

HERRAMIENTA SOFTWARE PARA EL DISEÑO ASISTIDO DE VÍAS

DIEGO ARMANDO VILLARREAL DÍAZ
SERGIO ANDRÉS HERNÁNDEZ CÁCERES

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECHANICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA

2010

HERRAMIENTA SOFTWARE PARA EL DISEÑO ASISTIDO DE VÍAS

DIEGO ARMANDO VILLARREAL DÍAZ
SERGIO ANDRÉS HERNÁNDEZ CÁCERES

Trabajo de grado para obtener el título de Ingeniero de Sistemas

DIRECTOR:

ALFONSO MENDOZA CASTELLANOS
PROFESOR ESCUELA DE INGENIERIA DE SISTEMAS E INFORMATICA - UIS

CODIRECTOR:

JORGE GOMEZ GOMEZ
PROFESOR ESCUELA DE INGENIERIA CIVIL - UIS

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECHANICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA

2010

*A mi Padre y Madre
fuentes de honestidad y deseo de
hacer las cosas lo mejor posible.*

*A mi Hermano que
me ha enseñado el valor
de mantener la calma
en los momentos difíciles.*

*A mi Nona por
su apoyo incondicional
en todo momento.*

*A Sergio Andrés
mi compañero de proyecto
por su perseverancia y profesionalismo.*

*Al Profesor Alfonso Mendoza
por permitirme formar parte
de este proyecto y el apoyo brindado.*

DIEGO ARMANDO VILLARREAL DÍAZ

*a mis padres, Carolina y Segundo
por ser un ejemplo de vida
y enseñarme el valor de la educación
no solo académica sino social y afectiva.*

*A mis hermanos, Rubén y Jenny
apoyo incondicional
en los momentos difíciles y ejemplo
de profesionales y estudiantes
con calidad y compromiso.*

*A todos mis compañeros, amigos y profesores
que creen en mis capacidades
porque me han motivado a dar más de mí.*

*A mis compañeros de proyecto y directores
Diego, Liz, Freddy, Jorge y Alfonso por
ser un equipo de trabajo cordial y comprensivo
con los cuales disfruté mucho al hacer este trabajo.*

SERGIO ANDRÉS HERNÁNDEZ CÁCERES

AGRADECIMIENTOS

ALFONSO MENDOZA CASTELLANOS

JORGE HERNANDO GÓMEZ GÓMEZ

GERMÁN GARCÍA VERA

ALVARO CASTELLANOS RIVERA

ALUMNOS DEL PROFESOR JORGE, GERMAN Y ALVARO

LIZ MAYDOLLY BARRERA ARDILA

FREDDY ALBERTO APARICIO SÁNCHEZ

JUAN CARLOS MANTILLA ROSAS

ANDRES LEONARDO CALPA GOMEZ

CARLOS GILBERTO MARTÍNEZ MARTÍNEZ

TABLA DE CONTENIDO

LISTA DE ILUSTRACIONES	12
INTRODUCCIÓN.....	19
1. DESCRIPCIÓN DEL PROYECTO	21
OBJETIVO GENERAL	22
OBJETIVOS ESPECÍFICOS	22
METODOLOGIA	23
XTREME PROPRAMMING (XP).....	24
ORGANIZACIÓN DEL LIBRO	26
2. MARCO TEÓRICO.....	27
2.1 DISEÑO VIAL	27
2.2 GENERACIÓN DE TERRENO.....	28
2.2.1 Levantamiento topográfico	28
2.2.2 Generación de Curvas De Nivel:.....	31
2.3 DISEÑO GEOMÉTRICO DE VÍAS	43
2.3.1 Criterios Diseño Geométrico de Vías	43
2.3.2 Diseño Planta	44
2.4- DISEÑO ASISTIDO POR COMPUTADORA (CAD).....	60
2.4.1 Definición	60
2.4.2 Antecedentes.....	60
2.4.3 Conceptos Básicos de una Herramienta CAD	61
2.4.4 Usando una Herramienta CAD	65
2.4.5 Ejemplos de Herramienta CAD	67
2.5- USO DE EXPRESIONES REGULARES	76
2.6 UML.....	80
2.6.1 Tipos de Diagrama	80
2.7 PATRONES DE DISEÑO.....	86
2.7.1 Objetivos de los patrones	87
2.7.2 Categorías de patrones	87
2.7.3 Descripción de los Patrones Usados en el proyecto.....	88
3. USO DE LA APLICACIÓN	103
3.1. DESARROLLO DE UN PROYECTO VIAL EN DIGRAVI	103
3.1.1. Generación de Terreno	104
3.1.2. Especificaciones de Proyecto.....	109
3.1.3. Diseño Horizontal	112
3.1.4. Transición de peralte	118
4 DESARROLLO DE LA APLICACIÓN	124
4.1. GENERACIÓN DE TERRENO.....	124
4.1.1. Análisis:	124
4.1.2. Requisitos:	126
4.1.3. Diseño:.....	127
4.1.4. Implementación:	138
4.2. ESPECIFICACIONES DE DISEÑO.....	142
4.2.1. Análisis:	142
4.2.2. Requisitos:	144
4.2.3. Diseño:.....	145
4.2.4. Implementación:	147
4.3. DISEÑO HORIZONTAL	148
4.3.1. Alineamiento Horizontal.....	149
4.3.2 Diseño de curvas horizontales	161
4.4. TRANCICIÓN DE PERALTE:	168

4.5. AYUDAS:	173
4.5.1 Análisis:	173
4.5.2 Requisitos:	174
4.5.3 Diseño:.....	174
4.4.4 Implementación:	176
5 LIBRERÍA MANEJO GRAFICOS	177
5.1 GUI EN JAVA.....	177
5.1.1 AWT	177
5.1.2 Swing	178
5.1.4 Guía para dibujar en Swing	180
5.1.5 Java2D.....	181
5.1.6 CADEMIA y Arquímedes	183
5.1.7 Diseño de la librería.....	184
6 TALLERES PRÁCTICOS	192
6.1 OBJETIVOS DEL TALLER.....	192
6.2 METODOLOGÍA	192
6.3 DOCUMENTO DEL TALLER.....	193
6.4 RESULTADO DE LOS TALLERES	193
6.4.1 Práctica generación de terreno, gráficador y GUI de la Aplicación.....	193
6.4.2 Practica especificaciones y ayudas.....	195
6.5 RECOMENDACIONES Y OBSERVACIONES	196
CONCLUSIONES.....	197
RECOMENDACIONES	198
REFERENCIAS BIBLIOGRÁFICAS.....	199

LISTA DE ILUSTRACIONES

ILUSTRACIÓN 1 LEVANTAMIENTO TOPOGRÁFICO	29
ILUSTRACIÓN 2 LEVANTAMIENTO DISTANCIAS FIJAS	29
ILUSTRACIÓN 3 CARTERA TOPOGRÁFICA DISTANCIAS FIJAS	29
ILUSTRACIÓN 4 LEVANTAMIENTO MÉTODO COTA REDONDA	30
ILUSTRACIÓN 5 EJE DE REFERENCIA	31
ILUSTRACIÓN 6 CARTERA COTA REDONDA	32
ILUSTRACIÓN 7 CARTERA DISTANCIAS FIJAS	33
ILUSTRACIÓN 8 PUNTOS CURVA DE NIVEL	36
ILUSTRACIÓN 9 CURVA DE NIVEL DEL TERRENO	39
ILUSTRACIÓN 10. ELEMENTOS CURVA CIRCULAR SIMPLE. TOMADO DE MANUAL DE DISEÑO GEOMÉTRICO DE CARRETERAS 2008	45
ILUSTRACIÓN 11. ELEMENTOS CURVA CIRCULAR COMPUESTA DE DOS RADIOS. TOMADO DE MANUAL DE DISEÑO GEOMÉTRICO DE CARRETERAS 2008	48
ILUSTRACIÓN 12. ELEMENTOS CURVA CIRCULAR COMPUESTA DE TRES RADIOS. TOMADO DE MANUAL DE DISEÑO GEOMÉTRICO DE CARRETERAS 2008	50
ILUSTRACIÓN 13. ELEMENTOS CURVA ESPIRAL-CIRCULAR-ESPIRAL, TOMADO DE MANUAL DE DISEÑO GEOMÉTRICO DE CARRETERAS 2008	53
ILUSTRACIÓN 14. ELEMENTOS CURVA ESPIRAL-ESPIRAL. TOMADO DE MANUAL DE DISEÑO GEOMÉTRICO DE CARRETERAS 2008	56
ILUSTRACIÓN 15. CALZADA GIRADA ALREDEDOR DEL EJE. TOMADO DE MANUAL DE DISEÑO GEOMÉTRICO DE CARRETERAS 2008	58
ILUSTRACIÓN 16. CALZADA GIRADA ALREDEDOR DEL BORDE INTERIOR. TOMADO DE MANUAL DE DISEÑO GEOMÉTRICO DE CARRETERAS 2008	59
ILUSTRACIÓN 17. CALZADA GIRA ALREDEDOR DEL BORDE EXTERNO. TOMADO DE MANUAL DE DISEÑO GEOMÉTRICO DE CARRETERAS 2008	59
ILUSTRACIÓN 18 DIBUJO CON CAPAS. TOMADO MANUAL REFERENCIA QCAD	61
ILUSTRACIÓN 19. TRES INSTANCIAS DE UN BLOQUE (INSERTS) CON DIFERENTES ATRIBUTOS, ÁNGULOS Y FACTOR DE ESCALA. TOMADO MANUAL REFERENCIA QCAD	62
ILUSTRACIÓN 20. COORDENADAS CARTESIANAS. TOMADO MANUAL REFERENCIA QCAD	63
ILUSTRACIÓN 21. COORDENADAS POLARES. TOMADO MANUAL REFERENCIA QCAD	63
ILUSTRACIÓN 22. DISEÑO 3D AUTOCAD. TOMADO DE CREATE A 3D ENVIRONMENT TO DRAW 3D MODELS 2010, DEL SITIO OFICIAL AUTOCAD	67
ILUSTRACIÓN 23. INTERFAZ DE USUARIO AUTOCAD. TOMADO DE GETTING STARTED WITH THE BASICS IN AUTOCAD, DEL SITIO OFICIAL AUTOCAD	68
ILUSTRACIÓN 24. DIBUJANDO EN AUTOCAD. TOMADO DE GETTING STARTED WITH THE BASICS IN AUTOCAD, DEL SITIO OFICIAL AUTOCAD	69
ILUSTRACIÓN 25. MANEJANDO CAPAS EN AUTOCAD. TOMADO DE MODIFY OBJECT PROPERTIES AND LAYERS IN AUTOCAD 2010 GETTING STARTED WITH THE BASICS IN AUTOCAD, DEL SITIO OFICIAL AUTOCAD	69
ILUSTRACIÓN 26. QCAD EN DIFERENTES PLATAFORMAS. IMAGEN MODIFICADA, ORIGINAL DEL SITIO OFICIAL QCADAUTOCAD	71
ILUSTRACIÓN 27. GUI QCAD. TOMADO DEL MANUAL DE REFERENCIA, DEL SITIO OFICIAL QCAD	72
ILUSTRACIÓN 28. GUI CADEMIA. TOMADO DEL SITIO OFICIAL CADEMIA	73
ILUSTRACIÓN 29. TRABAJANDO CON CADEMIA. TOMADO DEL SITIO OFICIAL CADEMIA	74
ILUSTRACIÓN 30 GUI ARCHÍMEDES. TOMADO DEL SITIO OFICIAL ARCHIMEDES	75
ILUSTRACIÓN 31. REPRESENTACIÓN PAQUETE UML	83
ILUSTRACIÓN 32. REPRESENTACIÓN CLASES DENTRO DE UN PAQUETE UML	83
ILUSTRACIÓN 33. INDICACIÓN DE CLASES DE OTROS PAQUETES	83
ILUSTRACIÓN 34. HERENCIA UML	84
ILUSTRACIÓN 35 ASOCIACION EN UML	84
ILUSTRACIÓN 36 ASOCIACION EN UN SOLO SENTIDO EN UML	85
ILUSTRACIÓN 37. EJEMPLO DE MULTIPLICIDAD	85
ILUSTRACIÓN 38. INTERFAZ EN UML	85
ILUSTRACIÓN 39. IMPLEMENTACIÓN DE UNA INTERFACE	86
ILUSTRACIÓN 40 DIAGRAMA PATRÓN ADAPTER	89
ILUSTRACIÓN 41. DIAGRAMA PATRÓN BRIDGE	91
ILUSTRACIÓN 42. DIAGRAMA PATRÓN COMPOSITE	93
ILUSTRACIÓN 43 ESTRUCTURA DEL PATRON FACADE. TOMADO DEL SITIO WEB HTTP://ES.WIKIPEDIA.ORG/WIKI/FACADE_(PATRÓN_DE_DISEÑO)	94
ILUSTRACIÓN 44 ESTRUCTURA PATRON OBSERVER. TOMADO DEL LIBRO PATRONES DE DESEÑO, GAMMA	95
ILUSTRACIÓN 45. ARQUITECTURA MVC	97
ILUSTRACIÓN 46. UNA APLICACIÓN JAVA SE EMPLEANDO MVC	97
ILUSTRACIÓN 47. MVC EN UML	98
ILUSTRACIÓN 48 PRESENTACIÓN DIGRAVI	103
ILUSTRACIÓN 49 DIAGRAMA DE FLUJO PROYECTO VIAL DIGRAVI	103
ILUSTRACIÓN 50 DIAGRAMA DE GENERACIÓN DE TERRENO DIGRAVI	104

ILUSTRACIÓN 51. MENÚ PARA LA GENERACIÓN DE TERRENO	105
ILUSTRACIÓN 52 PLANO DE EJEMPLO	106
ILUSTRACIÓN 53. CARGANDO UNA CARTERA EN FORMATO EXCEL	106
ILUSTRACIÓN 54. VISUALIZACIÓN DE UNA CARTERA CARGADA PREVIAMENTE	107
ILUSTRACIÓN 55. AYUDA SOBRE LA GENERACIÓN DE TERRENO.....	108
ILUSTRACIÓN 56. VISUALIZACIÓN DE LAS CURVAS DE NIVEL RESULTANTES DE CARTERA.....	109
ILUSTRACIÓN 57 ESPECIFICACIONES DE PROYECTO	109
ILUSTRACIÓN 58 DIAGRAMA ESPECIFICACIONES DE PROYECTO DIGRAVI	110
ILUSTRACIÓN 59 EJEMPLO DE VENTANA DE AYUDA.....	111
ILUSTRACIÓN 60 MENSAJE DE CONFIRMACIÓN.....	112
ILUSTRACIÓN 61 DISEÑO HORIZONTAL.....	113
ILUSTRACIÓN 62 PUNTOS FINALES	114
ILUSTRACIÓN 63 LINEA DE CEROS.....	114
ILUSTRACIÓN 64 TRAZADO DE PROYECTO	115
ILUSTRACIÓN 65 FIN ALINEAMIENTO	116
ILUSTRACIÓN 66 DIAGRAMA DISEÑO DE CURVAS HORIZONTALES DIGRAVI	116
ILUSTRACIÓN 67 DISEÑO DE CURVAS HORIZONTALES.....	117
ILUSTRACIÓN 68 CURVA CIRCULAR SINPLE	117
ILUSTRACIÓN 69 GRAFICA Y CARTERA	118
ILUSTRACIÓN 70 TRANSICIÓN DE PERALTE	118
ILUSTRACIÓN 71 INFORMACIÓN DE CURVA PARA TRANSICIÓN.....	119
ILUSTRACIÓN 72 TIPO DE TRANSICIÓN	119
ILUSTRACIÓN 73 DIAGRAMA TRANSICIÓN DE PERALTE DIGRAVI	120
ILUSTRACIÓN 74 TRANSICIÓN ALREDEDOR DEL EJE.....	121
ILUSTRACIÓN 75 OVNI TRANSICIÓN ALREDEDOR DEL EJE	121
ILUSTRACIÓN 76 TRANSICIÓN DEL BORDE INTERNO.....	122
ILUSTRACIÓN 77 OVNI TRANSICIÓN DEL BORDE INTERNO	122
ILUSTRACIÓN 78 TRANSICIÓN DEL BORDE EXTERNO	123
ILUSTRACIÓN 79 OVNI TRANSICIÓN DEL BORDE EXTERNO.....	123
ILUSTRACIÓN 80. CASO DE USO GENERACIÓN DE TERRENO	127
ILUSTRACIÓN 81. CARTERA DE DISTANCIA FIJAS.....	128
ILUSTRACIÓN 82. CARTERA DE COTA REDONDA.....	131
ILUSTRACIÓN 83. DIAGRAMA DE CLASE PUNTO	133
ILUSTRACIÓN 84. DIAGRAMA DE CLASE CARTERAVIAL	134
ILUSTRACIÓN 85. DIAGRAMA DE CLASE TRAZADODLAVIA Y EJEREFERENCIAL.....	135
ILUSTRACIÓN 86. DIAGRAMA DE CLASE LEEREJEDREFENCIA	136
ILUSTRACIÓN 87. DIAGRAMA DE CLASE EDITORES DE CARTERA	136
ILUSTRACIÓN 88. BOCETO UI	137
ILUSTRACIÓN 89. PAQUETE CARTERAS	138
ILUSTRACIÓN 90. PAQUETE CLASEBASE	138
ILUSTRACIÓN 91 PAQUETE IO.....	139
ILUSTRACIÓN 92. PAQUETE INTERFACE	139
ILUSTRACIÓN 93. PAQUETE AYUDAS	139
ILUSTRACIÓN 94. PAQUETE ELEMENTOVIAL	140
ILUSTRACIÓN 95. PAQUETE EXPREGULARES	140
ILUSTRACIÓN 96. PAQUETE PROYECTOVIAL.....	140
ILUSTRACIÓN 97. PAQUETE VISUALIZACIÓN	141
ILUSTRACIÓN 98. PAQUETE ÁRBOLDPROYECTO.....	141
ILUSTRACIÓN 99. PAQUETE COMPONENTESGRAFICOS	142
ILUSTRACIÓN 100. CASO DE USO ESPECIFICACIONES DE DISEÑO	145
ILUSTRACIÓN 101. DIAGRAMA DE CLASE DE ESPDPROYECTO	146
ILUSTRACIÓN 102 BOCETO ESPECIFICACIONES	147
ILUSTRACIÓN 103 PAQUETE AYUDAS	147
ILUSTRACIÓN 104 PAQUETE EXPREGULARES	147
ILUSTRACIÓN 105 PROYECTOVIAL	148
ILUSTRACIÓN 106 PAQUETE ÁRBOLDPROYECTO.....	148
ILUSTRACIÓN 107. CASO DE USO DISEÑO HORIZONTAL	153
ILUSTRACIÓN 108. DIAGRAMA DE CLASE DISEÑO HORIZONTAL	154
ILUSTRACIÓN 109 INSTRUCCIÓN Y CONTROLADOR	155
ILUSTRACIÓN 110 COMANDO ALTURAS	156
ILUSTRACIÓN 111 COMANDO INICIO FIN VIAL	157
ILUSTRACIÓN 112 COMANDO LINEA DE CEROS	157
ILUSTRACIÓN 113 COMANDO TRAZADO DE PROYECTO.....	158
ILUSTRACIÓN 114 INTERFAZ DE USUARIO ALINEAMIENTO HORIZONTAL	159
ILUSTRACIÓN 115 PAQUETE CARTERAS	159
ILUSTRACIÓN 116 PAQUETE HERRAMIENTA GRÁFICA	160
ILUSTRACIÓN 117 PAQUETE INSTRUCCIONES	160
ILUSTRACIÓN 118 PAQUETE PROYECTOVIAL	160
ILUSTRACIÓN 119 PAQUETE VISUALIZACIÓN	160
ILUSTRACIÓN 120 PAQUETE DISEÑO HORIZONTAL.....	161

ILUSTRACIÓN 121 DIAGRAMA DE CASOS DE USO DISEÑO DE CURVAS HORIZONTALES	163
ILUSTRACIÓN 122 CURVA HORIZONTAL	163
ILUSTRACIÓN 123 CURVA CIRCULAR SIMPLE.....	164
ILUSTRACIÓN 124 CURVAS COMPUESTAS 2 Y 3 RADIOS	165
ILUSTRACIÓN 125 CURVAS ESPIRALIZADAS.....	166
ILUSTRACIÓN 126 PAQUETE CARTERAS	166
ILUSTRACIÓN 127 PAQUETE CURVA HORIZONTAL.....	167
ILUSTRACIÓN 128 PAQUETE PROYECTO VIAL.....	167
ILUSTRACIÓN 129PAQUETE DISEÑO HORIZONTAL.....	168
ILUSTRACIÓN 130 CASO DE USO TRANSICIÓN DE PERALTE	169
ILUSTRACIÓN 131 DIAGRAMA DE CLASES PTRANSICIONPERALTE	170
ILUSTRACIÓN 132 DIAGRAMA DE CLASES BUILDING UTILIDADES	171
ILUSTRACIÓN 133 BOCETO TRANSICIÓN DE PERALTE	171
ILUSTRACIÓN 134 PAQUETE TRANSICIÓN DE PERALTE	172
ILUSTRACIÓN 135 PAQUETE PROYECTO VIAL TRANSICIÓN	172
ILUSTRACIÓN 136 PAQUETE VISUALIZACIÓN TRANSICIÓN.....	172
ILUSTRACIÓN 137 PAQUETE TRANSICIÓNDEPERALTE	173
ILUSTRACIÓN 138 DIAGRAMA DE CLASE AYUDAS	175
ILUSTRACIÓN 139 MÉTODOS CLASE VISUALIZADORAYUDA.....	175
ILUSTRACIÓN 140. PAQUETE AYUDAS.....	176
ILUSTRACIÓN 141. DIAGRAMA CLASES DE LAS ENTIDADES.....	185
ILUSTRACIÓN 142. DIFERENTES IMPLEMENTACIONES DE ELEMENTO	185
ILUSTRACIÓN 143. DIAGRAMA DE CLASE PARA LA CAPA.....	186
ILUSTRACIÓN 144. DIAGRAMA DE CLASE DE DIBUJO.....	187
ILUSTRACIÓN 145. DIAGRAMA DE CLASE DEL LIENZO	188
ILUSTRACIÓN 146. DIAGRAMA DE CLASE DEL CONTROLADORINSTRUCCION.....	189
ILUSTRACIÓN 147. DIAGRAMA DE CLASE DE INSTRUCCIÓN.....	190
ILUSTRACIÓN 148. ALGUNAS IMPLEMENTACIONES DE INSTRUCCIÓN PARA GENERAR ENTIDADES	191
ILUSTRACIÓN 149. ALGUNAS IMPLEMENTACIONES DE INSTRUCCIÓN PARA MANIPULAR LIENZO.....	191
ILUSTRACIÓN 150 TALLERES DE DIGRAVI.....	193

LISTA DE TABLAS

TABLA 1. PERALTE MÁXIMO	57
TABLA 2. FUNCIONES COMUNES HERRAMIENTA CAD	65
TABLA 3. VISIBILIDAD CLASE	81
TABLA 4. MULTIPLICIDAD DE CLASE	85
TABLA 5 VALORES EJEMPLO ESPECIFICACIONES DE PROYECTO	111
TABLA 6 TRANSICIÓN ALREDEDOR DEL EJE.....	121
TABLA 7 TRANSICIÓN DEL BORDE INTERNO.....	122
TABLA 8 TRANSICIÓN DEL BORDE EXTERNO	123
TABLA 9 POSIBLES VALORES QUE PUEDE TOMAR EL CAMPO ABSCISA	129
TABLA 10 POSIBLES VALORES QUE PUEDE TOMAR EL CAMPO DEFELXION.....	130
TABLA 11. PRO Y CONTRA AWT	179
TABLA 12. PRO Y CONTRA SWING	179
TABLA 13 TABLA CON LOS METODOS GRAPHICS2D PARA REALZAR FORMAS.....	182
TABLA 14. ATRIBUTOS CAPA	186

TABLA DE ANEXOS

ANEXO 1 CASOS DE USO	200
ANEXO 2 CARTERAS	214
ANEXO 3 TALLERES	221
ANEXO 4 PROPUESTA INTERFAZ GRÁFICA.....	234

RESUMEN

TÍTULO:

Herramienta software para el diseño asistido de vías ¹

AUTORES:

Sergio Andrés Hernández Cáceres

Diego Armando Villarreal Díaz²

PALABRAS CLAVES:

Diseño, Geométrico, CAD, GEOM, Topografía, Simulador, Terreno, Educacional.

DESCRIPCIÓN:

DIGRAVI (Diseño gráfico de Vías) es una herramienta de diseño asistido por computadora para el diseño geométrico de carreteras, elaborado como tesis de grado en la Universidad Industrial de Santander. DIGRAVI ha sido desarrollado por estudiantes de la escuela de Ingeniería de Sistemas de la UIS, y está inspirado en las metodologías de enseñanza de las áreas de topografía y vías de la escuela de Ingeniería Civil de la misma universidad, haciendo uso del conocimiento de los profesores y estudiantes de estas áreas.

La aplicación permite la digitalización de terrenos por medio de carteras topográficas, elaboradas siguiendo los métodos de distancias fijas y cota redonda, dichas carteras son ingresadas haciendo uso de un editor de hojas de cálculo como Excel o el editor interno de la aplicación, el cuál está diseñado para recibir y validar los datos de cartera en la notación que se usa regularmente en un levantamiento topográfico.

A partir de la cartera ingresada puede generarse un plano de curvas de nivel del terreno, Adicionalmente se pueden definir las especificaciones de la vía según las normativas GEOM (Manual de Diseño Geométrico de Carreteras del INVIAS) versión 2008 y realizar las fases de Alineamiento horizontal, diseño de curvas Horizontales y Transición de peralte.

¹ Trabajo de Grado en la Modalidad de Investigación

² Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas e Informática.

SUMMARY

TITLE:

SOFTWARE TOOL FOR AIDED ROUTE DESIGN³

AUTHORS:

Sergio Andrés Hernandez Caceres

Diego Armando Villarreal Diaz⁴

KEYWORDS:

Geometric, Design, CAD, GEOM, Topography, Simulator, Field, Educational.

ABSTRACT:

DIGRAVI (Graphical Routes Design) is a computer-aided designer tool for road geometric design, elaborated as a thesis for the “Universidad Industrial de Santander” UIS. DIGRAVI has been developed by students of the Systems Engineering’s school of UIS, and It is inspired in educational methodologies such as topography and routes, from the Civil Engineering’s school at the same university, using as theoretical support teacher’s knowledge and student’s experiences in these areas.

The application allows the digitalization of areas through topographic portfolios, which are elaborated following the fixed-distances method and round-Heights method, the input portfolios could be make trough the use of an excel’s spreadsheet or the application’s editor which was design to validate the input portfolio using a portfolio standard notation.

From these portfolios, the application generates a plane containing a set of contour lines for a specific area. The application allows the definition of a route according to the regulations of GEOM (Manual of Geometric Design of Roads of the INVIAS), version 2008, and the realization of the phases of horizontal Alignment, horizontal curve Design and banking Transition.

³ Graduation paper, Research mode

⁴ Faculty of Physical-Mechanical Engineering. School of Engineering and Computer Science.

INTRODUCCIÓN

La creación de nuevas vías de tránsito terrestre en nuestro país es de gran importancia, puesto que estas vías influyen de manera enorme en el desarrollo de las comunidades que hacen uso de ellas. En lo que respecta a las actividades de transporte, turismo, el acceso a los recursos, y la venta de los productos que la población genera, se hacen imposibles sin la construcción de carreteras que le permitan a la población abrirse a las demás comunidades.

Ante esta fuerte necesidad, la Universidad Industrial de Santander (UIS) prepara a sus Ingenieros Civiles con el conocimiento para desarrollar estas obras, conforme a las necesidades que tiene nuestra población Colombiana.

En el proceso de enseñanza, nuestra universidad, se ve la necesidad de usar herramientas con enfoque educacional, que le permitan al estudiante afianzar sus conceptos y poner a prueba sus conocimientos en el diseño de una vía. Con el ánimo de responder a esta necesidad, se plantea este proyecto de grado, el cual ofrece una herramienta diseñada con base a la metodología de enseñanza utilizada en la universidad.

ESPECIFICACIONES DEL PROYECTO

TITULO:

HERRAMIENTA SOFTWARE PARA EL DISEÑO ASISTIDO DE VÍAS

DIRECCIÓN:

DIRECTOR: Profesor Escuela Ingeniería de Sistemas. ALFONSO MENDOZA CASTELLANOS

Director Grupo de Investigación en ingeniería Biomédica (GIIB)

CODIRECTOR: Profesor Escuela Ingeniería Civil. JORGE GOMEZ GOMEZ

AUTORES:

DIEGO ARMANDO VILLARREAL DÍAZ

SERGIO ANDRÉS HERNÁNDEZ CÁCERES

ENTIDADES INTERESADAS EN EL PROYECTO:

Escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander

Escuela de Ingeniería Civil de la Universidad Industrial de Santander

Grupo de Investigación en Ingeniería Biomédica (GIIB)

1. DESCRIPCIÓN DEL PROYECTO

La escuela de ingeniería civil en su proceso de enseñanza del área de topografía y vías, hace uso de diferentes herramientas software para la representación de terrenos y la realización del diseño de vías de comunicación. Entre estas herramientas se encuentran Autocad, Hedica y Excel, las cuales han sido de gran utilidad para el proceso de enseñanza; sin embargo Autocad y Excel son herramientas de propósitos generales y no orientados a la educación y si bien es cierto que Hedica posee un enfoque pedagógico, han surgido nuevas necesidades que no satisface. El proceso de generación de terreno requiere de la realización de una serie de interpolaciones sobre una cartera obtenida del terreno, este proceso se realiza manualmente por lo cual está sujeto a múltiples errores y un gasto excesivo de tiempo, tiempo que podría ser usado para el afianzamiento de los conceptos. Para la visualización del terreno se emplea Autocad o se dibuja sobre un plano físico. La utilización de Autocad es cómoda pero no tiene un enfoque educacional, y el dibujo manual sí es pedagógico pero consume mucho tiempo. Por otro lado, en el área de vías comienzan el proceso con un plano físico el cual calcan en Autocad, o en su defecto lo consiguen ya digitalizado; Autocad a pesar de ser programa de gran utilidad, posee una licencia de gran costo para la universidad, esta metodología no le permite al estudiante afianzar sus conceptos de topografía.

Se cuenta también con el programa Hedica, un desarrollo propio de la universidad, hecho como proyecto de grado en 2004, el cual tiene un enfoque educativo, y permite calcular los elementos viales requeridos en la asignatura de vías II, pero también con el inconveniente de que estos elementos no vienen integrados a un proyecto vial, sino que se calculan por separado y deben luego ser graficados en Autocad 2004. Adicional a esto Hedica presenta problemas de incompatibilidad que impiden su instalación en algunos ordenadores, y la versión Autocad 2004 se ha ido descontinuuando del mercado. Una solución posible a los problemas mencionados sería modificar Hedica para contemplar estos casos, la cual no fue posible realizar por que no se contaba con documentación ni código fuente de

Hedica. El diseño de las vías de comunicación debe acogerse siempre a las normativas colombianas que se estipulan como parámetros de seguridad y comodidad en el diseño. Las normativas GEOM fueron modificadas en 2008 y han sido publicadas en 2009, por lo cual las herramientas que usan los estudiantes actualmente no están actualizadas a estas normativas.

OBJETIVO GENERAL

Construir una herramienta software que satisfaga las necesidades básicas del diseño vial, de acuerdo a los requerimientos obtenidos con la colaboración de estudiantes y profesores de la escuela de ingeniería civil, actualizado a las normativas colombianas GEOM 2008. Planteado como proyecto base para el desarrollo subsecuente de varios módulos en el campo del diseño vial.

OBJETIVOS ESPECÍFICOS

- Documentar los procesos del diseño vial.
- Generar documentos con las especificaciones de requisitos.
- Integrar a la interfaz de usuario ayudas, tanto de uso de la aplicación así como información concerniente al diseño vial.
- Desarrollar la aplicación, de tal forma que permita su uso en distintos sistemas operativos.
- Asistir la generación de terreno, especificaciones de planta y peralte, diseño horizontal, y transversal de una vía.
- Diseñar y realizar talleres de socialización de la aplicación con los estudiantes de las asignaturas topografía y diseño vial.

METODOLOGIA

Los conocimientos necesarios para diseñar una vía son aportados por dos estudiantes de civil, los cuales trabajan conjuntamente bajo la supervisión del codirector de proyecto Jorge Hernando Gómez Gómez, profesor de la escuela de ingeniería civil, de acuerdo a la metodología con que actualmente se enseña a los estudiantes de ingeniería Civil de la UIS en la materia “Vías II” para la cual será utilizado el software como herramienta pedagógica. Las especificaciones de requisitos del software son debatidas por el grupo de estudiantes, desarrolladores, colaboradores y docentes para ser aprobadas por el Director y Codirector del proyecto. Este estudio de requisitos se ve plasmado en diagramas UML, y en los manuales de ayuda y operación del software que son entregados con el mismo. La implementación del Software es realizada por los mismos desarrolladores de este proyecto: Diego Armando Villareal Díaz Y Sergio Andrés Hernández Cáceres, en el lenguaje de programación Java.

Para la realización del proyecto, se plantea una continua interacción de las etapas: marco teórico, especificación, implementación y prueba, lo que implica reuniones frecuentes con los dos estudiantes de civil y los directores, contando también con talleres de prueba, dirigidos a los estudiantes de Ingeniería Civil, quienes son los usuarios finales de la aplicación. Se hará uso de la metodología Xtreme programming dado que permite de una alta interacción y continua revisión de las actividades desarrolladas, se requiere de la vinculación de los conocedores de la temática en el área de vías, así mismo se comienza el proyecto sabiendo que debe amoldarse a las nuevas normativas GEOM 2008 a publicarse en 2009. Xtreme Programming está diseñado para proyectos con requisitos cambiantes, y alta interacción con el usuario, por lo cual se considera que la metodología antes mencionada que se ajusta convenientemente a nuestras necesidades.

XTREME PROPRAMMING (XP)

Xp es una metodología de desarrollo de software cuyo objeto es el mejorar la calidad del software y la respuesta ante cambios de en los requerimientos de los clientes, entre los mejores atributos de Xp son:

- Simplicidad
- Comunicación
- Programación por parejas
- Retroalimentación
- Coraje
- Respeto

En Xp se realizan entregas en cortos ciclos de desarrollo, con esto se busca el crear espacios para la adición de nuevos requisitos por parte del cliente. Algunos elementos frecuentes en Xp son:

- Programar en parejas
- Realizar pruebas frecuentes sobre todo el código
- Simplicidad y claridad en el código
- Comunicación frecuente entre el cliente y grupo de desarrolladores El aspecto más sobresaliente de Xp es su simplicidad en las reglas.

Xp es similar a un rompecabezas: Hay muchas piezas. Individualmente las piezas no tienen mucho sentido, pero cuando se combinan se forma una figura completa. Las reglas pueden parecer en primer lugar poco elegante e incluso despreocupadas, pero se fundamentan valores y principios sólidos.

Fases de XP:

- Planificar
- Diseñar
- Codificar

- Probar

La programación extrema proporciona una serie de ventajas a la hora de afrontar ciertos proyectos, que deben ser tenidas muy en cuenta, porque el uso de la XP en ciertos contextos permite el ahorro de tiempo y recursos.

Ventajas y Desventajas

- El productos resultantes son usables con mayor rapidez.
- El proceso de integración es continuo, por lo que el esfuerzo final para la integración es nulo.
- Se consigue integrar todo el trabajo con mayor facilidad.
- Se atienden las necesidades del usuario con mayor exactitud. Esto se logra gracias a las continuas versiones que se ofrecen al usuario.
- Los productos son más fiables y robustos contra los fallos gracias al diseño de los test de forma previa a la codificación.
- Se obtiene código más simple y más fácil de entender, reduciendo el número de errores.
- Gracias a la filosofía del “pair programming” (programación en parejas), se consigue que los desarrolladores apliquen las buenas prácticas que se les ofrecen con la metodología XP.
- Gracias al “refactoring”[1] es más fácil modificar los requerimientos del usuario.
- El contemplar nuevos requerimientos en la marcha del proyecto, puede implicar hacer cambios sobre faces que ya han sido terminadas; esto conlleva a retomar trabajo anterior y una demora sustancial en el desarrollo del proyecto. (por esta razón es de gran importancia que el cliente tenga muy claro los requisitos que debe cumplir el software y se definan tempranamente de la forma más concreta y formal posible).
- El mezclar grupos interdisciplinarios en el trabajo de desarrollo vuelve más compleja la labor de coordinación del trabajo, puesto que para avanzar se hace necesaria la disposición de mas personas.

ORGANIZACIÓN DEL LIBRO

Este libro se organiza de la siguiente forma:

- El capítulo 2 describe las temáticas que fueron necesarias para el desarrollo del proyecto, y que forman las bases de conocimiento sobre las cuales se desarrolla el producto entregable. Tales temas abarcan el proceso de construcción de una vía, herramientas de diseño asistido por computadora (CAD), expresiones regulares, patrones de diseño y UML
- El capítulo 3 describe el funcionamiento de Digravi, con el desarrollo de un proyecto de ejemplo.
- El capítulo 4 comprende el proceso de desarrollo de la aplicación desde la especificación de requisitos hasta el proceso de implementación.
- El capítulo 5 describe como se realizó el desarrollo de la librería gráfica usada por Digravi.
- El capítulo 6 Pruebas de la aplicación con los estudiantes (Talleres), describe el momento de las pruebas que se hicieron con los estudiantes para asegurar la buena calidad del software, y encontrar posibles errores.

2. MARCO TEÓRICO

2.1 DISEÑO VIAL

El desarrollo de un proyecto de diseño vial implica un alto nivel de interacción con la comunidad que será beneficiaria del proyecto, con el objetivo de conocer a fondo sus necesidades y de esta forma lograr un buen balance entre:

- Seguridad
- Movilidad
- Comodidad
- Estética
- Economía
- Compatibilidad con el medio ambiente

Una vía se define como una infraestructura del transporte especialmente acondicionada dentro de una franja de terreno denominada derecho de vía, con el propósito de permitir la circulación de vehículos de manera continua en el espacio y el tiempo, con niveles adecuados de seguridad y comodidad.

El desarrollo de un proyecto vial conlleva la realización de tres fases:

Fase 1. Pre-factibilidad: se obtienen los primeros datos de información y documentación de la zona, así como un reconocimiento previo del terreno a nivel topográfico y unos cálculos preliminares que permitan brindar recomendaciones en las fases posteriores. El objetivo de esta fase es evaluar si el proyecto es viable económicamente.

Fase 2. Factibilidad: se realiza un análisis más profundo de todo el conjunto de diseño de la carretera, el cual tiene un fuerte enfoque técnico; por ejemplo se realizan levantamientos de precisión topográfica de las alternativas contempladas, que luego permitan dibujar el trazado tanto en planta como en perfil de la rasante a desarrollar finalmente. Con estos datos se hacen también unas primeras aproximaciones en el cálculo de volúmenes, movimientos de tierra, construcción de obras de arte, de drenaje o puentes, estimándose sus cantidades

de obra y dándose unas aproximaciones mínimas aceptables. El objetivo de esta fase es definir si el proyecto es rentable a partir de datos más detallados.

Fase 3. Diseños definitivos: en esta fase se escoge la alternativa que finalmente va a cumplir con todos los requerimientos del proyecto, siendo integrada entonces por: un diseño geométrico completo, cantidades de materiales y costos de la obra, especificaciones de la construcción, un presupuesto y un pliego de condiciones dado para la licitación. Además, con ella se presentan todos los planos de la propuesta (localización, planta, perfil, secciones transversales, diagrama de masas y detalles de obras adicionales), informes completos del proyecto tanto del diseño como del modo de construcción. El objetivo de esta fase es el de elaborar los diseños detallados que permitan construir la vía.

2.2 GENERACIÓN DE TERRENO

La generación del terreno ocupa la actividad de descripción del terreno en cuanto a sus alturas, ubicación y orientación. A partir de esta descripción es posible diseñar una vía de acuerdo a la disponibilidad del terreno para realizar la obra. Para este fin se realiza un levantamiento topográfico y los datos generados en éste son ingresados en un modelador digital el cual genera una serie de curvas de nivel que representan el terreno. Sobre éstas curvas siguiendo las normativas vigentes se procede a realizar el diseño de la vía.

2.2.1 Levantamiento topográfico

Para ubicar un objeto es necesario contextualizarlo en un terreno, del cual se deben saber tanto sus distancias horizontales como verticales. La forma de conocer estos datos es a través de la práctica, mediante levantamientos en campo con la ayuda de instrumentos diseñados para apoyar estos trabajos tales como niveles, miras, estaciones, plomadas, GPS o tamanuas.

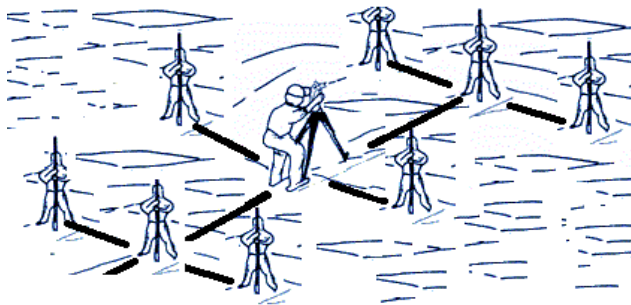


Todos los datos tomados allí se consignan en unas bitácoras denominadas carteras, las cuales varían de acuerdo al tipo de dato a detallar y se llenan de abajo hacia arriba en primera instancia y posteriormente en el procesamiento de cálculos en oficina se organizan en orden contrario. Para el diseño y construcción de una vía (de manera pedagógica), **la Comisión**

ILUSTRACIÓN 1 LEVANTAMIENTO TOPOGRÁFICO

topográfica⁵ une un punto inicial y final mediante la ubicación de un alineamiento que se traza de acuerdo a la pendiente media máxima, dada por el nivel de mano (Abney), siendo el trazado segmentado longitudinalmente procurando conservar una distancia determinada (tomando cada punto el nombre y valor de una abscisa en el sentido de avance), en la cual se toman datos transversalmente de alturas y distancias. Este proceso se puede ejecutar mediante los métodos de distancia fija o cotas redondas.

2.2.1.1 Método de distancias fijas:



Este método toma una distancia fija en el sentido transversal de cada abscisa y se le toman sus respectivas alturas con la ayuda de una mira y el nivel de precisión, dejando que en oficina se procesen cálculos que permitan hallar cotas redondas con interpolaciones para finalmente unir puntos de un

ILUSTRACIÓN 2 LEVANTAMIENTO DISTANCIAS FIJAS

mismo valor que le den forma a las curvas de nivel en un plano. El registro realizado de los datos que se toman en un levantamiento topográfico con cartera de distancias fijas es el siguiente:

K0+371.83	-70°57'09"	1662.16	1660.40	1658.60	1656.68	1654.86	1653.33	1652.00	1650.69	1649.57	1648.66	1647.76	1646.90	1646.03
K0+360		1659.40	1657.91	1656.41	1654.91	1653.38	1651.83	1650.29	1648.86	1647.47	1646.11	1644.80	1643.49	1642.18
K0+340		1656.71	1655.30	1653.95	1652.49	1651.15	1649.87	1648.51	1647.14	1645.75	1644.31	1642.87	1641.44	1640.00
K0+320		1654.39	1653.00	1651.57	1650.14	1648.72	1647.30	1645.87	1644.48	1643.09	1641.73	1640.44	1639.17	1637.89
K0+300		1652.50	1651.10	1649.70	1648.31	1646.86	1645.40	1643.93	1642.50	1641.04	1639.60	1638.16	1636.73	1635.29
K0+280		1651.70	1650.58	1649.33	1647.90	1646.48	1645.02	1643.55	1642.08	1640.61	1639.10	1637.59	1636.15	1634.67
K0+260		1650.44	1649.06	1647.62	1646.18	1644.75	1643.33	1641.89	1640.49	1639.04	1637.60	1636.18	1634.78	1633.34
K0+240		1650.00	1648.52	1647.05	1645.59	1644.11	1642.62	1641.13	1639.67	1638.17	1636.79	1635.22	1633.74	1632.26
K0+220		1649.10	1647.62	1646.15	1644.68	1643.20	1641.73	1640.26	1638.79	1637.32	1635.86	1634.38	1632.91	1631.44
K0+212.72	-41°04'59"	1649.03	1647.57	1646.06	1644.54	1643.03	1641.52	1640.00	1638.48	1637.00	1635.52	1634.05	1632.57	1631.10
K0+200		1646.79	1645.22	1643.65	1642.10	1640.57	1639.13	1637.73	1636.41	1635.25	1634.16	1633.19	1632.23	1631.27
K0+180		1644.06	1642.60	1641.19	1639.80	1638.58	1637.02	1635.64	1634.25	1632.72	1631.11	1629.48	1627.94	1626.81
K0+160		1643.28	1641.83	1640.36	1638.88	1637.40	1635.92	1634.44	1632.96	1631.48	1630.00	1628.52	1627.05	1625.13
K0+140		1642.16	1640.66	1639.16	1637.66	1636.18	1634.74	1633.33	1631.89	1630.43	1628.98	1627.54	1626.10	1624.08
K0+120		1642.19	1640.70	1639.21	1637.72	1636.24	1634.75	1633.26	1631.77	1630.28	1628.79	1627.31	1625.67	RIO
K0+100		1642.58	1641.09	1639.60	1638.12	1636.63	1635.14	1633.65	1632.16	1630.67	1629.18	1627.69	1626.21	RIO
K0+80		1641.77	1640.32	1638.95	1637.55	1636.14	1634.70	1633.42	1631.78	1630.32	1628.86	1627.40	1625.92	RIO
K0+60		1641.23	1639.77	1638.31	1636.86	1635.40	1633.94	1632.48	1631.00	1629.52	1628.03	1626.54	1624.11	RIO
K0+40		1639.02	1637.77	1636.51	1635.25	1634.00	1632.64	1631.25	1629.83	1628.41	1626.99	1625.59	1624.22	RIO
K0+20		1635.42	1634.39	1633.25	1632.04	1630.78	1629.53	1628.17	1627.01	1625.67	1624.00	RIO	RIO	RIO
K0+00		1630.77	1630.02	1629.06	1628.04	1627.05	1626.05	1624.94	RIO	RIO	RIO	RIO	RIO	RIO
ABSCISA	DEFLEXIÓN	30	25	20	15	10	5	EJE	5	10	15	20	25	30
		IZQUIERDA						CENTRAL	DERECHA					

ILUSTRACIÓN 3 CARTERA TOPOGRÁFICA DISTANCIAS FIJAS

⁵ Una comisión topográfica es el grupo de personas que realizan el trabajo de campo (levantamiento topográfico)

2.2.1.2 Método de cota redonda:

Cotas redondas toma a través de la interpolación en campo alturas cerradas y se registra la distancia a la que fue ubicada, apoyándose igualmente en la mira y el nivel de precisión para este trabajo. Con ello, al llegar a la oficina se plasman de una vez las



ILUSTRACIÓN 4 LEVANTAMIENTO MÉTODO COTA REDONDA

curvas de nivel que grafican un plano. A continuación se observa una cartera de cotas fijas, allí observamos que en cada abscisa hay dos filas, ya que en la primera fila es necesario registrar la distancia, que varía para cada una, y en la segunda la respectiva cota.

Ko+180					13.91	10.43	0	0.53	1.88	9.49	13.36				
					1009	1010	1010.379	1011	1012						
Ko+160	11.33'				14.78	9.26	0	3.68	9.87						
					1008	1009	1009.35	1010	1011						
Ko+140					12.48	3.01	0	4.16	10.37						
					1005	1006	1007.914	1007	1008	1009	1010				
Ko+120	-14.6'				20.13	6.253	0	1.24	9.947	12.88	15.829				
					1005	1006	1006.938	1007	1008	1009	1010	1010	1009		
Ko+100					12.608	3.452	0	0.155	2.655	5.155	7.655	8.649	13.55		
					1004	1005	1005.716	1006	1007	1008					
Ko+080					11.66	3.741	0	2.84	9.76	13.396					
					1003	1004	1004.119	1005	1006	1006	1005				
Ko+060	-11'				11.002	3.741	0	2.565	7.898	8.105	13.62				
					1003	1002	1002.757	1003	1004	1005					
Ko+040					8.24	8.09	0	1.55	7.95	14.11					
					1003	1002	1001.514	1001	1000						
Ko+020					16.4	9.44	0	2.076	6.117						
					1002	1001	1000	999.967	999						
KO+00					12.212	7.003	0.224	EJE	11.817						
ABSCISA	DEFLEXION	IZQUIERDA				CENTRAL			DERECHA						

Después de implementar estos trabajos en la práctica se procesa todo y grafica en un plano uniendo puntos de igual altura, permitiendo así bosquejar la realidad del terreno sobre el cual se va a proyectar una vía.

Posteriormente y con el diseño geométrico se localizan puntos mediante coordenadas y alturas para la construcción de la carretera y los cálculos de movimientos de tierra que ayudan a plantear el costo de la obra.

En la actualidad mediante equipos sofisticados de tecnología como el GPS se puede determinar con mayor prontitud los datos de un terreno y apoyados en programas avanzados que traen los equipos ubicar puntos en un plano del mismo. Sin embargo, son cosas que no se consiguen en el mercado de manera fácil y que no se manejan sin el debido conocimiento del procesamiento en la práctica.

2.2.2 Generación de Curvas De Nivel:

El proceso de elaboración de las curvas de nivel del consta de tres actividades a saber: la primera de ellas es la transformación de las notaciones usadas en el levantamiento de terreno tales como abscisas cotas y deflexiones, a un sistema de coordenadas rectangulares (x, y, z), lo que facilita su manejo y posterior representación gráfica. La segunda actividad abarca el proceso de interpolación de datos en búsqueda de la ubicación de las alturas requeridas para realizar las curvas de nivel. Y como tercero es necesario tomar el conjunto de puntos obtenidos de las dos fases anteriores y agruparlos los puntos con la misma altura, estos se organizan lógicamente para que su unión sucesiva genere cada curva de nivel del terreno sin cruzarse con otras curvas ni unirse sobre áreas sin información.

2.2.2.1 Transformación entre sistemas de coordenadas

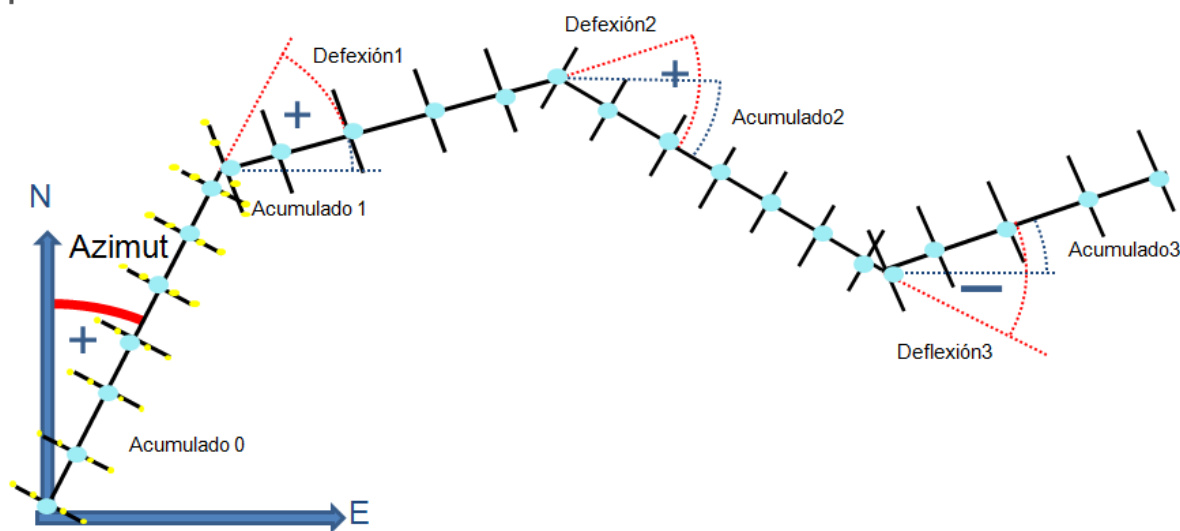


ILUSTRACIÓN 5 EJE DE REFERENCIA

Primero es necesario definir Los componentes básicos de la cartera de entrada: encontramos el eje de referencia marcado en azul es un conjunto ordenado de puntos, cada uno de estos puntos se compone de una abscisa una cota y una deflexión transversalmente a cada punto del eje se encuentra una sección transversal a izquierda y derecha, cada sección transversal está conformada también por puntos, la agrupación de estos se denomina cartera, cada una de las carteras está compuesta por puntos.

Cartera: Agrupación de Puntos topográficos ordenados, estos puntos indican una posición en el espacio y se definen en función de los otros, es decir son dependientes.

Punto: Un punto es una representación de un lugar físico, su ubicación es dependiente de sus valores y del punto anterior, se compone de Abscisa, Cota y deflexión.

Abscisa: Es la distancia recorrida comenzando por el primer punto de la cartera a través de todos los puntos de la de la misma hasta llegar al punto al que pertenece, es decir la distancia acumulada desde el comienzo de la cartera.

Cota: Es la altura a la cual se encuentra el terreno en ese mismo punto, es la única coordenada que no necesita de una transformación especial.

Deflexión: es un cambio de la dirección del eje de referencia marcado en ese punto, la deflexión es un ángulo, se mide en grados o en radianes en sentido horario desde la proyección en el sentido anterior del eje de referencia hacia el nuevo sentido del mismo (ver Imagen).

La organización de estos puntos puede variar dependiendo del método de levantamiento topográfico como se muestra en las graficas:

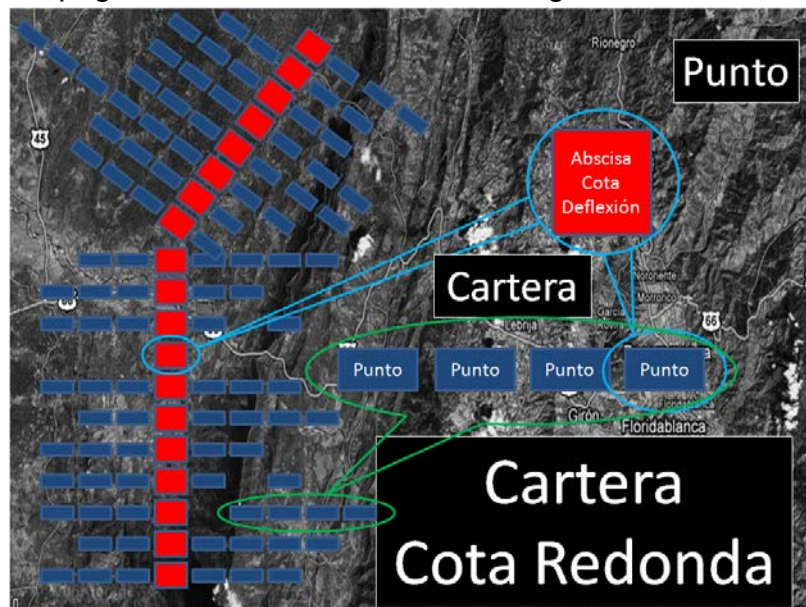


ILUSTRACIÓN 6 CARTERA COTA REDONDA

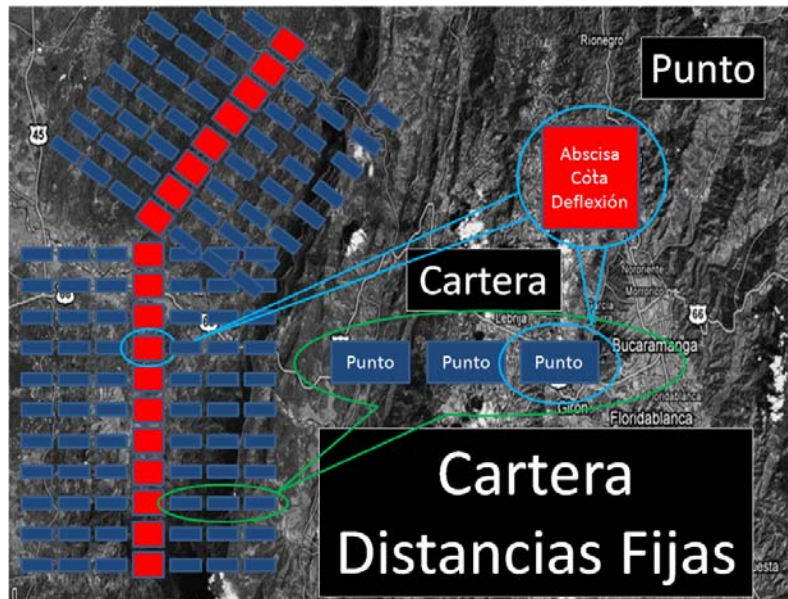


ILUSTRACIÓN 7 CARTERA DISTANCIAS FIJAS

Se plantea un modelo matemático que permita convertir los puntos de estas carteras en puntos de un sistema de coordenadas rectangulares (x, y, z), primero es necesario definir la información de entrada es decir la cartera.

El ángulo de inclinación que forma el eje central de la cartera respecto al eje vertical del sistema de coordenadas (y norte) se llamará *azimut* y el ángulo que forma el eje central de respecto del eje horizontal (x este) es llamado *acumulado*, el acumulado variara de acuerdo a la inclinación del eje central. Su valor inicial es de $acumulado = \frac{\pi}{2} - azimut$

Para el eje central de la cartera y su conversión a puntos 3D se usara la siguiente notación:

ejeCentral[longitud] = arreglo de puntos del eje de referencia.

ejeCentral3D[longitud]

■ arreglo de puntos del eje de referencia en el sistema de coordenada rectangulares (x, y, z)

Donde, *longitud* ■ cantidad de puntos que tiene el eje de referencia

■ cantidad de tuplas de la cartera topografica

De tal manera que:

ejeCentral[0].abscisa

■ Valor de la abscisa del primer punto de la cartera (longitud valor real)

ejeCentral3D[0].x

■ coordenada rectangular x del primer punto de la cartera (longitud real Este)

ejeCentral[0].cota

■ valor de la cota del primer punto de la cartera (longitud valor real)

ejeCentral3D[0].z

■ coordenada rectangular y del primer punto de la cartera (longitud real altura)

ejeCentral3D[i].z = ejeCentral[i].cota; para todo $0 \leq i < longitud$

Dado que la altura del terreno en ese punto es igual a la cota.

ejeCentral[0].deflexión

■ valor de la deflexión del primer punto de la cartera (angulo valor real)

ejeCentral3D[0].y

■ ***coordenada rectangular y del primer punto de la cartera (longitud real norte)***

Para las carteras de las secciones transversales se usa la siguiente notación:

izquierda[longitud][x]

■ ***matriz con los puntos de las secciones transversales izquierda***

izquierda3D[longitud][x]

■ ***matriz con los puntos en el sistema de coordenadas rectangulares de las secciones transversales izquierda***

donde *x* es la la cantidad de puntos que contiene un sección trasversal específica

longitud_{0i} = cantidad de puntos de la primera sección transversal izquierda

longitud_{1i} = cantidad de puntos de la segunda sección transversal izquierda

:

longitud_{ni} =

cantidad de puntos de la (*n* - 1) éstima sección transversal izquierda

izquierda[0][0]

■ ***el punto mas cercano de la primera sección transversal izquierda***

izquierda[0][longitud_{0i}]

■ ***el punto mas lejano de la primera sección transversal izquierda***

izquierda[longitud][0]

■ ***el punto mas cercano de la última sección transversal izquierda***

izquierda[longitud][longitud_{0i}]

■ ***el punto mas lejano de la última sección transversal izquierda***

En Cada punto puede accederse a su respectiva abscisa cota y deflexión para ***izquierda*** y a los valores de *y*, *x* y *z* para ***izquierda3D*** de la misma forma definida para ***ejeCentral***.

derecha[longitud][x]

■ ***matriz con los puntos de las secciones transversales derecha***

derecha3D[longitud][x]

■ ***matriz con los puntos en el sistema de coordenadas rectangulares de las secciones transversales derecha***

donde *x* es la la cantidad de puntos que contiene un sección trasversal específica

longitud_{0d} = cantidad de puntos de la primera sección transversal derecha

longitud_{1d} = cantidad de puntos de la segunda sección transversal derecha

:

longitud_{nd} =

cantidad de puntos de la (*n* - 1) éstima sección transversal derecha

Los índices y el acceso a los valores de ***derecha*** y ***derecha3D*** son los mismos usados para ***izquierda*** y ***izquierda3D***.

Ya habiendo definido los valores de entrada ***ejeCentral***, ***izquierda*** y ***derecha*** a es posible a partir de estos arreglos generar la información de ***ejeCentral3D***, ***izquierda3D*** y ***derecha3D***, es necesario también definir algunas operaciones básicas.

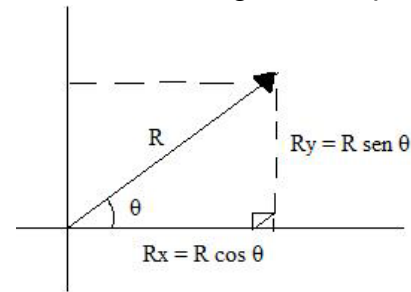
polar2Rec(longitud, ángulo) = [x, y] Coordenadas rectangulares a partir de coordenadas polares.

$$x = longitud * \cos(\text{ángulo})$$

$$y = longitud \sin(\text{ángulo})$$

$$\text{regresar} - [x, y]$$

Fin polar2Rec



transversales(indice, x, y) Ingresa las secciones transversales del punto hallado anterioridad

con

$$\text{angIzquierdo} = \text{acumulado} + \frac{\pi}{2}$$

para todos los t entre 0 y longitud_{qz}

$$[\text{delta}_{xt}, \text{delta}_{yt}] = \text{polar2rec}(\text{izquierda}[\text{indice}][t].\text{absclsa}, \text{angIzquierdo})$$

$$\text{izquierda3D}[\text{indice}][t].x = \text{delta}_{xt} + x$$

$$\text{izquierda3D}[\text{indice}][t].y = \text{delta}_{yt} + y$$

$$\text{izquierda3D}[\text{indice}][t].z = \text{izquierda}[\text{indice}][t].\text{cota}$$

Fin para

$$\text{angDerecho} = \text{acumulado} - \frac{\pi}{2}$$

para todas las t entre 0 y longitud_{qd}

$$[\text{delta}_{xd}, \text{delta}_{yd}] = \text{polar2rec}(\text{derecha}[\text{indice}][t].\text{absclsa}, \text{angDerecho})$$

$$\text{derecha3D}[\text{indice}][t].x = \text{delta}_{xd} + x$$

$$\text{derecha3D}[\text{indice}][t].y = \text{delta}_{yd} + y$$

$$\text{derecha3D}[\text{indice}][t].z = \text{derecha}[\text{indice}][t].\text{cota}$$

Fin para

Fin transversales

Haciendo uso de lo anterior se define el cuerpo central del algoritmo:

valores de entrada: *aztmu*, *ejeCentral*, *izquierda* y *derecha*

$$\text{acumulado} = \frac{\pi}{2} - \text{aztmu} \quad \text{ejeCentral3D}$$

$$[x_0, y_0] = \text{polar2rec}(\text{ejeCentral}[0].\text{absclsa}, \text{ejeCentral}[0].\text{deflexión})$$

$$\text{ejeCentral3D}[0].x = x_0$$

$$\text{ejeCentral3D}[0].y = y_0$$

$$\text{ejeCentral3D}[0].z = \text{ejeCentral}[0].\text{cota}$$

transversales(0, x_0 , y_0)

$$\text{acumulado} = \text{acumulado} - \text{ejeCentral}[0].\text{deflexión}$$

para todos los t entre 0 y longitud - 1

$$[\text{delt}_x, \text{delt}_y] = \text{polar2rec}(\text{ejeCentral}[t+1].\text{absclsa} - \text{ejeCentral}[t].\text{absclsa}, \text{acumulado})$$

$$\text{ejeCentral3D}[t+1].x = \text{ejeCentral3D}[t].x + \text{delt}_x$$

$$\text{ejeCentral3D}[t+1].y = \text{ejeCentral3D}[t].y + \text{delt}_y$$

$$\text{ejeCentral3D}[t+1].z = \text{ejeCentral}[t+1].\text{cota}$$

Si *ejeCentral*[$t+1$].*deflexión* es diferente de 0


```

        acumulado = acumulado - ejeCentral[l + 1].deflexión
    Fin si
    transversales(l + 1, ejeCentral3D[l + 1].x, ejeCentral3D[l + 1].y)
Fin para

```

2.2.2.2 Interpolación

Como en el caso anterior, vamos a definir primero la notación de la información que vamos a utilizar:

inicio = cota o altura de inicio de las curvas de nivel del terreno

(ninguna curva se encontrará a una altura más baja de este valor) "longitud valor real"

fin = cota o altura de finalización de las curvas de nivel del terreno

(ninguna curva de nivel se encontrará a una altura mayor de éste valor) "longitud valor real"

esquidistancia

■ la diferencia de alturas fija entre cada curva de nivel con sus curvas contiguas "longitud valor real"

De la misma forma son accesibles los valores de (y ,z). También es necesaria una

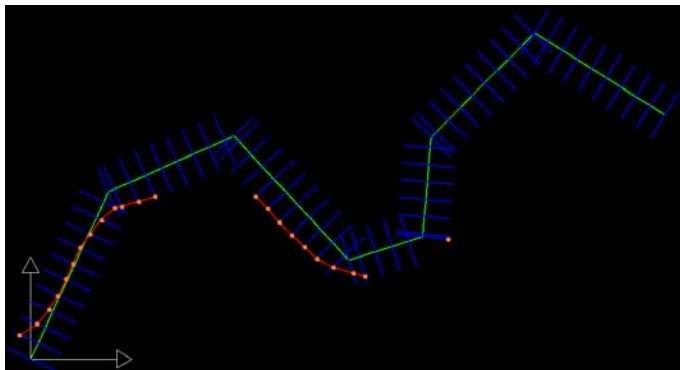


ILUSTRACIÓN 8 PUNTOS CURVA DE NIVEL

la línea verde y las secciones transversales en azul, para encontrar la curva de nivel 1634 de este caso se requiere encontrar los puntos naranjas que se ven dentro del gráfico a partir de la información de los puntos que conforman las secciones transversales azules y se utiliza para esta tarea la tercera coordenada (z altura).

En cada sección transversal podrían interpolarse varios puntos. Una cartera contiene varias secciones transversales y una representación del terreno puede tener varias curvas de nivel.

puntosCurvas[n][longitud][d]

■ puntos interpolados para las curvas de nivel del terreno

Donde n como en el caso anterior es la cantidad de curvas de nivel que se van a realizar, longitud de la misma forma que en la sección anterior es la cantidad de tuplas que tiene la cartera de datos (cantidad de secciones transversales) y d es la cantidad de puntos que se interpolaron sobre una sección transversal específica para una curva de

nivel determinada.

puntosCurvas[0][0][0].x

■ la coordenada x del primer punto de la primera sección transversal de la primera curva de nivel del terreno

puntosCurvas[0][0][d₀].x

■ la coordenada x del último punto de la primera sección transversal de la primera curva de nivel del terreno

puntosCurvas[0][longitud₀][d₀].x

■ la coordenada x del último punto de la última sección transversal de la primera curva de nivel del terreno

puntosCurvas[n][longitud₀][d₀].x

■ la coordenada x del último punto de la última sección transversal de la última curva de nivel del terreno

De forma semejante se acceden a los datos de (y, z). Habiendo definido la notación del modelo para interpolar los puntos de las curvas de nivel del terreno es posible plantear un algoritmo matemático que a partir de los resultados de la sección anterior **EjeCentral3D, izquierda3D y derecha3D** y de los valores iniciales **inicio, fin y equidistancia** interpolé los puntos de las curvas de nivel del terreno dando como resultado **puntosCurvas** para este fin primero vamos a definir unas subrutinas:

obtenerPunto(largo, ancho)= obtiene el punto3D en los índices especificados

si ancho es negativo

ancho = -1 - ancho

regresar izquierda3D[largo][ancho]

fin si

ancho es mayor que 0

ancho = ancho - 1

regresar derecha3D[largo][ancho]

fin si ancho es mayor

si ancho es igual a 0

regresar ejeCentral3D[largo]

fin si ancho

fin obtenerPunto

interpolador(largo1, ancho1, largo2, ancho2, altura)

punto1 = obtenerPunto(largo1, ancho1)

punto2 = obtenerPunto(largo2, ancho2)

si p1 o p2 son vacíos

retorna vacío

fin si p1 o p2

si altura se encuentra entre punto1.z y punto2.z

distanciaX = punto2.x - punto1.x

distanciaY = punto2.y - punto1.y

distanciaZ = punto2.z - punto1.z

mx = distanciaX / distanciaZ

```

my =  $\frac{\text{distanciaY}}{\text{distanciaZ}}$ 
delta = altura * punto1.z
x = punto1.x + delta * mx
x = punto2.x + delta * my
retorno.x = x
retorno.y = y
retorno.z = altura
Retornar

si no
    retornar vacio
fin si altura
fin interpolar

inicio
Entradas: EjeCentral3D, Izquierda3D, derecha3D, inicio, fin y equidistancia
para toda la altura desde inicio hasta fin cada equidistancia
    para cada tupla desde 0 hasta longitud
        para todo punto desde 0 hasta longitudtupla i - 1
            interpolado = interpolado(tupla, -(punto
                + 1), tupla, -(punto + 2), altura)
            si interpolado es distinto de vacio
                puntosCurvas[(altura
                    - inicio) modulo equidistancia][tupla][dtupla]
                    = interpolado
            fin si interpolado es distinto
        fin para todo punto
        interpolado = interpolado(tupla, -1, tupla, -0, altura)
        si interpolado es distinto de vacio
            puntosCurvas[(altura
                - inicio) modulo equidistancia][tupla][dtupla]
                    = interpolado
        fin si interpolado es distinto
        interpolado = interpolado(tupla, 0, tupla, 1, altura)
        si interpolado es distinto de vacio
            puntosCurvas[(altura
                - inicio) modulo equidistancia][tupla][dtupla]
                    = interpolado
        fin si interpolado es distinto
    fin para cada tupla
    para todo punto desde 0 hasta longitudtupla d - 1
        interpolado = interpolado(tupla, (punto
            + 1), tupla, (punto + 2), altura)
        si interpolado es distinto de vacio

```

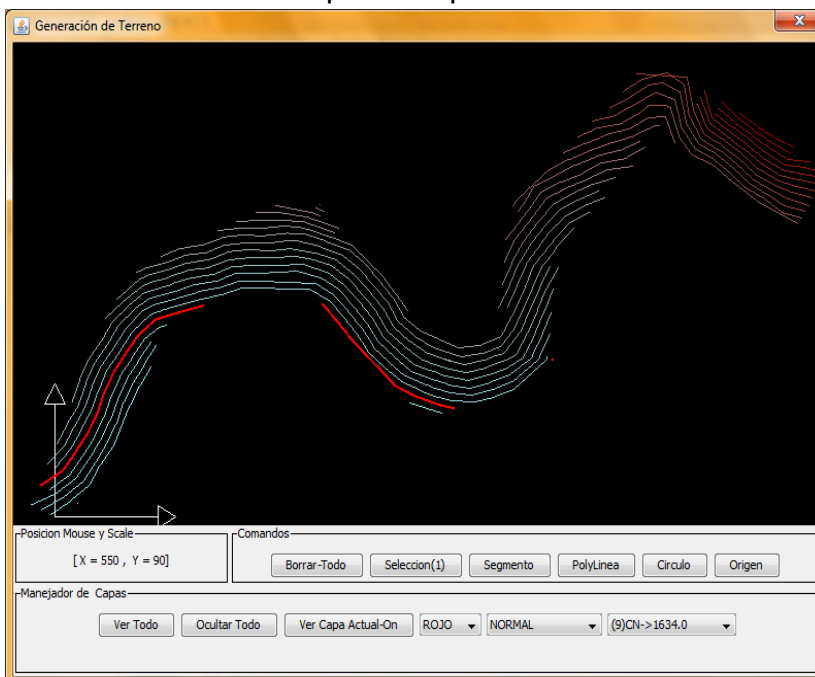
```

puntosCurvas[(altura
- inicio) modulo equidistancia] [tupla] [d_tupla]
= interpolado
fin si interpolado es distinto
fin para todo punto
fin para cada tupla
fin para cada altura
fin

```

2.2.2.3 Organización de Puntos:

La información con la que se representa una curva de nivel se define a partir de un



conjunto de puntos con la misma altura. En este conjunto, los puntos no se unen todos de forma arbitraria, sino en un orden formando trazos sobre el plano. Estos trazos, a pesar de estar a la misma altura, pueden separarse por un abrupto en el terreno como una falla geográfica o porque llega hasta el límite del plano. Un ejemplo de esto se ve en la figura tal donde se resalta en

ILUSTRACIÓN 9 CURVA DE NIVEL DEL TERRENO

rojo la curva de nivel 1634. Debido a esto, un trazo es un conjunto ordenado de uno o más puntos que se unen secuencialmente. Una curva de nivel es un conjunto de uno o más trazos que se encuentran a la misma altura y una representación digital del terreno como la que se muestra en la figura tal se conforma de un conjunto de una o más curvas de nivel.

$curvas[n][t][p] = \text{conjunto de curvas de nivel del terreno}$

donde:

$$n = \frac{fin - inicio}{equidistancia}$$

■ es la cantidad de curvas de nivel que se encontrarán en el gráfico

$t = \text{conjunto de trazos que tiene una curva de nivel en particular}$

(depende del terreno y de los límites de la información tomada del mismo)

p

■ cantidad de puntos que contiene un trazado específico de una determinada curva de

nivel (depende de la cantidad de información que se ha tomado del terreno)

De tal forma que:

Curvas[0][0][0].x = la coordenada x del primer punto del primer trazo de la primera curva de nivel del terreno

Curvas[0][0][p₀].x = la coordenada x del último punto del primer trazo de la primera curva de nivel del terreno

Curvas[0][t₀][p₀].x = la coordenada x del último punto del último trazo de la primera curva de nivel del terreno

Curvas[n][t₀][p₀].x = la coordenada x del último punto del último trazo de la última curva de nivel del terreno

Con base en los resultados obtenidos en la sección anterior

puntosCurvas, n, longitud, abscisado n

intercambio() "intercambia los puntos de los corredores de la carretera cuando algunos están más cerca de otros evitando que las curvas de nivel se crucen"

para toda curva desde 0 hasta n

para toda tupla de 0 hasta longitud

nroPuntos = d_{tupla curva}

nroPuntos2 = d_{tupla+1 curva}

nroPuntos3 = d_{tupla+2 curva}

para los i entre 0 y nroPuntos

para los j entre 0 y nroPuntos2

x₁ = puntosCurvas[curva][tupla + 1][j].x

- puntosCurvas[curva][tupla][t].x

y₁ = puntosCurvas[curva][tupla + 1][j].y

- puntosCurvas[curva][tupla][t].y

distancia₁ = $\sqrt{x_1^2 + y_1^2}$

para los k entre 0 y nroPuntos3

x₂ = puntosCurvas[curva][tupla + 2][k].x

- puntosCurvas[curva][tupla][t].x

y₂ = puntosCurvas[curva][tupla + 2][k].y

- puntosCurvas[curva][tupla][t].y

distancia₂ = $\sqrt{x_2^2 + y_2^2}$

si distancia₁ es mayor que distancia₂

temporal = puntosCurvas[curva][tupla + 1][j]

puntosCurvas[curva][tupla + 1][j] = puntosCurvas[curva][tupla + 2][k]

puntosCurvas[curva][tupla + 2][k] = temporal

fin si distancia

fin para k

fin para j

fin para los i

fin para toda tupla

fin para toda curva

fin intercambio

mirarHacia(curva.corredor.abscisado)

```

puntosCorredor = puntosCurvas[curva][corredor]
puntosTrazos = curvas[curva].getTrazos
    = puntos finales de todos los trazos de la curva
nroTrazos = puntosTrazos.longitud( )
    = cantidad de puntos dentro del arreglo
nroPuntos = puntosTrazos.longitud( )
si nroPuntos = 0
    fin mirarHacia
fin si nroPuntos
miranLosTrazos[nroTrazos]
    = arreglo de enteros con el índice los puntos más cercanos
miranLosPuntos[nroPuntos]
    = arreglo de enteros con el índice los trazos más cercanos
para cada trazo desde 0 hasta nroTrazos
    distancia1 = M = "valor muy grande"
    miranLosTrazos[trazo] = -1;
    para cada punto entre 0 y nroPuntos
        distancia2
        =  $\sqrt{(ptosCorredor[punto].x - ptosTrazos[trazo].x)^2 + (ptosCorredor[punto].y - ptosTrazos[trazo].y)^2}$ 
        si distancia2 es menor que distancia1
            y distancia2 es menor que abscisado
                distancia1 = distancia2
                miranLosTrazos[trazo] = punto;
        fin si distancia2 es menor
    fin para cada punto
fin para cada trazo
para cada punto entre 0 y nroPuntos
    distancia1 = M
    miranLosPuntos[punto] = -1;
    para cada trazo entre 0 y nroTrazos
        distancia2
        =  $\sqrt{(ptosTrazos[trazo].x - ptosCorredor[punto].x)^2 + (ptosTrazos[trazo].y - ptosCorredor[punto].y)^2}$ 
        si distancia2 es menor que distancia1
            y distancia2 es menor abscisado
                distancia1 = distancia2
                miranLosPuntos[punto] = trazo
        fin si la distancia2 es menor
    fin para cada trazo
fin para cada punto

para cada trazo desde 0 asta nroTrazos

```

```

    si miranLosTrazos[trazo] es diferente de 1
        si trazo es igual a miranLosPuntos[miranLosTrazos[trazo]]
            curvas[curva][trazo]
                = ptoCorredor[miranLosTrazos[trazo]]
        si no
            miranLosTrazos[trazo] = -1;
        fin si trazo
    fin si el punto que miro me mira
fin para cada trazo
completarCurva(curva, corredor, miranLosTrazos, miranLosPuntos)
fin mirarHacia

completarCurva(curva, corredor, miranLosTrazos[ ], miranLosPuntos[ ])
    trazos = curvas[curva].getTrazos
        = puntos finales de todos los trazos de la curva curva

    para cada punto desde 0 hasta miranLosPuntos.longitud()
        si miranLosPuntos[punto] es igual a - 1
            punto0 = puntosCurvas[curva][corredor][punto]
            para cada trazo desde 0 hasta miranLosTrazos.longitud( )
                si miranLosTrazos[trazo] es diferente de - 1
                    
$$distancia = \sqrt{(trazos[trazo].x - punto_0.x)^2 + (trazos[trazo].y - punto_0.y)^2}$$

                    si distancia es menor que salto
                        curvas[curva][trazo].agregar(punto0,)
                            = "agregar el punto al trazo"
                    fin si distancia
                fin si miranLosTrazos[trazo]
            fin para cada trazo
            si miranLosPuntos[punto] es igual a - 1
                nuevoTrazo[ ] = arreglo de puntos
                nuevoTrazo.agregar(punto0)
                curvas[curva].agregar(nuevoTrazo)
            fin si miranLosPuntos[punto]
        fin si miranLosPuntos[punto]
    fin para cada punto
fin completarCurva

inicio( )
    intercambio()
    para cada curva desde 0 hasta n
        para cada corredor desde 0 hasta longitud
            mirarHacia(curva, corredor, absclada)
        fin para cada corredor

```

fin para cada curva
fin

2.3 DISEÑO GEOMÉTRICO DE VÍAS

El diseño geométrico es “el proceso de correlacionar los elementos físicos de la vía con las condiciones de operación de los vehículos y las características del terreno”

Entre los elementos físicos se encuentran:

- Vehículos
 - o Dimensiones
 - o Velocidad
 - o Conductor
- Terreno
 - o Pendiente longitudinal
 - o Pendiente Transversal

2.3.1 Criterios Diseño Geométrico de Vías

- Topografía y clases de terrenos
- Volúmenes de tránsito
- Clasificación de las vías [Según el ancho de zona]
- Topografía y clases de terrenos.
- Volúmenes de tránsito
- Clasificación de vías. Ancho de vía (zona de carreteras).
- Vehículo de diseño.
- Proyección del tránsito
- Velocidad de diseño.
- Capacidad.
- Seguridad

2.3.2 Diseño Planta

Para realizar el diseño de una vía es necesario el realizar un trazo inicial el cual es básicamente una serie de rectas consecutivas. Los puntos en donde las rectas consecutivas cambian de inclinación se conocen como deflexiones. En estos puntos es necesario el realizar un ajuste al trazado para que el cambio de la vía no sea súbito; es aquí donde se inicia el proceso de diseñar curvas circulares y/o espirales.

2.3.2.1 Curvas Horizontales⁶

2.3.2.1.1 Curva Circular Simple

Es aquella que está compuesta por un arco circular de radio único, lo que hace que sea constante e inversamente proporcional a este último.

2.3.2.1.1.1 Elementos Curva Circular Simple:

⁶ Tomado del libro DISEÑO Y GRAFICACIÓN DE VIAS DE COMUNICACIÓN ASISTIDO POR UNA HERRAMIENTA DE PROGRAMACIÓN: "DIGRAVI" (DISEÑADOR Y GRAFICADOR DE VÍAS) FASE 1

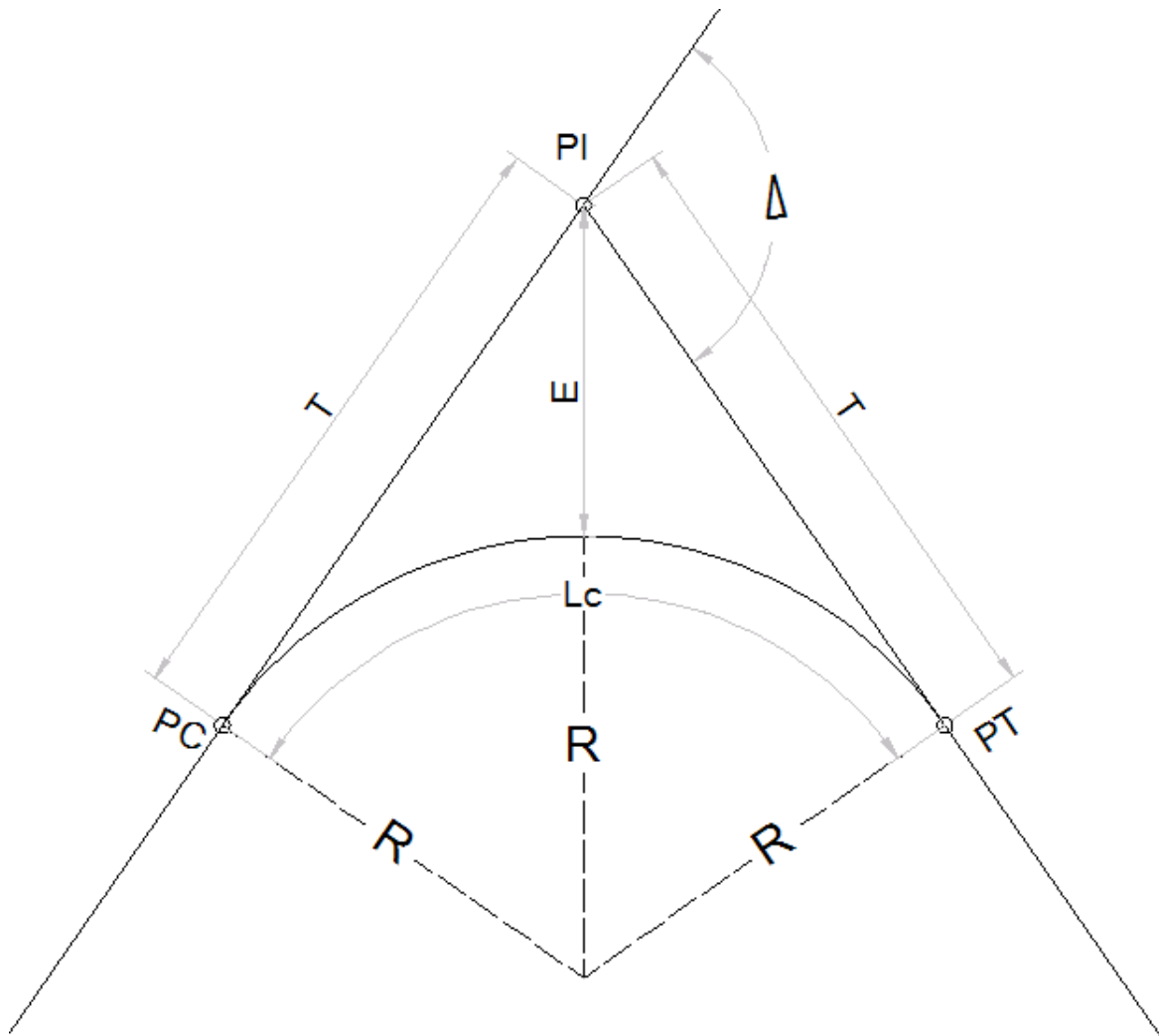


ILUSTRACIÓN 10. ELEMENTOS CURVA CIRCULAR SIMPLE. TOMADO DE MANUAL DE DISEÑO GEOMÉTRICO DE CARRETERAS 2008

PC: Punto de inicio de la curva circular.

PI: Punto de intersección entre dos alineamientos rectos horizontales.

PT: Punto de salida de la curva circular.

T: Tangente de la curva (segmentos entre **PC-PI** y **PI-PT**), calculada en metros.

Δ: Ángulo de deflexión en el **PI**, dado en grados o radianes.

R: Radio de curvatura, originada en metros.

E: Externa de la curva circular, expresada en metros.

Lc: Longitud de la curva circular, dada en metros.

C: Cuerda, que es la distancia constante entre las estaciones redondas de una curva.

G: Grado de curvatura, que representa el ángulo central de curvatura subtendido por una cuerda escogida como unidad.

CL: Cuerda Larga de la curva circular expresada en metros.

2.3.2.1.1.2 Relaciones entre Elementos Curva Circular Simple

$$PC = PI - T, \quad PT = PC + Lc$$

$$R = \frac{\frac{C}{2}}{\operatorname{sen}\left(\frac{G}{2}\right)}$$

$$T = R \operatorname{tg} \frac{\Delta}{2}$$

$$CL = 2R \operatorname{sen} \frac{\Delta}{2}$$

$$E = T \operatorname{tg} \frac{\Delta}{4}$$

$$G = \frac{C \Delta}{Lc}$$

2.3.2.1.1.3 Localización en el plano de la Curva

Con el objetivo de asignar algunas coordenadas (x , y) que permitan posterior construcción de la curva el método que se emplea es descrito a continuación:

Para localizar los diferentes puntos es necesario tener una coordenada de inicio PC, la idea es generar una coordenada cada C.

1. se calcula un ángulo y se suma al acumulado α
2. se calcula el g acumulado
3. se calcula una longitud M, $M = 2R\text{sen}(g\text{Acumulado} / 2)$
4. se calcula la coordenada (x,y),

$$X = PC.X + M\text{sen}(\text{azimut} + \alpha)^7$$

$$Y = PC.Y + M \cos(\text{azimut} + \alpha)$$

Para realizar los dos primeros pasos hay que tener en cuenta dos casos especiales que ocurren al principio y final de la curva, en estos dos lugares no se logra la longitud C.

Calculo de coordenadas para el primer y último tramo:

Para el primer tramo calcula la distancia entre la abscisa del PC y la abscisa de la primera cuerda, que llamaremos d.

Para el último tramo se calcula la distancia entre la abscisa del PT y la abscisa de la ultima cuerda, que llamaremos d.

$$\alpha_k = \alpha_{k-1} + \frac{(d)(g)}{2C}$$

$$g\text{Acumulado}_k = g\text{Acumulado}_{k-1} + 2\text{arcsen}\left(\frac{d}{2R}\right)$$

Calculo de coordenadas para tramos intermedios:

$$\alpha_k = \alpha_{k-1} + \frac{(g)}{2}$$

$$g\text{Acumulado}_k = g\text{Acumulado}_{k-1} + \frac{g}{2}$$

2.3.2.1.2 Curva Circular Compuesta

⁷ Azimut es ángulo que se mide desde el norte en sentido horario.

Son aquellas curvas que están compuestas en su trayectoria por la unión de dos o más curvas simples, contando así con múltiples radios.

2.3.2.1.2.1 Curva Circular Compuesta de Dos Radios

2.3.2.1.2.1.1 Elementos Curva Circular Compuesta de Dos Radios

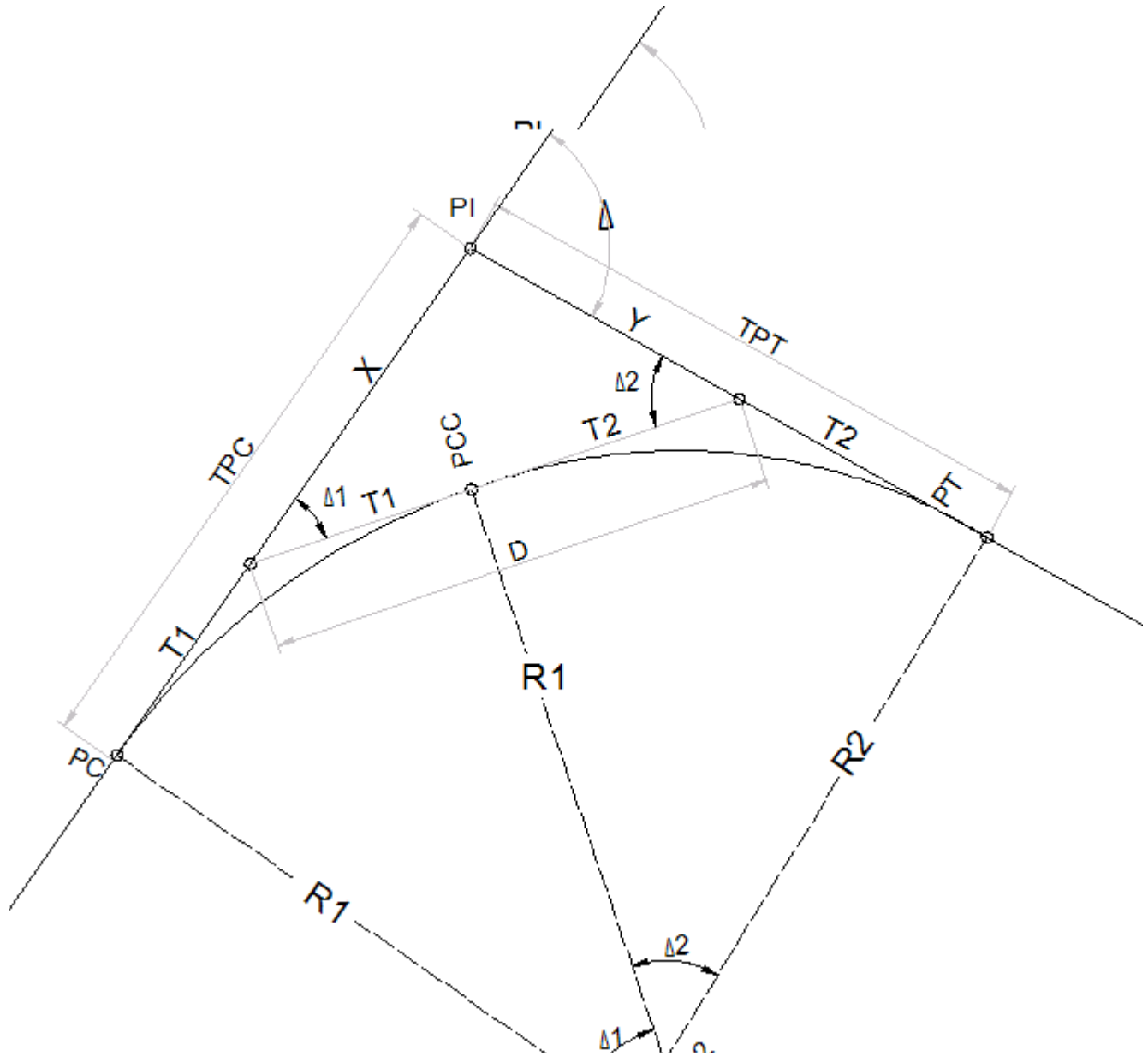


ILUSTRACIÓN 11. ELEMENTOS CURVA CIRCULAR COMPUESTA DE DOS RADIOS. TOMADO DE MANUAL DE DISEÑO GEOMÉTRICO DE CARRETERAS 2008

PC: Punto de inicio de la curva circular compuesta.

PI: Punto de intersección entre dos alineamientos rectos horizontales.

PCC: Punto de unión entre dos curvas simples.

PT: Punto de salida de la curva circular.

T1: Tangente de entrada de la curva compuesta (**PC-PI**), calculada en metros.

T2: Tangente de salida de la curva compuesta (**PI-PT**), calculada en metros.

Δ : Ángulo de deflexión en el **PI**, dado en grados o radianes.

Δ_1 : Ángulo de deflexión curva inicial, dado en grados o radianes.

Δ_2 : Ángulo de deflexión curva final, dado en grados o radianes.

R1: Radio de curvatura inicial originada, en metros.

R2: Radio de curvatura final originada, en metros.

TPC: Tangente de curva inicial, dada en metros.

TPT: Tangente de curva final, dada en metros.

Lc: Longitud de cada curva circular que comprende la compuesta, dada en metros.

G: Grado de curvatura, que representa el ángulo central subtendido por una cuerda escogida como unidad, para cada una de las curvas circulares.

2.3.2.1.2 Relaciones entre Elementos Curva Circular Compuesta de Dos Radios

Son similares a los de la curva simple solo que en este caso es si se juntaran dos curvas simples

$$PC = PI - T_1, \quad PCC = PC + LC_1, \quad PT = PCC + LC_2$$

$$\Delta = \Delta_1 + \Delta_2$$

$$T_1 = \frac{R_1 - R_2 \cos \Delta - (R_1 - R_2) \cos \Delta_1}{\sin \Delta}, \quad T_2 = \frac{R_2 - R_1 \cos \Delta + (R_1 - R_2) \cos \Delta_2}{\sin \Delta}$$

2.3.2.1.2.2 Curva Circular Compuesta de Tres Radios

2.3.2.1.2.2.1 Elementos Curva Circular Compuesta de Tres Radios

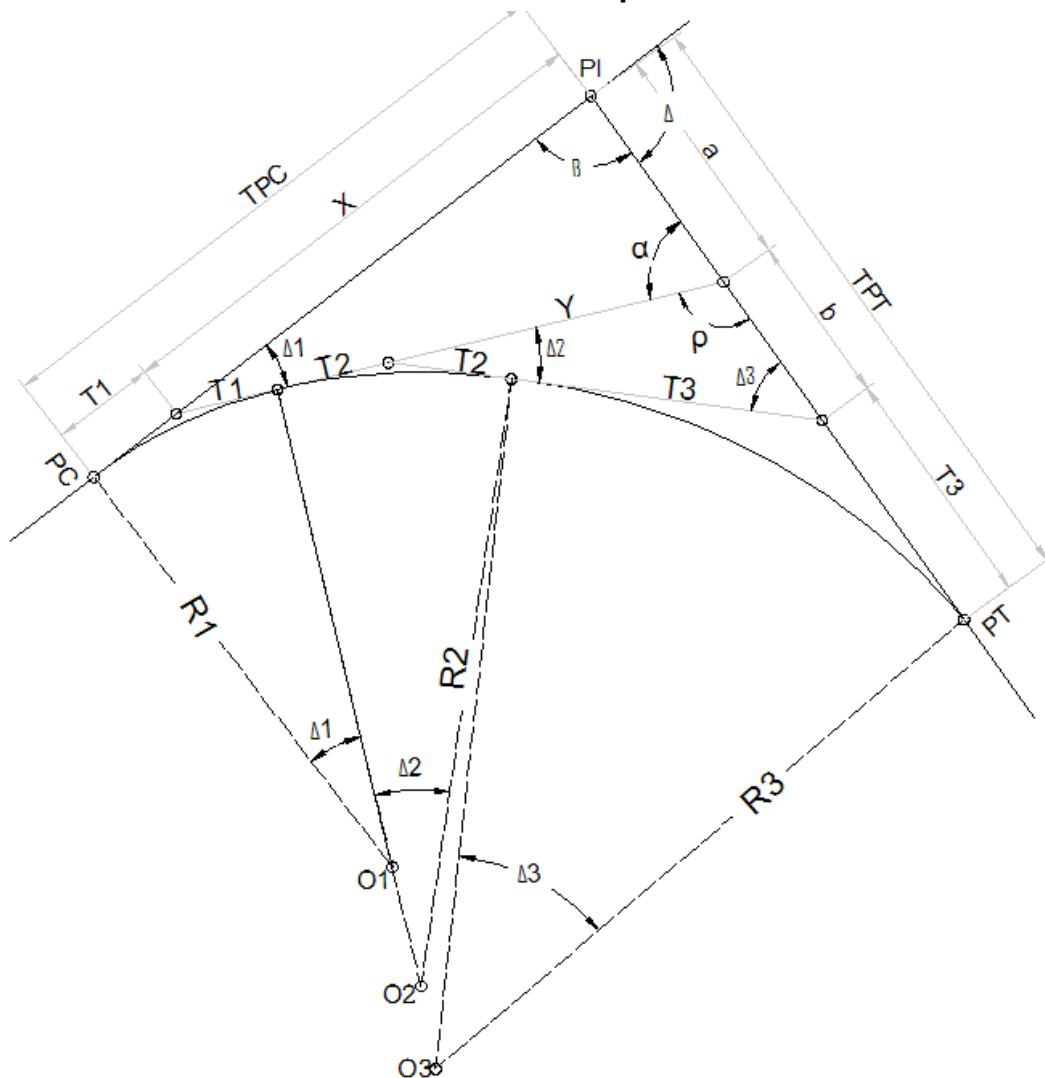


ILUSTRACIÓN 12. ELEMENTOS CURVA CIRCULAR COMPUESTA DE TRES RADIOS. TOMADO DE MANUAL DE DISEÑO GEOMÉTRICO DE CARRETERAS 2008

PC: Punto de inicio de la curva circular compuesta.

PI: Punto de intersección entre dos alineamientos rectos horizontales.

PCC1: Punto de unión inicial entre dos curvas simples.

PCC2: Punto de unión final entre dos curvas simples.

PT: Punto de salida de la curva circular.

T1: Tangente de entrada de la curva compuesta (**PC-PI**), calculada en metros.

T2: Tangente de salida de la curva compuesta (**PI-PT**), calculada en metros.

Δ : Ángulo de deflexión en el PI, dado en grados o radianes. (**$\Delta = \Delta_1 + \Delta_2 + \Delta_3$**)

Δ_1 : Ángulo de deflexión curva inicial, dado en grados o radianes.

Δ_2 : Ángulo de deflexión curva intermedia, dado en grados o radianes.

Δ_3 : Ángulo de deflexión curva final, dado en grados o radianes

R1: Radio de curvatura inicial, originada en metros.

R2: Radio de curvatura intermedio originada, en metros.

R3: Radio de curvatura intermedio originada, en metros.

Lc: Longitud de cada curva circular que comprende la compuesta, dada en metros.

G: Grado de curvatura, que representa el ángulo central de curvatura subtendido por una cuerda escogida como unidad, para cada una de las curvas circulares.

2.3.2.1.2.2 Relaciones entre Elementos Curva Circular Compuesta de Tres Radios

PC = PI – T1, PCC1 = PC + LC1, PCC2 = PCC1 + LC2, PT = PCC2 + Lc3

$$T_1 = \frac{R_1 - R_3 \cos \Delta - (R_1 - R_2) \cos \Delta_1 - (R_2 - R_3) \cos(\Delta_1 + \Delta_2)}{\text{sen} \Delta}$$

$$T_2 = \frac{R_3 - R_1 \cos \Delta + (R_1 - R_2) \cos(\Delta_2 + \Delta_3) + (R_2 - R_3) \cos \Delta_3}{\text{sen} \Delta}$$

2.3.2.1.2.3 Localización en el plano de la Curva Compuestas

Las curvas compuestas son una sucesión de curvas simples, usando el principio de dividir un problema en problemas más pequeños y conocidos, se localiza cada una de las curvas que la componen (ver 2.3.2.1.1.3 Localización en el plano de la Curva).

2.3.2.1.3 Curva Espiralizada

2.3.2.1.3.1 Curva Espiral-Circular-Espiral

Constituida por la curva que une dos líneas rectas, con un ángulo de deflexión (Δ), a través de dos segmentos de espiral clotoide y uno más en medio de ellas circular. Los tramos de espiral pueden mantener un mismo parámetro de espiral(A) o bien ser distintos, volviendo la curva asimétrica.

2.3.2.1.3.1.1 Elementos Curva Espiral-Circular-Espiral

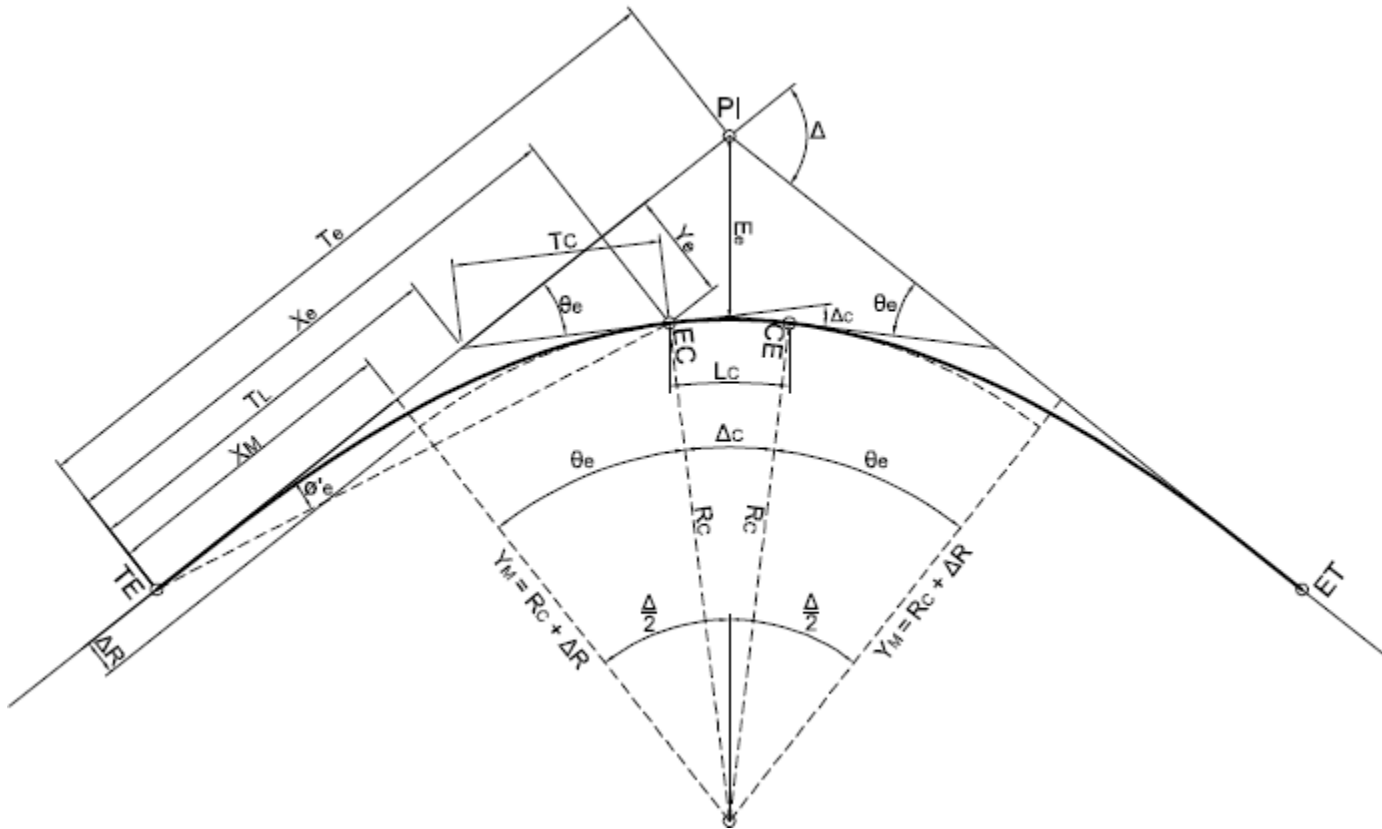


ILUSTRACIÓN 13. ELEMENTOS CURVA ESPIRAL-CIRCULAR-ESPIRAL, Tomado de manual de diseño geométrico de carreteras 2008

PI: punto de intersección de los alineamientos rectos.

Le: Longitud de la curva espiral.

RC: Radio de la curva espiral.

θ_e : Ángulo de deflexión o ángulo al centro de la espiral.

θ : Ángulo de desviación en un punto de la espiral ubicado a una distancia dada desde el origen.

(x, y): Coordenadas en un punto de la espiral ubicado a una distancia dada desde el origen. **ϕ' :** Deflexión de un punto de la espiral ubicado a una distancia desde el

origen y con coordenadas (**x**, **y**). El ángulo de desviación (**θ**), coordenadas (**x**, **y**) y el ángulo de deflexión se usan para localizar el empalme espiral en el campo.

(**Xe**, **Ye**): Coordenadas en el punto **EC** y **CE** del empalme espiral ubicado a una distancia **Le** desde el origen.

ΔR: Disloque de la espiral. Si es menor a 0.25 cm no requiere espiralizar la curva.

(**XM**, **YM**): Coordenadas del centro del arco circular cuyo radio es **RC**.

TL: Tangente larga de la espiral, expresada en metros.

TC: Tangente corta de la espiral, dada en metros.

Øe': Ángulo de la cuerda larga de la espiral o deflexión total del empalme espiral.

CLE: Cuerda larga de la espiral total desde el origen hasta el **EC**.

Δ: Deflexión en el **PI** o deflexión total del empalme:

Δ C: Deflexión del tramo circular o ángulo al centro del empalme circular.

LC: Longitud del empalme circular. Esta debe ser mínimo 0.556*VCH.

Te: Tangente de la espiral.

Ee: Externa de la espiral.

2.3.2.1.3.1.2 Relaciones entre Elementos Curva Espiral-Circular-Espiral

TE= PI- Te, EC= TE+ Le, CE= EC+ LC, ET= CE+ Le

$$Le = 2(Rc)(\theta e), \theta = \theta e \left(\frac{d}{Le} \right)^2,$$

$$Xe = Le \left(1 - \frac{\theta^2}{10} + \frac{\theta^4}{216} - \frac{\theta^6}{9360} + \frac{\theta^8}{685440} - \dots \right), Ye = Le \left(\frac{\theta}{3} - \frac{\theta^3}{42} + \frac{\theta^5}{1320} - \frac{\theta^7}{75600} + \dots \right)$$

d es la distancia aun punto sobre la espiral

$$\theta' = \arctan\left(\frac{Y}{X}\right), \Delta R = Ye + Rc(\cos \theta e - 1), X_M = Xe - (Rc)\text{sen}(\theta e), Y_M = Rc + \Delta R$$

$$T_L = Xe - \frac{Ye}{\tan(\theta e)}, T_c = \frac{Ye}{\text{sen}(\theta e)}, CL_e = \sqrt{Xe^2 + Ye^2}, \Delta c = \Delta - 2\theta e, Lc = (\Delta c)(Rc)$$

$$T_e = (Rc + \Delta R) \tan\left(\frac{\Delta}{2}\right) + X_M, E_e = \frac{Rc + \Delta R}{\cos\left(\frac{\Delta}{2}\right)} - Rc$$

En curvas espirales que sean asimétricas los elementos son los mismos simplemente que para cada segmento espiral se debe buscar el valor de sus elementos de acuerdo a sus características.

2.3.2.1.3.1.3 Localización en el plano de la Curva Espiral-Circular-Espiral

La curva espiral-circular-espiral se compone de tres partes la central que es una curva circular simple y las altéales que son secciones espiralizadas. La sección central al ser una curva circular simple como se ya se vio (ver 2.3.2.1.1.3 Localización en el plano de la Curva).

Para las secciones espirales se emplea la siguiente metodología:

1. se calcula ángulo esprital α , $\alpha = \frac{(\beta)(d)}{Le}$, donde β es el ángulo de deflexión máximo y d es la distancia entre la abscisa actual y la inicial.
2. se calcula $z = \frac{(3.1)(10)^{-3}(3)^3 + (2.3)(10)^{-8}(5)^5}{3600}$
3. se calcula $\delta = \frac{\theta e}{3} - z$
4. se calcula X y Y

$$X = d\left(1 - \frac{\theta^2}{10} + \frac{\theta^4}{216} - \frac{\theta^6}{9360} + \frac{\theta^8}{685440} - \dots\right), Y = d\left(\frac{\theta}{3} - \frac{\theta^3}{42} + \frac{\theta^5}{1320} - \frac{\theta^7}{75600} + \dots\right)$$

La ultima sección espiralizada se calcula de la misma manera en se acaba de mostrar.

2.3.2.1.3.2 Curva Espiral-Espiral

Esta curva es aquella que se da entre dos alineamientos rectos, por la unión de dos ramas espirales, que bien pueden ser iguales o diferentes, por lo que su parámetro (A) será constante a lo largo del recorrido o variable de acuerdo al caso.

2.3.2.1.3.2.1 Elementos Curva Espiral-Espiral

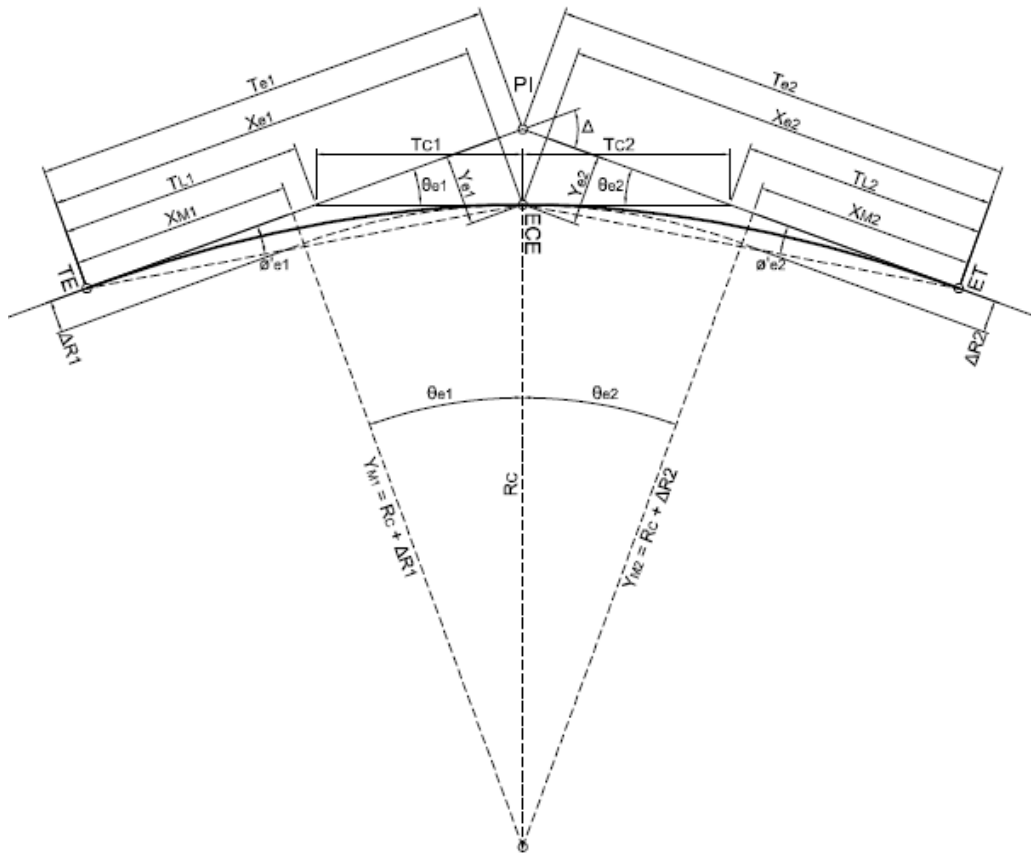


ILUSTRACIÓN 14. ELEMENTOS CURVA ESPIRAL-ESPIRAL. Tomado de manual de diseño geométrico de carreteras 2008

Los elementos son iguales para una espiral-círculo-espiral, simplemente que su Δ y L_c , no existirán.

2.3.2.1.3.2.2 Relaciones entre Elementos Curva Espiral-Espiral

El ángulo de deflexión total debe cumplir que:

$\Delta = 2 \cdot \theta_e$ para simétricas.

$\Delta = \theta e_1 + \theta e_2$ para asimétricas.

2.3.2.2 Peraltes

Se denomina así a la inclinación transversal de la vía con respecto a la horizontal, para disminuir los efectos de la fuerza centrífuga de un vehículo al transitar por un tramo en curva.

Para cada tipo de carretera, se dan unas pautas de peraltes máximos de acuerdo a la velocidad específica de la curva recorrida

TABLA 1. PERALTE MÁXIMO

Tipo Carretera	Peralte Máximo
Primarias y Secundarias	8%
Terciarias	6%

2.3.2.3 Transición Peralte

Para tratar de una manera más confortable una curva horizontal, se toma una longitud de transición que pasa el borde externo de la vía del bombeo normal, al peralte máximo de la misma, de una forma suavizada. Esta longitud está compuesta por dos tramos principales: la distancia (N) necesaria para levantar dicho borde, del bombeo a la nivelación con el eje de la carretera, denominado aplanamiento y la distancia (L) necesaria para pasar de dicho punto al peralte total en la curva circular. La magnitud total de la transición se da por la expresión:

$$L_t = L + N$$

$$N = (BN \cdot 6) / e$$

Tomándose (BN) como el bombeo normal para el tipo de vía, que generalmente se usa como el 2% para vías pavimentadas y (e) como el peralte máximo expresado en porcentaje que toma la curva.

2.3.2.3.1 Métodos de realización de la transición de peralte

Se puede dar por rotación sobre el eje de la carretera el cual es el más utilizado o por rotación alrededor de uno de los bordes de la vía (externo o interno), el cual es usado para proyectos multicarriles con separador central o para facilitar el drenaje en vías de calzada sencilla, debido a las condiciones de las pendientes transversales de la misma. Las ilustraciones de cada caso están a continuación:

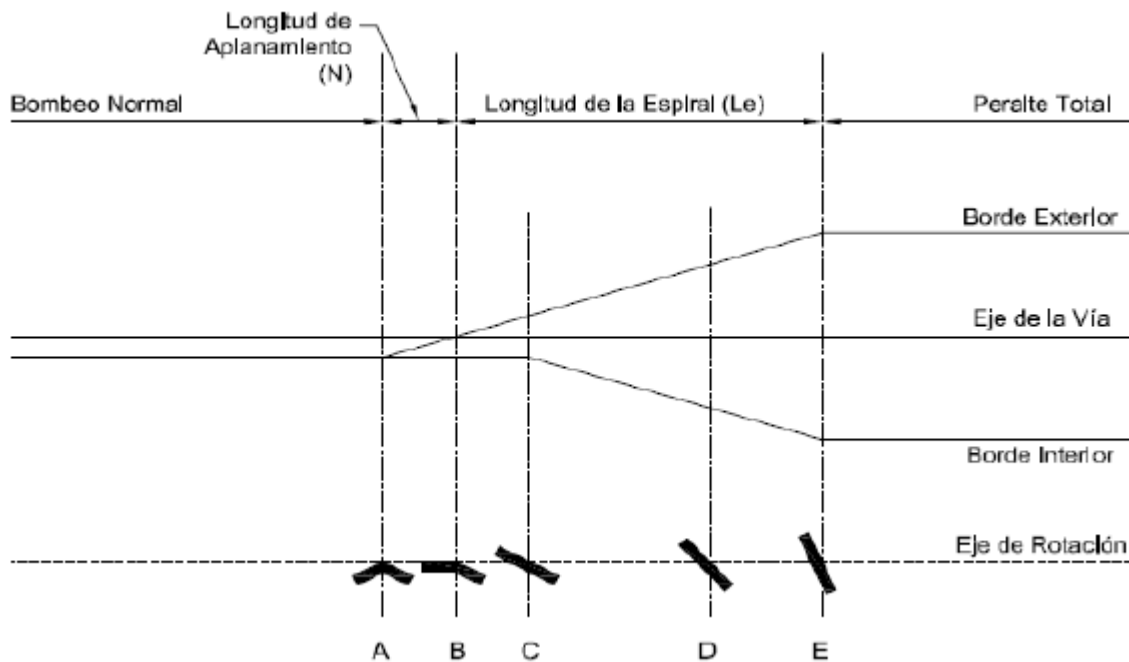


ILUSTRACIÓN 15. CALZADA GIRADA ALREDEDOR DEL EJE. TOMADO DE MANUAL DE DISEÑO GEOMÉTRICO DE CARRETERAS 2008

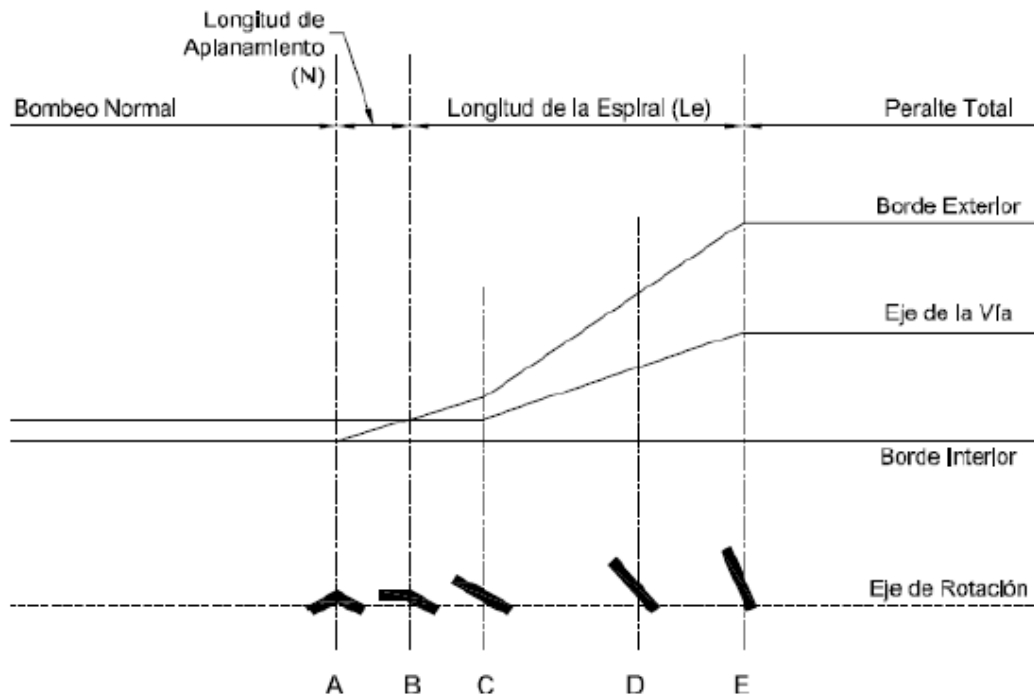


ILUSTRACIÓN 16. CALZADA GIRADA ALREDEDOR DEL BORDE INTERIOR. TOMADO DE MANUAL DE DISEÑO GEOMÉTRICO DE CARRETERAS 2008

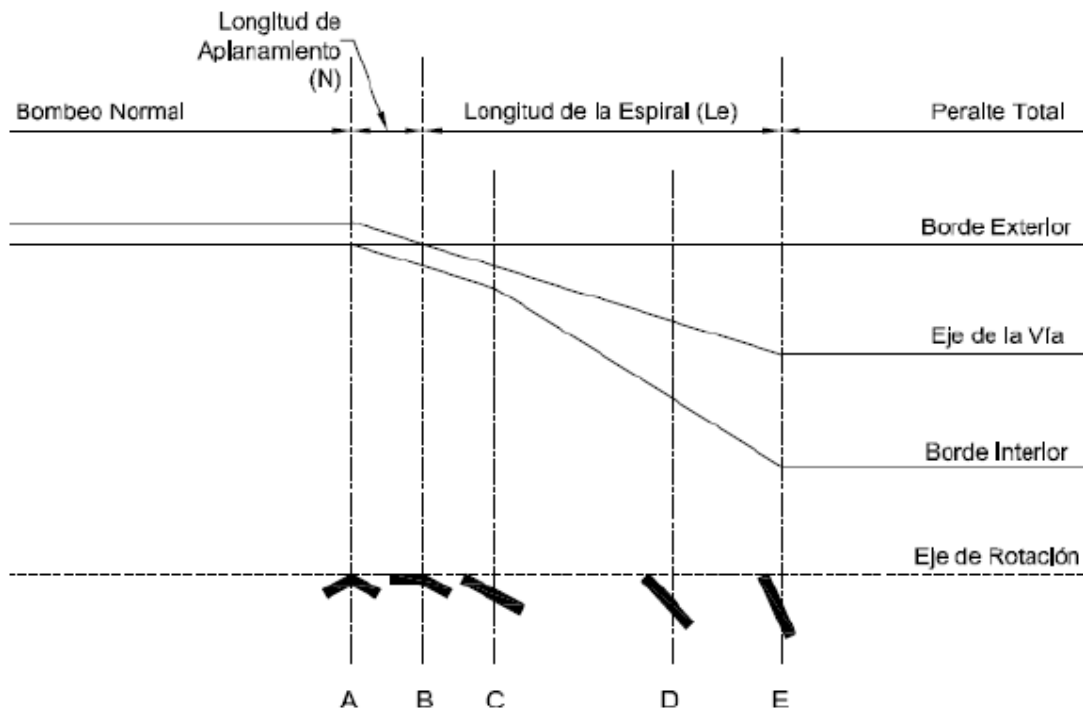


ILUSTRACIÓN 17. CALZADA GIRA ALREDEDOR DEL BORDE EXTERNO. TOMADO DE MANUAL DE DISEÑO GEOMÉTRICO DE CARRETERAS 2008

2.4- DISEÑO ASISTIDO POR COMPUTADORA (CAD)

2.4.1 Definición

El diseño asistido por computador CAD (computer assisted design) es el uso de tecnología de computación para diseñar objetos, reales o virtuales. Generalmente una herramienta CAD implica más que solo formas. Permitiendo trabajar con materiales, procesos, dimensiones y tolerancia acordes a las necesidades de la labor realizada.

2.4.2 Antecedentes

Los fabricantes del sector CAD siempre han sido punteros en aprovechar la tecnología informática más avanzada. Técnicas como el diseño vectorial, la organización de los proyectos en capas, la medición automatizada, el trabajo directo con objetos y procedimientos, la ampliación de los programas con extensiones especializadas o el diseño con modelos 3D, tienen su origen en aplicaciones de CAD, aunque en la actualidad se pueden encontrar en otros tipos de programas.

La evolución y desarrollo de las aplicaciones CAD han estado íntimamente relacionados con los avances del sector informático. El nacimiento del CAD, lo podemos situar al final periodo de los ordenadores de primera generación, pero tiene su pleno desarrollo a partir de la aparición de los ordenadores de cuarta generación en que aparecen los circuitos de alta escala de integración LSI (Large Scale Integration) y ya están desarrollados plenamente los lenguajes de alto nivel. Están desarrolladas: la memoria virtual utilizando sistemas de memoria jerárquicamente estructurados, la multiprogramación y la segmentación con el propósito de permitir la ejecución simultánea de muchas partes del programa.

A destacar, el gran interés estratégico que desde el principio ha tenido el CAD para las empresas, por el impacto enorme en la productividad. Las grandes empresas desde el principio han apostado por el CAD y ello supone importantes inversiones, que lógicamente potencian y convierten el CAD en un producto estratégico con un gran mercado.

2.4.3 Conceptos Básicos de una Herramienta CAD⁸

2.4.3.1 Entidades

Son objetos gráficos en un sistema CAD. Generalmente las entidades soportadas por la mayoría de los sistemas CAD son: puntos, líneas, círculos y arcos de elípticos. CAD más complejos y específicos incluyen polylíneas, texto, dimensiones, curvas Bézier y Spline.

2.4.3.2 Atributos

Cada entidad tiene determinados atributos tales como color, tipo de línea y ancho de línea.

2.4.3.3 Capas

Un concepto básico del dibujo asistido por computador el uso de capas para organizar y estructural el dibujo. Toda entidad de un dibujo se encuentra en una capa y cada capa puede contener cualquier número de entidades. Generalmente todas las entidades con funcionamientos comunes o con atributos comunes son construidas en las mismas capas. Cada capa tiene unos atributos (color, ancho de línea, estilo de línea) al crea una entidades en una capa generalmente hereda los atributos de la capa.

Históricamente, en el dibujo manual, se ha usado una estrategia similar. Diferentes sistemas de construcción, tales como el cableado y aire acondicionado fueron frecuentemente dibujados en diferentes laminas transparentes. Estas láminas fueron apiladas una tras otra para producir el dibujo final.

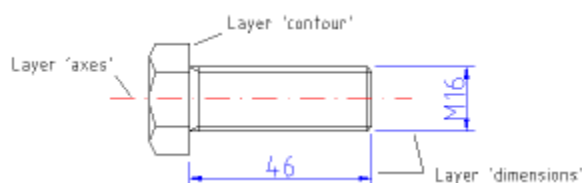


ILUSTRACIÓN 18 DIBUJO CON CAPAS. TOMADO MANUAL REFERENCIA QCAD

⁸ Extraído de la documentación de QCAD,
http://www.qcad.org/qcad/manual_reference/chapter04.html

2.4.3.4 Bloque

Un bloque es como se llama a un grupo de entidades. Un bloque puede ser insertado en un dibujo en varias veces con diferente posición, atributos, factor de escala y ángulo de rotación. Cada instancia de un bloque es llamada generalmente como 'Inserts'. Un Inserts tiene atributos como otras entidades. Una entidad que es parte de un Inserts puede tener sus propios atributos, heredándolos de los atributos de la clase o el Insert a la cual pertenece. Una vez creados, un Inserts es todavía dependiente del bloque al cual instancia. La utilidad de los Inserts radica en que una vez se modificó un bloque todos los Insert que fueron previamente creados partiendo de él serán modificados para reflejar los cambios hechos.



ILUSTRACIÓN 19. TRES INSTANCIAS DE UN BLOQUE (INSERTS) CON DIFERENTES ATRIBUTOS, ÁNGULOS Y FACTOR DE ESCALA. TOMADO MANUAL REFERENCIA QCAD

2.4.3.5 Dibujando en un CAD

En muchos aspectos, un CAD es similar al dibujo tradicional. Cuando se dibuja un plano o una vista un objeto en un papel, se usan herramientas como reglas para dibujar líneas. Un sistema CAD ofrece muchas herramientas para lograr un mismo objetivo. La gran ventaja de un sistema CAD es el hecho de que se puede cambiar una entidad fácilmente luego de ser creado. Al usar una herramienta CAD no es recomendable el realizar de forma directa el gráfico deseado, es preferible el crear auxiliares que ayuden a definir y verificar parte del dibujo que se busca generar.

2.4.3.6 Sistema de Coordenadas

El sistema de coordenadas cartesiano es el sistema por defecto usado a menos que otro se especifique. En el sistema de coordenadas cartesiano, la posición de un punto puede ser descrita por su distancia a los ejes X y Y. Las coordenadas cartesianas son generalmente escritas en el formato x-ordenada, y-ordenada

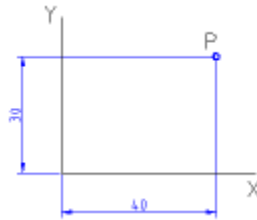


ILUSTRACIÓN 20. COORDENADAS CARTESIANAS. TOMADO MANUAL REFERENCIA QCAD

Las coordenadas polares usan distancia y ángulo para describir la posición de un punto, generalmente el ángulo 0 se encuentra dirigido hacia la derecha de la pantalla (este).

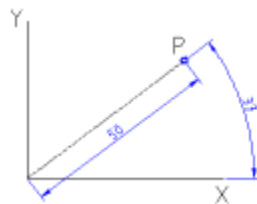


ILUSTRACIÓN 21. COORDENADAS POLARES. TOMADO MANUAL REFERENCIA QCAD

2.4.3.7 Edición Básica

2.4.3.7.1 Construcción de Entidades

Las entidades pueden ser añadidas a un dibujo empleando varias herramientas de dibujo o duplicando entidades existentes. Dibujar una entidad es necesario el definir todos los puntos y parámetros que definen una entidad, por ejemplo al dibujar es suficiente con indicar dos puntos o bien definir un punto una pendiente.

2.4.3.7.2 Borrar entidades

Borrar una entidad significa removerla del dibujo.

2.4.3.7.3 Selección de Entidades

Antes de que una entidad pueda ser borrada, duplicada, o transformando, deber ser seleccionado. La selección de entidades es una de las básicas operaciones de un CAD. Sin embargo, el seleccionar la entidad deseada para una operación no es una operación trivial.

2.4.3.7.4 Modificación de Entidades

Existen entidades que pueden ser modificadas de muchas maneras. Las modificaciones básicas incluyen translación, rotación, reflexión, y escala. Estas operaciones no alteran las características geométricas de las entidades afectadas.

2.4.3.8 Visualizando en un CAD

A diferencia del dibujo manual, en un CAD no es necesario el definir de antemano el tamaño o la escala del área de dibujo. No hay escala en el dibujo. Únicamente al imprimir es necesario el especifica una escala de dibujo con el objetivo de ajustarlo el dibujo en el papel. El modelo de dibujo no es afectado por esto y siempre permanece en el escala original 1:1.

En la pantalla, el usuario puede ajustar la región que se visualiza de le dibujo acercándose para ver más detalle o alejándose para tener una visión más amplia. Otra operación de visualización importante en un CAD es la translación de la región de visualización. Para ver otras porciones del dibujo sin cambiar la escala de visualización, esto se consigue trasladando el rectángulo de visualización hasta el punto deseado.

2.4.4 Usando una Herramienta CAD

Al igual que ocurre con el dibujo manual, con un programa de CAD, se puede conseguir cualquier composición, por muy compleja que sea, creando cuantos objetos gráficos básicos sean precisos, enlazados entre sí, hasta formar las figuras adecuadas al proyecto. Un programa de CAD es capaz de crear, modificar e imprimir figuras geométricas elementales (líneas, arcos, rectángulos, elipses, etc.) llamadas entidades, con propiedades individuales propias (color, tipo de línea, medidas, etc.).

El proceso general de trabajo es:

1. Subdividir el dibujo en entidades gráficas básicas
2. Seleccionar la función que genera la entidad deseada e introducir los parámetros necesarios.
3. Modificar las propiedades de la entidad o grupo de entidades con el objeto de lograr las características deseadas.

Un resumen de las funciones más importantes en los programas de CAD de carácter general puede ser el siguiente:

TABLA 2. FUNCIONES COMUNES HERRAMIENTA CAD

TIPO	FUNCIONES
DIBUJO	Punto-Línea-Arco-Círculo-Elipse-Curva-Rectángulo-Polígono-Polilínea-Texto-Croquis
EDICIÓN	Borrar-Copiar-Estirar-Deshacer/Rehacer-Girar-Mover-Simetría-Escala-Partir-Matrices-Enlace-Chaflán-Des/Agrupar-Texto
AYUDA AL DIBUJO	Retícula-Variables-Fijar puntos-Modos de referencia-Capas-Líneas de construcción-Selección de objetos-Coordenadas

	(absolutas, relativas, polares)-Entrada con teclado-Entrada con ratón-Entrada con tableta digitalizadora-Unidades-Precisión-Colores
VISUALIZACIÓN	Encuadre-Zoom-Pre visualización-Redibujando-Vistas-In/Visibilidad
DIMENSIONADO Y MEDICIÓN	Cota horizontal-Cota vertical-Cota alineada-Cota angular-Nota-Punto-Distancia-Perímetro-Área-Ángulo-Parámetros
SÍMBOLOS	Seleccionar-Previsualización-Insertar-Deshacer-Escala-Atributos-Editar
LÍNEAS Y TRAMAS	Rayados-Tramas-Tipos de líneas-Espesores de líneas-Ajustes-Editar
TEXTOS	Tipos de letra-Ajustes-Editor-Importar-Símbolos especiales
3D/SÓLIDOS	Primitivas-Revolución-Traslación-Operaciones lógicas
TRAZADO E IMPRESIÓN	Escala-Fichero-Ventana-Color-Pluma-Impresora-Trazador de plumillas
MACROS Y LENGUAJES DE PROGRAMACIÓN	Teclas de función-Macros-Personalización de la interfaz-Lenguaje de programación
FICHEROS DE INTERCAMBIO	ASCII-IGES-DXF
CONTROL DE FICHEROS	Pre visualizar-Cargar-Salvar-Insertar-Mezclar
AYUDAS GENERALES	Manuales-Ayuda-Tutorial -Ejemplos

VARIOS	Animación-Captura de pantallas-Librerías de símbolos-Bases de datos-Digitalización de dibujos-Módulos de ampliación-Modelización realista (rendering)
--------	---

2.4.5 Ejemplos de Herramienta CAD

2.4.5.1 AutoCAD ⁹

Autodesk AutoCAD es un programa de diseño asistido por ordenador (CAD "Computer Aided Design"; en inglés, Diseño Asistido por Computadora) para dibujo en 2D y 3D. Actualmente es desarrollado y comercializado por la empresa Autodesk.

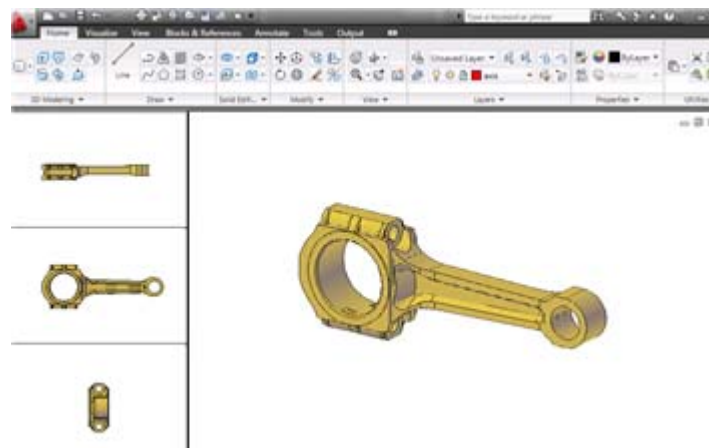


ILUSTRACIÓN 22. DISEÑO 3D AUTOCAD. TOMADO DE CREATE A 3D ENVIRONMENT TO DRAW 3D MODELS 2010, DEL SITIO OFICIAL AUTOCAD

AutoCAD gestiona una base de datos de entidades geométricas (puntos, líneas, arcos, etc.) con la que se puede operar a través de una pantalla gráfica en la que se muestran éstas, el llamado editor de dibujo.

La interacción del usuario se realiza a través de comandos, de edición o dibujo, desde la línea de órdenes, a la que el programa está fundamentalmente orientado.

⁹ Tomado del sitio Oficial Autocad, <http://usa.autodesk.com/>

Las versiones modernas del programa permiten la introducción de éstas mediante un GUI, que automatiza el proceso.

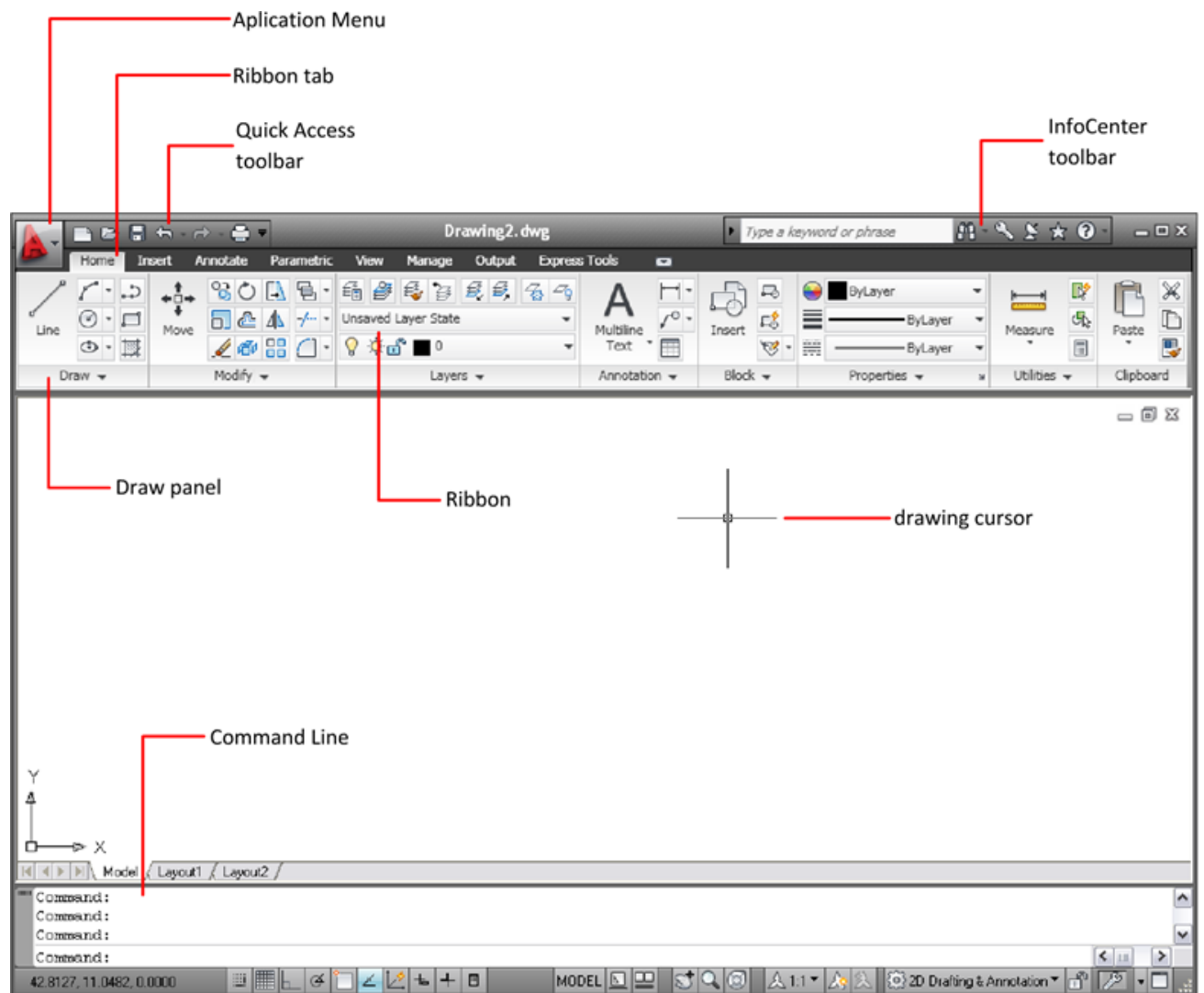


ILUSTRACIÓN 23. INTERFAZ DE USUARIO AUTOCAD. TOMADO DE GETTING STARTED WITH THE BASICS IN AUTOCAD, DEL SITIO OFICIAL AUTOCAD

Como todos los programas de CAD, procesa imágenes de tipo vectorial, aunque admite incorporar archivos de tipo fotográfico o mapa de bits, donde se dibujan figuras básicas o primitivas (líneas, arcos, rectángulos, textos, etc.), y mediante herramientas de edición se crean gráficos más complejos.

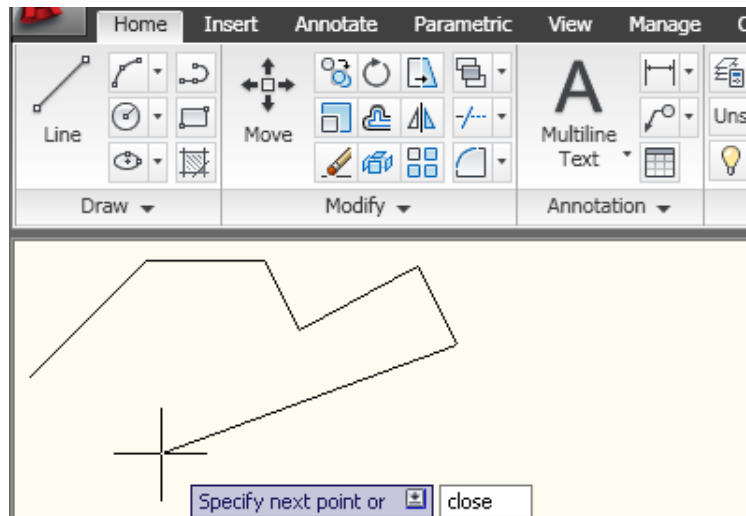


ILUSTRACIÓN 24. DIBUJANDO EN AUTOCAD. TOMADO DE GETTING STARTED WITH THE BASICS IN AUTOCAD, DEL SITIO OFICIAL AUTOCAD

El programa permite organizar los objetos por medio de capas o estratos, ordenando el dibujo en partes independientes con diferente color y grafismo. El dibujo de objetos seriados se gestiona mediante el uso de bloques, posibilitando la definición y modificación única de múltiples objetos repetidos.

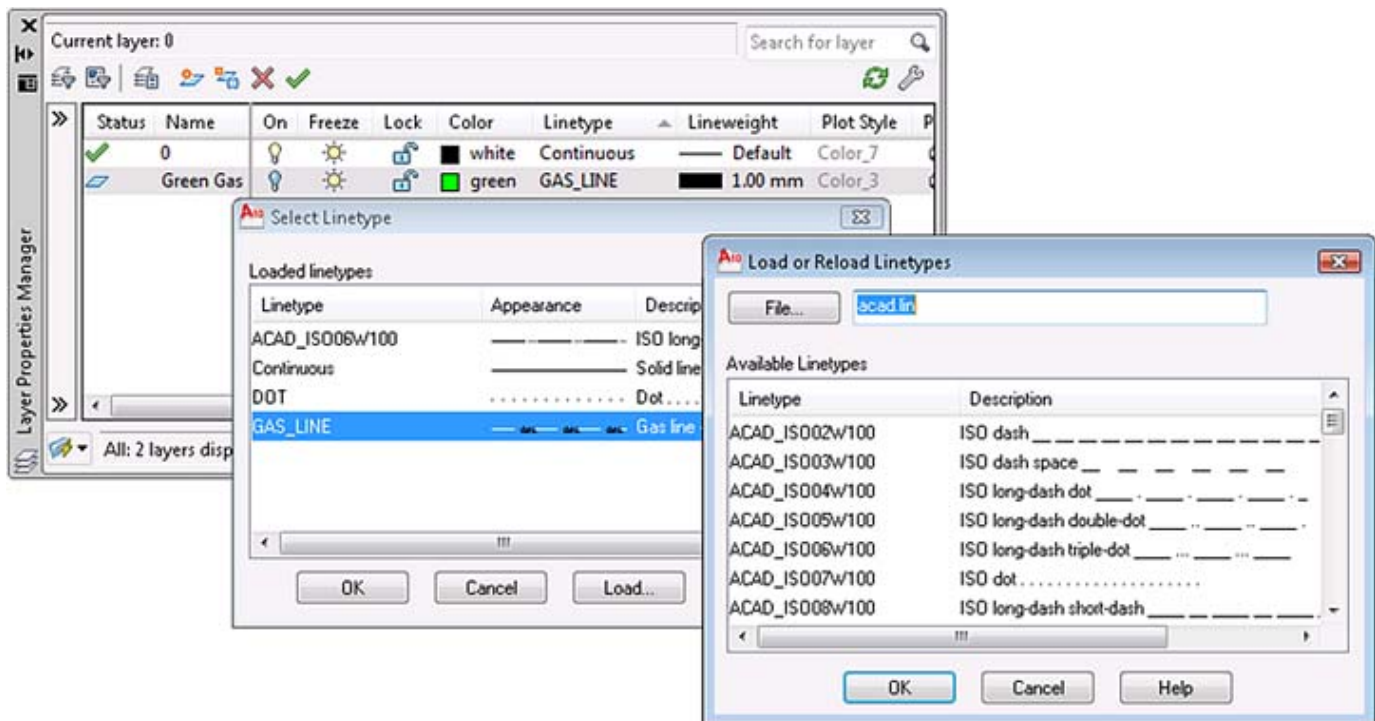


ILUSTRACIÓN 25. MANEJANDO CAPAS EN AUTOCAD. TOMADO DE MODIFY OBJECT PROPERTIES AND LAYERS IN AUTOCAD 2010 GETTING STARTED WITH THE BASICS IN AUTOCAD, DEL SITIO OFICIAL AUTOCAD

Parte del programa AutoCAD está orientado a la producción de planos, empleando para ello los recursos tradicionales de grafismo en el dibujo, como color, grosor de líneas y texturas tramadas. AutoCAD, a partir de la versión 11, utiliza el concepto de espacio modelo y espacio papel para separar las fases de diseño y dibujo en 2D y 3D, de las específicas para obtener planos trazados en papel a su correspondiente escala. La extensión del archivo de AutoCAD es .dwg, aunque permite exportar en otros formatos (el más conocido es el .dxf). Maneja también los formatos IGES y STEP para manejar compatibilidad con otros software de dibujo.

El formato .dxf permite compartir dibujos con otras plataformas de dibujo CAD, reservándose AutoCAD el formato .dwg para sí mismo. El formato .dxf puede editarse con un procesador de texto básico, por lo que se puede decir que es abierto. En cambio, el .dwg sólo podía ser editado con AutoCAD, si bien desde hace poco tiempo se ha liberado este formato (DWG), con lo que muchos programas CAD distintos del AutoCAD lo incorporan, y permiten abrir y guardar en esta extensión, con lo cual lo del DXF ha quedado relegado a necesidades específicas.

Es en la versión 11, donde aparece el concepto de modelado sólido a partir de operaciones de extrusión, revolución y las booleanas de unión, intersección y sustracción. Este módulo de sólidos se comercializó como un módulo anexo que debía de adquirirse aparte. Este módulo sólido se mantuvo hasta la versión 12, luego de la cual, AutoDesk, adquirió una licencia a la empresa Spatial, para su sistema de sólidos ACIS.

El formato .dwg ha sufrido cambios al evolucionar en el tiempo, lo que impide que formatos más nuevos .dwg puedan ser abiertos por versiones antiguas de AutoCAD u otros CADs que admitan ese formato. La última versión de AutoCAD hasta la fecha es el AutoCAD 2010, y tanto él como sus productos derivados (como Architectural DeskTop ADT o Mechanical DeskTop MDT) usan un nuevo

formato no contemplado o trasladado al OpenDWG, que sólo puede usar el formato hasta la versión 2000.

2.4.5.2 QCAD¹⁰

QCad es una herramienta (CAD) para diseño 2D, se encuentra disponible en diferentes plataformas: Windows, Mac OS X y muchos Linux y sistemas Unix

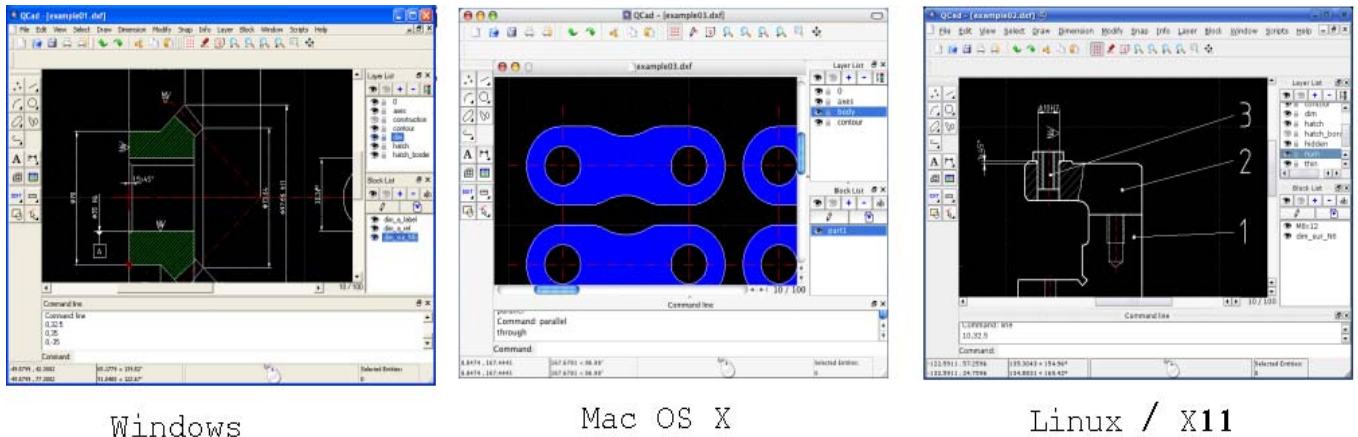


ILUSTRACIÓN 26. QCAD EN DIFERENTES PLATAFORMAS. IMAGEN MOFIDICADA, ORIGIANAL DEL SITIO OFICIAL QCADAUTOCAD

QCad fue desarrollado por RibbonSoft. El desarrollo de QCad empezó en octubre de 1999, partiendo de código de CAM Expert. QCad 2, diseñado para "hacer QCad más productivo, más amigable al usuario, más flexible y para incrementar su compatibilidad con otros productos", empezó su desarrollo en mayo de 2002.

Buena parte de la interfaz y de los conceptos sobre su uso son iguales que los de AutoCAD, aunque quizás más fácil de entender en QCad.

¹⁰ Tomado del sitio oficial de QCAD,

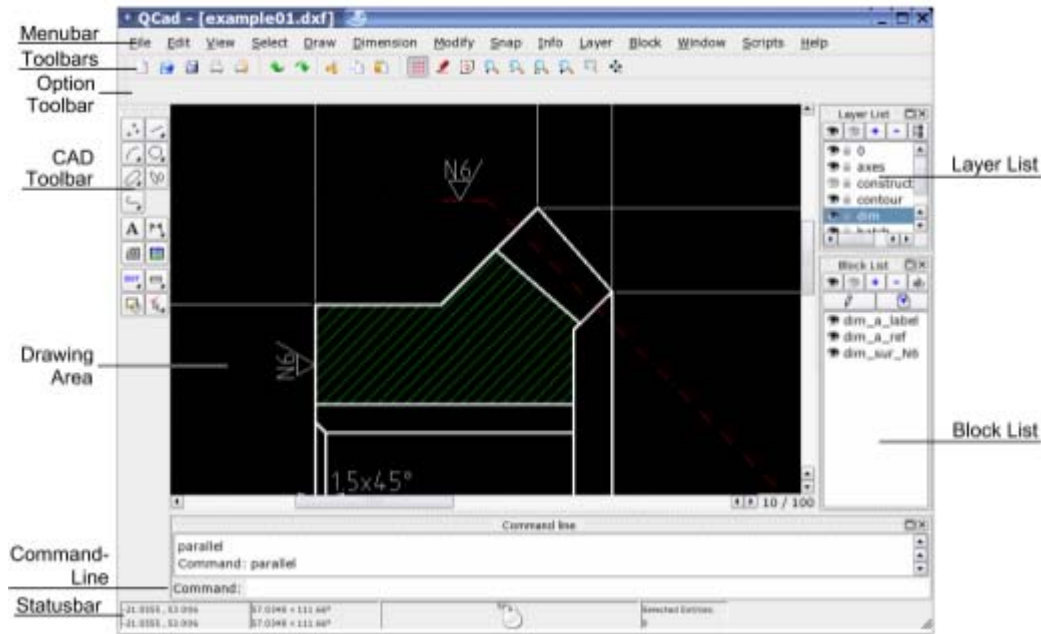


ILUSTRACIÓN 27. GUI QCAD. TOMADO DEL MANUAL DE REFERENCIA, DEL SITIO OFICIAL QCAD

Qcad utiliza el formato del archivo de AutoCAD DXF internamente y para guardar e importar archivos. Los archivos se pueden exportar en varios formatos, pero perceptiblemente los que faltan son el VRML y SVG.

2.4.5.2.1 Características:

- Soporte completo para capas y bloques (inserciones).
- Consola para inserción de coordenadas y ejecución de instrucciones.
- Incluye 35 fuentes CAD
- Unidades métricas e imperiales
- Entrada/salida DXF
- Ajuste a objetos (extremos, centros, intersecciones, etc.).
- Rellenos sólidos y rayados.
- Impresión a escala
- Múltiples niveles de «deshacer»/«rehacer».
- Más de 40 herramientas de construcción
- Más de 20 herramientas de modificación

- Varios modos de creación de líneas, arcos, círculos, elipses, paralelas, ángulos bisectores, etc.
- Herramientas potentes de selección y modificación (mover, rotar, espejo, recortar, estirar, etc.).
- Object snaps
- Herramientas de medición
- Biblioteca de piezas con más de 4800 piezas CAD
- Interfaz de programación
- Importación y exportación de mapas de bits (JPEG, PNG, etc.).
- Creación de ficheros PS que pueden portarse fácilmente a PDF.
- Interfaz de usuario traducida a múltiples idiomas: alemán, catalán, checo, danés, eslovaco, español, estonio, francés, griego, holandés, húngaro, inglés, italiano, polaco, ruso y turco.

2.4.5.3 CADEMIA¹¹

CADEMIA fue originalmente escrito con un propósito educativo y de investigación. CADEMIA es un software construido de manera modular y ofrece muchas posibilidades para añadir nuevas funcionalidades. Gracias a su flexibilidad CADEMIA es fácilmente de personalizar.

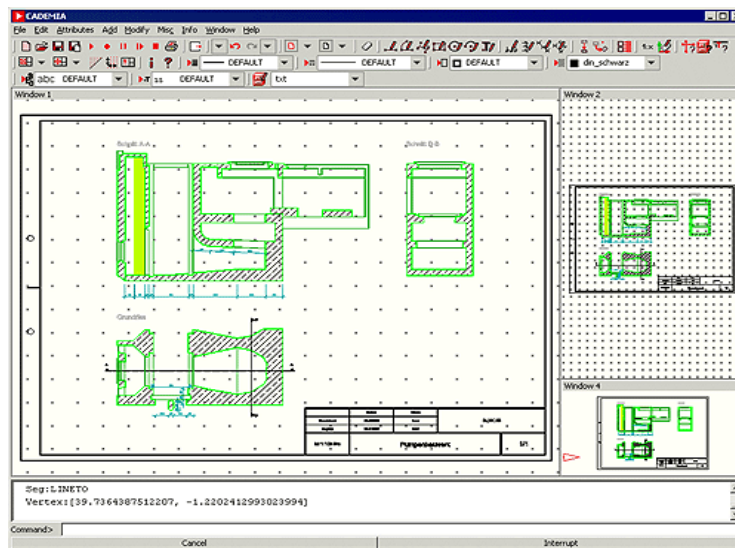


ILUSTRACIÓN 28. GUI CADEMIA. TOMADO DEL SITIO OFICIAL CADEMIA

¹¹ Tomado del sitio oficial de CADEMIA, <http://www.cademia.org/>

CADEMIA es programado en java y por lo tanto es independiente de la plataforma. Lo que lo convierte una alternativa económica a los CAD propietarios.

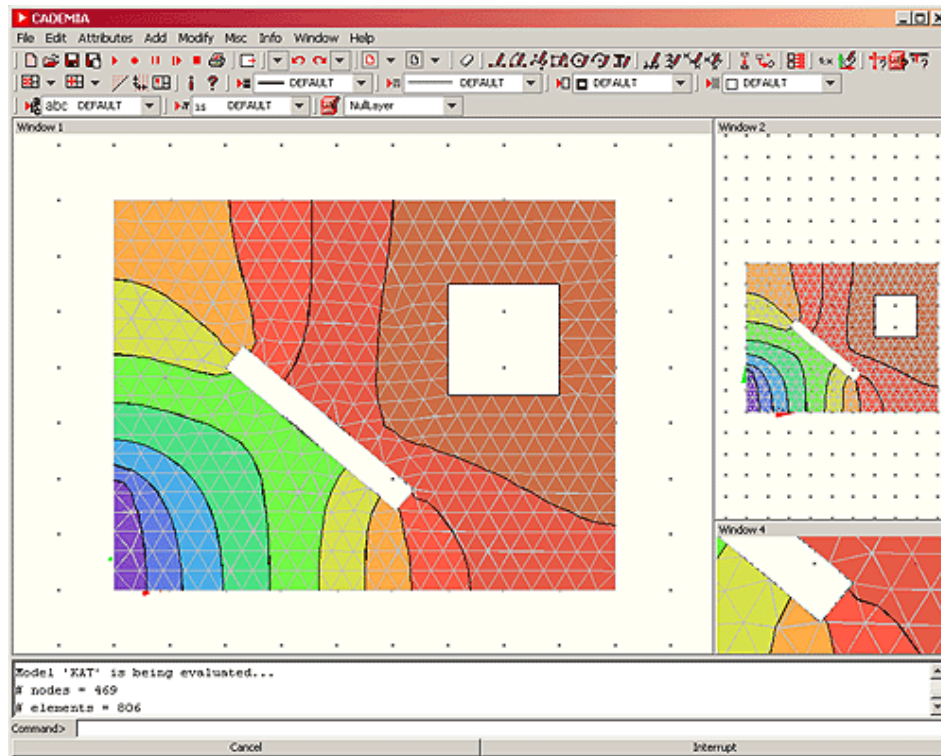


ILUSTRACIÓN 29. TRABAJANDO CON CADEMIA. TOMADO DEL SITIO OFICIAL CADEMIA

2.4.5.3.1 Características

- No necesita ningún Hardware especial para su funcionamiento
- No necesita de la instalación de software adicional, simplemente ejecutar el archivo JAR.
- Fácil de usar, menos complejo que el software existen
- Varias posibilidades de entrada:
 - o Texto
 - o Barra de Menú
 - o Barra de herramientas
 - o Mouse
- Gestión de Macros
- Historial de comandos(hacer/deshacer)
- Atributos grafico/dibujo ajustados a los estándares.

2.4.5.4 Archimedes¹²

Es un CAD enfocado en la arquitectura, desarrollado bajo la supervisión de arquitectos y firmas de arquitectos.

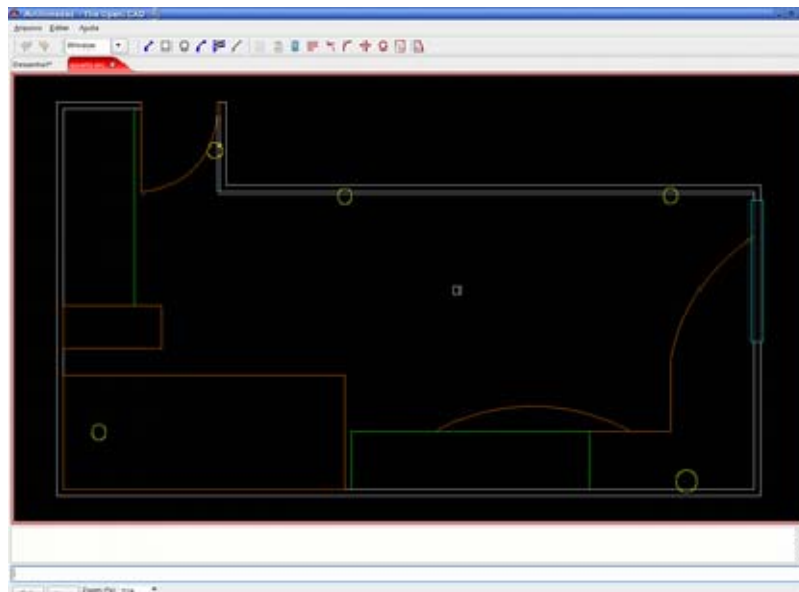


ILUSTRACIÓN 30 GUI ARCHÍMEDES. TOMADO DEL SITIO OFICIAL ARCHIMEDES

2.4.5.4.1 Características

- Herramientas básicas de dibujo
 - o Líneas, Polylineas, arcos y círculos.
 - o Texto Editable
 - o Explode (Transformar una figura en elementos más sencillos)
 - o Offset (Crear copias de una figura trasladadas)
- Funciones avanzadas de un CAD
 - o Recortar
 - o Filleting
 - o Medir áreas

¹² Tomado del sitio oficial de

- Miscelánea
 - o Auto salvado
 - o Exporta a SVG y PDF
 - o Disponible en Inglés, portugués y italiano.

2.5- USO DE EXPRESIONES REGULARES

Las expresiones regulares son usadas en el transcurso de nuestra aplicación como un mecanismo de control y corrección de errores en la recepción de datos por parte del usuario, las expresiones regulares o Regular expression en inglés, permiten definir un patrón muy flexible con el cual podemos comparar los datos ingresados por el usuario y saber si se ajustan a las características deseadas.

El uso de expresiones regulares es muy común en los compiladores para los lenguajes de programación y en los intérpretes de comando, el uso de éstas también permite reconocer si una línea de código está mal escrita, si le falta por ejemplo un punto y coma y le permite diferenciar las palabras reservadas del lenguaje de programación y dibuja en distintos colores las variables, los tipos de datos, las palabras reservadas, los comentarios. ¿Cómo se diferencian estas palabras en un texto? parece magia pero no lo es, es el uso de expresiones regulares.

Expresión regular¹³: una expresión regular es definida en términos de la teoría de lenguajes formales. Las expresiones regulares constan de constantes y operadores que conforman un conjunto de sentencias o cadenas y las operaciones sobre estos conjuntos respectivamente. La siguiente definición es estándar, fue tomada de los siguientes libros de Teoría de Lenguajes formales

^{a b} Hopcroft, Motwani & Ullman (2000).

^a Sipser (1998)

¹³ Tomado del sitio WEB http://en.wikipedia.org/wiki/Regular_expression

Dado un alfabeto finito Σ , son definidas las siguientes constantes:

- (el conjunto vacío) $\emptyset$
- (cadena vacía) ϵ , la cual no tiene ningún carácter.
- (carácter literal) a en Σ denota algún carácter en el lenguaje.

son definidas también las siguientes operaciones:

- (concatenación) RS denotan el conjunto $\{\alpha\beta \mid \alpha \text{ en } R \text{ y } \beta \text{ en } S\}$. por ejemplo $\{\text{"ab"}, \text{"c"}\} \{\text{"d"}, \text{"ef"}\} = \{\text{"abd"}, \text{"abef"}, \text{"cd"}, \text{"cef"}\}$.
- (alternativa) $R \mid S$ denota el conjunto unión de R y S . por ejemplo $\{\text{"ab"}, \text{"c"}\} \mid \{\text{"ab"}, \text{"d"}, \text{"ef"}\} = \{\text{"ab"}, \text{"c"}, \text{"d"}, \text{"ef"}\}$.
- (Estrella Kleene) R^* Es el conjunto de todas las cadenas que pueden ser hechas al hacer concatenaciones de cero o mas cadenas de R . por ejemplo, $\{\text{"ab"}, \text{"c"}\}^* = \{\epsilon, \text{"ab"}, \text{"c"}, \text{"abab"}, \text{"abc"}, \text{"cab"}, \text{"cc"}, \text{"ababab"}, \text{"abcab"}, \dots\}$.

En el desarrollo de las operaciones expresadas anteriormente se asume que la estrella Kleene tiene la más alta prioridad, seguida por la unión de conjuntos. Si no hay ambigüedad los paréntesis pueden ser omitidos, Por ejemplo: $(ab) \mid c$ puede ser escrito como abc , y $a \mid (b(c^*))$ puede ser escrito como $a|bc^*$. Muchos libros usan $\square$, $+$, o $\square$ como alternativa en remplazo de la barra vertical.

Ejemplos:

- $a|b^*$ denota $\{\epsilon, a, b, bb, bbb, \dots\}$
- $(a|b)^*$ denota todas las cadenas que solo tienen los símbolos a y b , incluyendo la cadena vacía: $\{\epsilon, a, b, aa, ab, ba, bb, aaa, \dots\}$
- $ab^*(c|\epsilon)$ denota el conjunto de cadenas que comienzan por a , seguidas de cero o mas b 's y final mente una c opcional: $\{a, ac, ab, abc, abb, abbc, \dots\}$

Lenguaje Formal: Un lenguaje formal por ejemplo L es definido sobre un alfabeto Σ , este es un subconjunto de Σ^* , el cual es un conjunto de palabras del alfabeto.

En matemáticas y ciencias de la computación generalmente no se trabaja con lenguajes naturales (como el que utilizamos diariamente), se maneja generalmente lenguajes formales pero comúnmente se refieren a ellos como Lenguajes solamente dado que se da por sentado que son formales. Los lenguajes formales son descritos por algunas leyes sintácticas, la definición de un lenguaje formal sería: un conjunto (podría ser infinito) de palabras de longitud finita, no mas y no menos que estas, hay muchas lenguas que pueden ser descritas por reglas, tales como lenguajes regulares o lenguajes libres del contexto.

El concepto de gramática formal es cercano al concepto de lenguaje el cual se encuentra descrito por reglas sintácticas.

Se realizaran algunas reglas que acoten el lenguaje de las deflexiones, una deflexión en una cartera es el ángulo que se forma cuando hay un cambio de dirección en la que se está haciendo el levantamiento del terreno (ver Ilustración plano de ejemplo)

Existe el lenguaje D definido sobre el alfabeto $\Sigma = \{ \varepsilon, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, ^\circ, ', ' ', ', ' \}$, Las cadenas del lenguaje deben someterse a las siguientes reglas:

- las cadenas no pueden comenzar con $^{\circ}$.
- las cadenas pueden o no tener el carácter $^{\circ}$ si lo tienen deben ser solo hasta 3 caracteres $^{\circ}$, estos caracteres deben ir anteceditos de uno o más caracteres numéricos.
- Si no se encuentra el carácter $^{\circ}$ entonces las cadenas pueden o no tener el carácter $'$ y el carácter $' '$ pero solo pueden tener uno de ellos a la vez y tal carácter solo puede aparecer una vez en cada palabra, si está al comienzo de una palabra deberá ir sucedido por uno o más caracteres numéricos.

De acuerdo a estas reglas las palabras $"45^{\circ}"$, $"45^{\circ}0^{\circ}0^{\circ}"$, $"29.9"$, $"'29,9"$, $"0.34"$, $".34"$, $"',34"$ hacen parte de este lenguaje pero no es el caso de $"23.45."$, $"^{\circ}45^{\circ}"$, $"',"$, $"',"$.

También tenemos las operaciones de concatenación, estrella e unión que vimos anteriormente, también le vamos a agregar las siguientes operaciones:

- operación R^+ que será la misma operación estrella excluyendo la sarta vacía ϵ , por ejemplo sea $R=\{"ab", "c"\}$ entonces $R^+ = \{"ab", "c", "abab", "abc", "cab", "cc", "ababab", "abcab", \dots\}$.
- la operación $R\{x,y\}$ donde x e y son enteros y las palabras de R se repiten de x a y veces, por ejemplo $R\{1,2\} = \{"ab", "abab", "abc", "c", "cc", "cab"\}$, en este ejemplo obligatoriamente debe haber una o dos palabras de R concatenadas.
- La operación $R?$ enuncia la existencia de una palabra de R o ninguna es decir $R?=\{ \epsilon, ab, c\}$.

A partir de estas reglas y operadores podemos formular una expresión regular que se ajuste a estas restricciones:

DEFLEXION = " **REAL**|-?**NUMERICO_M**^o){1,3} "

Donde la expresión **REAL** es un número real que puede o no tener signo y punto decimal:

REAL="((-?**NUMERICO_M**?)((**.**|**,**)**NUMERICO_M**)?)"

Y **NUMERICO_M** es una cadena no vacía de caracteres numéricos:

NUMERICO_M="(0|1|2|3|4|5|6|7|8|9)+";

Al tener un patrón de comparación podemos decir que cualquier cadena o sarta que no se ajuste a la expresión anterior no está bien formulada y por tanto es necesario pedirle al usuario que corrija su error. También este conocimiento nos proporciona herramientas para tratar las expresiones bien formadas, extraer por aparte por ejemplo los fragmentos entre los caracteres ° buscar y remplazar de las frases los caracteres que no pertenezcan al alfabeto.

El uso de expresiones regulares en este proyecto puede extrapolarse a nuevas funcionalidades del mismo tales como un intérprete de comandos, pero para nuestra primera fase no se hacía necesaria esta herramienta, los desarrolladores de las siguientes fases pueden basarse en el trabajo ya hecho en este tema para sacar provecho de él.

2.6 UML

Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para elaborar un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

2.6.1 Tipos de Diagrama

2.6.1.1 Diagramas de Estructura

Enfatizan en los elementos que deben existir en el sistema modelado:

Diagrama de clases

Diagrama de componentes

Diagrama de objetos

Diagrama de estructura compuesta (UML 2.0)

Diagrama de despliegue

Diagrama de paquetes

2.6.1.1.1 Diagramas de Clases:

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará

en el sistema, y los componentes que se encargaran del funcionamiento y la relación entre uno y otro.

Los elementos que intervienen son:

- Clases
- Relaciones de dependencia, generalización y asociación.
- Paquetes
- Interfaces

2.6.1.1.1 Representación de una clase

En este diagrama las clases se representan mediante un rectángulo que se divide en tres partes:

- La parte superior contiene el nombre de la clase
- La parte media contiene los atributos de la clase, en la forma:
 - o visibilidad nombreAtributo: tipo
- la parte inferior se encuentran los métodos o operaciones que puede realizar la clase, en la forma:
 - o visibilidad nombreMétodo(parámetros:tipo): tipoRetorno.

La visibilidad, de atributos y métodos, puede ser:

TABLA 3. VISIBILIDAD CLASE

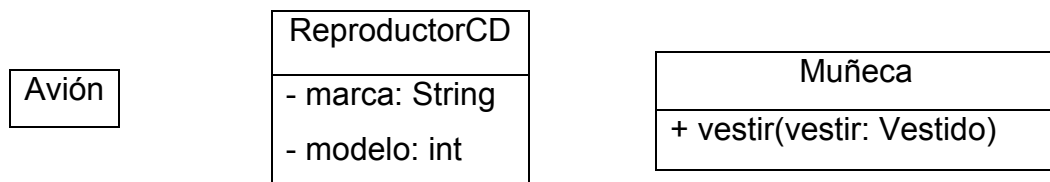
Símbolo	Representa
-	Privado
+	Publico
#	Protegido
~	Paquete

Ejemplo Clase:

AUTOMOVIL
marca: String - modelo: int - cantidadGasolina: doublé
+ Carro(marca:String, modelo:int) ~ getModelo():int + tanquear(double):void

Para representar una clase no es necesario el usar siempre los tres compartimientos puede ser suficiente el solo mostrar el nombre o bien acompañarlo del recuadro de atributos o operaciones. El objetivo de esto es enfocar la representación hacia la explicación de un comportamiento determinado de la clase.

Ejemplo:



2.6.1.1.2 Modificadores

- Los atributos y métodos estáticos se subrayan.
- Para indicar que una clase es abstracta o final se debe colocar un comentario en la parte superior del rectángulo debajo del nombre de la clase. Este comentario va entre llaves {abstract}.
- Los métodos abstractos también llevan un comentario al final.
- Los atributos y métodos final llevan el comentario {frozen}

2.6.1.1.1.3 Paquetes

- Representación

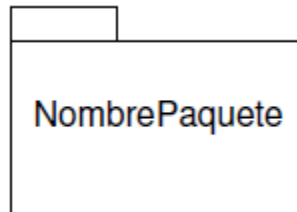


ILUSTRACIÓN 31. REPRESENTACIÓN PAQUETE UML

- Las clases que son contenidas por un paquete se representa

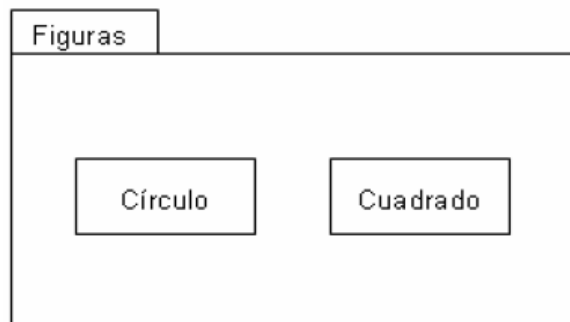


ILUSTRACIÓN 32. REPRESENTACIÓN CLASES DENTRO DE UN PAQUETE UML

- En un diagrama de clases, cuando se desea incluir una clase de otro paquete se escribe, en el compartimento correspondiente al nombre de la clase, el nombre del paquete seguido de dos puntos.

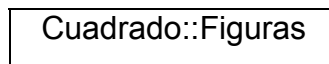


ILUSTRACIÓN 33. INDICACIÓN DE CLASES DE OTROS PAQUETES

2.7.1.1.1.4 Herencia

La herencia se representa con una línea continua que va de la clase hija a la clase padre, y que termina en un triángulo vacío.

Ejemplo:

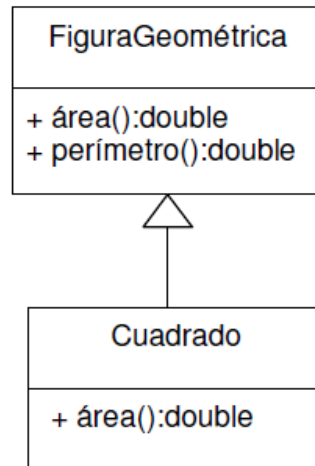


ILUSTRACIÓN 34. HERENCIA UML

En el ejemplo anterior la clase Cuadrado es hija de la clase Figura Geométrica y hereda todos sus atributos y métodos. En este caso sólo es necesario mostrar los métodos que se sobrescriben. Es decir, Cuadrado sobrescribe el método área, pero no el método perímetro, aunque tiene ambos métodos.

2.6.1.1.5 Asociación

- Una asociación (relación entre dos clases) se representa como una línea continua entre dos clases, y puede tener el nombre de la relación sobre esta línea. Ejemplo:

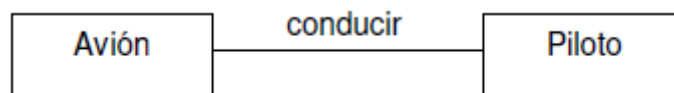


ILUSTRACIÓN 35 ASOCIACION EN UML

- Para mostrar que la relación sólo tiene un sentido se muestra con una flecha que indica el sentido de la relación.

Ejemplo:

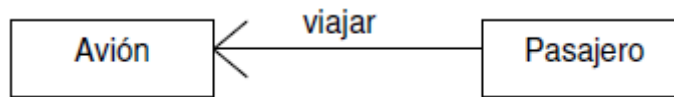


ILUSTRACIÓN 36 ASOCIACION EN UN SOLO SENTIDO EN UML

- Para determinar cuántos objetos de cada clase participan en una relación (multiplicidad), se escribe el rango de la forma: límiteInferior..límiteSuperior. Por ejemplo:

TABLA 4. MULTIPLICIDAD DE CLASE

0..1	Cero o uno (indica una relación opcional)
8-10	De ocho a diez
*	Cero o muchos

- La multiplicidad se muestra en el extremo “destino” de la relación, ejemplo:

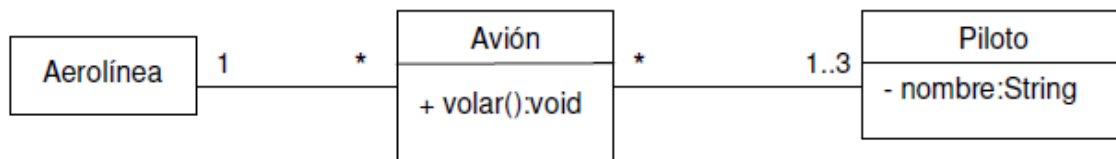


ILUSTRACIÓN 37. EJEMPLO DE MULTIPLICIDAD

2.6.1.1.6 Interfaces

Una interfaz se puede representar como una clase. En el primer compartimento del rectángulo debe ir la palabra “interfaz” encerrada en símbolos de << y >>. En los otros compartimentos, se trabaja igual que con las clases.

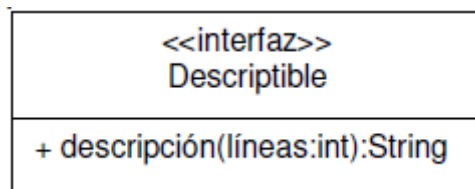


ILUSTRACIÓN 38. INTERFAZ EN UML

Para mostrar que una clase implementa una interfaz, se utiliza una línea punteada desde la clase hacia la interfaz, que termina en un triángulo vacío.

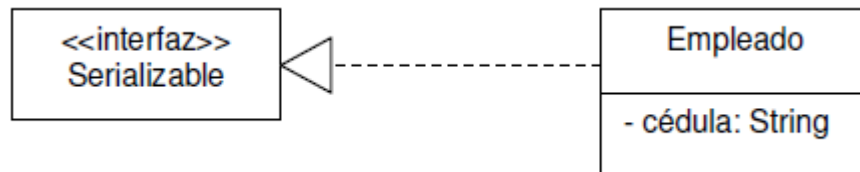


ILUSTRACIÓN 39. IMPLEMENTACIÓN DE UNA INTERFACE

2.7 PATRONES DE DISEÑO

Según Alexander “cada patrón describe un problema que ocurre una y otra vez en nuestra entorno, así como la solución a ese problema, de tal modo que se puede aplicar esta solución un millón de veces, sin hacer lo mismo dos veces.

Un patrón es un par/problema solución con nombre que se puede aplicar en nuevos contextos, con consejos acerca de cómo aplicarlo en nuevas situaciones y discusiones sobre sus compromisos

Según Gamma un patrón tiene cuatro elementos esenciales:

- Nombre del patrón que permite su identificación y para posterior referencias.
- El problema describe cuando aplicar el patrón. Explicar el problema y su contexto.
- La solución describe los elementos que constituyen el diseño, sus relaciones, responsabilidades y colaboraciones.
- Las consecuencias son los resultados así como las ventajas e inconvenientes de aplicar el patrón.

2.7.1 Objetivos de los patrones

Los patrones de diseño pretenden:

- Proporcionar catálogos de elementos reusables en el diseño de sistemas software.
- Evitar la reiteración en la búsqueda de soluciones a problemas ya conocidos y solucionados anteriormente.
- Formalizar un vocabulario común entre diseñadores.
- Estandarizar el modo en que se realiza el diseño.
- Facilitar el aprendizaje de las nuevas generaciones de diseñadores condensando conocimiento ya existente.

Asimismo, no pretenden:

- Imponer ciertas alternativas de diseño frente a otras.
- Eliminar la creatividad inherente al proceso de diseño.
- No es obligatorio utilizar los patrones, solo es aconsejable en el caso de tener el mismo problema o similar que soluciona el patrón, siempre teniendo en cuenta que en un caso particular puede no ser aplicable. Abusar o forzar el uso de los patrones puede ser un error.

2.7.2 Categorías de patrones

Según la escala o nivel de abstracción:

- Patrones de arquitectura: Aquéllos que expresan un esquema organizativo estructural fundamental para sistemas software.
- Patrones de diseño: Aquéllos que expresan esquemas para definir estructuras de diseño (o sus relaciones) con las que construir sistemas software.
- Idiomas: Patrones de bajo nivel específicos para un lenguaje de programación o entorno concreto.

2.7.3 Descripción de los Patrones Usados en el proyecto¹⁴

2.7.3.1 Singleton (Único)

Garantiza que una clase tenga una instancia, y proporciona un punto de acceso global a ella.

2.7.3.1.2 Aplicabilidad

- Solo debe haber una instancia de una clase, y éste debe ser accesible a los clientes desde un punto de acceso conocido.
- La única instancia debería ser extendida mediante herencia, y los clientes deberían ser capaces de usar una instancia sin modificar su código

2.7.3.1.3 Participantes

- Singleton
 - o define un método “instancia” que permite que los clientes accedan a su única instancia.
 - o puede ser responsable de crear su única instancia.

2.7.3.1.4 Consecuencias

- Acceso controlado a la única instancia
- Permite el refinamiento de operaciones. Se puede crear una subclase de la clase Singleton, y es fácil configurar una aplicación con una instancia de esta clase extendida.
- Permite un número variable de instancias. El patrón hace que sea fácil cambiar de opinión y permitir más de una instancia de la clase Singleton. Además podemos usar el mismo en foque para controlar el numero de instancias que usa la aplicaron. Solo se necesitaría cambiar la operación que otorga acceso a la instancia del Singleton

¹⁴ Esta seccion recopila el contenido del Libro Patrones de Diseño de Gamma y el sitio WEB http://es.wikipedia.org/wiki/Patrón_de_diseño

2.7.3.2 Adapter

Se utiliza para transformar una interfaz en otra, de tal modo que una clase que no pudiera utilizar la primera, haga uso de ella a través de la segunda.

2.7.3.2.1 Propósito

Convierte la interfaz de una clase en otra interfaz que el cliente espera. *Adapter* permite a las clases trabajar juntas, lo que de otra manera no podría hacerlo debido a sus interfaces incompatibles.

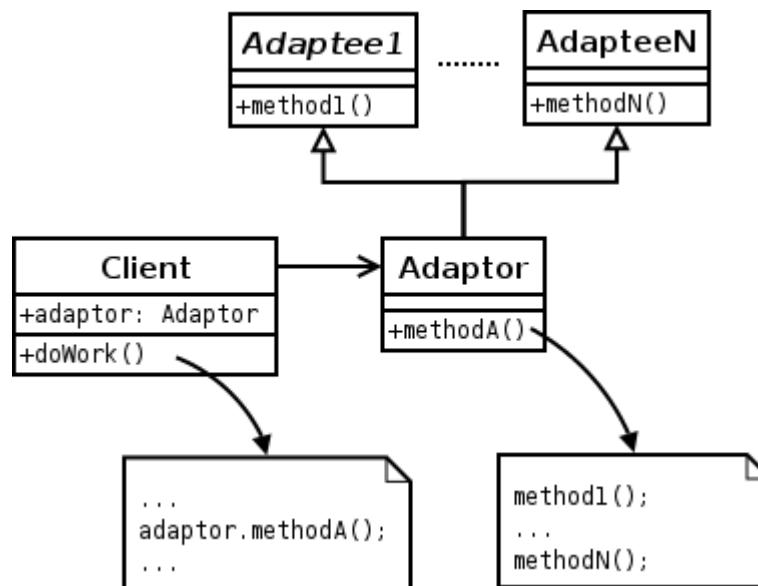


ILUSTRACIÓN 40 DIAGRAMA PATRÓN ADAPTER

2.7.3.2.2 Aplicabilidad

Usar el patrón Adapter cuando:

- Se desea usar una clase existente, y su interfaz no se iguala con la necesitada.
- Cuando se desea crear una clase reusable que coopera con clases no relacionadas, es decir, las clases no tienen necesariamente interfaces compatibles.

2.7.3.2.3 Consecuencias

*Los adaptadores de clase y objetos tienen diferentes trade-off*¹⁵

- Un adaptador de clase:
 - o Adapta *Adaptee* a *Target* encargando a una clase *Adaptee* concreta. Como consecuencia, una clase adaptadora no funcionará cuando se desea adaptar una clase y todas sus subclases.
 - o Permite a los *Adapter* sobrescribir algo de comportamiento de *Adaptee*, ya que *Adapter* es una subclase de *Adaptee*.
- Un adaptador de objeto:
 - o Permite que un único *Adapter* trabaje con muchos *Adaptees*, es decir, el *Adapter* por sí mismo y las subclases (si es que la tiene). El *Adapter* también puede agregar funcionalidad a todos los *Adaptees* de una sola vez.
 - o Hace difícil sobrescribir el comportamiento de *Adaptee*. Esto requerirá derivar *Adaptee* y hacer que *Adapter* se refiera a la subclase en lugar que al *Adaptee* por sí mismo.

Aquí hay otras cuestiones a considerar cuando se utiliza el patrón *Adapter*:

- *¿Cuanta adaptación hace el Adapter?* *Adapter* varía en la cantidad de trabajo que hace para adaptar *Adaptee* a la interfaz *Target*. Hay un espectro de trabajo posible, desde una simple conversión (por ejemplo, cambiando los nombres de las operaciones) hasta soportando un conjunto de operaciones enteramente diferentes. La cantidad de trabajo que *Adapter* hace depende de cuánto de similar tienen la interfaz *Target* con *Adaptee*.
- *Adaptadores Pluggables* Una clase es más reusable cuando minimizas la suposición que otras clases deben hacer para utilizarla. Mediante la construcción en una clase de la adaptación de una interfaz, eliminas la suposición de que otras clases ven la misma interfaz. Puesto de otra manera, la adaptación de la interfaz nos permite incorporar nuestra clase en sistemas existentes que pueden esperar diferentes interfaces para la clase.

¹⁵ *trade-off* = pros y contras, ventajas y desventajas

2.7.3.3 Bridge

El patrón **Bridge**, también conocido como Handle/Body, es una técnica usada en programación para desacoplar una abstracción de su implementación, de manera que ambas puedan ser modificadas independientemente sin necesidad de alterar por ello la otra.

Esto es, se desacopla una abstracción de su implementación para que puedan variar independientemente.

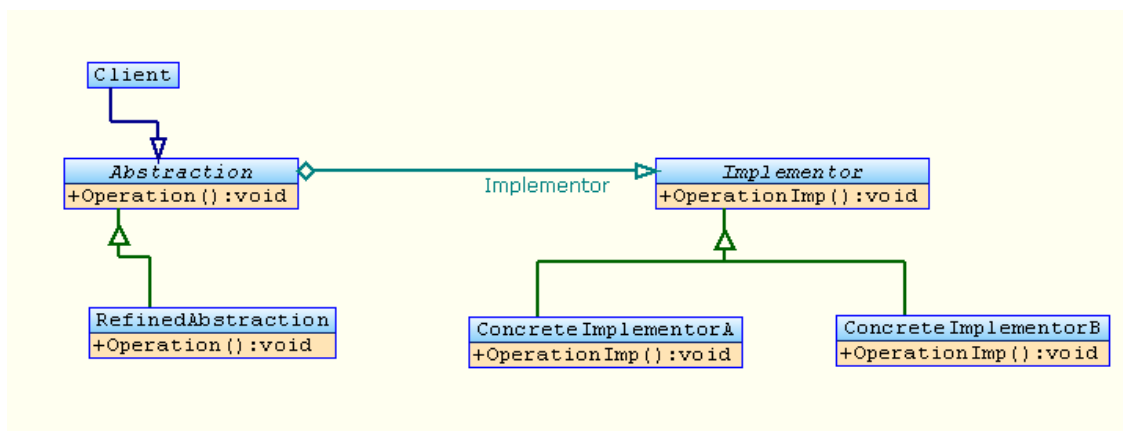


ILUSTRACIÓN 41. DIAGRAMA PATRÓN BRIDGE

2.7.3.3.1 Aplicabilidad

Se usa el patrón *Bridge* cuando:

- Se desea evitar un enlace permanente entre la abstracción y su implementación. Esto puede ser debido a que la implementación debe ser seleccionada o cambiada en tiempo de ejecución.
- Tanto las abstracciones como sus implementaciones deben ser extensibles por medio de subclases. En este caso, el patrón *Bridge* permite combinar abstracciones e implementaciones diferentes y extenderlas independientemente.
- Cambios en la implementación de una abstracción no deben impactar en los clientes, es decir, su código no debe tener que ser recompilado.

- Se desea compartir una implementación entre múltiples objetos (quizá usando contadores), y este hecho debe ser escondido a los clientes.

2.7.3.3.2 Participantes

- **Abstraction** define una interface abstracta. Mantiene una referencia a un objeto de tipo *Implementor*.
- **RefinedAbstraction** extiende la interface definida por *Abstraction*
- **Implementor** define la interface para la implementación de clases. Esta interface no se tiene que corresponder exactamente con la interfaz de *Abstraction*; de hecho, las dos interfaces pueden ser bastante diferente. Típicamente la interfaz *Implementor* provee sólo operaciones primitivas, y *Abstraction* define operaciones de alto nivel basadas en estas primitivas.
- **ConcreteImplementor** implementa la interfaz de *Implementor* y define su implementación concreta.

2.7.3.3.3 Consecuencias

- *Desacopla interfaz e implementación*: una implementación no es limitada permanentemente a una interface. La implementación de una abstracción puede ser configurada en tiempo de ejecución. Además le es posible a un objeto cambiar su implementación en tiempo de ejecución. Desacoplando *Abstraction* e *Implementor* también elimina las dependencias sobre la implementación en tiempo de compilación. Cambiar una clase de implementación no requiere recompilar la clase *Abstraction* ni sus clientes. Esta propiedad es esencial cuando te debes asegurar la compatibilidad binaria entre diferentes versiones de una biblioteca de clases. Es más, este desacoplamiento fomenta las capas, que pueden conducir a un sistema mejor estructurado. La parte de alto nivel de un sistema sólo tiene que conocer *Abstraction* e *Implementor*.
- *Mejora la extensibilidad*: se puede extender las jerarquías de *Abstracciones* *Implementor* independientemente.

- *Esconde los detalles de la implementación a los clientes.*

2.7.3.4 Composite

El patrón Composite sirve para construir objetos complejos a partir de otros más simples y similares entre sí, gracias a la composición recursiva y a una estructura en forma de árbol.

Esto simplifica el tratamiento de los objetos creados, ya que al poseer todos ellos una interfaz común, se tratan todos de la misma manera.

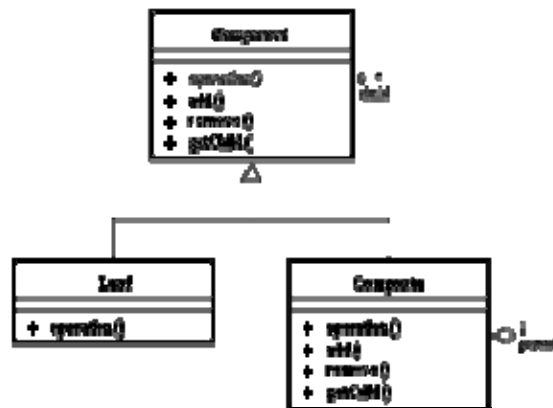


ILUSTRACIÓN 42. DIAGRAMA PATRÓN COMPOSITE

2.7.3.4.1 Aplicabilidad

- Debe ser usado cuando los clientes quieren ignorar la diferencia entre agrupaciones(composiciones) de objetos y objetos individuales

2.7.3.4.2 Participantes

- **Componente:** es la abstracción para todos los componentes, incluyendo los compuestos. Declara la interfaz para los objetos en la composición. Implementa los comportamientos por defecto común a todas las clases, como sea apropiado. Declara un interfaz para acceder y gestionar sus componentes hijo.

- Hoja: Representa las implementaciones del componente, implementa todos los métodos del componente.
- Compuesto: Representa un objeto compuesto (Componente que tiene Hijos). Implementa todos los métodos del componente, generalmente delegándolos a sus hijos.

2.7.3.4.3 Consecuencias

- Simplifica el cliente. Los clientes pueden tratar uniformemente a las estructuras compuestas y a los objetos individuales.
- Facilita la adición de nuevos tipos de componentes, si se define nuevas clases de compuestos, esta funcionan automáticamente con la estructura y código del cliente
- Un desventaja es que se dificulta el restringir los componentes de un compuesto

2.7.3.5 Facade

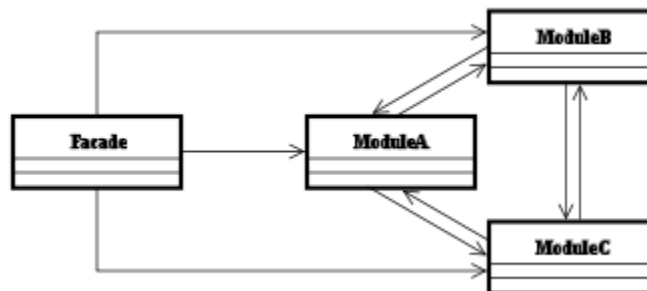


ILUSTRACIÓN 43 ESTRUCTURA DEL PATRON FACADE. TOMADO DEL SITIO WEB
[HTTP://ES.WIKIPEDIA.ORG/WIKI/FACADE_\(PATRÓN_DE_DISEÑO\)](http://es.wikipedia.org/wiki/Facade_(patrón_de_diseño))

Proporciona una interfaz unificada para un conjunto de interfaces de un subsistema, define una interfaz de alto nivel que hace que el subsistema sea mas fácil de usar.

2.7.3.5.1 Aplicabilidad

- Definir una interfaz simple para un subsistema complejo.
- Independizar subsistemas al proporcionarse una fachada para su comunicación.
- Definir un punto de entrada en cada nivel del subsistema

2.7.3.5.2 Participantes

Fachada: se encarga de delegar al subsistema que cubre las peticiones que a el se realizan

Clases del subsistema: implementa las funciones del subsistema, realiza las labores que le encomienda la fachada, no tiene referencia a la fachada

2.7.3.5.3 Consecuencias

- Facilita el uso de complejos subsistemas, ocultando al cliente los componentes que lo conforman.
- Promueve un débil acoplamiento entre el subsistema y sus clientes.
- No impide el uso de clases del subsistema en caso de ser necesario. De este modo se puede elegir entre facilidad de uso y generalidad.

2.7.3.5 Observer

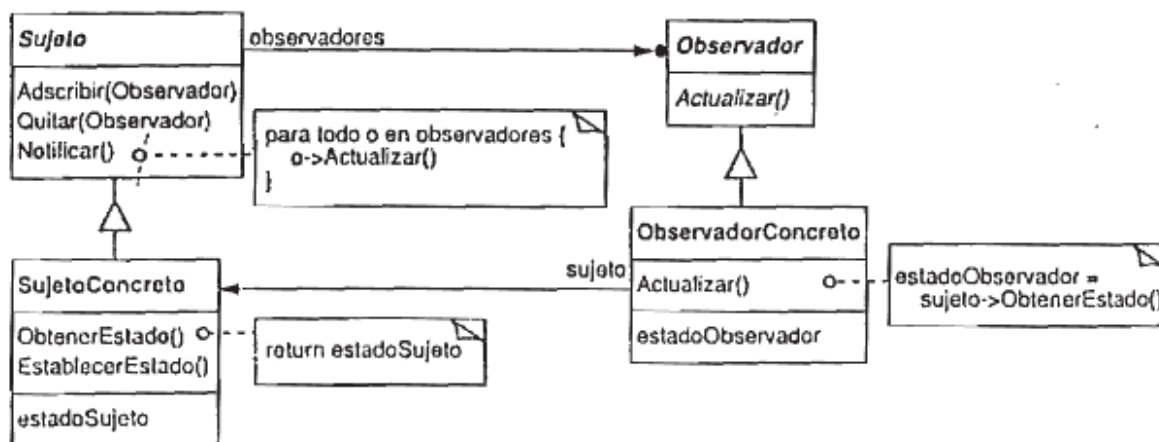


ILUSTRACIÓN 44 ESTRUCTURA PATRON OBSERVER. TOMADO DEL LIBRO PATRONES DE DISEÑO, GAMMA

Crea un vinculo entre objetos de tal forma que cuando un objeto se modifica este notifica a todos los objetos que se encuentren relacionados con el.

2.7.3.5.1 Aplicabilidad

- Cuando producto de una modificación de un objeto otro requiera se modificado también

- Cuando se necesite que uno objeto notifique a otro sin saber que objetos son a los que tiene que informar

2.7.3.5.2 Participantes

Sujeto: proporciona una interfaz para asignar y quitar objetos observadores

Observador: define la interfaz para actualizar los objetos que deben ser notificados ante cambios de un sujeto

SujetoConcreto: envía una notificación a sus observadores cuando cambia su estado

ObservadorConcreto: implementa la interfaz de actualización del observador para mantener su estado consistente con el del sujeto

2.7.3.5.3 Consecuencias

- Permite el modificar a los sujetos y observadores de forma independiente.
- El sujeto no conoce la clase concreta de ningun observador. Por tanto el acoplamiento entre sujeto y observadores es minima

2.7.3.6 Modelo Vista Controlador

Modelo Vista Controlador (MVC) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

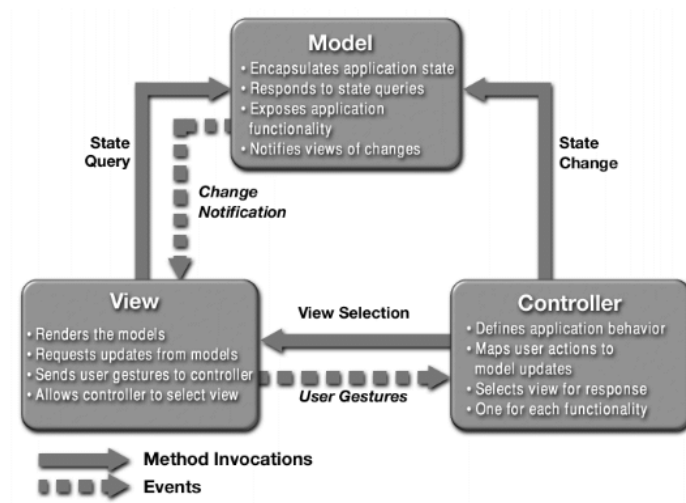


ILUSTRACIÓN 45. ARQUITECTURA MVC

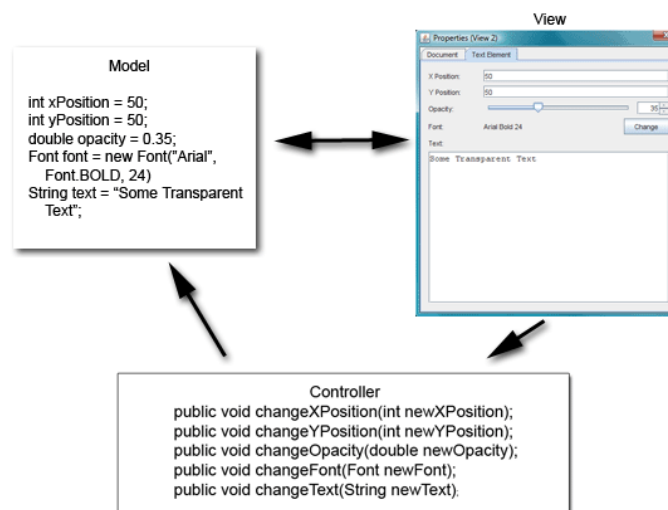


ILUSTRACIÓN 46. UNA APLICACIÓN JAVA SE EMPLEANDO MVC

La principal característica del MVC es que el Modelo, las Vistas y los Controladores se tratan como entidades separadas; esto hace que cualquier cambio producido en el Modelo se refleje automáticamente en cada una de las Vistas.

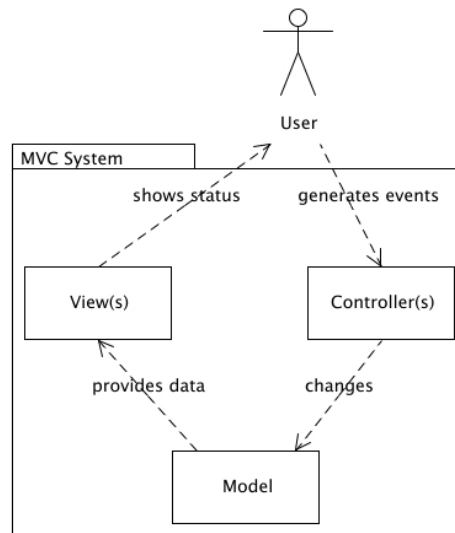


ILUSTRACIÓN 47. MVC EN UML

El *Modelo* es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el Modelo.

La *Vista* es el objeto que maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa con el Modelo a través de una referencia al propio Modelo.

El *Controlador* es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo.

Diseñar una aplicación siguiendo la arquitectura MVC literalmente no es muy recomendable. Por ejemplo en programas de que donde se requiere un elevado manejo de grafica , como un juego, posiblemente se junten las clases de la Vista y el Modelo mucho más de lo que se esperaría al usar MVC, adicionalmente con frecuencia se combina la vista con el Controlados [Swing-Java].

Flujo de una Aplicación usa MVC

Aunque se pueden encontrar diferentes implementaciones de MVC, el flujo que sigue el control generalmente es el siguiente:

1. El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace)
2. El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler) o callback.
3. El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario (por ejemplo, el controlador actualiza el carro de la compra del usuario). Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.
4. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo (por ejemplo, produce un listado del contenido del carro de la compra). El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, el patrón de observador puede ser utilizado para proveer cierta indirección entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier cambio. Un objeto vista puede registrarse con el modelo y esperar a los cambios, pero aun así el modelo en sí mismo sigue sin saber nada de la vista. El controlador no pasa objetos de dominio (el modelo) a la vista aunque puede dar la orden a la vista para que se actualice. Nota: En algunas implementaciones la vista no tiene acceso directo al modelo, dejando que el controlador envíe los datos del modelo a la vista.
5. La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

2.7.3.6.1 Aplicabilidad

- Sistemas CAD
- Aplicaciones Web
- Gestión de base de datos

2.7.3.6.2 Consecuencias

Ventajas del patrón

- Construcción de software sostenible.
- Localización ágil de errores.
- Diseño modular, poco acoplado, favoreciendo la reutilización.
- Hay una clara separación entre los componentes de un programa; lo cual nos permite implementarlos por separado
- La conexión entre el Modelo y sus Vistas es dinámica; se produce en tiempo de ejecución, no en tiempo de compilación.
- La aplicación está implementada modularmente.
- Sus vistas muestran información actualizada siempre.
- El programador no debe preocuparse de solicitar que las vistas se actualicen, ya que este proceso es realizado automáticamente por el modelo de la aplicación.
- Si se desea hacer una modificación al modelo del dominio, como aumentar métodos o datos contenidos, sólo debe modificarse el modelo y las interfaces del mismo con las vistas, no todo el mecanismo de comunicación y de actualización entre modelos.
- Las modificaciones a las vistas no afectan en absoluto a los otros módulos de la aplicación.
- MVC es bastante utilizado en la actualidad en marcos de aplicación orientados a objeto desarrollados para construir aplicaciones de gran tamaño; Java Swing, Apache Struts, Microsoft ASP.NET, las transformaciones XSL o incluso los documentos LATEX siguen este patrón de diseño.

- MVC está demostrando ser un patrón de diseño bien elaborado pues las aplicaciones que lo implementan presentan una extensibilidad y una mantenibilidad únicas comparadas con otras aplicaciones basadas en otros patrones

Fortalezas

- Se presenta la misma información de distintas formas.
- Las vistas y comportamiento de una aplicación deben reflejar las manipulaciones de los datos de forma inmediata.
- Debería ser fácil cambiar la interfaz de usuario (incluso en tiempo de ejecución).
- Permitir diferentes estándares de interfaz de usuario o portarla a otros entornos no debería afectar al código de la aplicación.

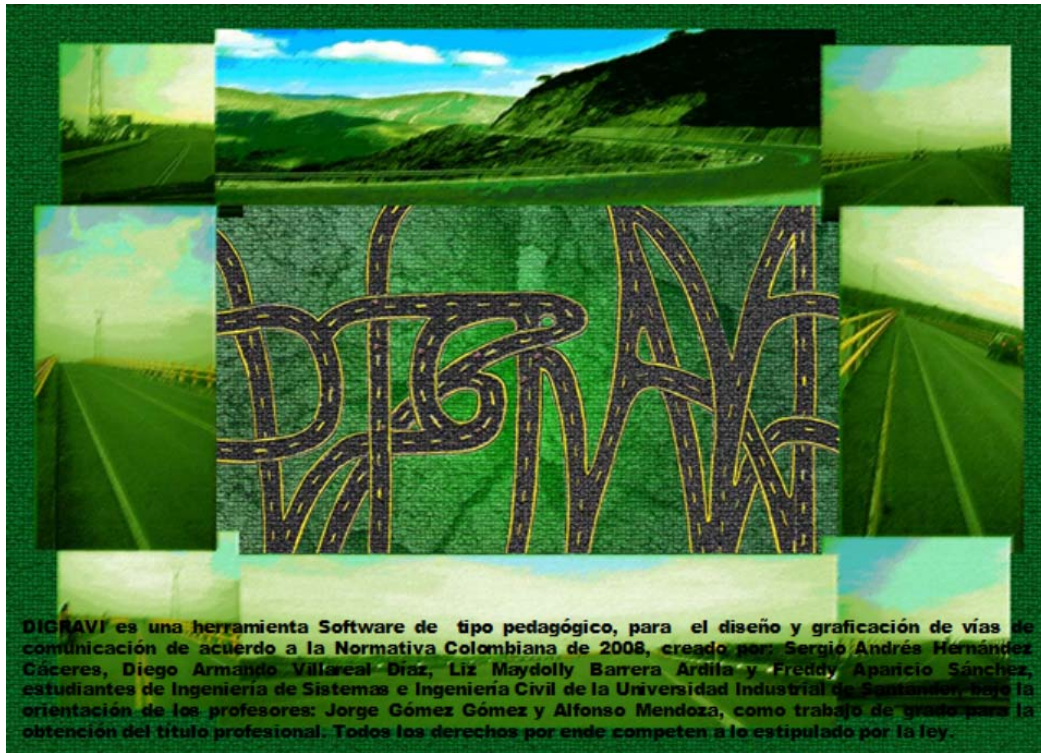
Desventajas:

- El tiempo de desarrollo de una aplicación que implementa el patrón de diseño MVC es mayor, al menos en la primera etapa, que el tiempo de desarrollo de una aplicación que no lo implementa, ya que MVC requiere que el programador implemente una mayor cantidad de clases que en un entorno de desarrollo común no son necesarias. Sin embargo, esta desventaja es muy relativa ya que posteriormente, en la etapa de mantenimiento de la aplicación, una aplicación MVC es muchísimo más sostenible, extensible y modificable que una aplicación que no lo implementa.
- MVC requiere la existencia de una arquitectura inicial sobre la que se deben construir clases e interfaces para modificar y comunicar los módulos de una aplicación. Esta arquitectura inicial debe incluir, por lo menos: un mecanismo de eventos para poder proporcionar las notificaciones que genera el modelo de aplicación; una clase Modelo, otra clase Vista y una clase Controlador genéricas que realicen todas las tareas de comunicación, notificación y actualización que serán luego transparentes para el desarrollo de la aplicación.

- MVC es un patrón de diseño orientado a objetos por lo que su implementación es sumamente costosa y difícil en lenguajes que no siguen este paradigma.

3. USO DE LA APLICACIÓN

DIGRAVI es una herramienta pensada para la asistencia de proyectos viales. El usuario puede desplazarse a través de sus módulos e ir completando cada etapa de diseño vial haciendo uso de la información de etapas anteriores de una forma



secuencial.

ILUSTRACIÓN 48 PRESENTACIÓN DIGRAVI

3.1. DESARROLLO DE UN PROYECTO VIAL EN DIGRAVI

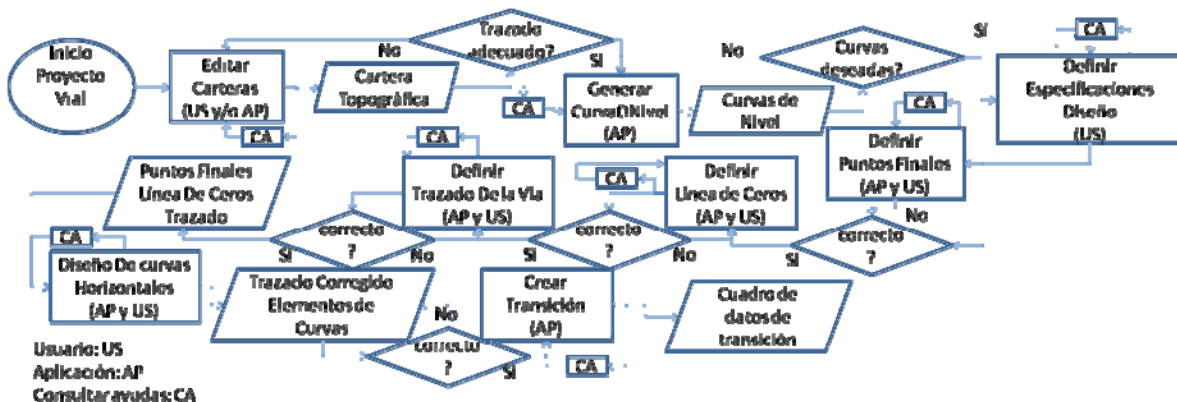


ILUSTRACIÓN 49 DIAGRAMA DE FLUJO PROYECTO VIAL DIGRAVI

El proyecto vial comienza sobre el terreno en el cual se va a diseñar. En este terreno se debe hacer un levantamiento topográfico, el cual quedará consignado en una cartera de campo que puede realizarse haciendo uso de los métodos de distancias fijas y cota redonda, procedimientos que se explican en detalle en la sección 2.2.1 *Levantamiento topográfico*. La aplicación a partir de ésta cartera recibida genera las curvas de nivel del terreno, las cuales son el resultado del primer módulo.

El segundo módulo es utilizado para definir las especificaciones del proyecto vial. Los valores de especificación serán usados a lo largo de todo el proyecto para realizar cálculos en los siguientes módulos. En éste módulo es importante el uso de las ayudas del proyecto para apoyar las decisiones del diseñador.

Después sigue la etapa de diseño horizontal la cual se separará en los sub módulos de alineamiento horizontal y diseño de curvas horizontales, en el alineamiento se utilizan los valores de especificación para definir el abscisado y el ancho del trazado de la vía, se marcan los puntos de inicio y fin del proyecto vial, la línea de ceros y el trazado de la vía. Sobre el trazado de proyecto el usuario puede diseñar las curvas horizontales que más sean convenientes para el trazado propuesto, escogiendo un tipo de curva y proponiendo sus parámetros de entrada. Cuando Se han diseñado las curvas horizontales es posible generar la transición para cada curva sobre el eje izquierdo, eje derecho y eje central.

3.1.1. Generación de Terreno

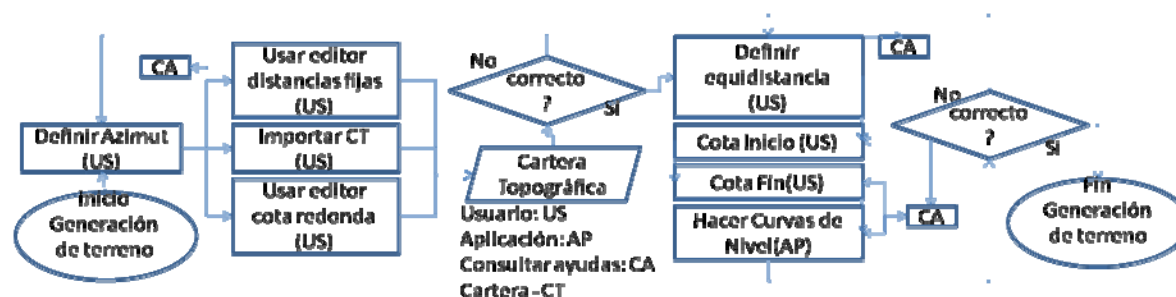


ILUSTRACIÓN 50 DIAGRAMA DE GENERACIÓN DE TERRENO DIGRAVI

En este módulo puede importarse una cartera en los formatos permitidos de cota redonda o distancias fijas, también es posible editarla internamente dentro de la aplicación. Para generar la cartera se requiere dar una orientación al terreno la cuál está dada por el azimut . Una vez creada la cartera, la aplicación deberá usar su modelo matemático para transformar los puntos de la notación de cada cartera en coordenadas rectangulares tridimensionales, las cuales serán usadas por el método de generación de curvas de nivel.

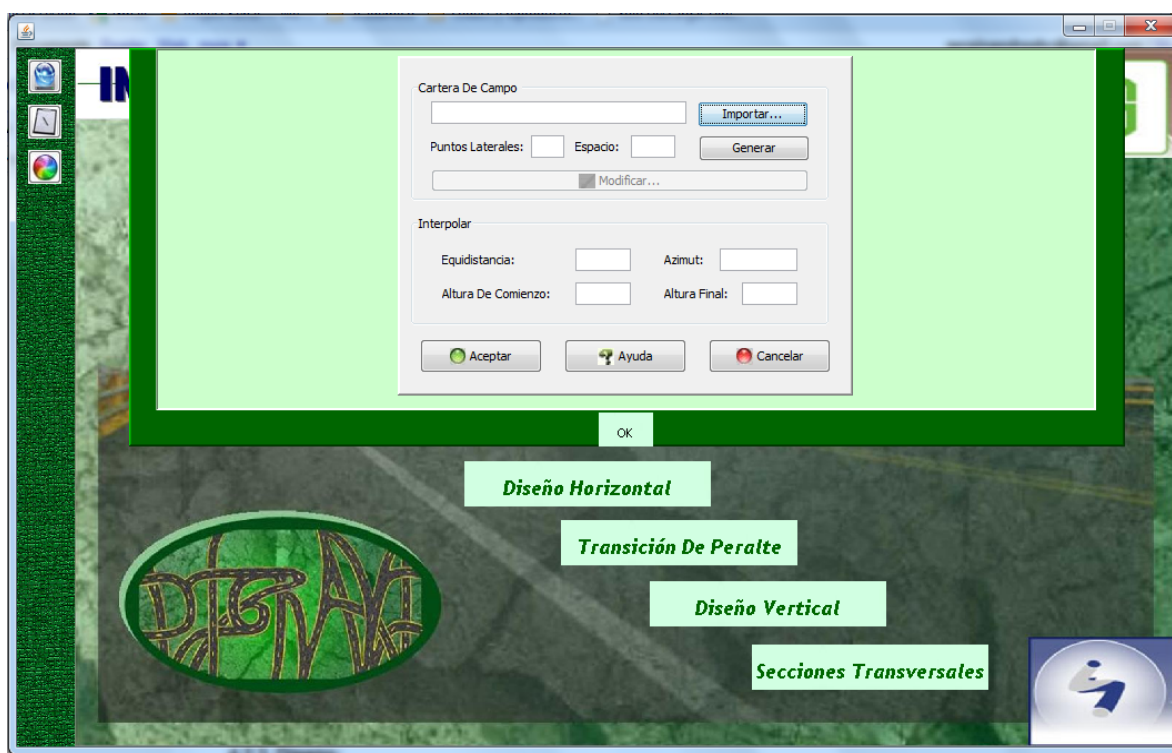


ILUSTRACIÓN 51. MENÚ PARA LA GENERACIÓN DE TERRENO

3.1.1.1. Ejemplo:

Se cargarán las carteras topográficas suministradas en ANEXO 2 CARTERAS. Primero se dirige a la cacilla azimut y se pone el ángulo medido en sentido horario desde al norte geográfico hasta el primer tramo del levantamiento topográfico, como se puede ver en la *Ilustración 47(Plano De Ejemplo) del plano del terreno el valor del azimut es de 7°*.

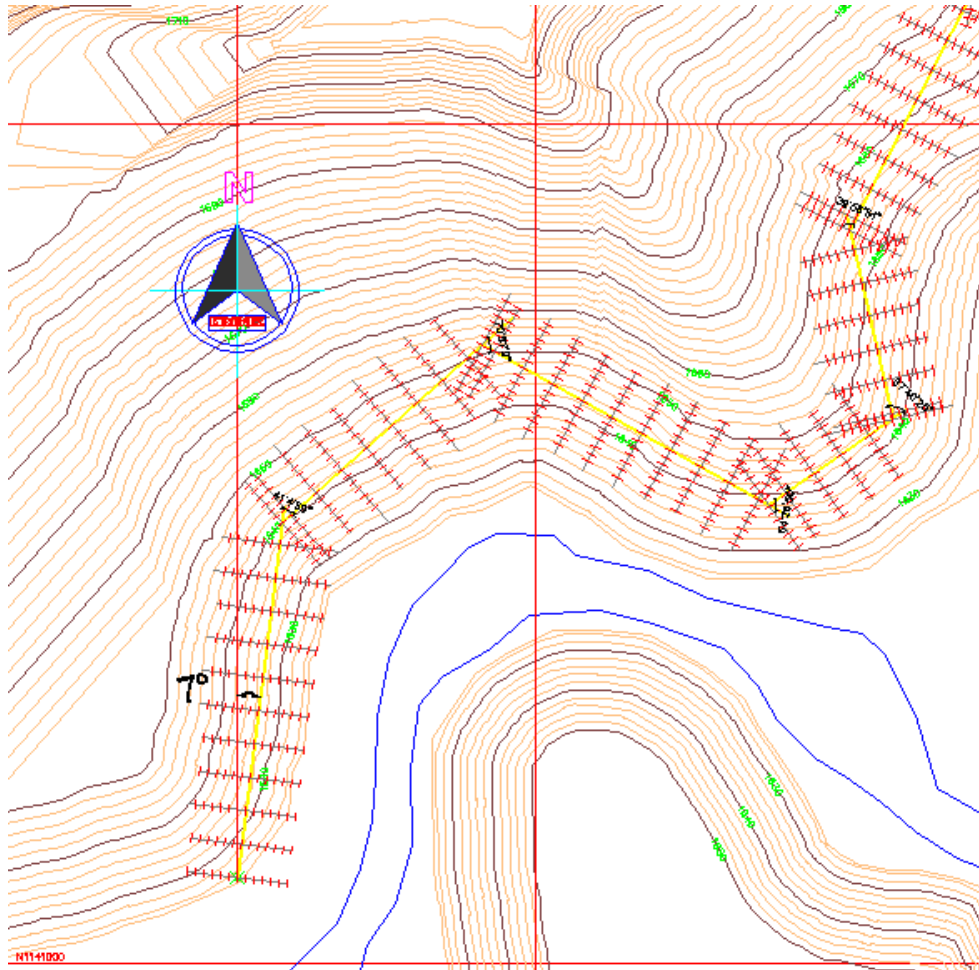


ILUSTRACIÓN 52 PLANO DE EJEMPLO

Después de hacer esto se debe ir al botón importar y accionarlo, la aplicación desplegará un selector de archivos en el cuál es posible buscar el archivo y abrirlo

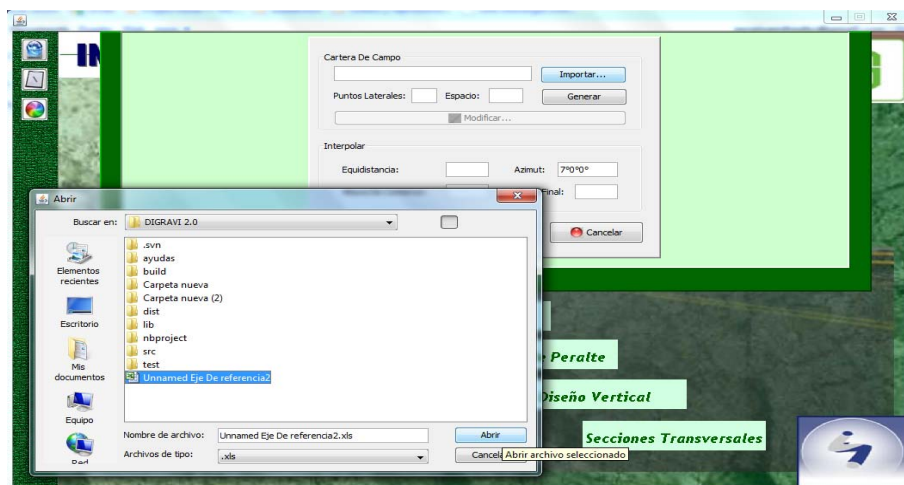
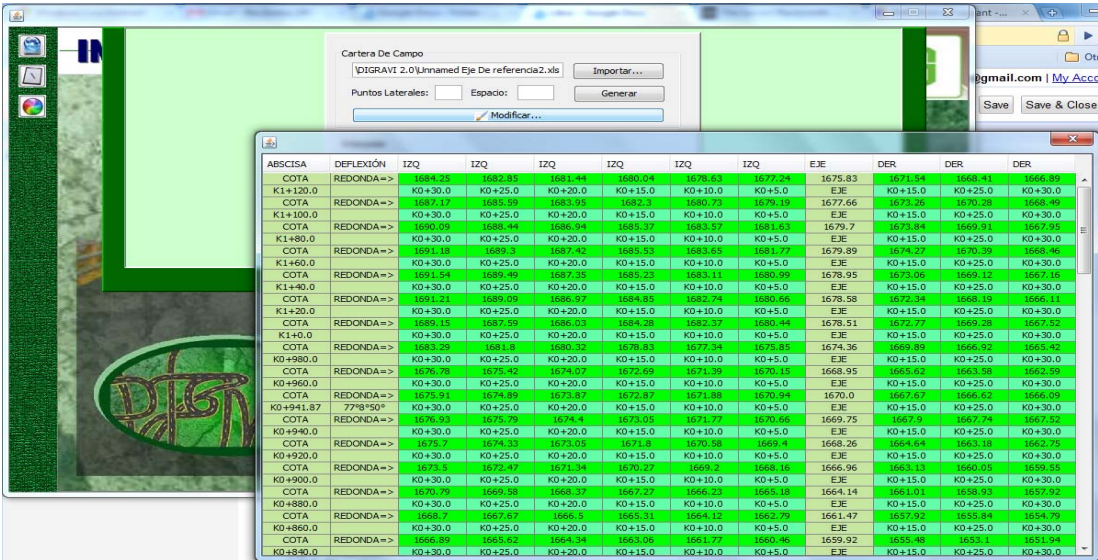


ILUSTRACIÓN 53. CARGANDO UNA CARTERA EN FORMATO EXCEL

Una vez seleccionado el archivo, este se carga dentro de la aplicación y con este crea el eje de referencia. Si no se presenta ningún problema con el archivo se habilitará el botón editar de lo contrario mostrará un mensaje con el error ocurrido. Editar despliega el editor de carteras de la aplicación con los datos de la cartera anteriormente cargada para editarlos.



The screenshot shows a software window titled 'Cartera De Campo'. It contains a text field with the path 'D:\GRAVI 2.0\Unnamed Eje De referencia2.xls', an 'Importar...' button, and 'Puntos Laterales:' labels with input fields for 'Espacio:' and 'Generar'. Below these is a 'Modificar...' button. In the background, a data table is visible with columns for stationing and deflection angles.

ABSCISA	DEFLEXIÓN =>	IZQ	IZQ	IZQ	IZQ	IZQ	IZQ	EJE	DER	DER	DER
COTA	REDONDA=>	1684.25	1682.85	1681.44	1680.04	1678.63	1677.24	1675.83	1671.34	1668.41	1666.89
K1+120.0		KD+30.0	KD+25.0	KD+20.0	KD+15.0	KD+10.0	KD+5.0	EJE	KD+15.0	KD+25.0	KD+30.0
COTA	REDONDA=>	1687.17	1685.99	1684.93	1683.95	1682.95	1681.95	1677.66	1673.26	1670.28	1668.46
K1+100.0		KD+30.0	KD+25.0	KD+20.0	KD+15.0	KD+10.0	KD+5.0	EJE	KD+15.0	KD+25.0	KD+30.0
COTA	REDONDA=>	1690.09	1688.44	1686.94	1685.37	1683.57	1681.63	1679.7	1673.84	1669.91	1667.95
K1+80.0		KD+30.0	KD+25.0	KD+20.0	KD+15.0	KD+10.0	KD+5.0	EJE	KD+15.0	KD+25.0	KD+30.0
COTA	REDONDA=>	1691.18	1689.3	1687.84	1686.55	1685.05	1683.27	1679.89	1674.07	1670.09	1668.46
K1+60.0		KD+30.0	KD+25.0	KD+20.0	KD+15.0	KD+10.0	KD+5.0	EJE	KD+15.0	KD+25.0	KD+30.0
COTA	REDONDA=>	1691.54	1689.49	1687.35	1685.23	1683.11	1680.99	1678.95	1673.06	1669.12	1667.16
K1+40.0		KD+30.0	KD+25.0	KD+20.0	KD+15.0	KD+10.0	KD+5.0	EJE	KD+15.0	KD+25.0	KD+30.0
COTA	REDONDA=>	1691.21	1689.09	1686.97	1684.85	1682.74	1680.66	1678.58	1672.64	1668.19	1666.11
K1+20.0		KD+30.0	KD+25.0	KD+20.0	KD+15.0	KD+10.0	KD+5.0	EJE	KD+15.0	KD+25.0	KD+30.0
COTA	REDONDA=>	1689.15	1687.95	1686.03	1684.26	1682.37	1680.44	1678.51	1672.77	1669.28	1667.52
K1+0.0		KD+30.0	KD+25.0	KD+20.0	KD+15.0	KD+10.0	KD+5.0	EJE	KD+15.0	KD+25.0	KD+30.0
COTA	REDONDA=>	1683.29	1681.8	1680.32	1678.85	1677.34	1675.85	1674.36	1669.89	1666.92	1665.42
KD+980.0		KD+30.0	KD+25.0	KD+20.0	KD+15.0	KD+10.0	KD+5.0	EJE	KD+15.0	KD+25.0	KD+30.0
COTA	REDONDA=>	1676.78	1675.42	1674.07	1672.69	1671.39	1670.15	1668.95	1665.42	1663.38	1662.59
KD+960.0		KD+30.0	KD+25.0	KD+20.0	KD+15.0	KD+10.0	KD+5.0	EJE	KD+15.0	KD+25.0	KD+30.0
COTA	REDONDA=>	1675.91	1674.89	1673.87	1672.87	1671.88	1670.94	1670.0	1667.67	1666.62	1666.09
KD+941.87	77°18'50"	KD+30.0	KD+25.0	KD+20.0	KD+15.0	KD+10.0	KD+5.0	EJE	KD+15.0	KD+25.0	KD+30.0
COTA	REDONDA=>	1676.93	1675.79	1674.4	1673.05	1671.77	1670.66	1669.75	1667.9	1667.74	1667.52
KD+940.0		KD+30.0	KD+25.0	KD+20.0	KD+15.0	KD+10.0	KD+5.0	EJE	KD+15.0	KD+25.0	KD+30.0
COTA	REDONDA=>	1675.7	1674.33	1673.05	1671.8	1670.58	1669.4	1668.26	1664.64	1663.18	1662.75
KD+920.0		KD+30.0	KD+25.0	KD+20.0	KD+15.0	KD+10.0	KD+5.0	EJE	KD+15.0	KD+25.0	KD+30.0
COTA	REDONDA=>	1673.5	1672.47	1671.34	1670.27	1669.2	1668.16	1666.96	1663.15	1660.05	1659.55
KD+900.0		KD+30.0	KD+25.0	KD+20.0	KD+15.0	KD+10.0	KD+5.0	EJE	KD+15.0	KD+25.0	KD+30.0
COTA	REDONDA=>	1670.75	1669.58	1668.37	1667.27	1666.23	1665.18	1664.14	1661.01	1658.93	1657.92
KD+880.0		KD+30.0	KD+25.0	KD+20.0	KD+15.0	KD+10.0	KD+5.0	EJE	KD+15.0	KD+25.0	KD+30.0
COTA	REDONDA=>	1668.7	1667.67	1666.5	1665.31	1664.12	1662.95	1661.47	1657.92	1655.84	1654.75
KD+860.0		KD+30.0	KD+25.0	KD+20.0	KD+15.0	KD+10.0	KD+5.0	EJE	KD+15.0	KD+25.0	KD+30.0
COTA	REDONDA=>	1666.95	1665.62	1664.34	1663.06	1661.77	1660.46	1659.92	1655.48	1653.1	1651.94
KD+840.0		KD+30.0	KD+25.0	KD+20.0	KD+15.0	KD+10.0	KD+5.0	EJE	KD+15.0	KD+25.0	KD+30.0

ILUSTRACIÓN 54. VISUALIZACIÓN DE UNA CARTERA CARGADA PREVIAMENTE

Los editores de las carteras funcionan bajo comandos:

- ENTER permite agregar una fila sobre la casilla que esté seleccionada.
- CONTROL + D es usado para remover la fila de la casilla que se encuentra seleccionada.
- CONTROL + I inserta una columna a la derecha de la casilla seleccionada
- CONTROL + R remueve la columna de la casilla que se encuentre seleccionada.
- CONTROL+C copiar
- CONTROL+V pegar
- SHIFT + FLECHAS selección de múltiples casillas
- CONTROL +ENTER Crea el eje de referencia a partir de la cartera lo guarda en el proyecto vial y cierra el editor.

En cualquier momento durante el uso de este módulo es posible accionar el botón de ayuda el cual nos mostrará la ayuda concerniente a la generación de terreno.

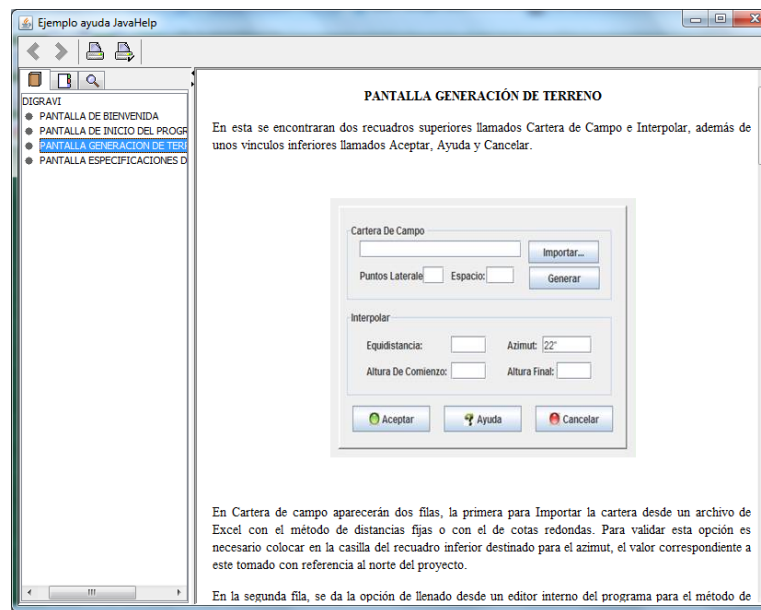


Ilustración 55. Ayuda sobre la Generación de terreno

Es posible suministrar algunos datos para el valor de la equidistancia, las alturas de comienzo y final, en caso de estar vacíos, y accionarse el botón aceptar. Se usan los valores por defecto: 2 para el valor de equidistancia y las cotas mínima y máxima que se encuentren en la cartera; al accionar el botón aceptar deberán visualizarse las curvas de nivel del terreno.

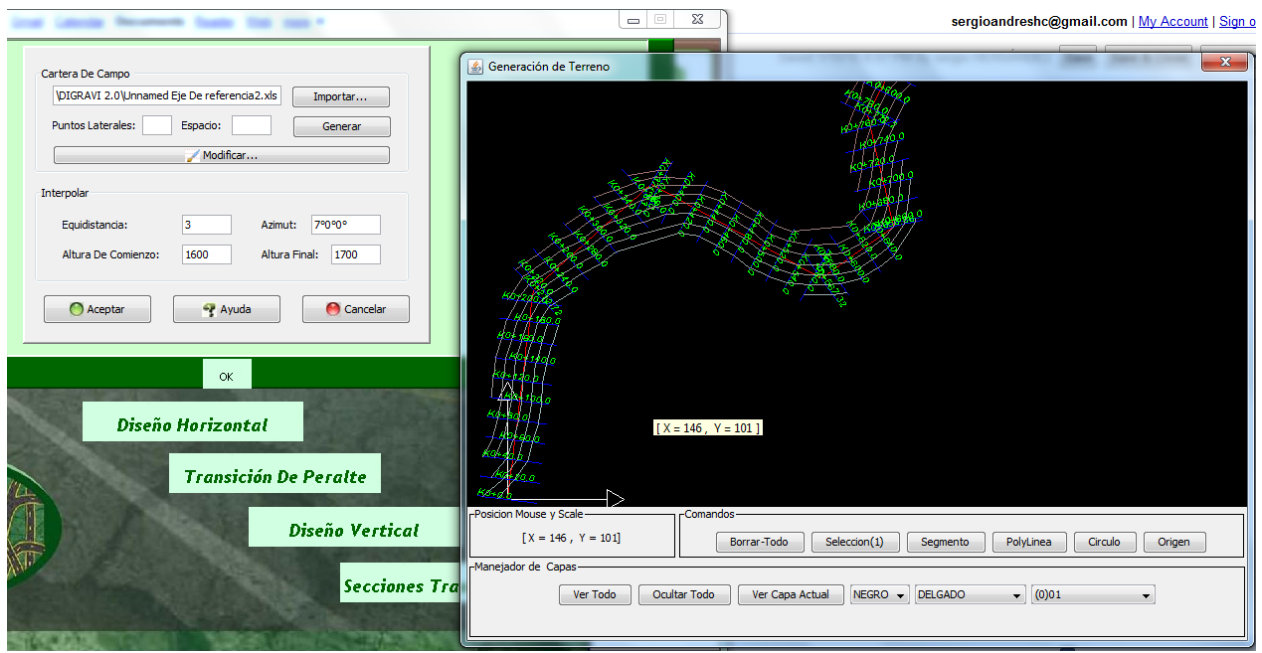


ILUSTRACIÓN 56. VISUALIZACIÓN DE LAS CURVAS DE NIVEL RESULTANTES DE CARTERA

3.1.2. Especificaciones de Proyecto

The screenshot shows a software window titled 'Especificaciones de Proyecto'. It contains several input fields and buttons. The 'Nombre Del Proyecto' field is 'Proyecto Sin Titulo 1'. The 'Tipo De vía' dropdown is set to 'PRINCIPAL DE DOS CALZADAS'. There are input fields for 'Velocidad De Diseño [Km/h]', 'Ancho De Banca [m]', 'Pendiente Maxima Horizontal [%]', 'Pendiente Minima Horizontal [%]', 'Tipo De terreno' (set to 'Por Definir Generación de terreno'), 'Peralte Máximo e.max [%]', 'Bombeo [%]', 'Longitud Circular Minima [m]', 'Longitud Espiral Minima [m]', 'Longitud Vertical Minima [m]', 'Abcisa [m]', 'Sobreancho [m]', and 'Punto De Referencia'. The 'Punto De Referencia' section includes fields for 'Norte [Mm]', 'Este [Mm]', and 'Altura [Mm]'. At the bottom, there are 'Cancelar' and 'Aceptar' buttons.

ILUSTRACIÓN 57 ESPECIFICACIONES DE PROYECTO

Al realizar el diseño de un proyecto vial cuestiones como el terreno, tipo de vía y la normativa vigente determina la manera en que se debe generar los diferentes elementos que componen el diseño, sin embargo esta labor depende en gran

medida de la pericia el diseñador, el cual con su experiencia debe elegir los parámetros que mejor se ajusten al diseño a realizar.

El objetivo de las especificaciones es el de definir las características globales del proyecto, permitiendo el realizar sugerencia al diseñador y configurar el comportamiento de la aplicación en el desarrollo de etapas posteriores.

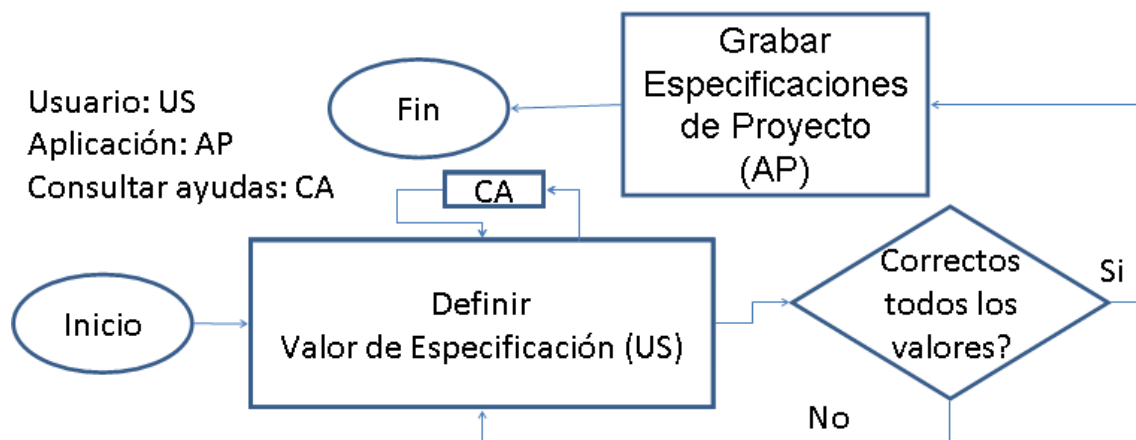


ILUSTRACIÓN 58 DIAGRAMA ESPECIFICACIONES DE PROYECTO DIGRAVI

3.1.2.1 Ejemplo:

Con el objetivo de guiar al usuario inexperto en el diseño de vías, junto a cada uno de los campos en los que se define las características générale del proyecto se encuentra un botón con un interrogante , al oprimirlo se despliega la ventana que contiene las ayudas de la aplicación, en ellas se encuentra información que orienta al usuario para que tome la decisión que más se ajuste a sus necesidades.

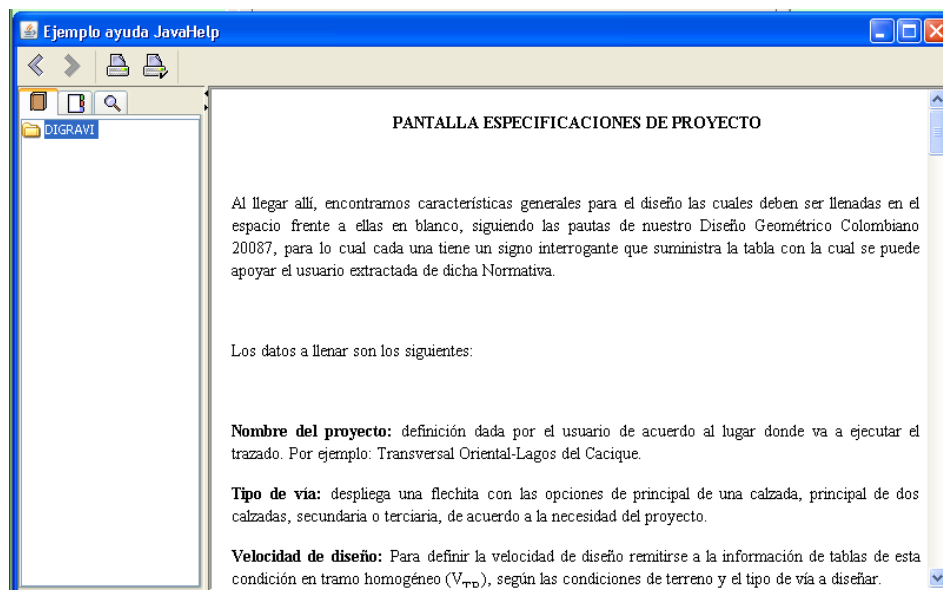


ILUSTRACIÓN 59 EJEMPLO DE VENTANA DE AYUDA

Al consultar con los ingenieros civiles Liz y Freddy, recomendaron el definir las especificaciones de seño como se muestra a continuación:

TABLA 5 VALORES EJEMPLO ESPECIFICACIONES DE PROYECTO

Campo	Valor
Nombre Proyecto	Vía Vereda Tuna
Tipo de vía	Secundaria
Velocidad de Diseño	70
Ancho Máximo Horizontal	7
Pendiente Máximo Horizontal	7
Pendiente Minima Horizontal	0.5
Peralte Máximo	8
Bombeo	2
Longitud Circular Minima	40
Longitud Espiral Minima	45
Longitud Vertical Minima	195
Abscisado	20
Sobreancho	3
Norte	50
Este	50
Altura	1624.94

Una vez definidas las especificaciones se oprime el botón aceptar, al hacerlo se obtiene el siguiente mensaje de confirmación (ILUSTRACIÓN 55 MENSAJE DE CONFIRMACIÓN), en caso de querer cambiar la configuración se puede oprimir el botón cancelar con lo cual se borran todos los campos

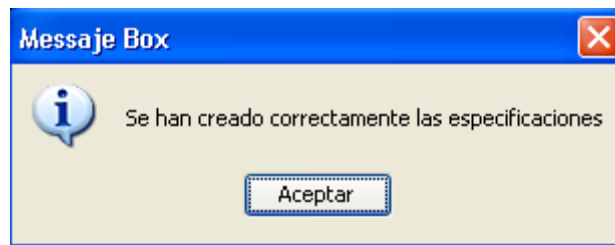


ILUSTRACIÓN 60 MENSAJE DE CONFIRMACIÓN

3.1.3. Diseño Horizontal

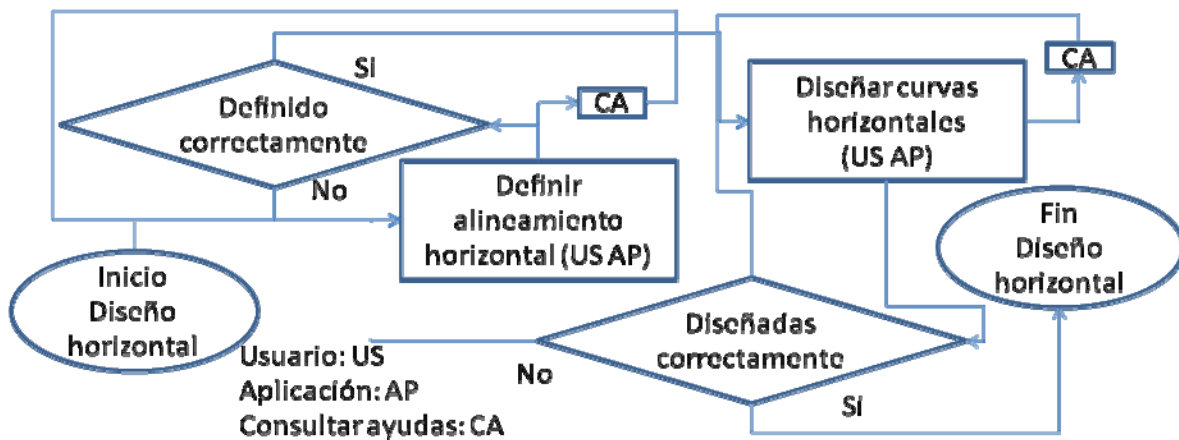


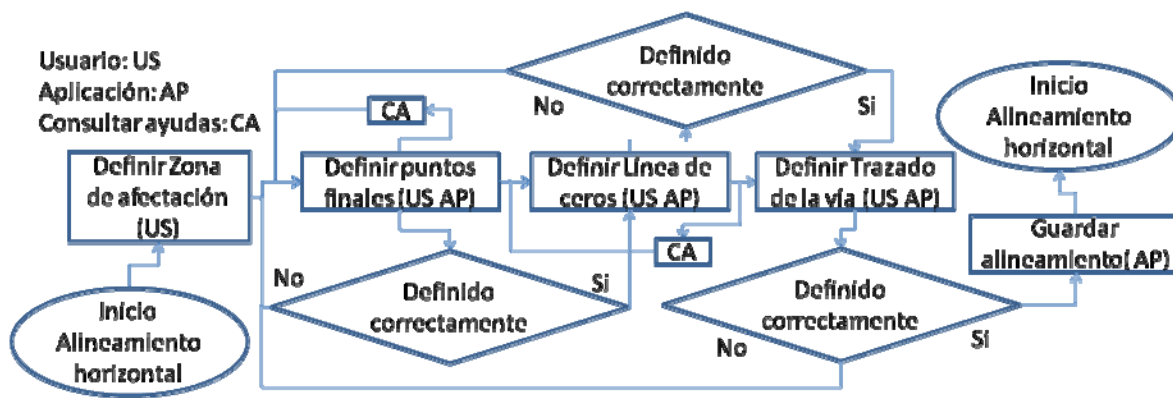
ILUSTRACIÓN 5 DIAGRAMA DISEÑO HORIZONTAL DIGRAVI

El diseño horizontal se compone básicamente de dos submódulos alineamiento donde definimos el trazado de la vía por donde se va a construir, y el módulo de diseño de curvas horizontales, que permite diseñar las curvas del trazado definido anteriormente.



ILUSTRACIÓN 61 DISEÑO HORIZONTAL

3.1.3.1. Alineamiento Horizontal



El sub módulo de alineamiento comienza definiendo por defecto el ancho de banca de especificaciones como ancho del trazado, el usuario ingresa un valor de zona de afectación a su criterio, y a continuación prosigue a definir los puntos De inicio y fin del proyecto vial sobre un lienzo que muestra las curvas de nivel del terreno, si el usuario está satisfecho con los puntos definidos puede pasar a realizar la línea de ceros uniendo el punto de inicio de la vía con el punto final saltando a través de las curvas de nivel. Finalmente el usuario dibuja el trazado procurando acomodarse a la línea de ceros, en cualquier momento del proceso el usuario puede acceder a las ayudas.

3.1.3.1.1. Ejemplo:

A partir del trabajo obtenido en los módulos de Especificaciones y Generación de terreno es posible realizar el alineamiento, primero el al seleccionar la cacilla de zona de afectación se carga el valor 6,6 a partir de la banca definida en el módulo anterior y se escoge la opción **1. Puntos Finales**.

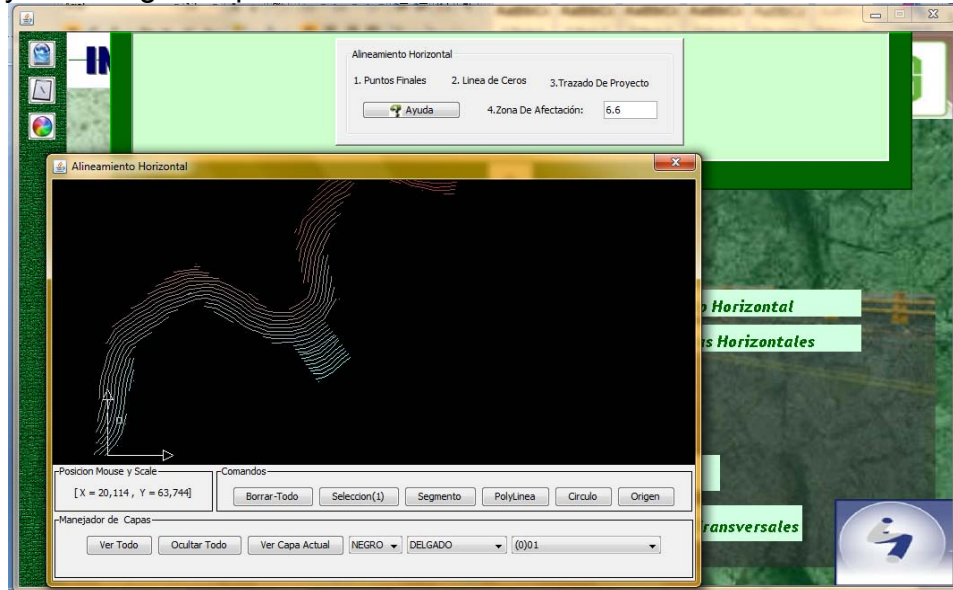


ILUSTRACIÓN 62 PUNTOS FINALES

Sobre la gráfica de las curvas de nivel del terreno es posible posicionar los puntos iniciales y finales de la vía diseñar haciendo uso del mouse, al terminar la aplicación pregunta al usuario si debe guardarlos, si la respuesta es afirmativa se prosigue a definir la línea de zeros.

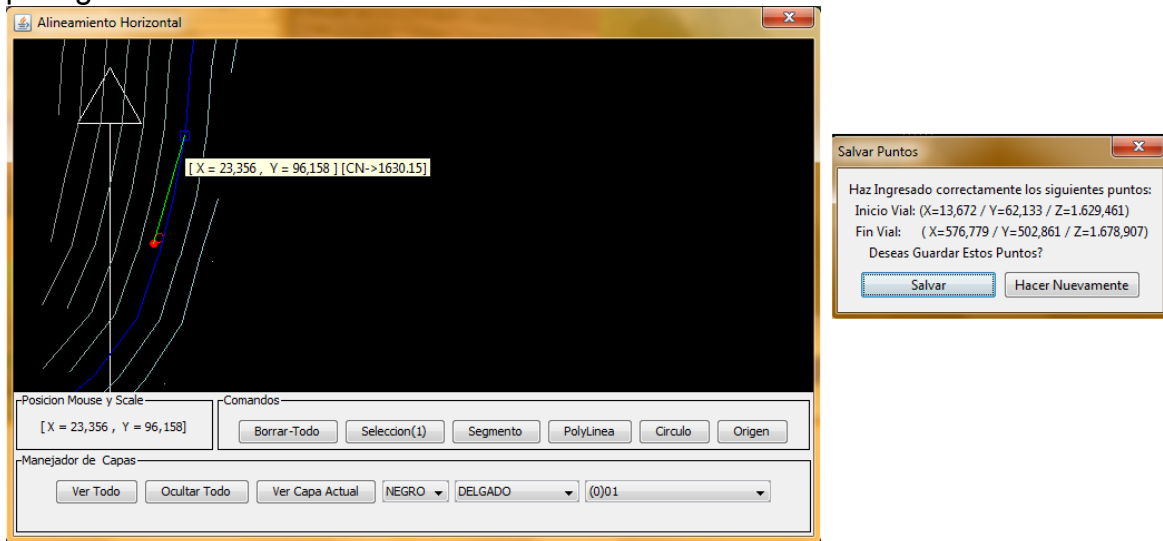


ILUSTRACIÓN 63 LÍNEA DE CEROS

La línea de va uniendo las curvas de nivel del terreno, conforme se mueve el cursor esta línea cambia de color cuando la pendiente excede la normativa, y cuando se posiciona sobre terreno sin información, cuando la línea de zeros se lleva hasta el punto final el punto se hace más grande y cambia de color, al

cliquearlo la aplicación pregunta si el usuario desea guardar la línea de ceros, en caso de si se prosigue con el trazado de proyecto.

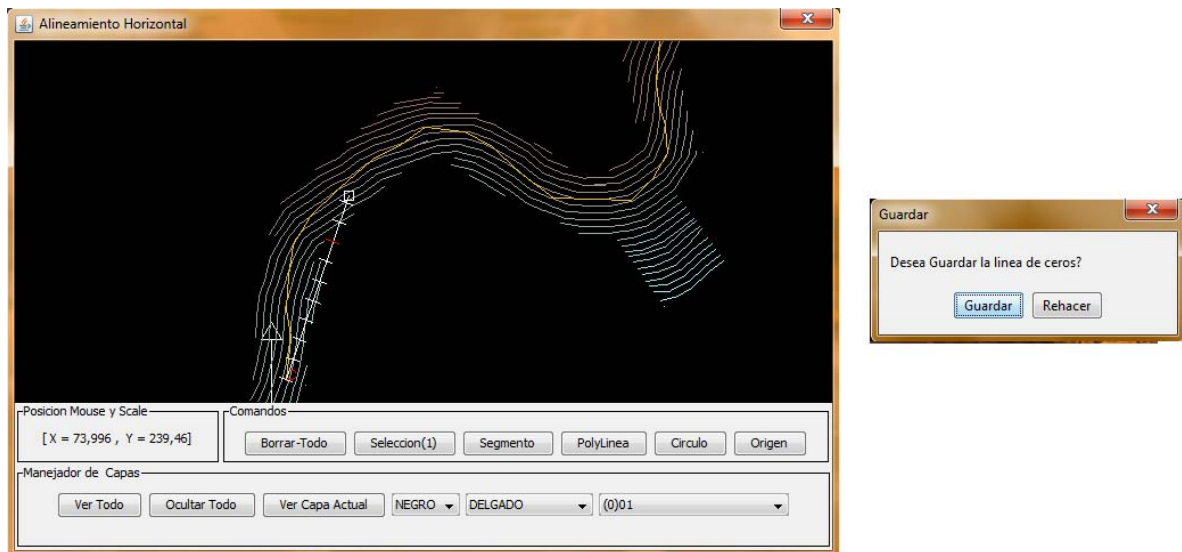
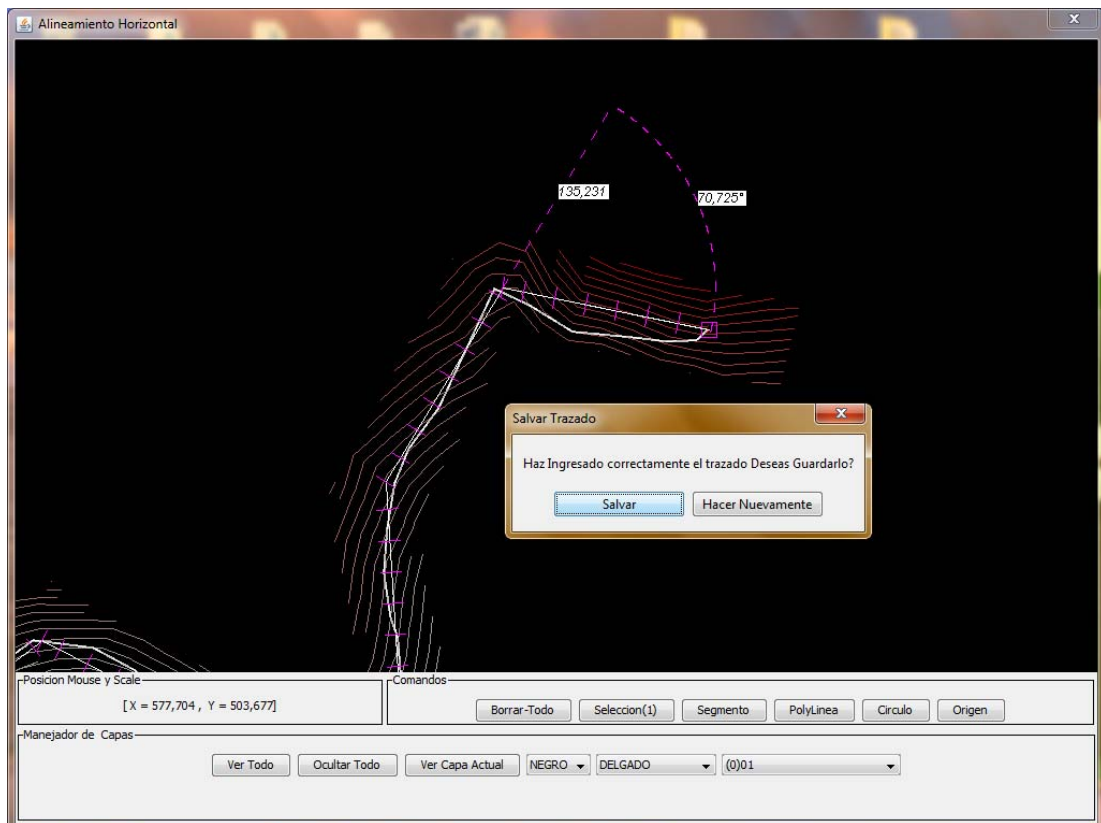


ILUSTRACIÓN 64 TRAZADO DE PROYECTO

El trazado de proyecto permite Dibujar la línea de proyecto sobre las curvas de nivel del terreno, nótese en la Ilustración 59 que el trazado se diferencia de la línea de ceros por que se encuentra subdividida por pequeñas secciones transversales, el ancho de estas secciones es del valor suministrado como zona de afectación y cambian de color cuando se encuentran sobre terreno sin información, la línea central cambia cuando se descende o asciende con la misma pendiente más de lo estipulado en la norma, el trazado muestra el valor de las deflexiones y la longitud de los tramos. Al unirse el trazado con el puto final la aplicación pregunta si debe guardarse, en caso de ser afirmativo se guarda el trazado y se prosigue al siguiente módulo de diseño de curvas horizontales.

ILUSTRACIÓN 65 FIN ALINEAMIENTO



3.1.3.2. Diseño de Curvas Horizontales

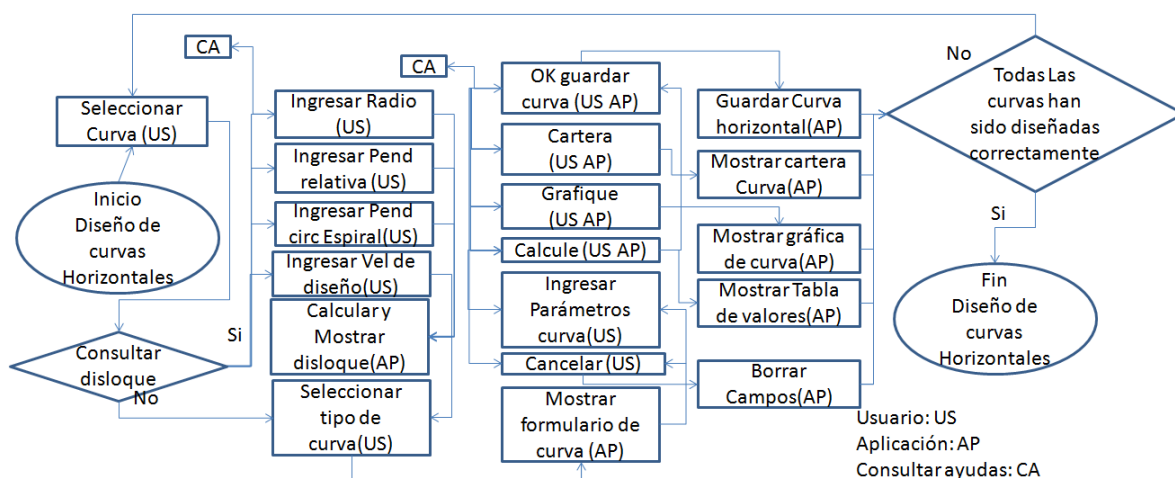


ILUSTRACIÓN 66 DIAGRAMA DISEÑO DE CURVAS HORIZONTALES DIGRAVI

Tomando de partida el trazado realizado en el módulo anterior, éste identifica las deflexiones del trazado y las enumera de 1 a n en el sentido de avance de la vía,

el usuario escoge la curva suministrando algún número, después decide si desea visualizar el disloque a demás de la información que ya suministra la aplicación como lo son la velocidad de diseño, entre tangencia y deflexión, para ver el disloque deben suministrars además el radio y las pendientes relativa y Circular espiral.

3.1.3.2.1. Ejemplo:

El trazado definido en al ejemplo anterior será usado para diseñar una curva horizontal, lo primero después de desplegar el módulo de Diseño de curvas Horizontales es escoger el número de curva por ejemplo 1, la aplicación mostrará la información de esta deflexión.

ILUSTRACIÓN 67 DISEÑO DE CURVAS HORIZONTALES

Usando los valores mostrados la aplicación calcula el disloque, como el valor es bajo es posible hacer un curva simple, en el formulario de curva simple debe ingresar el valor de la cuerda y uno de los parámetros que se encuentran para escoger, se usa para éste ejemplo una cuerda de 10 m y un radio de 98m, con estos datos los cálculos obtenidos son los siguientes.

Elemento	Valor
Curva	1
PC	K0+104,404
PI	K0+147,847
PT	K0+186,155
Delta	47°48'47"
Cuerda	10
Externa	9,198
Grado	5,849
Longitud	81,751
Radio	98,003
Tangente	43,443

ILUSTRACIÓN 68 CURVA CIRCULAR SINPLE

Después de calcular los elementos es posible graficar la curva, ver su cartera y si el usuario se encuentra conforme con el resultado puede accionar el botón OK para guardar la curva en el proyecto.

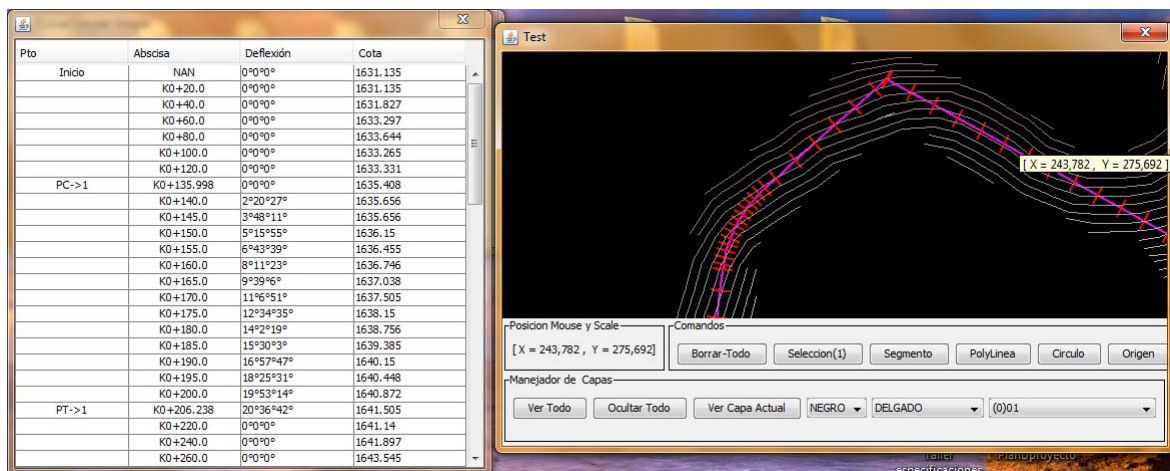


ILUSTRACIÓN 69 GRAFICA Y CARTERA

3.1.4. Transición de peralte



ILUSTRACIÓN 70 TRANSICIÓN DE PERALTE

Para realizar la transición hay que tener en cuenta el tipo de curva y algunos parámetros adicionales como lo son la pendiente relativo y longitudinal, Digravi internamente conoce el tipo de cada una de las curvas que se diseñan por lo cual basta con indicarle el número de curva para que el determine el tipo que le corresponde (ver ILUSTRACIÓN 66 INFORMACIÓN DE CURVA PARA TRANSICIÓN).

Curva Horizontal

Curva No:	1
PENDIENTE RELATIVA (Pr):	0.96
PENDIENTE LONGITUDINAL:	4

ILUSTRACIÓN 71 INFORMACIÓN DE CURVA PARA TRANSICIÓN

Hay tres tipos de transición alrededor del eje, girando el borde interno y girando el borde externo, por cual se tiene que seleccionar el tipo de transición deseada (ver ILUSTRACIÓN 67 TIPO DE TRANSICIÓN).

Método de Transición

- ☒ ALREDEDOR DEL EJE
- ☐ GIRANDO BORDE INTERIOR
- ☐ GIRANDO BORDE EXTERIOR

ILUSTRACIÓN 72 TIPO DE TRANSICIÓN

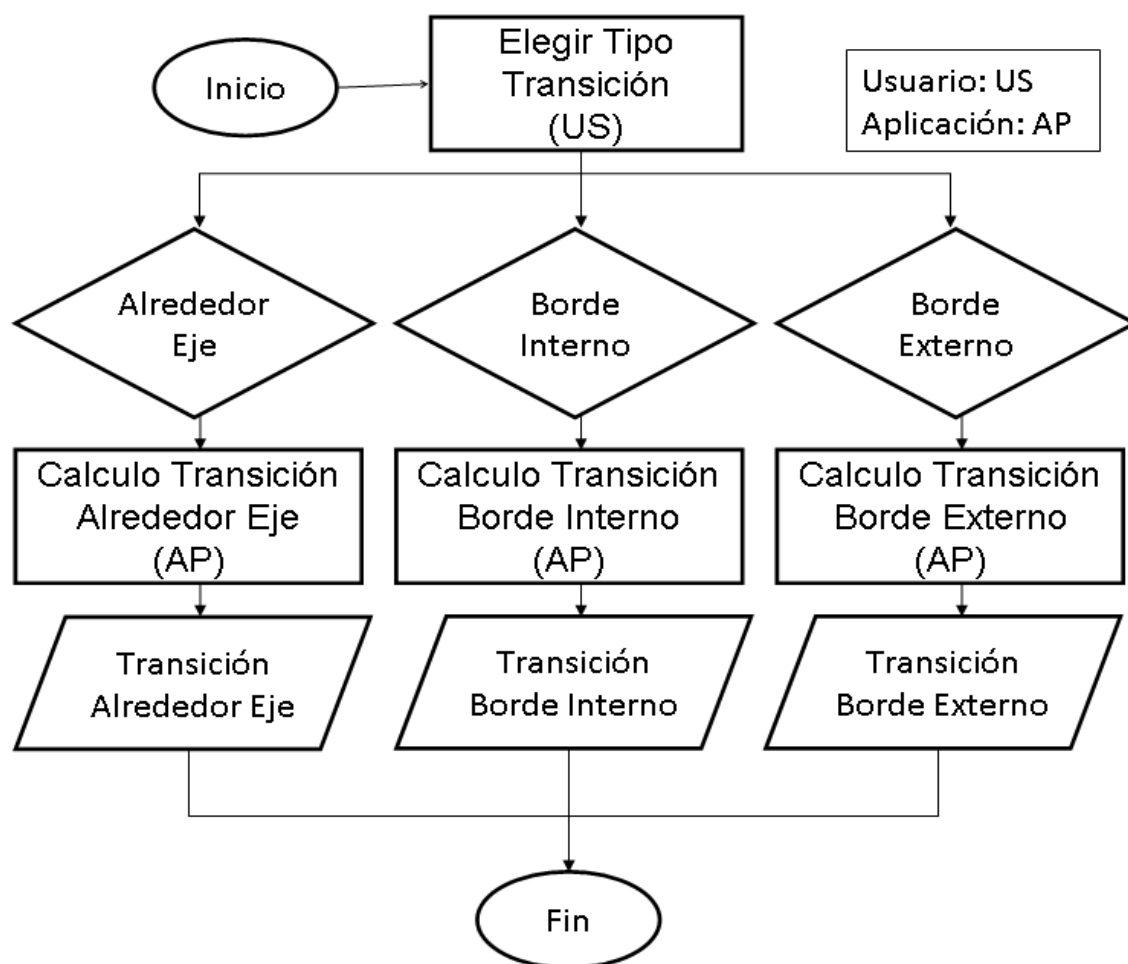


ILUSTRACIÓN 73 DIAGRAMA TRANSICIÓN DE PERALTE DIGRAVI

3.1.4.1 Ejemplo:

Se calculara la transición de peralte para la primera curva con los siguiente valores:

Campo	Valor
Curva No	1
Pendiente Relativa	0.96
Pendiente Longitudinal	4
Alrededor Eje	Seleccionado
Girando Borde Interno	Seleccionado
Girando Borde Externo	Seleccionado

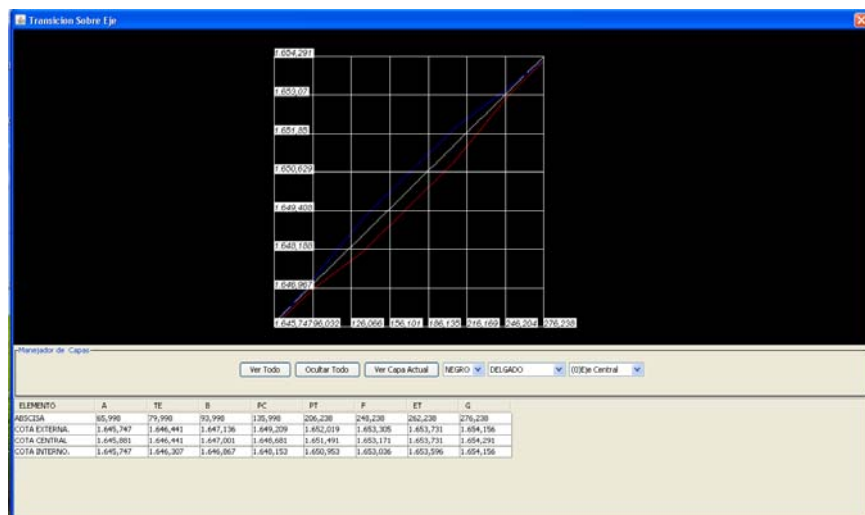


ILUSTRACIÓN 74 TRANSICIÓN ALREDEDOR DEL EJE

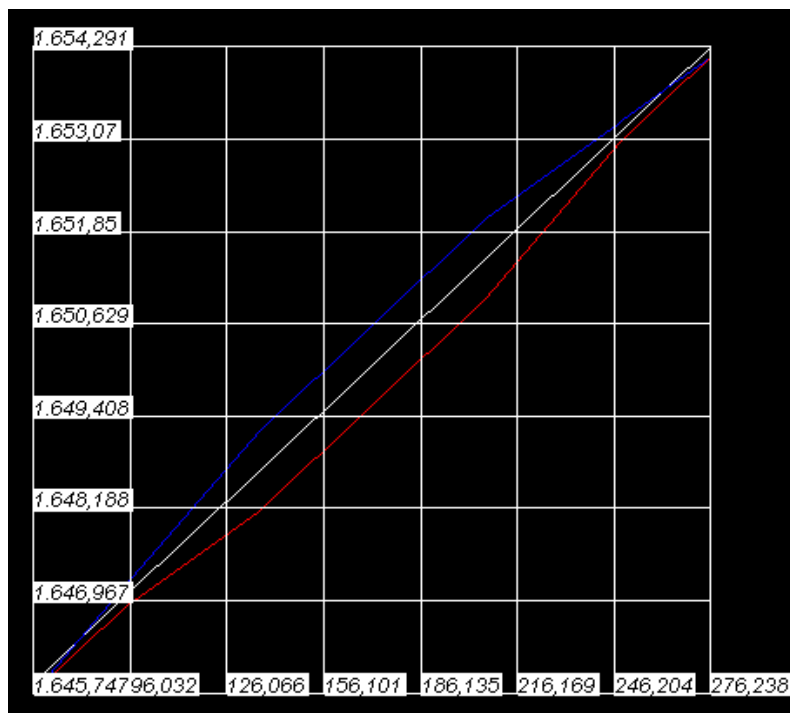


ILUSTRACIÓN 75 OVNI TRANSICIÓN ALREDEDOR DEL EJE

TABLA 6 TRANSICIÓN ALREDEDOR DEL EJE

ELEMENTO	A	TE	B	PC	PT	F	ET	G
ABSCISA	65,998	79,998	93,998	135,998	206,238	248,238	262,238	276,238
COTA EXTERNA.	1.645,747	1.646,441	1.647,136	1.649,209	1.652,019	1.653,305	1.653,731	1.654,156
COTA CENTRAL	1.645,881	1.646,441	1.647,001	1.648,681	1.651,491	1.653,171	1.653,731	1.654,291
COTA INTERNO.	1.645,747	1.646,307	1.646,867	1.648,153	1.650,953	1.653,036	1.653,596	1.654,156

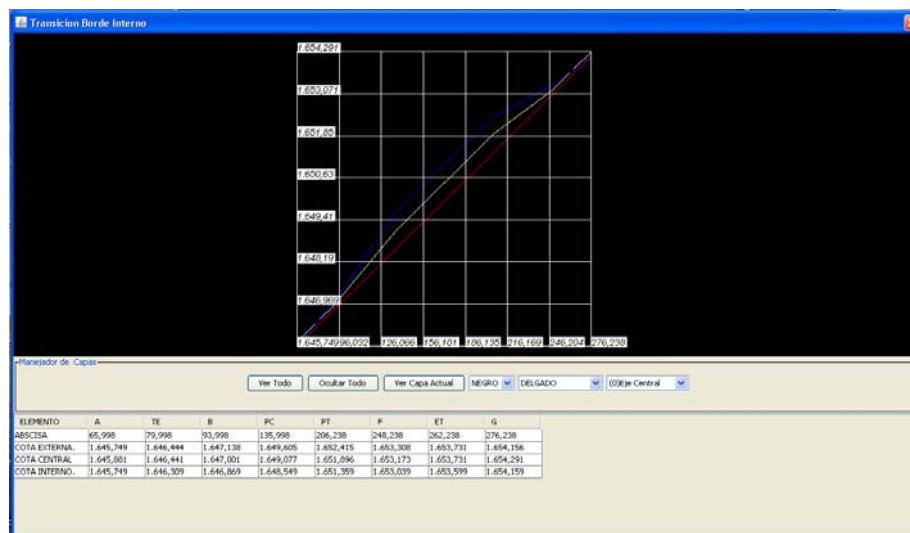


ILUSTRACIÓN 76 TRANSICIÓN DEL BORDE INTERNO

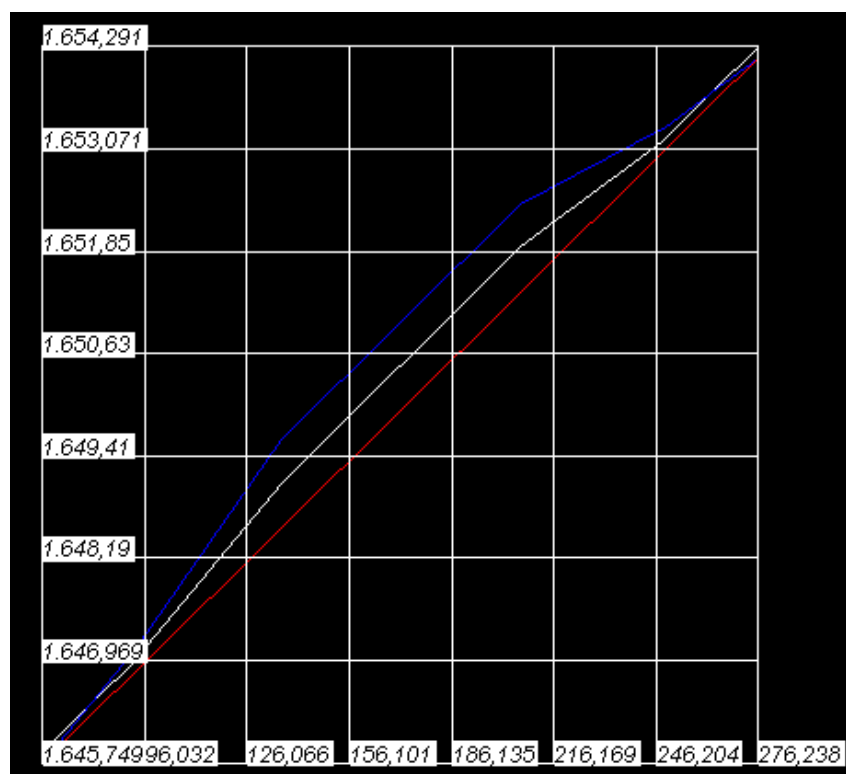


ILUSTRACIÓN 77 OVNI TRANSICIÓN DEL BORDE INTERNO

TABLA 7 TRANSICIÓN DEL BORDE INTERNO

ELEMENTO	A	TE	B	PC	PT	F	ET	G
ABSCISA	65,998	79,998	93,998	135,998	206,238	248,238	262,238	276,238
COTA EXTERNA.	1.645,749	1.646,444	1.647,138	1.649,605	1.652,415	1.653,308	1.653,731	1.654,156
COTA CENTRAL	1.645,881	1.646,441	1.647,001	1.649,077	1.651,896	1.653,173	1.653,731	1.654,291
COTA INTERNO.	1.645,749	1.646,309	1.646,869	1.648,549	1.651,359	1.653,039	1.653,599	1.654,159

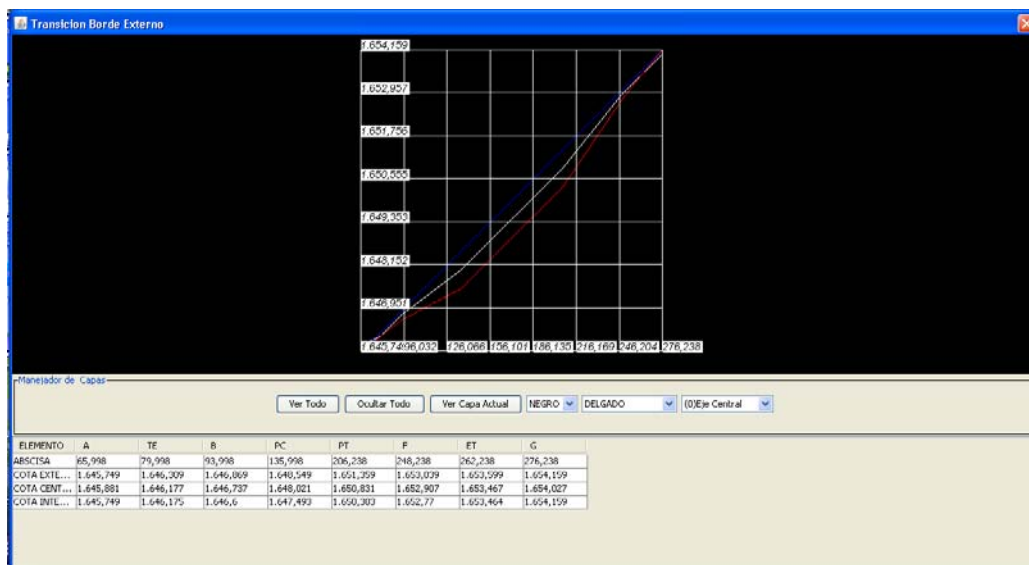


ILUSTRACIÓN 78 TRANSICIÓN DEL BORDE EXTERNO

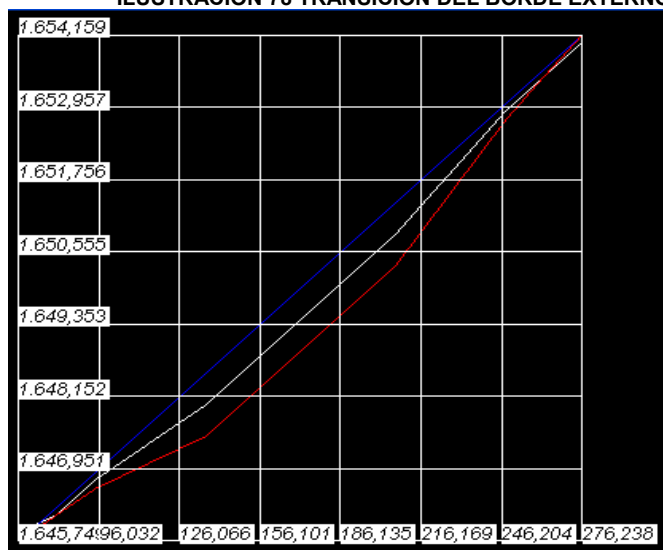


ILUSTRACIÓN 79 OVNI TRANSICIÓN DEL BORDE EXTERNO

TABLA 8 TRANSICIÓN DEL BORDE EXTERNO

ELEMENTO	A	TE	B	PC	PT	F	ET	G
ABSCISA	65,998	79,998	93,998	135,998	206,238	248,238	262,238	276,238
COTA EXTE...	1.645,749	1.646,309	1.646,869	1.648,549	1.651,359	1.653,039	1.653,599	1.654,159
COTA CENT...	1.645,881	1.646,177	1.646,737	1.648,021	1.650,831	1.652,907	1.653,467	1.654,027
COTA INTE...	1.645,749	1.646,175	1.646,6	1.647,493	1.650,303	1.652,77	1.653,464	1.654,159

4 DESARROLLO DE LA APLICACIÓN

4.1. GENERACIÓN DE TERRENO

4.1.1. Análisis: El diseño de una carretera comienza con un terreno en el cual se desea realizar una carretera que una dos lugares específicos, como se muestra en el marco teórico (*ver 2.1 DISEÑO VIAL*), después de tomar en consideración el impacto los pros y contras de hacer la vía se debe pensar en la manera de modelar el terreno, de llevarlo a la aplicación para poder interactuar con él. Ante esta tarea se presentaron para consideración varias opciones:

1. Importar un archivo de Autocad o de alguna Herramienta de Diseño asistido el cual puede ser modificado y elaborado en dicha aplicación (*ver 2.4 DISEÑO ASISTIDO POR COMPUTADORA CAD*), en este caso la aplicación se volvería un poco dependiente de Autocad.

Esta es la forma en que se hace actualmente; el archivo de Autocad se calca de un plano o se consigue de algún conocido con lo cual se privan de la actividad de generar el terreno por ellos mismos. Por otra parte, el calcar un plano de curvas de nivel en Autocad prueba la destreza del alumno en el manejo de la herramienta, pero no le aporta en su aprendizaje, conceptualización y en el conocimiento de un levantamiento topográfico, no se sabría cómo se generó el plano porque el plano ya existe y sólo se está calcando, las ventajas de éste método son sencillez y rapidez.

Por otra parte se presenta el mismo inconveniente que ya se viene presentando y es que la herramienta se vuelve dependiente de otra, lo cual no es provechoso por dos razones: en primer lugar, no hay garantía de que el formato de estas herramientas se siga manteniendo con las nuevas versiones, lo que ocurre actualmente con la herramienta en uso, la cual requiere de una versión de Autocad antigua, desactualizada y difícil de conseguir, y en segunda

instancia, El uso de Autocad incrementaría los costos y las complicaciones para los estudiantes por que tendría que conseguirse e instalarse una herramienta extra que tiene un alto costo.

2. Dar como entrada a la aplicación un plano con las curvas de nivel del terreno para que de este modo la aplicación realice un tratamiento de la imagen identificando en ella las curvas y símbolos del plano para su digitalización, esta opción es crítica en el caso de la obtención de los planos originales ya que éstos tienen un alto costo, y por su tamaño se hacen difíciles de escanear; los mapas deberían conservarse en muy buen estado para garantizar su adecuado reconocimiento.

Esta propuesta tiene a su favor el hecho de que, como la anterior, es fácil, rápida y además no depende de la adquisición de otra herramienta, solo de conseguir la imagen de un plano con las curvas de nivel, pero por el otro lado se tomó en consideración que para que la calidad en el reconocimiento de las curvas fuera lo suficientemente buena, deben las imágenes estar en un estado impecable, sin pliegues, manchones ni otros factores de ruido; de otra forma las curvas obtenidas de esta manera no serán confiables. Para este fin deben comprarse al CODAZZI los planos de los terrenos específicos para escanear y tener un escáner de grandes dimensiones para poder escanear un plano completo, ya que en un escáner normal habría que doblar el plano para escanearlo por partes y como se dijo anteriormente esto desmejoraría la calidad de la imagen. Además La adquisición tanto del escáner como de los planos es bastante costosa para los estudiantes.

Otro factor a tener en cuenta es que, como en el caso anterior es estudiante no tendría que hacer ningún esfuerzo por entender de donde salió el plano solo comprarlo o conseguirlo.

3. Realización de levantamientos topográficos a partir de los cuales los estudiantes deben generar una cartera de campo con las alturas del terreno. Con base en esta cartera se generan las curvas de nivel del terreno.

El levantamiento topográfico se realiza en un terreno real utilizando los implementos y las metodologías enseñadas por la universidad en el área de topografía; estos implementos han sido adquiridos por la universidad para tal fin. A pesar de que el proceso es un apoco más demorado y largo, es realizado por los mismos estudiantes; realizar directamente un levantamiento les permite utilizar los conocimientos adquiridos en la universidad para la elaboración de un plano, esta propuesta no requiere de ninguna otra herramienta sino lápiz y papel y la disposición de los estudiantes.

4. Cargar en la aplicación una imagen del plano como guía, de tal manera que el estudiante con una herramienta de polilínea calque las curvas de nivel del terreno de la imagen en el fondo.

Por las razones expuestas anteriormente, este método de calcado no permite al estudiante utilizar los conocimientos que emplea en la labor de hacer un levantamiento topográfico.

por estas razones se decidió por la realización de un levantamiento que diera como resultado una cartera topográfica, opción que le aporta conocimiento práctico a los estudiantes y disminuye los costos tanto para la universidad como para el estudiante y que al mismo tiempo le permite a la herramienta ser independiente de otros programas.

4.1.2. Requisitos:

La aplicación debe permitir la digitalización de un terreno a partir de una cartera de campo, dicha cartera deberá estar en el formato de distancias fijas o cota redonda.

La aplicación debe estar en capacidad de importar un archivo de Excel en dicho formato y generar las curvas de nivel del terreno a partir de éste.

La aplicación debe tener también un editor propio independiente que permita la creación de esta misma cartera en los formatos antes mencionados, la cartera creada dentro de la aplicación debe ser exportable a un archivo Excel.

La aplicación debe tener un módulo titulado “Generación de terreno” con las siguientes características:

1. Debe contener un botón **importar** que permita escoger, a través de un selector de archivos, el archivo de Excel a importar.
2. Debe contener una caja de texto que permita visualizar la dirección del archivo que se ha escogido.
3. Debe contener una caja de texto donde permita ingresar el valor del **azimut** en notación de radianes o grados, minutos y segundos; también debe validar esta información verificando su correcta sintaxis.
4. Un botón aceptar que al ser presionado calcule las curvas de nivel del terreno a partir de la cartera generada o importada en el programa, y que posteriormente grafique las respectivas curvas.
5. En el transcurso de las anteriores acciones debe manejarse los posibles errores y mostrar en casos de ocurrencia un mensaje informativo.

Los requisitos enumerados anteriormente pueden visualizarse con mayor claridad en el siguiente diagrama de casos de uso (Ver ANEXO 1 CASOS DE USO)

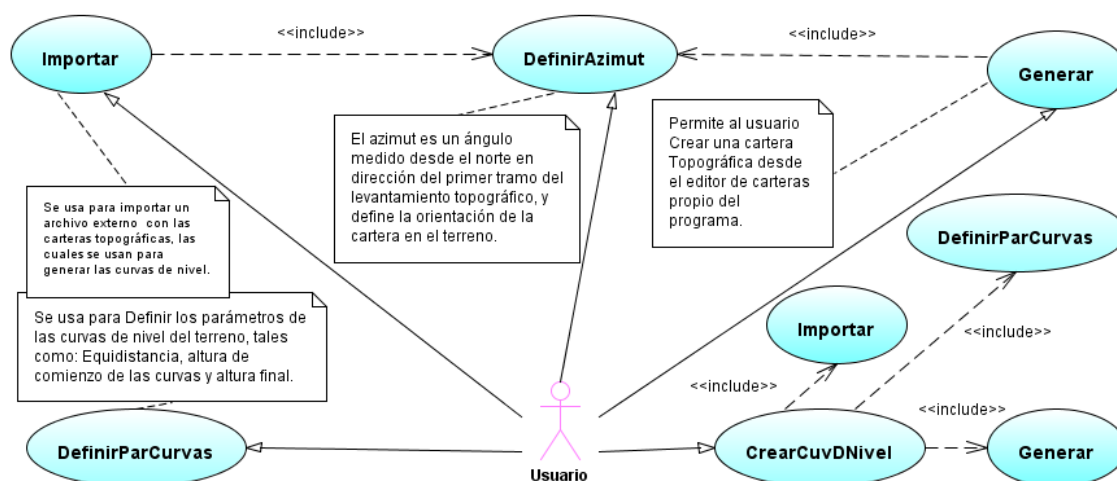


ILUSTRACIÓN 80. CASO DE USO GENERACIÓN DE TERRENO

4.1.3. Diseño:

4.1.3.1. Formato Archivos De Excel

Para el diseño de este módulo, se necesitó primero de la comprensión de los métodos de elaboración de carteras de distancias fijas y cota redonda, la

información de las carteras topográficas comúnmente se recolecta por los estudiantes de topografía en lápiz y papel, una vez realizada la práctica en el terreno, los estudiantes deben pasar en limpio la cartera; generalmente utilizan la herramienta ofimática Excel para esta tarea, y ha demostrado ser muy sencilla y agradable de usar para ellos, razón por la cual hemos decidido brindar compatibilidad de la aplicación con los archivos de Excel y además que el editor interno de la aplicación permita crear la cartera de una manera semejante.

Primero proponemos un formato específico tanto para las carteras de distancias fijas como para las de cota redonda, de tal manera que la aplicación pueda leerlas adecuadamente y pueda identificar el significado de cada uno de los campos.

ABSCISA	DEFLEXIÓN	IZQUIERDA	CENTRAL	DERECHA
30	25	20	15	10
5			5	
				5
				10
				15
				20
				25
				30

ILUSTRACIÓN 81. CARTERA DE DISTANCIA FIJAS

Filas 1 y 2: conforman el encabezado de la cartera, que irónicamente debe ir en las últimas dos filas. Esto se acostumbra por convención dado que se asemeja al recorrido que debe trazar el estudiante que toma los datos a través del terreno, y el abscisado se va llenando en sentido ascendente en la medida en que se toman

los datos desde el punto de comienzo hasta el punto final, como muestra la flecha en la imagen.

Abscisa: La casilla 1 de la fila 2 del encabezado está marcada con la palabra “ABSCISA” y se encuentra combinada con la casilla 1 de la fila 1. Todas las casillas que se encuentran sobre esta columna serán abscisas del terreno, y serán marcadas con la notación de abscisa “Km+n” donde m y n son números reales, o “n” donde n es un numero real. Por ejemplo:

Km+n	Real[m]	DESCRIPCION
K12+120	12120	Escritura ideal sin espacios ni caracteres de ruido
K 12 + 120	12120	Escritura con espacios
K1 2+1 b 20	12120	Escritura con espacios y con ruido
k0+120	120	Escritura en minúscula
Ko+120	120	Escritura con caracteres truncados 0→o
Kmt 0+1 20.34m	120.34	Escritura con caracteres truncados 0→0, con caracteres de ruido con espacios con caracteres en minúscula y números reales.
12,0 = 12.0	12	escritura con números reales coma decimal

TABLA 9 POSIBLES VALORES QUE PUEDE TOMAR EL CAMPO ABSCISA

Cualquier otro tipo de entrada dar como respuesta “SYNTAX ERROR” dado que está mal formulada.

Deflexión: La casilla 2 de la fila 2 del encabezado está marcada con la palabra “DEFLEXIÓN” y se encuentra combinada con la casilla 2 de la fila 1. Todas las casillas que se encuentran sobre esta columna serán deflexiones del terreno y serán marcadas con la notación de deflexión “G°M°S°” donde G, M y S son enteros, o “n” donde n es un numero real en radianes, Por ejemplo:

ANGULO		DESCRIPCION
GRADOS	RADIANES	

$45^{\circ}45'45'' = 45.7625$	0.7987	Escritura ideal sin espacios ni caracteres de ruido
$4\ 5^{\circ}4\ 5^{\circ}45' = 45.7625$	0.7987	Escritura con espacios
$r4\ t\ 5^{\circ}45' = 4\ 5u^{\circ}45' = 45.7625$	0.7987	Escritura con espacios y con ruido
$r4\ t\ 5'45 = 4\ 5u'45$	0.7987	escritura con caracteres truncados °→'

TABLA 10 POSIBLES VALORES QUE PUEDE TOMAR EL CAMPO DEFELXION

Cualquier otro tipo de entrada dar como respuesta “SYNTAX ERROR” dado que está mal formulada.

Izquierda: Sobre la sección 3 en la fila 2 se encuentran los valores reales que enuncian las distancias fijas en las cuales se tomarán alturas en cada sección transversal Izquierda (*ver 2.2.1.1 Método de distancias fijas*). En esta misma sección, la fila 1 que se encuentra debajo de la anterior estará marcada con la palabra “IZQUIERDA” y se encontrará combinada con las demás celdas que se encuentren debajo de las distancias fijas de la izquierda. La notación que reciben las distancias fijas es la de un real positivo que ya ha sido explicada anteriormente.

En todas las filas sobre la sección 3 deberán escribirse los valores de las alturas que se encuentran en las distancias fijas del valor real presente en la base de la columna que se esté llenando.

Derecha: Sobre la sección 5 en la fila 2 se encuentran los valores reales que enuncian las distancias fijas en las cuales se tomarán alturas en cada sección transversal Derecha. En esta misma sección la fila 1 que se encuentra debajo de la anterior estará marcada con la palabra “DERECHA”, y se encontrará combinada con las demás celdas que se encuentren debajo de las distancias fijas de la

derecha. La notación que reciben las distancias fijas es la de un real positivo que ya ha sido explicada anteriormente.

En todas las filas sobre la sección 3 deberán escribirse los valores reales de las alturas que se encuentran en las distancias fijas del valor real presente en la base de la columna que se esté llenando.

Central: En la Fila 1 de la sección 4 se encuentra la palabra “CENTRAL”, la cual demarcara la posición del eje de referencia central. En la fila 2 sobre esta misma sección se encontrará la palabra “EJE” la cual servirá como guía para diferenciar los valores fijos de las secciones izquierda y derecha. Los demás valores que se encuentran en esta columna son las alturas que se encuentran a lo largo del eje sobre el valor de la abscisa de esa fila.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
102	KO+240		30	25	20	15	10	5	EJE	5	10	15	20	25	30		
103			1649.1	1647.62	1646.15	1644.68	1643.2	1641.73	1640.26	1638.79	1637.32	1635.86	1634.38	1632.91	1631.44		
104	KO+220		30	25	20	15	10	5	EJE	5	10	15	20	25	30		
105			1649.03	1647.57	1646.06	1644.54	1643.03	1641.52	1640	1638.48	1637	1635.52	1634.05	1632.57	1631.1		
106	KO+212.72	410459°	30	25	20	15	10	5	EJE	5	10	15	20	25	30		
107			1646.79	1645.22	1643.65	1642.1	1640.57	1639.13	1637.73	1636.41	1635.25	1634.16	1633.19	1632.23	1631.27		
108	KO+200		30	25	20	15	10	5	EJE	5	10	15	20	25	30		
109			1644.06	1642.6	1641.19	1639.8	1638.58	1637.02	1635.64	1634.25	1632.72	1631.11	1629.48	1627.94	1626.81		
110	KO+180		30	25	20	15	10	5	EJE	5	10	15	20	25	30		
111			1643.28	1641.83	1640.36	1638.88	1637.4	1635.92	1634.44	1632.96	1631.48	1630	1628.52	1627.05	1625.13		
112	KO+160		30	25	20	15	10	5	EJE	5	10	15	20	25	30		
113			1642.16	1640.66	1639.16	1637.66	1636.18	1634.74	1633.33	1631.89	1630.43	1628.98	1627.54	1626.1	1624.08		
114	KO+140		30	25	20	15	10	5	EJE	5	10	15	20	25	30		
115			1642.19	1640.7	1639.21	1637.72	1636.24	1634.75	1633.26	1631.77	1630.28	1628.79	1627.31	1625.87	1624.1		
116	KO+120		30	25	20	15	10	5	EJE	5	10	15	20	25	30		
117			1642.58	1641.09	1639.6	1638.12	1636.63	1635.14	1633.65	1632.16	1630.67	1629.18	1627.69	1626.21	1624.1		
118	KO+100		30	25	20	15	10	5	EJE	5	10	15	20	25	30		
119			1641.77	1640.32	1638.85	1637.35	1635.84	1634.7	1633.42	1631.78	1630.32	1628.86	1627.4	1625.92	1624.1		
120	KO+80		30	25	20	15	10	5	EJE	5	10	15	20	25	30		
121			1641.23	1639.77	1638.31	1636.86	1635.4	1633.94	1632.48	1631	1629.52	1628.03	1626.54	1624.11	1622.1		
122	KO+60		30	25	20	15	10	5	0	5	10	15	20	25	30		
123			1639.02	1637.77	1636.51	1635.25	1634	1632.64	1631.25	1629.83	1628.41	1626.99	1625.59	1624.22	1622.1		
124	KO+40		30	25	20	15	10	5	0	5	10	15	20	25	30		
125			1635.42	1634.99	1633.25	1632.04	1630.78	1629.53	1628.17	1626.78	1625.32	1623.86	1622.4	1620.92	1619.1		
126	KO+20		30	25	20	15	10	5	0	5	10	15	20	25	30		
127			1630.77	1630.02	1629.06	1628.04	1627.05	1626.05	1624.94	1623.78	1622.52	1621.18	1619.84	1618.4	1616.92		
128	KO+00		30	25	20	15	10	5	EJE	5	10	15	20	25	30		
129	ABSCISA	DEFLEXIÓN	IZQUIERDA					CENTRAL		DERECHA							

ILUSTRACIÓN 82. CARTERA DE COTA REDONDA

Fila 1: conforma el encabezado de la cartera, el cual se encuentra de la misma manera que la cartera de distancias fijas en la última fila, y solo abarca esta fila, a diferencia de la cartera anterior cuyo encabezado tenía dos filas.

Abscisa: La casilla 1 de la fila 1 en el encabezado está marcada con la palabra “ABSCISA”, todas las casillas que se encuentran sobre esta columna serán abscisas del terreno y serán marcadas de la misma manera que se ha explicado con la cartera de distancias fijas, con la Diferencia de que debe ir dejándose vacía una fila de por medio como muestra la columna 1 (*ver ILUSTRACIÓN 77. CARTERA DE COTA REDONDA*). Esto en razón a que cada sección transversal agregada tendrá su propia fila de distancias a diferencia de la cartera anterior cuyas distancias eran fijas y se ponían en el encabezado una sola vez para toda la cartera.

Deflexión: La casilla 2 de la fila 1 del encabezado está marcada con la palabra “DEFLEXIÓN”, indicando las deflexiones del terreno, y serán marcadas como explicamos en la cartera anterior. De la misma manera que la columna de Abscisas de esta cartera, las deflexiones deben intercalarse dejando una casilla en blanco en la parte de arriba (*ver ILUSTRACIÓN 77. CARTERA DE COTA REDONDA* filas 105 y 106 columna 2).

Izquierda: Sobre la sección 3 en la fila 1 se encuentra la palabra “IZQUIERDA”, se encontrará combinada con las demás celdas que se encuentren debajo de los valores de cota redonda de la izquierda. Los valores de una sección transversal abarcan 2 filas: la primera tendrá el valor de la distancia a la cota redonda, por ejemplo supongamos que a lo largo de la sección transversal el terreno tiene una altura de 1600 que es la cota redonda, y la distancia a ese punto es de 3.45m entonces en la fila de abajo se pone 3.45 y en la de arriba 1600; de la misma manera se ponen los valores para todas las cotas cerradas que se encuentren a lo largo de esa sección transversal.

Derecha: Sobre la sección 5 en la fila 1 se marca con la palabra “DERECHA”, se encontrará combinada con las demás celdas que se encuentren debajo de las

secciones transversales derechas. Deberán llenarse de la misma manera que las secciones izquierdas con su respectivo cambio de dirección.

Central: En la Fila 1 de la sección 4 se encuentra la palabra “CENTRAL” la cual demarcará la posición del eje de referencia central, en la fila 2 sobre esta misma sección se encontrará la palabra “EJE”, la cual servirá como guía para diferenciar los valores de las secciones izquierda y derecha. Los demás valores que se encuentran en esta columna son las alturas que le encuentran a lo largo del eje sobre el valor de la abscisa de esa fila. Esta columna, al igual que las de abscisa y deflexión, también deben dejar un espacio intermitente vacío o con la palabra “EJE” o 0 (ver ILUSTRACIÓN 77. CARTERA DE COTA REDONDA)

4.1.3.2. Carteras:

Dentro de la aplicación debe haber un objeto que reciba este tipo de información y la guarde ordenadamente para el uso por parte de la aplicación, además que tenga métodos de acceso rápidos a esta información.

Primero definimos un punto que en la cartera defina una posición específica, el cual está conformado por una abscisa, una cota y una deflexión.

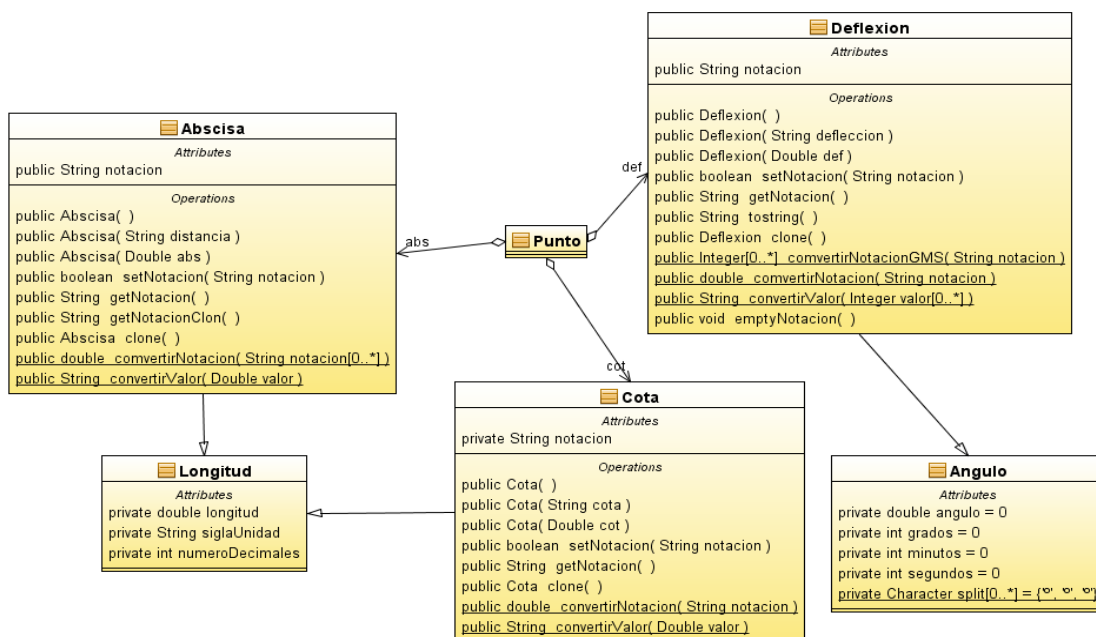


ILUSTRACIÓN 83. DIAGRAMA DE CLASE PUNTO

Un Conjunto Ordenado de puntos forma una cartera. Esta colección de puntos permite ordenar y buscar información de ellos, además de interpolar valores entre los mismos a partir de la información dada.

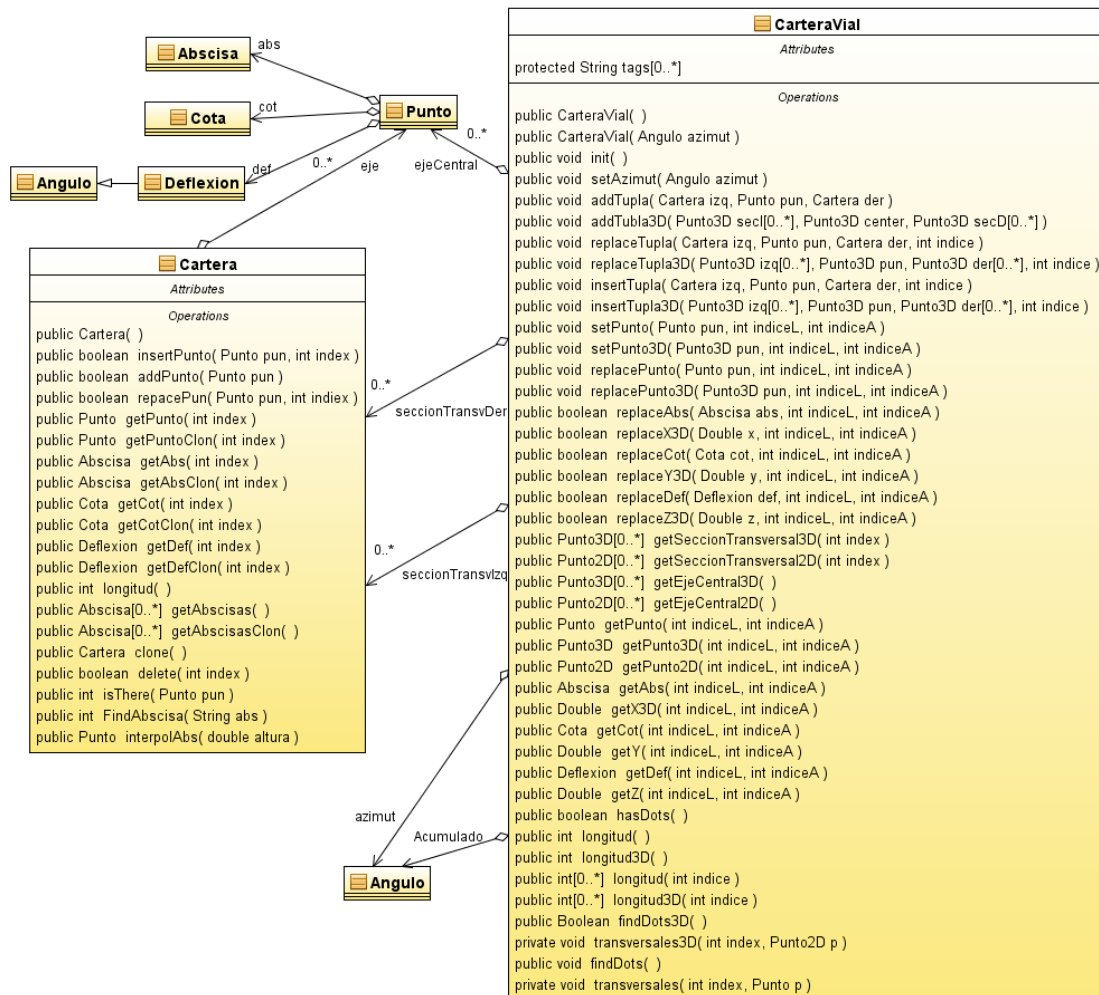


ILUSTRACIÓN 84. DIAGRAMA DE CLASE CARTERA VIAL

Una CarteraVial utiliza estas carteras como secciones transversales y para definir su orientación requiere de un ángulo inicial llamado *azimut* (ver *ILUSTRACIÓN 47 PLANO DE EJEMPLO*). La cartera vial es una estructura flexible que puede recibir la información de distintos tipos de carteras, tiene métodos especiales que le permiten transformar los puntos viales en puntos de coordenadas rectangulares, así como métodos para dar su información en distintas presentaciones como

puntos viales y puntos de coordenadas rectangulares a lo largo y ancho de la cartera.

A partir de la cartera podemos extender el eje de referencia y el trazado de la vía, estas son carteras con un gran parecido pero su significado es diferente y cada una tiene un tratamiento especial de los datos (2.2 GENERACIÓN DE TERRENO)

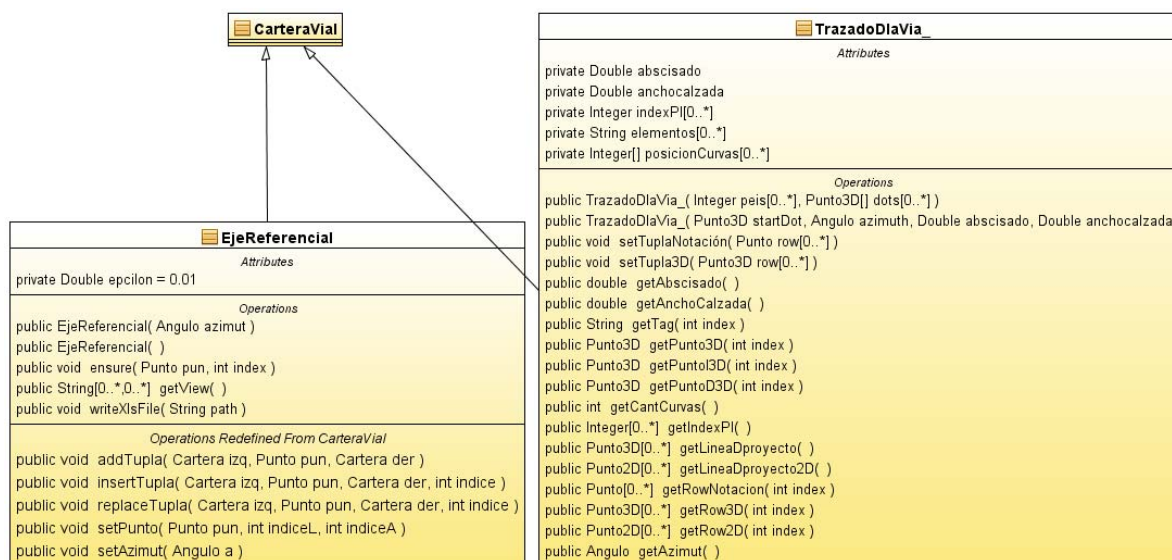


ILUSTRACIÓN 85. DIAGRAMA DE CLASE TRAZADODLAVIA Y EJEREFERENCIAL

Los editores específicos de cota redonda y distancias fijas se diferencian en la forma en que agregan y borran columnas y filas, dado que la colocación de los valores para cada formato es distinta para cada uno.

4.1.3.3. Lectura de Archivos:

Teniendo ya el contenedor de esta información, podemos encargarnos del ingreso de los datos a la aplicación, para lo cual se elabora una clase que maneje los archivos de Excel y permita conectarse con el archivo externo, y también otra clase que, a través de la clase anterior, lea los archivos, identifique su contenido, y de acuerdo a ello ingrese la información del eje de referencia.

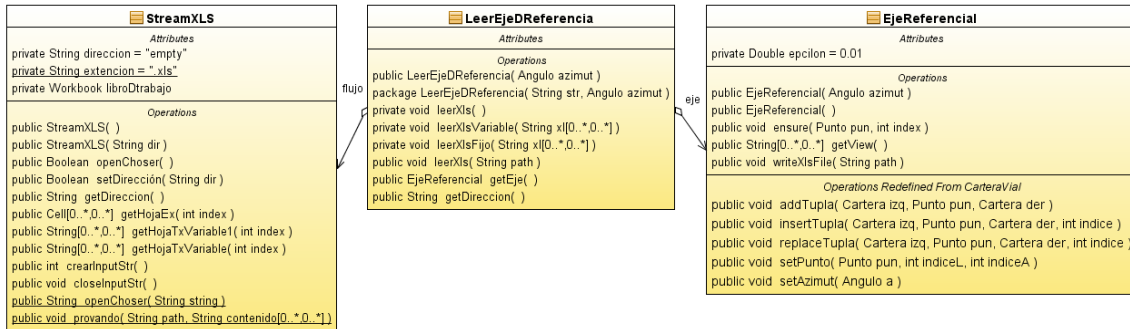


ILUSTRACIÓN 86. DIAGRAMA DE CLASE LEEREJEDREFERENCIA

4.1.3.4. Editores Internos:

Un editor a groso modo debe mostrarse al usuario utilizando el formato semejante al establecido anteriormente para las carteras de distancias fijas y cotas redondas, también debe poder redimensionarse en la medida en que se agregan mas filas o mas columnas, y ser consciente de cuántas secciones tiene y el ancho máximo de las secciones a derecha y a izquierda.

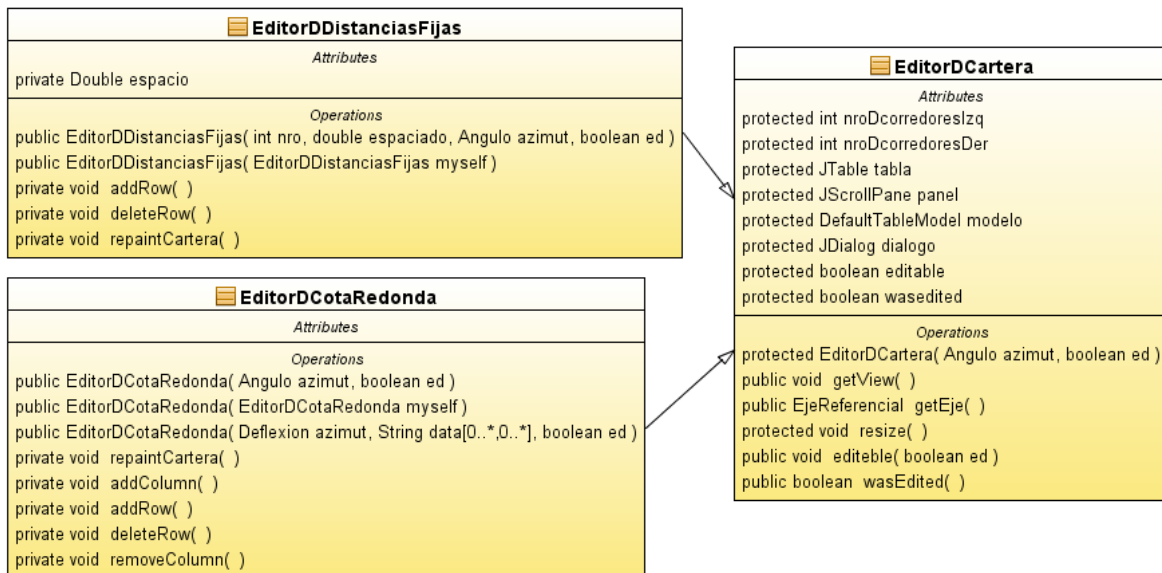


ILUSTRACIÓN 87. DIAGRAMA DE CLASE EDITORES DE CARTERA

Los editores específicos de cota redonda y distancias fijas se diferencian en la forma en que agregan y borran columnas y filas, dado que la colocación de los valores para cada formato de cartera varía de uno a otro.

4.1.3.5. Interfaz Gráfica:

La *Ilustración 46* muestra el boceto propuesto para el módulo de generación de terreno. En la parte superior presenta las dos posibles opciones de ingreso de la cartera: importar un archivo de Excel o generar la cartera con el editor interno del programa. En la sección de abajo se encuentran las casillas donde se puede ingresar el valor del azimut y modificar los valores de generación de las curvas de nivel, tales como equidistancia, cotas de comienzo y finalización de las curvas de nivel. El botón aceptar debe desplegar el gráfico de las curvas de nivel del terreno, el botón Cancelar debe borrar cualquier información que se halla ingresado a este módulo, y el botón de ayudas debe desplegar la ayuda tanto de uso del módulo como de conceptos viales necesarios relacionados con la generación de un terreno a partir de carteras topográficas.

Cartera De Campo

Importar...

Puntos laterales ☐ Esp ☐ Generar

Modificar

Interpolar

Equidistancia Azimut

Altura De Comienzo Altura final:

Aceptar ? Ayuda X Cancelar

ILUSTRACIÓN 88. BOCETO UI

4.1.4. Implementación:

4.1.4.1. Carteras:

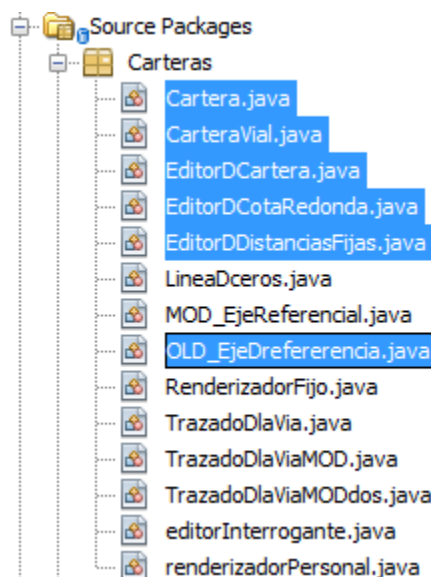


ILUSTRACIÓN 89. PAQUETE CARTERAS

En este paquete se agrupan las clases que se usan para representar en la aplicación las carteras usadas para generar el terreno, tales como: Cartera, CarteraVial, OLD_EjeDReferencia, así como los editores de la aplicación para recibir las carteras antes mencionadas: EditorDCartera, EditorDCotaRedonda, EditorDDistanciasFijas.

4.1.4.2. ClaseBase:

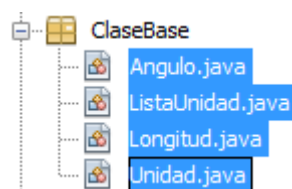


ILUSTRACIÓN 90. PAQUETE CLASEBASE

Contiene los componentes básicos de las magnitudes que conforman una cartera. Una Deflexión, por ejemplo, hereda su funcionalidad y propiedades de un ángulo, las abscisas y cotas son longitudes que están representadas con una unidad de medida.

4.1.4.3. IO:

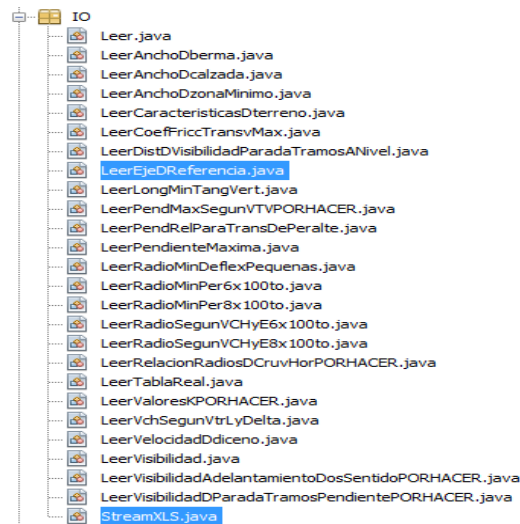


ILUSTRACIÓN 91 PAQUETE IO

De este paquete se utiliza la clase StreamXLS para establecer los flujos de entrada y salida entre los archivos de Excel y la aplicación. LeerEjeDReferencia utiliza la información extraída por StreamXLS para crear el eje de referencia.

4.1.4.4. Interface:

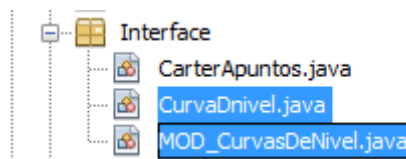


ILUSTRACIÓN 92. PAQUETE INTERFACE

Contiene las clases que implementa la funcionalidad y atributos de: CurvaDNivel del terreno, y MOD_CurvasDNivel. Agrupa y maneja el conjunto de curvas de nivel de de un plano, permite crear estas curvas a partir de un eje de referencia.

4.1.4.5. Ayudas:

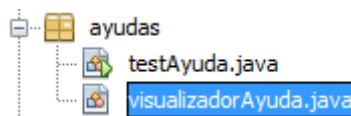


ILUSTRACIÓN 93. PAQUETE AYUDAS

La clase VisualizadorAyuda permite mostrar la ayuda concerniente a este módulo tanto en cuestión de manejo de la aplicación como de conceptos viales.

4.1.4.6. ElementoVial:

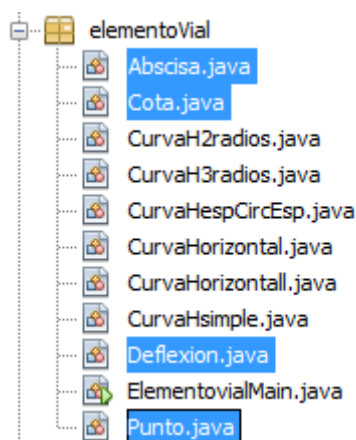


ILUSTRACIÓN 94. PAQUETE ELEMENTOVIAL

En este paquete se agrupan los elementos componentes de una cartera vial la cual se conforma de puntos viales, quienes a su vez se componen de Abscisa, Cota y Deflexión.

4.1.4.7. ExpRegulares:

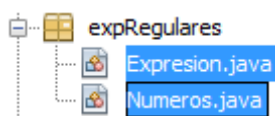


ILUSTRACIÓN 95. PAQUETE EXPREGULARES

Las clases Expresión y Números, son utilizadas por el módulo de generación para validar la información encontrada en las carteras y algunos otros tratamientos sobre sertas de datos.

4.1.4.8. ProyectoVial:

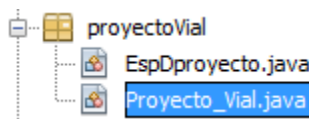


ILUSTRACIÓN 96. PAQUETE PROYECTOVIAL

En este paquete encontramos la clase `Proyecto_Vial`, la cual se encarga de manejar todos los objetos concernientes a un proyecto vial completo, en este caso

el eje de referencia, administrando su actualización y permitiendo a otros módulos de la aplicación acceder a información que ha sido construida en módulos anteriores.

4.1.4.9. Visualización:

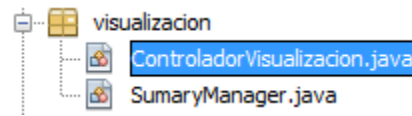


ILUSTRACIÓN 97. PAQUETE VISUALIZACIÓN

Este paquete es la interfaz que permite usar a alto nivel las funcionalidades del motor gráficador. ControladorVisualizacion maneja las herramientas del gráficador para mostrar las curvas de nivel del terreno.

4.1.4.10. ÁrbolDProyecto:

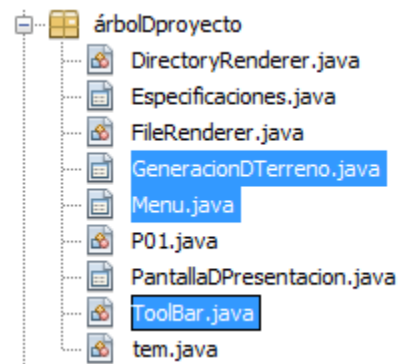


ILUSTRACIÓN 98. PAQUETE ÁRBOLDPROYECTO

El árbol de proyecto contiene la interfaz grafica que interactúa con el usuario. Es la parte visible del programa y que acciona el trabajo de las clases antes mencionadas.

4.1.4.11. ComponentesGraficos:

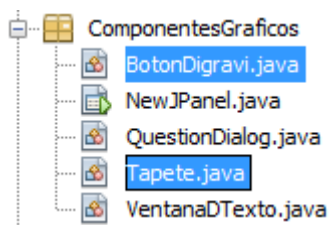


ILUSTRACIÓN 99. PAQUETE COMPONENTESGRAFICOS

Los componentes gráficos son extensiones de `JComponents` normales, estos han sido modificados para armonizar con la propuesta de diseño de la aplicación.

4.2. ESPECIFICACIONES DE DISEÑO

4.2.1. Análisis:

Las especificaciones de diseño toman partido fuertemente en el campo educacional. Dado que la definición de estos valores debe ser criterio del diseñador, debe ser apoyado por las normativas y los conceptos aprendidos en clase.

Las normativas se extraen del GEOM. También se encuentran algunos valores que son definidos por fórmulas, y otros a criterio del diseñador, pero que se encuentran sustentados en una teoría que ha sido impartida en clase.

El objetivo de este modulo es permitir al usuario definir las especificaciones de diseño viales que utilizará a lo largo de los siguientes módulos. Estas serán usadas en cálculos posteriores de elementos viales, pero a la vez que el estudiante hace esta tarea, debe ser consciente de los valores que propone; para ello la aplicación debe suministrar la información necesaria. Para tal fin se consideraron las siguientes opciones:

1. Realizar un manejo en el cual se valida en cada casilla con relación a las tablas de valores, y no permite colocar otros valores fuera de los rangos. Haciendo uso de valores de tablas internas de la aplicación.

El trabajo automático es bastante diligente, por tal razón es muy popular entre los estudiantes, pero por razones discutidas con el docente se ha decidido que no es conveniente dado que los estudiantes no se preocupan por pensar en los valores y el significado real que estos tienen, sólo en el manejo de la herramienta y los valores que esta da. Por tal razón realizan los trabajos de una manera mecánica, sin permitirse probar por fuera de las normativas para poder ver las implicaciones que esto tiene sobre los resultados finales, de tal manera que el nivel de interacción, pruebas y error que alcanza el estudiante es muy corto debido que la herramienta está proponiendo en todo momento y no le permite pensar. Otro inconveniente que presenta esta propuesta es que las tablas están estáticas dentro del programa y no es posible modificar sus valores. Con un nuevo cambio de las normativas GEOM esta herramienta quedaría desactualizada.

2. Realizar este mismo manejo pero con las nuevas normativas de GEOM y que se lean las tablas de archivos externos de tal manera que no queden fijas dentro del programa, y puedan ser modificables en futuro, lo cual brindaría una gran flexibilidad.

Con esta opción es posible actualizar la aplicación y hacer cambios en sus valores, pero seguiría sesgándose al estudiante en sus criterios de diseño, sin permitirle ir más allá.

3. Desplegar en cada casilla de decisión la tabla correspondiente en la cual se seleccione el valor correspondiente.

La presente opción es un poco más libre dado que el estudiante escoge de la misma tabla, pero también se encuentran inconvenientes. En el caso de que por cuestiones de la geografía deban ponerse valores que no se encuentran en la tabla o que fueran valores intermedios, seguiría siendo una opción muy rígida.

4. Buscar automáticamente el valor a partir de las tablas y valores anteriores, lo cual definitivamente castraría la labor de aprendizaje del estudiante, no es conveniente.
5. Presentar las tablas acompañadas de la teoría, cargada desde un archivo externo, concerniente a cada valor a ingresar, de tal manera que el estudiante consulte en cada caso y decida bajo su criterio.

Para este caso es muy sencillo mostrar las tablas en modo visualización, en conjunto con la teoría, y organizar estos documentos para cargarse en las ayudas de la aplicación; de esta manera, si el estudiante coloca cualquier valor y sus diseños presentan anormalidades, él va querer ver las ayudas para solucionar su problema.

En razón a las anteriores consideraciones, se decidió por la opción número 5, sencilla pero que le permite una gran libertad al usuario y es fácil de actualizar.

4.2.2. Requisitos:

La Aplicación debe tener disponible la información teórica, formulas y tablas concernientes a las etapas de diseño geométrico comprendidas en la aplicación, deben poderse acceder desde un lugar visible en cada etapa de diseño vial y ser pertinente y completa para este caso.

La aplicación debe permitir el ingreso de cada valor de especificación en un formulario. Frente a cada valor debe verse un botón de ayuda que remita a la información que facilite la decisión de este valor.

La aplicación debe contener un botón aceptar al ser presionado debe validar todos los campos y marcar los errores en cada uno, en caso de estar correctos todos debe guardar los valores de especificación en Proyecto_Vial.

Debe existir un botón cancelar que borre los valores del formulario y los ingresados en el proyecto

En el módulo de especificación deben poderse definir:

1. Nombre de proyecto
2. tipo de vía
3. Velocidad de diseño
4. Ancho de Banca
5. Pendiente Máxima Horizontal
6. Pendiente Mínima Horizontal
7. Mostrar el tipo de terreno que haya sido cargado
8. Peralte máximo
9. Bombeo
10. Longitud circular mínima
11. Longitud espiral mínima
12. Longitud vertical mínima
13. Abscisado
14. Sobreancho
15. Punto de referencia

Los requisitos antes mencionados se pueden ver mejor en el siguiente diagrama de casos de uso:

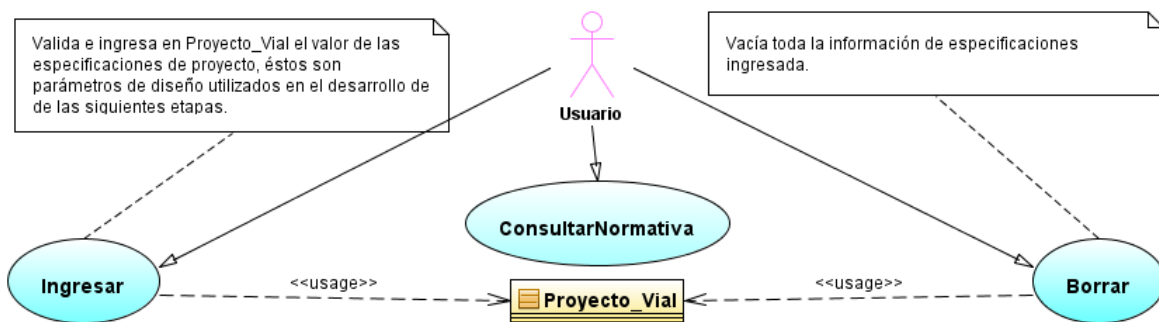


ILUSTRACIÓN 100. CASO DE USO ESPECIFICACIONES DE DISEÑO

4.2.3. Diseño:

4.2.3.1. Especificaciones de Diseño:

Para este módulo primero debemos diseñar una clase en la que se definan las constantes a tener en cuenta como especificación, las cuales deberán ser llamadas de la misma manera desde los demás módulos. Así mismo, deben darse los espacios para ingresar cada tipo de dato. Las especificaciones más elaboradas como terreno y tipo de carretera constituyen nuevos objetos que se agregan a esta misma clase, y la especificación de debe hacer parte del proyecto vial en el cual se va a aplicar. El proyecto vial también tendrá muchos más atributos como se verá más adelante.

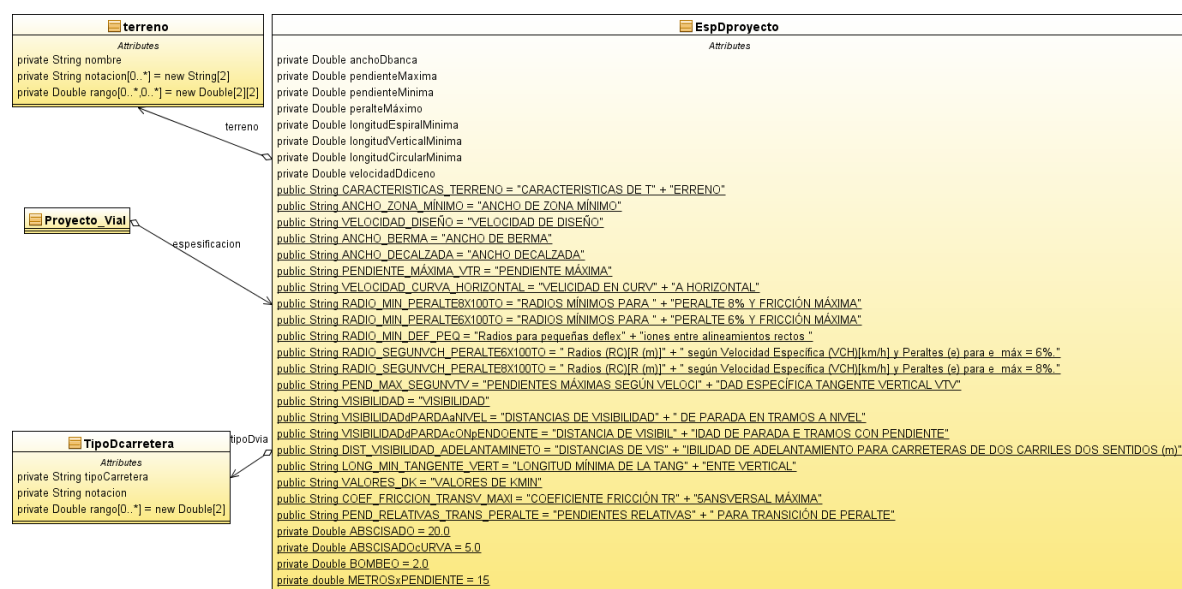


Ilustración 101. Diagrama de Clase de EspDproyecto

4.2.3.2. Manejo de las Ayudas:

Las Ayudas que se utilizan tanto en especificaciones como en los demás módulos usan la clase VisualizadorAyuda el cual maneja los eventos de los botones de ayuda (ver sección 4.4 Ayudas)

4.2.3.3. Interfaz Gráfica:

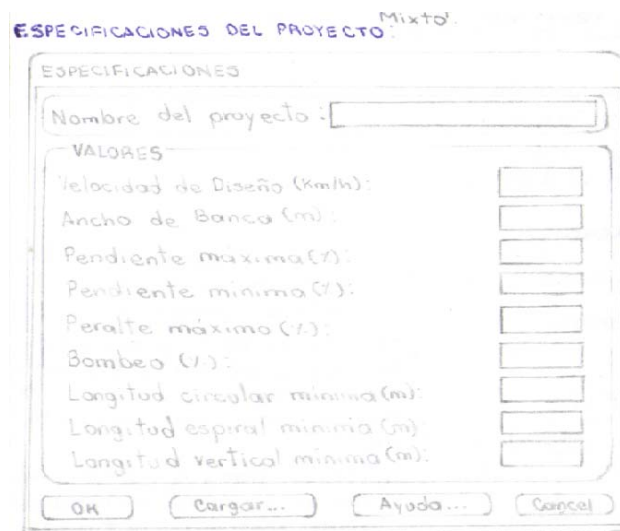


ILUSTRACIÓN 102 BOCETO ESPECIFICACIONES

4.2.4. Implementación:

4.2.4.1. Ayudas:

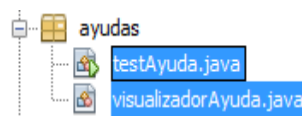


ILUSTRACIÓN 103 PAQUETE AYUDAS

Este paquete es usado por el módulo de especificaciones para darle funcionalidad a cada botón de ayuda que se muestra en especificación. VisualizadorAyuda.java despliega un JComponentes que carga las paginas HTML con el contenido de cada una de las ayudas. TestAyuda permite probar el comportamiento de las ayudas en la fase de desarrollo.

4.2.4.2. ExpRegulares:

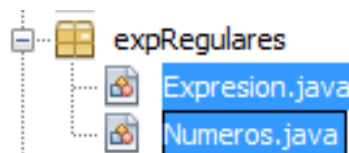


ILUSTRACIÓN 104 PAQUETE EXPREGULARES

La clase números se usa para validar los datos de usuario en el formulario de especificaciones de diseño. La clase expresión se usa para manejar sartas de control.

4.2.4.3. ProyectoVial:

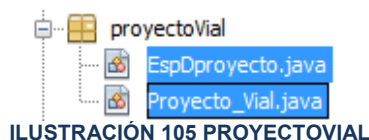


ILUSTRACIÓN 105 PROYECTO VIAL

El Proyecto_Vial guarda los valores del proyecto en construcción, de tal forma que puedan ser accesibles desde otras etapas de diseño que requieran de ésta información. A través de sus banderillas es posible corroborar si la información producto de otras etapas se encuentra disponible. La clase EspDproyecto guarda organizadamente los valores de especificación de proyecto, y el Proyecto_Vial tiene entre sus atributos la especificación.

4.2.4.4. ÁrbolDProyecto:

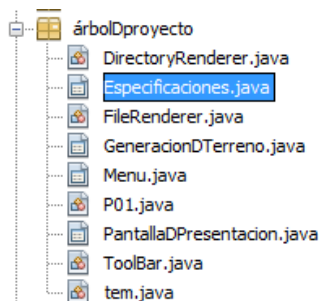


ILUSTRACIÓN 106 PAQUETE ÁRBOLDPROYECTO

Especificaciones es el panel con el cual interactúa el usuario para ingresar sus valores de especificación. A través de él se da uso a las demás clases listadas con anterioridad.

4.3. DISEÑO HORIZONTAL

El diseño horizontal consta de dos partes a saber: el alineamiento horizontal, que es donde se define por dónde va cruzar la carretera, y el diseño de curvas horizontales, que permite diseñar las curvas que deben darse para suavizar las deflexiones en el terreno y garantizar la seguridad en el trayecto.

4.3.1. Alineamiento Horizontal

4.3.1.1. Análisis:

El alineamiento consta de 3 partes: primero la definición de los puntos inicial y final del proyecto, estos deben unirse por la carretera que se diseñe, el segundo es la línea de ceros que demarca el camino con la pendiente adecuada para la vía, y el último paso el trazado de la vía sobre el cual se hará la vía finalmente.

Los puntos de inicio y fin de la carretera han sido evaluados en fases anteriores del proyecto, y el estudiante debe marcarlos sobre el gráfico; la línea de ceros debe dibujarse desde el punto de comienzo hasta el punto final, saltando a través de las curvas de nivel; la pendiente que tiene esta línea no puede exceder los valores establecidos en las especificaciones, y el trazado de la vía, el cual se dibuja sobre la línea de ceros tratando de conservar la trayectoria de esta misma que es la línea guía.

Se ha decidido reproducir el proceso de elaboración de alineamiento horizontal en forma grafica a través de un esquema de instrucciones sobre el gráficator. Esto debido a que el proceso actual es muy interactivo y didáctico y de la manera propuesta las instrucciones podrían ser dedicadas a este proceso con las herramientas que esta amerita.

4.3.1.2. Requisitos:

Actualmente este proceso se hace sobre un plano, ya sea digital o físico, y requiere posicionar adecuadamente los puntos sobre este mismo. Para ubicarse espacialmente, el estudiante utiliza su punto de referencia, el cual, en la aplicación se obtendría de las especificaciones. Adicional a esto, debe calcular las alturas de estos mismos a partir de las curvas de nivel en el plano, las cuales se generan en el módulo de generación de terreno.

El módulo de alineamiento horizontal debe tener las opciones de:

1. Definir los puntos de inicio y fin de la vía

2. Línea de ceros

3. Trazado de Proyecto.

Estas tres actividades deben hacerse secuencialmente en el orden dado anteriormente. Al escoger cualquiera de ellas, debe revisarse la realización de las anteriores y en caso de no estar completado remitirse a esta acción primero. En el caso de que ya se hayan hecho pasos posteriores, o el mismo, debe preguntar si desea reescribir de ese paso en adelante.

Al escoger cada actividad debe desplegarse un lienzo. En este lienzo deben dibujarse las curvas de nivel del terreno junto con las gráficas que se generan de las actividades anteriores que se hayan realizado, puesto que son necesarias para realizar la siguiente.

La actividad de escoger puntos de inicio y fin vial debe realizarse de la siguiente manera:

1. Se despliega el lienzo con las curvas de nivel del terreno y el eje de coordenadas.

2. El usuario mueve el Mouse sobre el lienzo en búsqueda del punto de inicio. La aplicación muestra las coordenadas del cursor sobre el lienzo y sobre la barra inferior del mismo.

3. El usuario encuentra las coordenadas buscadas y da un clic en ese punto. A continuación se dibuja un círculo alrededor de dicho punto, acompañado de la letra C.

4. Se repiten los puntos 2 y 3, esta vez para el punto final, dibujando un círculo acompañado de una F.

5. Se despliega un cuadro informativo con las coordenadas rectangulares (x, y, z) de los dos puntos, preguntando si la información ingresada es la deseada; en caso de ser afirmativo debe guardarse y seguir a la siguiente actividad, en caso de no serlo comenzar la actividad de nuevo.

La actividad de realizar la línea de ceros debe desarrollarse de la siguiente manera:

1. Se despliega el lienzo con los gráficos para la actividad anterior más los puntos C y F.

2. Se dibuja una línea que une el punto de inicio de la vía con el cursor del mouse. El usuario puede ver las coordenadas del cursor, y el color de la línea varía en la medida en que el cursor se desplaza en alguno de estos casos:

- Sobre partes del plano sin información o que no se encuentran sobre una curva de nivel.
- Sobre una curva de nivel a la cual podría llegar una línea con pendiente aceptable.
- Sobre una curva de nivel a la cual podría llegar una línea con pendiente no aceptable.

3. Cuando el usuario hace clic sobre un lugar en el plano, se traza una línea fija desde el punto de inicio al punto de clic, en caso de que este clic fuese dado sobre una curva de nivel, y guarda las coordenadas este punto. De otra forma no hace nada.

4. Se repiten los puntos 2 y 3, pero cambiando el punto de inicio por el último punto de la línea de ceros.

5. Se repite el punto cuatro hasta que el usuario posicione el Mouse sobre el punto final de la vía y de un clic, cuando el mouse se ponga sobre el punto final la línea debe cambiar a otro color.

6. Cuando estén unidos el punto inicial y el final, debe mostrarse un cuadro preguntando si debe guardarse la línea de ceros creada, en caso afirmativo, debe guardarse la línea de ceros y proseguir a la siguiente etapa, en caso contrario, se borra la línea y debe comenzarse de nuevo.

La actividad de realizar el trazado de la vía debe realizarse de la siguiente manera:

1. Se despliega el lienzo con los gráficos para la actividad anterior más la línea de ceros
2. Se dibuja una línea que une el punto de inicio de la vía con la posición actual del cursor, la cual cambia a medida que se desplaza el cursor. Se dibujan también en sentido transversal de esta línea cada abscisado, una línea del largo de la zona de afectación de la vía, la cual debe poder definirse en la misma interfaz gráfica del alineamiento horizontal.

El usuario puede ver las coordenadas del cursor y el color de las líneas variar en la medida en que el cursor se desplaza:

- Sobre partes del plano sin información.
- Sobre partes del plano que podrían formar una línea con una longitud aceptable y tienen información a lo largo de la línea y de las secciones transversales.
- Hay información pero la longitud no es aceptable.

3. El usuario hace clic sobre una parte del plano. Si no hay información no hace nada, si hay información busca las alturas en los puntos a lo largo de la línea central y de las secciones transversales en caso de encontrarse todas las cotas agrega este tramo al trazado y su gráfica queda fija.

4. Se repite el punto 3 hasta que se dé clic sobre el punto final, en ese mismo momento se muestra un mensaje preguntando si se desea guardar el trazado o volver a hacerlo y se realiza la acción escogida.

En el transcurso de todas las actividades debe ser posible regresarse a un paso anterior mediante el comando control +z.

La representación gráfica de estos requisitos puede verse en resumen en el siguiente diagrama, cuyos casos de uso se explican en detalle en la tabla de casos de uso (Ver ANEXO 1 CASOS DE USO).

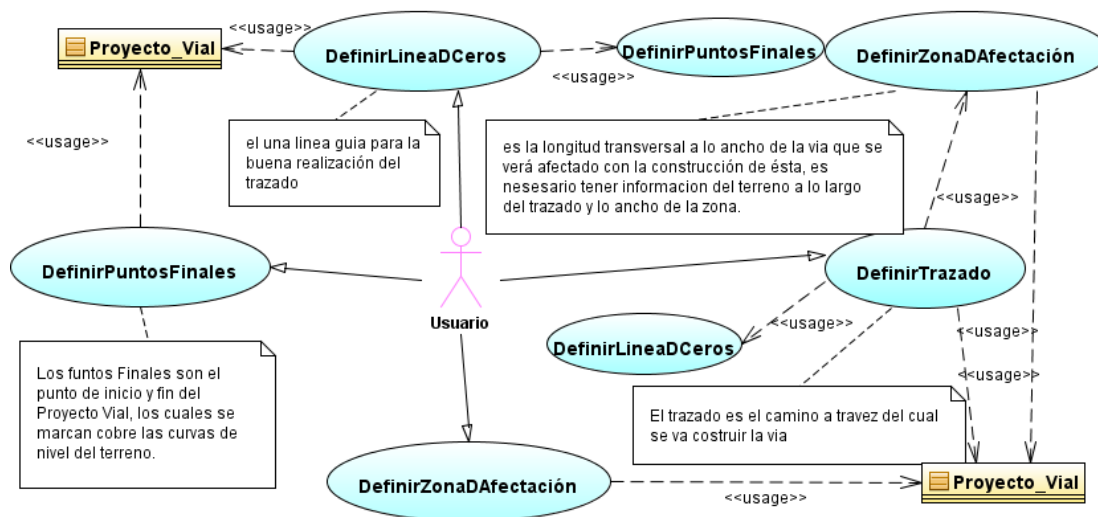


ILUSTRACIÓN 107. CASO DE USO DISEÑO HORIZONTAL

4.3.1.3. Diseño:

4.3.1.3.1. Línea de Ceros y Trazado de Proyecto:

Las clases que son necesarias para recibir los datos de usuario son:

La línea de ceros cuya información debe soportar: un arreglo de puntos tridimensionales, la pendiente de normativa que debe cumplir, su propia pendiente, y la equidistancia de las curvas sobre las cuales se está trazando.

El trazado de proyecto debe tener en cuenta el abscisado que subdivide longitudinalmente la vía, el ancho de la calzada que demarca los bordes externos de la vía transversalmente, información de la ubicación de las deflexiones (PI), banderillas de información de elementos viales, y posición de inserción de las curvas horizontales.

El Proyecto_Vial debe contener tanto la línea de ceros como el trazado de la vía, de tal manera que pueda ser accesible desde los demás módulos que lo

requieran. Las anteriores características son plasmadas en el siguiente diagrama de clases:

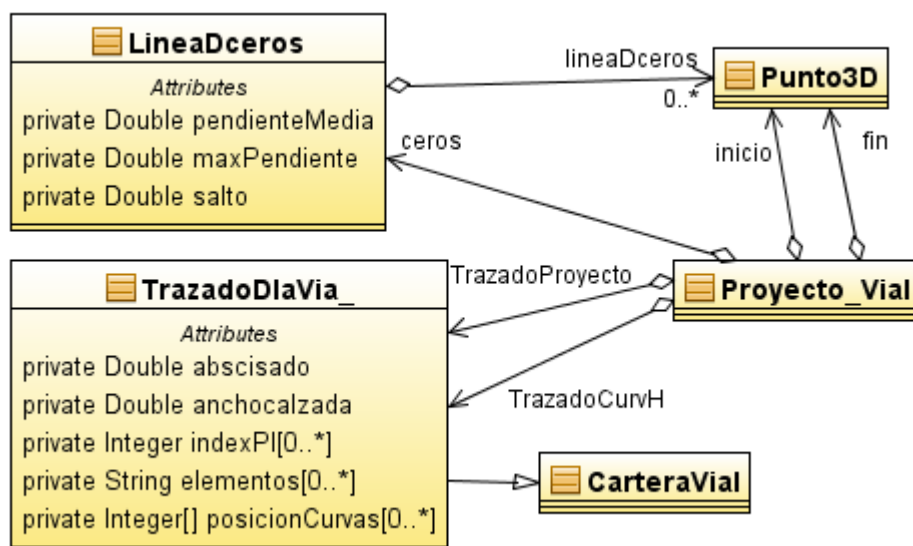


ILUSTRACIÓN 108. DIAGRAMA DE CLASE DISEÑO HORIZONTAL

4.3.1.3.1. Instrucciones:

El esquema de instrucciones se usa para permitir que el usuario interactúe con el lienzo de la herramienta gráfica. La clase Instrucción está dotada de banderillas que le permiten diferenciar entre los distintos eventos que el usuario puede accionar con el mouse y con el teclado.

Cada instrucción que se cree debe implementar métodos de acción para inicializar la instrucción con los valores que requiera, debe definir banderillas de control, métodos que manéjenlos eventos de mouse y teclado, métodos que dibujen graficas temporales y fijas, y métodos que finalicen la instrucción, que retrocedan o adelanten modificaciones sobre el lienzo.

La clase Controladorinstrucciones contiene un lienzo al cual controla, así mismo el lienzo tiene un controlador de instrucciones y el lienzo puede tener varias instrucciones activas a las cuales responde con ayuda de su controlador, el controlador de instrucciones puede tener varias instrucciones, y una activa a la

```

classDiagram
    class Instruccion {
        <<interface>>
        +Attributes
        +Operations
    }
    class ControladorInstrucciones {
        +Attributes
        +Operations
    }
    class Lienzo {
    }
    Instruccion <|-- ControladorInstrucciones
    ControladorInstrucciones --> Lienzo : cl
    ControladorInstrucciones --> Lienzo : Lienzo
  
```

UML Class Diagram: Instruccion Interface and ControladorInstrucciones Class

Instruccion Interface:

- Attributes:**
 - `public Short EVENTO_MOUSE_CLICKED = 1`
 - `public Short EVENTO_MOUSE_PRESSED = 2`
 - `public Short EVENTO_MOUSE_RELEASED = 3`
 - `public Short EVENTO_MOUSE_ENTERED = 4`
 - `public Short EVENTO_MOUSE_EXITED = 5`
 - `public Short EVENTO_MOUSE_DRAGGED = 6`
 - `public Short EVENTO_MOUSE_MOVED = 7`
 - `public Short EVENTO_MOUSEWHEEL_MOVED = 8`
 - `public Short EVENTO_KEY_TYPED = 9`
 - `public Short EVENTO_KEY_PRESSED = 10`
 - `public Short EVENTO_KEY_RELEASED = 11`
- Operations:**
 - `public void iniciar()`
 - `public boolean seNecesitaMasParametros()`
 - `public void reporteEventoMouse(Short claseEventoMouse, MouseEvent e)`
 - `public void reporteEventoMouseWheel(Short claseEventoMouse, MouseWheelEvent e)`
 - `public void reporteEventoKeyEvent(Short claseEventoMouse, KeyEvent e)`
 - `public void finalizar()`
 - `public void cancelar()`
 - `public boolean drawHelper()`
 - `public void setHelper(Capa c)`
 - `public boolean retroceder()`
 - `public boolean aplicar()`

ControladorInstrucciones Class:

- Attributes:**
 - `private boolean libre = true`
- Operations:**
 - `public void setInstruccion(Instruccion inst)`
 - `public void mouseClicked(MouseEvent e)`
 - `public void mousePressed(MouseEvent e)`
 - `public void mouseReleased(MouseEvent e)`
 - `public void mouseEntered(MouseEvent e)`
 - `public void mouseExited(MouseEvent e)`
 - `public void mouseDragged(MouseEvent e)`
 - `public void mouseMoved(MouseEvent e)`
 - `public void mouseWheelMoved(MouseWheelEvent e)`
 - `public void keyTyped(KeyEvent e)`
 - `public void keyPressed(KeyEvent e)`
 - `public void keyReleased(KeyEvent e)`
 - `private void eventoMouse(Short claseEventoMouse, MouseEvent e)`
 - `private void eventoTeclado(Short claseEventoMouse, KeyEvent e)`
 - `private void eventoMouseWheel(Short claseEventoMouse, MouseWheelEvent e)`
 - `private void drawCapaHelper()`

Associations:

- ControladorInstrucciones to Instruccion:** Generalization relationship (solid line with hollow triangle arrow pointing to Instruccion).
- ControladorInstrucciones to Lienzo:** Association relationship with role `cl` (arrow pointing to Lienzo).
- ControladorInstrucciones to Lienzo:** Association relationship with role `Lienzo` (arrow pointing to Lienzo).

4.3.1.3.2. Comando Alturas:

El comando `alturas` también requiere de métodos que busquen las alturas dibujen temporalmente el cuadro y que hagan operaciones de respaldo.

155

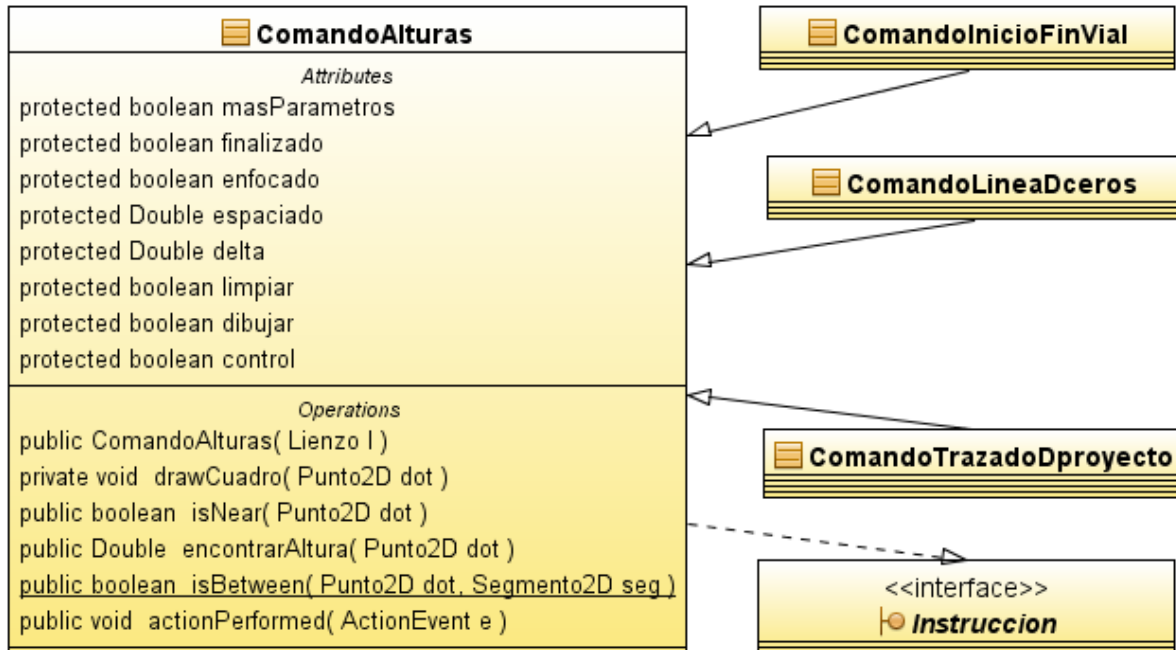


ILUSTRACIÓN 110 COMANDO ALTURAS

4.3.1.3.2. Comando Inicio Fin Vial:

El comando Inicio y fin vial permite dibujar los puntos de inicio y fin de la vía confirmando su valides con el usuario, debe mostrar las coordenadas rectangulares (x, y, z) al usuario y guardar después los datos en el proyecto vial.

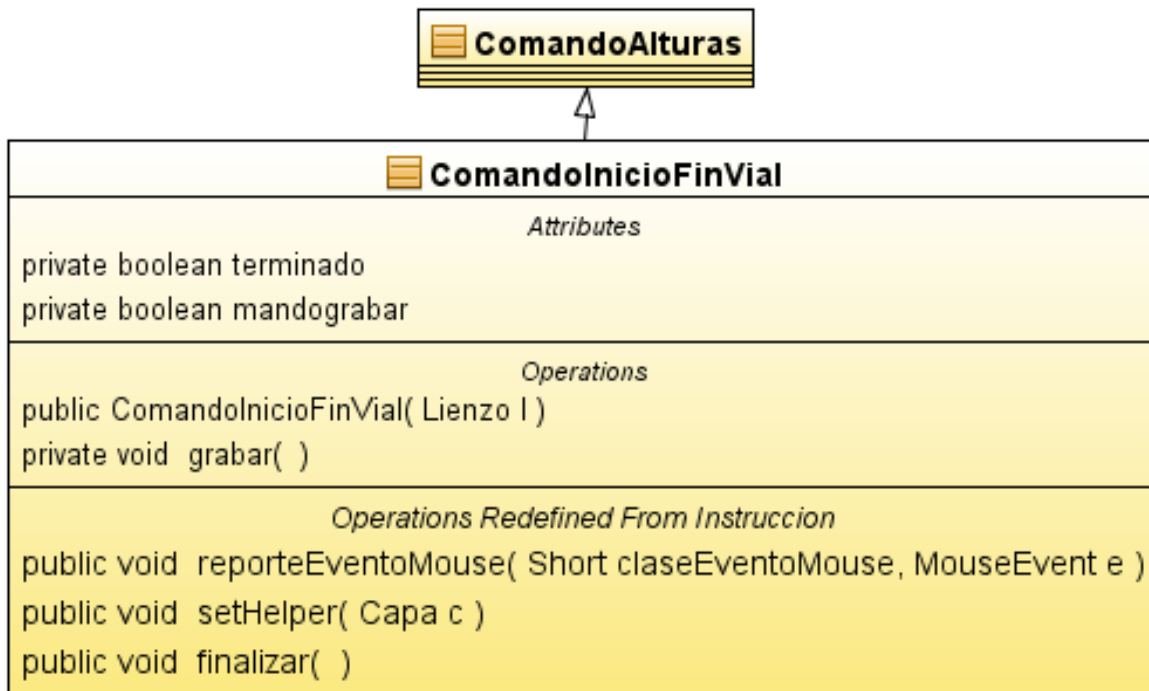


ILUSTRACIÓN 111 COMANDO INICIO FIN VIAL

4.3.1.3.3. Comando Línea de Ceros:

El comando de línea de ceros debe gestionar su pendiente y la equidistancia de las curvas de nivel del terreno, además de redefinir los eventos de manejo del mouse y teclado para agregar las líneas que unen los puntos.

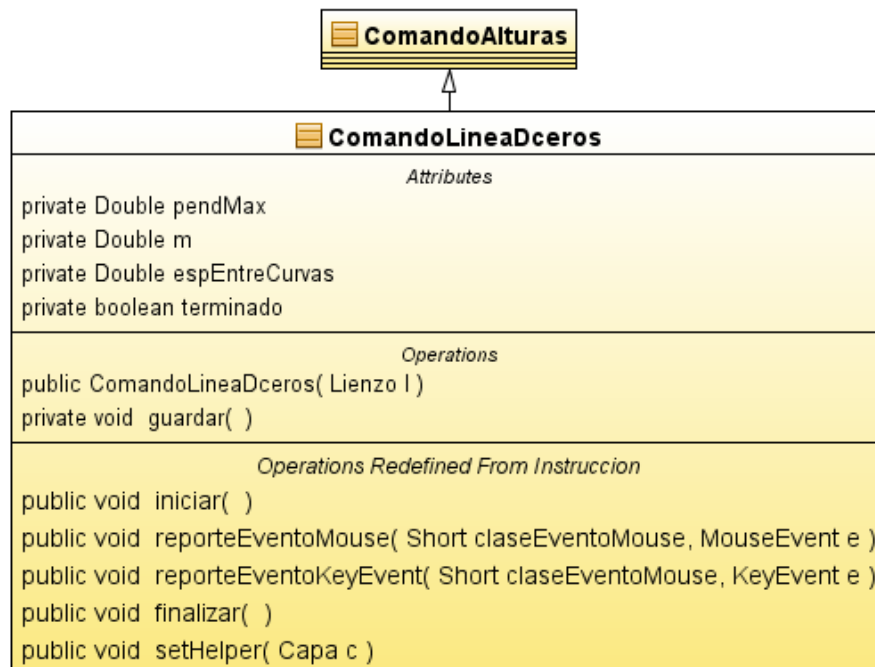


ILUSTRACIÓN 112 COMANDO LINEA DE CEROS

4.3.1.3.4. Comando Trazado de Proyecto:

El comando de trazado de proyecto requiere de información adicional para las secciones transversales, por esta razón contiene un arreglo de puntos tridimensionales e información del abscisado, de la zona de afectación, los metros de normativa por pendiente, los índices de deflexión y la longitud, todo esto con el fin de tener la información suficiente para crear el Trazado de la vía.

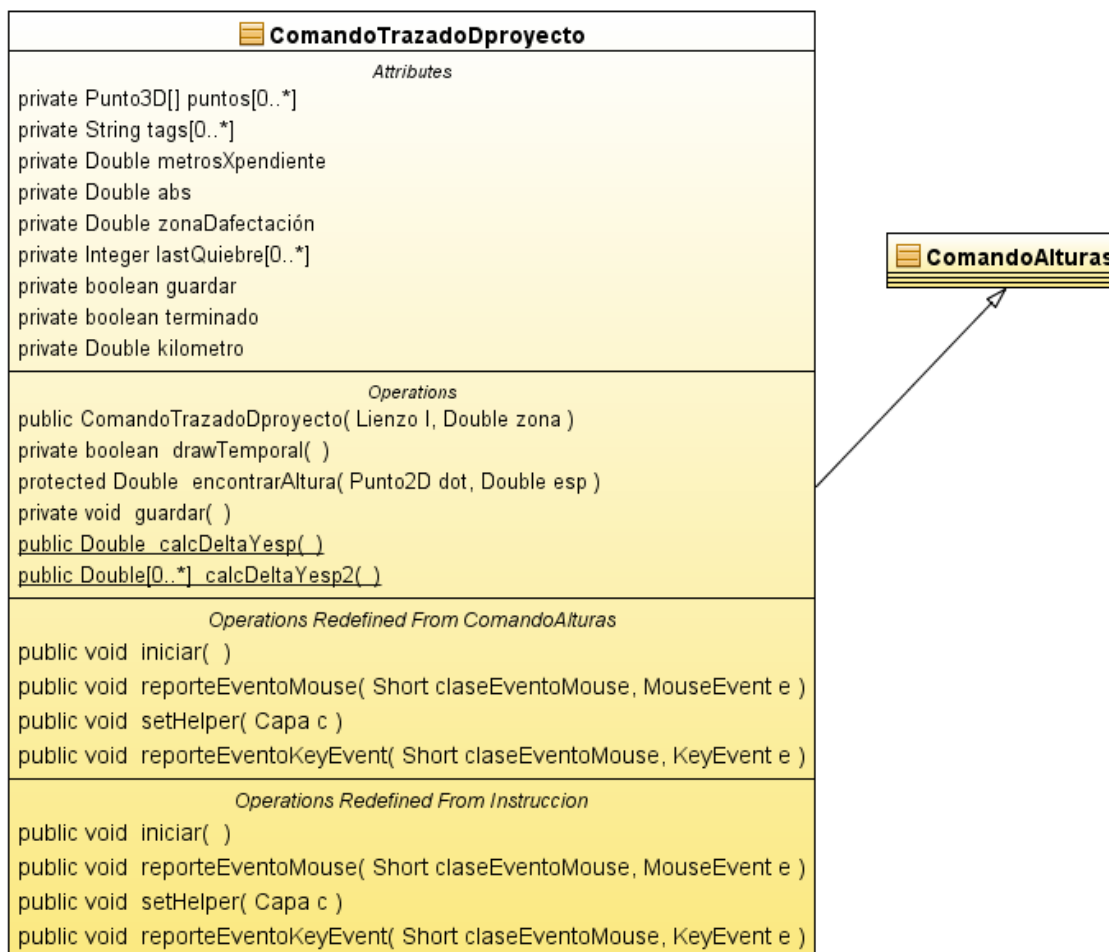


ILUSTRACIÓN 113 COMANDO TRAZADO DE PROYECTO

4.3.1.3.5. Interfaz de usuario:

La interfaz de usuario es muy sencilla debe tener la opción de desplegar el lienzo para marcar el trazado la línea de ceros y los puntos.

DISEÑO HORIZONTAL:

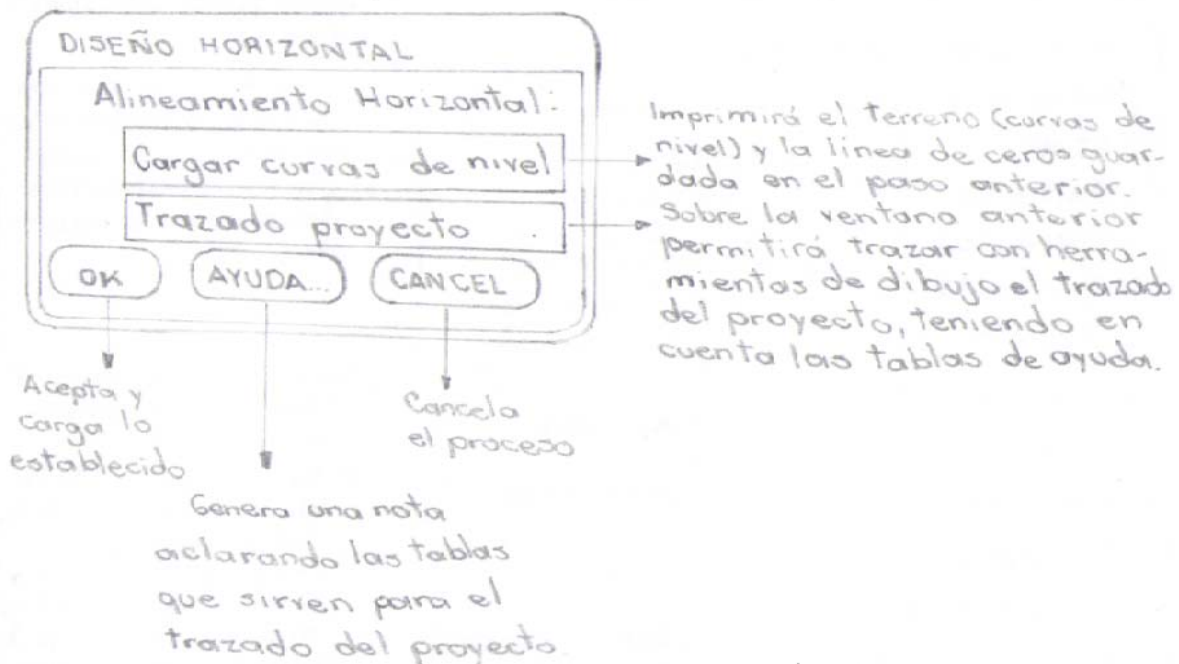


ILUSTRACIÓN 114 INTERFAZ DE USUARIO ALINEAMIENTO HORIZONTAL

4.3.1.4 Implementación:

4.3.1.4.1. Carteras:

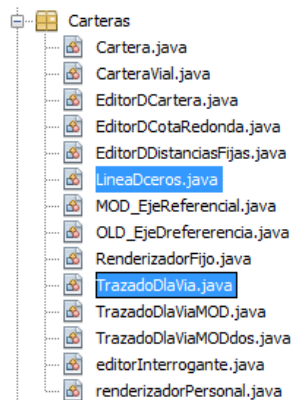
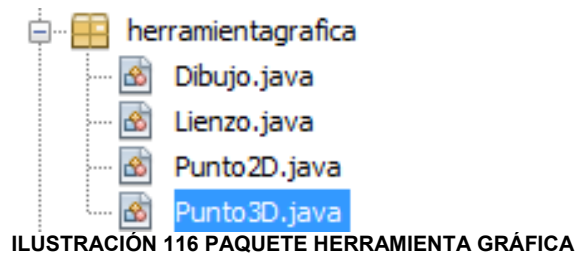


ILUSTRACIÓN 115 PAQUETE CARTERAS

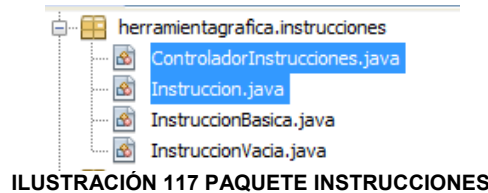
En el paquete cartera encontramos la línea de cerros que guarda el conjunto de puntos tridimensionales de la línea y las funcionalidades para calcular su pendiente

4.3.1.4.2. Herramienta Gráfica:

Dentro de herramienta gráfica se encuentran todo el gráficator incluyendo la clase Punto3D que se usa para definir los puntos de inicio y fin de la vía, los puntos que conforman la línea de cerros y el trazado de proyecto.

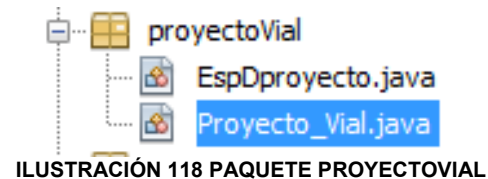


4.3.1.4.3. Instrucciones:



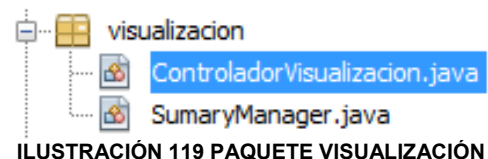
Instrucciones contiene la clase Controladorinstrucciones que maneja las instrucciones de un lienzo y les permite acceder a los eventos que ocurren sobre el lienzo, La interfaz Instrucción es la plantilla que permite crear nuevas instrucciones compatibles con el controlador.

4.3.1.4.4. Proyecto Vial:



El ProyectoVial toma participación en cada módulo puesto que la clase Proyecto_Vial es el puente de información entre todos los módulos, donde se guardan los resultados finales de cada fase y se acceden desde fases posteriores.

4.3.1.4.5. Visualización:



La clase ControladorVisualizacion es bastante usada por que permite desplegar el lienzo que va a interactuar con el usuario.

4.3.1.4.6. Diseño Horizontal:

Dentro del paquete diseño horizontal, se encuentra la clase AlineamientoHorizontal, esta clase es el componente gráfico que se muestra al usuario.

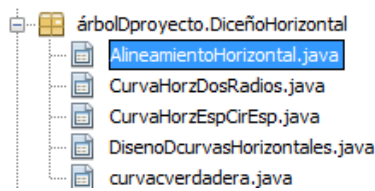


ILUSTRACIÓN 120 PAQUETE DISEÑO HORIZONTAL

4.3.2 Diseño de curvas horizontales

Después de completar el alineamiento horizontal se continua con el diseño de las curvas horizontales, el cual debe hacerse en cada punto donde se encuentren deflexiones en el trazado de proyecto que se ha hecho en la fase anterior, las curvas horizontales se diseñan con el propósito de suavizar el cambio de dirección del trazado dándole seguridad y comodidad al trayecto.

4.3.2.1. Análisis:

Para cumplir con la tarea de diseñar cada curva se acordó mostrar los posibles tipos de curvas a diseñar de tal manera que el usuario escoja la deflexión y el tipo de curva a diseñar, al escoger cada deflexión se le suministra información de la misma, con el fin de que el usuario tome una decisión. Cuando el usuario escoge una tipo de curva se debe desplegar el formulario específico de cada curva, para que el usuario diseñe.

De esta forma el usuario cuenta con buenos criterios para diseñar, antes de escoger un tipo de curva, puede consultar la información pertinente.

4.3.2.2. Requisitos:

El módulo de diseño de curvas horizontales debe tener dos aéreas: una de información y otra de selección de tipos de curvas.

El área de Información debe contener las siguientes casillas para ingreso de datos por parte del usuario:

- Numero de la curva.
- Radio.
- Pendiente relativa

- Peralte circular espiral
- Velocidad de diseño

Junto a las casillas antes mencionadas deben mostrarse otras casillas con la siguiente información del trazado:

- Entretangencia.
- Banca.
- Deflexión.

Junto a las casillas llenadas por el usuario y las casillas llenadas por la aplicación debe mostrarse la casilla de disloque, la cual es llenada por la aplicación usando la información en las demás casillas para calcular su valor.

Al escoger la casilla de número de curva debe mostrarse el rango de posibles deflexiones a diseñar, al digitar un valor correcto, deben llenarse automáticamente las casillas que llena la aplicación automáticamente, y deben habilitarse las que llena el usuario, así como la sección de tipos de curvas para escoger

La sección de tipos de curvas debe contener para escoger las opciones de:

- Circular simple.
- Dos radios.
- Tres radios.
- Espiral – espiral.
- Espiral – circular – espiral
- El botón de ayuda.

Al escoger cada una de las curvas debe desplegarse un formulario específico de cada curva cada uno con la posibilidad de ingresar los datos necesarios para calcular ese tipo de curva, la opción de calcular los elementos de la curva, de ver la gráfica de la curva, ver la cartera, y guardar o borrar del proyecto la curva que se está diseñando. El diagrama que engloba los requisitos aquí descritos de aprecia en la ilustrasen tal se explican en detalle sus casos de uso en la sección tal.

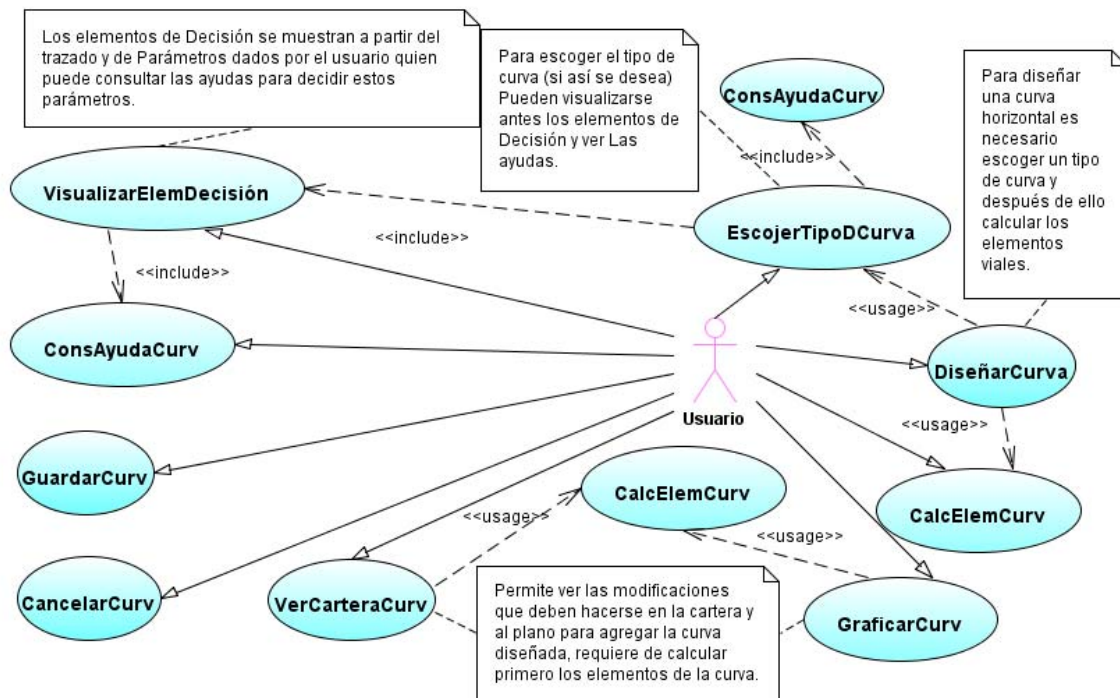


ILUSTRACIÓN 121 DIAGRAMA DE CASOS DE USO DISEÑO DE CURVAS HORIZONTALES

4.3.2.3. Diseño:

Para cada curva se diseña una clase que recibe y calcula sus propios parámetros y un formulario para interactuar con el usuario.

4.3.2.3.1. Curva Horizontal:

La curva circular contiene información de las abscisas del Punto de intersección, Punto de Comienzo de la curva, Punto terminal y deflexión de la curva.

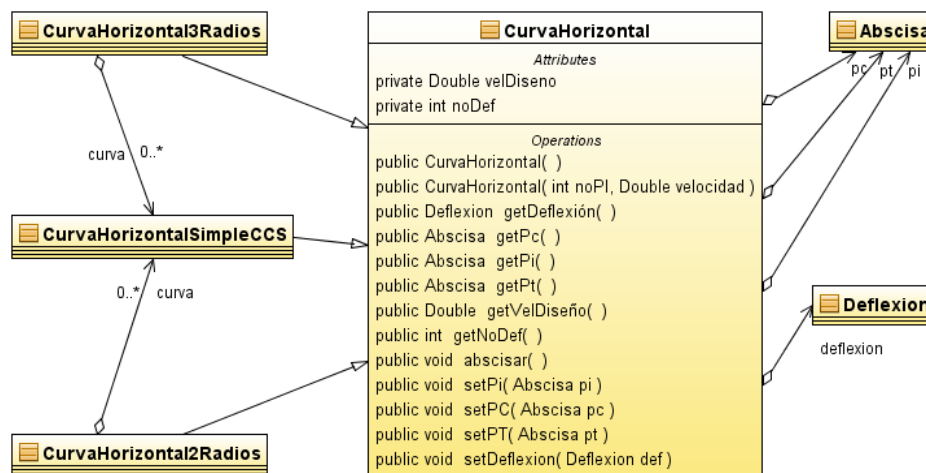


ILUSTRACIÓN 122 CURVA HORIZONTAL

4.3.2.3.2. Circular simple.

La curva circular simple contiene los valores de sus elementos viales métodos get y set para los mismos, y métodos de cálculo de estos valores a partir de los otros.

 CurvaHorizontalSimpleCCS
<i>Attributes</i> <pre> private Double cuerda_C private Double radio_R private Double externa_E private Double tangente_T private Double longitud_L private Double flecha_F private Double CL </pre>
<i>Operations</i> <pre> public CurvaHorizontalSimpleCCS() public CurvaHorizontalSimpleCCS(int no, Double velocidad, Double cuerda) public Double getCuerda_C() public Double getExterna_E() public Angulo getGrado_G() public Double getLongitud_L() public Double getRadio_R() public Double getTangente_T() public Double getFlecha_F() public Double getCL() public void setCuerda(Double c) protected void calGxCyR() private void calRxCyG() private void calRxTyDelta() private void calTxRyDelta() public void calTxEyDelta() private void calLxC_GyDelta() private void calGxC_LyDelta() private void calExTyDelta() private void calFxRyDelta() private void calCLXRyDelta() private void calPC() private void calPT() public void parametroG(Double g) public void parametroR(Double r) public void parametroE(Double e) public void parametroT(Double t) public void parametroL(Double L) public void abscisar2() </pre>
<i>Operations Redefined From CurvaHorizontal</i> <pre> public void abscisar() </pre>

ILUSTRACIÓN 123 CURVA CIRCULAR SIMPLE

4.3.2.3.3. Compuestas de Dos y Tres radios:

Las curvas compuestas se elaboran con la unión de 2 y 3 curvas circulares simples, es por ello que sus clases se agregan de la misma forma, haciendo cambios en los puntos de comienzo y fin de la vía.

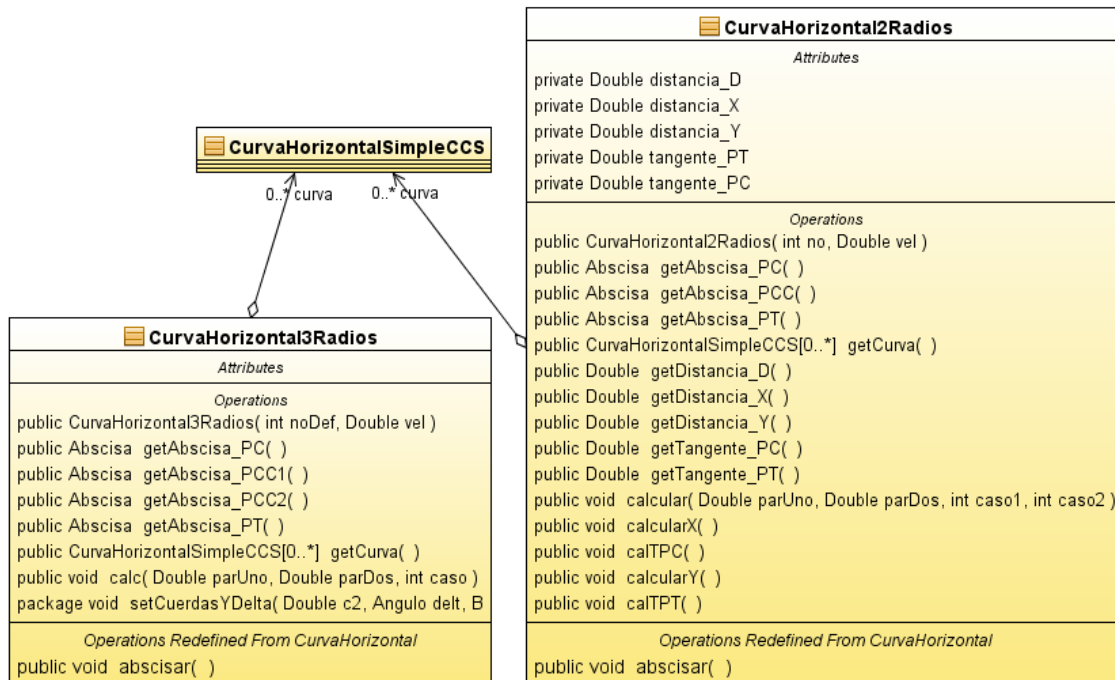


ILUSTRACIÓN 124 CURVAS COMPUESTAS 2 Y 3 RADIOS

4.3.2.3.4. Espiral – espiral Espiral – circular – espiral:

Las curvas espiralizadas, contienen otros elementos de curva, pero básicamente el funcionamiento es el mismo, los atributos enuncian los puntos donde comienzan y terminan sus secciones y las magnitudes que describen su forma. Los métodos implementados guardan y acceden a estos valores y calculan estos mismos a partir de otros ya existentes.

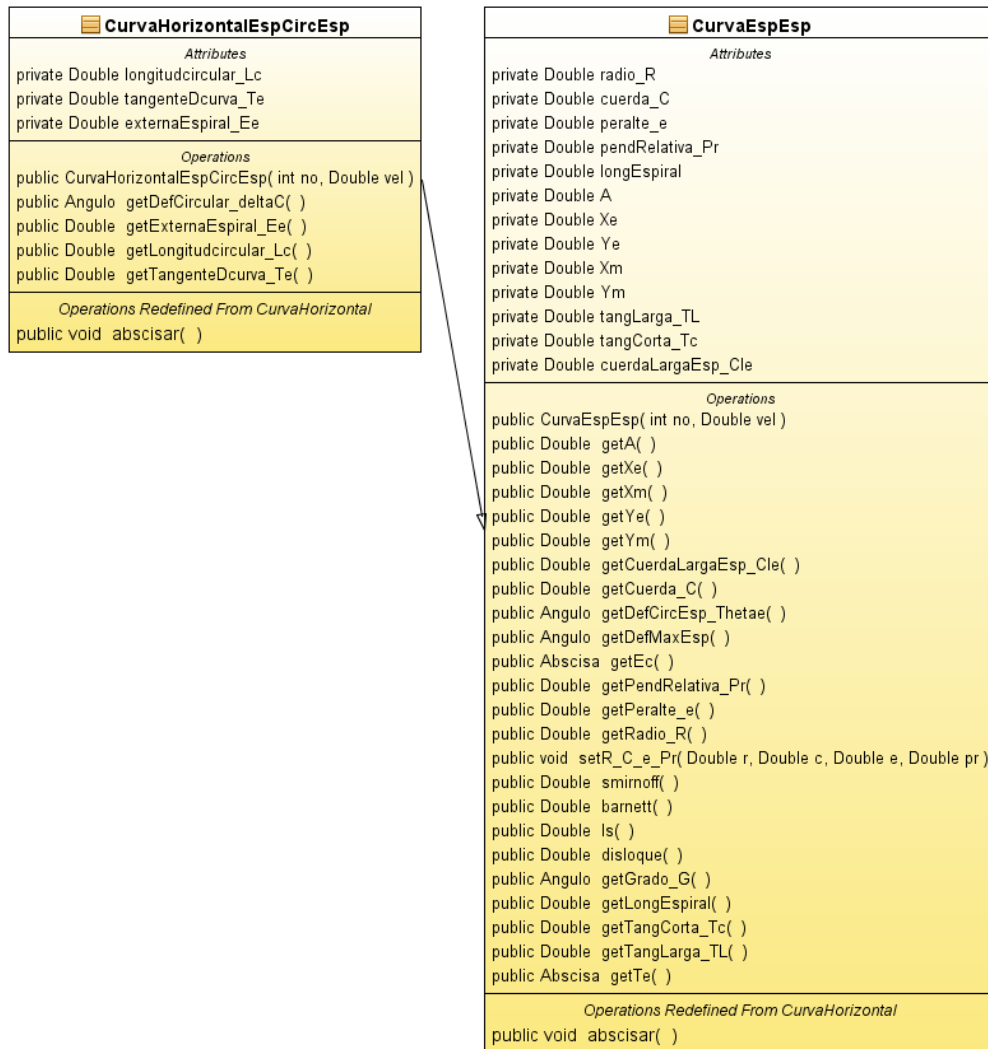


ILUSTRACIÓN 125 CURVAS ESPIRALIZADAS

4.3.2.4. Implementación:

4.3.2.4.1. Carteras:

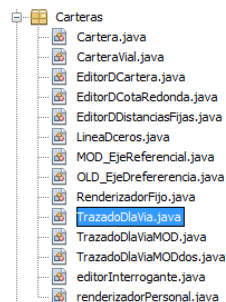


ILUSTRACIÓN 126 PAQUETE CARTERAS

La clase TrazadoDlaVia implementa métodos de inserción de filas, de duplicación y de registro de índices con el fin de ingresar las correcciones en la cartera, que implican el diseño de curvas horizontales.

4.3.2.4.2. Curva horizontal:

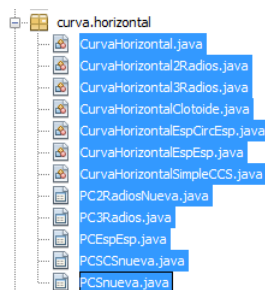


ILUSTRACIÓN 127 PAQUETE CURVA HORIZONTAL

En el paquete de curva horizontal se agrupan las clases pertinentes a cada tipo de curva que se contempla en la aplicación, acompañada de su respectivo formulario que es la interfaz grafica a través de la cual el usuario utiliza cada tipo de curva. Cada clase tiene los atributos de sus elementos, los métodos específicos de cálculo de estos mismos elementos, y métodos de gravado y acceso a los elementos. Los formularios contienen casillas de ingreso de datos, casillas de muestra de datos por parte de la aplicación, y los botones de acción:

- Calcular: muestra una tabla con los elementos de la curva calculados por la aplicación.
- Graficar: después de haber calculado, muestra un lienzo con la gráfica de la curva.
- Cartera: Muestra las correcciones que se agregaran a la cartera producto del diseño de la curva.
- Aceptar: guarda en el proyecto vial la curva diseñada.
- Cancelar: Borra del proyecto vial la curva diseñada y limpia todas las casillas del formulario.
- Ayuda: Despliega la información de ayuda para el diseño de la respectiva curva.

4.3.2.4.3. Proyecto vial:

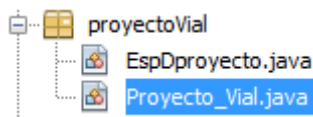


ILUSTRACIÓN 128 PAQUETE PROYECTO VIAL

La clase Proyecto_Vial Contiene los El trazado original de la vía y el trazado corregido con las curvas horizontales diseñadas por el usuario, se accede a estos dos trazados a través de los métodos get de ésta clase.

4.3.2.4.4. Diseño Horizontal:

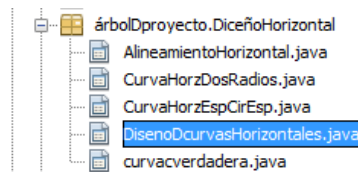


ILUSTRACIÓN 129PAQUETE DISEÑO HORIZONTAL

DiseñoDcurvasHorizontales es el formulario que se muestra al usuario para que consulte información de tipos de curvas, y escoja un tipo de curva en particular.

4.4. TRANCICIÓN DE PERALTE:

La transición de peralte cubre la fase de corrección de la inclinación trasversal de la vía para los tramos de curva (*ver tal lado*), para cada curva diseñada en el proyecto vial debe realizarse una transición del peralte. Esta transición puede hacerse alrededor del eje o por el borde interno o externo de la vía.

4.4.1. Análisis:

La transición de peraltado, se realizara primero escogiendo la curva a la cual se va a diseñar la transición después de ello se permite escoger el tipo de transición:

- Alrededor del eje.
- Borde externo.
- Borde interno

Después de ello el usuario ingresa los datos necesarios para calcular la transición, y la aplicación le muestra al usuario la gráfica del ovni y la tabla de valores de transición.

4.4.2. Requisitos:

Debe diseñarse un formulario que contenga:

- Una casilla de número de curva
- Una casilla de pendiente relativa

- Una casilla de pendiente longitudinal
- Una sección para escoger método de transición:
 - o Alrededor del eje.
 - o Girando borde exterior.
 - o Girando borde interior.
- Un botón ok.
- Botón ayuda.
- Botón cancelar.

El presente formulario debe interactuar con el usuario de la siguiente manera:

El usuario ingresa en la casilla de número de curva una número entero cuya curva ya haya diseñado, ingresa los valores de pendientes relativa y longitudinal, a continuación escoge un tipo de transición. Al accionar el botón aceptar la aplicación deberá validar el valor en el campo de número de curva y pendientes, en caso de estar incorrecto borrar el campo y mostrar un mensaje de error en caso de estar correctos realizar los cálculos de la transición de peralte, y desplegar un grafico del ovni de dicha transición acompañado de la tabla de valores de la transición.

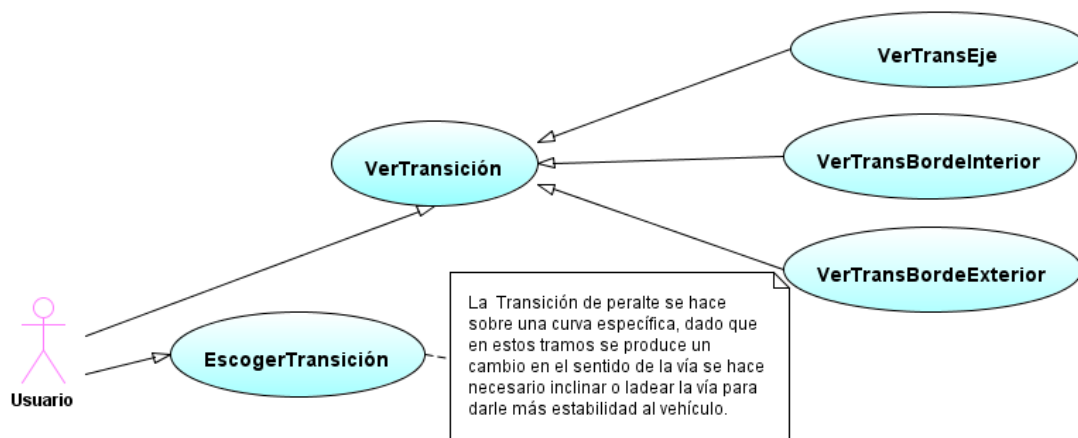


ILUSTRACIÓN 130 CASO DE USO TRANSICIÓN DE PERALTE

4.4.3. Diseño:

4.4.3.1. Transición de Peralte:

La elaboración de la transición de peralte es específica para cada tipo de curva, de acuerdo al tipo de transición aje o bordes interno o externo se calcula de una

forma distinta, pero se pueden generalizar algunas funcionalidades en TransiciónPeralte como muestra el diagrama, lo primero son las banderillas que todas las clases pueden usar para referirse a los tipos de transiciones, y como segundo se encuentran los métodos que permiten acceder a las columnas de la transición, a los gráficos a las tablas y en general los servicios que debe prestar cualquier tipo de transición sin importar los cálculos internos que deba hacer.

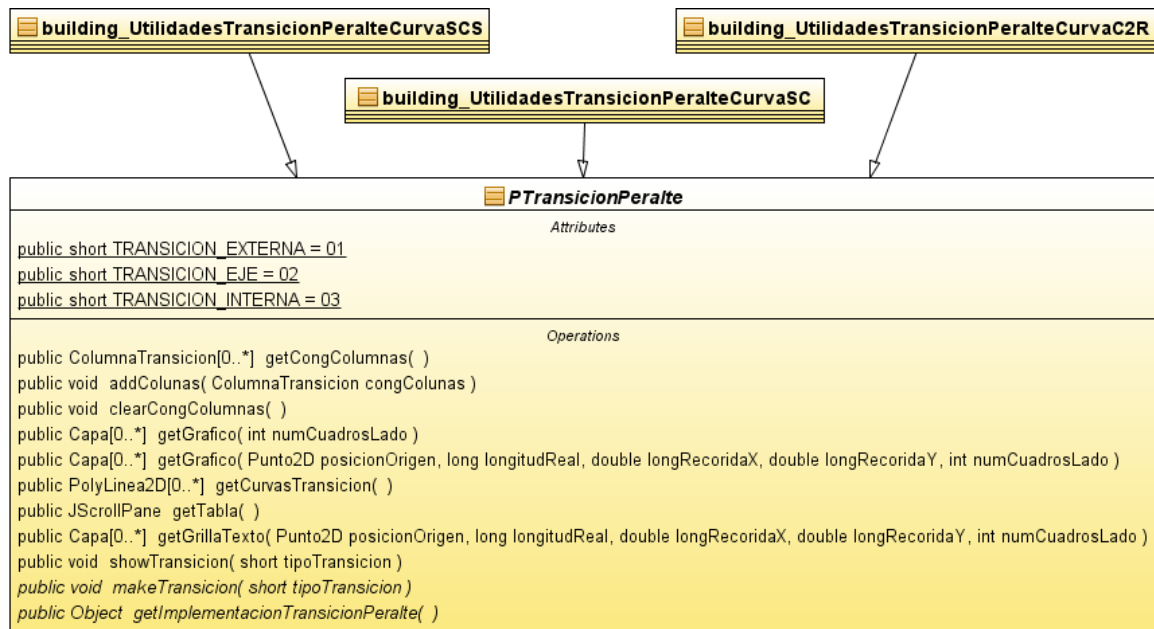


ILUSTRACIÓN 131 DIAGRAMA DE CLASES PTRANSICIONPERALTE

4.4.3.2. Utilidades Transición:

Cada implementación específica tiene sus propios atributos particulares necesarios para calcular la transición de peralte, en el constructor recibe cada uno de los valores para calcular la transición, y el método make de cada una se encarga de hacer el trabajo en cada caso, sabiendo el tipo de transición, los valores calculados por el método make son regresados a través del método getimplementacion ().

 building_UtilidadesTransicionPeralteCurvaSC
<i>Attributes</i>
package double cotaPI package double banca package double bombeo package double peralteMaximo package long cuerda package double deflexion package double tangente package double longitud package double pendRel package double pendLon
<i>Operations</i>
public building_UtilidadesTransicionPeralteCurvaSC(CurvaHsimple curv, Double penRel, Double penLong, Double cotaPI) public building_UtilidadesTransicionPeralteCurvaSC()
<i>Operations Redefined From PTransicionPeralte</i>
public void makeTransicion(short tipoTransicion) public Object getImplementacionTransicionPeralte()

 building_UtilidadesTransicionPeralteCurvaSCS
<i>Attributes</i>
package double cotaPI package double banca package double bombeo package double peralte package double peralteMaximo package long cuerda package double deflexion package double tangenteE package double longitud package double ls package double pendRel package double pendLon
<i>Operations</i>
public building_UtilidadesTransicionPeralteCurvaSCS(CurvaHespCircEsp curv, Double penRel, Double penLong, Double cotaPI) public building_UtilidadesTransicionPeralteCurvaSCS()
<i>Operations Redefined From PTransicionPeralte</i>
public void makeTransicion(short tipoTransicion) public Object getImplementacionTransicionPeralte()

 building_UtilidadesTransicionPeralteCurvaC2R
<i>Attributes</i>
package double cotaPI package double banca package double bombeo1 package double bombeo2 package double peralteMaximo package double e1 package double e2 package double tangPC package double tangPT package double pendRel package double pendLon
<i>Operations</i>
public building_UtilidadesTransicionPeralteCurvaC2R(CurvaH2radios curv, Double penRel, Double penLong, Double cotaPI, double b1, double b2, double e1, double e2) public building_UtilidadesTransicionPeralteCurvaC2R()
<i>Operations Redefined From PTransicionPeralte</i>
public void makeTransicion(short tipoTransicion) public Object getImplementacionTransicionPeralte()

ILUSTRACIÓN 132 DIAGRAMA DE CLASES BUILDING UTILIDADES

4.4.3.3. Interfaz de Usuario:

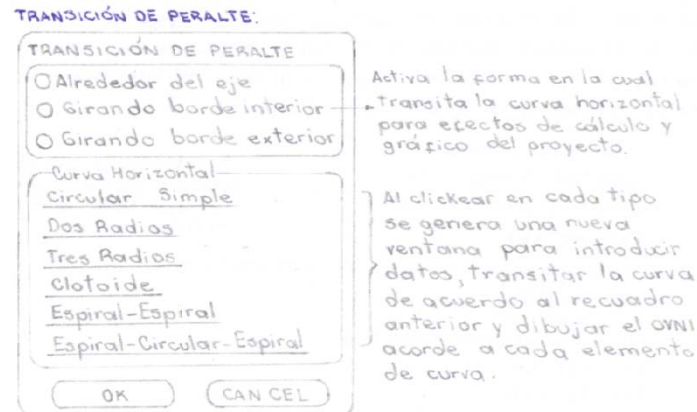


ILUSTRACIÓN 133 BOCETO TRANSICIÓN DE PERALTE

4.4.4. Implementación:

4.4.4.1. Transición de peralte:

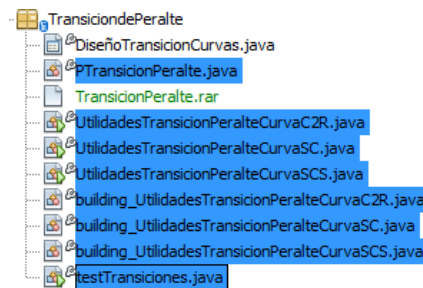


ILUSTRACIÓN 134 PAQUETE TRANSICIÓN DE PERALTE

El paquete de transición de peralte contiene la clase PTransiciónPeralte que sirve de plantilla para los building que recibiendo ciertos valores regresan los cálculos de la transición de peralte para cada tipo de curva específica, también se encuentran las clases utilidades y test que han sido usadas como prueba para corroborar el correcto funcionamiento de las anteriores clases.

4.4.4.2. Proyecto vial:

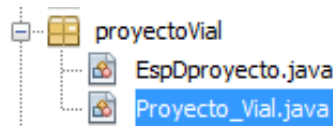


ILUSTRACIÓN 135 PAQUETE PROYECTO VIAL TRANSICIÓN

El proyecto vial implementa los métodos que permiten acceder a las curvas horizontales diseñadas en el módulo anterior, de esta manera recibe los elementos de dicha curva para realizar la transición.

4.4.4.3. Visualización de la Transición:

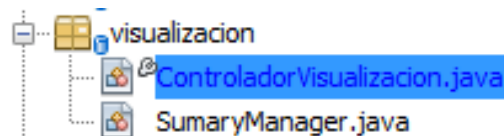


ILUSTRACIÓN 136 PAQUETE VISUALIZACIÓN TRANSICIÓN

El control de visualización recibe la información del tipo de transición, el número de curva, y las pendientes relativa y longitudinal a través de la interfaz de usuario, con esta información hace uso de las clases antes listadas, quienes le suministran los cálculos de la transición que debe presentar al usuario a través de un gráfico y una tabla de valores en el lienzo.

4.4.4.1. Transición de peralte GUI:

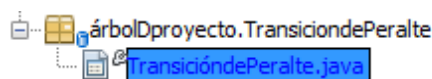


ILUSTRACIÓN 137 PAQUETE TRANSICIÓN DE PERALTE

Contiene el formulario de transición de peralte que utiliza el usuario para pedir sus transiciones.

4.5. AYUDAS:

4.5.1 Análisis:

Al usar la aplicación pueden surgir algunas inquietudes relacionadas a su uso o al diseño de carreteras, con el objetivo de responder a estas inquietudes surge la necesidad de ofrecer un espacio en la aplicación para realizar este tipo de consultas.

Para la realización de este módulo se consideraron las siguientes alternativas:

1. Crear formularios swing empleando un editor de UI como el Net beans para su generación. Esta alternativa presenta varios inconvenientes entre los que se destacan el tiempo necesario para la generación de cada formulario y la necesidad de contar con un editor de UI para java.
2. Usar la cualidad de los componentes swing para desplegar información en formato HTML, un archivo HTML puede ser creado con una gran variedad de herramientas que van desde Dreamweaver hasta un simple editor de texto, sin embargo hay algunos inconvenientes al emplear etiquetas como `<img></img>`, esta etiqueta permite la visualización de imágenes pero al ser usada en un componente swing no realiza ninguna acción, lo que conduce a la no visualización de imágenes¹⁶.
3. Usar un navegador de Internet nativo, esta idea permite el realizar páginas de ayudas que aprovechan al máximo las cualidades de los documentos HTML,

¹⁶ Es posible que esto se pueda solucionar, esto se escapa de los objetivos de este proyecto

sin embargo el hecho de delegar a otra herramienta puede afectar la portabilidad que se quiere promover en la aplicación.

4. Usar una librería en java que permita el manejo de las ayudas, esta opción tiene el inconveniente de en primer lugar encontrar la librería y en segundo lugar lo complejo que puede ser su manejo.

Al analizar las diferentes alternativas se elige la 4, la razón, java cuenta con una librería muy completa que facilita la generación de ventanas de ayudas Java Help.

Java Help usa archivos HTML para la generación de contenido en las ventanas de ayudas además ofrece una barra lateral para búsqueda y lista de los títulos contenidos en las ayudas

4.5.2 Requisitos:

- Generar una ventana de ayuda al oprimir un botón.

4.5.3 Diseño:

Se usará la librería JavaHelp para la generación de ventanas de ayudas, con el objetivo de facilitar su manejo se hará uso del patrón de diseño Facade (ver 2.8.4.5 Facade), con el cual se crea una interfaz a través de la cual se accede a las ayudas. Se elige este enfoque con el objeto de ocultar el uso de JavaHelp y de esta forma permitir el futuro cambio la librería o la forma en que las ayudas son generadas.

Para este caso la clase facade sería visualizadorAyuda, la cual internamente usa las clases HelpBroker y HelpSet de la librería JavaHelp.

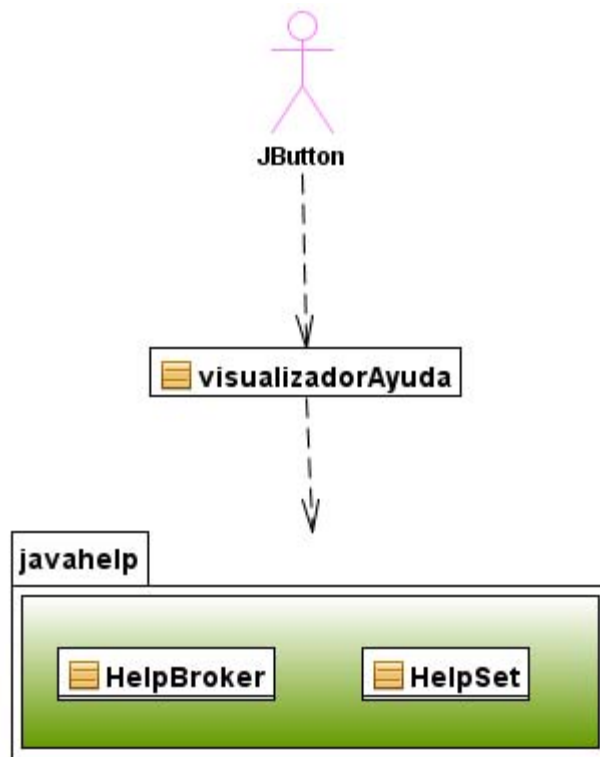


ILUSTRACIÓN 138 DIAGRAMA DE CLASE AYUDAS

Con el objetivo de seleccionar la ayuda a desplegar la clase **visualizadorAyuda** cuenta con dos métodos para la visualizador de ayuda, en el primero se especifica solo el botón que desplegara las ayudas, mostrando por defecto de las ayudas concernientes a la pantalla inicial del Digravi. El segundo método recibe un parámetro adicional en el que se le especifica la ayuda a visualizar.



ILUSTRACIÓN 139 MÉTODOS CLASE VISUALIZADORAYUDA

Los dos métodos son estáticos por lo cual no es necesario el crear una instancia de la clase **visualizadorAyuda**, internamente la clase realiza un manejo similar al realizado en el patrón de diseño Singleton para su funcionamiento.

4.4.4 Implementación:

4.4.4.1 Paquete Utilizado

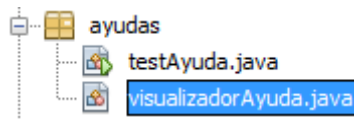


ILUSTRACIÓN 140. PAQUETE AYUDAS

Ayudas: El VisualizadorAyuda permite mostrar la ayuda concerniente a este módulo tanto en cuestión de manejo de la aplicación como de conceptos viales.

5 LIBRERÍA MANEJO GRAFICOS

Al considerar los requisitos que debería satisfacer la aplicación objeto de este proyecto, se observa la necesidad de generar grafico que brinden un soporte para el completo desarrollo de algunas etapas, por ejemplo:

- El la generación de terreno requiere de la visualización de curvas de nivel, con el objetivo de revisar el comportamiento de la cartera topográfica introducida.
- En el alineamiento horizontal, es necesario el visualizar las curvas de nivel para sobre ellas:
 - o Definir el punto inicial y final de la vía
 - o Definir la línea de ceros
 - o Definir el trazada del proyecto
- Al generar las diferentes tipos de curvas horizontales, es necesaria su visualización en el alineamiento.
- En la transición de peralte es necesario su visualización.

De lo anterior se identifican dos necesidades Fundamentales la visualización y la manipulación de los gráficos visualizados, java fue el lenguaje elegido para el desarrollo de este proyecto, se procede a revisar que herramientas ofrece java para la cumplir con estas dos necesidades.

5.1 GUI EN JAVA

Al crear aplicaciones en java hay que recordar que se para el desarrollo de interfaz de usuario(GUI) se cuenta con dos grupos de componentes AWT(Abstract Window Toolkit) y Swing.

5.1.1 AWT

Se emplea para desarrollo de GUI en Java. AWT es librería portable para aplicaciones stand-alone y/o applets. Provee la conexión entra la aplicación java y

el GUI nativo. AWT proporciona un alto nivel de abstracción para los desarrolladores java dado que oculta los detalles del GUI nativo.

Algunas de las características del AWT son:

- Un gran repertorio de componentes para la interfaz de usuario.
- Un robusto modelo para el manejo de eventos.
- Gráficos y herramientas para dibujar, incluyendo Shape, color y tipos de fuentes.
- Manejo flexible de Layout gracias a su independencia del tamaño o resolución de la pantalla.
- Clase para la transferencia de datos, cortar-y-pegar a través del clipboard nativo de la plataforma.

5.1.2 Swing

Swing implementa un grupo de componentes GUI que se fundamentados en AWT y proporciona un Look and Feel pluggable. Swing esta completamente implementado en el lenguaje java, y es basado en el JDK 1.1 lightweight UI Framework.

Algunas de las características del Swing son:

- Todas las características del AWT.
- 100% en java, cuenta con todos los componentes disponibles en AWT (Botones, Barras de desplazamiento, rótulos, etc.).
- Un amplio repertorio de componentes de alto nivel (tales como tree view, list box y tabbed panes).
- Diseñada completamente en java, no depende de peer [Clases e interfaces que conectan los componentes AWT independientes de la plataforma de su implementación dependiente de la plataforma].
- Look and Feel pluggable.

- 5.1.3 AWT Vs. Swing

TABLA 11. PRO Y CONTRA AWT

AWT	
A favor	En contra
- Velocidad: se beneficia de un rendimiento en velocidad al usar componentes nativos. - Look and Feel: los componentes AWT reflejan el Look and Feel del OS donde se ejecuta.	- Portabilidad: el uso de componentes nativos conlleva a limitaciones específicas de la plataforma. Algunos componentes pueden no funcionar en todas o algunas plataformas. - Desarrollo de terceros: la mayoría de fabricantes de componentes, basan el desarrollo de nuevos componentes en Swing. Hay un menor número de componentes AWT disponibles. - Características: AWT no soporta características como iconos y tool-tips.

TABLA 12. PRO Y CONTRA SWING

Swing	
A favor	En contra
- Portabilidad: diseñado completamente en java proporciona pocas limitaciones específicas de la plataforma.	- Desempeño: los componentes swing son generalmente más lentos y lleno de errores que AWT. - Look and Feel: a pesar de que a

- Comportamiento: diseñado completamente en java permite un gran rango de comportamientos en sus componentes dado que no se encuentra limitado a un comportamiento nativo. - Look and Feel: el hecho de que look and feel sea pluggable proporciona un sencillo conjunto de componentes GUI que pueden automáticamente tener el Look and Feel de cualquier plataforma OS.	los componentes Swing se les asignen un Look and Feel del OS donde se ejecuta, pueden no parecer como sus contrapartes nativas.
---	--

Teniendo en cuenta los objetivos de este proyecto es más conveniente el usar Swing, dada su portabilidad y amplio rango de características para personalizar.

5.1.4 Guía para dibujar en Swing

Con el objetivo de generar ciertas figuras personalizadas es necesario el revisar la forma en que se dibujan los diferentes componentes Swing.

- 1- Para los componentes Swing, paint() puede ser invocado por el sistema o la aplicación; update() nunca es invoca por un componente swing.
- 2- Los programas puede realizar llamados a paint() invocando repaint(), pero no deberían llamar a paint() directamente.
- 3- En componentes con complejas salidas, repaint() debería ser invocado con argumentos que definan solo el rectángulo se necesita actualizar, sin embargo la versión sin argumento, ocasionara que se repinte completo el componente.
- 4- La implementación Swing de paint() ocasiona el llamado:
 - 1) paintComponent()

- 2) `paintBorder()`
- 3) `paintChildren()`

Una extensión de un componente swing que implemente su propio código de pintado debería ser colocado en el método `paintComponent()`.

- 5- Una extensión de un componente swing que haya delegado UI (incluyendo `JPanel`), generalmente deberá invocar `super.paintComponent ()` en su implementación de `paintComponent ()`.
- 6- Swing incorpora double-buffering mediante la propiedad `doubleBuffered` de `JComponent`, la cual por defecto se encuentra en `true` para todos los componentes.
- 7- Es muy recomendable que double-buffering se encuentre habilitado para todos los componentes Swing.
- 8- Componente con generen complejos dibujos deberán usar adecuadamente el rectángulo de visualización, con el objeto de solo se dibuje lo necesario.

5.1.5 Java2D

Es un API que proporciona gráficos en dos dimensiones, texto y el procesamiento de imágenes para aplicaciones java a través de la extensión de AWT. Este completo paquete de dibujo incluye un buen repertorio figuras geométricas básicas, texto y imágenes en un flexible, completo Framework para el desarrollo de interfaz de usuario interesantes. Los objetos de Java2D existen en un plano llamado el espacio de usuario. Cuando un objeto es dibujado en una pantalla o impresora, las coordenadas del usuario son transformada en coordenadas del espacio del dispositivo.

Características de API Java2D:

- Un modelo de dibujo uniforme para pantallas e impresoras
- Un amplio repertorio de primitivas geométricas, como curvas, rectángulos, y elipses, además de un mecanismo para dibujar cualquier forma geométrica.

- Un modelo de composición que proporciona control sobre como los objetos se traslapan al dibujarse.
- Fácil gestión de los colores.
- Control en la calidad de los dibujos generados através de pautas de dibujo.

Para usar el API de Java2D es necesario primero el definir el contexto grafico con el que se desea trabajar, el cual en los componentes Swing se puede obtener de la siguiente manera

```
public class ImagePanel extends JPanel {

    protected void paintComponent(Graphics g) {
        // Paint the default look of the button.
        super.paintComponent(g);
        Graphics2D g2 = (Graphics2D) g;

        // Your custom painting here.
        g2.drawImage(foregroundImage, x, y, this);
    }
}
```

Graphics2D proporciona dos grupos de métodos:

- 1) Para dibujar formas
- 2) Afectan la forma en que se dibuja

En el primer grupo se encuentran métodos como:

TABLA 13 TABLA CON LOS METODOS GRAPHICS2D PARA REALIZAR FORMAS

Método	Descripción
draw	Dibuja el exterior de una forma geométrica primitiva usando los atributos stroke y paint.
fill	Dibuja cualquier forma geométrica primitiva rellenado su interior con el color o patrón especificado por el atributo paint.
drawString	dibuja cualquier cadena de texto. El atributo font se usa para convertir la

	fuelle a glyphs que luego se rellenan con el color o patrón especificados por el atributo paint.
drawImage	dibuja la imagen especificada.

Además, **Graphics2D** soporta los métodos de renderizado de **Graphics** para formas particulares, como **drawOval** y **fillRect**.

En el segundo grupos se encuentran métodos como:

- **setStroke**
- **setPaint**
- **setComposite**
- **setTransform**
- **setClip**
- **setFont**
- **setRenderingHints**

5.1.6 CADEMIA y Arquímedes

Java2D es una paquete poderoso, sin embargo presenta el usarlo de forma directa presenta algunos inconvenientes:

1. Es necesario el reescribir el mismo código en varias veces.
2. Java2D las figuras que se representan usando las coordenadas del espacio de usuario.
3. Si una figura cambia es necesario el llamar directamente el método de dibujo que corresponda para que se visualice la modificación.

Al observar esto se hace necesario el buscar herramientas permiten visualización y manipulación de lo visualizado. Este comportamiento es el que observa en las herramientas CAD, en el repertorio de herramientas CAD analizadas se encuentran 2 que son open source y además se encuentran desarrollados en java CADEMIA y Arquimides.

Al analizar el diseño de estas dos herramientas se obtuvieron las siguientes conclusiones:

1. Uso de varios patrones de diseño factory, adapter, bridge, composite, delegate (ver 2.8 Patrones de Diseño).
2. Uso de la arquitectura MVC(ver 2.8.4.6 Modelo Vista Controlador) para el funcionamiento general de la aplicación.
3. Las entidades emplean el patrón de adapter y composite.
4. La generación visual de las entidades se delega a un paquete grafico.
5. Se utiliza un sistema de coordenadas dinámico para la ubicación de las diferentes entidades.
6. Una capa es un clase contenedora de entidades que tiene además unos atributos booleanas para la visualización y manipulación de entidades.
7. Las herramienta que manipula y crea entidades son sencilla interfaces que reciben algunos parámetros y una referencia de la entidad que se vera afectada. Los parámetros pueden venir del Mouse o del teclado por lo cual se emplea un mediador para su interpretación.

5.1.7 Diseño de la librería

La arquitecta inicial elegida para el diseño de la librería fue MVC sin embargo con el objetivo de simplificar el desarrollo se opto por un enfoque M (CV) como el usado por Swing, dado el fuerte enfoque visual de la librería es más conveniente este enfoque ya que no es sencilla el separa la visualización del control que sobre ella se debe tener.

El modelo como tal se encuentra formado por las entidades y capas.

Una entidades representación de un concepto, cada entidad presenta atributos y comportamientos propias, sin embargo todas las entidades presentan algunos comportamientos comunes, como el contar con una representación grafica, el pertenecer a una capa, etc.

Teniendo en cuenta lo anterior se diseñan las entidades empleando el patrón adapter (ver 2.8.4.2 Adapter), donde la Elemento es el adaptador y una entidad es una extensión de esta clase.

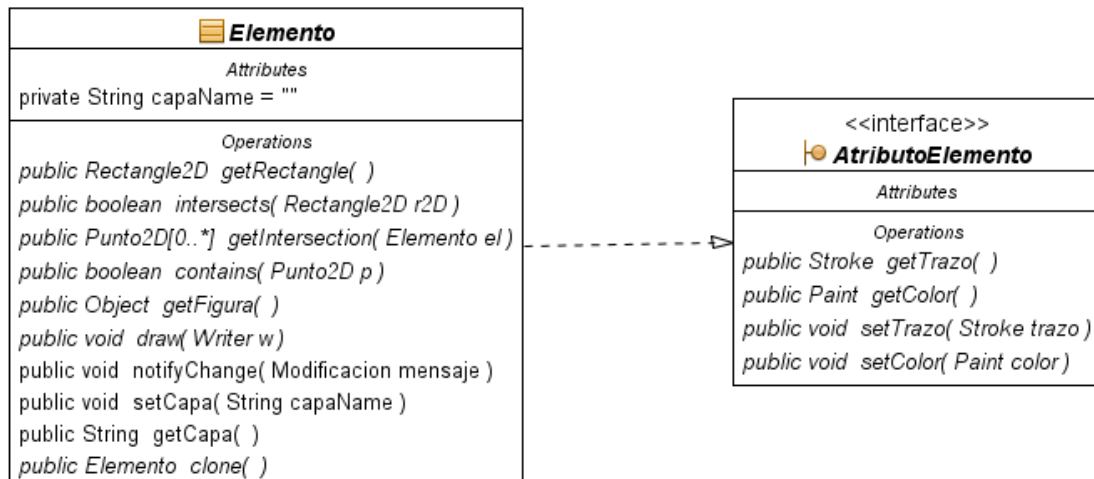


ILUSTRACIÓN 141. DIAGRAMA CLASES DE LAS ENTIDADES

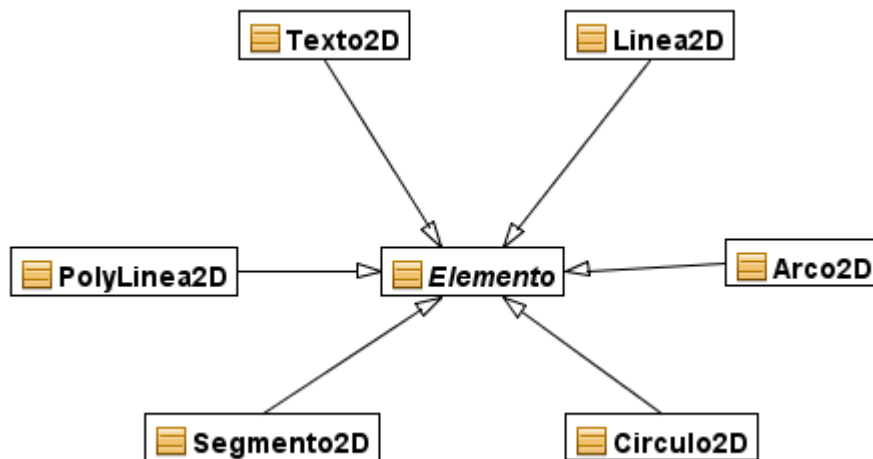


ILUSTRACIÓN 142. DIFERENTES IMPLEMENTACIONES DE ELEMENTO

Una capa es una agrupación de elementos, por defecto cuenta con una configuración para las entidades que son añadidas y no poseen algún atributo definido. Una de las funciones de la capa es el permitir o no la modificación de las entidades que en ella se encuentra a esta cualidad se le denomina cerradura, otra

función es la permitir o no la visualización de las entidades a esta cualidad se le denomina visibilidad.

TABLA 14. ATRIBUTOS CAPA

Atributo	Objetivo
Nombre	Identificar de forma única a una capa
Visibilidad	Gestiona la visibilidad de una capa al usuario.
Cerradura	Gestiona la posibilidad de agregación y remover elementos de una capa

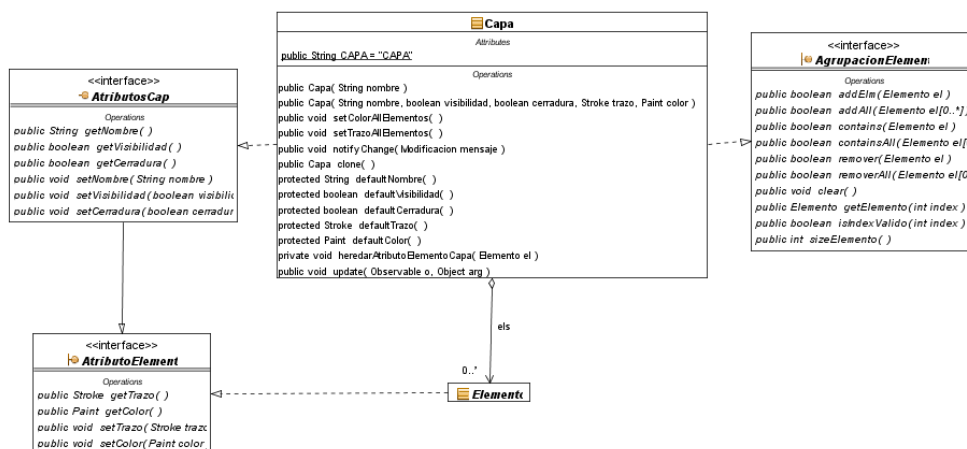


ILUSTRACIÓN 143. DIAGRAMA DE CLASE PARA LA CAPA

Una capa y un elemento se implemento el modelo observador-observado (ver 2.8.4.6 Observer), gracias a esto cada vez que una entidad sufra alguna modificación la capa se informa de la modificación para que tome las medidas necesarias

Para la gestión de las capas y elementos se realiza a través de la clase Dibujo que es más que un simple contenedor, pues permite el elegir una capa como actual y realizar modificaciones a esta clase a través de sus métodos convirtiéndolos en atajos para modificar las capas.

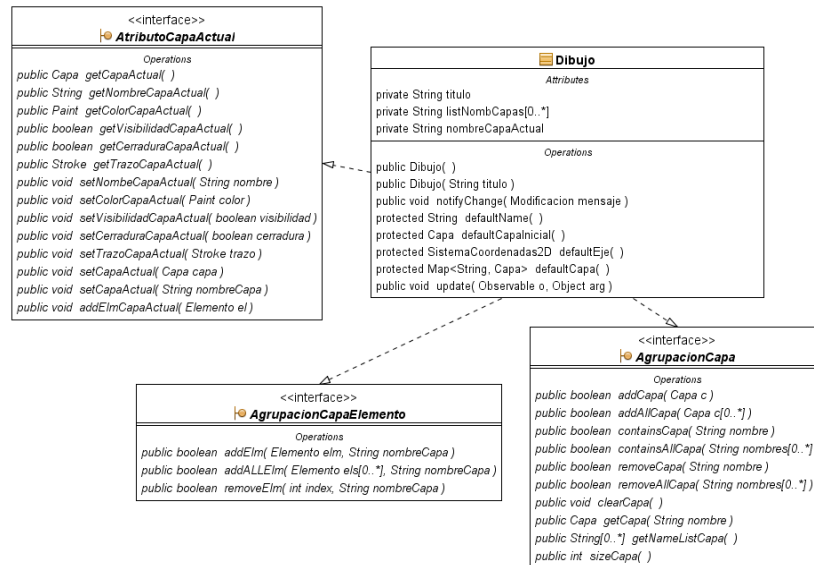


ILUSTRACIÓN 144. DIAGRAMA DE CLASE DE DIBUJO

Entre Dibujo y capa se implemento el modelo Observador-observado, con el objetivo de que se informe al dibujo las modificaciones que ocurran en la capa, en este punto se observa una característica interesante, y el hecho de al modificarse un elemento este informa a la capa, luego la capa puede informar al dibujo que a ocurrido una modificación y de esta forma mantener a los niveles superiores actualizados.

Hasta este punto, no se mencionado nada de la representación grafica de los componentes, hasta el momento se ha enfocado más en la administración de los diferentes elementos que forman la parte conceptual de la librería, a continuación se comentara sobre la generación de gráficos como tal.

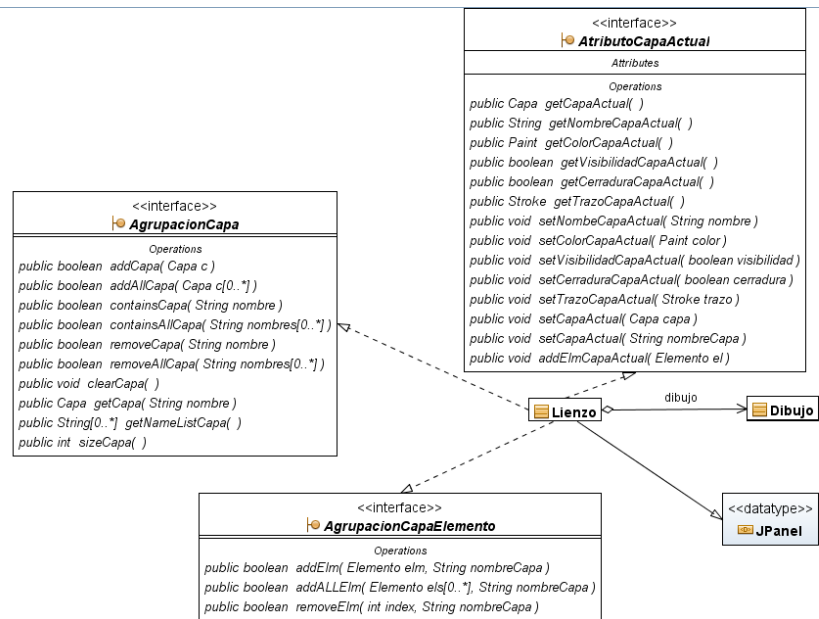


ILUSTRACIÓN 145. DIAGRAMA DE CLASE DEL LIENZO

El clase lienzo proviene de un componente swing, como se comento antes se modificara el método de dibujo con el objetivo de poder visualizar las diferentes entidades y elementos que pueda necesitar un grafico determinado.

Lienzo implementa las interfaces similares a Dibujo, en este sentido Lienzo delega el comportamiento de estos métodos a la instancia de Dibujo que posee.

Como se dijo antes la arquitectura usada para el diseño de esta librería seria un MVC donde la visualización y el controlador se unirían, es por esto que el Lienzo dispone de métodos para modificar la escala y la región que actualmente se visualiza.

Entre la clase Lienzo y Dibujo se implemento el esquema Observador-observado.

La manipulación de entidades es un punto importante para los objetivos de esta librería, y se va a cimentar en el modelo Observador-Observado con el que se ha venido trabajando, la idea es realmente sencilla es recordar que al modificar un atributo en una entidad el observador de la entidad informara de la modificación,

iniciando una cada de llamadas que culminan con el cambio de apariencia de la entidad objeto de la modificación en la interfaz de usuario.

El usuario interactúa con el Lienzo a través de del Mouse y el teclado, teniendo esto en mente se plantea el capturar los eventos que dispare el usuario, esta responsabilidad se delega a una implementación de ControladorInstrucciones el lienzo.

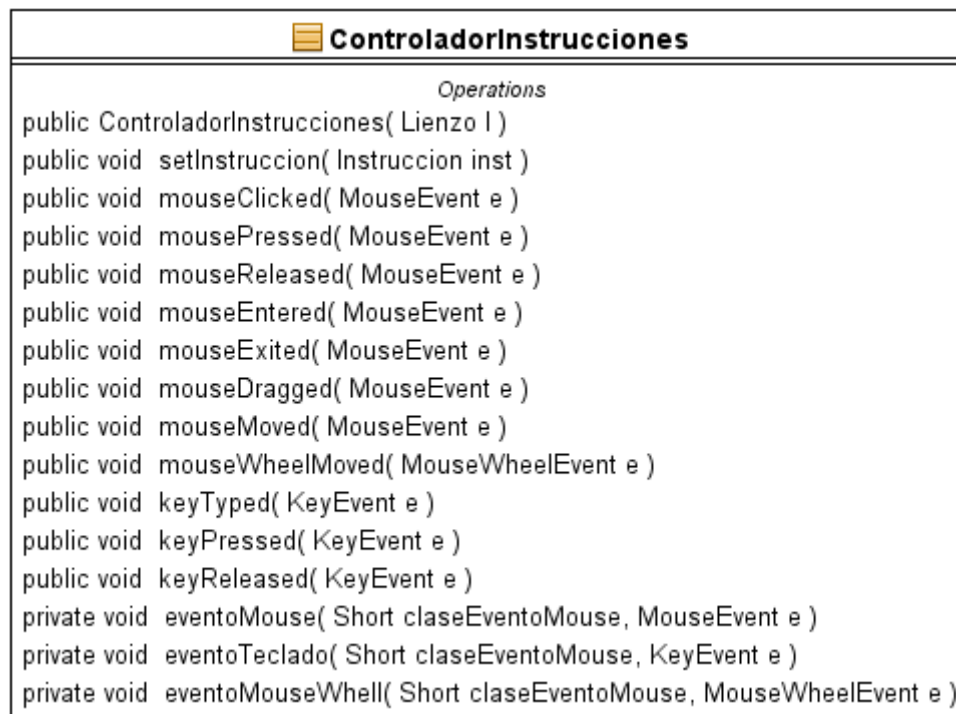


ILUSTRACIÓN 146. DIAGRAMA DE CLASE DEL CONTROLADORINSTRUCCION

Con esto se logra crear interfaz a través de la cual todos los eventos deben del Mouse y teclado deben pasar, esta interfaz recibe como uno de sus parámetros una instancia Instrucción, la implementación de ControllInstruccion debe administrar la instancia de Instrucción informándole de los eventos de Mouse y teclado que dispare el usuario sobre el Lienzo.

La interface Instrucción posee métodos para informarse de eventos de Mouse y teclado que ocurran, una instrucción puede encontrarse en diferentes estados el cual dependerá de los parámetros recibidos.

El ciclo de una instrucción inicia con el llamado el método `iniciar()`, cada vez que ocurre algún evento por parte del usuario se invoca el método `seNecesitaMasParametros()`, en caso afirmativo y de acuerdo al tipo de evento que ocurra se informa a la Instrucción, en caso negativo se procede a invocar el método `finalizar()`, es bueno el comentar que la diferencia entre el método `cancelar()` y `finalizar()` radica en que al `finalizar()` se realizan modificaciones a una entidad mientras que con `cancelar` no.

Los métodos `retroceder()` y `aplicar()` se emplean manejar el historial

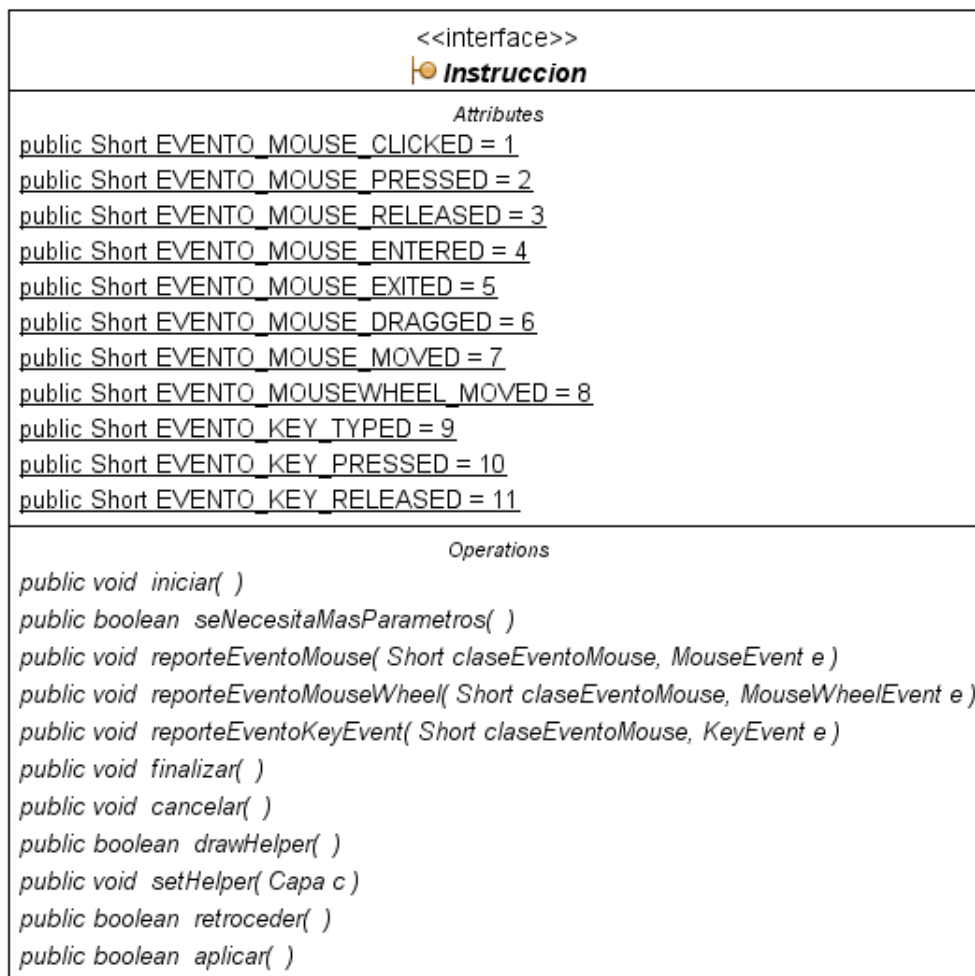


ILUSTRACIÓN 147. DIAGRAMA DE CLASE DE INSTRUCCIÓN

Las instrucciones resultan ser una poderosa herramienta para crear nueva funcionalidades a la librería pues permite el crear un mecanismo fácil para crear y modificar entidades

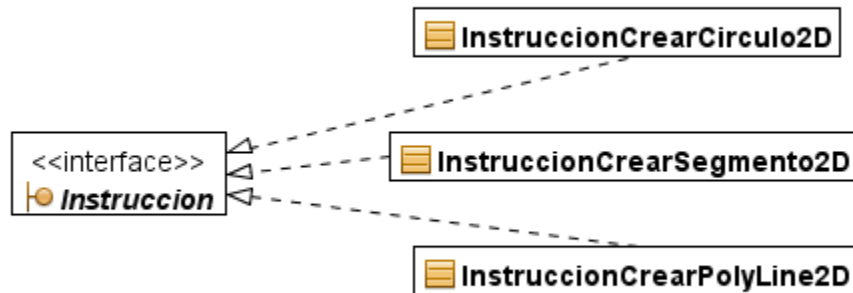


ILUSTRACIÓN 148. ALGUNAS IMPLEMENTACIONES DE INSTRUCCIÓN PARA GENERAR ENTIDADES

Sin embargo se puede las funcionalidades que puede aportar una instancia de Instrucción no se limitan a la manipulación directa de entidades también permiten la implementación se funcionalidades al lienzo como lo son:

- El zoom
- La Translación
- El mostrar la posición

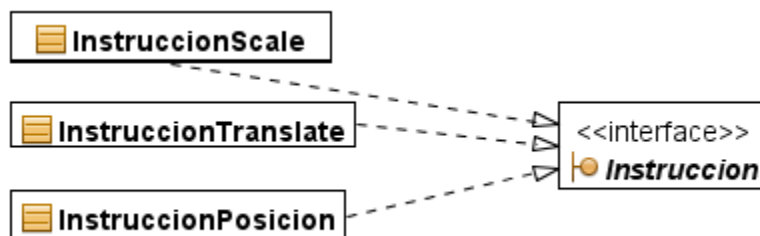


ILUSTRACIÓN 149. ALGUNAS IMPLEMENTACIONES DE INSTRUCCIÓN PARA MANIPULAR LIENZO

6 TALLERES PRÁCTICOS

Se programan con el profesor Jorge Gómez y Germán García la realización de algunas prácticas en las que sus estudiantes usan y realizan observaciones de la herramienta objeto de este proyecto, esto con el fin de corroborar el buen funcionamiento de DIGRAVI.

6.1 OBJETIVOS DEL TALLER

- Socializar el proyecto
- Corregir fallos
- Promover el uso de Digravi
- Mejorar la interfaz de usuario
- Revisar la usabilidad de la aplicación

6.2 METODOLOGÍA

Se realiza una reunión con el docente para acordar la fecha y los temas a desarrollar durante el taller. Con el objetivo de que cada estudiante disponga de un equipo para la realización de la actividad, se ha tomado la decisión de realizarlos en el CENTIC.

Al inicio se hace entrega de un documento digital a cada estudiante, en el que se encuentran las diferentes actividades a desarrollar junto con una serie de preguntas sobre el uso de la aplicación y el desarrollo del taller. Antes de comenzar a desarrollarlo se realizan algunas observaciones y recomendaciones con respecto al mismo.

Al finalizar se pide a los estudiantes que envíen por correo electrónico el documento del taller con sus observaciones, sugerencias y recomendaciones.

6.3 DOCUMENTO DEL TALLER

El documento usado durante el taller tiene el objeto de indicar de forma detallada los pasos a realizar para el cumplimiento de la actividad previamente acordada con el docente, y adicionalmente recopilar información de uso de la aplicación por parte del usuario así como sus comentarios y aportes, los documentos de talleres se encuentran disponibles en el *ANEXO 3 TALLERES*.

6.4 RESULTADO DE LOS TALLERES

Para fortuna del proyecto los usuarios se encuentran bastante receptivos y curiosos respecto a la aplicación



ILUSTRACIÓN 150 TALLERES DE DIGRAVI

6.4.1 Práctica generación de terreno, gráficator y GUI de la Aplicación

El propósito de ésta práctica es enseñar a los usuarios a generar una representación digital del terreno, y visualizarla a través de las herramientas gráficas con las que cuenta la aplicación. Permitir que el usuario interactúe con los formatos de ingreso de datos, y los modifique viendo claramente las implicaciones sobre la representación resultante, para éste fin primero se realiza una

capacitación mostrando el formato de ingreso de datos con el fin de que cada usuario realice previamente una cartera topográfica y trabaje con ella en la actividad, transformándola con los editores internos de DIGRAVI (*Ver 4.1.3.1. Formato Archivos De Excel*).

Una vez creadas las carteras el usuario puede generar las curvas de nivel con diferente equidistancia¹⁷ y cambiando los límites de cotas¹⁸ de inicio y fin de alturas. Se busca también de esta práctica realizar un sondeo de opinión referente a la interfaz de usuario en sus aspectos estéticos y funcionales, (ver ANEXO 3 *Taller Generación de Terreno*).

6.4.1.1 Observaciones

- Es complejo para el usuario regirse a un formato estandarizado para la realización de carteras, externo a la aplicación.
- Se detectaron anomalías en la lectura de los datos a través de los archivos de Excel, los datos con separación decimal cambiaban su valor al cargarse en la aplicación ejemplo: “50.0” → “500” o “50,0” → “500”. Esto lógicamente provocaba comportamientos erróneos en la aplicación.
- Para los usuarios es difícil identificar el logo dado que la letra es muy curveada
- Algunos usuarios no se sienten cómodos con los colores y fondos de DIGRAVI, esperaban algo más moderno y llamativo.

6.4.1.2 Mejoras

- Realizar plantillas de Excel que le faciliten al usuario crear al cartera en forma externa a la aplicación delimitando su uso al formato que DIGRAVI utiliza, fomentar el uso de los editores internos sobre los cuales la aplicación puede ejercer más control y detección de errores, y fortalecer las ayudas con videos de ejemplo que expliquen de forma explícita.

¹⁷ Equidistancia: diferencia de Alturas fija entre las curvas de nivel del terreno que conforma un plano de curvas

¹⁸ Cota: altura del terreno en un punto específico

- Se encontraron las causas del funcionamiento extraño en la lectura de datos, el cual era ocasionado por falta de definición de la codificación en la lectura de archivos externos, al definir la adecuada codificación el problema ha sido solucionado.
- Se ha planteado una propuesta para una nueva interfaz gráfica inspirada en las observaciones hechas por los usuarios en los talleres, la cual será implementada en la siguiente versión de DIGRAVI (*ver ANEXO 4 Propuesta interfaz gráfica*).

6.4.2 Practica especificaciones y ayudas

El propósito de ésta práctica es familiarizar al usuario con los conceptos de especificación de proyecto. Fomentar el uso de las ayudas como apoyo en las decisiones de diseño y hacerle ver las implicaciones producidas por la definición de ciertos valores de especificación.

Mostrar a los usuarios el funcionamiento de las ayudas y la forma como pueden ellos mismos modificarlas, para aportar en su actualización y corrección de los contenidos de la misma. Recolectar información del uso de la aplicación y propuestas de actualización de las ayudas en sus contenidos.

6.4.2.1 Observaciones

- Algunos usuarios con buen dominio del tema hacen buenos aportes conceptuales, que no están contemplados dentro de las ayudas actuales de la aplicación.
- Distribuir el material de talleres es dispendioso y consume buena parte del tiempo disponible, además dispersa la atención de los estudiantes.
- En los talleres prácticos las preguntas y observaciones pueden presentarse simultáneamente, contar con varias personas para resolver dudas y recibir las observaciones sería muy conveniente.

- Algunos estudiantes, llenan los valores de especificaciones con datos exagerados, lo que ocasiona comportamientos no deseados en los módulos siguientes.

6.4.2.2 Mejoras propuestas

- Capacitar a los estudiantes para que puedan modificar con supervisión del docente las ayudas de la aplicación, y de esta manera enriquecer el conocimiento que puede brindar la aplicación.
- Se puede hacer uso del portal de la escuela Albatros para distribuir los talleres con anterioridad, se pueden crear grupos de distribución de correo electrónico.
- Los talleres se realizan con el acompañamiento del profesor, el auxiliar de la materia y los desarrolladores del proyecto.
- La política de manejo de la aplicación, se enfoca en permitir la prueba y error de distintos valores, con el fin de asociar los criterios pedidos con los resultados obtenidos. Se plantea para la siguiente versión de la aplicación una ventanilla informativa que se actualice al enfocar cada campo, y despliegue una descripción rápida del campo.

6.5 RECOMENDACIONES Y OBSERVACIONES

- Es importante la presencia del docente en el desarrollo de la práctica. El docente tiene claridad en los conceptos de vías, lo que lo convierte en la persona idónea para resolver inquietudes.
- La persona que se encargara del taller debe poseer conocimientos en el uso de la aplicación y en el área de vías.
- Es recomendable acordar con el docente un porcentaje en la materia por el desarrollo del taller, de esta manera el estudiante se ve incentivado a participar.
- Se recomienda el uso del CENTIC para el desarrollo del taller, Ya que posee salas con la cantidad de equipos necesarios, además de ayudas didácticas como video beam.

CONCLUSIONES

- Se ha observado que existen distintos perfiles de usuario como lo son los estudiantes, profesores, auxiliares docentes y los mismos estudiantes después como profesionales. Los perfiles de profesor auxiliar y profesional requerirían de herramientas más activas en el proceso de calificación, y en la elaboración de trabajos repetitivos caso en el cual debería permitirse un diseño más fluido.
- El método actual de interpolación da como resultado curvas que presentan en ocasiones discrepancias respecto al terreno debido a que se trabaja con un corredor vial¹⁹ que por cuestiones de logística cuenta con pocos datos y no permite tener información que marque tendencias entre los corredores.
- Se observa en las prácticas el uso de formatos de escritura de datos distintos al de la aplicación que son conceptualmente correctos, así como también se ven errores que se repiten frecuentemente. Los formatos de recepción de datos se han ampliado constantemente desde el inicio de la aplicación hasta la fecha para contemplar este tipo de variaciones, pero con el uso frecuente de la aplicación seguirán encontrándose más casos como estos que plantean una nueva oportunidad para mejorar cada vez más la capacidad de reconocimiento de la aplicación.
- En el desarrollo del proyecto se requirió de la cooperación de estudiantes, profesores y coautores quienes nos asistieron en los aspectos conceptuales del área de vías y en la adquisición de información, el manejo de grupos grandes de colaboradores fue bastante difícil y el hecho de ser un grupo interdisciplinario y con distintos perfiles aumento aún más el reto dado que cada persona tenía horarios de disponibilidad y necesidades distintas acordar tiempos y reuniones no hubiera sido posible sin el apoyo de herramientas de comunicación como el correo electrónico que nos permitía realizar conferencias en línea. También utilizamos herramientas de elaboración de documentos en web y repositorios de código en línea para poder realizar trabajos conjuntos, manejar eficientemente los archivos y no perder los aportes de información de todos los integrantes. Adicional a esto fue necesario el uso de herramientas de administración remota para la resolución y la presentación también de los avances en el proyecto.

¹⁹ Un corredor vial es la sección transversal a lo ancho de una vía de comunicación.

RECOMENDACIONES

- Para una mayor participación del usuario es necesario proporcionar una buena información por lo cual debe implementarse una pantalla en la que despliegue información concerniente a cada campo de ingreso de datos que se encuentre seleccionado.

Fomentar a través de esta herramienta el uso frecuente de las ayudas y su colaboración por parte de los usuarios en el proceso de elaboración de los contenidos de las mismas.

- En la medida de que un usuario de DIGRAVI consiga un buen dominio de la aplicación pueden crearse perfiles expertos, para ello deben cargarse todas las tablas de normativas en la aplicación de tal manera que algunos datos puedan ser insertados de forma automática para trabajos repetitivos, buena parte de éste trabajo ya está implementado y puede ser aprovecharlo en futuros proyectos.
- Permitir que el usuario cargue archivos de configuración de parámetros usados frecuentemente, de tal manera que la aplicación tome información de su uso y avance al ritmo del dominio que el usuario tiene sobre la temática.
- La generación de las curvas de nivel del terreno puede mejorar adicionando distintos métodos de levantamiento topográfico, interpolación y suavización, trabajo que puede hacerse mediante la investigación de las áreas de topografía y análisis numérico.
- Pueden hacerse estudios más profundos en los formatos de ingreso de los datos en cuanto a la detección inteligente de errores a través de expresiones regulares, actualmente es posible descartar los caracteres no correspondientes a los campos y el orden de ellos.
- Se puede agregar a DIGRAVI un intérprete de comandos, para ello es necesario crear los componentes gráficos que muestren este intérprete y crear las expresiones validas de comandos con el uso de la teoría de lenguajes formales.

REFERENCIAS BIBLIOGRÁFICAS

Libros

[1] UNIVERSIDAD INDUSTRIAL DE SANTANDER. Diseño Y Graficación De Vías De Comunicación Asistido Por Una Herramienta De Programación: “Digravi” (Diseñador Y Gráficador De Vías) Fase 1,2009

[2] BRAVO Paulo Emilio, Trazado y Localización de Carreteras, Quinta Edición, Bogotá, 1984

[3] LARMAN Graig. Applying UML and Patterns. Tercera Edición. Pearson Education Inc, 2005

[4] STUMPF Robert V. - TEAGUE LAVETTE C. Object-Oriented Systems Analysis and Design with UML. Pearson Prentice Hall. 2005

[5] MARTIN Rober C. UML para programadores Java, Pearson Prentice Hall, 2003

[7] BRAVO Paulo Emilio. Diseño de carreteras técnica y análisis del proyecto. Sexta Edición, Bogotá, 1998.

[8] MINISTERIO DE TRANSPORTE E INSTITUTO NACIONAL DE VIAS. Manual de Diseño Geométrico de carreteras, Bogotá, 2008.

Enlaces

[1] Diseño Geométrico de Vías

<http://doblevia.wordpress.com/disenio-geometrico-de-vias/>

[2] Diseño asistido por ordenador.

<http://personal.telefonica.terra.es/web/cad/cad.htm>

ANEXO 1 CASOS DE USO

Menú Principal

Nombre:	Escoger Etapa De Diseño
Autor:	Sergio Andrés Hernández Cáceres
Prioridad:	0
Fecha:	22/11/09
Descripción:	Permite acezar a las distintas etapas del diseño Vial.
Actores:	Usuario, Menú, Proyecto_Vial.
Precondiciones:	El usuario debe haber iniciado la aplicación DIGRAVI y encontrarse frente al menú principal.
Flujo Alternativo:	Flujo Normal:
	1. El usuario acciona el botón de la etapa específica.
	2. Se muestra en la pantalla la etapa vial seleccionada por el usuario.
	3. El usuario realiza la etapa vial correspondiente.
	4. El usuario cierra nuevamente la etapa después de terminar.
	5. El usuario se encuentra nuevamente frente al menú principal de la aplicación.
Poscondiciones:	La información de la etapa seleccionada y desarrollada por el usuario se encuentra ahora disponible para ser usada en otras etapas, puede también ser salvada en un archivo o ser visualizada nuevamente en la misma etapa.

Nombre:	Salvar Proyecto Vial
Autor:	Sergio y Diego
Prioridad:	0
Fecha:	21/11/09
Descripción:	Permite salvar el proyecto Vial con todas las modificaciones hechas hasta el momento.
Actores:	Usuario, Proyecto_Vial
Precondiciones:	El usuario debe haber iniciado la aplicación DIGRAVI y encontrarse frente al menú principal.
Flujo Alternativo:	Flujo Normal:
	1. El usuario acciona el botón Salvar.
	2. Se Despliega un selector de Archivos en la pantalla.
3. El usuario cierra el selector y no escoge ningún nombre de archivo valido.	3. El usuario escoge o crea un nombre de archivo valido, en el cual se salvará la información que ha sido ingresada previamente por el usuario en Proyecto_Vial.
4. Se aborta el caso, se muestra un	4. Se pide la información ingresada por el usuario a Proyecto_Vial.

mensaje informativo.	
	5. Se crea un archivo de datos con el nombre suministrado por al usuario.
	6. Se guarda la información entregada por el Proyecto_Vial en el archivo elegido por el usuario.
	7. Se cierra el archivo de datos.
Poscondiciones:	La información del proyecto desarrollado por el usuario está disponible en el archivo especificado por el usuario.

Nombre:	Cargar Proyecto Vial
Autor:	Sergio Andrés Hernández Cáceres
Prioridad:	0
Fecha:	22/11/09
Descripción:	Permite Cargar el proyecto Vial guardado con anterioridad
Actores:	usuario, Proyecto_Vial
Precondiciones:	El usuario debe haber iniciado la aplicación DIGRAVI y encontrarse frente al menú principal, el usuario debe también haber Salvado un Proyecto vial con anterioridad
Flujo Alternativo:	Flujo Normal:
	1. El usuario acciona el botón Cargar
	2. Se Despliega un selector de Archivos en la pantalla.
3. El usuario cierra el selector y no escoge ningún nombre de archivo valido	3. El usuario escoge, un nombre de archivo valido de un proyecto guardado anteriormente, del el cual se cargará la información al Proyecto Vial.
4. se aborta el caso, se muestra un mensaje informativo	4. Se abre el archivo, con el nombre suministrado por el usurario.
	5. Se lee la información del proyecto vial a cargar.
6. La información encontrada no se puede cargar	6. Se guarda la información En Proyecto_Vial
7. se aborta el caso, se muestra un mensaje informativo	7. Se muestra al usuario una lista de la información cargada
Poscondiciones:	La información del proyecto cargado por el usuario está disponible para usar dentro de la aplicación.

Especificaciones:

Nombre:	Ingresar
Autor:	Sergio Andrés Hernández Cáceres
Prioridad:	0
Fecha:	22/11/09
Descripción:	Valida e ingresa en Proyecto_Vial el valor de las especificaciones de proyecto.
Actores:	usuario, Menú, Proyecto_Vial
Precondiciones:	El usuario debe haber iniciado la aplicación DIGRAVI y escoger Especificaciones en el menú principal.
Flujo Alternativo:	Flujo Normal:
1. El usuario, acciona las ayudas concernientes a cada valor de la especificación.	
2. Se despliega el asistente de ayudas con la información teórica pertinente a este valor y las respectivas restricciones de las normativas Colombianas.	
	3. El usuario ingresa todos los datos de especificaciones en los campos marcados para ello.
	4. El usuario acciona el botón aceptar.
	5. Se verifican todos los datos ingresados.
7. Se marcan con un mensaje las casillas con datos errados, se regresa al paso 1.	6. Los datos están todos correctos y se ingresan al Proyecto_Vial.
Poscondiciones:	La información de los valores de especificación está guardada en Proyecto_Vial y puede ser utilizada en las siguientes etapas de diseño.

Nombre:	Borrar
Autor:	Sergio Andrés Hernández Cáceres

Prioridad:	0
Fecha:	22/11/09
Descripción:	Vacía todos los campos de especificaciones para volver a ingresarlos.
Actores:	usuario, Menú, Proyecto_Vial
Precondiciones:	El usuario debe haber iniciado la aplicación DIGRAVI y escoger Especificaciones en el menú principal.
Flujo Alternativo:	Flujo Normal:
	1. El usuario acciona el botón cancelar.
	2. Se borran del formulario todos los campos de especificaciones.
	3. Se borra de Proyecto_Vial los valores de Especificación.
	4.El usuario acciona el botón aceptar.
Poscondiciones:	El modulo de Especificaciones regresa a su estado original, ahora es posible ingresar nuevamente los valores de especificaciones.

Generación De Terreno:

Nombre:	DefinirAzimut
Autor:	Sergio Andrés Hernández Cáceres
Prioridad:	0
Fecha:	06/01/2010
Descripción:	Permite definir el azimut del levantamiento topográfico, El azimut es un ángulo medido desde el norte en dirección del primer tramo del levantamiento topográfico, éste ángulo define la orientación de la cartera en el terreno.
Actores:	usuario, Modulo de Generación de Terreno.
Precondiciones:	El usuario debe escoger el módulo de generación de terreno en el menú principal.
Flujo Alternativo:	Flujo Normal:
	1. El usuario ingresa el valor del azimut en el campo especificado.
	2. El modulo de generación de terreno valida el dato.
3. El dato obtenido no es correcto, Se despliega un mensaje informativo.	3. El azimut se guarda en Proyecto_Vial.

4. se borra el dato del campo de ingreso, se aborta el caso.	4. Se reescribe el azimut sobre el campo haciendo uso de la notación del programa.
Poscondiciones:	El azimut ha sido definido y guardado exitosamente y puede ser utilizado para darle orientación a una cartera topográfica.

Nombre:	Importar
Autor:	Sergio Andrés Hernández Cáceres
Prioridad:	0
Fecha:	06/01/2010
Descripción:	Importa un archivo externo con una cartera topográfica escrita en el formato que es soportado por la aplicación y usa esta cartera para generar el terreno.
Actores:	usuario, Modulo de Generación de Terreno
Precondiciones:	El usuario debe escoger el módulo de generación de terreno en el menú principal.
Flujo Alternativo:	Flujo Normal:
	1. El usuario acciona la opción Importar en el módulo de generación de terrenos.
	2. Se verifica la existencia de un azimut.
3. El azimut no está definido, se muestra un mensaje informativo.	
4. Se usa el caso DefinirAzimut.	
	5. Se despliega un selector de archivos.
6. el usuario no selecciona un archivo valido, se aborta el caso.	6. El usuario selecciona un archivo valido, su dirección se copia en el campo importar.
	7. Se importa el archivo que se encuentra en el campo importar.
8. El archivo no se ha podido leer, se muestra un mensaje de error.	8. Se lee el archivo.
9. No ha sido posible crear la cartera, se muestra un mensaje de error.	9. La cartera Topográfica se crea y guarda correctamente en Proyecto_Vial.
Poscondiciones:	La cartera Topográfica ha sido creada y guardada exitosamente y puede ser utilizada para generar las curvas

	de nivel del terreno.
--	-----------------------

Nombre:	Generar
Autor:	Sergio Andrés Hernández Cáceres
Prioridad:	0
Fecha:	06/01/2010
Descripción:	Permite la creación de una cartera topográfica utilizando el editor interno de la aplicación.
Actores:	usuario, Modulo de Generación de Terreno
Precondiciones:	El usuario debe escoger el módulo de generación de terreno en el menú principal.
Flujo Alternativo:	Flujo Normal:
	1. El usuario acciona la opción generar en el módulo de generación de terrenos.
	2. Se verifica la existencia de un azimut.
3. El azimut no está definido, se muestra un mensaje informativo.	
4. Se usa el caso DefinirAzimut.	
	5. Se reciben los datos necesarios dependiendo del método usado para crear la cartera: (distancias fijas o Cota redonda).
	6. Se despliega el editor de cartera del método correspondiente.
	7. El usuario ingresa los datos de la cartera topográfica y acciona el comando de guardar.
8. Los datos ingresados no son correctos, se marcan las casillas del editor que contengan datos errados con un mensaje de error, regreso al paso 7.	8. Los datos en el editor son validados positivamente.
9. No ha sido posible crear la	9. Los datos son correctos, la cartera topográfica es creada y guardada en el Proyecto_Vial.

cartera, se muestra un mensaje de error.	
Poscondiciones:	La cartera Topográfica ha sido creada y guardada exitosamente, y puede ser utilizada para generar las curvas de nivel del terreno.

Nombre:	Definir Parámetros de Curvas
Autor:	Sergio Andrés Hernández Cáceres
Prioridad:	0
Fecha:	06/01/2010
Descripción:	Recibe del usuario los parámetros requeridos para crear las curvas de nivel del terreno: (Equidistancia, altura de comienzo de las curvas, altura de final).
Actores:	usuario, Modulo de Generación de Terreno
Precondiciones:	El usuario debe escoger el módulo de generación de terreno en el menú principal.
Flujo Alternativo:	Flujo Normal:
	1.El usuario ingresa los datos en los campos destinados para los parámetros de las curvas de nivel.
	2. Se verifican los datos ingresados.
3. Los datos ingresados no son correctos, los campos de los datos se marcan con un mensaje.	
	3. Los datos ingresados son correctos y pueden usarse para la creación de las curvas de nivel.
Poscondiciones:	Los parámetros para la creación de las curvas de nivel han sido definidos, a partir de estos y de la cartera topográfica es posible generar las curvas de nivel del terreno.

Nombre:	Crear Curvas De Nivel
Autor:	Sergio Andrés Hernández Cáceres
Prioridad:	0
Fecha:	06/01/2010
Descripción:	Crea las curvas de nivel del terreno a partir del azimuth de la cartera topográfica y de los parámetros de curvas recibidos.
Actores:	usuario, Modulo de Generación de Terreno
Precondiciones:	El usuario debe escoger el módulo de generación de terreno en el menú principal.
Flujo Alternativo:	Flujo Normal:
	1. El usuario acciona la opción de crear curvas de nivel en el módulo de generación de terreno.

	2. Verificar La existencia del azimuth
3. El azimuth no ha sido definido, se utiliza el caso DefinirAzimut.	
	4.Verificar la existencia de la cartera topográfica.
5. La cartera topográfica no se ha ingresado, se usa el caso Importar o Generar.	
	6. tomar los datos de los campos de parámetros de curvas de nivel.
7. Los campos se encuentran vacios, generar las curvas de nivel con los valores por defecto.	7.Generar las curvas de nivel del terreno haciendo uso del azimuth, la cartera topográfica y los parámetros recibidos anteriormente.
	8. Guardar la representación de las curvas de nivel del terreno en Proyecto_Vial y graficar las curvas de nivel del terreno en un lienzo para mostrar al usuario.
Poscondiciones:	Las curvas de nivel del terreno han sido creadas y guardadas en Proyecto vial, pueden ser visualizadas por el usuario y pueden ser usadas para crear sobre ellas un proyecto vial.

Diseño De Curvas Horizontales:

Nombre:	Visualizar Elementos de Decisión
Autor:	Sergio Andrés Hernández Cáceres
Prioridad:	0
Fecha:	20/01/2010
Descripción:	Muestra la información del trazado y junto con parámetros ingresados por el usuario puede mostrar los elementos para decidir por el tipo de curva que se acomode mejor a la vía.
Actores:	usuario, Módulo de Diseño de Curvas Horizontales
Precondiciones:	El usuario debe haber seleccionado el Módulo de Diseño de Curvas Horizontales en el menú de la aplicación.
Flujo Alternativo:	Flujo Normal:
	1. En el campo de # de curva el usuario marca un numero de curva que se encuentre en el trazado.
	2. Se buscan y cargan los valores de entre tangencia deflexión también los valores de especificación a tener en cuenta para la decisión de una curva.

3. El usuario utiliza el caso ConsAyudaCurv.	
	4. El usuario llena los datos faltantes.
	5. Se calcula el disloque con la información suministrada y se muestra en su respectivo campo.
Poscondiciones:	El Usuario tiene suficiente información para escoger una tipo de curvo horizontal.

Nombre:	Calcular Elementos Curva
Autor:	Sergio Andrés Hernández Cáceres
Prioridad:	0
Fecha:	20/01/2010
Descripción:	Permite Calcular y Visualizar los valores característicos de cada curva los cuales Son necesarios para realizar las correcciones de la cartera y en el plano.
Actores:	usuario, Panel de la curva Horizontal
Precondiciones:	El usuario debe escoger la curva Horizontal del Módulo de curvas Horizontales.
Flujo Alternativo:	Flujo Normal:
	1. Se despliega el formulario de la curva horizontal escogida.
	2. El usuario visualiza los valores que tiene y los que debe ingresar.
3. El usurario consulta las ayudas concernientes al tipo de curva horizontal escogida.	
	4. El usuario llena las casillas de parámetros solicitados.
	5. El usuario acciona el botón calcular.
	6. Se Calculan los valores de la curva y se muestran en una tabla.
Poscondiciones:	El usuario puede ver los elementos de la curva que está diseñando, puede usarlos para graficar la curva o para generar la cartera.

Nombre:	Guardar Curva
Autor:	Sergio Andrés Hernández Cáceres
Prioridad:	0
Fecha:	20/01/2010
Descripción:	Guarda la curva diseñada en Proyecto_Vial para con sus valores y ubicación.
Actores:	usuario, Panel de la curva Horizontal

Precondiciones:	El usuario debe escoger la curva Horizontal del Módulo de curvas Horizontales.
Flujo Alternativo:	Flujo Normal:
1. El usuario acciona el botón cargar, se despliega un mensaje informativo deben calcularse los elementos primero.	1. El usuario utiliza el caso CalcElemCurv para realizar los cálculos de curvas.
	2. El usuario sierra o minimiza la tabla de de elementos de la curva.
	3. el usuario acciona el botón guardar en el formulario de la curva.
Poscondiciones:	4. La curva con todos los elementos ha sido guardada en el proyecto vial así como su ubicación en la cartera modificada.

Nombre:	Cancelar Curva
Autor:	Sergio Andrés Hernández Cáceres
Prioridad:	0
Fecha:	20/01/2010
Descripción:	Regresan todos los valores del formulario de la curva a su estado original para volver a diseñar.
Actores:	usuario, Panel de la curva Horizontal
Precondiciones :	El usuario debe escoger la curva Horizontal del Módulo de curvas Horizontales.
Flujo Alternativo:	Flujo Normal:
	1. La aplicación muestra el formulario de la curva.
	2. El usuario ingresa ciertos valores en el formulario y ase uso de los demás casos de uso.
	3. El usuario decide volver a diseñar la curva y acciona el botón cancelar.
	4. Se borra toda la información ingresada por el usuario en el formulario.
	5. se borra la curva si es que se diseño antes alguna.
Poscondiciones :	El usuario puede volver a diseñar la curva.

Nombre:	Ver Cartera Curva
Autor:	Sergio Andrés Hernández Cáceres
Prioridad:	0
Fecha:	20/01/2010
Descripción:	Permite ver las modificaciones que implican en la cartera de la vía el diseño de la curva.
Actores:	usuario, Panel de la curva Horizontal
Precondiciones:	El usuario debe escoger la curva Horizontal del Módulo de curvas Horizontales.
Flujo Alternativo:	Flujo Normal:
	1. La aplicación muestra el formulario de la curva.
	2. El Usuario hace uso del caso de uso CalcElemCurv.
	3. El usuario acciona el botón Cartera.
	4. Se calculan y se muestran las modificaciones en la cartera.
Poscondiciones:	El usuario puede ver cómo queda su curva sobre la cartera de la vía y tiene un criterio de decisión para modificarla o guardarla así.

Nombre:	Graficar Curva
Autor:	Sergio Andrés Hernández Cáceres
Prioridad:	0
Fecha:	20/01/2010
Descripción:	Grafica la curva horizontal diseñada sobre el trazado de la vía.
Actores:	usuario, Panel de la curva Horizontal
Precondiciones:	El usuario debe escoger la curva Horizontal del Módulo de curvas Horizontales.
Flujo Alternativo:	Flujo Normal:
	1. La aplicación muestra el formulario de la curva.
	2. El Usuario hace uso del caso de uso CalcElemCurv.
	3. El usuario acciona el botón Graficar.
	4. Se calculan y los puntos de la curva y se redibuja el lienzo con el trazado y la nueva curva.
Poscondiciones:	El usuario puede confirmar visualmente como se ve la curva que acaba de diseñar de esta manera puede decidir si modificarla o no.

Alineamiento horizontal:

Nombre:	Definir Puntos Finales
Autor:	Sergio Andrés Hernández Cáceres
Prioridad:	0
Fecha:	06/01/2010
Descripción:	Define sobre una representación gráfica de las curvas de nivel del terreno los puntos de comienzo y fin de la vía.
Actores:	usuario, Proyecto_Vial
Precondiciones:	El usuario debe haber escogido la fase de alineamiento Horizontal en el menú principal.
Flujo Alternativo:	Flujo Normal:
	1. El usuario acciona el botón puntos Finales.
	2. se verifica la existencia de las curvas de nivel del terreno
3. No se han definido las curvas, se muestra un mensaje de error, se aborta el caso.	3. se muestra el lienzo con las curvas de nivel dibujadas sobre él.
	4. El usuario marca con el mouse los puntos sobre el lienzo.
	5. Se captura las coordenadas de los puntos marcados y se muestran al usuario para su verificación.
6. El usuario decide hacerlos de nuevo, regresa al paso 2.	6. El usuario verifica los puntos y estos se guardan en Proyecto_Vial.
Poscondiciones:	Los puntos de inicio y fin de la vía ya están definidos, ahora se puede diseñar una vía entre estos 2 puntos.

Nombre:	Definir Línea de Ceros
Autor:	Sergio Andrés Hernández Cáceres
Prioridad:	0
Fecha:	06/01/2010
Descripción:	Permite dibujar sobre el lienzo la línea de ceros del proyecto, comenzando desde el punto de inicio uniendo las curvas de nivel del terreno y finalizando en el punto final.
Actores:	usuario, Proyecto_Vial
Precondiciones:	El usuario debe haber escogido la fase de alineamiento Horizontal en el menú principal.
Flujo Alternativo:	Flujo Normal:
	1. El usuario acciona el botón línea de ceros.
	2. se verifica la existencia de los puntos iniciales y finales del proyecto.

3. No se han definido las curvas, se usa el caso DefinirPuntosFinales.	3. se muestra el lienzo con las curvas de nivel y los puntos finales.
	4. El usuario une las curvas de nivel comenzando desde el punto de inicio hasta el punto final de la vía procurando conservar la pendiente adecuada de acuerdo a los cambios de color de la línea.
	5. Se muestra un mensaje confirmándose: guardar o hacer nuevamente.
6. el usuario decide hacerlos de nuevo, regresa al paso 3.	6. El usuario decide guardar, la línea de ceros se guarda en Proyecto_Vial.
Poscondiciones:	La línea de Ceros ya está creada, ahora se puede dibujar el trazado sobre ésta.

Nombre:	Definir Zona De Afectación
Autor:	Sergio Andrés Hernández Cáceres
Prioridad:	0
Fecha:	06/01/2010
Descripción:	El usuario puede definir el ancho alrededor del centro de la vía que se verá afectado con el diseño y la construcción, por esta razón es necesario que a lo ancho del trazado contemos con información de las curvas de nivel.
Actores:	usuario, Proyecto_Vial
Precondiciones:	El usuario debe haber escogido la fase de alineamiento Horizontal en el menú principal.
Flujo Alternativo:	Flujo Normal:
	1. El usuario se posiciona sobre el campo zona de afectación.
	2. El usuario Ingresa el valor de la zona de afectación.
	3. el usuario selecciona alguna otra opción del módulo de alineamiento.
	4. Se valida el dato en la casilla de zona de afectación.
5. El dato no es válido, se muestra un mensaje de error en el campo, se aborta el caso.	5. El dato es válido, El Módulo guarda el valor de la zona de afectación.
Poscondiciones:	La zona de afectación ha sido definida y puede ser utilizada por el comando de trazado de proyecto para garantizar la existencia de información a lo ancho de la vía.

Nombre:	Definir Trazado
Autor:	Sergio Andrés Hernández Cáceres
Prioridad:	0
Fecha:	06/01/2010
Descripción:	Asiste en el proceso de dibujar el trazado de la vía entre los puntos inicial y final sobre las curvas de nivel y la línea de ceros conservando el ancho de la zona de afectación.
Actores:	usuario, Proyecto_Vial
Precondiciones:	El usuario debe haber escogido la fase de alineamiento Horizontal en el menú principal.
Flujo Alternativo:	Flujo Normal:
	1. El usuario acciona la opción de Trazado de la Vía en el modulo de Alineamiento.
	2. Se muestra el lienzo con las curvas de nivel del terreno los puntos de inicio y fin de la vía y la línea de ceros.
	3. Es usuario dibuja el trazado procurando apegarse a la línea de ceros, la línea que se dibuja cambia de color conforme el largo o ancho del trazado se salen de las curvas o no cumplen con la normativa.
	4. Se despliega un mensaje que pregunta, Guardar o realizar el trazado nuevamente.
5. el usuario decide hacer el trazado nuevamente, se regresa al paso 2.	5. El usuario decide guardar, se guarda el trazado en Proyecto_Vial.
Poscondiciones:	El trazado de la vía ha sido creado y se encuentra disponible en Proyecto_Vial para ser utilizado en las fases de diseño posteriores.

ANEXO 2 CARTERAS

Cartera de Ejemplo Distancias Fijas

K1+137.28		1683.00	1681.49	1678.94	1678.47	1676.96	1675.47	1674.00	1672.64	1671.29	1669.95	1668.67	1667.48	1666.38
K1+120		1684.25	1682.85	1681.44	1680.04	1678.63	1677.24	1675.83	1674.41	1672.99	1671.54	1669.97	1668.41	1666.89
K1+100		1687.17	1685.59	1683.95	1682.30	1680.73	1679.19	1677.66	1676.12	1674.66	1673.26	1671.88	1670.28	1668.49
K1+080		1690.09	1688.44	1686.94	1685.37	1683.57	1681.63	1679.70	1677.76	1675.80	1673.84	1671.87	1669.91	1667.95
K1+060		1691.18	1689.30	1687.42	1685.53	1683.65	1681.77	1679.89	1678.02	1676.15	1674.27	1672.33	1670.39	1668.46
K1+040		1691.54	1689.49	1687.35	1685.23	1683.11	1680.99	1678.95	1676.99	1675.03	1673.06	1671.10	1669.12	1667.16
K1+020		1691.21	1689.09	1686.97	1684.85	1682.74	1680.66	1678.58	1676.50	1674.42	1672.34	1670.26	1668.19	1666.11
K1+000		1689.15	1687.59	1686.03	1684.28	1682.37	1680.44	1678.51	1676.57	1674.64	1672.77	1671.00	1669.28	1667.52
K0+980		1683.29	1681.80	1680.32	1678.83	1677.34	1675.85	1674.36	1672.87	1671.38	1669.89	1668.40	1666.92	1665.42
K0+960		1676.78	1675.42	1674.07	1672.69	1671.39	1670.15	1668.95	1667.77	1666.69	1665.62	1664.59	1663.58	1662.59
K0+941.87	77° 08' 50"	1675.91	1674.89	1673.87	1672.87	1671.88	1670.94	1670.00	1669.16	1668.33	1667.67	1667.15	1666.62	1666.09
K0+940		1676.93	1675.79	1674.40	1673.05	1671.77	1670.66	1669.75	1669.12	1668.49	1667.90	1667.62	1667.74	1667.52
K0+920		1675.70	1674.33	1673.05	1671.80	1670.58	1669.40	1668.26	1666.92	1665.65	1664.64	1663.85	1663.18	1662.75
K0+900		1673.50	1672.47	1671.34	1670.27	1669.20	1668.16	1666.96	1665.71	1664.36	1663.13	1661.93	1660.05	1659.55
K0+880		1670.79	1669.58	1668.37	1667.27	1666.23	1665.18	1664.14	1663.11	1662.08	1661.01	1659.94	1658.93	1657.92
K0+860		1668.70	1667.67	1666.50	1665.31	1664.12	1662.79	1661.47	1660.22	1659.06	1657.92	1656.88	1655.84	1654.79
K0+840		1666.89	1665.62	1664.34	1663.06	1661.77	1660.46	1659.92	1657.92	1656.70	1655.48	1654.28	1653.10	1651.94
K0+820		1665.09	1663.82	1662.47	1661.17	1659.90	1658.59	1657.28	1655.98	1654.68	1653.39	1652.14	1650.88	1649.46
K0+800		1662.99	1661.74	1660.49	1659.28	1658.07	1656.89	1655.69	1654.43	1653.13	1651.81	1650.50	1648.89	1647.10
K0+780		1664.26	1662.85	1661.45	1660.04	1658.64	1657.23	1655.83	1654.43	1653.00	1651.66	1650.42	1648.90	1647.26
K0+772.20	39° 56' 51"	1664.42	1663.02	1661.61	1660.21	1658.81	1657.40	1656.00	1654.60	1653.19	1651.83	1650.65	1649.24	1647.56
K0+760		1661.27	1659.86	1658.59	1657.41	1656.32	1655.18	1654.00	1652.87	1651.77	1650.85	1649.88	1648.35	1646.86
K0+740		1659.25	1657.85	1656.45	1655.02	1653.57	1652.16	1650.86	1649.60	1648.38	1647.17	1645.95	1644.58	1643.22
K0+720		1657.18	1655.87	1654.41	1652.99	1651.63	1650.42	1649.02	1647.53	1646.06	1644.58	1643.10	1641.64	1640.21
K0+700		1654.55	1653.51	1652.21	1651.03	1649.83	1648.34	1646.85	1645.39	1643.96	1642.53	1641.11	1639.70	1638.33
K0+680		1651.36	1650.71	1650.07	1648.79	1647.44	1646.11	1644.78	1643.41	1642.00	1640.66	1639.26	1637.78	1636.32
K0+660		1647.44	1646.86	1646.27	1645.53	1644.64	1643.64	1642.32	1640.96	1639.60	1638.28	1636.91	1635.52	1634.30
K0+656.86	(-)67° 40' 25 "	1646.47	1645.85	1645.23	1644.62	1644.01	1643.02	1642.00	1640.62	1639.14	1637.87	1636.54	1635.36	1633.78
K0+640		1650.93	1650.24	1649.04	1647.56	1646.07	1644.59	1643.10	1641.75	1640.13	1638.64	1637.23	1635.75	1634.10
K0+620		1651.68	1650.54	1649.22	1647.75	1646.27	1644.79	1643.31	1641.83	1640.36	1638.89	1637.41	1635.93	1634.31
K0+600		1650.68	1649.63	1648.31	1646.98	1645.65	1644.32	1642.93	1641.51	1640.06	1638.63	1637.29	1635.55	1634.31
K0+580		1647.48	1646.26	1645.04	1643.82	1642.60	1641.37	1640.15	1638.89	1637.59	1636.28	1634.95	1633.62	1632.30

K0+567.32	(-)'64°26' 50"	1645.33	1644.10	1642.88	1641.66	1640.44	1639.21	1638.00	1636.78	1635.56	1634.33	1633.11	1631.58	1630.58
K0+560		1646.88	1645.59	1644.28	1642.98	1641.67	1640.37	1639.06	1637.76	1636.46	1635.15	1633.85	1632.54	1631.24
K0+540		1649.80	1648.50	1647.19	1645.89	1644.58	1643.27	1641.96	1640.55	1639.14	1637.73	1636.31	1634.35	1633.37
K0+520		1651.72	1650.36	1648.93	1647.45	1645.97	1644.49	1643.02	1641.54	1640.06	1638.59	1637.12	1635.65	1634.17
K0+500		1652.84	1651.18	1649.83	1648.30	1646.79	1645.28	1643.77	1642.27	1640.76	1639.17	1637.75	1636.24	1634.74
K0+480		1652.57	1650.10	1648.58	1647.11	1645.64	1644.17	1642.70	1641.23	1639.76	1638.29	1636.82	1635.37	1633.91
K0+460		1652.00	1649.97	1648.49	1647.00	1645.51	1644.00	1642.54	1641.07	1639.58	1638.10	1636.62	1635.13	1633.65
K0+440		1651.76	1649.74	1648.50	1647.11	1645.63	1644.14	1642.65	1641.16	1639.68	1638.19	1636.71	1635.22	1633.73
K0+420		1652.90	1650.88	1649.29	1648.00	1646.59	1645.21	1643.87	1642.52	1641.17	1639.82	1638.47	1637.13	1635.79
K0+400		1656.72	1654.89	1653.07	1651.25	1649.57	1648.23	1646.89	1645.55	1644.19	1642.92	1641.70	1640.52	1639.42
K0+380		1660.68	1658.80	1657.00	1655.23	1653.44	1651.65	1649.93	1648.95	1648.00	1646.88	1645.82	1644.94	1644.06
K0+371.83	70'57'09"	1662.16	1660.40	1658.60	1656.68	1654.86	1653.33	1652.00	1650.69	1649.57	1648.66	1647.76	1646.90	1646.03
K0+360		1659.40	1657.91	1656.41	1654.91	1653.38	1651.83	1650.29	1648.86	1647.47	1646.11	1644.80	1643.49	1642.18
K0+340		1656.71	1655.30	1653.95	1652.49	1651.15	1649.87	1648.51	1647.14	1645.75	1644.31	1642.87	1641.44	1640.00
K0+320		1654.39	1653.00	1651.57	1650.14	1648.72	1647.30	1645.87	1644.48	1643.09	1641.73	1640.44	1639.17	1637.89
K0+300		1652.50	1651.10	1649.70	1648.31	1646.86	1645.40	1643.93	1642.50	1641.04	1639.60	1638.16	1636.73	1635.29
K0+280		1651.70	1650.58	1649.33	1647.90	1646.48	1645.02	1643.55	1642.08	1640.61	1639.10	1637.59	1636.15	1634.67
K0+260		1650.44	1649.06	1647.62	1646.18	1644.75	1643.33	1641.89	1640.49	1639.04	1637.60	1636.18	1634.78	1633.34
K0+240		1650.00	1648.52	1647.05	1645.59	1644.11	1642.62	1641.13	1639.67	1638.17	1636.79	1635.22	1633.74	1632.26
K0+220		1649.10	1647.62	1646.15	1644.68	1643.20	1641.73	1640.26	1638.79	1637.32	1635.86	1634.38	1632.91	1631.44
K0+212.72	41'04'59"	1649.03	1647.57	1646.06	1644.54	1643.03	1641.52	1640.00	1638.48	1637.00	1635.52	1634.05	1632.57	1631.10
K0+200		1646.79	1645.22	1643.65	1642.10	1640.57	1639.13	1637.73	1636.41	1635.25	1634.16	1633.19	1632.23	1631.27
K0+180		1644.06	1642.60	1641.19	1639.80	1638.58	1637.02	1635.64	1634.25	1632.72	1631.11	1629.48	1627.94	1626.81
K0+160		1643.28	1641.83	1640.36	1638.88	1637.40	1635.92	1634.44	1632.96	1631.48	1630.00	1628.52	1627.05	1625.13
K0+140		1642.16	1640.66	1639.16	1637.66	1636.18	1634.74	1633.33	1631.89	1630.43	1628.98	1627.54	1626.10	1624.08
K0+120		1642.19	1640.70	1639.21	1637.72	1636.24	1634.75	1633.26	1631.77	1630.28	1628.79	1627.31	1625.67	RIO
K0+100		1642.58	1641.09	1639.60	1638.12	1636.63	1635.14	1633.65	1632.16	1630.67	1629.18	1627.69	1626.21	RIO
K0+80		1641.77	1640.32	1638.95	1637.55	1636.14	1634.70	1633.42	1631.78	1630.32	1628.86	1627.40	1625.92	RIO
K0+60		1641.23	1639.77	1638.31	1636.86	1635.40	1633.94	1632.48	1631.00	1629.52	1628.03	1626.54	1624.11	RIO
K0+40		1639.02	1637.77	1636.51	1635.25	1634.00	1632.64	1631.25	1629.83	1628.41	1626.99	1625.59	1624.22	RIO
K0+20		1635.42	1634.39	1633.25	1632.04	1630.78	1629.53	1628.17	1627.01	1625.67	1624.00	RIO	RIO	RIO
K0+00		1630.77	1630.02	1629.06	1628.04	1627.05	1626.05	1624.94	RIO	RIO	RIO	RIO	RIO	RIO
ABSCISA	DEFLEXIÓN	30	25	20	15	10	5	EJE	5	10	15	20	25	30
		IZQUIERDA						CENTRAL	DERECHA					

Cartera de Ejemplo Cota Redonda

		1683	1681.49	1678.94	1678.47	1676.96	1675.47	1674	1672.64	1671.29	1669.95	1668.67	1667.48	1666.38
K1+137.28		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1684.25	1682.85	1681.44	1680.04	1678.63	1677.24	1675.83	1674.41	1672.99	1671.54	1669.97	1668.41	1666.89
K1+120		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1687.17	1685.59	1683.95	1682.3	1680.73	1679.19	1677.66	1676.12	1674.66	1673.26	1671.88	1670.28	1668.49
K1+100		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1690.09	1688.44	1686.94	1685.37	1683.57	1681.63	1679.7	1677.76	1675.8	1673.84	1671.87	1669.91	1667.95
K1+080		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1691.18	1689.3	1687.42	1685.53	1683.65	1681.77	1679.89	1678.02	1676.15	1674.27	1672.33	1670.39	1668.46
K1+060		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1691.54	1689.49	1687.35	1685.23	1683.11	1680.99	1678.95	1676.99	1675.03	1673.06	1671.1	1669.12	1667.16
K1+040		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1691.21	1689.09	1686.97	1684.85	1682.74	1680.66	1678.58	1676.5	1674.42	1672.34	1670.26	1668.19	1666.11
K1+020		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1689.15	1687.59	1686.03	1684.28	1682.37	1680.44	1678.51	1676.57	1674.64	1672.77	1671	1669.28	1667.52
K1+000		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1683.29	1681.8	1680.32	1678.83	1677.34	1675.85	1674.36	1672.87	1671.38	1669.89	1668.4	1666.92	1665.42
K0+980		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1676.78	1675.42	1674.07	1672.69	1671.39	1670.15	1668.95	1667.77	1666.69	1665.62	1664.59	1663.58	1662.59
K0+960		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1675.91	1674.89	1673.87	1672.87	1671.88	1670.94	1670	1669.16	1668.33	1667.67	1667.15	1666.62	1666.09
K0+941.87	77° 08' 50"	30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1676.93	1675.79	1674.4	1673.05	1671.77	1670.66	1669.75	1669.12	1668.49	1667.9	1667.62	1667.74	1667.52
K0+940		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1675.7	1674.33	1673.05	1671.8	1670.58	1669.4	1668.26	1666.92	1665.65	1664.64	1663.85	1663.18	1662.75

K0+920		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1673.5	1672.47	1671.34	1670.27	1669.2	1668.16	1666.96	1665.71	1664.36	1663.13	1661.93	1660.05	1659.55
K0+900		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1670.79	1669.58	1668.37	1667.27	1666.23	1665.18	1664.14	1663.11	1662.08	1661.01	1659.94	1658.93	1657.92
K0+880		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1668.7	1667.67	1666.5	1665.31	1664.12	1662.79	1661.47	1660.22	1659.06	1657.92	1656.88	1655.84	1654.79
K0+860		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1666.89	1665.62	1664.34	1663.06	1661.77	1660.46	1659.92	1657.92	1656.7	1655.48	1654.28	1653.1	1651.94
K0+840		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1665.09	1663.82	1662.47	1661.17	1659.9	1658.59	1657.28	1655.98	1654.68	1653.39	1652.14	1650.88	1649.46
K0+820		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1662.99	1661.74	1660.49	1659.28	1658.07	1656.89	1655.69	1654.43	1653.13	1651.81	1650.5	1648.89	1647.1
K0+800		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1664.26	1662.85	1661.45	1660.04	1658.64	1657.23	1655.83	1654.43	1653	1651.66	1650.42	1648.9	1647.26
K0+780		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1664.42	1663.02	1661.61	1660.21	1658.81	1657.4	1656	1654.6	1653.19	1651.83	1650.65	1649.24	1647.56
K0+772.20	39° 56' 51"	30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1661.27	1659.86	1658.59	1657.41	1656.32	1655.18	1654	1652.87	1651.77	1650.85	1649.88	1648.35	1646.86
K0+760		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1659.25	1657.85	1656.45	1655.02	1653.57	1652.16	1650.86	1649.6	1648.38	1647.17	1645.95	1644.58	1643.22
K0+740		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1657.18	1655.87	1654.41	1652.99	1651.63	1650.42	1649.02	1647.53	1646.06	1644.58	1643.1	1641.64	1640.21
K0+720		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1654.55	1653.51	1652.21	1651.03	1649.83	1648.34	1646.85	1645.39	1643.96	1642.53	1641.11	1639.7	1638.33
K0+700		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1651.36	1650.71	1650.07	1648.79	1647.44	1646.11	1644.78	1643.41	1642	1640.66	1639.26	1637.78	1636.32
K0+680		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1647.44	1646.86	1646.27	1645.53	1644.64	1643.64	1642.32	1640.96	1639.6	1638.28	1636.91	1635.52	1634.3

K0+660		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1646.47	1645.85	1645.23	1644.62	1644.01	1643.02	1642	1640.62	1639.14	1637.87	1636.54	1635.36	1633.78
K0+656.86	-67° 40' 25 "	30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1650.93	1650.24	1649.04	1647.56	1646.07	1644.59	1643.1	1641.75	1640.13	1638.64	1637.23	1635.75	1634.1
K0+640		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1651.68	1650.54	1649.22	1647.75	1646.27	1644.79	1643.31	1641.83	1640.36	1638.89	1637.41	1635.93	1634.31
K0+620		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1650.68	1649.63	1648.31	1646.98	1645.65	1644.32	1642.93	1641.51	1640.06	1638.63	1637.29	1635.55	1634.31
K0+600		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1647.48	1646.26	1645.04	1643.82	1642.6	1641.37	1640.15	1638.89	1637.59	1636.28	1634.95	1633.62	1632.3
K0+580		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1645.33	1644.1	1642.88	1641.66	1640.44	1639.21	1638	1636.78	1635.56	1634.33	1633.11	1631.58	1630.58
K0+567.32	-64°26' 50"	30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1646.88	1645.59	1644.28	1642.98	1641.67	1640.37	1639.06	1637.76	1636.46	1635.15	1633.85	1632.54	1631.24
K0+560		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1649.8	1648.5	1647.19	1645.89	1644.58	1643.27	1641.96	1640.55	1639.14	1637.73	1636.31	1634.35	1633.37
K0+540		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1651.72	1650.36	1648.93	1647.45	1645.97	1644.49	1643.02	1641.54	1640.06	1638.59	1637.12	1635.65	1634.17
K0+520		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1652.84	1651.18	1649.83	1648.3	1646.79	1645.28	1643.77	1642.27	1640.76	1639.17	1637.75	1636.24	1634.74
K0+500		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1652.57	1650.1	1648.58	1647.11	1645.64	1644.17	1642.7	1641.23	1639.76	1638.29	1636.82	1635.37	1633.91
K0+480		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1652	1649.97	1648.49	1647	1645.51	1644	1642.54	1641.07	1639.58	1638.1	1636.62	1635.13	1633.65
K0+460		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1651.76	1649.74	1648.5	1647.11	1645.63	1644.14	1642.65	1641.16	1639.68	1638.19	1636.71	1635.22	1633.73
K0+440		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1652.9	1650.88	1649.29	1648	1646.59	1645.21	1643.87	1642.52	1641.17	1639.82	1638.47	1637.13	1635.79

K0+420		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1656.72	1654.89	1653.07	1651.25	1649.57	1648.23	1646.89	1645.55	1644.19	1642.92	1641.7	1640.52	1639.42
K0+400		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1660.68	1658.8	1657	1655.23	1653.44	1651.65	1649.93	1648.95	1648	1646.88	1645.82	1644.94	1644.06
K0+380		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1662.16	1660.4	1658.6	1656.68	1654.86	1653.33	1652	1650.69	1649.57	1648.66	1647.76	1646.9	1646.03
K0+371.83	70'57'09"	30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1659.4	1657.91	1656.41	1654.91	1653.38	1651.83	1650.29	1648.86	1647.47	1646.11	1644.8	1643.49	1642.18
K0+360		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1656.71	1655.3	1653.95	1652.49	1651.15	1649.87	1648.51	1647.14	1645.75	1644.31	1642.87	1641.44	1640
K0+340		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1654.39	1653	1651.57	1650.14	1648.72	1647.3	1645.87	1644.48	1643.09	1641.73	1640.44	1639.17	1637.89
K0+320		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1652.5	1651.1	1649.7	1648.31	1646.86	1645.4	1643.93	1642.5	1641.04	1639.6	1638.16	1636.73	1635.29
K0+300		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1651.7	1650.58	1649.33	1647.9	1646.48	1645.02	1643.55	1642.08	1640.61	1639.1	1637.59	1636.15	1634.67
K0+280		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1650.44	1649.06	1647.62	1646.18	1644.75	1643.33	1641.89	1640.49	1639.04	1637.6	1636.18	1634.78	1633.34
K0+260		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1650	1648.52	1647.05	1645.59	1644.11	1642.62	1641.13	1639.67	1638.17	1636.79	1635.22	1633.74	1632.26
K0+240		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1649.1	1647.62	1646.15	1644.68	1643.2	1641.73	1640.26	1638.79	1637.32	1635.86	1634.38	1632.91	1631.44
K0+220		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1649.03	1647.57	1646.06	1644.54	1643.03	1641.52	1640	1638.48	1637	1635.52	1634.05	1632.57	1631.1
K0+212.72	41'04'59"	30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1646.79	1645.22	1643.65	1642.1	1640.57	1639.13	1637.73	1636.41	1635.25	1634.16	1633.19	1632.23	1631.27
K0+200		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1644.06	1642.6	1641.19	1639.8	1638.58	1637.02	1635.64	1634.25	1632.72	1631.11	1629.48	1627.94	1626.81

K0+180		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1643.28	1641.83	1640.36	1638.88	1637.4	1635.92	1634.44	1632.96	1631.48	1630	1628.52	1627.05	1625.13
K0+160		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1642.16	1640.66	1639.16	1637.66	1636.18	1634.74	1633.33	1631.89	1630.43	1628.98	1627.54	1626.1	1624.08
K0+140		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1642.19	1640.7	1639.21	1637.72	1636.24	1634.75	1633.26	1631.77	1630.28	1628.79	1627.31	1625.67	RIO
K0+120		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1642.58	1641.09	1639.6	1638.12	1636.63	1635.14	1633.65	1632.16	1630.67	1629.18	1627.69	1626.21	RIO
K0+100		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1641.77	1640.32	1638.95	1637.55	1636.14	1634.7	1633.42	1631.78	1630.32	1628.86	1627.4	1625.92	RIO
K0+80		30	25	20	15	10	5	EJE	5	10	15	20	25	30
		1641.23	1639.77	1638.31	1636.86	1635.4	1633.94	1632.48	1631	1629.52	1628.03	1626.54	1624.11	RIO
K0+60		30	25	20	15	10	5	0	5	10	15	20	25	30
		1639.02	1637.77	1636.51	1635.25	1634	1632.64	1631.25	1629.83	1628.41	1626.99	1625.59	1624.22	RIO
K0+40		30	25	20	15	10	5	0	5	10	15	20	25	30
		1635.42	1634.39	1633.25	1632.04	1630.78	1629.53	1628.17	1627.01	1625.67	1624	RIO	RIO	RIO
K0+20		30	25	20	15	10	5	0	5	10	15	20	25	30
		1630.77	1630.02	1629.06	1628.04	1627.05	1626.05	1624.94	RIO	RIO	RIO	RIO	RIO	RIO
K0+00		30	25	20	15	10	5	EJE	5	10	15	20	25	30
ABSCISA	DEFLEXIÓN	IZQUIERDA						CENTRAL	DERECHA					

ANEXO 3 TALLERES

Taller Generación de Terreno:

NOMBRE:		CÓDIGO		CURSO:		
NOTA: ES IMPORTANTE QUE CONTESTE TODAS LAS PREGUNTAS CON SU JUSTIFICACIÓN						
Realimentación del Vídeo						
Pregunta	Calificación [0 a 10]	Comentario/Observaciones /Sugerencias				
¿El video fue claro?						
¿Como le pareció la edición del video?						

1 - Copiar la Carpeta Digravi en el equipo donde se realizara el taller.

2 - Dentro de la carpeta Digravi se encuentran algunos archivos necesarios para el desarrollo del taller, revise que la carpeta contenga estos archivos, de ser posible abra los para analizar su contenido:

2.1 - Un archivo en Excel llamado [cartera] , en el se encuentra la cartera que se usara para el desarrollo del taller.

2.2 - Un archivo de Autocad llamada [curvas de nivel] donde se encuentra, las curvas de nivel originales de donde se obtuvo la cartera del ejemplo.

2.3 - Una imagen de las curvas de nivel, el nombre del archivo es [curvas de nivel].

3- Iniciar Digravi, doble click sobre el archivo de nombre Digravi



Pregunta	Calificación [0 a 10]	Comentario/Observaciones /Sugerencias
¿Que opina del logo Digravi?		

¿Que opina de los colores del Digravi ?		
¿Que opina de la imagen de fondo que se muestra en el menú principal del Digravi?		

4-Seleccionar botón de generación de terreno

Generación Del Terreno

en la ventana desplegada inserte el valor del azimuth en el campo Azimut

Azimut:

oprimimos el botón Importar

Importar...

Busque el archivo [cartera] que se encuentra en la carpeta del Taller y selecciónelo.

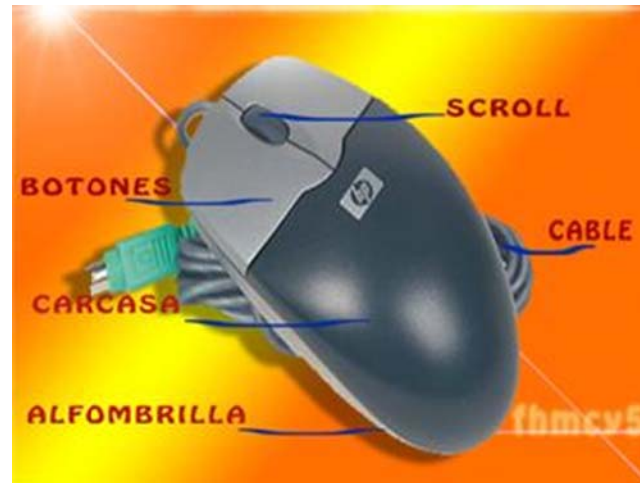
4-1 Para Visualizar y Editar los datos cargados, oprima el botón editar

Modificar...

5- Oprima el botón Aceptar, para visualizar las curvas de nivel generadas a partir de la cartera usada

Aceptar

6-En la ventana que se despliega podrá visualizar las curvas de nivel generadas apartir de la cartera de Excel.



6-1 Cambiar la escala de visualización del Grafica: Para Cambiar la escala del gráfico use el scroll del mouse.

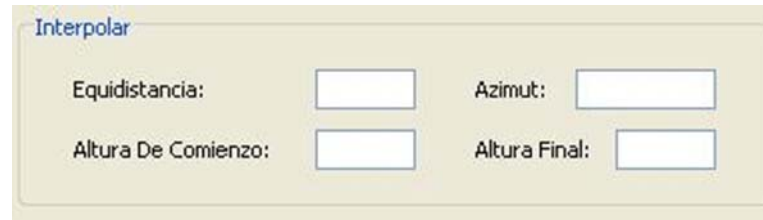
6-2 Trasladarse por el Grafico: Para trasladarse oprime y mantenga presionado el scroll al mismo tiempo mueva el mouse hacia la dirección a lo que se desea trasladarse

6-3 Use la barra de herramienta para ocultar/hacer visible todas las capas que conforman el grafico, si desea trabajar con una capa en particular selecciónela como capa actual.



6-4 Cierra la ventana del gráfico

7-En la ventana de generación de terreno llenemos completamente sección de interpolación



con los valores:

Campo	Valor
Equidistancia	
Altura de Comienzo:	
Altura Final	

Oprima el boton aceptar



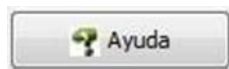
de la ventana de generación de terreno, ahora use valores de su agrado para completar los campos.

Pregunta	Calificacion [0 a 10]	Comentario/Observaciones /Sugerencias
----------	-----------------------	---------------------------------------

¿Observo algún comportamiento anómalo al usar valores diferentes a los mencionados, en caso afirmativo por favor realice una descripción de la anomalía incluyendo los valores usados?	NO APLICA	
--	-----------	--

8- Vamos a realizar algunas modificaciones a la cartera en excel, vamos a eliminar algunas filas y colimnas [desde el explorador de window abramos los archivos "cartera" y "cartera modificada" como puede observar en "cartera modificada" se han eliminado algunas filas, ahora desde Digravi cargue el archivo "cartera modificada", recuerde que primero debe definir un azimut [si ya han abierto una cartera , aqui se puede explicar lo de sobre escribir]]

9-Usemos las ayudas, en Digravi los botones que tiene un icono de interrogante tienen por ejemplo el llamar las ayudas de la aplicación, en estas ayudas se podrán consultar aspectos relacionados con el diseño vial [Apartes el geom] y orientacion sobre el uso de las diferentes herramientas del Digravi, en la ventana de generación de terreno oprima el botón ayuda



10-Usando el editor de carteras:

Realimentacion:

Pregunta	Calificacion [0 a 10]	Comentario/Observaciones /Sugerencias
¿ha considerado intuitivo el módulo de generación de terreno?		

¿La disposición y funcionalidad de los menús es la adecuada?		
¿Es comoda de usar?[sugerencias de como podria se]		
El editor de carteras del Digravi, ¿como le parecio?		
¿Que opina de la forma en que Digravi descenden las ventanas, sobre el tablero de trabajo en cada una de las etapas de diseño vial? ¿Deberian visualizarse de otra forma?		

Taller Especificaciones:

NOMBRE:		CÓDIGO		CURSO:		
NOTA: ES IMPORTANTE QUE CONTESTE TODAS LAS PREGUNTAS CON SU JUSTIFICACIÓN DADO QUE ES LA RAZÓN PRIMORDIAL DE ÉSTA PRÁCTICA.						
El objetivo de ésta práctica es Probar el módulo de especificaciones de diseño así como las ayudas del programa, modificarlas para aportar en el proceso de conocimiento. También permitirá el uso del módulo de alineamiento horizontal para interactuar en forma gráfica con la aplicación en la definición del trazado. Por último Vamos a Probar el diseño de curvas horizontales.						

- 1 - Copiar la Carpeta DIGRAVI 2.0 en el equipo donde se realizara el taller.
- 2 - Dentro de la carpeta Digravi se encuentran algunos archivos necesarios para el desarrollo del taller, revise que la carpeta contenga estos archivos:

2.1 - Un archivo en Excel llamado [cartera] , en el se encuentra la cartera que se usara para el desarrollo del taller.

2.1 - El ejecutable con el programa DIGRAVI 2.0

2.3 - Una carpeta llamada lib

2.4 - Una Carpeta llamada Ayudas.

2.5 - Un archivo de Datos llamado ejemplo.dat

3- Iniciar Digravi, doble click sobre el archivo de nombre Digravi



4 - como en el taller anterior trabajamos la generación de terreno, en este vamos a cargar la cartera rápidamente y seguiremos con el resto del taller.

4.1 - Abrir el módulo de generación de terreno.

4.2 - Posicionarse sobre la cacilla azimuth en la parte inferior del formulario que se acaba de desplegar e ingresar el valor 7”

4.3 – Accionar el botón importar en la parte superior del formulario, se desplegará un selector de archivos del cual debe escogerse el archivo de Excel llamado cartera.xls

4.4 – accionar el botón aceptar, se desplegarán un grafico con las curvas de nivel del terreno.

5 – utilizar las opciones que presenta la ventana de curvas de nivel, ésta actividad busca concientizar al usuario de las herramientas que puede usar con la herramienta gráfica. La gráfica se transada cliqueando y arrastrando con el botón del scroll, zoom rotando el mismo scroll.

5.1 – para que sirve el botón Círculo

5.2 – para que sirve el botón Polylinea

5.3 – para que sirve el botón Segmento

5.4 – para que sirve el botón Selección (1)

5.6 – para que sirve el botón Ocultar Todo, ver todo.

5.7 – para qué sirven los comboBox marcados con NERO, DELGADO Y (0)01

5.8 – Deslice el cursor del mouse sobre las curvas de nivel del terreno, se despliegan unos tags junto al cursor, que indican?

5.9 – para que sirve el botón Ver Capa Actual

Pregunta	Calificacion [0 a 10]	Comentario/Observaciones /Sugerencias
La grafica generada esta completa? Que le faltaría?		
Las herramientas gráficas son suficientes? Cuales faltarían?		
Que otras herramientas seria bueno tener?	no aplica	

6 – Pasemos ahora al módulo de especificaciones, vamos a crear un proyecto con el nombre de cada uno.

6.1. – vamos a hacer una vía Secundaria y a partir de este sencillo dato vamos a apoyarnos en las ayudas para completar todos los valores de especificación, el primer botón de ayuda frente a la velocidad de diseño abrirá las ayudas con la tabla VELOCIDADES DE DISEÑO (KPH). Vamos a escoger la velocidad más baja marcada en la tabla para una vía secundaria con el tipo de terreno que acabamos de cargar el tipo de terreno se muestra en la mitad del formulario.

6.2 – en ancho de banca vamos a hacer lo mismo vamos a picar sobre el botón de ayuda del ancho de banca ANCHOS RECOMENDADOS PARA CALZADA SEGÚN VTR es la tabla que debe desplegarse en este caso y vamos a buscar los valores validos para la velocidad escogida anteriormente con el tipo terreno que cargamos y el tipo de carretera y escogemos el más bajo como en el caso anterior.

6.2 – en el campo de Pendiente Máxima Horizontal vamos la tabla PENDIENTES MÁXIMAS SEGÚN VELOCIDAD ESPECÍFICA TANGENTE VERTICAL VTV vamos a hacer el mismo procedimiento que con las anteriores de acuerdo a los parámetros anteriores escogemos el mínimo valor que presenta la tabla.

6.3 – El campo de pendiente mínima horizontal ya no presenta una tabla si no dos posibles valores de acuerdo al tipo de terreno que tenemos, entonces escogemos la adecuada a nuestro terreno.

6.4 – Así como en el campo anterior nos presenta dos opciones de acuerdo al terreno, escogemos la que nos sirve.

6.5 – El bombeo se define según la tabla TIPO DE SUPERFICIE DE RODADURA en nuestro caso vamos a diseñar una vía en Superficie de concreto hidráulico o asfáltico.

6.6 – Huy la longitud Circular mínima no tiene ninguna indicación, que mal pero no nos vamos a varar por eso, vamos a hacerle un aporte a DIGRAVI en las ayudas la verdad es muy fácil, es como hacer un documento de Word podemos personalizar las ayudas de DIGRAVI a nuestra conveniencia.

6.6.1 – Vamos a abrir Un documento de Word que se encuentra en la carpeta ayudas llamado mis ayudas. Vamos a revisar que este bien donde la sección de longitud Circular mínima ya encontramos la información que necesitamos.

6.6.2 – Al final de la sección roja vamos a agregar un texto que diga Información adicionada por: su nombre.

6.6.3 – Vamos a ir a guardar como, se va desplegar un selector de archivos dentro de este vamos a abrir la carpeta HTML y en la parte inferior se ve un campo llamado tipo en este campo vamos a seleccionar Página web filtrada, se van a mostrar varias páginas web vamos a escoger la VENTANA 4, nos va a preguntar que si la sobrescribimos, le decimos que si a todo.

6.6.4 – Ahora vamos a cerrar el documento de Word y nos vamos de nuevo a la aplicación, tenga en cuenta los valores de especificación que a puesto hasta ahora, vaya a la barra verde de la izquierda se encuentra el botón salvar, vamos a salvar el proyecto con el nombre de cada uno y después serramos la aplicación.

6.6.4 – Ahora vamos a cerrar el documento de Word y nos vamos de nuevo a la aplicación, tenga en cuenta los valores de especificación que a puesto hasta ahora, vaya a la barra verde de la izquierda se encuentra el botón salvar, vamos a salvar el proyecto con el nombre de cada uno y después serramos la aplicación.

6.6.5 – Volvamos a abrir DIGRAVI, en la misma barra oprimimos el botón cargar y cargamos el proyecto, nos dirigimos a especificaciones volvemos a poner los valores que llevábamos despliegue de nuevo las ayudas de longitud Circular mínima y confirme que la modificación ya está, calculamos la longitud y la ingresamos.

6.7 – los siguientes campos los podemos seguir llenando con el apoyo de las ayudas como hemos venido haciendo.

Pregunta	Calificación [0 a 10]	Comentario/Observaciones /Sugerencias
La Información en las ayudas es pertinente? Falta información?		
Es fácil la forma de modificar las ayudas?		
Con texto es suficiente? O sería mejor videos? Otros documentos relacionados como GEOM?		
Siente que las ayudas le ayudaron a afirmar sus conocimientos en el área de topografía y vías?		
Como le pareció el visualizador de las ayudas? Cuál sería la mejor forma de mostrarlas?		
Que sería bueno tener en las ayudas	no aplica	

Taller de Alineamiento Horizontal:

NOMBRE:		CÓDIGO		CURSO:		
NOTA: ES IMPORTANTE QUE CONTESTE TODAS LAS PREGUNTAS CON SU JUSTIFICACIÓN DADO QUE ES LA RAZÓN PRIMORDIAL DE ÉSTA PRÁCTICA.						

El objetivo de éste taller es el de probar el módulo de alineamiento horizontal definir los puntos inicial y final de la vía línea de ceros y trazado de proyecto.

1 - Vamos a dirignos al módulo de Diseño Horizontal, dentro de este encontraremos el módulo de Alineamiento Horizontal, accionamos este módulo y se nos despliega un formulario con tres opciones: puntos iniciales, línea de ceros y trazado de la vía, también vamos a tener un campo llamado zona de afectación.

2 – Primero vamos a dirignos a al campo de zona de afectación y vamos a seleccionarlo, como anteriormente hemos completado el módulo de especificaciones automáticamente se llenara con el valor del ancho de banca, El valor de zona de afectación es el entorno alrededor del eje central de la vía que se verá afectado con la construcción de la misma, por ende debe corroborarse la existencia de información a lo ancho de la zona de afectación en el mismo instante en que se está elaborando el trazado, lo mínimo que se debe asegurar es el valor de la banca y se puede aumentar si el usuario lo desea.

3 – Después de esto vamos a seleccionar la opción de puntos finales, la cual nos despliega un gráfico con las curvas de nivel del terreno, notemos que al mover el cursor por el gráfico el icono puede cambiar al reloj de espera rodeado de un cuadro rojo Cuando esta la parte negra de la pantalla ausente de curvas de nivel, al cliquear en este momento no encuentro ninguna respuesta porque estoy posicionando los puntos sobre un espacio sin información. Cuando el cursor para sobre las curvas puede verse el icono en cruz rodeado de un cuadro blanco cuando se encuentra sobre dos curvas y crus rodeada de un cuadro azul cuando se encuentra sobre la línea de una curva de nivel del terreno, en estos dos últimos casos si se pueden posicionar los puntos con un clic.

4 – Cuando termine de colocar los puntos la aplicación mostrara las coordenadas rectangulares y la altura de los dos puntos, pregunta si están correctos o deben ser definidos nuevamente.

Pregunta	Calificacion [0 a 10]	Comentario/Observaciones /Sugerencias
Fue sencillo posicionar los puntos?		
Fue fácil de encontrar la herramienta zoom?		

Fue fácil de encontrar la herramienta arrastrar la imagen?		
Fue fácil de encontrar la herramienta los comandos control + M control + L?		

5 – La segunda etapa se llama línea de ceros, podemos seleccionarla del formulario anterior o seguirla sobre la misma grafica si no hemos cerrado el lienzo, de la misma manera que la anterior etapa tanto el cursor como la línea que se dibuja cambian al desplazarse el cursor sobre el lienzo la línea roja me indica ausencia de información la verde está bien y puede agregarse a la línea de ceros y la amarilla puede agregarse pero está en una pendiente muy alta o muy baja.

6 – comenzando desde el punto de inicio de la vía trace la línea de ceros uniando con clic las curvas de nivel del terreno hasta llegar al punto final, cuando se posiciones sobre el punto final la línea se pondrá de color amarillo y el punto se agrandará un poco, cuando cliquee el punto final la aplicación preguntará si desea guardar la línea de ceros o hacerla nuevamente, si estamos conformes con esa línea vamos a guardarla.

Pregunta	Calificacion [0 a 10]	Comentario/Observaciones /Sugerencias
Fue sencillo posicionar los puntos?		
Fue fácil de encontrar la herramienta zoom?		
Fue fácil de encontrar la herramienta arrastrar la imagen?		
Fue fácil de encontrar la herramienta los comandos control + M control + L?		

5 – La segunda etapa se llama línea de ceros, podemos seleccionarla del formulario anterior o seguirla sobre la misma grafica si no hemos cerrado el lienzo, de la misma manera que la anterior etapa tanto el cursor como la línea que se dibuja cambian al desplazarse el cursor sobre el lienzo la línea roja me indica ausencia de información la verde está bien y puede agregarse a la línea de ceros y la amarilla puede agregarse pero está en una pendiente muy alta o muy baja.

6 – comenzando desde el punto de inicio de la vía trace la línea de ceros uniendo con clic las curvas de nivel del terreno hasta llegar al punto final, cuando se posiciones sobre el punto final la línea se pondrá de color amarillo y el punto se agrandará un poco, cuando cliquee el punto final la aplicación preguntará si desea guardar la línea de ceros o hacerla nuevamente, si estamos conformes con esa línea vamos a guardarla.

Pregunta	Calificacion [0 a 10]	Comentario/Observaciones /Sugerencias
Fue sencillo definir la línea de ceros? Como podria ser mejor?		

7 - La tercera etapa es el trazado de proyecto, este se traza sobre las curvas de nivel del terreo tratando de conservar la misma trayectória de la línea de ceros, sus colores cambian de manera similar a los de la línea de ceros y las secciones transversales que lo atraviezan corrovoran que haya información a lo ancho del alineamiento, tan pronto como el usuario termine de unir el punto inicial con el final la aplicación pedirá guardar los cambios o hacer nuevamente.

Pregunta	Calificacion [0 a 10]	Comentario/Observaciones /Sugerencias
Fue sencillo definir El trazado de proyecto? Como podria ser mejor?		

Taller Diseño de Curvas Horizontales:

Taller de Transición de peralte:

ANEXO 4 PROPUESTA INTERFAZ GRÁFICA

Logo:



Menú principal:

DIGRAVI v2.0

ESPECIFICACIONES

Nombre Del Proyecto: Proyecto Sin Título 1	Velocidad De Diseño [km/h]:	Peralte Máximo e.max [%]:	Longitud Vertical Mínima [m]:	☒ Aceptar
Tipo De vía: PRINCIPAL DE DOS CALZADAS	Ancho De Banca [m]:	Bombeo [%]:	Abscisado [m]:	☒ Modificar
	Pendiente Máxima Horizontal [%]:	Longitud Circular Mínima [m]:	Sobreebanco [m]:	
	Pendiente Mínima Horizontal [%]:	Longitud Espiral Mínima [m]:	Azimuth:	

Punto De Referencia Norte [M... Este [M... Altura [M...]

Comandos:

Manejador de Capas: NEGRO DELGADO (0)01

Escribir los comentarios de cada uno de los items. por ejemplo: cuando esta parado con el cursor en Nombre de proyecto: el mensaje seria. "debes ingresar el nombre del proyecto"

Posicion Mouse y Scale
Mouse Fuera del Lienzo

Generación de terreno:

DIGRAVI v2.0

ESPECIFICACIONES GENERACION DE TERRENO.

Carta De Campo

Puntos Laterales: Espacio:

Interpolar

Equidistancia:

Altura De Comienzo:

Altura Final:

Comandos

Manejador de Capas

NEGRO DELGADO (0)01

Escribir los comentarios de cada uno de los items. por ejemplo: cuando esta parado con el cursor en Nombre de proyecto: el mensaje seria. "debes ingresar el nombre del proyecto"

Posicion Mouse y Scale
Mouse Fuera del Lienzo

Diseño horizontal:

DIGRAVI v2.0

ESPECIFICACIONES GENERACION DE TERRENO. DISEÑO HORIZONTAL

4.Zona De Afectación:

ALINEAMIENTO HORIZONTAL

PUNTO DE REFERENCIA LINEA DE CEROS LINEA DE PROYECTO

Curva No: Radio: Estructura: Pendiente: Velocidad:

Sección: Deflexión: Ángulo: Escala:

TEST DE SELECCION

TIPOS DE CURVAS

Comandos

Manejador de Capas

NEGRO DELGADO (0)01

Escribir los comentarios de cada uno de los items. por ejemplo: cuando esta parado con el cursor en Nombre de proyecto: el mensaje seria. "debes ingresar el nombre del proyecto"

Posicion Mouse y Scale
Mouse Fuera del Lienzo