

Prototipo Software para Apoyar La Planificación Prospectiva de Proyectos, Basado en el Método
de Estrategia de los Actores

Johan Mauricio Cepeda Ortiz

Trabajo de grado para optar al título de Ingeniero de Sistemas

Director:

Jaime Octavio Albarracín Ferreira

Magíster en ingeniería de Sistemas.

Codirector:

Diego Armando Villarreal

Ph.D Computer Science.

Universidad Industrial de Santander

Facultad de Ingenierías Físico-Mecánicas

Escuela de Ingeniería de Sistemas e Informática

Bucaramanga

2019

Contenido

	Pág.
Introducción	13
1. Descripción del Proyecto	15
1.1 Definición del Problema	15
1.2 Justificación del Proyecto	16
1.3 Objetivos	17
1.3.1 Objetivo General.....	17
1.3.2 Objetivos específicos	17
2. Marco Teórico y Metodológico	18
2.1 Marco Teórico.....	18
2.1.1 Planificación prospectiva.	18
2.1.2 Métodos de la planificación prospectiva.....	21
2.1.3 Soluciones existentes.	26
2.1.4 Estándar de la IEEE 830	28
2.1.5 Lenguaje unificado de modelado (UML)	29
2.1.6 Arquitectura MVC (Modelo-Vista-Controlador).....	31
2.1.7 Herramientas utilizadas para el desarrollo	35
2.2 Marco Metodológico.....	43

2.2.1 Prototipado evolutivo.....	43
2.2.2 Fases del proyecto.....	45
2.2.2.1 Recolección y refinamiento de requisitos.	45
2.2.2.2 Diseño rápido.	46
2.2.2.3 Construcción de la herramienta.	46
2.2.2.4 Evaluación de la herramienta por parte del cliente.	47
2.2.2.5 Refinamiento de la herramienta.	47
2.2.2.6 Producto de ingeniería.	48
3. Resultados del Proyecto.....	48
3.1 Requisitos Establecidos para la Herramienta.....	49
3.1.1 Requisitos para el desarrollo.....	49
3.2 Diseño de la Herramienta.....	52
3.2.1 Diagrama de casos de uso.....	52
3.2.2 Diagrama de estados.	53
3.2.3 Diagrama de actividades.....	54
3.3 Desarrollo de la Herramienta.....	55
3.3.1 Módulo de ingreso.	59
3.3.2 Módulo de gestión de proyectos.	61
3.3.3 Módulo de datos.....	62
3.3.4 Módulo de gestión de usuarios.	62
3.3.5 Módulo generar Matriz de Influencia Directa (MID).	63
3.3.6 Módulo de ayuda.....	65

3.3.7 Módulo para editar informe.	66
3.3.8 Roles de la herramienta.....	67
3.4 Desarrollo e Implementación.....	69
3.5 Arquitectura del Sistema.....	70
3.6 Pruebas de la Herramienta	71
4. Conclusiones.....	73
5. Recomendaciones	74
Referencias Bibliográficas	76

Lista de Figuras

	Pág.
Figura 1. Fases de planeación estratégica prospectiva.....	21
Figura 2. Identificación de variables esenciales	23
Figura 3. Análisis de estrategia de los actores	25
Figura 4. Estándar de la IEEE 830.....	28
Figura 5. Lenguaje unificado de modelado (UML).....	29
Figura 6. Arquitectura MVC (Modelo-Vista-Controlador).....	31
Figura 7. Net.	35
Figura 8. C Sharp.....	38
Figura 9. SQL Server.....	40
Figura 10. Bootstrap 3.0.....	42
Figura 11. Ciclo de vida de la metodología prototipado evolutivo.....	44
Figura 11. Diagrama de casos de uso – Crear proyecto.....	53
Figura 12. Diagrama de estados – Crear proyecto.....	54
Figura 13. Diagrama de actividades – Crear proyecto.....	55
Figura 14. Inicio de la herramienta web PJA.....	56
Figura 15. Descripción del método juego de los actores de la herramienta web PJA.....	57
Figura 16. Sección de contacto de la herramienta web PJA.....	58
Figura 17. Interfaz de ingreso de la herramienta web PJA.....	58
Figura 18. Interfaz de registro de la herramienta web PJA.....	60

Figura 19. Interfaz de ingreso de la herramienta web PJA..... 60

Figura 20. Módulo Gestión de proyectos de la herramienta web PJA..... 61

Figura 21. Módulo Gestión de usuarios de la herramienta web PJA..... 63

Figura 22. Módulo Matriz (MID) de la herramienta web PJA 65

Figura 23. Módulo de ayuda de la herramienta web PJA..... 66

Figura 24. Módulo Editar un informe de la herramienta web PJA..... 67

Figura 25. Menú del rol Administrador de la herramienta web PJA..... 68

Figura 26. Menú del rol Usuario de la herramienta web PJA..... 69

Figura 27. Arquitectura implementada para Planeación Juego de Actores. 71

Lista de Tablas

	Pág.
Tabla 1. Objetivos específicos	17
Tabla 2. Definición de abreviaturas utilizadas.....	49
Tabla 3. Requisito funcional Crear proyecto para la herramienta PJA.....	50
Tabla 4. Requisitos propuestos para la herramienta PJA.....	50
Tabla 5. Formulario de caso de prueba del requisito 04 de la herramienta.	72
Tabla 6. Formulario dos de caso de prueba del requisito 04 de la herramienta web	72
Tabla 7. Formulario tres de caso de prueba del requisito 04 de la herramienta web	73

Lista de Apéndices

“Ver Apéndices Adjuntos en el CD y pueden visualizarse en la Base de Datos de la
Biblioteca – UIS”

Apéndice A. Documento de especificación de requerimientos de la herramienta web Planeación
Juego de Actores (PJA)

Apéndice B. Documento de diseño de la herramienta web Planeación Juego de Actores (PJA)

Apéndice C. Documento de casos de pruebas de la herramienta web Planeación Juego de Actores
(PJA)

Apéndice D. Interfaces diseñadas de la herramienta web Planeación Juego de Actores (PJA)

Resumen

Título: Prototipo Software para Apoyar La Planificación Prospectiva de Proyectos, Basado en el Método de Estrategia de los Actores.*

Autor: Johan Mauricio Cepeda Ortiz**

Palabras Claves: Metodología de Prospección, Mactor, Juego de los Actores, MIC MAC, Análisis Estructural, Prototipado Evolutivo, Matriz de Influencia Directa, Matriz Actores por Objetivos.

Descripción:

Gracias a la experiencia que tiene el grupo de investigación en Gestión de la Innovación Tecnológica y del Conocimiento (INNOTEC), con las herramientas que maneja la metodología de planeación prospectiva, se logró identificar la oportunidad de crear una herramienta que beneficie tanto a las organizaciones, como a los diferentes actores que participan en un proyecto, ya que se busca anticipar el futuro y reducir el margen de fracaso con el uso de esta metodología.

En respuesta a esto, se decidió crear una herramienta que utilice la metodología estrategia planeación prospectiva, y para lograr tal propósito se utilizó la metodología prototipado evolutivo y el uso del estándar IEEE 830 para la especificación de requisitos, delimitando de esta manera el alcance del sistema.

La presente herramienta da la posibilidad de gestionar toda la información que se realice en la etapa de planificación prospectiva, y por medio de diagramas, matrices y cuadros, se brinda la posibilidad de realizar un estudio sobre los comportamientos futuros de los actores que participan en un proyecto, siempre con el objetivo de anticipar los errores que se podrán presentar en la ejecución de un proyecto.

* Trabajo de Grado

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas e Informática Director: Jaime Octavio Albarracín Ferreira, Magíster en ingeniería de Sistemas.

Abstract

Title: Software Prototype to Support Prospective Project Planning, Based on the Actor Strategy Method.*

Author: Johan Mauricio Cepeda Ortiz**

Keywords: Prospecting Methodology, Mactor, Actor Game, MIC MAC, Structural Analysis, Evolutionary Prototyping, Direct Influence Matrix, Actors by Objectives Matrix.

Description:

Thanks to the experience of the research group in Technological Innovation and Knowledge Management (INNOTECH), with the tools that the prospective planning methodology manages, it was possible to identify the opportunity to create a tool that benefits both organizations, as to the different actors that participate in a project, since it seeks to anticipate the future and reduce the margin of failure with the use of this methodology.

In response to this, it was decided to create a tool that uses the prospective planning strategy methodology, and to achieve this purpose the evolutionary prototyping methodology was used and the use of the IEEE 830 standard for the specification of requirements, thus delimiting the scope of the system .

The present tool gives the possibility of managing all the information that is carried out in the prospective planning stage, and through diagrams, matrices and tables, it is possible to carry out a study on the future behaviors of the actors that participate in a project, always with the objective of anticipating the errors that may arise in the execution of a project.

* Bachelor Thesis.

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas e Informática Director: Jaime Octavio Albarracín Ferreira, Magíster en ingeniería de Sistemas.

Introducción

Para las organizaciones es importante que antes de iniciar un proyecto, ya se pueda contar con la información detallada y estructurada del tema sobre el que se tomarán decisiones, así como los objetivos y actores que dan paso a los posibles escenarios que se van a presentar a futuro con relación a la situación que se estudia.

Existe una metodología que trata toda esta problemática, y se llama estrategia de planeación prospectiva. Esta metodología trata de anticipar el futuro, de que todas las acciones que se vayan a realizar, no se dejen al azar, sino que cada paso que se vaya a dar, ya haya sido previamente estudiado. Al trabajar con herramientas que se enfoquen en esta metodología, lo que se busca es poder reducir al más mínimo el porcentaje de fracaso en un proyecto a futuro.

Es así como desde la experiencia que tiene el grupo de investigación en gestión de la innovación tecnológica y del conocimiento (INNOTEC), de la UIS, con respecto a la apropiación de las herramientas que se manejan en la metodología de planeación prospectiva, se ha identificado la necesidad de crear una herramienta que beneficie tanto a las organizaciones como a los diferentes actores que participan en un proyecto.

En respuesta a esta necesidad, se ha propuesto la creación de una herramienta que soporte las acciones relacionadas con la implementación de la metodología planeación prospectiva. Sin embargo, esta metodología cuenta con diferentes métodos, y para efectos de este trabajo se hizo énfasis en el método denominado juego de los actores, ya que brinda información importante acerca de las metas, alianzas, batallas, ventajas y desventajas de todos los actores que participan en un proyecto.

Por otro lado, actualmente existe una herramienta en el mercado que maneja la metodología planeación prospectiva, y se enfoca en el método estrategia de los actores. Este software se llama Método de análisis de juego de los actores (Lipsor MACTOR), y gracias a que el grupo de investigación INNOTECH tiene experiencia en el trabajo con ella, se logró obtener información acerca de la herramienta. Posteriormente, con el tiempo se llegó a realizar un estudio riguroso tanto en la literatura del método, como también en el uso de la herramienta software, esto con el fin de poder entender sus ventajas y falencias.

Adicionalmente, se buscó la forma en que esta nueva herramienta software, llegara a tener cambios significativos con respecto a la anterior. Para ello, se decidió que fuera de aplicación web, ya que se identificó, que hay falencias en cuanto a la recolección y generación de información de interés.

Para ello se concluyó que la mejor opción, era hacerla con un diseño intuitivo para el usuario, posibilitando así que los participantes o usuarios de un proyecto logran interactuar y gestionar sus datos e información directamente en la herramienta.

Posteriormente, se creó un módulo de ayuda, el cual va a estar acompañando tanto al administrador, como al usuario (actor) en todo momento del estudio de su proyecto. Esto se hace con la intención de que la persona que gestione su estudio en la herramienta, logre comprender la metodología prospectiva y más concretamente el método de la estrategia de actores. Además, se quiere que el usuario pueda tener claro los pasos que debe realizar y como debe realizarlos, para que el estudio tenga un resultado satisfactorio.

1. Descripción del Proyecto

1.1 Definición del Problema

Actualmente existe una herramienta software de tipo escritorio que utiliza la metodología de planeación prospectiva. Esta es una metodología que cuenta con diversos métodos. Específicamente, la herramienta anteriormente nombrada está basada en el método llamado estrategia de los actores, tiene como nombre Lipsor MACTOR, y es utilizada por el grupo de investigación INNOTECH.

Para este grupo de investigación, es indispensable que esta herramienta le brinde ciertas ventajas al momento de enfrentar un proyecto, con el fin de poder adelantarse a los posibles escenarios a futuro. Adicionalmente, se busca que el actor encargado en la administración de la herramienta, pueda entender el método, para que así se pueda facilitar el ingreso información, y comprender los resultados obtenidos.

En base al trabajo que se ha venido realizando con herramientas que manejan esta metodología, se han podido identificar una serie de inconvenientes para quienes la usan:

- Al momento de superar la cantidad máxima (trece actores), el software tiende a presentar resultados erróneos. Esto se debe a la gran cantidad de información que se debe manejar y estudiar.
- El actor encargado de la herramienta, no logra entender el módulo de resultados que se generan al momento de concluir un estudio.

- Además, se observó que el actor que va crear un proyecto, está en la obligación de gestionar, generar e ingresar la información de los diferentes actores que van a participar en un proyecto.

Por consiguiente, se hace necesario la creación de una nueva herramienta software que maneje la misma metodología, pero que tenga como fin poder superar las dificultades que se lograron identificar, de tal forma que se contribuya con la gestión de datos que deben hacer los actores que participan en un proyecto (nuevas posibilidades y ventajas al momento de ingresar la información).

1.2 Justificación del Proyecto

Tomando como referente el trabajo del grupo de investigación INNOTEC, se ha evidenciado un déficit en las herramientas que actualmente se utilizan para la etapa de planificación prospectiva. Por ende, se ha planeado desarrollar una herramienta que supla esas necesidades.

Cabe recordar que la metodología de planeación prospectiva es una disciplina que pronostica acciones futuras. Estudia el futuro para comprenderlo y poder influir en él. Además, trata de imaginar o proyectar los futuros posibles (Pinto, 2008).

En ese sentido, se debe tener en cuenta que con el uso de los métodos de planificación prospectiva (análisis estructural y estrategia de los actores), es posible llevar a cabo un análisis exhaustivo sobre el futuro de un proyecto mediante la ejecución de tres etapas importantes como lo son: la construcción de las bases, identificación de cuestiones principales en el juego y la construcción de escenarios (Pinto, 2008).

Con respecto a lo anterior, se identifica como alternativa de mitigación, el desarrollo de una herramienta software que soporte las acciones que se ejecutan haciendo uso de esta metodología, la cual, a través de diagramas, matrices y cuadros ilustrativos de la información, proporcione a los usuarios la facilidad del análisis sobre los comportamientos futuros de un proyecto.

Adicionalmente, hay que tener en cuenta que el objetivo principal de estrategia de los actores, es facilitarle al actor la ayuda para la toma de decisión sobre sus futuras alianzas, conflictos, metas, fortalezas y debilidades en un entorno de trabajo (Pinto, 2008).

Gracias a este método, podemos crear los escenarios que se van a presentar en un futuro, con el fin de poder realizar cambios en el presente para que no afecte la realización total de los objetivos de los actores y del proyecto.

1.3 Objetivos

1.3.1 Objetivo General. Desarrollar una herramienta software para apoyar la planificación prospectiva de proyectos, basado en el método de estrategia de los actores, haciendo uso de la tecnología .NET

1.3.2 Objetivos específicos

Tabla 1.

Objetivos específicos

No	Objetivo	Cumplimiento
1	Identificar los requisitos funcionales de la herramienta siguiendo la norma IEEE 830.	Se identificaron los requisitos funcionales de la herramienta, siguiendo la norma IEEE 830. (Ver anexo A).

2	Diseñar, a través del lenguaje unificado de modelado, los requisitos funcionales con prioridad alta.	Se diseñó, a través del lenguaje unificado de modelado, los requisitos funcionales con prioridad alta. (Ver anexo B).
3	Implementar en C# .NET y SQL SERVER los requisitos funcionales diseñados para la primera versión de la herramienta.	Se Implementó en C# .NET y SQL SERVER los requisitos funcionales diseñados para la primera versión de la herramienta. (Ver anexo A sección 3.2).
4	Realizar pruebas funcionales de la herramienta.	Se realizaron pruebas funcionales de la herramienta. (Ver anexo C).

2. Marco Teórico y Metodológico

2.1 Marco Teórico

En esta sección el lector podrá encontrar información relacionada con los principales temas que brindan soporte al desarrollo de este proyecto. Entre estos se encuentran: Planificación prospectiva, métodos de planificación prospectiva, soluciones existentes, estándar de la IEEE 830, lenguaje unificado de modelado, modelo vista-controlador, y herramientas utilizadas para el desarrollo.

2.1.1 Planificación prospectiva. La prospectiva es la disciplina que indica las acciones futuras desde un punto de vista ambiental, social, económico, científico y tecnológico (Pinto, 2008). Uno de los fundadores de la prospectiva es Gastón Berger. Él la define como la ciencia que estudia el futuro para comprenderlo y poder influir en él (Pinto, 2008).

Esta trata de imaginar o proyectar los escenarios futuros posibles, con el fin de planificar las acciones necesarias, para evitar acelerar los sucesos.

Algunas de las características del enfoque moderno de la prospectiva son (Forciniti & Elbaum, 2001):

- Está dirigida a la acción y a la definición de prioridades con un enfoque preventivo y de anticipación de los problemas.
- Parte del supuesto de que no hay uno, sino varios futuros posibles.
- Adopta una visión global y sistémica, dado que entiende los fenómenos sociales en su complejidad e interdependencia.
- Toma en cuenta los factores cualitativos como el análisis sobre el comportamiento de los actores.

En conclusión, se debe entender como un sistema que, en su recorrido más importante, viene desde el futuro hacia el presente. Esto nos permite configurar el futuro que deseamos, reflexionando sobre el presente, para finalmente desarrollar las estrategias de acción que se van a utilizar para poder alcanzar el objetivo (Forciniti & Elbaum, 2001).

Para hacer el análisis prospectivo hay que pasar por diferentes métodos como es, el análisis estructural y estrategia de los actores.

Gracias a la tecnología, se han desarrollado diversas herramientas para estos métodos que sistematizan toda la información que se genera en esta etapa. Esta información se almacena para ser estudiada y analizada de tal forma que se pueda llegar a diferentes conclusiones, previendo así la corrección de errores que se podrían presentar en el futuro (Pinto, 2008).

El proceso de la planificación prospectiva estratégica comprende tres etapas principales: construcción de las bases, identificación de cuestiones principales en juego, y construcción de escenarios (Pinto, 2008).

Etapas de planificación prospectiva

La planificación prospectiva, cuenta con diferentes etapas que son importantes para su desarrollo.

Construcción de las bases e identificación de variables:

En esta etapa, el método consiste en establecer y analizar el sistema objeto de estudio. El propósito de esto, es identificar los puntos y aspectos fundamentales, que representan apuestas para el futuro, sobre las cuáles los actores podrían elaborar sus estrategias. El análisis estructural es la herramienta principal en esta etapa del proceso (Arcade, Godet, Meunier, & Roubelat, 2004).

Identificación de las cuestiones principales en el juego, y preguntas claves para el futuro

En esta etapa, el punto es identificar los verdaderos mecanismos que regulan la existencia y evolución de ciertas variables. Si se toma en cuenta la estrategia de los actores, se puede entender mejor las evoluciones observadas, y ampliar el tamaño de futuras evoluciones. Es posible que ciertas cuestiones en juego parezcan estar generando potencialmente alianzas-conflictos que serán determinados en el futuro. Por lo tanto, es importante formular preguntas claves, que se relacionen con la evolución a largo plazo del sistema estudiado.

Elaboración de escenarios exploratorios

Un escenario se construye a partir de una serie de hipótesis posibles sobre cada una de las preguntas claves. El objetivo desde este proyecto, frente al tema de escenarios, es explorar y luego reducir el espacio de los mismos teniendo en cuenta las exclusiones que resultan, por ejemplo, de

posibles incompatibilidades entre ciertas hipótesis. Luego se hace necesario analizar el grado de coherencia de los escenarios preseleccionados (Arcade, Godet, Meunier, & Roubelat, 2004).



Figura 1. Fases de planeación estratégica prospectiva Adaptado de: Fases de planeación estratégica prospectiva. Revista del Postgrado en Informática “http://www.revistasbolivianas.org.bo/scielo.php?pid=S3333-77772014000100013&script=sci_arttext”

2.1.2 Métodos de la planificación prospectiva. Los métodos esenciales para realizar el análisis prospectivo son: análisis estructural y estrategia de actores (Ossorio, 2003).

Análisis estructural

Es una herramienta diseñada para vincular ideas que permiten describir el sistema gracias a una matriz que une todos sus componentes. Mediante el análisis de estas relaciones el método permite destacar las variables que son esenciales para la evolución del sistema. Tiene la ventaja de incitar a la reflexión dentro del grupo, y hacer que las personas analicen ciertos aspectos que

algunas veces son poco imaginativos. Se aplica al estudio cualitativo de sistemas extremadamente diferentes (Ossorio, 2003).

El sistema estudiado se presenta como un conjunto de elementos interrelacionados (variables/factores). El análisis estructural comprende tres etapas que son, inventario de variables y factores, descripción de las relaciones entre variables e identificación de variables esenciales (Ossorio, 2003).

Inventario de variables/factores

Consiste en definir el alcance del estudio, es decir, del sistema a ser estudiado. La segunda etapa hace referencia a la consolidación del inventario de todas las variables y factores internos o externos, que caracterizan al sistema (Arcade, Godet, Meunier, & Roubelat, 2004).

Descripción de las relaciones entre variables

El método consiste en vincular las variables en una tabla de doble entrada (Arcade, Godet, Meunier, & Roubelat, 2004).

Identificación de variables esenciales

Consiste en definir las variables que son primordiales en un proyecto como son las influyentes, dependientes y esenciales en la evolución del sistema (Arcade, Godet, Meunier, & Roubelat, 2004).

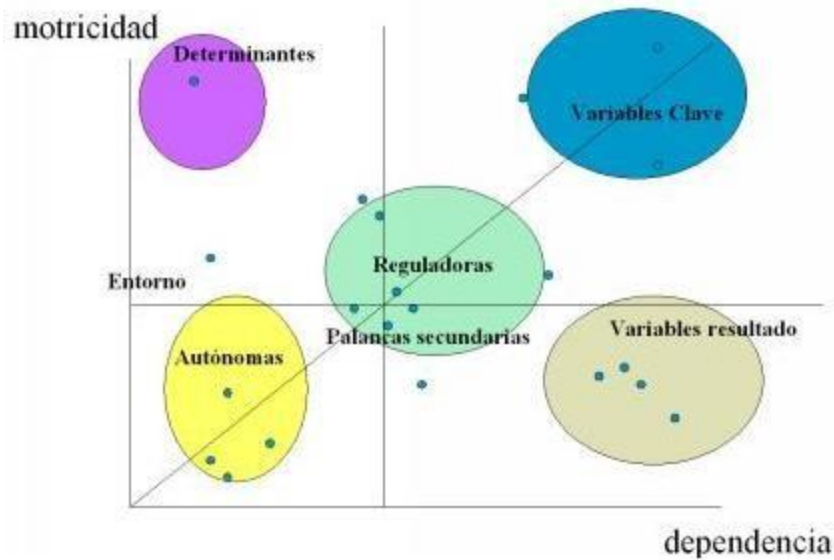


Figura 2. Identificación de variables esenciales. Adaptado de: Fuente: Identificación de variables esenciales. “<http://www.prospectiva.eu/blog/593>”

Estrategia de los actores

Esta etapa es esencial hasta tanto no se realice un estudio completo retrospectivo, de lo contrario se hace imposible llevar a cabo un pensamiento prospectivo adecuado (Ossorio, 2003). Esto significa que todas las variables y cuestiones claves identificadas con anterioridad, que se relacionan con la elaboración de una base de datos, debería ser lo más extensa posible.

Este método, permite identificar los eventos principales que señalan el camino del futuro, para tener una mejor visión general de la interacción de eventos y una mejor comprensión de las relaciones entre los actores. También gracias a este método se pueden identificar aquellos actores que ejercen una influencia y controlan de una u otra manera las citadas variables (Ossorio, 2003).

Las ventajas que ofrece, está orientada a permitir tener mejor conocimiento sobre el posicionamiento de los diferentes actores involucrados en los procesos de toma de decisiones, así

como a generar confianza entre los actores, valorar las relaciones de fuerza entre ellos y estudiar sus convergencias-divergencias (Ossorio, 2003).

La estrategia de los actores consiste en una serie de fases tales como: selección y definición del proyecto, identificación de actores, análisis estratégico de actores, planteamiento de estrategias basadas en el análisis político de los actores, identificación de oportunidades-amenazas, propuesta de estrategias relacionadas con el entorno y selección final de grupo de estrategias (Ossorio, 2003).

Selección y definición del proyecto

Se trata de identificar los objetivos priorizados, así como sus estrategias y mecanismos. Los proyectos no pueden ser declaraciones poco precisas o generales. Se requiere una definición clara de los mecanismos, identificando objetivos priorizados, y mecanismos diseñados para el logro de los objetivos planteados (Arcade, Godet, Meunier, & Roubelat, 2004).

Identificación de los actores

Hace referencia a las acciones de identificación de todos los actores involucrados en el desarrollo del proyecto, su posición con relación a cada uno de los objetivos y su poder o capacidad de acción (Arcade, Godet, Meunier, & Roubelat, 2004).

Análisis estratégico de actores

Está orientado a proponer supuestos sobre la posibilidad de que cualquier proyecto implica cambios que afectan las reglas bajo las cuales los actores han estado trabajando, llegando a generar transformaciones relacionadas con los incentivos y modificando así la distribución de flujos futuros de costos y beneficios entre distintos grupos (Arcade, Godet, Meunier, & Roubelat, 2004).

Las estrategias basadas en el análisis político de los actores

Esta fase tiene como finalidad diseñar estrategias capaces de incidir sobre el poder y la posición de los actores. Se determina la oportunidad o amenaza que cada uno de los actores representa desde el punto de vista de su lugar en la jerarquía (Arcade, Godet, Meunier, & Roubelat, 2004).

Oportunidades, amenazas y estrategias del entorno

El objetivo es definir estrategias para considerar las oportunidades y las amenazas que pueden afectar lo realizado (tanto en el corto, como mediano y largo plazo) (Arcade, Godet, Meunier, & Roubelat, 2004).

Selección final del grupo de estrategias

A partir del análisis anterior, surge un grupo de estrategias con sus objetivos y acciones (Arcade, Godet, Meunier, & Roubelat, 2004).

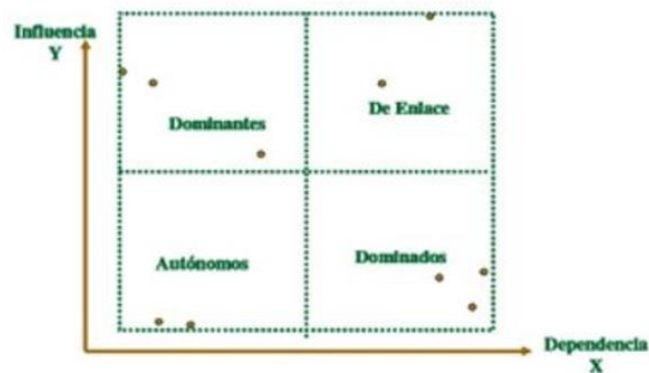


Figura 3. Análisis de estrategia de los actores. Adaptado de: Análisis de estrategia de los actores.

“<https://prospecciondelme rcadoblog.wordpress.com/2016/03/02/acerca-de-este-blog/>”

2.1.3 Soluciones existentes. Actualmente en el mercado hay cuatro herramientas que trabajan en el análisis prospectivo con los métodos de análisis estructural, estrategia de los actores y elaboración de escenarios. Las herramientas más usadas en el mercado actualmente son (Godet, 2007):

Micmac

El análisis estructural es una herramienta de estructuración de una reflexión colectiva. Ofrece la posibilidad de describir un sistema con ayuda de una matriz que relaciona todos sus elementos constitutivos. Este método tiene por objetivo, hacer aparecer la revisión variable influyente y dependientes, y por ello las variables esenciales a la evolución del sistema.

La herramienta Micmac está muy bien estructurada, su velocidad de respuesta es muy buena, pero tiene ciertas deficiencias en su manejo, ya que si el usuario no se documenta no llegara a comprenderla (Godet, 2007).

Mactor

El método de análisis de juego de actores, Mactor, busca valorar las relaciones de fuerza entre los actores, y estudiar sus convergencias y divergencias con respecto a un cierto número de posturas y de objetivos asociados.

A partir de este análisis, el objetivo de la utilización del método Mactor es el de facilitar a un actor una ayuda para la decisión de la puesta en marcha de su política de alianzas y de conflictos.

Mactor es una herramienta concreta y muy bien estructurada. Es fácil de manejar y comprender. La deficiencia de esta herramienta está en la capacidad de almacenar información, ya

que tiene un número límite de actores, y esto podría afectar a proyectos que sobrepasan ese límite y requieren de los resultados que ofrece la herramienta (Godet, 2007).

Smic-prob expert

Los métodos de impactos cruzados probabilistas vienen a determinar las probabilidades simples y condicionadas de hipótesis o eventos, así como las probabilidades de combinaciones de estos últimos, teniendo en cuenta las interacciones entre los eventos y/o hipótesis.

El objetivo de estos métodos no es solamente el de hacer destacar los escenarios más probables, sino también el de examinar las combinaciones de hipótesis que serán excluidas a priori (Godet, 2007).

Multipol

Como todo método multicriterio, el método Multipol pretende comparar diferentes acciones o soluciones a un problema en función de criterios y de políticas múltiples. El objetivo del Multipol es también aportar ayuda a la decisión construyendo un tablero de análisis simple y evolutivo de las diferentes acciones o soluciones que se le ofrecen al que debe tomar la decisión (Godet, 2007).

2.1.4 Estándar de la IEEE 830



Figura 4. Estándar de la IEEE 830. Fuente: Estándar de la IEEE 830. Documento de Requerimientos de Software estándar IEEE 830-1998 “<https://ingenieriadesoftwareutmachala.wordpress.com/2017/01/20/documento-de-requerimientos-de-software-estandar-ieee-830-1998/>”

El estándar presenta recomendaciones para enfocar la especificación de requisitos software (ERS). En la IEEE se indica que un buen documento de requisitos debe contemplar toda la información presentada en este estándar y, aunque propone una organización de dicha información, no exige estrictamente el formato que debe seguirse. Los principales objetivos de una ERS son (IEEE, 2008):

- Ayudar a los clientes a describir claramente lo que se desea obtener mediante un determinado software.
- Ayudar a los desarrolladores a entender que quiere exactamente el cliente.
- Servir de base para desarrollo de estándares de ERS particulares para cada organización.

Las características deseables para una buena ERS según la IEEE son las siguientes (IEEE Std 830-1998) (IEEE, 2008):

- Correcta
- No ambigua
- Completa

- Verificable
- Consistente
- Clasificada
- Modificable
- Explorable
- Utilizable durante las tareas de mantenimiento y uso.

2.1.5 Lenguaje unificado de modelado (UML)



Figura 5. Lenguaje unificado de modelado (UML). Fuente: Lenguaje unificado de modelado (UML)”[http://www .uml.org/what-is-uml.htm](http://www.uml.org/what-is-uml.htm)”

UML es un lenguaje gráfico que ayuda a especificar, visualizar y documentar esquemas de sistemas de software, incluyendo su estructura y diseño. Se puede usar UML para el modelado de negocios y modelado de otros sistemas no-software también (Unified Modeling Language [UML], 2005).

En UML se puede modelar una gran cantidad de aplicaciones, que se ejecuten en cualquier tipo y combinación de hardware, sistema operativo, lenguaje de programación, y red (Unified Modeling Language [UML], 2005).

UML define trece tipos de diagramas, divididos en tres categorías: Seis tipos de diagramas representan la estructura estática de la aplicación; tres representan los tipos generales de comportamiento; y cuatro representan los diferentes aspectos de las interacciones (Unified Modeling Language [UML], 2005):

- Diagramas de estructura; incluye el diagrama de clases, diagrama de objetos, diagrama de componentes, diagrama de estructuras compuestas, diagrama de paquetes, y diagrama de despliegue.

- Diagramas de Comportamiento; incluyen el diagrama de casos de uso (utilizado por algunas metodologías durante la recopilación de requisitos); Diagrama de actividades, y diagrama de estados.

- Diagramas de interacción; derivados en general de los diagramas de comportamiento, incluyen el diagrama de secuencia, diagrama de comunicación, diagrama de tiempos, y el diagrama general de interacciones.

2.1.6 Arquitectura MVC (Modelo-Vista-Controlador)

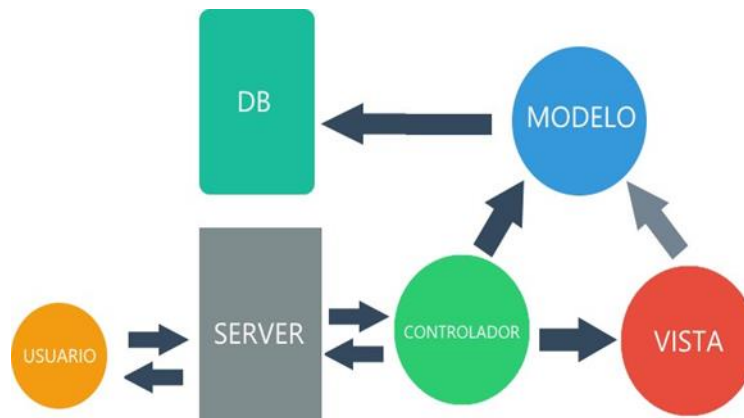


Figura 6. Arquitectura MVC (Modelo-Vista-Controlador). Adaptado de: Arquitectura MVC (Modelo-Vista-Controlador)”<http://michelletorres.mx/mvc-y-su-importancia-en-la-web/>”

El patrón Modelo – Vista – Controlador fue inventado en el contexto de separación entre la interfaz gráfica y el código del funcionamiento de una aplicación, también fue posteriormente aplicado a los lenguajes que están basados en objetos.

En el paradigma MVC, las entradas del usuario, los modelos del mundo exterior y la retroalimentación visual son explícitamente separados y manejados por tres tipos de objetos, cada uno especializado para un conjunto de tareas específicas (Sandoval, 2014).

El objetivo primordial del patrón es dar soporte a los modelos funcionales y mapas mentales de la información relevante para los usuarios, permitiendo un modelo que facilite la consulta y manejo de los mismos. La única manera de construir artefactos manejables es ayudar al usuario a construir modelos del sistema. Pero esto es imposible si el modelo mental no ha sido diseñado dentro del artefacto desde el principio. Intentar adicionar los modelos mentales del usuario cuando ya se ha avanzado en el desarrollo puede ser imposible (Sandoval, 2014).

Las Ventajas de la arquitectura MVC son: (Sandoval, 2014)

- La implementación se realiza de forma modular.
- 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.
- Cualquier modificación que afecte al dominio, como aumentar métodos o datos contenidos, implica una modificación sólo en 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 al modelo de dominio, simplemente se modifica la representación de la información, no su tratamiento.
- 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.

Las Desventajas de la arquitectura MVC son: (Sandoval, 2014)

- Para desarrollar una aplicación bajo el patrón de diseño MVC es necesario una mayor dedicación en los tiempos iniciales del desarrollo. Normalmente el patrón exige al programador desarrollar un mayor número de clases que, en otros entornos de desarrollo, 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 mucho más mantenible, 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.

Partes que involucran la arquitectura MVC (Sandoval, 2014).

- **Modelo**

El modelo es un conjunto de clases que representan la información del mundo real que el sistema debe reflejar. Es la parte encargada de representar la lógica de negocio de una aplicación. Así, por ejemplo, un sistema de administración de datos geográficos tendrá un modelo que representara la altura, coordenadas de posición, distancia, etc. sin tomar en cuenta ni la forma en la que esa información va a ser mostrada ni los mecanismos que hacen que esos datos estén dentro del modelo, es decir, sin tener relación con ninguna otra entidad dentro de la aplicación.

El modelo, a nivel teórico, no debe tener conocimiento acerca de la existencia de las vistas y del controlador. Esta situación es interesante, pero de difícil aplicación práctica, pues deben existir interfaces que permitan a los módulos comunicarse entre sí, por lo que SmallTalk sugiere que el modelo en realidad esté formado por dos submódulos: El modelo del dominio y el modelo de la aplicación.

Modelo de Dominio

Se entiende por modelo de dominio al conjunto de clases definidas a través del análisis de la situación real del problema que queremos solucionar. Si seguimos el ejemplo anterior, sobre datos geográficos, pertenecerían a este dominio las clases; Río, Montaña, Mar, etc...El modelo del dominio no debería tener relación con nada externo a la información que contiene.

Modelo de la aplicación

El modelo de la aplicación es un conjunto de clases que sirven de puente en la relación de las vistas con el modelo de dominio. Tienen conocimiento de las vistas e implementan los mecanismos necesarios para notificar a éstas los cambios que se pudieren dar en el modelo del dominio. El modelo de la aplicación es llamado también coordinador de la aplicación.

- **Vista**

Las vistas son las encargadas de la representación de los datos, contenidos en el modelo, al usuario. La relación entre las vistas y el modelo son de muchas a uno, es decir cada vista se asocia a un modelo, pero pueden existir muchas vistas asociadas al mismo modelo. De esta manera, por ejemplo, se puede tener una vista mostrando la hora del sistema como un reloj analógico y otra vista mostrando la misma información como un reloj digital.

La vista solo necesita la información requerida del modelo para realizar un despliegue. Cada vez que se realiza una actuación, que implica una modificación del modelo de dominio, la vista cambia a través de notificaciones generadas por el modelo de la aplicación. Sencillamente, es la representación visual del modelo que redibuja las partes necesarias cuando se produce una modificación del mismo.

- **Controlador**

El controlador es el encargado de interpretar y dar sentido a las instrucciones que realiza el usuario, realizando actuaciones sobre el modelo. Si se realiza algún cambio, comienza a actuar, tanto si la modificación se produce en una vista o en el modelo. Interactúa con el Modelo a través de una referencia al propio Modelo.

2.1.7 Herramientas utilizadas para el desarrollo

- Net



Figura 7. Net. Adaptado de: NET”<https://www.dizof.com/blog/38>”

La plataforma .NET de Microsoft está diseñada para que se puedan desarrollar componentes software utilizando casi cualquier lenguaje de programación, de forma que lo que escribamos en un lenguaje pueda utilizarse desde cualquier otro de la manera más transparente posible (utilizando servicios web como middleware). Esto es, en vez de estar limitados a un único lenguaje de programación, permitimos cualquier lenguaje de programación, siempre y cuando se adhiera a unas normas comunes establecidas para la plataforma .NET en su conjunto. De hecho, existen compiladores de múltiples lenguajes para la plataforma .NET: Visual Basic .NET, C#, Managed C++, Oberon, Component Pascal, Eiffel, Smalltalk, Cobol, Fortran, Scheme, Mercury, Mondrian/Haskell, Perl, Python, SML.NET... (Berzal & Cortijo, s.f.).

La plataforma .NET apuesta por un futuro en el que las aplicaciones se ejecutan de manera distribuida en Internet. Así, una aplicación se ejecuta en un solo servidor y no existen múltiples copias de la misma. Además, una misma aplicación puede "adornarse" con distintas interfaces para que, desde diferentes dispositivos (teléfonos móviles, PDAs, portátiles, etc.) pueda accederse a la misma. La plataforma .NET no es más que un conjunto de tecnologías para desarrollar y utilizar componentes que nos permitan crear formularios web, servicios web y aplicaciones Windows (Berzal & Cortijo, s,f.).

- **Ventajas de framework .NET**

Las características principales y ventajas que tienen la plataforma Microsoft .NET Framework son (Berzal & Cortijo, s,f.):

Programación Orientada a Objetos: La plataforma fue construida aplicando el paradigma de Programación Orientada a Objetos (POO). El núcleo de lenguajes como C# están basados en los principios POO.

Soporte para múltiples lenguajes: En .NET, la verdadera interoperabilidad entre lenguajes es posible gracias a las capacidades que tiene la plataforma como herencia entre lenguajes (Cross-Language Interoperability) que junto con un sistema de tipos unificado (Common Type System), que veremos en otro post, hace que la integración entre el código escrito en diferentes lenguajes sea total. Esto permite que se puedan usar otros paradigmas de programación tales como la programación funcional con F# o lenguajes dinámicos como Ruby o Python.

Fácil desarrollo basado en componentes: En la plataforma .NET es más fácil implementar componentes o bibliotecas de componentes que comparten funcionalidades. La unidad de código compartido en .NET se denomina ensamblado (assembly), que lleva información de la versión y todos los metadatos necesarios para usarlo.

Simplifica el despliegue de las aplicaciones: En contraste con las aplicaciones basadas en componentes COM, no es necesario el registro de los ensamblados, Con un “Xcopy Deployment” es suficiente, es decir con copiar los ensamblados vale. Se ha eliminado por completo el clásico problema de DLL HELL, gracias a que múltiples versiones de un ensamblado pueden coexistir en la misma máquina. Un ejemplo de esto es el propio .NET Framework, que podemos tener diferentes versiones instaladas.

Soporte para Biblioteca de Clases Base (Base Class Library): .NET Framework viene con un conjunto de bibliotecas de clases que proveen bloques básicos para construir aplicaciones, todas se proporcionan de manera consistente y están diseñadas bajo los principios de la POO. Ejemplos de estas bibliotecas incluyen el uso de colecciones, manipulación de texto, acceso a bases de datos, manipulación del sistema de archivos, etc.... Implementación de varios tipos de aplicaciones: Gracias a la Biblioteca de Clases Base (BCL) es muy fácil el poder implementar cualquier tipo de aplicación. Ya sean basadas en escritorio (Windows Forms y Windows Presentation Foundation [WPF]), aplicaciones Web (ASP.NET), servicios Web (Windows Communication Foundation [WCF]), aplicaciones para dispositivos inteligentes (Compact Framework), aplicaciones móviles para Windows Phone, aplicaciones Cloud (Microsoft Azure) y más...

Interoperabilidad con código existente: La plataforma .NET tiene muy buen soporte para reutilizar componentes de software existentes que fueron escritos por tecnologías antiguas como los componentes COM y para proporcionar acceso a funciones del API de Win32 nativas del sistema operativo a través de un mecanismo llamado P/Invoke.

Manejo de excepciones: El tratamiento de errores en el .NET Framework se proporciona a través de un mecanismo conocido como el Error Handling. Esto elimina las comprobaciones

manuales que se hacían en la programación de Win32 y basada en COM con los llamados HRESULT.

Modelo de seguridad mejorada: El motor en tiempo de ejecución de la plataforma .NET tiene un mecanismo de seguridad integrada llamado Code Access Security (CAS). Este modelo de seguridad es ortogonal a los mecanismos de seguridad proporcionados por el sistema operativo, como acceso control lista (ACL) y tokens de seguridad de Windows.

Soporte para comunicación con Servicios Web: Desde el inicio de la plataforma .NET Framework, ha tenido soporte para servicios web como una forma de realizar llamadas a procedimientos remotos basadas en Simple Object Access Protocol (SOAP) y XML y poder establecer una Arquitectura Orientada a Servicios (SOA).

- **C Sharp**



Figura 8. C Sharp. Adaptado de: C SHARP. Honey Comb Software” <https://honeycombsoft.com/Article/Detail/4>”

C Sharp es un lenguaje elegante, con seguridad de tipos y orientado a objetos, que permite a los desarrolladores crear una gran variedad de aplicaciones seguras y sólidas que se ejecutan en .NET Framework .NET. Puede usar C# para crear aplicaciones cliente de Windows, servicios web

XML, componentes distribuidos, aplicaciones cliente- servidor, aplicaciones de base de datos y muchas, muchas más cosas. Visual C# proporciona un editor de código avanzado, prácticos diseñadores de interfaz de usuario, un depurador integrado y muchas otras herramientas que facilitan el desarrollo de aplicaciones basadas en el lenguaje C# y .NET Framework (Saisang, Cai; et al, 2015).

- **Ventajas de C Sharp**

Las características principales y ventajas que tienen la plataforma Microsoft .NET Framework son (Saisang, Cai; et al, 2015):

Declaraciones en el espacio de nombres: Al empezar a programar algo, se puede definir una o más clases dentro de un mismo espacio de nombres.

Tipos de datos: En C# existe un rango más amplio y definido de tipos de datos que los que se encuentran en C, C++ o Java.

Atributos: Cada miembro de una clase tiene un atributo de acceso del tipo público, protegido, interno, interno protegido y privado.

Pase de parámetros: Aquí se puede declarar a los métodos para que acepten un número variable de parámetros. De forma predeterminada, el pase de parámetros es por valor, a menos que se use la palabra reservada ref, la cual indica que el pase es por referencia.

Métodos virtuales y redefiniciones: Antes de que un método pueda ser redefinido en una clase base, debe declararse como virtual. El método redefinido en la subclase debe ser declarado con la palabra override.

Propiedades: Un objeto tiene intrínsecamente propiedades, y debido a que las clases en C# pueden ser utilizadas como objetos, C# permite la declaración de propiedades dentro de cualquier clase.

Inicializador: Un inicializador es como una propiedad, con la diferencia de que, en lugar de un nombre de propiedad, un valor de índice entre corchetes se utiliza en forma anónima para hacer referencia al miembro de una clase.

Control de versión: C# permite mantener múltiples versiones de clases en forma binaria, colocándolas en diferentes espacios de nombres. Esto permite que versiones nuevas y anteriores de software puedan ejecutarse en forma simultánea.

- **SQL Server**



Figura 9. SQL Server. Adaptado de: SQL SERVER. Revista Digital” <http://bibliotecaprofesional.com/que-es-sql-server/>”

Microsoft SQL Server es un sistema de gestión de base de datos relacional (RDBMS) producido por Microsoft. Su principal lenguaje de consulta es Transact-SQL, una aplicación de las

normas ANSI / ISO estándar Structured Query Language (SQL) utilizado por ambas Microsoft y Sybase (Santamaría, Hernández, & Javier, 2015).

Características de Microsoft SQL Server (Santamaría, Hernández, & Javier, 2015):

- Soporte de transacciones.
- Escalabilidad, estabilidad y seguridad.
- Soporta procedimientos almacenados.
- Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información.
- Permite administrar información de otros servidores de datos.

Este sistema incluye una versión reducida, llamada MSDE con el mismo motor de base de datos, pero orientado a proyectos más pequeños, que en su versión 2005 pasa a ser el SQL Express Edition, que se distribuye en forma gratuita (Santamaría, Hernández, & Javier, 2015).

Microsoft SQL Server constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son Oracle, Sybase ASE, PostgreSQL o MySQL (Santamaría, Hernández, & Javier, 2015).

- **Bootstrap 3.0**



Figura 10. Bootstrap 3.0. Adaptado de: BOOSTST RAP 3.0”<https://mycoderweb.com/desarrollo-web/bootstrap-logo/>”

Bootstrap 3.0 es un framework o conjunto de herramientas de software libre para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como, extensiones de JavaScript opcionales adicionales (Bootstrap, s.f.).

La mayor ventaja es que se pueden crear interfaces que se adapten a los distintos navegadores (responsive design) donde se apoyaría en un framework potente con numerosos componentes webs. (Wikipedia s.f.)

Las características principales son: (Bootstrap, s.f.)

Bootstrap ofrece una serie de plantillas CSS y ficheros Javascript que permiten integrar el framework de forma sencilla y potente en los proyectos webs.

- Permite crear interfaces que se adapten a los diferentes navegadores, tanto de escritorio como tablets y móviles a distintas escalas y resoluciones.
- Se integra perfectamente con las principales librerías Javascript, por ejemplo, JQuery.

- Ofrece un diseño sólido usando LESS y estándares como CSS3/HTML5.
- Es un framework ligero que se integra de forma limpia en el proyecto actual.
- Funciona con todos los navegadores, incluido Internet Explorer usando HTML Shim para que reconozca los tags HTML5.
- Dispone de distintos layout predefinidos con estructuras fijas a 940 píxeles de distintas columnas o diseños fluidos.

2.2 Marco Metodológico

2.2.1 Prototipado evolutivo. En el proyecto se trabajó con la metodología prototipado evolutivo, ya que permitió que parte del sistema, se construyera de manera rápida. Esto facilitó la comprensión de ciertos aspectos en los que se minimizaba el riesgo y la incertidumbre, gracias a que a medida que se iba evolucionando, se iba refinando poco a poco los requisitos del software que se estaba desarrollando. Es una de las metodologías ágiles más usada, gracias a que desarrolla el concepto del sistema a medida que progresa el proyecto.

Para el desarrollo de esta metodología se inició con la definición de los objetivos globales para el software, luego se identificaron los requisitos conocidos y las áreas del esquema en donde fue necesario tener más definición. Este modelo se utilizó para que el cliente pudiera dar sus puntos de vista y sea parte del software. Es básicamente prueba y error ya que, si el cliente no le gustaba una parte del prototipo, significa que la prueba falló, por lo cual se debía corregir el error que se tenga hasta que el cliente quedará satisfecho. Además, se construyeron diferentes prototipos en poco tiempo, usando los programas adecuados. Una gran ventaja es que no se debió utilizar mucho

dinero, ya que se escogió un prototipo, el cual fue en el que se basó el verdadero desarrollo del software.

Gracias a la interacción que ocurrió constantemente con el cliente, el prototipo se fue ajustando a los cambios que él quiere, y esto permitió que al mismo tiempo se entendiera mejor lo que se debía hacer, y el cliente así pudo ver los resultados a corto plazo.



Figura 11. Ciclo de vida de la metodología prototipado evolutivo. Adaptado de: Fuente: Roger S. Pressman. Ingeniería del Software, un enfoque práctico

Las ventajas de esta metodología son: (McConnell, 1996)

- Es ideal para sistemas que no tienen bien definidos los requisitos.
- La especificación se puede mostrar de forma creciente.
- Es ideal para proyectos cuyos requisitos cambian con rapidez.

- Cuando el cliente no puede especificar el conjunto total de requisitos.
- Se puede ir creando versiones desechables hasta quedarnos con una versión que resuelva los inconvenientes de las versiones que desechamos.

En el desarrollo de la herramienta, se fue haciendo un análisis de la literatura sobre la planificación prospectiva y método estrategia de los actores. También se hizo uso de algunas herramientas que manejaban estos métodos de planificación. Se realizaron estas actividades con el fin de poder identificar las necesidades que hay, y observar claramente los requisitos de prioridad alta que se deben implementar en la herramienta.

2.2.2 Fases del proyecto. Teniendo en cuenta que la metodología que se utilizó es la de prototipado evolutivo, se dividió el plan de trabajo para el análisis, diseño y desarrollo de la herramienta en seis fases. La fase dos hasta la fase cinco fueron de forma iterativa, ya que esta herramienta con el transcurso de su desarrollo se le fueron haciendo ajustes necesarios. Esto se hizo con el fin de ir retroalimentado las fases, hasta que se pudiera encontrar la versión que resolviera todos los problemas que se fueran encontrando en el desarrollo. Basándonos en el ciclo de vida de esta metodología, las fases que se trabajaron en la herramienta fueron:

2.2.2.1 Recolección y refinamiento de requisitos. En esta fase, se recolectó la información suficiente que dio pauta al análisis de requisitos. Esta información fue de acuerdo sobre un contexto el cual contiene una problemática y se decidió darle una solución a esta. Luego se hizo una lista de objetivos y requisitos, los cuales se le hicieron un análisis para tener conocimientos de los alcances y las variables de la herramienta. Posteriormente se hizo un estudio en el cual se analizó

y se escogieron las herramientas tecnológicas que mejor se adecuaran al plan de trabajo de la herramienta.

Para la primera fase, se realizó un análisis a fondo a la herramienta MACTOR, esto con el fin de obtener una visión general de los procesos que se realizaron con respecto a la metodología prospectiva. Como resultado de este análisis, se encontraron nuevos requisitos a implementar, los cuales generan más facilidad en la usabilidad de la herramienta y proveen mayores ventajas para los usuarios. Gracias a esta fase, se logró hacer un planteamiento esquemático que fue como base para el diseño de la herramienta.

2.2.2.2 Diseño rápido. En la fase dos, que se denomina como diseño rápido, se procedió a realizar un pequeño diseño de la herramienta. Esto se hizo para dar una idea de cómo podrá verse el sistema que se iba a desarrollar. También se analizó la capacidad y el modelo de datos que se iba a implementar, y posteriormente se procedió a diseñarlo.

Para la fase dos, teniendo ya una especificación de requisitos, se procedió a implementarlos al lenguaje de modelado unificado (UML). En base a los resultados que se obtuvieron en la fase de análisis, se realizó el diseño de los requisitos, los cuales fueron modelados en los diagramas de caso de uso, actividades y secuencia.

Posteriormente, se realizó un prototipo en papel o un Mockup en la herramienta Balsamiq, los cuales fueron desarrollados de acuerdo a la fase de análisis y diseño de requisitos.

2.2.2.3 Construcción de la herramienta. Una vez realizado el diseño rápido en base al prototipo de papel o el mockup, se llevó a cabo esta etapa, la fue donde se implementen los requisitos analizados y el diseño rápido.

Aquí se desarrolló la herramienta, la cual se tuvo una idea más clara y real de lo que se pretendía llegar a realizar.

En esta fase para el desarrollo de la herramienta se procedió al manejo del lenguaje de programación C#, el framework de .NET y el motor de bases de datos SQL Server. Posteriormente se trabajó en la implementación las diferentes interfaces, como los módulos de la herramienta en torno web.

Estas interfaces y módulos se fueron construyendo a partir del análisis y diseño de requisitos que se obtuvieron en las fases anteriores.

Posteriormente se realizó la validación y aplicación mediante pruebas y determinación de errores. Esto se hace con el fin de verificar que la herramienta cumpla con los requisitos y diagramas UML establecidos en las fases de análisis y diseño.

Se llevaron a cabo varias etapas de pruebas en búsqueda de poder tener un producto solido en funcionamiento y así, posteriormente se procedió seguir a la siguiente fase.

2.2.2.4 Evaluación de la herramienta por parte del cliente. Cuando ya se tenía un primer prototipo de la herramienta, se procedió a mostrar al cliente, quien en este caso fue el grupo de investigación Innotec. Para ello, se acordó una reunión en la cual se tuvo como fin realizar una evaluación de la herramienta, y poder observar si cumplía con los requisitos mencionados tanto en la fase de análisis como en la de diseño.

2.2.2.5 Refinamiento de la herramienta. Una vez finalizada la evaluación de la herramienta, se procedió a realizar un refinamiento de la misma, en donde se modificaron algunos puntos, para que se realizaran los procesos de manera diferente. Esta fase fue muy importante ya

que en este punto se logró ver que la herramienta cumplía o no con las expectativas del cliente, por lo cual se hicieron ciertos ajustes refinamiento en algunos requisitos.

2.2.2.6 Producto de ingeniería. Una vez finalizado este ciclo, se procedió a entregar el producto de ingeniería, con todas sus pruebas de funcionalidad.

Adicionalmente, en esta fase, también se creó la documentación de todas las etapas del proyecto y se presenta el documento con sus respectivas correcciones.

3. Resultados del Proyecto

Para dar puesta en marcha a la herramienta software con aplicación web, Planeación Juego de Actores (PJA), se tomó como punto de partida una herramienta que existe actualmente en el mercado y maneja la metodología planeación prospectiva, cuyo nombre es Lipsor MACTOR. Para esta herramienta se manejó la misma metodología y con el fin de adecuarse a la necesidad actual que tiene el grupo de investigación INNOTECH.

Adicionalmente, se hizo uso de la herramienta Lipsor MACTOR por un periodo de tiempo, en el cual se crearon actores y proyectos de ejemplo, con el fin de observar detalladamente su trabajo. Esto se realizó en busca de llegar a entender su funcionamiento e identificar de manera más clara las dificultades que esta presenta.

Posteriormente, luego de hacer pruebas y de enumerar los déficits que se identificaron en la herramienta anterior, se realizó un análisis de cuáles son los requisitos que se van a implementar en la herramienta Planeación Juego de Actores (PJA), para que de esta manera se pueda fortalecer

y potencializar un producto más completo, el cual llegue a brindar una mejor satisfacción en el cliente.

3.1 Requisitos Establecidos para la Herramienta

Tomando en cuenta que la metodología que se trabajó es prototipado evolutivo, y estudiando la primera fase que es recolección y refinamiento de requisitos, fue como se logró obtener el listado de requisitos funcionales que son prioridad alta para el desarrollo de la herramienta, ya que en esta fase es donde se recolecta la información suficiente que darán pauta al análisis de requisitos.

La siguiente Tabla muestra las abreviaturas utilizadas en el plan de versiones. El documento de requisitos se encuentra en el Apéndice A.

Tabla 2.

Definición de abreviaturas utilizadas.

Termino	Abreviación	Descripción
Requisito funcional	RF	Hace referencia a una tarea que desempeña el sistema y que está directamente involucrada con el problema a resolver.
Diagramas de Caso de Uso	DCU	Describe el procedimiento a realizar para ejecutar la tarea a la que el usuario accede en la herramienta.

3.1.1 Requisitos para el desarrollo. Los requisitos tanto funcionales, como los no funcionales, se encuentran estipulados y especificados en el Documento Especificación de Requisitos Software, el cual se puede encontrar en el Apéndice A.

En la tabla 3, se presenta de forma específica el contenido del requisito funcional Crear proyecto, que es de prioridad alta de la herramienta Planeación Juego de Actores (PJA).

Tabla 3.

Requisito funcional Crear proyecto para la herramienta PJA.

Número de requisito:	RF-04
Nombre del requisito:	Permitir la creación de proyectos.
Descripción:	El portal web debe permitir al usuario la creación de uno o más proyectos.
Restricciones:	1-Al ingresar datos erróneos o no válidos en el formulario de creación se debe generar el aviso de dicho error. 2-Para que se pueda crear un proyecto se le debe asignar unos actores. 3-El usuario con rol Administrador será el único quien podrá crear un proyecto.
Prioridad:	Alta

En la tabla 4, se presenta de forma general los requisitos funcionales de la herramienta Planeación Juego de Actores (PJA).

Tabla 4.

Requisitos propuestos para la herramienta PJA.

Id Requisitos	Requisitos	DCU
RF-01	Permitir registrarse	
RF-02	Permitir inicio de sesión	
RF-03	Recuperar contraseña de inicio de sesión	
RF-04	Permitir la creación de proyectos	
RF-05	Permitir visualización de proyectos	
RF-06	Permitir edición de un proyecto	
RF-07	Permitir eliminación de un proyecto	
RF-08	Permitir asignación de usuarios a un proyecto	
RF-09	Permitir visualización de perfiles de usuario	
RF-10	Permitir la creación de perfil de usuario	
RF-11	Permitir la edición de perfil de usuario	
RF-12	Permitir la eliminación de perfil de usuario	

Id Requisitos	Requisitos	DCU
RF-13	Permitir visualización de notificaciones.	
RF-14	Permitir la creación de notificaciones.	
RF-15	Permitir la edición de notificaciones.	
RF-16	Permitir la eliminación de notificaciones.	
RF-17	Permitir la configuración de privacidad para las notificaciones.	
RF-18	Permitir ingreso al menú administrador.	
RF-19	Permitir ingreso al menú usuario.	
RF-20	Permitir ingreso al menú de espera.	
RF-21	Agregar usuarios a un proyecto.	
RF-22	Eliminar usuarios de un proyecto.	
RF-23	Editar información de usuario.	
RF-24	Permitir la opción de entender resultados.	
RF-25	Redactar informe	
RF-26	Ingresar al menú de ayuda.	
RF-27	Crear información del módulo de datos.	
RF-28	Editar información del módulo de datos.	
RF-29	Eliminar información del módulo de datos.	
RF-30	Generar información de la matriz de influencia directa e indirecta (MIDI)	
RF-31	Visualizar resultados de matriz de influencia directa e indirecta (MIDI)	
RF-32	Eliminar resultados de matriz de influencia directa e indirecta (MIDI)	
RF-33	Visualizar resultados de relación fuerza de la matriz de influencia directa e indirecta (MIDI)	
RF-34	Eliminar resultados relación fuerza de la matriz de influencia directa e indirecta (MIDI)	
RF-35	Visualizar resultados de balance neto de la matriz de influencia directa e indirecta (MIDI)	
RF-36	Eliminar resultados balance neto de la matriz de influencia directa e indirecta (MIDI)	
RF-37	Visualizar resultados fuerza máxima de la matriz de influencia directa e indirecta (MIDI)	
RF-38	Eliminar resultados fuerza máxima de la matriz de influencia directa e indirecta (MIDI)	

3.2 Diseño de la Herramienta

Tomando en cuenta que la metodología que se trabajó es prototipado evolutivo, y estudiando la segunda fase que es la de diseño rápido, se procedió a realizar un pequeño diseño de la herramienta.

Adicionalmente en esta fase 2, se tomó como base la especificación de requisitos (generada en la fase de análisis), y se procedió a realizar su implementación al lenguaje de modelado unificado (UML).

A continuación, se presentan los diagramas UML elaborados en el desarrollo de la herramienta Planeación Juego de Actores (PJA), conformados por diagramas de casos de uso, diagramas de estados y diagrama de actividades. (Ver Apéndice B)

3.2.1 Diagrama de casos de uso. Los diagramas de casos de uso UML se elaboraron con el fin de especificar el comportamiento, interacción y comunicación de los usuarios con el sistema y viceversa. Dichos diagramas se construyeron bajo el soporte de la herramienta Software STAR UML.

Cabe recordar que todos los diagramas de casos de uso de la herramienta se encuentran en el documento Apéndice B– Diseño.

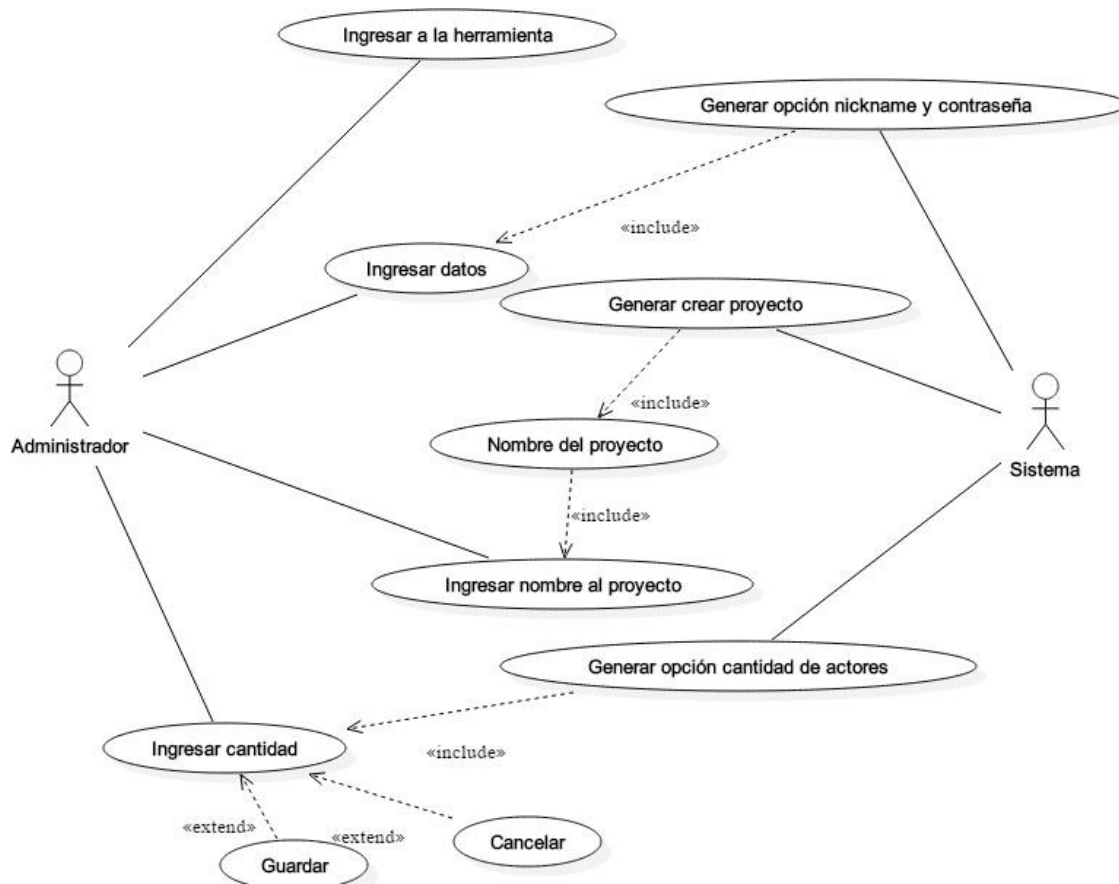


Figura 11. Diagrama de casos de uso – Crear proyecto

3.2.2 Diagrama de estados. Estos diagramas han sido elaborados para soportar la explicación sobre todos los estados posibles en los que puede ingresar un objeto particular y la manera en que modifica el estado del objeto como resultado de los eventos que llegan a él.

Un diagrama de estados es un diagrama utilizado para determinar cada una de las rutas o caminos que puede tomar un movimiento de información luego de ejecutarse cada proceso.

Permite identificar bajo qué pruebas se ejecuta cada uno de los procesos y en qué momento podrían tener una variación. El diagrama de estados permite visualizar de una forma ordenada la ejecución de cada uno de los procesos. Dichos diagramas se construyeron bajo el soporte de la herramienta Software STAR UML.

Cabe recordar que todos los diagramas de Estados de la herramienta se encuentran en el documento Apéndice B- Diseño.

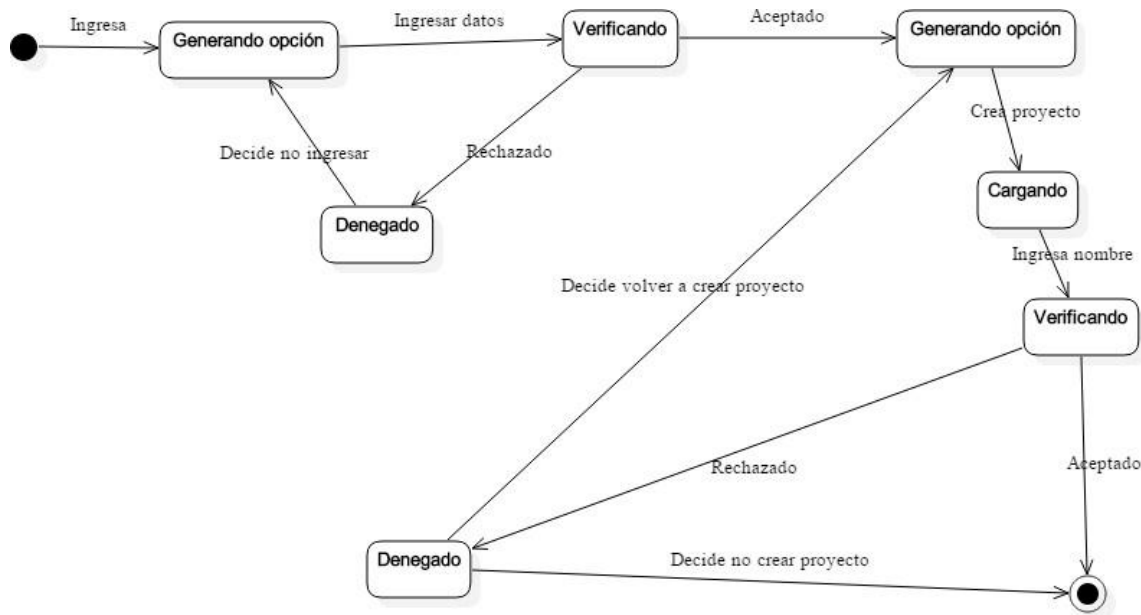


Figura 12. Diagrama de estados – Crear proyecto

3.2.3 Diagrama de actividades. Los diagramas de actividades se elaboraron para mostrar la secuencia de actividades de la herramienta. Estos diagramas muestran el flujo de trabajo desde el punto de inicio hasta el punto final detallando muchas de las rutas de decisiones que existen en el progreso de eventos contenidos en la actividad. Estos también pueden usarse para detallar situaciones donde el proceso paralelo puede ocurrir en la ejecución de algunas actividades. Dichos diagramas se construyeron bajo el soporte de la herramienta Software STAR UML.

Cabe recordar que todos los diagramas de actividades de la herramienta se encuentran en el documento Apéndice B – Diseño.

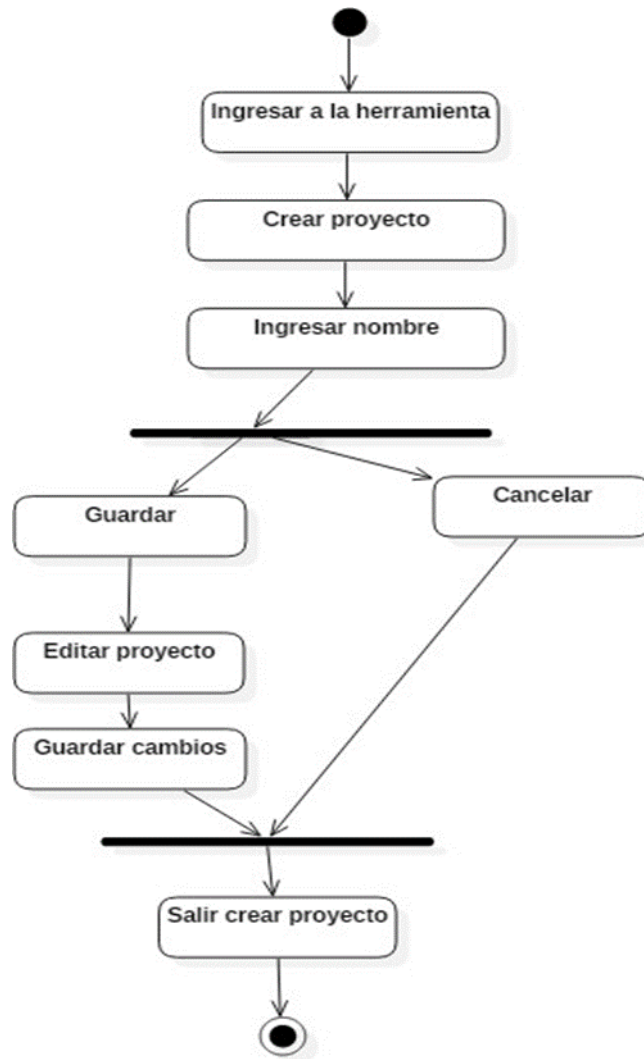


Figura 13. Diagrama de actividades – Crear proyecto

3.3 Desarrollo de la Herramienta

Ante la necesidad identificada de desarrollar una herramienta software con aplicación web, soportada en la apropiación de la metodología estrategia prospectiva, y más específicamente utilizando el método estrategia de los actores, en la cual se pueda gestionar toda la información que se realice en la etapa de planificación, se presentará por medio de diagramas, matrices y cuadros, un esquema general de las acciones que se requieren para determinar los resultados que

se buscan en actividades de prospectiva, es decir, de obtener datos relacionados con los comportamientos futuros de un proyecto.

El objetivo principal de la herramienta, es facilitarle al actor la ayuda para la toma de decisión sobre sus futuras alianzas, conflictos, metas, fortalezas y debilidades en un entorno de trabajo.

A continuación, se mostrará el contenido que es público para cualquier usuario.

- En la interfaz de inicio, el usuario podrá encontrar el menú de inicio, el cual tendrá las diferentes opciones como: registrar una cuenta nueva, ir a la sección de método (etapas de la metodología prospectiva), ir a contacto, e iniciar sesión (si ya tiene su cuenta creada), para acceder a su correspondiente menú.

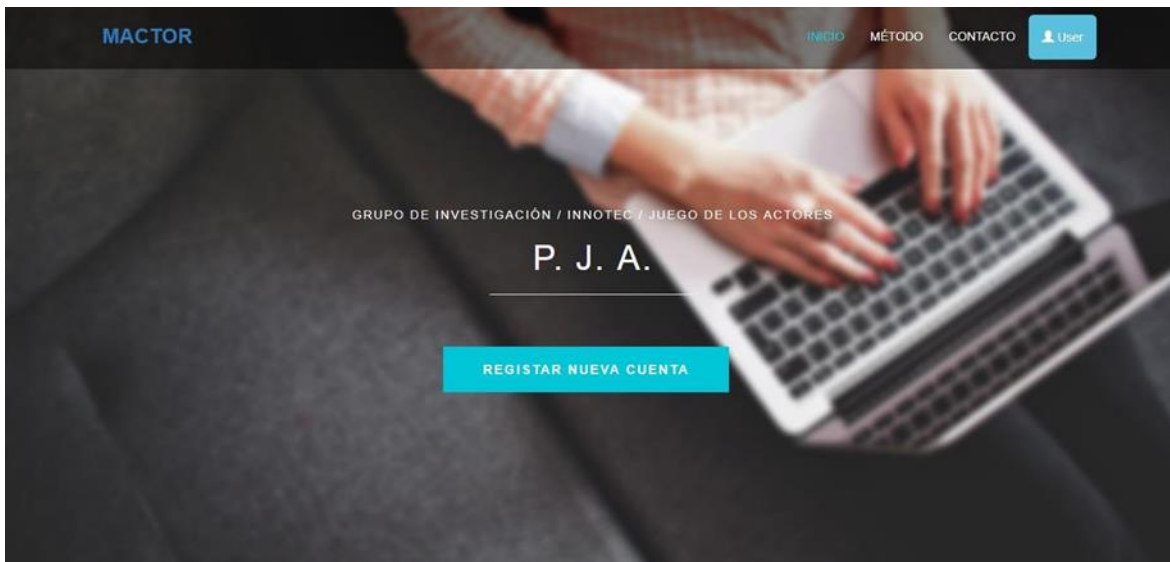


Figura 14. Inicio de la herramienta web PJA.

- En la opción “MÉTODO”, el usuario podrá encontrar información detallada de la metodología Estrategia Prospectiva (juego de los actores). Con esta información, el usuario podrá

tener un conocimiento previo de cómo es el funcionamiento de herramienta, y cuáles son las etapas de trabajo que realiza en el estudio de cada proyecto.



Figura 15. Descripción del método juego de los actores de la herramienta web PJA.

- En la opción “CONTACTO”, el actor podrá encontrar la información respectiva del grupo de investigación INNOTECH (teniendo en cuenta que son los colaboradores del proyecto). Adicionalmente encontrará allí un formulario, el cual podrá ser llenado por si hay dudas con respecto al manejo de la herramienta, o si quiere enviar alguna sugerencia de mejora con respecto a la herramienta o al uso de la metodología.

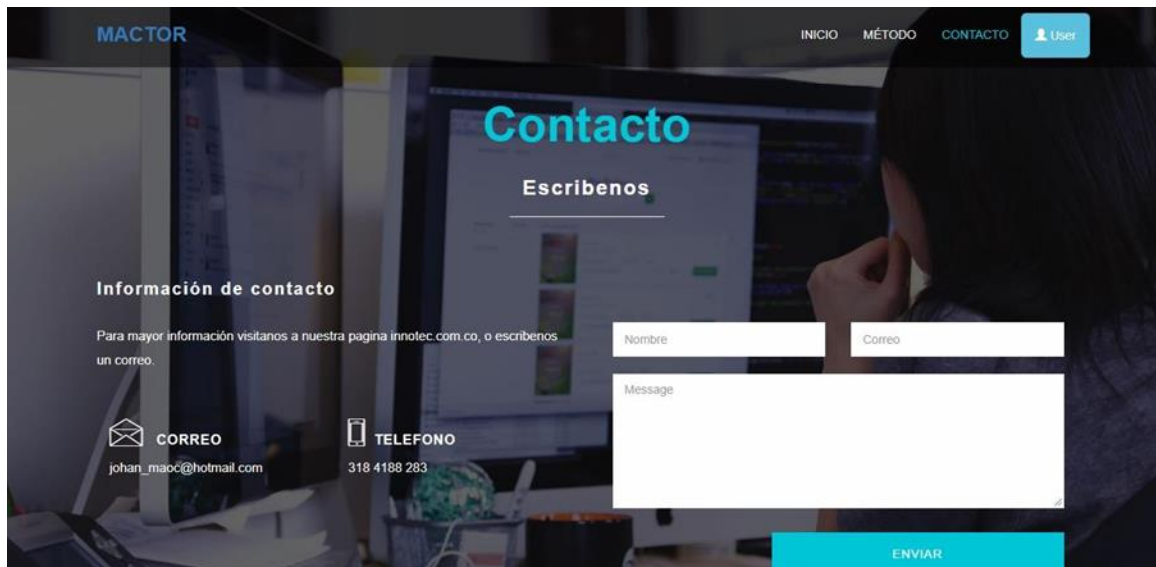


Figura 16. Sección de contacto de la herramienta web PJA

- EL actor podrá encontrar en la opción “INICIAR SESIÓN”, un formulario el cual deberá ser llenado con su cuenta y contraseña. Si el actor no tiene su propia cuenta, deberá registrarse y crear una para acceder al menú correspondiente.



Figura 17. Interfaz de ingreso de la herramienta web PJA

3.3.1 Módulo de ingreso. En el módulo de ingreso, el usuario tendrá la posibilidad de crear su propia cuenta con su respectiva contraseña. También va a poder solicitar el cambio de contraseña, ya sea por olvido de esta, ó simplemente por generar un cambio.

El sistema estará encargado de validar si los datos generados por el actor son correctos o incorrectos, permitiéndole así el acceso a la herramienta.

Para que el actor pueda crear su cuenta y contraseña, primero tiene que ir al formulario de registro y crear su cuenta, para que automáticamente la herramienta lo registre y lo lleve a la interfaz de ingreso.

Sí el actor se registra con el rol administrador, el sistema estará encargado de redireccionarlo al menú del administrador, donde tendrá la opción de la creación del proyecto y muchas más funcionalidades (revisar a más detalle en 3.4 Roles).

Sí ya hay un actor con el rol de administrador, el ingreso de los nuevos registros será automáticamente tomados con el rol de usuario. Posteriormente, si un actor no está registrado en ningún proyecto, deberá esperar a que el administrador de la herramienta lo agregue en uno, y así cuando el actor vuelva a ingresar, el sistema estará encargado de permitir el acceso al menú de usuario. Allí se podrá generar toda la recolección de información y muchas más funcionalidades (revisar a más detalle en 3.4 Roles).



Figura 18. Interfaz de registro de la herramienta web PJA



Figura 19. Interfaz de ingreso de la herramienta web PJA

3.3.2 Módulo de gestión de proyectos. El actor con rol administrador, será el único que podrá tener acceso a este módulo, en el cual va a poder tener la opción de ingresar en “crear proyecto”.

Cuando el actor ingrese en esta opción, automáticamente se generará un formulario el cual deberá ser realizado para la creación de un proyecto. En el formulario irá la información básica del proyecto como es: nombre, una breve descripción y la fecha de su creación.

Adicionalmente en este módulo podrá consultar el listado de proyectos que han sido cargados en la herramienta, y tendrá acceso para ver tanto los detalles del proyecto, como la de los actores. El sistema también le dará la opción de llegado el caso eliminar un proyecto o simplemente editar la información del mismo. Cabe resaltar de que no se tendrá la opción de editar la información que haya generado los usuarios.



Figura 20. Módulo Gestión de proyectos de la herramienta web PJA

3.3.3 Módulo de datos. En este módulo, sólo podrá tener acceso aquel actor que tenga como rol Usuario, y que ya el administrador de la herramienta lo haya registrado en un proyecto.

El actor al ingresar en este módulo, el sistema le va a generar un formulario, el cual deberá ser llenado con todos los datos que se requieran allí. La información que ingrese el actor es muy importante, ya que de esta información dependerá los resultados que se obtengan en el estudio y análisis del proyecto, por tal motivo se pide al actor que el ingreso de la información sea coherente.

Cuando el sistema valide la información, mostrará la ficha que se generó, esto se hará con el fin de que el actor pueda ver de forma dinámica la información que ha ingresado. Adicionalmente el sistema le dará la opción al administrador de poder suprimir la información, seguir añadiendo o cerrar y guardar la ficha generada.

3.3.4 Módulo de gestión de usuarios. Para este módulo, sólo tendrá acceso el actor que tenga rol administrador. Al ingresar, automáticamente el sistema le va a generar el listado de todos los actores que actualmente están participando en los proyectos, y que han sido cargados en la herramienta. Tendrá la opción de listarlos de acuerdo al proyecto que pertenece. Adicionalmente podrá ver todos los detalles y toda la información de cada actor. Llegado el caso podrá eliminar a un actor, pero no podrá añadir o editar la información que haya ingresado.

Luego de ser llenado el formulario con toda la información requerida, se va a generar un listado con todos los proyectos que han sido cargados en la herramienta. Él podrá visualizar una descripción de cada uno, esto con el fin de que pueda identificar en cual va a participar.

Cuando el actor ya haya identificado el proyecto en el cual quiere participar, podrá ingresar en la opción “Unirse al proyecto”. En el momento en que el actor ya pertenezca a uno, el

administrador deberá ser el encargado de ingresar la información del actor, esto con el fin de realizar el estudio.



Figura 21. Módulo Gestión de usuarios de la herramienta web PJA

3.3.5 Módulo generar Matriz de Influencia Directa (MID). Para este módulo sólo tendrá acceso el actor que tenga rol administrador. Al ingresar en este módulo, el sistema automáticamente le va a generar el listado de los proyectos que han sido cargados en la herramienta, dándole la opción de poder elegir uno. Al elegir un proyecto, el administrador va a poder ingresar en la matriz de influencias directas y poder llenarla con la información que los usuarios generaron en la herramienta. Posteriormente la herramienta hará el análisis y estudio con la información ingresada, y le notificara al administrador los resultados que se obtuvieron del proyecto.

Luego de obtener los resultados del análisis y estudio de la herramienta, el sistema generará al usuario un botón “Entender Resultados”, el cual le indicara una explicación de los resultados

obtenidos, además, también le indicara las ventajas y desventajas de los resultados que se obtuvieron, para que sirven estos a futuro, y, por último, cuáles son los objetivos de analizar y estudiar esta información.

Posteriormente, el administrador podrá elegir la opción si desea hacer públicos los resultados a los demás usuarios o no. Esto se hace con el fin de poder tener una mayor seguridad de los resultados que haga la herramienta. Además, el administrador podrá tener la opción de elegir cual es el momento indicado para compartirlos.

Adicionalmente, el administrador y usuario podrá encontrar que los diferentes resultados de la matriz MIDI, se dividen en cinco partes, las cuales son:

- Resultados de matriz MIDI.
- Resultados de las relaciones de fuerza.
- Resultados de la balanza neta de influencias.
- Resultados de matriz de máxima influencias directas e indirectas.
- Resultados de relaciones de fuerza máxima.

En cada resultado, el administrador y usuario, podrá recurrir al botón “Entender Resultados”, el cual será de gran ayuda, como se indicó anteriormente.

Posteriormente, el actor encontrará un cuadro de texto, el cual será generado por el sistema. Esto se hace con el fin de que él pueda redactar una síntesis del estudio que hizo la herramienta. Allí podrá llenar con comentarios esta ficha y el sistema le dará la opción de aceptarla o anularla. Esta parte es importante, ya que el administrador al ingresar estos comentarios, le servirá de gran ayuda al momento de entender y explicar los resultados que se obtuvieron del proyecto a los demás usuarios que participan en él.



Figura 22. Módulo Matriz (MID) de la herramienta web PJA

3.3.6 Módulo de ayuda. Este módulo es muy importante, ya que allí los actores podrán encontrar todo el contenido acerca de la herramienta y el paso a paso del método estrategia de los actores. Se busca que el módulo sea simple y fácil de entender por los actores que hagan uso de la herramienta.

Los actores van a encontrar el paso a paso con las indicaciones pertinentes para que puedan ingresar toda la información necesaria para el análisis y estudio del proyecto. También va a poder encontrar ciertas recomendaciones, las cuales le van a ser de gran ayuda en el momento de gestionar toda la toma de datos.

Este módulo de ayuda está pensado para brindar o facilitar a los actores el uso de la herramienta, y por ello es que se ha planeado que sea un módulo en línea, el cual va a ser de gran ayuda para todos los usuarios.

1.1. Interfaz de inicio

En la interfaz de inicio el usuario, va poder encontrar cinco botones muy utiles para encontrar información tanto de la metodología, como del metodo. Además allí encontrará el botón para poder encontrar información por si tiene dudas o alguna sugerencia para poder mejorar. También podrá encontrar el botón de ingreso y el de crear una nueva



centa.
de los botones de la interfaz de inicio.

- 1** Botón de inicio

INICIO

Este botón sirve para siempre ir al inicio de la herramienta.
- 2** Botón método

MÉTODO

Figura 23. Módulo de ayuda de la herramienta web PJA

3.3.7 Módulo para editar informe. En este módulo, el administrador tendrá diferentes opciones con respecto al estudio que se realizó. El sistema le dará la opción de poder configurar el informe, seleccionando los resultados de su interés y también podrá configurar los formatos de los resultados. En general, él podrá ir seleccionando las partes que le interesó del estudio y borrar las que no le interesa. Posteriormente el sistema le genera la opción de guardar estudio o de suprimir todo el análisis que obtuvo la herramienta.

The screenshot displays the MACTOR web application interface. At the top, there is a navigation menu with the following items: INICIO, GESTIÓN DE PROYECTOS, GESTIÓN DE USUARIOS, MATRIZ (MID), INFORME, AYUDA, and a user profile section for 'admin1@hotmail.com' with a 'Salir' button. The main content area features a large heading 'Realizar Informe' in blue. Below this is a table with the following data:

Código	Nombre	Descripción	Fecha	Operaciones
1	girod	proyecto para comercializar productos	28/08/2018 0:00:00	[Icono de editar] [Icono de borrar]
3	Agropolis	Proyecto para el agro	26/04/2012 0:00:00	[Icono de editar] [Icono de borrar]
7	Colciencias	proyecto	28/08/2016 0:00:00	[Icono de editar] [Icono de borrar]
25	sicaac	Proyecto para la contratación	03/08/2015 0:00:00	[Icono de editar] [Icono de borrar]
26	sti	Proyecto para la mejorar	20/02/2015 0:00:00	[Icono de editar] [Icono de borrar]

At the bottom of the page, there is a footer section with three columns: 'GRUPO DE INVESTIGACIÓN STI - UIS', 'UBICACIÓN', and 'CONTACTO'. The 'CONTACTO' column includes the email 'gruposti@uis.edu.co', the phone number '(7) 6351006', and the PBX number '(7) 6344000 Extensión. 2873-1304'.

Figura 24. Módulo Editar un informe de la herramienta web PJA.

3.3.8 Roles de la herramienta

- **Rol administrador**

Este rol va a ser tomado por el actor que va a crear los diferentes proyectos en la herramienta. Además, va a tener acceso a toda la información y resultados del estudio. Podrá añadir o suprimir la información que crea conveniente. Tiene el poder de decidir cuantos actores va a tener el proyecto y podrá elegir si la información que se recolectó de los usuarios va a ser pública o privada para los demás usuarios. También es el encargado de estipular cuales son los objetivos del proyecto, y si el estudio de la herramienta es satisfactorio o si decide suprimir el análisis de la herramienta.

Adicionalmente, el administrador podrá encontrar una sección en la cual se le brindará la opción de crear una conclusión acerca del análisis y estudio que hizo la herramienta, como también exportar el estudio del proyecto y poder elegir cual parte del estudio quiere eliminar.



Figura 25. Menú del rol Administrador de la herramienta web PJA.

- **Rol usuario**

Este rol va a ser para el actor con rol usuario, el cual participa activamente en un proyecto específico.

Al momento de ingresar en la herramienta, quedará registrado en la lista de actores y estará en estado de espera, para que el administrador lo agregue en un proyecto y así, logre acceder al menú “usuario”, para que posteriormente pueda ingresar toda su información.



Figura 26. Menú del rol Usuario de la herramienta web PJA

3.4 Desarrollo e Implementación

Esta herramienta se desarrolló, para que los diferentes actores puedan ver desde distintas perspectivas su proyecto, adicionalmente también podrán entenderlo y ver las mejores opciones a tomar. Esto se hace con el fin de que se pueda reducir el margen de fracaso de un proyecto.

Además, se desarrolló con un diseño que hace sencilla su navegabilidad, esto con el fin de que se pueda entender el paso a paso de la herramienta; como se está aplicando el método estrategia de los actores y sobre todo que tenga una facilidad al momento de la recolección de información, ya que es lo más importante para poder hacer el análisis y el estudio.

En la herramienta se crearon dos roles, como se había especificado anteriormente, los dos usuarios son: Administrador y usuario.

Adicionalmente, la herramienta cuenta con diferentes módulos, los cuales van a ser llenados con información por los actores que participan en el proyecto.

La herramienta le indicará al actor cual es el procedimiento adecuado a seguir, esto con el fin de tener un análisis concreto de los posibles escenarios que se van a presentar. Los módulos que se diseñaron para la primera versión de la herramienta son: (Los módulos fueron explicados detalladamente en la sección anterior)

- Módulo de ingreso.
- Módulo de ayuda.
- Módulo de gestión de proyectos.
- Módulo proyecto.
- Módulo de gestión de usuarios.
- Módulo generar matriz de influencia directa (MID).
- Modulo para editar un informe.

3.5 Arquitectura del Sistema

La plataforma escogida para el desarrollo de la herramienta web, es el paquete de herramientas de Visual Studio .NET. Teniendo en cuenta que es uno de los lenguajes de programación que ofrece un conjunto adecuado de funcionalidades para el desarrollo de aplicaciones Web, y que cuenta con una comunidad muy activa en la internet.

Se utilizó ASP .NET con arquitectura MVC, Lenguaje de Programación C Sharp y base de datos SQL Server, como herramientas de desarrollo.

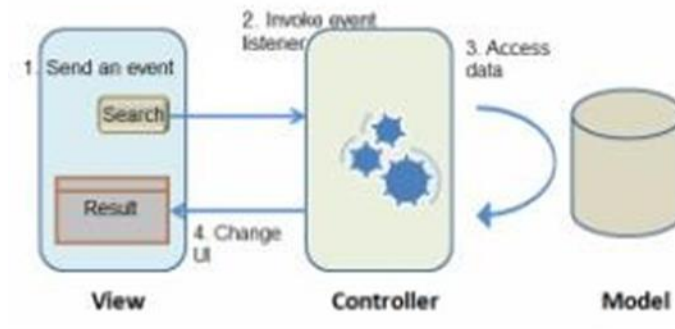


Figura 27. Arquitectura implementada para Planeación Juego de Actores.. Adaptado de: Fuente: Arquitectura implementada para Planeación Juego de Actores. Patrón de Diseño MVC (Modelo Vista Controlador) y DAO (Data Access Object) “<https://jossjack.wordpress.com/2014/06/22/patron-de-diseno-mvc-modelo-vista-controlador-y-dao-data-access-object/>”

3.6 Pruebas de la Herramienta

Para la herramienta web, se verificó que cumpliera con los requisitos y casos de uso establecidos en la fase de análisis y diseño, donde se llevaron a cabo diferentes pruebas con el fin de poder observar el comportamiento de la herramienta a la hora de ejecutar cada una de sus funcionalidades, acciones que permitieron realizar una debida evaluación de los resultados y los ajustes necesarios.

Se realizaron casos de prueba de las funcionalidades a los módulos de la herramienta, haciendo uso del siguiente formulario para registrar cada caso de prueba.

Tabla 5.

Formulario de caso de prueba del requisito 04 de la herramienta.

Formulario para caso de prueba			
ID	CP04	ID Requisito	04
Objetivo	Describe el objetivo del caso de prueba	Fecha de creación	10/10/2018
Actores	Administrador		
Precondiciones	Haber iniciado sesión con su cuenta, contraseña y dirigirse al modulo Gestión de proyectos.		
N° de Entradas	Entradas	Resultado esperado	
2	Hacer clic en el botón "Crear Proyecto"	El sistema redireccionara al actor a un formulario, en el cual ingresara los datos y descripción del proyecto	

Tabla 6.

Formulario dos de caso de prueba del requisito 04 de la herramienta web

Formulario para caso de prueba			
ID	CP04-02	ID Requisito	04
objetivo	Permitir la creación de proyectos.	Fecha de creación	10/10/2018
Actores	Administrador		
Precondiciones	CP04-01		
N° de Entradas	Entradas	Resultado esperado	
2	Dar clic en el botón "Crear nuevo proyecto"	El sistema redirecciona a un formulario, el cual contiene los datos y una breve descripción del proyecto.	

Tabla 7.

Formulario tres de caso de prueba del requisito 04 de la herramienta web

Formulario para caso de prueba			
ID	CP04-03	ID Requisito	04
objetivo	Permitir la creación de proyectos.	Fecha de creación	10/10/2018
Actores	Administrador		
Precondiciones	CP04-02		
N° de Entradas	Entradas	Resultado esperado	
2	Hacer clic en el botón "Guardar"	EL sistema guardara en la base de datos el proyecto.	

4. Conclusiones

Para la realización de la herramienta software de aplicación web, Planeación Juego de los Actores (PJA), se trabajó con la metodología prototipado evolutivo, en la cual se realizó una correcta planificación en cada una de sus fases. Esto en busca de ir obteniendo un producto final que se vaya ajustando a las necesidades del cliente, además, ir cumpliendo satisfactoriamente con sus expectativas.

En la etapa de análisis y diseño, se evidenció la importancia de la utilización de herramientas como diagramas UML y Balsamiq, las cuales fueron de gran ayuda para poder describir y visualizar un plano del sistema, llegando así a vincular la idea del cliente con la del desarrollador para la especificación de los requisitos de la herramienta.

Se evidenció a través de este proyecto, la importancia de la ingeniería de sistemas, ya que se acogió una importante metodología que estudia los posibles escenarios a presentar a futuro, y se combinó con la tecnología, llegando así a optimizar tanto recursos humanos, como económico.

Se desarrolló la primera versión web de una herramienta software que maneja la metodología Estratégica planeación juego de los actores. Esto en busca de brindar un seguimiento más cercano a los diferentes actores que participan activamente en un proyecto, asesorando en todo momento y entregando información valiosa, acerca de sus alianzas, conflictos, ventajas y desventajas a futuro.

Se evidenció el correcto funcionamiento de la herramienta software web, Planeación Juego de los Actores (PJA), mediante casos de pruebas. Estas pruebas se realizaron con datos reales a los diferentes módulos, cumpliendo con los requisitos establecidos en la herramienta.

5. Recomendaciones

Teniendo en cuenta que todo proyecto software o todo proyecto que esté vinculado con tecnología debe estar en continua evolución y mejora, se plantean a continuación las recomendaciones para la herramienta web Planeación Juego de los Actores (PJA).

- Crear el módulo Matriz de Actores por Objetivos (MAO). Es una fase importante del método estrategia de los actores, ya que brinda información valiosa acerca de los objetivos del proyecto con respecto a los actores que han sido inscritos. No se realizó este módulo, ya que se espera implementar en una versión futura de la herramienta.

- Es recomendable realizar capacitaciones a los diferentes usuarios. Esto en busca de que se haga un buen uso y que se puedan adaptar a los diferentes servicios que ofrece la herramienta.
- Es de suma importancia mantener en constante actualización los datos de los proyectos y actores que estén inscritos en la herramienta, ya que, al ingresar nuevos datos, el estudio estará en constante cambio.
- Para continuar con el desarrollo de la herramienta, es vital que se sigan dando proyectos orientados a obtener nuevas versiones del resultado que se alcanzó en el proyecto.

Referencias Bibliográficas

- Arcade, J., Godet, M., Meunier, F., & Roubelat, F. (2004). *Análisis estructural con el método MICMAC, y estrategia de los actores con el método Mactor*. (M. Mendieta, Trad.) Buenos Aires, Argentina: Metodología de Investigación de Futuros. Obtenido de <https://bit.ly/350Nk1D>
- Berzal, F., & Cortijo, F. (s.f.). *Curso de C#: La plataforma .NET*. Obtenido de Fernando Berzal Galiano: <https://bit.ly/33LIKUQ>
- Bootstrap. (s.f.). *Designed for everyone, everywhere*. Obtenido de <https://getbootstrap.com/docs/3.3/>
- Forciniti, L., & Elbaum, J. (2001). *La prospectiva. que es y para que sirve*. Buenos Aires: Direccion Nacional de Planificacion y Evaluacion, Gobierno de Argentina. Obtenido de <https://bit.ly/32MHn79>
- Godet, M. (2007). *Metodos de prospectiva: los programas*. Obtenido de La prospective: <https://bit.ly/2OiRP18>
- IEEE. (2008). *Especificación de Requisitos según el estándar de IEEE 830*. Obtenido de Universidad Complutense de Madrid: <https://bit.ly/2qU6M1g>
- McConnell, S. (1996). *Desarrollo y gestión de Proyectos Informaticos*. Madrid: McGraw Hill. Obtenido de <https://bit.ly/2OaI7xy>
- Ossorio, A. (2003). *Planeamiento estratégico* (Quinta ed.). Buenos Aires, Argentina: Dirección de Planeamiento y Reingeniería Organizacional. Obtenido de <https://bit.ly/2Kn4oaq>

Pinto, J. (2008). Las herramientas de la prospectiva estratégica: usos, abusos y limitaciones.

Cuadernos de Administración(40), 47-56. Obtenido de <https://bit.ly/2CIc1nJ>

Saisang, Cai; et al. (2015). *Introducción al lenguaje C# y .NET Framework*. Obtenido de Microsoft:

<https://bit.ly/2KoaWW1>

Sandoval, M. (2014). *Qué es MVC y por que es tan usado en el desarrollo Web*. Obtenido de

Michelle Torres Blog: <https://bit.ly/351zYCj>

Santamaría, J., Hernández, & Javier. (2015). Microsoft SQL Server. *SQL Server vs MySQL*, 1-6.

Obtenido de <https://bit.ly/2Qfi3UK>

Unified Modeling Language [UML]. (2005). *Introduction to OMG'S Unified Modeling Language*.

Obtenido de Object Management Group: <https://www.uml.org/what-is-uml.htm>