

**DISEÑO Y DESARROLLO SOFTWARE PARA EL MANEJO DE INFORMACIÓN Y REGISTRO DE TIEMPO  
USANDO DISPOSITIVOS RFID, DE LOS PARTICIPANTES DE LAS PRUEBAS ATLÉTICAS QUE REALIZA  
LA UNIVERSIDAD INDUSTRIAL DE SANTANDER BASADO EN PROTOTIPO UTILIZADO EN LA XXV  
CARRERA ATLÉTICA DE LA UIS**

**JUAN CAMILO FIDEL BARAJAS RUEDA**

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

**2017**

**DISEÑO Y DESARROLLO SOFTWARE PARA EL MANEJO DE INFORMACIÓN Y REGISTRO DE TIEMPO  
USANDO DISPOSITIVOS RFID, DE LOS PARTICIPANTES DE LAS PRUEBAS ATLÉTICAS QUE REALIZA  
LA UNIVERSIDAD INDUSTRIAL DE SANTANDER BASADO EN PROTOTIPO UTILIZADO EN LA XXV  
CARRERA ATLÉTICA DE LA UIS**

**JUAN CAMILO FIDEL BARAJAS RUEDA**

**Trabajo de grado presentado como requisito  
para optar al título de Ingeniero de Sistemas**

**Director**

**ENRIQUE TORRES LÓPEZ**

**Especialista en ingeniería de software**

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

**2017**

## AGRADECIMIENTOS

*Agradezco a Dios por darme la oportunidad de conocer el atletismo, darme la fortaleza para afrontar cada competencia, guiarme en el buen camino, conocer a personas maravillosas, realizar mi proyecto con la pasión de hacer mejor este deporte en Santander y llevar en alto el nombre de la Universidad Industrial de Santander.*

*A la memoria de mi padre Manuel Barajas López y mi mamá Claudia Rueda Quijano que ha sido una mujer fuerte en cada reto, mi motivación en mis estudios, paciencia y el apoyo en cada etapa de mi vida junto a mis hermanos, ayudándome a persistir en alcanzar este logro.*

*A mi director de proyecto el ingeniero Enrique Torres López, por darme la oportunidad de continuar este proyecto con el gran apoyo de la DSI.*

*Al ingeniero Humberto Ruiz Roa quien siempre se presentó dispuesto a colaborar y enseñarme.*

*Al profesor Fabio Villafrades González, quien no dudo en apoyar este proyecto, guiándome y ser un amigo, instruyendo para presentar siempre lo mejor posible cada objetivo.*

*A mis profesores, por su gran labor en cada año, mostrarme cada uno con su experiencia las oportunidades para seguir avanzando.*

*A mis amigos y compañeros, por sus consejos y gratos momentos vividos en esta etapa.*

**Juan Camilo Fidel Barajas Rueda**

## CONTENIDO

	Pág.
<b>INTRODUCCIÓN</b> .....	14
<b>1. DEFINICIÓN DEL PROBLEMA</b> .....	15
<b>2. JUSTIFICACIÓN</b> .....	16
<b>3. OBJETIVOS</b> .....	17
<b>3.1 OBJETIVO GENERAL</b> .....	17
<b>3.2 OBJETIVOS ESPECÍFICOS</b> .....	17
<b>4. ALCANCES Y LIMITACIONES</b> .....	18
<b>5. MARCO TEÓRICO</b> .....	19
<b>5.1 CONCEPTO SISTEMA DE INFORMACIÓN</b> .....	19
5.1.1 Arquitectura MVC ( MODELO – VISTA - CONTROLADOR).....	19
5.1.2 Ventajas del esquema MVC.....	19
5.1.3 Desventajas del esquema MVC. ....	20
<b>5.2 ¿QUE HACE UN SISTEMA DE INFORMACIÓN?</b> .....	21
5.2.1 Ciclo de vida de un sistema .....	21
<b>5.3 PLANEACIÓN DEL PROYECTO</b> .....	21
5.3.1 Análisis del sistema: .....	21
5.3.2 Diseño del sistema .....	21
5.3.3 Desarrollo del sistema y documentación .....	22
5.3.4 Pruebas del sistema .....	22
5.3.5 La implantación del sistema .....	22
5.3.6 El mantenimiento del sistema: .....	22
<b>5.4 BASES DE DATOS</b> .....	22
5.4.1 Informix.....	23
<b>5.5 MARCO CONCEPTUAL</b> .....	23
5.5.1 Código electrónico de producto (EPC).....	23
5.5.2 La tecnología RFID. ....	24
5.5.3 Funcionamiento general de un sistema RFID. ....	24

5.5.4 IMPRESORAS RFID .....	25
5.5.5 Ventajas de la tecnología RFID sobre el Código de Barras.....	26
5.5.6 Regulación de frecuencias.....	26
<b>5.6 TECNOLOGIAS DE DESARROLLO .....</b>	<b>27</b>
5.6.1 Diseño orientado a objetos con UML .....	27
5.6.2 Diagramas de casos de uso.....	27
5.6.3 Diagrama de clases.....	27
5.6.4 Diagrama de secuencia.....	28
<b>5.7 MODELO ENTIDAD/RELACIÓN .....</b>	<b>28</b>
<b>5.8 PLATAFORMA DE DESARROLLO .....</b>	<b>28</b>
5.8.1 JAVA EE.....	29
5.8.2 SEAM .....	29
5.8.3 Servidor de Aplicaciones JBOSS (JBOSS-AS).....	29
5.8.4 EJB (Enterprise Java Bean).....	29
5.8.5 Hibernate .....	30
5.8.6 JavaServer Faces (JSF) .....	30
5.8.7 Seguridad .....	31
<b>5.9 HISTORIA DE LA TECNOLOGÍA EN CARRERAS ATLÉTICAS: .....</b>	<b>31</b>
<b>5.10 ¿CÓMO FUNCIONA EL SISTEMA DE CHIPS? .....</b>	<b>31</b>
5.10.1 Elementos del sistema de identificación: .....	31
<b>5.11 IMAGEN DE ANTENAS.....</b>	<b>32</b>
<b>5.12 PRUEBAS ATLÉTICAS EN BUCARAMANGA.....</b>	<b>32</b>
<b>6. DISEÑO METODOLÓGICO .....</b>	<b>33</b>
<b>6.1 PROTOTIPO EVOLUTIVO .....</b>	<b>33</b>
<b>6.2 LENGUAJE DE MODELADO UNIFICADO .....</b>	<b>34</b>
6.2.1 Diagramas de UML. ....	34
6.2.2 Diagramas de casos de uso.....	35
6.2.3 Diagramas de secuencias. ....	35
<b>6.3 ESTÁNDARES DE PROGRAMACIÓN .....</b>	<b>35</b>
6.3.1 Modelo de datos. ....	35
6.3.2 Nombres de las tablas. ....	35
6.3.3 Clases. ....	36

6.3.4 Páginas XHTML .....	36
6.3.5 Organización de Directorios. ....	36
<b>7. DESARROLLO DE LA HERRAMIENTA, ADMINISTRACIÓN Y MANTENIMIENTO .....</b>	<b>37</b>
<b>7.1 PRIMER PROTOTIPO .....</b>	<b>37</b>
<b>7.2 PROTOTIPO FINAL.....</b>	<b>39</b>
7.2.1 Esquema de seguridad Universidad Industrial de Santander .....	40
7.2.2 Entorno de Navegación .....	42
7.2.3 Entorno de Control de Datos.....	42
7.2.4 Auditoría .....	42
7.2.5 Interfaz Gráfica.....	43
7.2.6 Requerimientos de objetivo. ....	46
7.2.7 Requerimientos del sistema de información.....	47
<b>7.3 DIAGRAMAS DE CASOS DE USO. ....</b>	<b>48</b>
7.3.1 Documentación de casos de USO del sistema. ....	51
7.3.2 Diseño y Análisis.....	51
7.3.2.3 Modelo de Procesos del Sistema.....	52
7.3.2.4 Implementación, Implantación y Pruebas Generales. ....	53
<b>7.4 MANTENIMIENTO Y ADMINISTRACIÓN .....</b>	<b>54</b>
7.4.1 Actividades de Soporte a Usuarios. ....	54
<b>8. PRUEBAS DEL SISTEMA.....</b>	<b>55</b>
<b>8.1 PRUEBAS DE VERIFICACIÓN.....</b>	<b>55</b>
8.1.1 Pruebas por componente.....	55
<b>8.2 PRUEBAS DE INTEGRACIÓN .....</b>	<b>55</b>
<b>8.3 PRUEBAS DE VALIDACIÓN .....</b>	<b>55</b>
<b>9. CONCLUSIONES.....</b>	<b>56</b>
<b>10. RECOMENDACIONES .....</b>	<b>57</b>
<b>BIBLIOGRAFÍA.....</b>	<b>58</b>

## LISTA DE TABLAS

	Pág.
Tabla 1. Descripción de las Entidades .....	52

## LISTA DE FIGURAS

	<b>Pág.</b>
Figura 1. Modelo vista controlador (MVC) .....	20
Figura 2. Descripción general de un sistema RFID .....	24
Figura 3. Stack de JBoss .....	28
Figura 4. Punto de control META - 2015 .....	32
Figura 5. Prototipo Evolutivo .....	33
Figura 6. Página inicial – Resultados .....	37
Figura 7. Crear categoría .....	38
Figura 8. Inscripción .....	38
Figura 9. Página inicial .....	43
Figura 10. Administrar Prueba .....	44
Figura 11. Administrar categoría .....	44
Figura 12. Inscripción .....	45
Figura 13. Resultados .....	45
Figura 14. Diagrama de Casos de Uso: Registrar y consultar .....	48
Figura 15. Diagrama de Casos de USO: Gestionar Prueba .....	49
Figura 16. Diagrama de Casos de USO: Cargar y generar resultados .....	49
Figura 17. Diagrama de Casos de Uso: Generar número .....	50
Figura 18. Diagrama de Casos de Uso completo del sistema .....	50
Figura 19. Diagrama E/R: Sistema completo .....	51
Figura 20. Diagrama de Secuencia: Inscripción .....	53

## LISTA DE ANEXOS

	<b>Pág.</b>
Anexo A. Tecnologías de desarrollo de páginas web dinámicas.....	61
Anexo B. Bases de datos .....	62
Anexo C. Leguaje de modelado unificado (UML) .....	65
Anexo D. Documentación de casos de uso del sistema.....	67
Anexo E. Diagramas de secuencia .....	69

## RESUMEN

**TÍTULO:** DISEÑO Y DESARROLLO SOFTWARE PARA EL MANEJO DE INFORMACIÓN Y REGISTRO DE TIEMPO USANDO DISPOSITIVOS RFID, DE LOS PARTICIPANTES DE LAS PRUEBAS ATLÉTICAS QUE REALIZA LA UNIVERSIDAD INDUSTRIAL DE SANTANDER BASADO EN PROTOTIPO UTILIZADO EN LA XXV CARRERA ATLÉTICA DE LA UIS\*.

**AUTOR:** JUAN CAMILO FIDEL BARAJAS RUEDA\*\*

**PALABRAS CLAVE:** Dispositivos RFID, Prueba atlética, Inscripción.

### DESCRIPCIÓN

CARRERA ATLÉTICA DE LA UIS, Bucaramanga, Colombia (1988), 26 ediciones celebrando el aniversario de la UIS “caminando, trotando y corriendo”. Carrera atlética de 5 km, considerada la carrera más antigua de Santander y “Universitaria” de Colombia, es un evento que ha logrado mantener su continuidad. Inició corriéndose en el mes de agosto, pero se institucionalizó, el primer domingo de marzo, dentro del programa de actividades de conmemoración del aniversario de la Universidad Industrial de Santander, con una participación superior a 3800 corredores en las diferentes categorías en el año 2016.

La primera Carrera Atlética de la UIS se realizó dentro del marco de la celebración de los 40 años de labores académicas de la Universidad Industrial de Santander. El Departamento de Cultura Física junto con los centros de estudio de las carreras de ingeniería Química, Eléctrica y Mecánica, programaron la Maratón “40 años UIS”.

Con el fin de atender las necesidades que se presentan en el registro de tiempos de la prueba atlética que organiza la Universidad Industrial de Santander se hizo reingeniería al prototipo utilizado en la versión 25 de la misma prueba, implementando módulos de administración para las categorías en la prueba, las inscripciones y consulta de resultados en línea para los participantes.

Apoyado en dispositivos de identificación por radiofrecuencias (RFID) para obtener la información de medición de tiempos automatizando este proceso que se realizaba de forma manual y desarrollando un prototipo funcional para la versión XVII de la carrera atlética de la UIS.

-----  
\* Trabajo de grado: Modalidad Trabajo de Investigación.

\*\*Facultad de Ingenierías Fisicomecánicas. Escuela de Ingeniería de Sistemas e Informática.

Director: Enrique Torres López.

## ABSTRACT

**TITLE: DESIGN AND DEVELOPMENT SOFTWARE FOR THE MANAGEMENT OF INFORMATION AND TIME REGISTRY USING RFID DEVICES, OF PARTICIPANTS OF THE ATHLETIC TESTS PERFORMED BY THE INDUSTRIAL UNIVERSITY OF SANTANDER BASED ON PROTOTYPE USED IN THE XXV ATHLETIC SCHOOL OF THE UIS \*.**

**AUTHOR: JUAN CAMILO FIDEL BARAJAS RUEDA \*\***

**KEY WORDS: RFID Devices, Athletic Testing, Inscription.**

### DESCRIPTION

**CAREER ATHLETIC OF THE UIS, Bucaramanga, Colombia (1988), 26 editions celebrating the anniversary of the UIS "walking, trotting and running". The 5 km athletic race, considered the oldest race in Santander and "University" in Colombia, is an event that has managed to maintain its continuity. It started running in August, but it was institutionalized on the first Sunday of March, within the program of activities commemorating the anniversary of the Industrial University of Santander, with a participation of more than 3800 runners in the different categories in 2016.**

**The first Athletic Race of the UIS was realized within the framework of the celebration of the 40 years of academic work of the Industrial University of Santander. The Department of Physical Culture together with the study centers of the chemical engineering, electrical and mechanical engineering, programmed the Marathon "40 years UIS".**

**In order to meet the needs presented in the time record of the athletic test organized by the Industrial University of Santander, the prototype used in version 25 of the same test was reengineered, implementing management modules for the categories in the Test, enrollment and online query results for participants.**

**Supported by Radio Frequency Identification (RFID) devices to obtain time measurement information by automating this process that was performed manually and developing a functional prototype for the XVII version of the UIS athletic career.**

-----  
\* Degree Work: Research Work Mode.

\*\* Faculty of Physicomechanical Engineering. School of Systems Engineering and Informatics.  
Director: Enrique Torres López.

## **INTRODUCCIÓN**

Debido al crecimiento de la participación de atletas en la carrera atlética de la Universidad Industrial de Santander, se hace necesario una mejora en el control y registro de tiempos. Es así, como surge la idea de crear una herramienta que en primer lugar facilite la interacción entre los interesados en el evento, participantes y organizadores de la prueba, desarrollando un prototipo mínimo funcional para la versión XXV de la carrera atlética de la Universidad Industrial de Santander llevado a cabo en el año 2015 en coordinación con el departamento de deportes y la administración general de la Universidad Industrial de Santander.

Dado a la inversión y la acogida del prototipo presentado en el año 2015, se propone por parte de la División de Servicios Informáticos (DSI), el diseño y desarrollo de un sistema de información que cumpla con los estándares y funciones de esta división.

La División de Servicios de Información de la Universidad Industrial de Santander tiene como fin la administración y el desarrollo de la tecnología de la información en los ámbitos académico y administrativo, definiendo las políticas necesarias para la gestión de la infraestructura de servicios informáticos institucionales, garantizando el adecuado uso de los recursos e impulsando la Innovación tecnológica de la Universidad.

Para llevar a cabo esta labor y proporcionar un sistema de información con contenido dinámico se ha contado con software licenciado de la DSI como JBoss Developer Studio, trabajando en lenguajes de programación JAVA , PHP , XML , acceso a servidores web , contando además con dispositivos de medición como antenas , tags RFID e infraestructura por parte del Departamento de Deportes de la Universidad Industrial de Santander.

En este documento se presenta el soporte teórico, metodológico y técnico del desarrollo de los servicios previamente mencionados, así como de las labores de administración y registro de usuarios desempeñadas durante el desarrollo del proyecto.

## **1. DEFINICIÓN DEL PROBLEMA**

Se ha venido acelerando la velocidad de cambio en la mayoría de las organizaciones, de allí que éstas necesitan ahora más información como soporte a la toma de decisiones. Por esta razón, el desarrollo de los sistemas de información viene jugando un papel importante y cada vez más preponderante para poder competir y subsistir en el medio, de esta forma se pretende obtener los datos que requieren la toma de decisiones de forma automatizada y ágil.

El sistema de medición sin utilizar medios electrónicos de la carrera atlética de la Universidad Industrial de Santander no es fiable, es laborioso y muy complejo logísticamente, que involucra a cientos de voluntarios. El desorden y la larga fila de corredores en la meta generan muchos inconvenientes a la hora de tomar el tiempo de llegada.

Por ende, se desea desarrollar un sistema de información a la medida que lleve el registro, estadísticas, históricos y el control de llegada de los corredores de la carrera atlética de la Universidad industrial de Santander haciendo uso de la tecnología RFID. Estableciendo para este proyecto un punto control, ubicado en la línea de meta conformado por una alfombra modular de lectura.

En 1995, la maratón de Rotterdam fue la primera maratón internacional que utilizó plenamente el sistema ChampionChip. La Maratón de Nueva York lo empezó a usar en 1999 y Boston en 1996. En Colombia, la Media Maratón de Bogotá introdujo su uso en el año 2006. Aunque existen otros sistemas utilizados en el mundo como el Chip Timing, D-tag y b-tag, entre otros, el chip sigue siendo uno de los más usados.

## **2. JUSTIFICACIÓN**

Se ha venido acelerando la velocidad de cambio del medio de casi todas las organizaciones, de allí que éstas necesitan ahora más información como soporte a la toma de decisiones. Es por eso que, los desarrollos de los sistemas de información vienen jugando un papel importante y cada vez más preponderante para poder competir y subsistir en el medio, de esta forma se pretende obtener los datos que requieren la toma de decisiones de forma automatizada y ágil.

El sistema de medición de la carrera atlética de la Universidad no es fiable, es laborioso y muy complejo logísticamente, que involucra a cientos de voluntarios. El gran número de participantes en las pruebas y la carencia de medios electrónicos e informáticos como apoyo para su administración generan muchos inconvenientes durante su realización.

Por ende, se desea desarrollar un sistema de información a la medida que lleve el registro, estadísticas, históricos y el control de llegada de los corredores de la carrera atlética de la Universidad Industrial de Santander con la tecnología RFID.

Como referencia del uso de chips se tiene: En 1995, la maratón de Rotterdam fue el primer maratón internacional que utilizó plenamente el sistema ChampionChip. La Maratón de Nueva York lo empezó a usar en 1999 y Boston en 1996, en Colombia la Media Maratón de Bogotá introdujo su uso en el año 2006, además el prototipo que se utilizó en el año 2015 para la versión XXV de la carrera atlética de la Universidad Industrial de Santander, implementos de medición disponibles de este evento en el inventario de la Universidad Industrial de Santander.

### **3. OBJETIVOS**

#### **3.1 OBJETIVO GENERAL**

Análisis del problema, diseñar e implementar el software que permita la administración y control de las pruebas atléticas que realiza la Universidad Industrial de Santander.

#### **3.2 OBJETIVOS ESPECÍFICOS**

1. Diseñar diagramas de modelos UML, basados en el estándar de desarrollo de la División de Servicios de Información DSI.
2. Desarrollar los módulos de inscripción, administración de categorías, generación de resultados y su consulta.
3. Garantizar la identificación de cada competidor y el funcionamiento del sistema de identificación por radiofrecuencias (RFID) generando informe de inscritos y participantes.
4. Desarrollar prototipo funcional para la XXVII Carrera Atlética de la Universidad Industrial de Santander para el año 2017.

#### **4. ALCANCES Y LIMITACIONES**

El módulo de inscripción permitirá realizar registro en línea de los participantes (estudiantes, egresados, trabajadores, profesores, comunidad en general) de la carrera atlética de la Universidad industrial de Santander, a su vez, da la opción de consultar, editar y eliminar registros, según el rol asignado dentro del sistema.

La administración de categorías, un módulo para el usuario administrador del evento donde se podrán crear, editar y eliminar categorías asignadas a una prueba.

Se desarrollan módulos administrativos de la prueba, pero también módulos públicos como la inscripción y el módulo de consulta de resultados donde se desplegarán los registros con criterios de búsqueda como identificación del participante, categoría, prueba, año y versión del evento.

Este sistema de información estará disponible en un servidor web de la Universidad Industrial de Santander, en donde se podrá acceder en línea.

El número de identificación de los competidores, asignado con el dispositivos RFID propuesto, para la competencia, será responsabilidad del comité designado para el evento, repartición y asignación de este material.

## **5. MARCO TEÓRICO**

### **5.1 CONCEPTO SISTEMA DE INFORMACIÓN**

Un sistema de información es un conjunto de personas, datos y procedimientos que funcionan en conjunto. Significa que los variados componentes buscan un objetivo común para apoyar las actividades de la organización. Estas incluyen las operaciones diarias de las empresas, la comunicación de los datos e informes, la administración de las actividades y la toma de decisiones. Se define como una arquitectura distribuida que permite a los usuarios finales obtener acceso a la información en forma transparente aún en entornos multiplataforma.

#### **5.1.1 Arquitectura MVC ( MODELO – VISTA - CONTROLADOR).**

MVC, Model - View - Controller o Modelo - Vista - Controlador un patrón de diseño de software para programación que propone separar el código de los programas por sus diferentes responsabilidades.

El modelo MVC le ayuda a crear aplicaciones que separan los diferentes aspectos de la aplicación (lógica de entrada, lógica de negocios y lógica de la interfaz de usuario), a la vez que proporciona un vago acoplamiento entre estos elementos. El modelo especifica dónde se debería encontrar cada tipo de lógica en la aplicación. La lógica de la interfaz de usuario pertenece a la vista. La lógica de entrada pertenece al controlador. La lógica de negocios pertenece al modelo. Esta separación le ayuda a administrar la complejidad al compilar una aplicación, ya que le permite centrarse en cada momento en un único aspecto de la implementación. Por ejemplo, se puede centrar en la vista sin estar condicionado por la lógica de negocios.

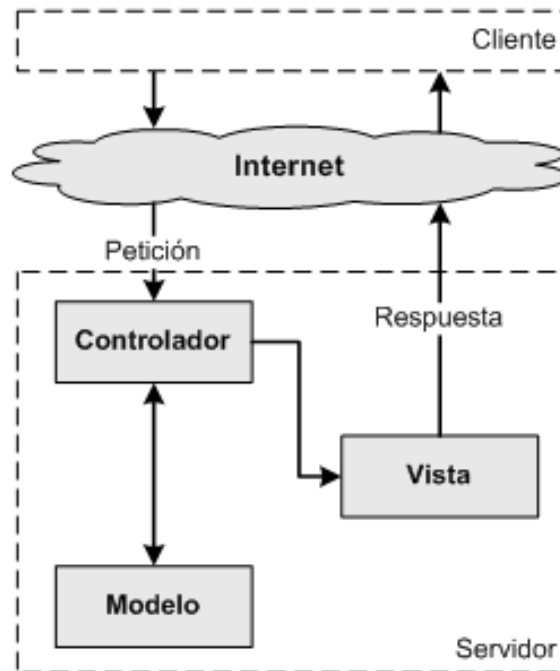
#### **5.1.2 Ventajas del esquema MVC**

1. La implementación se realiza de forma modular.
2. 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.
3. 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.
4. Las modificaciones a las vistas no afectan al modelo de dominio, simplemente se modifica la representación de la información, no su tratamiento.
5. 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.

### 5.1.3 Desventajas del esquema MVC.

1. 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 fácil de mantener, extensible y modificable que una aplicación que no lo implementa.
2. 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.
3. 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.

Figura 1. Modelo vista controlador (MVC).



Fuente: MVC. [En línea].

<http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/122> [Noviembre de 2016]

## **5.2 ¿QUE HACE UN SISTEMA DE INFORMACIÓN?**

Un sistema de información ejecuta algunas actividades generales. En primer término, recibe datos de fuentes internas o externas de la empresa como elementos de entrada, después transforma los datos para producir la información para el futuro usuario, tal vez sea un gerente, un administrador o un miembro del cuerpo directivo.

Un sistema de información es un conjunto de elementos orientados al tratamiento y administración de datos e información, organizados y listos para su uso posterior, generados para cubrir una necesidad o un objetivo.

### **5.2.1 Ciclo de vida de un sistema**

Se conocen como las etapas del ciclo de vida clásico de los sistemas de información:

1. Planeación del proyecto
2. Análisis del sistema
3. Diseño del sistema
4. Desarrollo del sistema y/o documentación
5. Pruebas del sistema
6. Implantación del sistema
7. Mantenimiento del sistema

## **5.3 PLANEACIÓN DEL PROYECTO**

Consiste en la definición del problema y la investigación que se hace acerca de las necesidades de los usuarios respecto del sistema con el fin de definir sus objetivos, alcance y justificación.

### **5.3.1 Análisis del sistema:**

Una vez desarrollada la planeación del proyecto en donde se definió que es lo que se quiere hacer, el cronograma de trabajo y el presupuesto, se procede con la etapa de análisis del sistema. En esta etapa se definen los requerimientos de información del mismo, los procesos que se piensan sistematizar, el flujo de los datos y la manera como se van a almacenar datos e información general.

### **5.3.2 Diseño del sistema**

El diseño de un sistema de información es el plan o modelo general del sistema, al igual que los planos de una edificación, el diseño consiste en todas las especificaciones que dan al sistema su forma y estructura; el diseñador de sistemas detallará e identificará las funciones a ser implementadas. Dichas especificaciones deben contemplar todos los componentes de administración, organización y tecnología del sistema solución.

### **5.3.3 Desarrollo del sistema y documentación**

El desarrollo de un sistema consiste en la traducción del diseño de dicho sistema a una forma entendible para la máquina, lo cual se logra mediante la generación de código fuente, que se puede obtener en muchos casos de manera automatizada (usando herramientas CASE) y en todos los casos usando el talento de los programadores que elaboran dicho código para que satisfagan las especificaciones del diseño.

### **5.3.4 Pruebas del sistema**

Una vez desarrollado un sistema de información se podría esperar que ya está listo para entrar en operación. No obstante, es absolutamente necesario efectuarle pruebas, pues es muy extraño el caso de que un sistema quede completamente listo y a satisfacción del usuario después de finalizada su etapa de desarrollo.

### **5.3.5 La implantación del sistema**

Una vez se ha definido que el producto está terminado, probado y para entrega, se procede con la fase de implantación donde se sube el sistema al servidor sugerido y su respectiva base de datos.

### **5.3.6 El mantenimiento del sistema:**

Esta es la fase más extensa en el tiempo y más costosa del ciclo de vida de los sistemas de información. Se puede decir que la fase de mantenimiento existirá mientras el sistema tenga vida. Por otra parte, es una fase que implica la dedicación de capacidad profesional durante todo el tiempo de vida de la aplicación para hacer correcciones y ajustes de acuerdo con los cambios del medio.

## **5.4 BASES DE DATOS**

Una base de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso, con una redundancia controlada y una estructura que refleja las interrelaciones y restricciones existentes en el mundo real. En la base de datos se almacena información considerada necesaria para una determinada organización o negocio.

Dentro de las bases de datos existen diferentes modelos, entre los cuales se encuentran bases de datos jerárquicas, de red y relacionales. Así mismo para tener conexión a estos modelos es necesario utilizar conectores. Los más comunes son ODBC, MDB y JDBC. (Ver Anexo B).

Existen diferentes manejadores de bases de datos como Informix, MySQL, ORACLE, FoxPro, Microsoft Access o PowerBuilder, en el desarrollo de los sistemas de información de la División de Servicios de Información DSI se utiliza Informix.

### 5.4.1 Informix

Informix es un DBMS actualmente comercializada y soportada por IBM. Fue adquirido en el 2001 a una compañía llamada Informix. Ha sido uno de los más importantes sistemas de base de datos a lo largo de la historia y actualmente es quizá uno de los más reconocidos. Este sistema ha sido considerado como uno de los más eficientes, de bajo costo y confiables del mercado.

El modelo que utiliza Informix es un modelo que soporta bases de datos con paralelismo horizontal y vertical. La forma en la que almacena la información le permite procesar los datos de forma paralela (XPS (for eXtended Parallel Server)). Esto le ha dado la ventaja de ser líder de mercado.

Algunos ejemplos de aplicaciones para Informix es el uso de base de datos de gran capacidad. Actualmente de usa Informix dentro de la nueva era del Cloud Computing. Las características de Informix - de alto rendimiento, confiabilidad, facilidad de uso y bajo costo, la convierten en un excelente complemento para estas aplicaciones en la “nube”.

También la creación de bases de datos para nuevas pequeñas y medianas empresas que les permitirá tener todos los beneficios de una base datos. Otras aplicaciones que ofrece Informix es en el uso de telecomunicaciones, organización de tareas en una empresa, organización de tiempos, etc.

## 5.5 MARCO CONCEPTUAL

### 5.5.1 Código electrónico de producto (EPC)

*“El EPC es un conjunto de números que identifica única e inequívocamente a un artículo en la cadena de suministro”* (Observatorio Nacional de las Telecomunicaciones y de la Sociedad de la Información ONTSI, 2009, p 44). Es utilizado en las empresas para conformar un sistema de seguimiento e identificación de mercancías en tiempo real. Sistema que combinado con la tecnología RFID crea información precisa y efectiva de lo relevante a cualquier producto.

El número o código de identificación es almacenado en un chip o tag RFID, unido a una antena que emite una señal por medio de radio frecuencia. Esta señal lleva consigo el código que luego es leído por un lector RFID y posteriormente procesado por medio de un software especializado. Al igual que el código de barras, el código electrónico de producto está dividido en una sección de dígitos que identifican al fabricante y otras secciones que identifican al producto y su versión e información adicional sobre el producto.

Las ventajas más notorias de EPC frente al código de barras son su mayor capacidad de almacenamiento, debido a esto se puede tener mucha más información relacionada con el producto; y la manera como se obtienen los datos almacenados, por lo que no necesita lectura directa del código si no que este es recibido por radio frecuencia. Por estas ventajas y otras razones, EPC es considerado la evolución del código de barras.

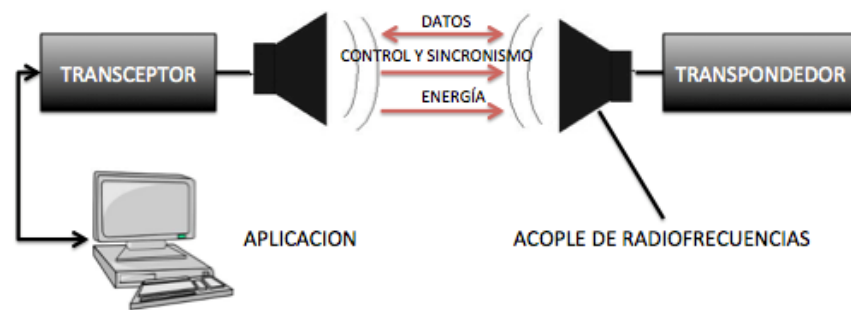
### 5.5.2 La tecnología RFID.

La tecnología de identificación por radio frecuencia o RFID, consiste en la identificación de determinados objetos a distancia sin necesidad de ningún contacto directo o visual. La variedad de aplicaciones en campos como el: empresarial, ganadero, agrícola, médico, entre muchos otros, ha hecho que esta tecnología tenga un rápido desarrollo e implementación en los últimos años. Landt, J (2001).

### 5.5.3 Funcionamiento general de un sistema RFID.

El modus operandi de un sistema RFID, está enfocado al almacenamiento y recuperación de datos de forma remota a través de ondas de radio. La operación del sistema involucra un transpondedor o etiqueta RFID; dispositivo que contiene el código de identificación, y transceptores o lectores RFID; dispositivos encargados principalmente de leer el código de las etiquetas. El esquema general se observa en la figura 2.

Figura 2. Descripción general de un sistema RFID



Fuente: Duran, V, Sandoval, A, Ariza, J, 2004

Generalmente, un transceptor está conformado por un módulo que trabaja a una frecuencia determinada para transmitir y recibir datos, una unidad de control encargada de sincronizar las señales enviadas y recibidas hacia y desde la etiqueta, y un elemento que se acopla a la etiqueta para comunicar los dos dispositivos. Suelen encontrarse además, con una conexión o interfaz de comunicación que les permite intercambiar información con un sistema externo capaz de procesarla y evocarla a una aplicación específica que utilice la información captada por estos módulos.

El transpondedor por su parte, es el dispositivo en el cual se encuentran almacenados los datos de identificación o código de un sistema RFID que recibe y automáticamente responde a una señal entrante. Está compuesto por un microchip, que es donde se guarda la información de caracterización y es el que permite que este elemento sea usado para labores de control y supervisión, y por un elemento de acople de radiofrecuencia tipo bobina o antena, que permite la comunicación inalámbrica con el transceptor al suministrarle la potencia necesaria para activarlo así como las señales de control y sincronismo. Los transpondedores, ven limitado su funcionamiento a un rango impuesto por el transceptor y mientras se encuentren fuera de este, su comportamiento

será netamente pasivo, activándose solamente cuando estén dentro del rango de respuesta del transceptor.

Es muy importante tener en cuenta que la función de otros servicios de radio no debe verse afectada por ningún dispositivo RFID, por lo cual, la gama de frecuencias de trabajo de estos sistemas se ve restringida considerablemente a frecuencias para aplicaciones industriales, científicas o médicas o para dispositivos de corto alcance conocidas como ISM<sup>1</sup> o SRD<sup>2</sup>.

Los transpondedores pueden ser de dos tipos: activos o pasivos. Ejemplos de un transponder pasivo son las etiquetas magnéticas, como las presentes en las tarjetas de crédito y en los elementos almacenados. Un transponder pasivo debe ser usado con un sensor activo que decodifique y comunique los datos contenidos en este. Una unidad de este tipo, utilizando tecnología RFID puede ser de un tamaño pequeño y su información puede ser leída a varios metros de distancia.

Los transponders activos simples son implementados en ubicación, identificación y en sistemas de navegación para aeronaves comerciales o privadas. Por ejemplo, un dispositivo RFID que transmite una señal codificada cuando recibe desde un punto de control un requerimiento. La señal de salida del transponder es rastreada de manera que siempre se detecta su presencia. Las frecuencias del reader y el transponder son pre asignadas y las distancia manejadas por este tipo de equipos pueden llegar a ser de miles de kilómetros.

El campo de frecuencias genera una corriente eléctrica sobre la bobina de recepción del dispositivo, esta señal es rectificadora y de esta manera se alimenta el circuito. Cuando esta llega a ser suficiente, el circuito transmite sus datos, el reader, entonces detecta los datos transmitidos por la tarjeta como una perturbación del propio nivel de la señal.

Para el caso de los transponders pasivos la señal recibida por el reader desde la tarjeta debe estar a un nivel mínimo de -60dB por debajo de la portadora de transmisión para ser leída efectivamente, las antenas para frecuencias UHF preparadas para capturar tags hasta 14 metros de distancia, pudiendo regular el alcance en función de las prestaciones del lector y del tag leído.

#### **5.5.4 IMPRESORAS RFID**

Existen dispositivos especializados y dedicados para escribir la información sobre las etiquetas RFID, estos son los programadores. Esta acción de escritura denominada programación puede realizarse una o varias veces para etiquetas de solo lectura o lectura/escritura respectivamente, y por lo general se lleva a cabo fuera del proceso.

Las impresoras RFID, algunas de estas son de tipo lectura/escritura, se encargan de codificar las etiquetas o tags e imprimir en su superficie los datos legibles como el código de barras u otra información escrita. Los materiales de las etiquetas para este proceso deben ser flexibles y permitir la impresión en su exterior. Estas impresoras incluyen un verificador que comprueba que los datos almacenados en el *chip* de la etiqueta sean válidos, recibe sus comandos y una vez que la etiqueta se encuentra en posición, el lector realiza una verificación del circuito distinguiendo las etiquetas

---

<sup>1</sup> Industrial-Scientific-Medical

<sup>2</sup> Short Range Devices

dañadas, expulsándolas y marcándolas con una leyenda que designa el fabricante. El módulo RFID de la impresora codifica el tag a través de los comandos del protocolo de comunicación.

### **5.5.5 Ventajas de la tecnología RFID sobre el Código de Barras**

1. No requiere una línea de visión.
2. No requiere de intervención humana (Ideal para automatizar).
3. Distancias de lectura de 1 a 10m.
4. Lectura simultánea de múltiples artículos (protocolo anticolidión).
5. Hasta 500 lecturas por minuto (más de 5 veces más rápido que un código de barras).
6. No le afectan los ambientes sucios.
7. Capacidad de lectura y escritura.

### **5.5.6 Regulación de frecuencias**

No hay ninguna corporación pública global que gobierne las frecuencias usadas para RFID. En principio, cada país puede fijar sus propias reglas.

Las principales corporaciones que gobiernan la asignación de las frecuencias para RFID son:

1. EE.UU.: FCC (Federal Communications Commission)
2. Canadá: DOC (Departamento de la Comunicación)
3. Europa: ERO, CEPT, ETSI y administraciones nacionales. Obsérvese que las administraciones nacionales tienen que ratificar el uso de una frecuencia específica antes de que pueda ser utilizada en ese país
4. Japón: MPHPT (Ministry of Public Management, Home Affairs, Post and Telecommunication)
5. China: Ministerio de la Industria de Información

Las etiquetas RFID de baja frecuencia (LF: 125 - 134 kHz y 140 - 148.5 kHz) y de alta frecuencia (HF: 13.56 MHz) se pueden utilizar de forma global sin necesidad de licencia. La frecuencia ultra alta (UHF: 868 - 928 MHz) no puede ser utilizada de forma global, ya que no hay un único estándar global. En Norteamérica, la frecuencia ultra elevada se puede utilizar sin licencia para frecuencias entre 908 - 928 MHz, pero hay restricciones en la energía de transmisión.

Existen regulaciones adicionales relacionadas con la salud y condiciones ambientales. Por ejemplo, en Europa, la regulación Waste Electrical and Electronic Equipment ("Equipos eléctricos y electrónicos inútiles"), no permite que se desechen las etiquetas RFID. Esto significa que las etiquetas RFID que estén en cajas de cartón deben ser quitadas antes de deshacerse de ellas.

## **5.6 TECNOLOGIAS DE DESARROLLO**

### **5.6.1 Diseño orientado a objetos con UML**

El lenguaje Unificado de Modelado (UML por sus siglas en inglés) es un lenguaje usado en el diseño de aplicaciones software que ayuda a especificar, visualizar, y documentar modelos de sistemas, incluyendo su estructura y diseño, de manera que satisfaga todos los requerimientos extraídos.

El modelado, es el diseño de las aplicaciones software antes de empezar con la codificación, es una parte fundamental en los grandes proyectos, y también es muy útil en el buen desarrollo de medianos y pequeños sistemas.

Para la realización óptima de estos modelos se trabaja mediante conjuntos de diagramas que permiten trabajar a altos niveles de abstracción.

La División de Servicios de Información ha establecido dentro de sus estándares de desarrollo de software la elaboración de los diagramas de casos de uso, de clases y de secuencia.

### **5.6.2 Diagramas de casos de uso**

El modelo de casos de uso especifica la funcionalidad que el sistema ha de ofrecer desde la perspectiva de los usuarios y lo que el sistema ha de realizar para satisfacer las peticiones de estos usuarios. El diagrama de casos de uso visualiza el comportamiento de un sistema, de un subsistema, o de una clase, tal como se muestra a un usuario.

Este modelo utiliza tres elementos básicos:

1. **Actor:** Es un tipo o categoría de usuarios; señala cuando un usuario realiza una tarea con el sistema. Un actor define los diferentes papeles que puede desempeñar un usuario.
2. **Casos de Uso:** Representa la secuencia de transacciones que se realizan en un diálogo con el sistema, y que se encuentran relacionadas por su comportamiento. La colección de todos los casos de uso relacionados con un sistema especifica todas las maneras en que se puede utilizar el sistema.
3. **Relaciones:** Para identificar la comunicación existente entre los actores, los casos de uso, y actores-caso de uso se utilizan varios tipos de relaciones las cuales pueden ser, según correspondan, relaciones de asociación, generalización, extensión e inclusión.

### **5.6.3 Diagrama de clases**

El diagrama de clases modela la vista estática del sistema, ya que no describe el comportamiento del sistema en función del tiempo. Un diagrama de clases recoge tanto los conceptos del dominio de la aplicación como aquellos que forman parte de la implementación de la aplicación.

Los elementos principales son:

1. **Clases:** Una clase es una descripción de un conjunto de objetos con las mismas propiedades, el mismo comportamiento, las mismas relaciones con otros objetos y la misma semántica. Una clase puede representar un concepto del mundo real o un concepto de implementación del sistema modelado.

2. Relaciones: Se definen como una abstracción de un conjunto de interrelaciones semánticas puntuales que se dan sistemáticamente entre diferentes tipos de objetos; las relaciones que se presentan entre las clases pueden ser de asociación, agregación o herencia, ésta última se puede encontrar mediante generalización o especialización.

#### 5.6.4 Diagrama de secuencia.

Los diagramas de secuencia son un tipo de diagramas de interacción. Se utilizan especialmente cuando se trata de sistemas en tiempo real, pueden ser excesivamente grandes si intervienen muchas clases.

Una clase en el diagrama de secuencia se representa con un rectángulo en cuyo interior aparece el nombre de la clase. Todas las clases involucradas en el diagrama de secuencia que se esté contrayendo se colocan una al lado de la otra. Debajo de cada clase se coloca una línea vertical. Entre las clases se pueden enviar mensajes, si se está en fase de análisis, o llamadas a métodos, si se está en fase de diseño.

### 5.7 MODELO ENTIDAD/RELACIÓN

El modelo Entidad/Relación (E/R) consiste en un conjunto de conceptos, reglas y notaciones que se utiliza en la fase de análisis de requisitos. El modelo E/R es el resultado del proceso de abstracción apoyado en un modelo de datos con el fin de describir la estructura de la base de datos de forma que se represente fielmente la parte del mundo real que se quiere almacenar.

### 5.8 PLATAFORMA DE DESARROLLO

La plataforma utilizada por la División de Servicios de Información es java EE.  
Los desarrollos en Java se realizan sobre el stack de JBoss: JBoss AS (Servidor de aplicaciones).

Figura 3. Stack de JBoss



Fuente: [en línea] <http://www.bdftech.com/tecnologia/java/> [citado Junio de 2016].

### **5.8.1 JAVA EE**

Los desarrolladores actualmente reconocen la necesidad de presentar aplicaciones distribuidas, transaccionales, y portables, que exploten la seguridad, velocidad y fiabilidad de las tecnologías del lado del servidor. El diseño de las aplicaciones en las empresas debe estar orientado a producir más con menos recursos. A través de Java Enterprise Edition, el desarrollo de estas aplicaciones se concibe de forma fácil y rápida, cumpliendo con las demandas de los desarrolladores.

El objetivo más importante de la plataforma Java EE es simplificar el desarrollo al proporcionar una base común para los diversos tipos de componentes en la plataforma Java EE. Los desarrolladores se benefician de las mejoras de productividad con más anotaciones y menos de configuración XML, más objetos Plain Old Java (POJO), y envases simplificado.

Actualmente la versión utilizada en sistemas de producción de la DSI es JAVA EE5 y se encuentra en proceso de actualización ya que desde el año 2015 fue lanzada la versión JAVA EE8, para este proyecto la DSI sugiere trabajar con la versión JAVA EE5, siendo SEAM el framework utilizado en esta versión.

JAVA EE 5 es una versión liberada de la plataforma de programación JAVA EE, cuyo objetivo es proporcionar al desarrollador con un potente conjunto de APIs y reducir tanto el tiempo de desarrollo, como la complejidad de las aplicaciones, aumentando su rendimiento.

### **5.8.2 SEAM**

SEAM fue diseñado con el propósito de eliminar la complejidad en la arquitectura y en los niveles API de desarrollo. Esto proporciona al desarrollador la capacidad de ensamblar cómodamente complejas aplicaciones Web mediante simples clases anotadas de java, un conjunto rico en componentes de interfaz de usuario, y algo de XML.

### **5.8.3 Servidor de Aplicaciones JBOSS (JBOSS-AS)**

**JBOSS** es un servidor de aplicaciones java creado por la comunidad de desarrolladores de código abierto; es una plataforma para la producción de aplicaciones empresariales, Web y portables con el fin de abarcar todo el repertorio ofrecido en JAVA EE 5, así como los servicios adicionales incluyendo: clustering, caching y persistencia.

### **5.8.4 EJB (Enterprise Java Bean)**

Es un componente que debe ejecutarse de un contenedor de EJBs y se diferencia bastante de un JavaBean normal. Un JavaBean es un objeto Java al cual accedemos de forma directa desde nuestro programa. Sin embargo, un EJB es un componente al cual no podemos acceder de una forma tan directa y siempre accedemos a través de algún tipo de intermediario. Este intermediario aportará una serie de servicios definidos por los estándares en los cuales el EJB se puede apoyar. Se va a construir en este post un EJB básico concretamente un EJB de Sesión que son los más habituales. Los EJB de sesión disponen de dos proxis (intermediarios) a través de los cuales se acceden a ellos.

1. Proxy Local: Es el intermediario que permite un acceso al EJB desde la misma máquina virtual.
2. Proxy Remoto: Es el intermediario que permite el acceso al EJB desde una máquina virtual remota.

Estos proxis son los encargados de dar acceso al EJB a todos los servicios adicionales que soporta el EJB Container como son Transaccionalidad, Seguridad etc. Para construir un EJB de Sesión se deberán definir las interfaces de acceso local y remoto en los cuales los proxis se apoyarán.

### **5.8.5 Hibernate**

Es una herramienta de mapeo objeto-relacional (ORM) para la plataforma Java (y disponible también para .Net con el nombre de NHibernate) que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) o anotaciones en los beans de las entidades que permiten establecer estas relaciones.

Hibernate es software libre, distribuido bajo los términos de la licencia GNU LGPL.

Además, busca solucionar el problema de la diferencia entre los dos modelos de datos coexistentes en una aplicación: el usado en la memoria de la computadora (orientación a objetos) y el usado en las bases de datos (modelo relacional). Para lograr esto permite al desarrollador detallar cómo es su modelo de datos, qué relaciones existen y qué forma tienen. Con esta información Hibernate le permite a la aplicación manipular los datos en la base de datos operando sobre objetos, con todas las características de la POO. Hibernate convertirá los datos entre los tipos utilizados por Java y los definidos por SQL.

### **5.8.6 JavaServer Faces (JSF)**

Es una tecnología y framework para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE. JSF usa JavaServer Pages (JSP) como la tecnología que permite hacer el despliegue de las páginas, pero también se puede acomodar a otras tecnologías como XUL (acrónimo de XML-based User-interface Language, lenguaje basado en XML para la interfaz de usuario).

JSF incluye:

1. Un conjunto de APIs para representar componentes de una interfaz de usuario y administrar su estado, manejar eventos, validar entrada, definir un esquema de navegación de las páginas y dar soporte para internacionalización y accesibilidad.
2. Un conjunto por defecto de componentes para la interfaz de usuario.
3. Dos bibliotecas de etiquetas personalizadas para JavaServer Pages que permiten expresar una interfaz JavaServer Faces dentro de una página JSP.
4. Un modelo de eventos en el lado del servidor.
5. Administración de estados.
6. Beans administrados.

### **5.8.7 Seguridad**

La plataforma Java EE dispone de reglas estándar para controlar el acceso. Estas son definidas por el desarrollador y se implementan al desplegar la aplicación en el servidor. Java EE pone a disposición de los desarrolladores mecanismos de acceso predefinidos para que ellos no tengan que implementarlos en sus aplicaciones. Una sola aplicación trabaja en diversos ambientes de desarrollo sin necesidad de modificar el código fuente.

### **5.9 HISTORIA DE LA TECNOLOGÍA EN CARRERAS ATLÉTICAS:**

Se inició con una tecnología llamada ChampionChip utilizada en 1993 por la organización de la carrera de las siete colinas en Nijmegen en Holanda. En 1995, La maratón de Rotterdam fue la primera maratón internacional que utilizó plenamente el sistema ChampionChip.

La Maratón de Nueva York lo empezó a usar en 1999 y Boston en 1996, en Colombia la Media Maratón de Bogotá introdujo su uso en el año 2006. Aunque existen otros sistemas utilizados en el mundo como el Chip Timing, D-tag y b-tag, entre otros, el chip sigue siendo uno de los más usados.

### **5.10 ¿CÓMO FUNCIONA EL SISTEMA DE CHIPS?**

La tecnología de chip consta de varias partes, entre ellas el chip y las antenas. Las antenas emisoras y receptoras están dentro de unas finas alfombras de tartán. Estas alfombras están situadas en la línea de meta y otros lugares de cronometraje y están conectadas a una caja amarilla a un lado de la vía que contienen baterías y aparatos eléctricos. Cada vez que un atleta cruce las alfombras, la antena envía un campo electromagnético, el Chip envía el código único que tiene cada atleta. El chip se energiza y envía su número de identificación y la hora o tiempo de paso por ese lugar.

#### **5.10.1 Elementos del sistema de identificación:**

1. El chip que debe llevar cada corredor, ubicado preferiblemente en el zapato del corredor.
2. Alfombras o sensores receptivos, que se colocan en los puntos de control o en la línea de llegada las cuales al paso del corredor detecta el chip y emite el código y tiempo.

### 5.11 IMAGEN DE ANTENAS

Figura 4. Punto de control META - 2015



En la figura se puede observar las antenas utilizadas en el año 2015, en el centro podemos ver una antena RFID para carreras atléticas tipo tapete, conectada con dos antenas en los extremos verticales, estas en función de identificar los tags dentro de los números de los competidores.

### 5.12 PRUEBAS ATLÉTICAS EN BUCARAMANGA

La Fundación Cardiovascular de Colombia con el propósito de ofrecer a la comunidad santandereana actividades que promuevan la salud cardiovascular y el sano esparcimiento, realiza eventos deportivos y culturales en el departamento desde el año 2004 cuando realizó la primera versión del  $\frac{1}{4}$  de Maratón Ciudad de Bucaramanga.

CARRERA ATLÉTICA DE LA UIS, Bucaramanga, Colombia (1988) 26 ediciones celebrando el aniversario de la UIS “caminando, trotando y corriendo”. Carrera atlética de 5 km, considerada la carrera más antigua de Santander y “Universitaria” de Colombia, es un evento que ha logrado mantener su continuidad. Inició corriéndose en el mes de agosto, pero se institucionalizó, el primer domingo de marzo, dentro del programa de actividades de conmemoración del aniversario de la Universidad Industrial de Santander, con una participación de más de 3800 corredores en las diferentes categorías para el año 2016.

La Iª Carrera Atlética de la UIS se realizó dentro del marco de la celebración de los 40 años de labores académicas de la Universidad Industrial de Santander. El Departamento de Cultura Física junto con los centros de estudio de las carreras de ingeniería Química, Eléctrica y Mecánica, programaron la Maratón “40 años UIS”.

## 6. DISEÑO METODOLÓGICO

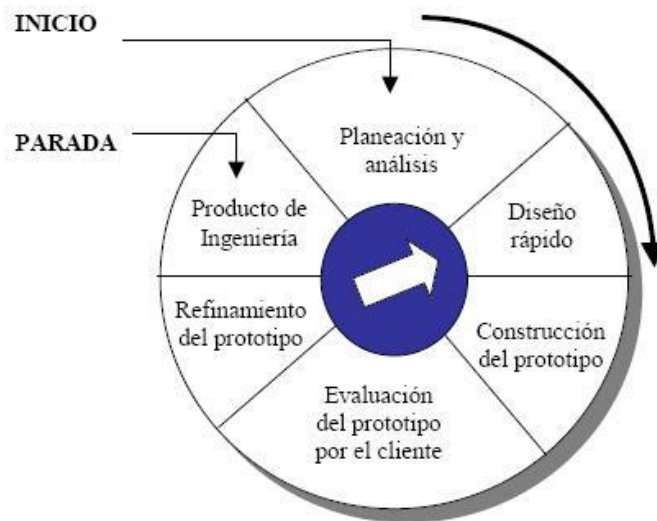
### 6.1 PROTOTIPO EVOLUTIVO

Este proyecto sigue los lineamientos de una investigación cualitativa donde se busca analizar, interpretar y describir la realidad de una población objeto de estudio. La población estudiada tiene un papel fundamental en la labor investigativa, ya que con base en sus opiniones, creencias e ideas, el conocimiento nace de la construcción y reconstrucción colectiva por parte de todas las personas partícipes.

La metodología de investigación que sirve al siguiente proyecto, es la investigación -acción. En esta metodología el investigador observa, analiza e interpreta las acciones realizadas por un grupo de sujetos inmersos en el sistema o fenómeno de estudio. A partir de esto el investigador elabora teorías basadas en la observación realizada, las cuales son posteriormente probadas en el sistema. Como resultado se obtiene experiencia, que será usada como realimentación para el nacimiento de nuevas teorías y acciones.

El desarrollo del prototipo inicial contiene las características visibles del sistema. Después de la presentación al usuario, este prototipo se convierte en la base de desarrollos posteriores, tomando en cuenta la realimentación dada por los usuarios del sistema. Este ciclo se mantiene hasta que se obtiene el producto final de software.

Figura 5. Prototipo Evolutivo.



Fuente: MVC. [En línea]. <https://requisitosdesoftware.wikispaces.com/D.+Prototipado+Evolutivo> [Junio de 2016]

La elección de esta metodología se debe a las siguientes razones:

1. Durante la primera etapa de los nuevos servicios es necesario tener una clara interpretación de las especificaciones dadas por los usuarios, pero muchas veces los usuarios no tienen una idea clara de lo que necesitan. Por esto la construcción de prototipos brinda la posibilidad de efectuar refinamientos de los requerimientos en forma sucesiva a fin de acercarse al producto deseado.
2. Otra ventaja es tener la posibilidad de realizar cambios en etapas tempranas y crear varios prototipos evaluables durante el desarrollo, obteniéndose de este modo una metodología integral para el proceso de evaluación del programa.
3. Esta metodología favorece la autocrítica al sistema, lo que conlleva a que se produzca muchas pruebas antes de dar un nuevo prototipo, así como mejoras rápidas a problemas que puedan surgir durante su uso.

Procedimiento a seguir para la metodología planteada:

1. Se toma nota de los requisitos para la construcción de los prototipos.
2. Se especifican en detalle los objetivos globales del software a realizar, mediante una reunión entre el desarrollador y el usuario, en la cual se identifican los requisitos básicos y se concluyen las áreas donde se necesita mayor definición.
3. Posteriormente se presenta al usuario el diseño de un prototipo que se enfoca en los aspectos visuales del software, métodos de entrada y formatos de salida, para proceder a la construcción.
4. El prototipo es evaluado por el usuario y se utiliza para filtrar los requisitos del software a desarrollar.
5. Se produce un proceso interactivo en el que el prototipo es depurado para satisfacer necesidades del usuario, de igual forma el desarrollador obtiene una mejor comprensión de lo que hay que hacer para la entrega del producto final de ingeniería requerido por el usuario.

## **6.2 LENGUAJE DE MODELADO UNIFICADO**

El Lenguaje de Modelado Unificado o Unified Modeling Language (UML), es el más utilizado en la actualidad. Es un lenguaje gráfico estándar para visualizar, especificar, construir y documentar un sistema para describir un modelo del sistema, incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

### **6.2.1 Diagramas de UML.**

Los diagramas UML utilizados en el desarrollo de este proyecto fueron: diagramas de casos de uso y diagramas de secuencias. Las principales razones por las cuales se prefirió UML como el lenguaje de modelado son:

1. UML tiene una notación gráfica muy expresiva que permite representar todas las fases de un proyecto informático: desde el análisis con casos de uso, el diseño con diagramas de clases, objetos, etc.
2. UML facilita el entendimiento de la información, la función y el comportamiento de un sistema, haciendo fácil el análisis de los requerimientos, ya que sirve de apoyo en los procesos de análisis

- de un problema.
3. UML permite a los creadores de sistemas realizar diseños que faciliten la comunicación a otras personas de manera convencional.
  4. UML permite generar un punto de comparación entre lo logrado y lo planificado.

### **6.2.2 Diagramas de casos de uso.**

Representación gráfica del entorno del sistema (actores) y su funcionalidad principal. Describe lo que hace el sistema desde el punto de vista de un observador externo, concentrándose en expresar lo que hace el sistema y no en dar respuesta de cómo lograr su comportamiento. (Ver Anexo C).

### **6.2.3 Diagramas de secuencias.**

Es aquel que muestra la forma en que los objetos interactúan entre sí al transcurrir el tiempo. Consta de objetos que se representan del modo usual: rectángulos con nombre (subrayado), mensajes representados por líneas continuas con una punta de flecha y el tiempo representado como una progresión vertical. (Ver Anexo C).

## **6.3 ESTÁNDARES DE PROGRAMACIÓN**

### **6.3.1 Modelo de datos.**

Es un lenguaje utilizado para la descripción de una base de datos, por lo general permite describir estructuras de datos de la base de datos (el tipo de datos que incluye la base y la forma en que se relacionan), las restricciones de integridad (las condiciones que los datos deben cumplir para reflejar correctamente la realidad deseada) y las operaciones de manipulación de los datos (agregar, borrar, modificar).

### **6.3.2 Nombres de las tablas.**

Los nombres de los campos, así como de las tablas de la base de datos, se escriben en minúsculas, exceptuando la primera letra de cada palabra que conforme su nombre; si es un nombre compuesto por dos o más palabras, los nombres tendrán en mayúscula la primera letra de cada palabra que la forma.

Se han definido tres categorías para las diferentes tablas que conforman la base de datos. Dada la categoría de la tabla, se antepondrá un prefijo a su nombre que permita conocer la categoría a la que pertenece. Las categorías son:

1. *Tabla básica:* Aquella cuyos registros son necesarios para el correcto funcionamiento de la base de datos. Estas tablas no experimentan muchos cambios en los datos. El prefijo a anteponer a los nombres de estas tablas es “nombre\_de\_tabla”, es decir la tabla que almacena los puntos que se tratan en un punto de control de la prueba es llamada “punto\_control”.
2. *Tabla de Relación:* Surge de la relación muchos a muchos de una o dos tablas cualquiera. Los nombres de las tablas de relación deberán ser descriptivos para cada relación. El prefijo a

anteponer a los nombres de estas tablas es “nombre1nombre2”, es decir la tabla de “CategoriaPrueba” es conocida como “categoría\_prueba”.

### **6.3.3 Clases.**

Los nombres de las clases deben ser sustantivos en plural, la primera letra de cada palabra debe ser mayúscula. Estos deben ser simples, descriptivos como por ejemplo: CategoriaPrueba.java, Inscrito.java.

### **6.3.4 Páginas XHTML.**

Los nombres de las paginas XHTML que componen el sistema de información serán escritos iniciando en minúscula, en caso de que el nombre del XHTML compuesto por dos o más palabras, entonces la primera de cada palabra interna debe ir en mayúscula, por ejemplo: administrarPrueba.xhtml, modalModificarInscrito.xhtml.

### **6.3.5 Organización de Directorios.**

Los directorios del sitio están organizados de tal manera que los archivos que se almacenen en ellos correspondan a lo que describe el nombre del directorio. Por ejemplo:

1. El sitio cuenta con un directorio llamado “carrerasEntidades”; en éste se encuentran almacenados todos los directorios, archivos de configuración y de las clases que tiene cada entidad definida.
2. El sitio cuenta con un directorio llamado View, donde están todos los archivos y páginas que se muestran en la interfaz gráfica del usuario, entre estos los directorios: estilos, imágenes, módulos, plantillas y recursos.

## 7. DESARROLLO DE LA HERRAMIENTA, ADMINISTRACIÓN Y MANTENIMIENTO

Como se mencionó anteriormente para el desarrollo de este proyecto se siguió la metodología de prototipo evolutivo. Al iniciar el proyecto se elaboró un primer prototipo durante la fase de requerimientos, el cual fue mejorado con la inclusión de nuevos requerimientos surgidos en la fase de desarrollo, a medida que se generaba un prototipo, el mismo era sometido a pruebas de funcionamiento y se le realizaban los refinamientos pertinentes a partir del resultado de dichas pruebas.

### 7.1 PRIMER PROTOTIPO

El primer prototipo elaborado consistió en el desarrollo de la interfaz de usuario de los módulos de fases, actividades, riesgos y entregables.

Dado que las actividades, riesgos y entregables están relacionadas directamente con las fases del proyecto, la primera interfaz en ser desarrollada fue la de planificación de las fases del proyecto.

Adicionalmente se implementaron las siguientes funcionalidades básicas:

1. Crear registros de fases, actividades, entregables y riesgos.
2. Ver información detallada de un registro seleccionado.
3. Modificar los registros creados.
4. Eliminar un registro seleccionado.

A continuación se muestra la interfaz presentada por el departamento de deportes de la Universidad Industrial de Santander para el año 2015:

Figura 6. Página inicial – Resultados

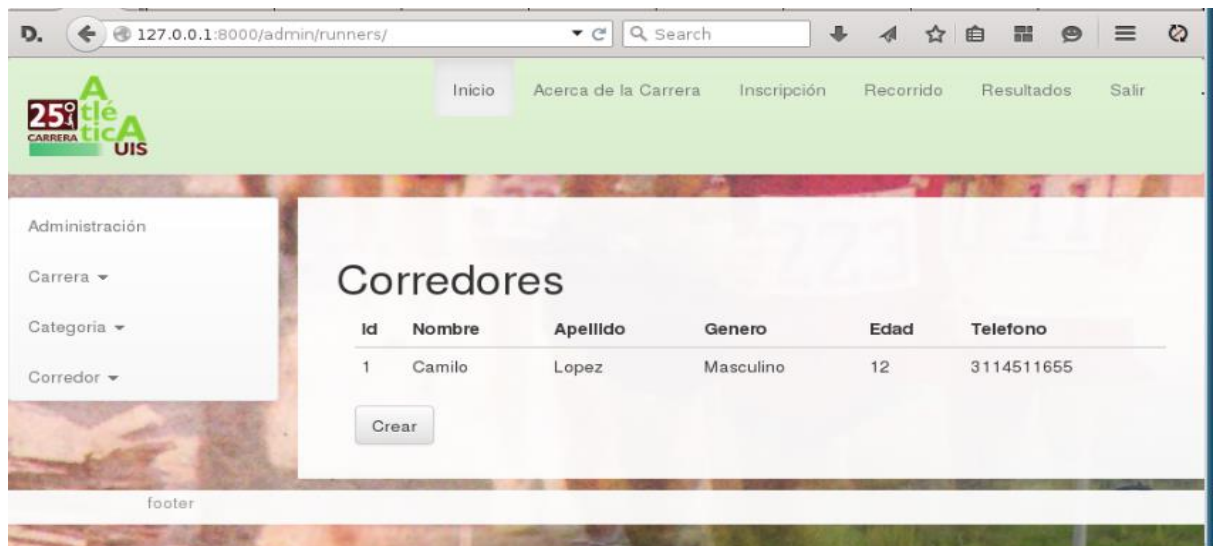


Figura 7. Crear categoría

The screenshot shows a web browser window with the URL `127.0.0.1:8000/admin/categories/add/`. The page features a green header with the logo for '25º Carrera Atlética UIS' and a navigation menu with items: Inicio, Acerca de la Carrera, Inscripción, Recorrido, Resultados, and Salir. On the left, a sidebar menu includes 'Administración', 'Carrera', 'Categoría', and 'Corredor'. The main content area is titled 'Categoría' and contains the following form fields: 'Nombre\*' (text input), 'Genero\*' (dropdown menu with 'Mixto' selected), 'Edad Mínima\*' (text input with a spinner), 'Edad Máxima\*' (text input with a spinner), and 'Color\*' (text input). At the bottom of the form are 'Cancelar' and 'Guardar' buttons.

Figura 8. Inscripción

The screenshot shows a web browser window with the URL `127.0.0.1:8000/admin/runners/add/`. The page features a green header with the logo for '25º Carrera Atlética UIS' and a navigation menu with items: Inicio, Acerca de la Carrera, Inscripción, Recorrido, Resultados, and Salir. On the left, a sidebar menu includes 'Administración', 'Carrera', 'Categoría', and 'Corredor'. The main content area is titled 'Inscripción' and contains the following form fields: 'Nombre\*' (text input), 'Apellidos\*' (text input), 'Tipo de Identificación\*' (dropdown menu), 'Numero de Identificación\*' (text input), 'Sexo\*' (dropdown menu), 'Fecha de Nacimiento\*' (text input), 'Edad\*' (text input with a spinner), 'Tipo de Sangre' (text input), 'Pais\*' (dropdown menu), 'Región\*' (dropdown menu), 'Ciudad\*' (dropdown menu), and 'Telefono\*' (text input).

## 7.2 PROTOTIPO FINAL

Este prototipo cumple a cabalidad con todos los objetivos estipulados en el plan del proyecto, incluyendo las políticas de seguridad instauradas por la DSI de la Universidad Industrial de Santander.

El esquema de seguridad está basado en la estructura de roles usuarios, este esquema se define ampliamente en el numeral 7.2.1.

Actualmente los sistemas de información desarrollados en la DSI no sólo deben cumplir con los estándares de diseño y programación, sino también deben implementar la seguridad que administra el control de acceso a cada sistema de información que apoya la gestión de la Universidad Industrial de Santander.

A continuación se realiza una breve descripción acerca del proceso de implementación del sistema de seguridad. En el siguiente numeral se tratará el tema con más detalle.

Para que todos los programas desarrollados puedan implementar el sistema de seguridad, básicamente es necesario identificar los roles asociados a los usuarios potenciales del Sistema de Información de la Carrera Atlética de la Universidad Industrial de Santander.

Una vez establecidos los roles se procede a establecer los permisos y menús de cada rol, tareas que posteriormente debe realizar el administrador del sistema de seguridad.

Cuando la información descrita en el párrafo anterior se encuentra registrada en la base de datos del sistema de seguridad se procede a configurar la aplicación de gestión de proyectos con el fin de adaptarla al esquema de seguridad definido específicamente para dicha aplicación.

Para el sistema de la prueba atlética de la UIS existen dos tipos de roles de usuarios principales:

1. Administrador: persona encargada de crear y modificar las tablas soportes que gestionan una prueba atlética.
2. Competidor: persona interesada en participar en el evento, inscribirse y tener medición de su tiempo y posición en la prueba.

### Módulo Administración

El prototipo definitivo abarca las siguientes funcionalidades:

1. Administración completa de crear tablas soporte para gestionar las pruebas atléticas definidas.
2. Gestión de pruebas, categorías, puntos de control, inscritos y tipos de pruebas.
3. Modificar información de las diferentes tablas del sistema de información.
4. Cargar los registros de tiempos, números de los competidores y generar resultados.

### Módulo Inscripción

El módulo de inscripción se encarga de adquirir y almacenar la información de los interesados en competir en una prueba atlética:

1. Gestión de inscritos, registro de información de competidores a la prueba.
2. Validación de la información en formulario de registro y respectivo guardado en base de datos.
3. Listado de inscritos y modificación o eliminación de registros por parte del administrador.

## Módulo Consulta Resultados

La consulta de los resultados de la prueba atlética comprende:

1. Una vez seleccionado el módulo de consulta tendrás disponibles criterios de selección que permitirán que los resultados de la búsqueda se ajusten a lo que necesitas, siendo por identificación del competidor, categoría, género o resultados generales.
2. Desplegar los registros relacionados con los datos cargados por el administrador y los competidores inscritos en una tabla de acuerdo al criterio de consulta seleccionado.

### **7.2.1 Esquema de seguridad Universidad Industrial de Santander**

Para este proyecto se utiliza el esquema de seguridad definido por la División de Servicios de Información para los diferentes sistemas de información que apoyan la gestión de la Universidad Industrial de Santander, el cual está basado en la estructura de roles – usuarios.

Los roles se establecen en cada una de las unidades académico administrativas, UAA, responsables de cada sistema, de acuerdo a las actividades que realizan. A cada uno de los roles definidos se le asocian los usuarios de acuerdo a las funciones que desempeñen.

#### **7.2.1.1 Estructura de la Base de Datos soporte**

La base de datos que soporta el esquema de seguridad contempla básicamente las siguientes tablas:

1. Sistema: Contiene información de los sistemas de información de la universidad. Para cada sistema se especifica: Nombre, descripción del sistema, fecha y hora de creación en la base de datos, fecha y hora de inicio de vigencia del sistema, fecha y hora de cierre de vigencia del sistema.
2. Rol: contiene información de los diferentes roles definidos para cada sistema de información, como: Nombre asignado al rol, descripción del rol, fecha y hora de creación, fecha y hora de inicio de vigencia del rol, fecha y hora de cierre de vigencia del rol.
3. Usuario: Contiene información de los posibles usuarios de los sistemas de información. Entre esta información está: tipo y número de documento de identidad del usuario, fecha y hora de creación del usuario, fecha y hora de inicio de vigencia del usuario, fecha y hora de cierre de vigencia del usuario.
4. Sistema–rol: Contiene los roles definidos para cada uno de los sistemas de información, indicando: rol, sistema, fecha y hora de creación del rol – sistema, fecha y hora de inicio de vigencia del rol en el sistema, fecha y hora de cierre de vigencia del rol en el sistema.
5. Rol–usuario: Contempla los usuarios asociados a cada uno de los roles definidos, considerando: Rol, usuario, fecha y hora de creación del rol – usuario, fecha y hora de inicio de vigencia del usuario en el rol, fecha y hora de cierre de vigencia del usuario en el rol.

6. Menú–rol–sistema: Contiene los menús asociados a los roles en los distintos sistema de información, contemplando: Sistema de información, nombre del menú, descripción del menú, fecha y hora de creación del menú, fecha y hora de inicio de vigencia del menú asociado al rol, fecha y hora de cierre de vigencia del menú asociado al rol.
7. Opción–menú–rol: Contempla las opciones definidas para cada una de los posibles menús establecidos para cada sistema de información. Contiene: Nombre de la opción, descripción de la opción, nombre del menú superior, nombre del menú que contiene la opción, nombre del programa a ejecutar cuando la opción es la de más bajo nivel, fecha y hora de creación de la opción del menú, fecha y hora de inicio de vigencia de la opción, fecha y hora de cierre de la opción.
8. Tabla–sistema: Contiene información de las tablas que conforman la base de datos que soporta cada uno de los sistemas de información. Considera: Sistema de información, nombre de la tabla, descripción de la tabla.
9. Tipo–permiso: Establece para cada tabla de un sistema de información, los roles que tienen permisos para incluir registros, para modificar registros o para eliminar registros en ella. Contiene: Sistema de información, nombre de la tabla, clase de permiso (inclusión, modificación, eliminación de registros), fecha y hora de creación del permiso, fecha y hora de inicio de vigencia del permiso, fecha y hora de fin de vigencia del permiso.
10. Acceso–tabla: Define para las tablas de un sistema de información si un rol tiene permiso sobre toda la información de la tabla o sobre una parte de esta. Considera: Sistema, nombre de la tabla, clase de acceso (total, parcial), fecha y hora de creación del permiso, fecha y hora de inicio de vigencia del permiso, fecha y hora de fin de vigencia del permiso.
11. Atributo–tabla: Establece los atributos sobre los cuales se debe controlar el acceso a una tabla, cuando a un rol se le concede permiso para hacer uso parcial de la información existente en una tabla. Contiene: Sistema de información, nombre de la tabla, nombre del atributo sobre el cual se controla el acceso a la tabla, descripción del atributo, fecha y hora de creación del atributo, fecha y hora de inicio de vigencia del atributo, fecha y hora de fin de vigencia del atributo.
12. Valor–atributo–proceso: Contiene los valores que deben tener los atributos definidos en cada tabla en la tabla atributo – tabla que permiten el acceso a la información asociada a estos valores. Específica: Sistema de información, nombre de la tabla, nombre del atributo, valor del atributo, descripción, fecha y hora de creación del valor del atributo, fecha y hora de inicio de vigencia del valor del atributo, fecha y hora de fin de vigencia del valor del atributo.
13. Acceso–sistema: Contempla el histórico de acceso que un usuario ha realizado a un sistema, identificando las opciones que ha seleccionado. Contiene: Login de usuario, rol, identificación de la sesión, sistema, opción seleccionada, fecha y hora de ingreso, fecha y hora de salida.

### **7.2.2 Entorno de Navegación**

Para cada sistema de información, la UAA responsable define los roles necesarios para el adecuado uso del sistema de información de acuerdo a las funciones que realice y establece los usuarios asociados a cada uno de ellos.

Para cada rol se define el menú de inicio, el cual le permite a cada usuario que hace parte de este rol, empezar la navegación por las distintas opciones que le ofrece el sistema, hasta llegar al nivel más bajo en el cual se ejecuta el proceso que soporta la actividad que desea realizar.

Este entorno está soportado por las siguientes tablas de las base de datos del esquema de seguridad: Rol, usuario, sistema, sistema-rol, usuario-rol, menú-rol, opción-menú rol.

### **7.2.3 Entorno de Control de Datos**

Para los roles definidos en cada uno de los sistemas de información se especifican las tablas a las cuales puede acceder, el tipo de transacción que puede realizar sobre estas tablas (inclusión, modificación o eliminación de registros), si tiene acceso total o parcial a la información que contiene la tabla.

Para el acceso a la información de la tabla de manera parcial, se debe establecer el atributo o atributos seleccionados, los valores que estos atributos deben tener para autorizar el acceso solicitado.

Este entorno está soportado por las siguientes tablas de las base de datos del esquema de seguridad: Rol, usuario, sistema, sistema-rol, usuario-rol, tabla-sistema, tipo-permisos, acceso-tabla, atributo-tabla, valor atributo proceso.

### **7.2.4 Auditoría**

Las tablas inscrito\_asistente\_punto\_control y resultado que conforman la base de datos soporte del esquema de seguridad tienen el historial de las transacciones realizadas sobre cada una de ellas.

El historial de las transacciones de cada tabla contiene información de los registros incluidos en la tabla, de los registros modificados y de los registros eliminados. Adicionalmente, en cada transacción se especifica: Fecha de la transacción, hora de la transacción, tipo de transacción (I/U/D), tipo y número de documento de identidad del usuario que realizó la transacción, login, rol asociado, dirección IP y MAC del equipo desde el cual llevó a cabo la transacción.

Al inicio el proyecto no se tenía una concepción clara de cómo sería el producto final, sin embargo, durante el desarrollo y evolución de los prototipos, las pruebas y análisis del sistema se pudo comprobar que se estaba acercando a los requerimientos establecidos, esto con el fin de enfocar exitosamente el desarrollo a la solución de las necesidades de los usuarios.

## 7.2.5 Interfaz Gráfica

Se presenta la interfaz gráfica de usuario con los estilos definidos en los sistemas de información de la DSI para la Universidad.

Figura 9. Página inicial

CONSTRUIAMOS FUTURO

Viernes, 27 de Enero de 2017 - 14:51

Contáctenos Visítenos Búsqueda Directorio Mapa del Sitio Guía de Navegación

Inicio La UIS Unidades Académicas Programas Académicos Investigación y Extensión Profesores Estudiantes Gestión Administrativa Eventos Emisoras

Sistema: Deportes  
Rol:  
Usuario:

**Sistema carrera atlética**  
Bienvenidos

**Administración**  
Inscripción  
Consultar Resultados

**Administración**  
Administrar categoría  
Administrar prueba  
Administrar inscrito  
Administrar tipoPrueba  
Administrar categoriaPrueba  
Administrar puntoControl  
Administrar inscritoCategoriaPrueba

**Objetivo**  
Promover el deporte como una cultura física y los buenos hábitos de conservación de la salud hacia una mejor calidad de vida de la población interna y externa de la Universidad.

**Presentación**  
El Departamento de Educación Física y Deportes de la UIS extiende una cordial invitación a toda la comunidad Universitaria y los diferentes entes a nivel regional, nacional e internacional para que participen en esta gran fiesta Deportiva, la XXXVII Carrera Atlética de la UIS.

**PARTICIPE CAMMIANDO TROTANDO O CORRIENDO**

Día del Evento: Domingo 26 de Marzo de 2017  
Hora: 8:30 am  
Distancia: 5 Kilómetros

Categoría	Rama	Distancia y Sitio de Salida	Llegada	Hora Salida
Estudiantes UIS	Masculino y Femenino	5 Kms Estadio la Flora	UIS	8:30 am
Estudiantes Universitarios Abierta	Masculino y Femenino	5 Kms Estadio la Flora	UIS	8:30 am
Profesores y Empleados UIS	Masculino y Femenino	5 Kms Estadio la Flora	UIS	8:30 am
Élite	Masculino y Femenino	5 Kms Estadio la Flora	UIS	8:30 am
Egresados UIS	Masculino y Femenino	5 Kms Estadio la Flora	UIS	8:30 am
Categoría Master "A" (40 - 49 años)	Masculino y Femenino	5 Kms Estadio la Flora	UIS	8:30 am
Categoría Master "B" (Mayor de 50 años)	Masculino y Femenino	5 Kms Estadio la Flora	UIS	8:30 am

**Inscripciones**  
Fecha: Hasta el Día Viernes 24 de Marzo  
Hora: 6:00 pm  
Lugar: Departamento de Deportes de la UIS, Coliseo de la UIS, Segundo Piso  
Telefax: 6450018 UIS, 334000 Extensión:: 2318 ~ 1143 Bucaramanga, Liga Santandereana de Atletismo  
Teléfonos: 3144704181 ~ 3153899828 Bucaramanga

**Hora de Salida y Recorrido**  
Salida para todas las Categorías: 8:30 am.  
Lugar de salida: Estadio de Atletismo La Flora  
Lugar de llegada: Estadio de Atletismo de la UIS.  
Distancia recorrida: 5 Kmts.

Esta página muestra la interfaz de inicio, información al público del evento, en la parte superior un menú a enlaces propios del sitio principal de la Universidad Industrial de Santander, además un menú vertical del sistema de información de la carrera atlética, los módulos de administración, inscripción y consultar resultados.

En las figuras 10 y 11 son muestra de las tablas soporte para la administración.

Adicionalmente la figura 12 muestra el formulario de inscripción para el usuario competidor.

Figura 10. Administrar Prueba

Sistema carrera atlética  
Administrar prueba

Crear prueba

Código: 2  
Nombre:   
Ciudad: Bucaramanga  
Tipo:   
Año de prueba:   
Versión:   
Tipo Usuario:   
Guardar Cancelar

Código	Prueba	Acción
1	XXXVII Carrera Atlética UIS	

Total Registros: 1

Figura 11. Administrar categoría

Sistema carrera atlética  
Administrar categoría

Crear categoría

Código: 13  
Nombre:   
Género:  Femenino  Masculino  
Debe pagar:  Si  No  
Valor pago: 0  
Guardar Cancelar

Código	Categoría	Acción
1	Estudiante UIS Femenino	
2	Estudiante UIS Masculino	
3	Estudiante Abierta Nacional F	
4	Estudiante Abierta Nacional M	
5	Profesores y Empleados UIS Fem	
6	Profesores y Empleados UIS Mas	
7	Élite Femenino	
8	Élite Masculino	
9	Master 'A' (40-49 AÑOS) Femeni	
10	Master 'A' (40-49 AÑOS) Mascul	
11	Master 'B' (Mayor de 50 años)	
12	Master 'B' (Mayor de 50 años)	

Total Registros: 12

Figura 12. Inscripción

En este modulo podras inscribirte

Sistema: Deportes  
Rol:  
Usuario:

**Administración**  
**Inscripción**  
**Consultar Resultados**

[Inicio](#)  
[Salir](#)

[Ocultar Menú](#)

### Sistema carrera atlética

En este modulo podras inscribirte

(\*) Información requerida \*

**Crear inscrito**

Tipo Documento: Cédula  
Código: 2090101  
Nombre:   
Apellido:   
Número de celular:   
Dirección:   
Correo electrónico:   
Género:  Femenino  Masculino  
EPS:   
Fecha de nacimiento:   
País: Colombia  
Tipo RH:   
Nombre contacto:   
Número contacto:

Bucaramanga - Colombia. Cra 27 calle 9. pbx: (57) (7) 6344000. nit: 890201213-4  
Resolución de pantalla recomendada 1024 x 768 pixeles

Administrador Web  
webadmin@uis.edu.co

Figura 13. Resultados

Consultar los resultados de la prueba

Sistema: Deportes  
Rol:  
Usuario:

**Administración**  
**Inscripción**  
**Consultar Resultados**

[Inicio](#)  
[Salir](#)

[Ocultar Menú](#)

### Sistema carrera atlética

Módulo de consultas

**Lista de resultados**

Número de atleta: No seleccionó una opción

(\*) Información requerida \*

Registro

Lista de resultados				
Id del resultado	Nombre	Apellido	Número de atleta	Tiempo
1	Juanita	Perez	2001	1700-02-01 09:54:00.0
2	Marina	Perez	2002	1700-02-01 09:54:00.0
3	Juan Camilo	Barajas Rueda	801	1700-02-01 09:54:00.0
5	Pepta	Perez	3501	1700-02-01 10:04:00.0
6	Juan Felipe	Rueda	3201	1700-02-01 10:04:00.0
7	Humberto	Ruiz	3801	1700-02-01 10:15:00.0
8	Enrique	Torres	3802	1700-02-01 10:18:00.0
9	Francisco	Russi	1	1700-02-01 10:30:00.0
10	Katherine	Ibargüen	301	1700-02-01 10:32:00.0
11	Maria	Garzón	2004	1700-02-01 11:05:00.0

Total Registros: 10

Bucaramanga - Colombia. Cra 27 calle 9. pbx: (57) (7) 6344000. nit: 890201213-4  
Resolución de pantalla recomendada 1024 x 768 pixeles

Administrador Web  
webadmin@uis.edu.co

### **7.2.6 Requerimientos de objetivo.**

El objetivo específico inicial y los requisitos que surgieron se dieron gracias a la realización de prototipos y la realimentación con los interesados en el proyecto, en este caso la administración de la carrera atlética de Universidad Industrial de Santander y la DSI. Para cada objetivo se listan los requerimientos detallados de este, los cuales se cumplieron para el prototipo final.

#### **7.2.6.1 Crear la inscripción en línea al evento.**

Objetivo Inicial:

1. Diseñar módulo de inscripción que permita a usuarios administradores registrar participantes o al usuario público registrase en línea a un evento.

Requisitos finales del objetivo:

2. Se implementa la interfaz para poder mediante un formulario realizar la inscripción.
3. Las preguntas del formulario deben ser de fácil entendimiento a cualquier usuario.
4. Se ajustan los estilos a los del sitio web de la Universidad Industrial de Santander versión 2016.

#### **7.2.6.2 Crear módulos administrativos para administrar eventos.**

Objetivo inicial:

1. Crear módulos para administrar categorías de la prueba.

Requisitos finales del objetivo:

2. Permitir crear, modificar o eliminar categorías de una prueba atlética.
3. Permitir administrar pruebas, puntos de control, tipos de pruebas y resultados del evento.
4. El usuario administrador podrá acceder a estos módulos y asignar las categorías que se presentaran en una prueba de acuerdo a los requerimientos de cada evento.

#### **7.2.6.3 Implementar módulo de consulta de resultados.**

Objetivo inicial:

1. Desarrollar módulo que permita generar y consultar resultados de un evento

Requisitos finales del objetivo:

2. Permitir visualizar tablas de resultados en una prueba atlética.
3. Brindarle al usuario la posibilidad de generar resultados de acuerdo a una prueba específica, categoría, género, número de participante, documento o versión de la prueba.
4. Permitir a usuario público consultar los resultados del evento.

#### **7.2.6.4 Implementar registro de datos en la prueba utilizando dispositivos RFID.**

Objetivo inicial:

1. Utilizar dispositivos RFID para hacer control de tiempos de los participantes del evento.

Requisitos finales del objetivo:

2. Configuración de dispositivos para la conexión con base de datos donde se guardarán y consultarán los datos de los participantes del evento.
3. Implementar dispositivos de acuerdo a especificaciones de la administración del evento.

#### **7.2.7 Requerimientos del sistema de información.**

Para aquellos servicios que lo requieran, se debe incluir la rutina de logueo de usuario y verificación de sesión, requerimiento de seguridad intrínseco del sistema de información de acuerdo a requerimientos de la DSI.

##### **7.2.7.1 Administración.**

Objetivos iniciales:

1. Efectuar labores de administración del sistema de información; teniendo en cuenta entre otras:
2. Generar Backups (copias de respaldo) diariamente de la Base de Datos.
3. Hacer seguimiento del uso que hacen los usuarios en cuanto a los servicios de inscrito del evento, para detectar usos indebido o incorrecto por parte de éste.
4. Actualizar periódicamente las Bases de Datos con el objeto de mantener la información en los resultados de acuerdo a la comunicación en tiempo real con los dispositivos RFID.
5. Atender consultas y sugerencias que los usuarios hagan para proponerlas como mejoramiento y ofrecimiento de nuevos servicios en el sistema de información del evento
6. Realizar una revisión constante de los archivos que se suben al sitio, eliminando los que no son necesarios para evitar saturación del portal.

Requisitos finales del objetivo:

Para usuarios con perfil de administrador:

1. Corregir cada uno de los fallos que se presentan en el sistema de información.
2. Mantenimiento a la base de datos para reorganizar tablas que se encuentran fragmentadas o crear nuevas que se necesitan para soportar los nuevos servicios.
3. Revisar la estructura de directorios, borrar los archivos que ya no se usan y agregar nuevos servicios implantados en el sistema de información.
4. Documentar el código de los archivos del sistema de información de manera que se pueda dar soporte.

Cada una de las actividades mencionadas en el objetivo inicial se llevó a cabo durante la práctica y desarrollo del proyecto.

### 7.2.7.2 Soporte a Usuarios.

Objetivo inicial:

1. Desempeñar labores de soporte a los usuarios del sistema de información del evento, brindando así solución a sus diferentes necesidades, conflictos o situaciones que se puedan presentar. Entre estas labores se destacan:
2. Capacitar usuarios administrativos en el uso de servicios dentro del sistema de información del evento, promoviendo así su utilización.
3. Agregar opción de información o ayuda en los formularios.

Requisitos finales del objetivo:

Para usuarios con perfil de administrador:

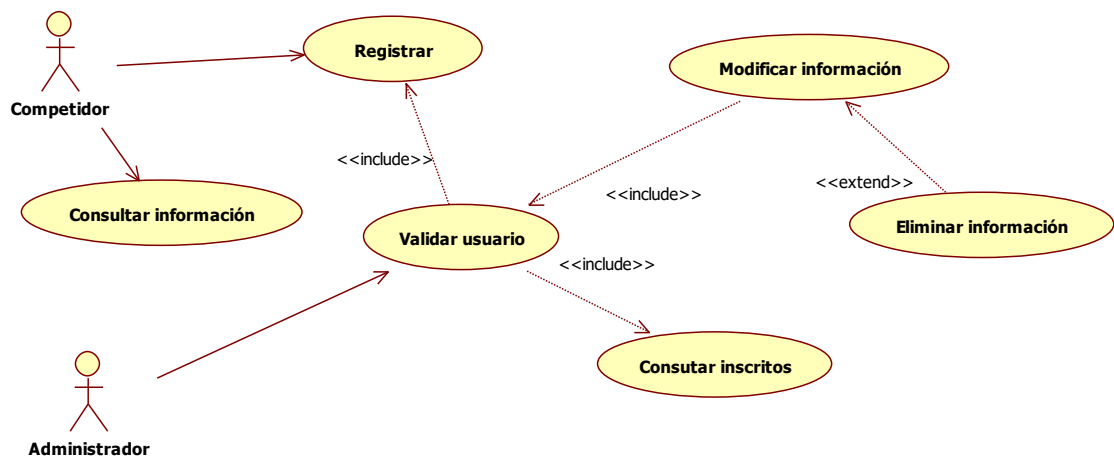
1. Conocer el total funcionamiento de los módulos administrativos del evento, a fin de dar solución a las posibles dificultades que se presenten.
2. Investigar las soluciones a nuevas dificultades que se presenten.

Cada una de las actividades mencionadas en el objetivo anterior se llevará a cabo durante la práctica. Además, se dará orientación a los usuarios sobre el uso de determinados servicios.

### 7.3 DIAGRAMAS DE CASOS DE USO.

Módulo de inscripción.

Figura 14. Diagrama de Casos de Uso: Registrar y consultar.



## Módulo de Administración

En este se administran las categorías, pruebas, puntos de control, tipos de pruebas e inscritos.

Figura 15. Diagrama de Casos de USO: Gestionar Prueba

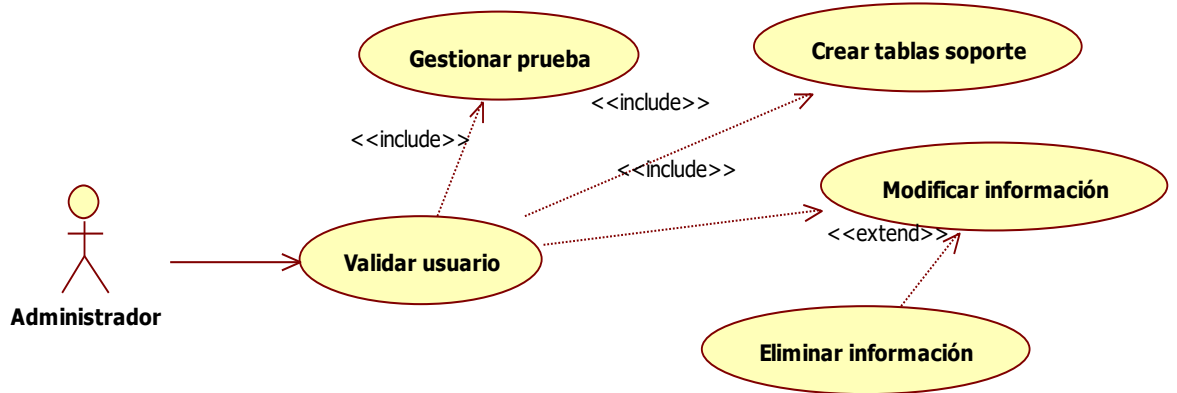


Figura 16. Diagrama de Casos de USO: Cargar y generar resultados

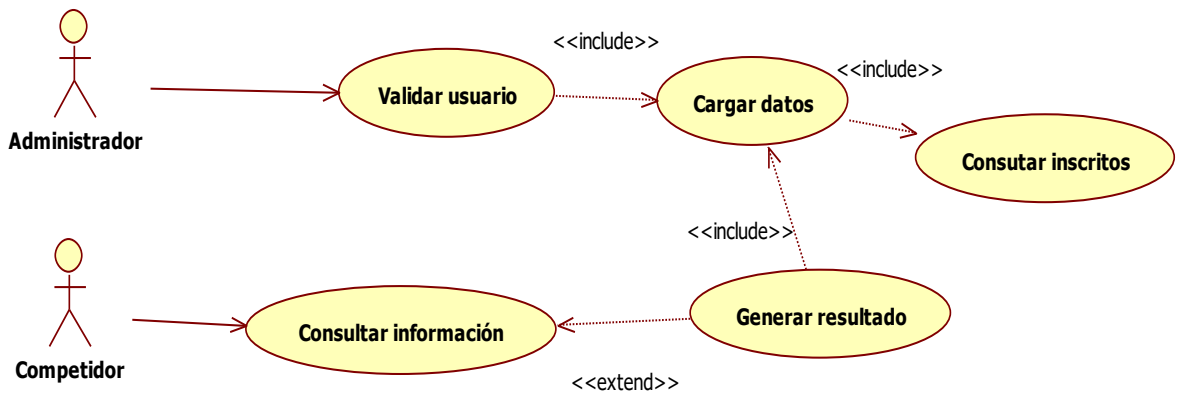


Figura 17. Diagrama de Casos de Uso: Generar número

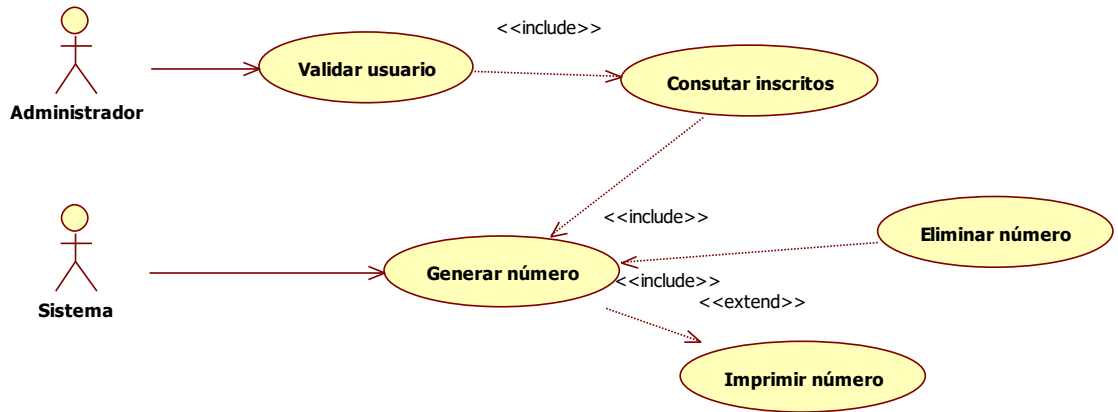
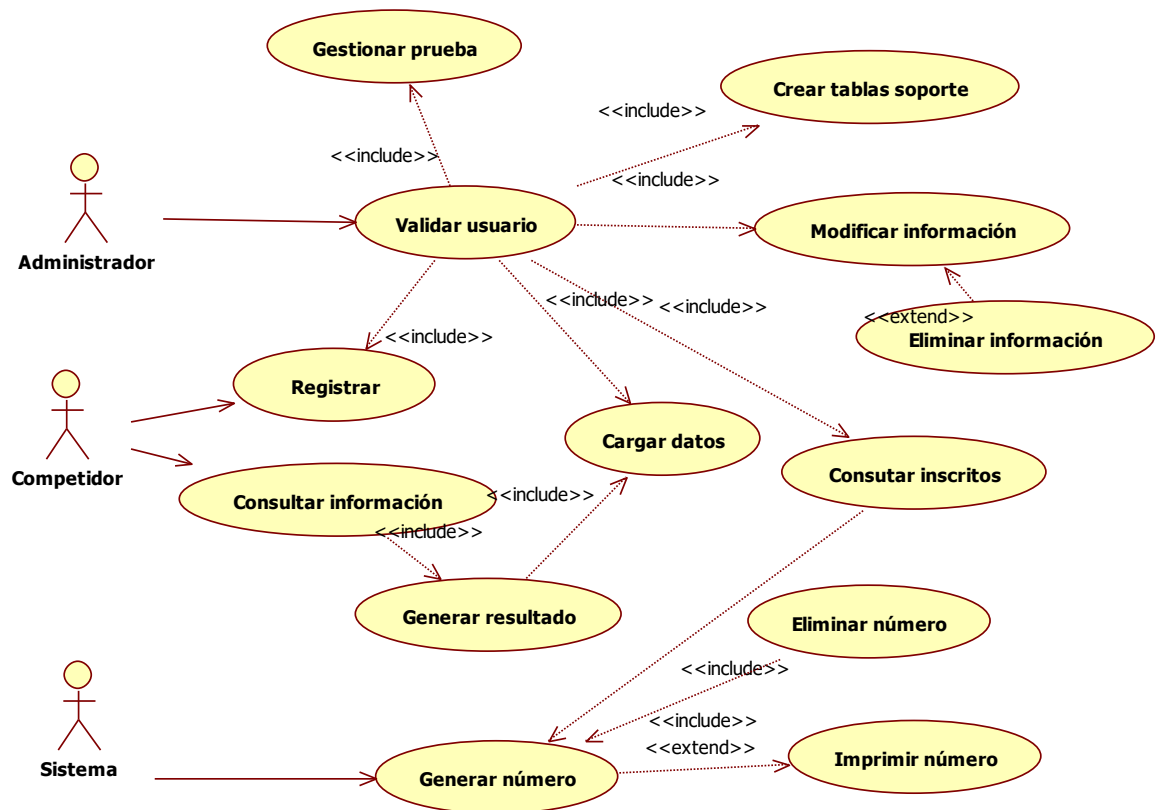


Figura 18. Diagrama de Casos de Uso completo del sistema



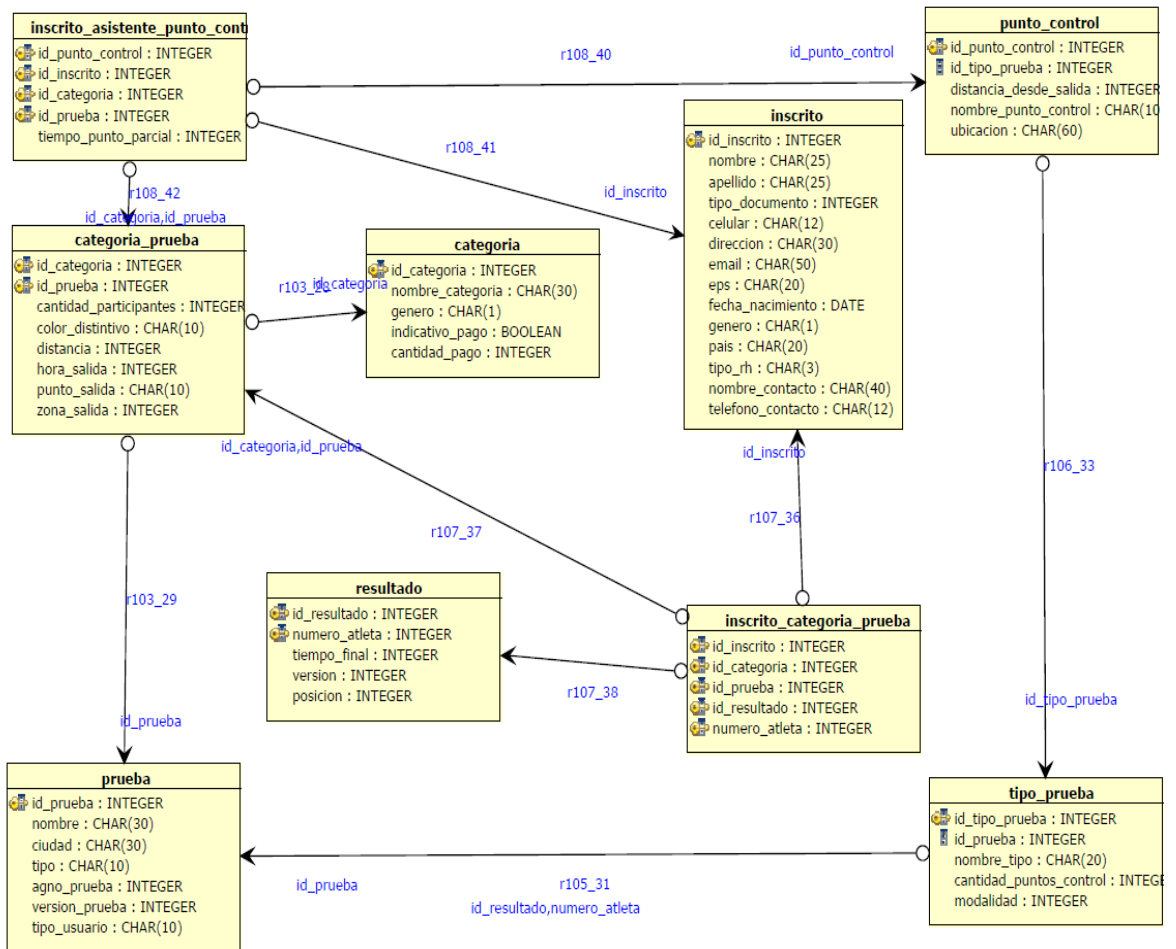
### 7.3.1 Documentación de casos de USO del sistema.

Una vez identificados los casos de uso, se procede a documentar sus pasos, describiendo cada una de las acciones que se llevan a cabo en los casos de uso anteriormente mencionados. Los detalles completos se documentan en los anexos de este libro. (Ver Anexo D).

### 7.3.2 Diseño y Análisis

#### 7.3.2.1 Diagrama Entidad/Relación de los servicios desarrollados.

Figura 19. Diagrama E/R: Sistema completo



### 7.3.2.2 Descripción de las Entidades.

Definición de las entidades utilizadas en la base de datos pruebas atléticas instalada en el servidor de la DSI denominado FAISAN.

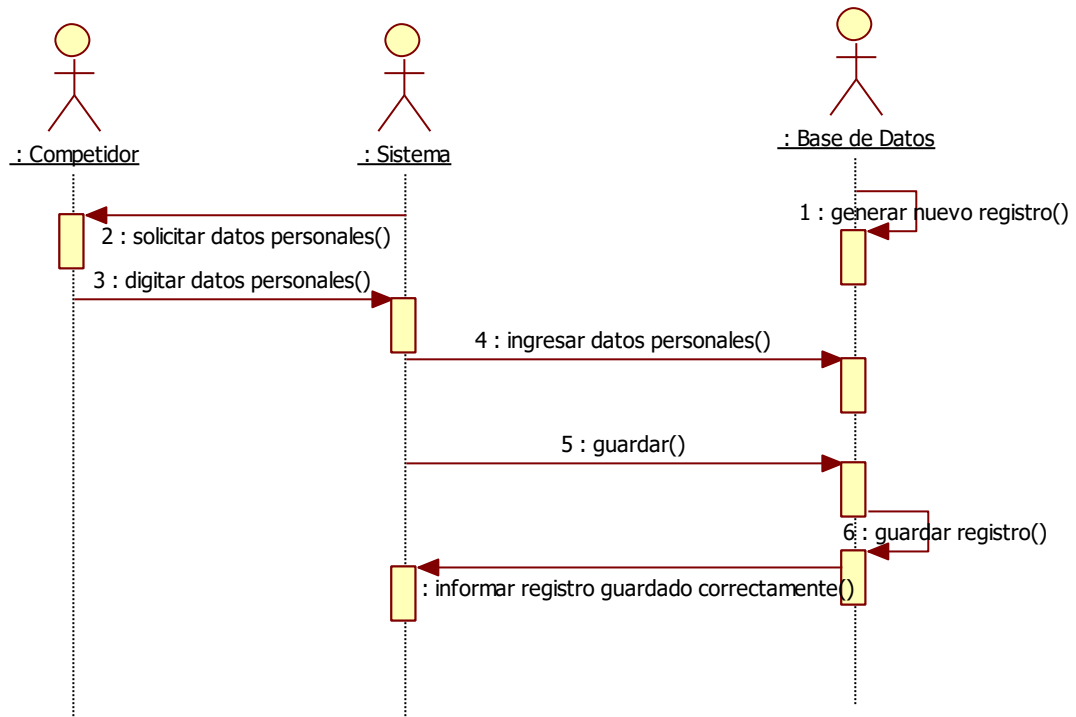
Tabla 1. Descripción de las Entidades

ENTIDAD	DESCRIPCIÓN
inscrito	Contiene la información de todos los usuarios registrados en el evento.
Categoría	Contiene las categorías establecidas para una prueba atlética.
prueba	Contiene la información de la prueba atlética.
resultado	Contiene los datos de los participantes en una prueba atlética.
punto_control	Contiene los puntos de control asignados a una prueba o tipo de prueba atlética.
categoria_prueba	Contiene la información relacionada de las entidades categoría y prueba.
inscrito_categoria_prueba	Contiene la información de los inscritos en una categoría a una prueba atlética.
tipo_prueba	Contiene los tipos de pruebas que pueden generarse en un evento deportivo.
inscrito_asistente_punto_control	Contiene la información de los inscritos que asistieron a una prueba y tienen registro en determinado punto de control.

**7.3.2.3 Modelo de Procesos del Sistema.** Para una mejor interpretación de los modelos de los procesos del sistema, en la figura 19 podremos ver el diagrama de secuencia del módulo de inscripción, se realizaron los diagramas de secuencias necesarios para cada caso de uso, en los que se explica con detalle los pasos para el funcionamiento de cada uno de los servicios. (Ver Anexo E).

Servicio de Inscripción.

Figura 20. Diagrama de Secuencia: Inscripción.



1. El controlador módulo inscripción despliega la interfaz en vista Inscripción.
2. El usuario registra sus datos en el formulario de inscripción.
3. En el controlador Inscripción se validan los datos ingresados del tipo de inscrito.
4. La base de datos retorna un mensaje de confirmación.

#### 7.3.2.4 Implementación, Implantación y Pruebas Generales.

Para la implementación de los servicios se utilizaron las siguientes herramientas:

1. Lenguaje de marcado para la elaboración de páginas web, HTML.
2. Lenguaje de programación orientado a la web, JSP.
3. Lenguaje de programación interpretado orientado a objetos, JavaScript.
4. Hojas de estilo en cascada, XML, CSS.
5. Lenguaje Java.
6. JBOSS Studio, IDE para desarrollar las clases de Java y los archivos JSP.
7. Servidor Faisan.
8. Manejador de Base de datos Informix.

Haciendo uso de las anteriores herramientas se diseñaron los módulos y se estructuraron los datos para el primer prototipo de la versión XXV de la carrera atlética de la Universidad de Santander. También se tomaron en cuenta sugerencias recibidas por miembros de la DSI, para realizar un posterior refinamiento a las interfaces.

Para el desarrollo se trabajó con la base de datos "pruebas\_atleticas" ubicada en el servidor FAISAN y con la base de datos centralizada.

Se llevaron a cabo pruebas para cada subsistema propuesto, verificando que los resultados correspondieran con lo esperado, de esta forma, se evidenció el correcto funcionamiento en la captura de datos, selección de ítems y almacenamiento de información.

La implantación de los servicios se realizó en primer lugar en el servidor local donde se realizan las pruebas a todos los componentes. Finalmente, se implementó este nuevo sistema de información para el acceso público en el servidor FAISAN.

## **7.4 MANTENIMIENTO Y ADMINISTRACIÓN**

### **7.4.1 Actividades de Soporte a Usuarios.**

Las DSI cuentan en sus sistemas de información con un servicio de consultas y sugerencias, a través del cual se responden preguntas y se resuelven problemas de los usuarios. También se resuelven problemas de manera presencial, donde el usuario acude directamente a las oficinas de la DSI ubicadas en el CENTIC dentro de la Universidad industrial de Santander y es atendido directamente por alguno de sus miembros, que le prestan la asesoría necesaria.

## **8. PRUEBAS DEL SISTEMA**

Para garantizar el correcto desarrollo de los servicios realizados, se realizaron las siguientes pruebas:

### **8.1 PRUEBAS DE VERIFICACIÓN**

Esta prueba es una de las más prácticas y más utilizadas mediante la cual se aplican diferentes técnicas para detectar errores en el sistema antes de ser utilizado. Se efectúa elaborando actividades para cada módulo desarrollado, realizando pruebas como: verificar que los campos que contienen datos obligatorios no queden vacíos, o examinar que los campos contengan el tipo de dato que le corresponde.

**8.1.1 Pruebas por componente.** Esta prueba se realizó para cada caso de uso de cada módulo desarrollado y descrito anteriormente.

Los resultados obtenidos con la realización de las pruebas, evidencian el cumplimiento de los objetivos propuestos, en la medida en que los servicios y mejoras funcionan correctamente.

### **8.2 PRUEBAS DE INTEGRACIÓN**

Una vez creados todos los servicios se verificó que cada uno funcionara correctamente y no interfirieran con el funcionamiento del portal de la Universidad Industrial de Santander. Se verificó que las consultas a la base de datos se hicieran de manera correcta, que en el sistema se visualizaran correctamente los servicios y que cada tipo de usuario tuviera acceso a ellos dependiendo del perfil y categoría al que está asociado.

### **8.3 PRUEBAS DE VALIDACIÓN**

Con el fin de probar el funcionamiento de cada servicio implementado se realizaron pruebas de validación elaboradas por integrantes de la DSI, administrativos y el director de proyecto. Para algunos de los módulos se observaron irregularidades las cuales fueron corregidas durante la marcha hasta obtener los resultados requeridos.

## **9. CONCLUSIONES**

1. El proyecto realizado va a contribuir de manera favorable, a la medición de tiempos de respuesta en la generación de resultados, control de participantes de la Carrera Atlética anual de la Universidad Industrial de Santander.
2. La implementación del registro en línea, va a contribuir a que los usuarios se integren y puedan junto a personas asignadas realizar inscripciones a eventos de la Universidad.

### Beneficios como estudiante

3. El aprendizaje de las herramientas para el diseño y desarrollo de este proyecto aplique lo aprendido en las diferentes asignaturas vistas en el transcurso de pregrado, aportando experiencia para mi vida profesional.
4. La satisfacción personal de contribuir con mi aporte utilizando los conocimientos adquiridos a la comunidad universitaria.

### Beneficios institucionales

5. Con el trabajo en equipo de diferentes miembros de la comunidad universitaria se aplica la ciencia y tecnología para este evento de trascendencia institucional.

## **10. RECOMENDACIONES**

1. Para mantener una buena comunicación con los usuarios de los portales, es fundamental dar pronta y precisa respuesta a las sugerencias de los mismos. De esta manera se logra dar soporte oportuno a sus inquietudes.
2. Se recomienda capacitar a los usuarios para que utilicen de una manera óptima los nuevos servicios, y con ellos ahorren tiempo y agilicen sus tareas relacionadas a los mismos.
3. Ampliación del sistema con los módulos que fueron excluidos del proyecto inicial.
4. Compromiso del Departamento de Deportes de apropiarse del sistema para ser usado en las competencias que realiza la Universidad Industrial de Santander.
5. Socialización del software entre los miembros de la comunidad universitaria y continuidad en este proyecto.
6. La utilización de los recursos adquiridos por el Departamento de Deportes en este proyecto para posteriores eventos.

## BIBLIOGRAFÍA

FDI . El patrón Modelo-Vista-Controlador (MVC). [En línea].  
<https://www.fdi.ucm.es/profesor/jpavon/poo/2.14.mvc.pdf> [citado el 06 Enero 2017]

JOHNSON, James. Bases de datos: Modelos lenguajes y diseño. 1ª ed. Oxford, 2000. Presenta temas de teoría de bases de datos: modelos y métodos de acceso, administración, diseño de aplicaciones.

ORACLE. Api de Java [en línea]. <http://docs.oracle.com/javase/7/docs/api/> [citado el 06 Enero 2017]

ORACLE CORPORATION. MySQL Documentation: MySQL Reference Manuals.[en línea].  
<http://dev.mysql.com/doc/> [citado el 06 Enero 2017]

PIATTINI, Mario, CALVO-MANZANO, José A., CERVERA, Joaquín, FERNANDEZ, Luis. Análisis y diseño detallado de Aplicaciones Informáticas de Gestión. Alfaomega, 2000.

PONCE DE LEÓN, Diego. Tutoriales HTML. [En línea]. <http://www.htmlquick.com/es/tutorials.html>  
[citado el 01 Mayo de 2016]

RICH, Clark, et al. Html element index [online]. Available from Internet:  
<http://html5doctor.com/element-index/> [citado el 03 Abril de 2016]

SCHMULLER, Joseph. Aprendiendo UML en 24 horas. 1ª ed. México: Alhambra Mexicana S.A, 2000.

STALLINGS, William. Sistemas Operativos: Aspectos Internos y Principios de Diseño. Madrid: Pearson Prentice Hall, 2005.

STUMPF, Robert, TEAGUE, Lavette. Object-Oriented Systems Analysis and Design with UML. Prentice Hall. 2004.

Componentes de interfaz de usuario dinámica e interactiva para crear aplicaciones. [En línea] [Bucaramanga, Colombia]: Elise Library V2. Noviembre 2016

WEITZENFELD, Alfredo. Ingeniería de Software Orientada a Objetos con UML, JAVA e Internet. Thomson International, 2003.

JBOSS . JBoss Application Server Official Documentation. [En línea]. <https://developer.jboss.org> [citado el 01 Mayo de 2016]

IMPINJ, «LTK Programmers Guide,» IMPINJ, 2002-2016. [En línea]. <http://www.impinj.com>. [Citado 06 Enero 2017].

IMPINJ, «Antena Hub,» IMPINJ, Abril 2002-2016. [En línea]. <http://www.impinj.com>. [Citado 06 Enero 2017].

RUNNING COLOMBIA, Calendario temporada 2017 [En línea] <http://runningcolombia.com>. [Citado 25 Enero 2017].

VILLAFRADES GONZÁLEZ, Fabio. Reseña histórica 25 años Carrera Atlética 1988-2015 UIS, 2015.

# ANEXOS

## Anexo A. Tecnologías de desarrollo de páginas web dinámicas

### TECNOLOGÍAS DE DESARROLLO DE PÁGINAS WEB DINÁMICAS

#### 1. CÓDIGO DEL LADO DEL CLIENTE (CLIENT SIDE SCRIPTS)

Código ejecutado por los navegadores, el cual los computadores clientes tienen instalados. Las tecnologías más comunes de este tipo son:

1. *JavaScript*: Lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C. Permite, crear ventanas, mostrar texto en movimiento y verificar las entradas a un formulario.
2. *Controles Activos*: tecnología Microsoft que permite la creación de aplicaciones Windows, como pueden ser Visual Basic Script o Visual C. Es la respuesta de Microsoft a los Applets de Java.
3. *Java Applets*: Programas escritos en lenguaje de programación Java, se incrustan en HTML y se ejecutan en el navegador gracias a la Máquina Virtual de Java (JVM) que lleva éste incorporado.

#### 2. CÓDIGO DEL LADO DEL SERVIDOR (SERVER SIDE SCRIPTS)

Código que se ejecuta en el servidor. Para su actividad el programa ejecuta y procesa los datos o peticiones que el usuario envía desde su navegador, para luego enviar los resultados del programa en una página HTML que el usuario verá normalmente en su navegador. Los más usados son:

1. ASP (Active Server Pages): Permite crear dinámicamente páginas Web mediante HTML, scripts, y componentes de servidor ActiveX reutilizables, requiere de un computador configurado como Servidor Web de Microsoft (Microsoft Web Server), el navegador del cliente es indiferente pues el trabajo se realiza del lado del servidor. De gran uso en la gestión de Bases de Datos ya que puede conectarse a SQL, Access, Oracle u otras.
2. PHP (PHP Hypertext Pre-processor): Lenguaje de programación interpretado, diseñado para la creación de páginas web dinámicas. Es un lenguaje de código abierto (Open Source) y gratuito. Su gran potencia se encuentra en la interacción con los motores de bases de datos como Oracle y MySQL.
3. JSP (Java Server Pages): tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo. Permiten la utilización de código Java mediante scripts.

## **Anexo B. Bases de datos**

### **BASES DE DATOS**

#### **1. MODELOS DE BASES DE DATOS**

Este modelo describe la estructura de una base de datos, donde se incluye conceptos como: entidades, atributos y relaciones, la mayoría de los modelos de datos poseen un conjunto de operaciones básicas para especificar consultar y actualizaciones. Las bases de datos se pueden clasificar de acuerdo a su modelo de administración de datos. Algunos modelos utilizados con frecuencia son:

##### **1.1 BASE DE DATOS JERÁRQUICA**

Estas bases de datos almacenan su información en una estructura escalonada, organizando los datos en forma similar a un árbol (visto al revés), en donde un nodo padre de información puede tener varios hijos, el nodo que no tiene padres es llamado raíz, y a los nodos que no tienen hijos se les conoce como hojas. Las bases de datos jerárquicas son especialmente útiles en el caso de aplicaciones que manejan un gran volumen de información y datos muy compartidos permitiendo crear estructuras estables y de gran rendimiento. Está limitado por su incapacidad de representar eficientemente la redundancia de datos.

##### **1.2 BASE DE DATOS DE RED**

En este modelo se permite que un mismo nodo tenga varios padres. Ofrece una solución eficiente al problema de redundancia de datos; sin embargo, la dificultad para administrar los datos en una base de datos de red ha conllevado a que sea un modelo usado más por programadores que por usuarios finales.

##### **1.3 BASE DE DATOS RELACIONAL**

Es el más utilizado para modelar problemas reales y administrar datos dinámicamente. Su fundamento es el uso de "relaciones". Estas relaciones podrían considerarse en forma lógica como conjuntos de datos, también llamados tuplas. Cada relación es una tabla que está compuesta por registros (las filas de una tabla), que representan las tuplas, y campos (las columnas de una tabla). Los datos pueden ser recuperados o almacenados mediante "consultas" que ofrecen una amplia flexibilidad y poder para administrar la información. El lenguaje más habitual para construir las consultas a bases de datos relacionales es el Lenguaje Estructurado de Consultas (Structured Query Language, SQL), un estándar implementado por los principales manejadores de bases de datos relacionales.

## 2. ACCESO A BASE DE DATOS

Para desarrollar aplicaciones que conecten bases de datos, se utilizan interfaces y programas estándar que envían demandas escritas en SQL, y procesan los resultados. Para conectarse a un motor de bases de datos determinado, se necesita un driver o una interfaz estándar que medie entre la aplicación y la base de datos.

## 3. CONECTORES MÁS UTILIZADOS

1. ODBC (Open DataBase Connectivity): Estándar de acceso a Bases de datos, su objetivo es hacer posible el acceder a cualquier dato desde cualquier aplicación, sin importar qué Sistema Gestor de Bases de Datos (DBMS por sus siglas en inglés) almacene los datos, logra esto al insertar una capa intermedia llamada manejador de Bases de Datos, entre la aplicación y el DBMS, el propósito de esta capa es traducir las consultas de datos de la aplicación en comandos que el DBMS entienda. Para que esto funcione tanto la aplicación como el DBMS deben ser compatibles con ODBC.
2. MDB: Esta aplicación permite trabajar con tablas de base de datos creadas en Access 97/2000. Es posible abrir tablas en SQL, visualizarlas, navegar, crear y borrar índices, fijar relaciones, copiar, etc.
3. JDBC (Java Database Connectivity): Interfaz de programación de aplicaciones que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java, independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede, utilizando el dialecto SQL del modelo de base de datos que se utilice.

Para el desarrollo de los portales y cada uno de sus módulos se empleó el conector JDBC. Uno de sus mayores beneficios es su capacidad para crear aplicaciones cuya programación sea independiente de la base de datos, es decir, pueden ser migradas a otro servidor de bases de datos sin complicaciones. Sin embargo, dos elementos siguen estando ligados a una base de datos en particular, el nombre de la clase que se usa para cargar el controlador JDBC y la dirección URL (Universal Resource Locator) para acceder a la base de datos.

Los servlets y las páginas JSP usan JDBC habitualmente de la misma manera que cualquier otra aplicación en Java, típicamente los datos del controlador JDBC, la cadena de conexión, el nombre de usuario y contraseña para conectarse a la base de datos son codificados dentro del programa.

Las operaciones primordiales realizadas durante la ejecución del JDBC son:

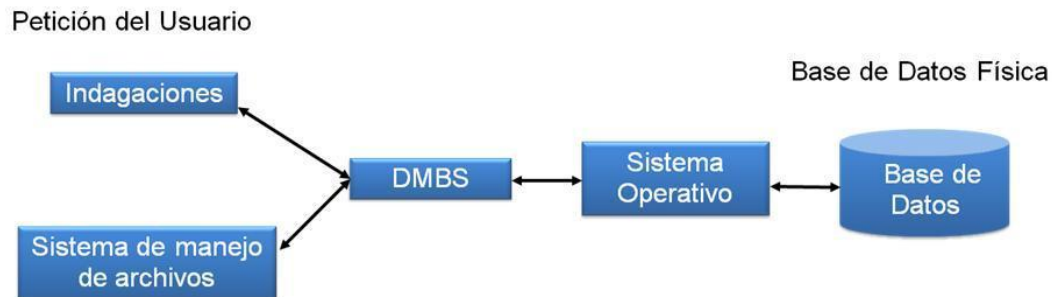
1. Cargar un controlador JDBC.
2. Utilizar el controlador para abrir una conexión con la base de datos.
3. Formular instrucciones SQL a través de la conexión.
4. Procesar los conjuntos de resultados devueltos por las operaciones SQL.

#### 4. MANEJADORES O GESTORES DE BASES DE DATOS

El sistema manejador de bases de datos es la porción más importante del software de un sistema de base de datos. Un DBMS es una colección de numerosas rutinas de software interrelacionadas, cada una de las cuales es responsable de alguna tarea específica.

Las funciones principales de un DBMS son:

1. Crear y organizar la Base de Datos.
  2. Establecer y mantener las trayectorias de acceso a la base de datos de tal forma que los datos puedan ser capturados rápidamente.
  3. Manejar los datos de acuerdo a las peticiones de los usuarios.
  4. Registrar el uso de las bases de datos.
  5. Interacción con el manejador de archivos a través de las sentencias en Lenguaje Manipulador de Datos (Data Manipulation Language, DML) al comando del sistema de archivos.
  6. Respaldo y recuperación: Consiste en contar con mecanismos implantados que permitan la recuperación fácilmente de los datos en caso de ocurrir fallas en el sistema de base de datos.
  7. Control de concurrencia: consiste en controlar la interacción entre los usuarios concurrentes para preservar la consistencia de los datos.
- 
1. Seguridad e Integridad: consiste en contar con mecanismos que permitan el control de la consistencia de los datos evitando que estos se vean perjudicados por cambios no autorizados o previstos.



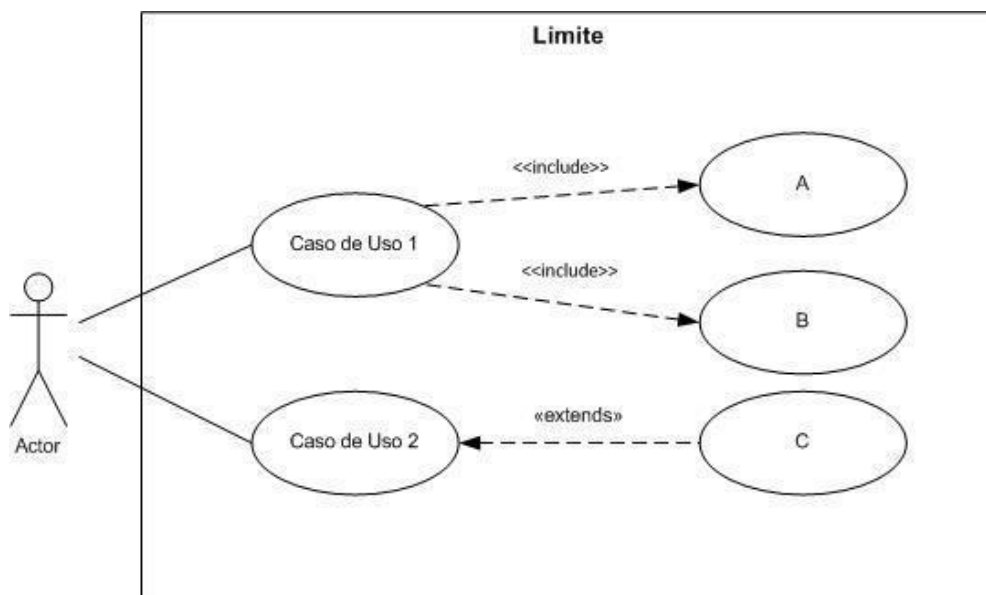
Función del DBMS

La figura muestra el DBMS como interfaz entre la base de datos física y las peticiones del usuario. El DBMS interpreta las peticiones de entrada/salida del usuario y las manda al sistema operativo para la transferencia de datos entre la unidad de memoria secundaria y la memoria principal. Es decir el DBMS es el corazón de la base de datos ya que se encarga del control total de los posibles aspectos que la puedan afectar.

## Anexo C. Leguaje de modelado unificado (UML)

### LEGUAJE DE MODELADO UNIFICADO (UML)

#### 1. DIAGRAMAS DE CASOS DE USO



Diagramas de Casos de Uso

**Actores:** Un actor en un caso de uso representa un rol, que alguien o algo puede desempeñar dentro un sistema y no un alguien o algo específico.

En este proyecto se destacan las siguientes clases de actores:

1. **Administradores:** Son usuarios que además de pertenecer a la categoría de usuarios tienen un perfil de administrador, con el cual pueden desempeñar ciertas labores diferentes a un usuario normal dentro del sitio. Estos son: los Auxiliares de administración del sistema de información, organizadores del evento, secretaria con ciertos privilegios. Dentro de esta categoría se incluye también el súper administrador.
2. **Usuario deportista:** Es el tipo de usuario común del sitio y a quien van dirigidos los servicios. Este usuario solo tiene control sobre sus privilegios.
3. **Sistema:** Es el sistema encargado de realizar determinadas funciones, que le solicita el usuario.

**Inclusión (include–uses):** Es una forma de interacción, un caso de uso dado puede "incluir" otro. Una inclusión es utilizada para indicar que un caso de uso depende de otro, es decir, la funcionalidad de determinado caso de uso se requiere para realizar las tareas de otro. En la figura 7 el caso de uso "Caso de uso 1" depende de los casos de uso "A" y "B".

**Extensión (extend):** Es otra forma de interacción, una extensión representa una variación de un caso de uso a otro, es decir, una dependencia específica entre los casos de uso, a través de la cual un caso de uso puede extender a otro.

## 2. DIAGRAMAS DE SECUENCIAS

**Objetos:** Se ubican en la parte superior del diagrama de izquierda a derecha y se acomodan de manera que simplifiquen al diagrama. La línea que está debajo de cada objeto será una línea discontinua conocida como la *línea de vida* de un objeto. Con la línea de vida se encuentra un pequeño rectángulo conocido como *activación*, el cual representa la ejecución de una operación que realiza el objeto.

**Mensaje:** Un mensaje que va de un objeto a otro pasa la línea de vida de un objeto a otro. Un objeto puede enviarse un mensaje a sí mismo. Un mensaje puede ser simple, sincrónico o asincrónico.

**Tiempo:** El diagrama representa al tiempo en dirección vertical. Inicia en la parte superior y avanza hacia la parte inferior. Un mensaje que esté más cerca de la parte superior ocurrirá antes que uno que esté cerca de la parte inferior.

**GUI:** (Siglas en Ingles) La interfaz gráfica de usuario; es la interfaz de interacción del usuario y en la que más interactividades se presentan con otros objetos.

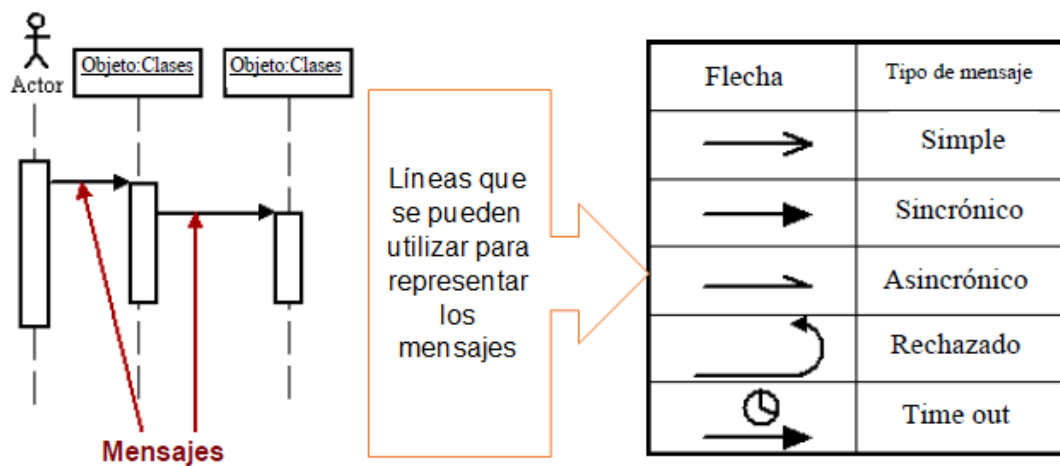


Diagrama de Secuencias

## Anexo D. Documentación de casos de uso del sistema

### DOCUMENTACIÓN DE CASOS DE USO DEL SISTEMA

#### 1. INSCRIPCIÓN

**Caso de uso: realizar inscripción.**

<b>TÍTULO</b>	<b>FUNCIONES PRIMARIAS</b>
Inscripción	Muestra formulario de inscripción.
Editar	Permite editar un registro de la tabla inscripción.
Eliminar	Permite eliminar un registro seleccionado.
Ayuda	Muestra información para administrar la inscripción.
Listar	Despliega un listado de los registros de la tabla de inscripción.

#### 2. ADMINISTRACIÓN

**Caso de uso: Administrar Prueba.**

<b>TÍTULO</b>	<b>FUNCIONES PRIMARIAS</b>
Administrar Prueba	Muestra tabla soporte de pruebas.
Editar	Permite editar un registro de la tabla prueba.
Eliminar	Permite eliminar un registro seleccionado.
Ayuda	Muestra información para administrar las pruebas.
Listar	Despliega un listado de los registros de la tabla de prueba.

**Caso de uso: Administrar Categoría.**

<b>TÍTULO</b>	<b>FUNCIONES PRIMARIAS</b>
Administrar Categoría	Muestra tabla soporte de las categorías.
Editar	Permite editar un registro de la tabla categoría.
Eliminar	Permite eliminar un registro seleccionado.
Ayuda	Muestra información para administrar las categorías.
Listar	Despliega un listado de los registros de la tabla de categoría.

### **Caso de uso: Administrar Punto de control.**

<b>TÍTULO</b>	<b>FUNCIONES PRIMARIAS</b>
Administrar PuntoControl	Muestra tabla soporte de los puntos de control.
Editar	Permite editar un registro de la tabla punto_control.
Eliminar	Permite eliminar un registro seleccionado.
Ayuda	Muestra información para administrar los puntos de control.
Listar	Despliega un listado de los registros de la tabla de punto_control.

### **Caso de uso: Administrar CategoriaPrueba.**

<b>TÍTULO</b>	<b>FUNCIONES PRIMARIAS</b>
Administrar CategoriaPrueba	Muestra tabla soporte de las categorías que se asociarán con alguna prueba.
Editar	Permite editar un registro de la tabla categoría_prueba.
Eliminar	Permite eliminar un registro seleccionado.
Ayuda	Muestra información para administrar las categorías relacionadas con pruebas.
Listar	Despliega un listado de los registros de la tabla de categoría_prueba.

### **Caso de uso: Administrar TipoPrueba.**

<b>TÍTULO</b>	<b>FUNCIONES PRIMARIAS</b>
Administrar TipoPrueba	Muestra tabla soporte de los tipos de pruebas.
Editar	Permite editar un registro de la tabla tipo_prueba.
Eliminar	Permite eliminar un registro seleccionado.
Ayuda	Muestra información para administrar los tipos de pruebas.
Listar	Despliega un listado de los registros de la tabla de tipo_prueba.

## **3. RESULTADOS**

### **Caso de uso: Consultar resultado.**

<b>TÍTULO</b>	<b>FUNCIONES PRIMARIAS</b>
Resultado	Muestra tabla de resultados con opción de búsqueda.
Resultado por categoría	Permite seleccionar una tabla de resultados por categoría seleccionada.
Resultado por género	Permite seleccionar una tabla de resultados por género seleccionada.
Resultado por ID	Muestra el resultado de un registro por identificación.

## Anexo E. Diagramas de secuencia

### DIAGRAMAS DE SECUENCIA

#### 1. Administración

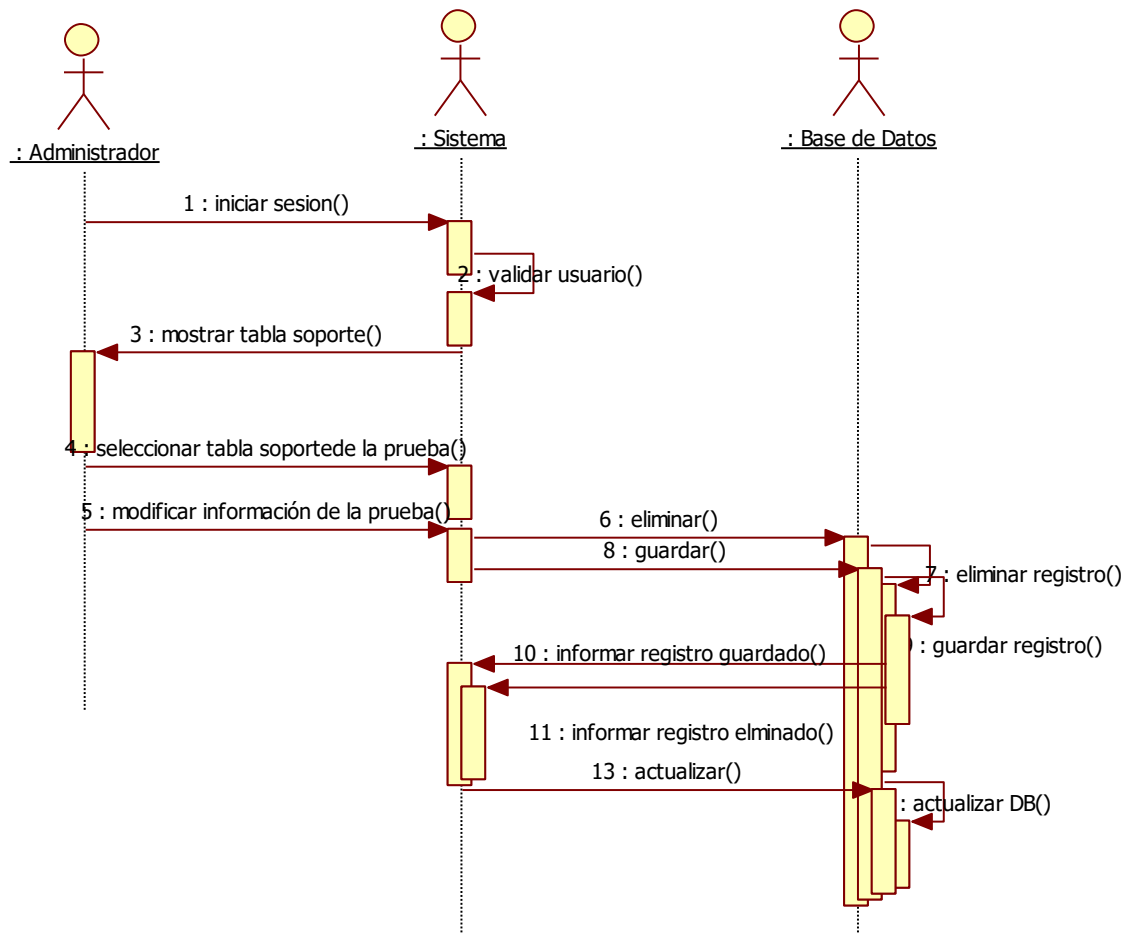
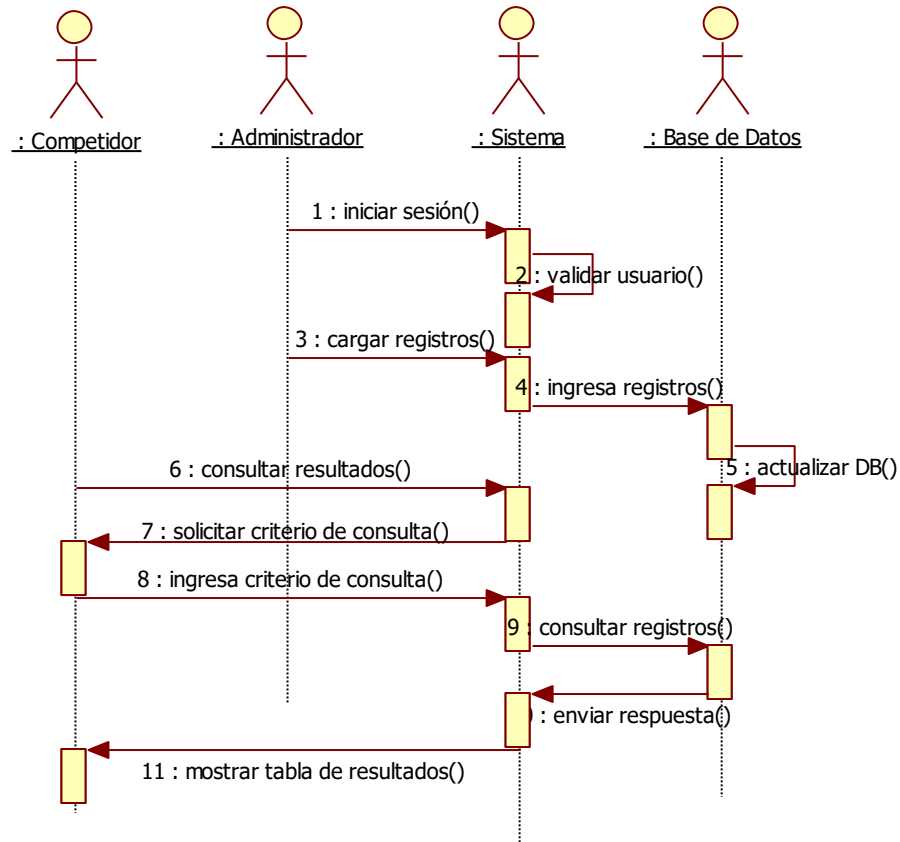


Diagrama de Secuencia: Administración.

1. El usuario administrador ingresa al Sistema de Información de la Carrera Atlética.
2. El sistema valida el usuario.
3. En el controlador Administración despliega menú administrador con tablas soporte de la prueba.
4. El administrador selecciona y modifica la tabla de la prueba.
5. El sistema guarda o elimina el registro seleccionado.
6. La base de datos es modificada y el sistema despliega mensaje de la actividad realizada.
7. La base de datos es actualizada.

## 2. Resultados

Diagrama de Secuencia: Consultar Resultado.



1. El usuario administrador ingresa al Sistema de Información de la Carrera Atlética.
2. El sistema valida el usuario.
3. El Administración carga los registros de tiempos.
4. El sistema ingresa los registros de tiempos a la base de datos.
5. La base de datos es actualizada.
6. El usuario competidor ingresa al módulo de consultar resultados.
7. El sistema solicita criterios de consulta de resultados.
8. El usuario competidor selecciona e ingresa los datos requeridos para la consulta.
9. El sistema consulta el registro a la base de datos.
10. La base de datos responde a la consulta.
11. El sistema muestra la tabla de los resultados solicitados por el usuario.