

**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**

**Facultad 09**



**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE  
INGENIERO EN CIENCIAS INFORMÁTICAS**

**TÍTULO:** Implementación del sistema automatizado GESTAPRO  
para el control de los recursos humanos en los polos productivos de  
la Facultad 9.

**AUTOR:** Cheyner Pérez Barreda

**TUTOR:** Ing. Adrián Gracia Aguila

**CO-TUTOR:** Ing. Yudiel Rodríguez Larrazabal

**CONSULTOR:** Ing. Kizzy Yinet Labrada Escalona

**Ciudad de la Habana, 1ro de julio del 2010**

**Año 52 de la Revolución**

## Pensamiento

*“La Revolución se lleva en el corazón, no en la boca para vivir de ella”*

*Ché*

## Dedicatoria

A mi abuela Unga.

A mis padres.

A la Revolución.

## Agradecimientos

A la Revolución y a Fidel por darme la posibilidad de convertirme en un profesional en esta excepcional universidad.

A mis padres por su cariño, dedicación, apoyo y ser mis guías en todos los años de mi vida.

A mi abuela Unga, que con su deseo de verme graduado, me impulsó a esforzarme cada día más.

A mi novia Daina por ser tan especial, por su amor, comprensión y ayuda en todo momento.

A mi hermana por ser mi ejemplo a seguir y darme consejos oportunos.

A mis segundos padres, Lidia y Juan, por poder contar con ellos siempre.

A mi tía, por ser como una madre para mí y brindarme toda su ayuda.

A toda mi familia por depositar su confianza en mí.

A Manuel Alejandro, el Chino, por ser un amigo incondicional.

A mi tutor, Adrián, por ayudarme en estos últimos meses y ser, además, un amigo.

A todos mis amigos, por compartir todos estos años juntos.

A la Universidad de las Ciencias Informáticas por formar profesionales integrales.

A todas las personas involucradas en el desarrollo de esta tesis,

Muchas Gracias.

## Declaración de autoría

Declaro que soy el único autor de este trabajo y autorizo a la Facultad 9 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

**Cheyner Pérez Barreda**

**Autor**

---

**Adrián Gracia Aguila**

**Tutor**

## Datos de contacto

Tutor: Ing. Adrián Gracia Aguila.

Sexo: Masculino.

Institución: Universidad de las Ciencias Informáticas.

Dirección de la institución: Carretera a San Antonio de los Baños, Km. 2 ½, Reparto:  
Torrens, Municipio: Boyeros, Provincia: Ciudad de La Habana.

Correo electrónico: agracia@uci.cu

Teléfono del trabajo: 835-8907

Categoría docente: Adiestrado.

Cargo del trabajador: Profesor.

Título de la especialidad de graduado: Ingeniero en Ciencias Informáticas.

Año de graduación: 2009.

Institución donde se graduó: Universidad de las Ciencias Informáticas.

## Resumen

Con la creación de la Universidad de las Ciencias Informáticas (UCI) se puso en práctica el principio de vincular al estudiantado con la producción trayendo consigo la creación de proyectos productivos con fines comerciales o docentes. Con el objetivo de agrupar estos proyectos teniendo en cuenta su tema de trabajo, en el año 2007, se crean los polos productivos trabajando en ellos una gran cantidad de personas de las cuales se necesita tener un control.

A la hora de controlar a dicho personal se genera un gran cúmulo de información, la cual generalmente es digital pero no existe una centralización y organización de la misma por lo que puede ser manipulada de forma errónea, puede no estar disponible en algún momento para su consulta y en algunos casos pueden existir pérdidas.

La presente investigación consiste en la realización de las actividades propias del rol de implementador con el objetivo de obtener el sistema automatizado GESTAPRO para la gestión de los recursos humanos de los polos productivos de la Facultad 9.

### **PALABRAS CLAVES**

Implementador, sistema automatizado GESTAPRO, Gestión de los Recursos Humanos.

## Índice

Capítulo 1 Fundamentación Teórica .....	5
1.1 Introducción.....	5
1.2 Conceptos asociados al dominio del problema.....	5
1.3 Paradigmas de Programación. ....	7
1.4 Lenguajes de Programación. ....	8
1.4.1 PHP .....	9
1.4.2 HTML.....	11
1.4.3 CSS.....	12
1.4.4 JavaScript .....	13
1.5 Sistemas Gestores de Bases de Datos.....	13
1.5.1 PostgreSQL.....	17
1.6 Servidor Web.....	19
1.6.1 Servidor Apache.....	19
1.7 Herramientas de desarrollo.....	20
1.7.1 Zend Studio .....	21
1.7.2 Dreamweaver.....	21
1.8 Metodologías de Desarrollo de Software .....	23
1.9 Conclusiones Parciales.....	24
Capítulo 2: Construcción del sistema.....	25
2.1 Introducción.....	25
2.2 Valoración crítica del diseño propuesto por el analista.....	25
2.2.1 Requisitos Funcionales.....	25
2.2.2 Requisitos No Funcionales. ....	28
2.2.3 Artefactos Obtenidos. ....	29
2.2.4 Arquitectura del sistema. ....	31
2.3 Diagramas de Componentes. ....	31
2.3.1 CU Autenticar.....	32
2.3.2 CU Gestionar Polo. ....	32
2.4 Estándares de Codificación. ....	35
2.5 Descripción de las principales clases y sus funcionalidades. ....	37
2.6 Conclusiones parciales.....	41
Capítulo 3: Validación de la solución propuesta. ....	42
3.1 Introducción.....	42
3.2 Descripción general de las pruebas de software. ....	42
3.3 Diseño de los casos de prueba para la validación de la solución. ....	42



3.3.1	Gestionar Polo. ....	43
3.3.2	Buscar Polo.....	47
3.3.3	Gestionar Rol. ....	49
3.3.4	Buscar Rol.....	56
3.4	Conclusiones parciales.....	57
	Conclusiones.....	58
	Recomendaciones .....	59
	Referencias Bibliográficas .....	60
	Bibliografía .....	63
	Glosario de Términos.....	67

## Introducción

Desde los inicios de la humanidad el hombre necesitaba interactuar con otros individuos en la realización de tareas que para su cumplimiento era imprescindible un cierto número de fuerza de trabajo. Cada una de estas tareas tenía un líder que era la persona encargada de controlar el trabajo a realizar y a las personas que se encontraban bajo su mando. Al principio, solo eran pequeñas labores pero con el aumento de la complejidad de las tareas y el desarrollo tecnológico alcanzado por la sociedad, el número de personas necesarias para estas actividades fue creciendo y la tarea de controlarlas se hizo más difícil.

Por esta problemática surgió el término de Recursos Humanos (a partir de ahora RRHH) el cual hace referencia al proceso de selección, formación, desarrollo y consecución de las personas cualificadas necesarias para conseguir los objetivos de la organización (1). Este término se desarrolló de tal manera que en las empresas y centros de trabajo se crearon departamentos con la finalidad de gestionar el personal de trabajo, lo que ha permitido obtener y coordinar a las personas de una organización, para alcanzar las metas establecidas.

Debido a la creación de las TIC's y su dinámica y rápida evolución, todas las esferas de la sociedad han sufrido cambios, ya que éstas facilitan la labor del hombre a la hora de desarrollar dichas actividades, además, a través de ellas se puede lograr la comunicación entre personas distantes entre sí. La gestión de RRHH no ha quedado exenta del desarrollo alcanzado por las TIC's y de esta manera han surgido sistemas automatizados para la gestión de RRHH en todo el mundo obteniéndose un mejor control sobre dichos recursos.

Desde la creación de la Universidad de las Ciencias Informáticas (a partir de ahora UCI) se comenzaron a desarrollar un conjunto de proyectos con el objetivo de producir sistemas de software, con fines comerciales o docentes. Estos proyectos se fueron agrupando en dependencia de sus objetivos y de esta forma se crearon en el año 2007 los polos productivos. Debido al gran número de personal que labora en dichos polos se hace necesario su control y el de las tareas que realizan en los proyectos de la Facultad 9. Como consecuencia de este control se genera mucha información la cual básicamente es digital pero no existe una centralización y organización de la

misma, por lo que puede ser manipulada de forma errónea, no estar disponible en algún momento para su consulta y en otras ocasiones han existido pérdidas por esta desorganización.

Teniendo en cuenta la problemática anterior se define como **problema a resolver**: El manejo inadecuado de los datos en la gestión de los recursos humanos en los polos productivos de la Facultad 9; para poder dar solución al problema anterior se hace necesario el estudio del proceso de gestión de los recursos humanos en la UCI siendo el **campo de acción** la gestión de los recursos humanos en la Facultad 9.

Se persigue el siguiente **objetivo general**: Implementar un sistema informático para la gestión de los recursos humanos de los polos productivos de la Facultad 9.

Por todo lo planteado la **idea a defender** es que si se implementa una aplicación web para gestionar los recursos humanos de los polos productivos de la Facultad 9 se va a lograr un mejor control del personal perteneciente a estos polos y de las actividades que realizan.

Para lograr el objetivo expuesto anteriormente y confirmar la anterior afirmación se desarrollarán las siguientes **tareas de investigación**:

1. Definición del marco teórico de la investigación.
2. Identificación de los estándares de codificación para la implementación del sistema informático GESTAPRO.
3. Descripción de los estándares de codificación para la implementación del sistema informático GESTAPRO.
4. Definición de las herramientas y el lenguaje de programación que se usarán para el desarrollo del sistema informático GESTAPRO.
5. Implementación del sistema automatizado GESTAPRO para la gestión de los recursos humanos de los polos productivos de la Facultad 9.
6. Validación del sistema automatizado GESTAPRO para la gestión de los recursos humanos de los polos productivos de la Facultad 9.

Como resultado de todas las tareas y actividades desarrolladas y además de un largo proceso de investigación se espera lo siguiente:

- Sistema informático GESTAPRO para la gestión de recursos humanos en los polos productivos.

En el transcurso y desarrollo de esta investigación científica se utilizarán un conjunto de métodos científicos para poder obtener, procesar y llegar a conclusiones. Dentro de éstos podemos mencionar los siguientes:

#### **Métodos científicos:**

##### **Teóricos:**

- **Analítico - Sintético:** Se utilizará para descomponer los elementos por separado como por ejemplo los casos de uso anteriormente diseñados y profundizar en el estudio de cada uno de estos elementos, para luego sintetizarlos y realizar la confección de la solución propuesta.
- **Histórico – lógico:** Se utilizará para el estudio del análisis y diseño realizado inicialmente y tomar este trabajo como base para el desarrollo de esta investigación.

##### **Empíricos:**

- **Observación:** Se utilizará para determinar los problemas existentes en la gestión de los recursos humanos en los polos productivos de la Facultad 9.

Este trabajo está dividido en 3 capítulos fundamentales:

**Capítulo 1: Fundamentación Teórica.** En este capítulo se abordan los conceptos asociados al problema. Se describen las principales tecnologías y metodologías existentes que se emplean para obtener la solución propuesta, además de describir las principales herramientas de desarrollo.

**Capítulo 2: Construcción del sistema.** En este capítulo se hace una valoración del diseño propuesto por el analista. Se establecen los estándares de codificación a

utilizar en el desarrollo de la aplicación. Se describen algunas clases a implementar, así como sus principales funcionalidades.

**Capítulo 3: Validación de la solución propuesta.** En este capítulo se aborda el tema sobre la construcción de casos de pruebas para determinar si el sistema cumple con los requerimientos.

# Capítulo 1 Fundamentación Teórica

## 1.1 Introducción

En este capítulo se enuncian conceptos asociados al problema así como la descripción de las metodologías de desarrollo de software, herramientas de desarrollo y lenguajes de programación que se utilizarán para obtener la solución propuesta.

## 1.2 Conceptos asociados al dominio del problema

Para una mejor comprensión del tema durante el desarrollo de la investigación se hace referencia a algunos conceptos y términos relacionados con la misma.

### **Recursos Humanos:**

“Proceso de selección, formación, desarrollo y consecución de las personas cualificadas necesarias para conseguir los objetivos de la organización.” (1)

### **Gestión:**

“Actividad profesional tendiente a establecer los objetivos y medios de su realización, a precisar la organización de sistemas, a elaborar la estrategia del desarrollo y a ejecutar la gestión del personal.” (2)

### **Gestión de Recursos Humanos:**

“La gestión de los recursos humanos se encarga de obtener y coordinar a las personas de una organización, de forma que consigan las metas establecidas. Para ello es muy importante cuidar las relaciones humanas.” (3)

“Actividades coordinadas para dirigir y controlar una organización, que permiten materializar la política laboral, que se aplican con la participación activa y efectiva de los trabajadores en la planificación, organización, dirección, control y evaluación de los recursos humanos, que determinan o inciden en el desempeño de la organización.” (4)

### **Objetivos de la gestión de Recursos Humanos: (3)**

- Atraer a los candidatos al puesto de trabajo que estén potencialmente cualificados.
- Retener a los mejores empleados.
- Motivar a los empleados.
- Ayudar a los empleados a crecer y desarrollarse en la organización.
- Aumento de la productividad.
- Mejorar la calidad de vida en el trabajo.
- Cumplimiento de la normativa y legislación.

#### **Sistema de Gestión de Recursos Humanos:**

“Uno de los subsistemas del Sistema de Administración de la Empresa que tiene como misión: concebir, planificar, coordinar, ejecutar y controlar las políticas y procedimientos para el ingreso, permanencia, promoción y terminación de las relaciones laborales de los dirigentes, técnicos y trabajadores en la empresa, conformando un todo ordenado y coherente con relaciones afines e interacción mutua para el logro de la misión, las estrategias y los objetivos establecidos en cada nivel.”

(5)

#### **Sistema Automatizado:**

Aplicaciones o programas que se crean con el objetivo de sustituir los procedimientos manuales administrativos, que una persona realiza en una organización de forma presencial, por un software que permita realizar los mismos procedimientos de manera no presencial y reducir el esfuerzo empleado por dicha persona.

#### **GESTAPRO:**

Sistema automatizado que gestiona información de estudiantes y profesores de las áreas productivas de la Facultad 9. Lleva el control del tiempo de máquina, evalúa el desempeño y lleva el control de eventos realizados por cada uno de ellos.

### 1.3 Paradigmas de Programación.

Un paradigma de programación es un modelo básico de diseño y desarrollo de programas, que permite producir programas con unas directrices específicas, tales como: estructura modular, fuerte cohesión, alta rentabilidad, etc. (6)

También se puede decir que son una colección de modelos conceptuales que juntos modelan el proceso de diseño y determinan la estructura de un programa.

Cada paradigma es distinto a otro por lo que supone nuevas reglas, límites, elementos y maneras de pensar. Ninguno es mejor que otro ya que cada cual tiene sus ventajas y desventajas con respecto a otros pero es cierto que en determinadas situaciones existen paradigmas que son más apropiados que otros.

Dentro de los principales Paradigmas de Programación se encuentran los siguientes:

**Paradigma Imperativo:** Modelo abstracto que consiste en un gran almacenamiento de memoria donde la computadora almacena una representación codificada de un cálculo y ejecuta una secuencia de comandos que modifican el contenido de ese almacenamiento. Algoritmos + Estructura de Datos = Programa.

**Paradigmas Procedimentales:** Modelos de Desarrollo: Orientado a Objetos, a Eventos, y a Agentes. Secuencia computacional realizada etapa a etapa para resolver el problema. Su mayor dificultad reside en determinar si el valor computado es una solución correcta del problema.

**Paradigmas Declarativos:** Modelos de Desarrollo: Funcional, Lógico y de Flujo de Datos. Se construye señalando hechos, reglas, restricciones, ecuaciones, transformaciones y otras propiedades derivadas del conjunto de valores que configuran la solución.

**Paradigmas Demostrativos:** Modelos de Desarrollo Genético. Cuando se programa bajo un paradigma demostrativo (también llamada programación por ejemplos), el programador no especifica procedimentalmente cómo construir una solución sino que presentan soluciones de problemas similares.



**Paradigma Funcional:** Modelo matemático de composición funcional donde el resultado de un cálculo es la entrada del siguiente, y así sucesivamente hasta que una composición produce el valor deseado.

**Paradigma Orientado a Objeto:** Disciplina de ingeniería de desarrollo y modelado de software que permite construir más fácilmente sistemas complejos a partir de componentes individuales. Objetos + Mensajes = Programa.

#### **1.4 Lenguajes de Programación.**

Un lenguaje de programación es un lenguaje diseñado para describir el conjunto de acciones consecutivas que un equipo debe ejecutar. Por lo tanto, un lenguaje de programación es un modo práctico para que los seres humanos puedan dar instrucciones a un equipo. (7)

La computadora funciona bajo el control de programas escritos en diferentes lenguajes de programación pero estos códigos deben ser llevados a lenguaje o código de máquina para poder ser entendidos por el procesador y consisten en datos binarios, es decir, formados por 0 y 1. Además, los lenguajes codificados en una computadora específica no pueden ser ejecutados en otra computadora diferente.

El primer lenguaje de programación utilizado fue el ensamblador y el mismo es muy similar al lenguaje de máquinas pero a la vez es muy poco comprensible para los seres humanos ya que están compuestos por códigos numéricos sin sentido nemotécnico por lo que se han desarrollado diferentes lenguajes para que sirvan de intermediarios entre el hombre y las máquinas.

Los lenguajes de programación facilitan el trabajo a la hora de programar, ya que disponen de formas adecuadas que permiten ser leídas y escritas por personas, a su vez resultan independientes del modelo de procesador a utilizar. Además, representan en forma simbólica y en manera de un texto los códigos que podrán ser leídos por una persona.

Existen estrategias que permiten ejecutar en una computadora un programa realizado en un lenguaje de programación simbólico. Los procesadores del lenguaje son los programas que permiten el tratamiento de la información en forma de texto, representada en los lenguajes de programación simbólicos.

Hay lenguajes de programación que utilizan compilador.

La ejecución de un programa con compilador requiere de dos etapas:

- 1) Traducir el programa simbólico a código máquina
- 2) Ejecución y procesamiento de los datos.

Otros lenguajes de programación utilizan un programa intérprete o traductor, el cual analiza directamente la descripción simbólica del programa fuente y realiza las instrucciones dadas.

El intérprete en los lenguajes de programación simula una máquina virtual, donde el lenguaje de máquina es similar al lenguaje fuente.

La ventaja del proceso intérprete es que no necesita de dos fases para ejecutar el programa, sin embargo, su inconveniente es que la velocidad de ejecución es más lenta ya que debe analizar e interpretar las instrucciones contenidas en el programa fuente.

Entre los lenguajes de programación más usados en el mundo podemos citar: Delphi, Java, C, Visual Basic, etc.

Para la implementación del sistema automático para la Gestión de RRHH de los Polos Productivos de la Facultad 9 GESTAPRO se utilizarán lenguajes de programación para el desarrollo de aplicaciones web como los son: PHP, HTML, JavaScript y CSS.

#### **1.4.1 PHP**

Definido en la arquitectura del sistema como lenguaje de programación, PHP, cuyas siglas responden a un acrónimo recursivo (**PHP Hypertext Preprocessor** o **Preprocesador de Hipertexto PHP**), es un lenguaje de script usado principalmente para ejecutarse en servidores web, rápido y con una gran librería de funciones.

Fue creado en 1994 por Rasmus Lerdorf e inicialmente significaba Personal Home Page pero como está desarrollado en política de código abierto, a lo largo de su historia ha tenido muchas contribuciones de otros desarrolladores. (8)

Este lenguaje está mundialmente difundido entre las comunidades de programadores, es gratuito y multiplataforma. (9)

Entre las características que le hacen un competidor muy fuerte para los demás productos podemos encontrar las siguientes: (10)

- Interfaces para una gran cantidad de sistemas de base de datos diferentes.
- Bibliotecas incorporadas para muchas tareas Web habituales.
- Bajo coste.
- Facilidad de aprendizaje y uso.
- Portabilidad.
- Acceso al código abierto.
- Integración de base de datos.

#### **Ventajas (11)**

- PHP corre en casi cualquier plataforma utilizando el mismo código fuente, pudiendo ser compilado y ejecutado en algo así como 25 plataformas, incluyendo diferentes versiones de Unix, Windows (95,98,NT, 2000,XP,etc) y Macs. Como en todos los sistemas se utiliza el mismo código base, los scripts pueden ser ejecutados de manera independiente al sistema operativo.
- La sintaxis de PHP es similar a la del C, por esto cualquiera con experiencia en lenguajes del estilo C podrá entender rápidamente PHP.
- PHP es completamente expandible. Está compuesto de un sistema principal, un conjunto de módulos y una variedad de extensiones de código.
- Cuenta con interfaces distintas para cada tipo de servidor. PHP actualmente se puede ejecutar bajo Apache, AOLServer, Roxen y THTTPD. Otra alternativa es configurarlo como módulo CGI.
- Puede interactuar con muchos motores de bases de datos tales como MySQL, MS SQL, Oracle, Informix, PostgreSQL, y otros muchos.

- Posee una gran variedad de módulos. Cuando un programador PHP necesite una interface para una librería en particular, fácilmente podrá crear una API para esta. Algunas de las que ya vienen implementadas permiten manejo de gráficos, archivos PDF, calendarios, XML, IMAP, POP, etc.
- PHP generalmente es utilizado como módulo de Apache, lo que lo hace extremadamente veloz. Está completamente escrito en C, así que se ejecuta rápidamente utilizando poca memoria.
- PHP es open source (código abierto), lo cual significa que el usuario no depende de una compañía específica para arreglar cosas que no funcionan, además no estás forzado a pagar actualizaciones anuales para tener una versión que funcione.

#### **1.4.2 HTML**

HTML (HyperText Markup Language o Lenguaje de Marcado de Hipertexto) es un lenguaje muy sencillo que permite presentar texto de forma estructurada y agradable, con enlaces (hyperlinks) que conducen a otros documentos o fuentes de información relacionadas, y con inserciones multimedia (gráficos, sonido). Es el lenguaje más utilizado para la presentación de textos estructurados en formato hipertexto, estándar de las páginas web.

Fue creado por Tim Berners-Lee en 1990 y tenía como objetivo facilitar a científicos de diferentes universidades el acceso a los documentos de investigación de cada uno de ellos. El proyecto obtuvo un éxito mucho mayor del que Tim Berners-Lee nunca hubiese llegado a imaginar.

HTML es utilizado por la totalidad de navegadores web del mercado con el fin de presentar al visitante de una página web el contenido de la misma tal como el diseñador quiere que se muestre a su público.

Debido a su estandarización por la organización sin ánimo de lucro W3C (World Wide Web Consortium) y reconocido este estándar por todas las empresas relacionadas con la Internet una página HTML se muestra de la misma manera en cualquier sistema operativo y en cualquier navegador.

### 1.4.3 CSS

CSS (Cascading Style Sheets u Hojas de Estilo en Cascada) es la tecnología desarrollada por el World Wide Web Consortium (W3C) con el fin de separar la estructura de la presentación que nos permite crear páginas web de una manera más exacta y permitiendo un control mayor del documento y sus atributos convirtiendo al HTML en un documento muy versátil.

Gracias a las CSS se pueden hacer muchas cosas que no se podrían hacer utilizando solamente HTML, como incluir márgenes, tipos de letra, fondos, colores, etc. Se puede decir que el HTML es la caja que muestra los contenidos y el CSS es la manera en que lo hace.

CSS se utiliza para dar estilo a documentos HTML y XHTML, separando el contenido de la presentación. Los estilos definen la forma de mostrar los elementos HTML y XHTML.

CSS permite a los desarrolladores Web controlar el estilo y el formato de múltiples páginas Web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento.

El uso de las CSS conceden las siguientes ventajas: (12)

- Control centralizado de la presentación de un sitio web completo con lo que se agiliza de forma considerable la actualización del mismo.
- Mejor accesibilidad para todas las personas.
- Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre o incluso a elección del usuario.
- El documento HTML es más claro de entender y se consigue reducir considerablemente su tamaño.
- Visualización de la Web en otros soportes electrónicos (telefonía celular).

#### **1.4.4 JavaScript**

JavaScript es un lenguaje de script multiplataforma, interpretado, orientado a objetos y se utiliza principalmente para el desarrollo de páginas web dinámicas.

JavaScript es un lenguaje pequeño y ligero, está diseñado para una fácil incrustación en otros productos y aplicaciones tales como páginas web.

Se trata de un lenguaje de programación que aunque se puede utilizar del lado del servidor es muy utilizado en el lado del cliente gracias a su compatibilidad con la mayoría de los navegadores modernos, proporcionando objetos para el control del navegador y su Modelo de Objeto Documento (Document Object Model, DOM).

Al ser un lenguaje de programación interpretado no requiere de compilación, solo se necesita del navegador para su interpretación.

Hay que mencionar que existen dos tipos de JavaScript; por un lado está el que se ejecuta en el lado del cliente, que es el que se conoce como JavaScript aunque técnicamente se llama Navigator JavaScript y también está el LiveWire JavaScript que es el que se ejecuta en el servidor siendo este más reciente.

Debido a su semejanza en cuanto a nombre se refiere, JavaScript no guarda relación alguna con el lenguaje de programación Java.

Entre las ventajas del uso de JavaScript, podemos mencionar las siguientes: (13)

- Es un código “interpretado” por el cliente.
- Es un lenguaje abierto.
- Es un código orientado a objetos.
- Es un código integrado a HTML.
- Reutilización de código de programación.

#### **1.5 Sistemas Gestores de Bases de Datos.**

Un Sistema Gestor de Bases de Datos (a partir de ahora SGBD) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las

aplicaciones. (14) Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos.

Un SGBD debe permitir: (14)

- Definir una base de datos: especificar tipos, estructuras y restricciones de datos.
- Construir la base de datos: guardar los datos en algún medio controlado por el mismo SGBD.
- Manipular la base de datos: realizar consultas, actualizarla, generar informes.

De esta manera, un SGBD debe ser un software de propósito general.

Algunos ejemplos de SGBD son Oracle, DB2, PostgreSQL, MySQL, MS SQL Server, etc.

Algunas de las características deseables en un Sistema Gestor de base de datos SGBD son: (14)

- Control de la redundancia: La redundancia de datos tiene varios efectos negativos (duplicar el trabajo al actualizar, desperdicia espacio en disco, puede provocar inconsistencia de datos) aunque a veces es deseable por cuestiones de rendimiento.
- Restricción de los accesos no autorizados: Cada usuario ha de tener unos permisos de acceso y autorización.
- Cumplimiento de las restricciones de integridad: El SGBD ha de ofrecer recursos para definir y garantizar el cumplimiento de las restricciones de integridad.
- Seguridad: La información almacenada en una base de datos puede llegar a tener un gran valor. Los SGBD deben garantizar que esta información se encuentra segura frente a usuarios malintencionados, que intenten leer información privilegiada; frente a ataques que deseen manipular o destruir la información; o simplemente ante la inexperiencia de algún usuario autorizado.

- Independencia: La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.
- Abstracción de la información: Los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. Da lo mismo si una base de datos ocupa uno o cientos de archivos, este hecho se hace transparente al usuario. Así, se definen varios niveles de abstracción.

Un sistema gestor de base de datos está compuesto de: (15)

### **El gestor de la base de datos**

Se trata de un conjunto de programas no visibles al usuario final que se encargan de la privacidad, la integridad, la seguridad de los datos y la interacción con el sistema operativo. Proporciona una interfaz entre los datos, los programas que los manejan y los usuarios finales. Cualquier operación que el usuario hace contra la base de datos está controlada por el gestor.

El gestor almacena una descripción de datos en lo que llamamos diccionario de datos, así como los usuarios permitidos y los permisos.

Tiene que haber un usuario administrador encargado de centralizar todas estas tareas.

### **Diccionario de datos**

Es una base de datos donde se guardan todas las propiedades de la base de datos, descripción de la estructura, relaciones entre los datos, etc.

El diccionario debe contener:

- La descripción externa, conceptual e interna de la base de datos.
- Las restricciones sobre los datos.
- El acceso a los datos.



- Las descripciones de las cuentas de usuario.
- Los permisos de los usuarios.
- Los esquemas externos de cada programa.

### **El administrador de la base de datos**

Es una persona o grupo de personas responsables del control del sistema gestor de base de datos.

Las principales tareas de un administrador son:

- La definición del esquema lógico y físico de la base de datos.
- La definición de las vistas de usuario.
- La asignación y edición de permisos para los usuarios.
- Mantenimiento y seguimiento de la seguridad en la base de datos.
- Mantenimiento general del sistema gestor de base de datos.

### **Los lenguajes**

Un sistema gestor de base de datos debe proporcionar una serie de lenguajes para la definición y manipulación de la base de datos. Estos lenguajes son los siguientes:

- Lenguaje de definición de datos (DDL). Para definir los esquemas de la base de datos.
- Lenguaje de manipulación de datos (DML). Para manipular los datos de la base de datos.
- Lenguaje de control de datos (DCL). Para la administración de usuarios y seguridad en la base de datos.

### 1.5.1 PostgreSQL

Para el almacenamiento de datos se utilizará PostgreSQL, que es ideal para bases de datos donde se requiera recibir gran cantidad de peticiones, por lo que es factible su uso en grandes bases de datos. (9)

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. (16)

Es un SGBD basado en el open source (código abierto) por lo que su desarrollo está basado en el trabajo de un gran número de desarrolladores esparcidos por todo el mundo trabajando en su mayoría de manera voluntaria.

Funciona en los sistemas operativos más usados en el mundo como los son Linux, Windows y Unix (Mac OS X, Solaris).

Entre sus principales características podemos encontrar las siguientes: (16)

- Implementación del estándar SQL92/SQL99.
- Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP), cadenas de bits, etc. También permite la creación de tipos propios.
- Incorpora una estructura de datos array.
- Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, etc.
- Permite la declaración de funciones propias, así como la definición de disparadores.
- Soporta el uso de índices, reglas y vistas.

- Incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.

Entre las ventajas de PostgreSQL podemos encontrar las siguientes: (17)

**Instalación ilimitada:** Es frecuente que las bases de datos comerciales sean instaladas en más servidores de lo que permite la licencia pero nadie puede demandar al cliente por violar acuerdos de licencia, puesto que no hay costo asociado a la licencia del software.

**Mejor soporte que los proveedores comerciales:** Además de las ofertas de soporte, existe una importante comunidad de profesionales y entusiastas de PostgreSQL de los que su compañía puede obtener beneficios y contribuir.

**Ahorros considerables en costos de operación:** El software ha sido diseñado y creado para tener un mantenimiento y ajuste mucho menor que los productos de los proveedores comerciales, conservando todas las características, estabilidad y rendimiento.

**Estabilidad y confiabilidad legendarias:** En contraste a muchos sistemas de bases de datos comerciales, es extremadamente común que compañías reporten que PostgreSQL nunca ha presentado caídas en varios años de operación de alta actividad.

**Extensible:** El código fuente está disponible para todos sin costo. Si su equipo necesita extender o personalizar PostgreSQL de alguna manera, pueden hacerlo con un mínimo esfuerzo, sin costos adicionales.

**Multiplataforma:** PostgreSQL está disponible en casi cualquier sistema operativo Unix, y una versión nativa de Windows está actualmente en estado beta de pruebas.

**Diseñado para ambientes de alto volumen:** PostgreSQL usa una estrategia de almacenamiento de filas llamada MVCC para conseguir una mejor respuesta en ambientes de grandes volúmenes.

**Herramientas gráficas de diseño y administración de bases de datos:** Existen varias herramientas gráficas de alta calidad para administrar las bases de datos (pgAdmin, pgAccess) y para hacer diseño de bases de datos (Tora, Rational Data Architect).

## 1.6 Servidor Web

### 1.6.1 Servidor Apache

Para trabajar con lenguaje PHP es necesario una tecnología de servidor que sea capaz de interpretarlo, para ello se selecciona la tecnología de servidor Apache. (9)

El Servidor Apache HTTP es un servidor Web de tecnología open source (código abierto) sólido y para uso comercial desarrollado por la Apache Software Foundation (Fundación del Software Apache).

El servidor Apache es un software que está estructurado en módulos. La configuración de cada módulo se hace mediante la configuración de las directivas que están contenidas dentro del módulo. Los módulos del Apache se pueden clasificar en tres categorías:

- **Módulos Base:** Módulo con las funciones básicas del Apache
- **Módulos Multiproceso:** son los responsables de la unión con los puertos de la máquina, aceptando las peticiones y enviando a los hijos a atender a las peticiones
- **Módulos Adicionales:** Cualquier otro módulo que le añada una funcionalidad al servidor.

Las funcionalidades más básicas se encuentran en el módulo base, siendo imprescindible un módulo multiproceso para manejar las peticiones. Se han diseñado varios módulos multiproceso para cada uno de los sistemas operativos sobre los que se ejecuta el Apache, optimizando el rendimiento y rapidez del código.

El resto de funcionalidades del servidor se consiguen por medio de módulos adicionales que se pueden cargar. Para adicionar un conjunto de utilidades al

servidor, simplemente hay que añadirle un módulo, de forma que no es necesario volver a instalar el software. (18)

**Ventajas: (19)**

- La arquitectura modular personalizable: Apache permite crear un servidor de la manera deseada.
- Administración: Los Archivos de configuración de Apache está en código ASCII, tienen un formato simple, y puede ser editado utilizando cualquier editor de texto. Se puede controlar el servidor desde la línea de comandos, lo que hace la administración remota muy conveniente.
- Extensible: El código fuente del servidor Apache están abiertos al público. Si existe alguna característica que se necesita, pero no existe en Apache, puede escribir un módulo de servidor para la aplicación.
- Eficiente: Gran parte del esfuerzo se ha puesto en la optimización de código C de Apache para el rendimiento. Como resultado, se ejecuta más rápido y consume menos recursos que muchos otros servidores.
- Portabilidad: Apache se ejecuta en una amplia variedad de sistemas operativos, incluyendo todas las variantes de UNIX, Windows 9x/NT, MacOS, entre otros.
- La estabilidad / fiabilidad: El código fuente de Apache está abierto al público. Cuando se encuentran errores, se comunican y se arreglan rápidamente. Se realizan actualizaciones y se anuncian a partir de entonces.
- Soporte: Apache es apoyado por el Grupo Apache, un gran número de usuarios dedicados, y por las empresas que comercializan versiones de Apache.

**1.7 Herramientas de desarrollo**

### **1.7.1 Zend Studio**

Es un IDE (Integrated Development Environment o Entorno de Desarrollo Integrado) destinado a desarrolladores profesionales. Es un IDE propietario, compatible con las plataformas Linux, MAC y Windows. Está definido en la arquitectura del sistema como IDE de desarrollo para la implementación del sistema.

El IDE, además de servir de editor de texto para páginas PHP, proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código. (9)

Incluye todos los componentes necesarios durante el ciclo de vida de una aplicación en PHP. Incluye editor, análisis, depuración, optimizadores de código y herramientas de base de datos. Zend Studio nos permite agilizar el desarrollo web y permite simplificar proyectos complejos.

Entre sus características podemos mencionar que tiene un excelente completamiento de código, coloreado de sintaxis del código, administración avanzada de proyectos, múltiples lenguajes, incorpora el Framework de Zend, manual de PHP, soporte para Servicios Web, PHP4, PHP5 y SQL. (20)

### **1.7.2 Dreamweaver**

Por cuestiones de tiempo y de mejorar la calidad y facilidad de programación del diseño de las páginas del producto se propone la utilización de una herramienta que aunque no forma parte del grupo de herramientas de la licencia libre, sin dudas es una de las mejores para realizar esta tarea; se habla del Dreamweaver. (9)

Dreamweaver es la herramienta de diseño de páginas web de tipo WYSIWYG (What you see is what you get, Inglés) (Lo que ves es lo que obtienes) más avanzada dado que proporciona una potente combinación de herramientas visuales de diseño, funciones de desarrollo de aplicaciones y soporte para la edición del código, características que permiten a los desarrolladores y diseñadores más o menos expertos crear rápidamente sitios Web y aplicaciones basados en estándares.

Aunque sea un experto programador de HTML el usuario que lo maneje, siempre se encontrarán en este programa razones para utilizarlo, sobre todo en lo que a productividad se refiere.

Desde su lanzamiento en 1997, Dreamweaver se ha convertido en la solución estándar de la industria para el desarrollo profesional de la Web y, actualmente, más de 3,2 millones de profesionales recurren a Dreamweaver para desarrollar sus sitios Web.

Esta herramienta cumple perfectamente el objetivo de diseñar páginas con aspecto profesional, y soporta gran cantidad de tecnologías, además muy fáciles de usar:

- Hojas de estilo y capas.
- JavaScript para crear efectos e interactividades.
- Inserción de archivos multimedia.

Entre las ventajas que posee podemos encontrar las siguientes: (21)

- Trabajar sin necesidad de tener conocimientos de programación HTML.
- Con el mismo editor puedes trabajar con páginas estáticas como dinámicas.
- Conectividad con base de datos.
- Ahorro de tiempo en desarrollo de proyectos.
- Herramientas de comprobación de errores en compatibilidad.
- FTP integrado.
- Aplicaciones prediseñadas listas para usarse solo configurando algunos parámetros.
- Uso de Estilos (CSS).
- Soporta diferentes lenguajes (XML, ColdFusion, ASP, XHTML, Ajax, etc.).

- Compatibilidad con dispositivos móviles

## 1.8 Metodologías de Desarrollo de Software

Las metodologías de desarrollo de software son el conjunto de procedimientos, técnicas, herramientas y un soporte documental que permiten producir y mantener un producto de software. En el proceso de desarrollo de software la metodología define quién debe hacer qué, cuándo y cómo debe hacerlo.

Una metodología puede seguir uno o varios modelos de ciclo de vida, es decir, el ciclo de vida indica qué es lo que hay que obtener a lo largo del desarrollo del proyecto pero no cómo hacerlo.

Entre los ciclos de vida más habituales podemos encontrar usando prototipos, cascada, iterativo e incremental (espiral), basado en componentes y de programación automática.

No existe una metodología de desarrollo de software universal. Debido a que cada grupo de desarrollo tiene características diferentes el proceso de desarrollo de software debe ser adaptable.

Existen dos clasificaciones para las metodologías existentes: las ágiles o ligeras y las tradicionales o robustas. Entre las ágiles podemos encontrar a Extreme Programming, SCRUM, Crystal Methodologies, entre otras y como ejemplo de las metodologías tradicionales podemos mencionar a Microsoft Solutions Framework (MSF) y al Proceso Unificado de Desarrollo (RUP, Rational Unified Process por sus siglas en inglés).

El Proceso Unificado de Desarrollo (RUP) es la metodología que más se destaca entre las tradicionales.

El sistema para el cual se centra el desarrollo de la investigación será llevado a cabo haciendo uso de la metodología de desarrollo de software Rational Unified Process (RUP). (9)

RUP fue creado por la empresa Rational (IBM) y permite seleccionar fácilmente el conjunto de componentes de proceso que se ajustan a las necesidades específicas del proyecto. Se podrán alcanzar resultados predecibles unificando el equipo con



procesos comunes que optimicen la comunicación y creen un entendimiento común para todas las tareas, responsabilidades y artefactos.

Diferente a otras metodologías comerciales, la plataforma RUP hace que el proceso sea práctico con bases de conocimiento y guías para ayudar en el despegue de la planificación del proyecto, integrar rápidamente a los miembros del equipo y poner en acción el proceso personalizado.

Una de las mejores prácticas centrales de RUP es la noción de desarrollar iterativamente. RUP organiza los proyectos en términos de disciplinas y fases, consistiendo cada una en una o más iteraciones. Con esta aproximación iterativa, el énfasis de cada flujo de trabajo variará a través del ciclo de vida. La aproximación iterativa ayuda a mitigar los riesgos en forma temprana y continua, con un progreso demostrable y frecuentes releases ejecutables.

Se caracteriza por ser iterativo e incremental, centrado en la arquitectura y dirigido por casos de uso. Cuenta con 4 fases: Inicio, Elaboración, Construcción y Transición, además cuenta con 9 flujos de trabajos, de los cuales 6 son de Ingeniería y los 3 restantes son de apoyo. En cada iteración se obtienen artefactos que son los resultados tangibles que se alcanzan como por ejemplo diagramas de actividades, código fuente, entre otros y para una mejor organización define roles que no es más que el papel a desempeñar por una persona.

### **1.9 Conclusiones Parciales.**

Con el estudio realizado en este capítulo y después de haber analizado y argumentado las características de las principales tecnologías, herramientas y metodologías a utilizar para la implementación de la solución propuesta, se puede afirmar que las mismas poseen características y ventajas con respecto a otras que facilitan la implementación del sistema, además, en su mayoría son software libre debido a lo cual la aplicación cuenta con un bajo costo de desarrollo.

## Capítulo 2: Construcción del sistema.

### 2.1 Introducción

En este capítulo se realiza una valoración del diseño propuesto por el analista, además se exponen los diagramas de componentes de algunos de los principales casos de uso. Se establecen los estándares de codificación a utilizar en el desarrollo de la aplicación y se describen brevemente algunas clases a implementar, así como sus principales funcionalidades.

### 2.2 Valoración crítica del diseño propuesto por el analista.

La obtención de los artefactos generados por el analista durante el flujo de trabajo de Análisis y Diseño del sistema GESTAPRO resultó de gran importancia ya que permitió una mejor comprensión de todos los aspectos relacionados con los requisitos funcionales y componentes reutilizables.

La principal fuente de información que brinda el analista es la especificación de requisitos. Los mismos pueden ser clasificados en funcionales y no funcionales.

Requisitos Funcionales: Capacidades o condiciones que el sistema debe cumplir.

Requisitos No Funcionales: Son propiedades o cualidades que el producto debe tener.

#### 2.2.1 Requisitos Funcionales.

Los requisitos funcionales descritos por el analista y que facilitaron la implementación del sistema son:

- ❖ RF1. Autenticar usuario.
- ❖ RF2. Gestionar Polo.
  - RF2.1. Registrar Polo.
  - RF2.2. Modificar datos del Polo.
  - RF2.3. Eliminar Polo.
  - RF2.4. Mostrar datos de un Polo.
- ❖ RF3. Buscar Polo.
- ❖ RF4. Gestionar Rol.

- RF4.1. Registrar Rol.
- RF4.2. Modificar Rol.
- RF4.3. Eliminar Rol.
- RF4.4. Mostrar Rol.
- ❖ RF5. Buscar Rol.
- ❖ RF6. Gestionar Categoría.
  - RF6.1. Registrar Categoría.
  - RF6.2. Modificar Categoría.
  - RF6.3. Eliminar Categoría.
  - RF6.4. Mostrar Categoría.
- ❖ RF7. Buscar Categoría.
- ❖ RF8. Gestionar PC.
  - RF8.1. Registrar PC.
  - RF8.2. Modificar PC.
  - RF8.3. Eliminar PC.
  - RF8.4. Mostrar PC.
- ❖ RF9. Buscar PC.
- ❖ RF10. Gestionar tiempo de máquina.
  - RF10.1. Registrar tiempo de máquina.
  - RF10.2. Modificar tiempo de máquina.
  - RF10.3. Eliminar tiempo de máquina.
  - RF10.4. Mostrar tiempo de máquina.
- ❖ RF11. Buscar tiempo de máquina.
- ❖ RF12. Gestionar Equipo de Producción e Investigación (EPI).
  - RF12.1. Registrar Equipo de Producción e Investigación (EPI).
  - RF12.2. Modificar Equipo de Producción e Investigación (EPI).
  - RF12.3. Eliminar Equipo de Producción e Investigación (EPI).
  - RF12.4. Mostrar Equipo de Producción e Investigación (EPI).
- ❖ RF13. Buscar Equipo de Producción e Investigación (EPI).
- ❖ RF14. Gestionar usuario.
  - RF14.1. Registrar usuario.
  - RF14.2. Modificar usuario.

- RF14.3. Eliminar usuario.
- RF14.4. Mostrar usuario.
- ❖ RF15. Buscar usuario.
- ❖ RF16. Mostrar datos personales.
- ❖ RF17. Gestionar tarea.
  - RF17.1. Registrar tarea.
  - RF17.2. Modificar tarea.
  - RF17.3. Eliminar tarea.
  - RF17.4. Mostrar tarea.
- ❖ RF18. Buscar tarea.
- ❖ RF19. Gestionar evento.
  - RF19.1. Registrar evento.
  - RF19.2. Modificar evento.
  - RF19.3. Eliminar evento.
  - RF19.4. Mostrar evento.
- ❖ RF20. Buscar evento.
- ❖ RF21. Gestionar curso optativo del segundo perfil.
  - RF21.1. Registrar curso optativo del segundo perfil.
  - RF21.2. Modificar curso optativo del segundo perfil.
  - RF21.3. Eliminar curso optativo del segundo perfil.
  - RF21.4. Mostrar curso optativo del segundo perfil.
- ❖ RF22. Buscar curso optativo del segundo perfil.
- ❖ RF23. Gestionar noticia.
  - RF23.1. Registrar noticia.
  - RF23.2. Modificar noticia.
  - RF23.3. Eliminar noticia.
  - RF23.4. Mostrar noticia.
- ❖ RF24. Buscar noticia.
- ❖ RF25. Gestionar evaluación.
  - RF25.1. Registrar evaluación.
  - RF25.2. Modificar evaluación.
  - RF25.3. Eliminar evaluación.
  - RF25.4. Mostrar evaluación.

- ❖ RF26. Buscar evaluación.
- ❖ RF27. Generar Reporte.
- ❖ RF28. Exportar Reporte.
- ❖ RF29. Imprimir Reporte.
- ❖ RF30. Ver tiempo de máquina asignado.
- ❖ RF31. Ver tareas asignadas.
- ❖ RF32. Ver las evaluaciones.
- ❖ RF33. Ver noticias publicadas en la semana.
- ❖ RF34. Buscar información en las noticias por un criterio dado por el usuario.

### **2.2.2 Requisitos No Funcionales.**

Los requerimientos no funcionales son las características que hacen al software atractivo para el usuario, los que dicen cuán rápido o confiable debe ser el mismo, ponen las pautas a seguir y son imprescindibles para que el sistema tenga aceptación por los usuarios. (22)

A continuación se especifican algunos de estos requerimientos.

#### **Requerimientos de software:**

1. El sistema deberá ser multiplataforma, especialmente para Windows XP y Linux.
2. El sistema deberá visualizarse en Internet Explorer 6.0 o superior y Mozilla 2.0 o superior.
3. El sistema deberá tener como gestor de Base de datos PostgreSQL 8.2.

#### **Requerimientos de hardware:**

- ❖ Estaciones de trabajo o PC Clientes.
  1. Periféricos: Mouse, teclado.
  2. Tarjeta de red.
  3. 128 MB de RAM o superior.
- ❖ Servidor de Aplicaciones.
  1. Periféricos: Mouse, teclado.
  2. Tarjeta de Red.
  3. 512 MB de RAM o más.
  4. 10 GB de espacio en disco.
- ❖ Servidor de Base Datos.

1. Periféricos: Mouse, teclado.
2. Tarjeta de Red.
3. 512 MB de RAM o superior.
4. 40 GB o más GB de disco.

**Restricciones de diseño o implementación:**

- 1- El sistema será implementado en el lenguaje PHP.
- 2- Para desarrollar el sistema se utilizará el IDE Zend Development Environment.
- 3- Para la modelación del sistema se usará el Visual Paradigm for UML 6.0 Enterprise Edition.

**Requerimientos de apariencia o interfaz externa:**

1. El sistema deberá tener una interfaz intuitiva, organizada y de fácil entendimiento para el usuario.
2. El sistema deberá presentar una interfaz agradable, que favorezca el estado de ánimo del cliente y que combine correctamente los colores, tipo de letra y tamaño y que los iconos estén en correspondencia con lo que representan.

**Requerimientos de Usabilidad:**

- 1- El sistema deberá ser utilizado por usuarios que tengan conocimiento acerca del funcionamiento y procesamiento de la información con que se trabaja en el sistema.

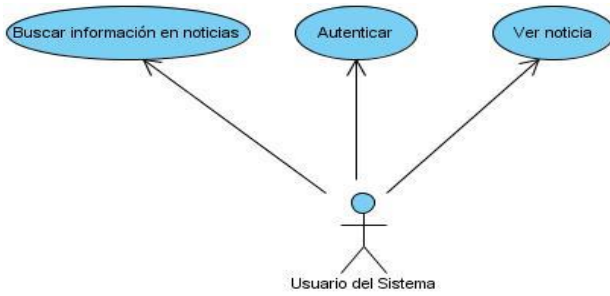
**Requerimientos de soporte:**

- 1- El sistema deberá ser revisado y actualizado cada 6 meses.

**2.2.3 Artefactos Obtenidos.**

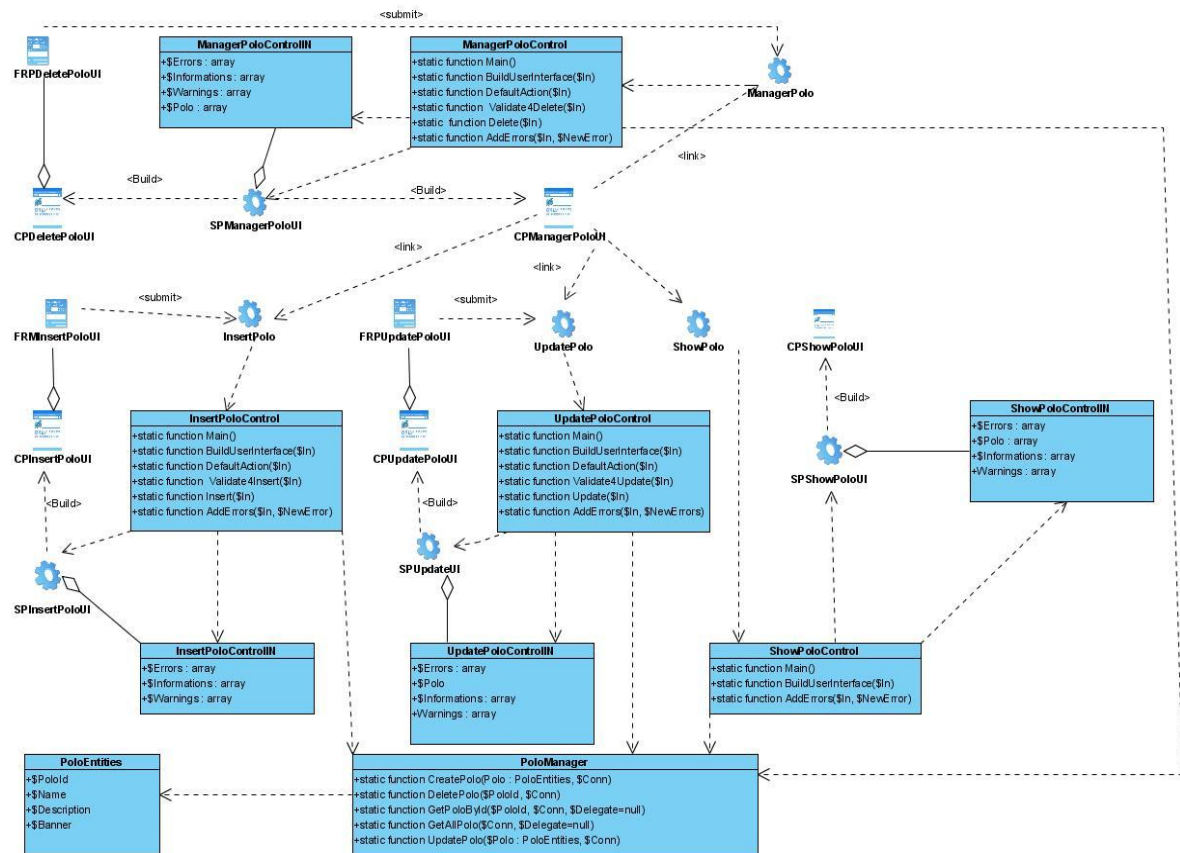
Durante la fase de elaboración, los requerimientos funcionales se convierten en casos de uso los cuales son ejecutados por actores del sistema construyéndose los diagramas de casos de uso del sistema donde se exponen las relaciones que existen entre los mismos.

A continuación se expone el DCUS Usuario del Sistema:



**Figura 1 DCUS Usuario del Sistema**

Entre los artefactos que se generaron están los diagramas de clases del diseño. A continuación, se muestra el diagrama de clases del diseño del CU Gestionar Polo en su escenario Modificar Polo.



**Figura 2 Diagrama de Clases del diseño CU Gestionar Polo.**

En esta figura se puede observar la correcta separación de las clases por escenarios permitiendo una mayor especialización de las mismas, además de las relaciones que existen entre las distintas clases facilitando la implementación del sistema.

De igual manera, se puede percibir la utilización de las clases que conforman los dos subsistemas que integran la Vista dentro del patrón arquitectónico Modelo-Vista-Controlador los cuales están descritos en la Vista Lógica de la arquitectura propuesta. Estas clases se utilizan para mostrar la información que el usuario desea ver y para brindar los recursos que se necesitan para la actualización de las vistas.

Para más información de los DCUS, de la descripción de cada caso de uso así como de los diagramas de clases del diseño referirse al Trabajo de Diploma “Análisis y diseño de un sistema automatizado para el control de los recursos humanos en los polos productivos de la Facultad 9” del autor Adrián Gracia Aguila. (22)

#### **2.2.4 Arquitectura del sistema.**

Como arquitectura para el sistema se optó por la arquitectura cliente-servidor seleccionándose como patrón arquitectónico Modelo-Vista-Controlador (MVC). Debido a la selección de ese patrón se agruparon las clases en 3 paquetes: Vista, Modelo y Controlador, las cuales están bien estructuradas con el objetivo de lograr un correcto funcionamiento de la aplicación.

Las clases pertenecientes al Modelo son las que representan la información con la que trabaja la aplicación, es decir, la lógica del negocio. En el caso de las que forman parte de la Vista son las encargadas de transformar los datos proveídos por el controlador en una página web que permite al usuario intercambiar con ella y en el caso del Controlador es el que se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo y actualiza la vista.

También se elaboraron Diagramas de Interacción y de Componentes, estos últimos de los casos de uso arquitectónicamente significativos, los cuales ayudan a comprender la estructura del sistema gracias a su buena elaboración, captando de ellos lo que debe realizar la aplicación.

#### **2.3 Diagramas de Componentes.**

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Estos diagramas muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Los componentes representan todos los tipos de



elementos software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, bibliotecas cargadas dinámicamente, entre otros. Las relaciones de dependencia se utilizan en los diagramas de componentes para indicar que un componente utiliza los servicios ofrecidos por otro componente. (23)

El sistema está dividido en cuatro partes fundamentales: Model, Control, View y Global, los cuales están representados en los diagramas de componentes. A continuación se hará referencia de como quedan organizadas las clases del sistema en términos de componentes para cada una de las cuatro partes mencionadas anteriormente mostrando los diagramas de componentes de algunos de los principales casos de uso.

### 2.3.1 CU Autenticar.

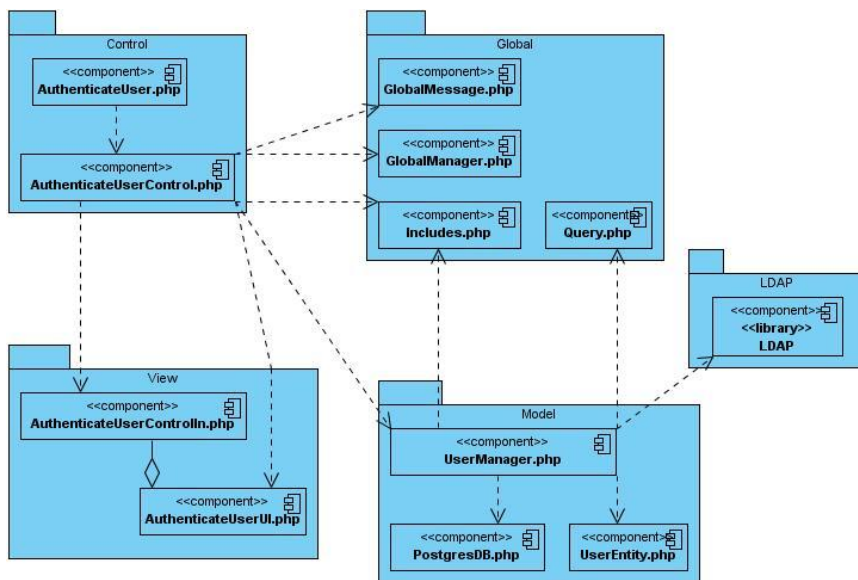


Figura 3 Diagrama de Componente CU Autenticar.

### 2.3.2 CU Gestionar Polo.

#### 2.3.2.1 Listar Polos.

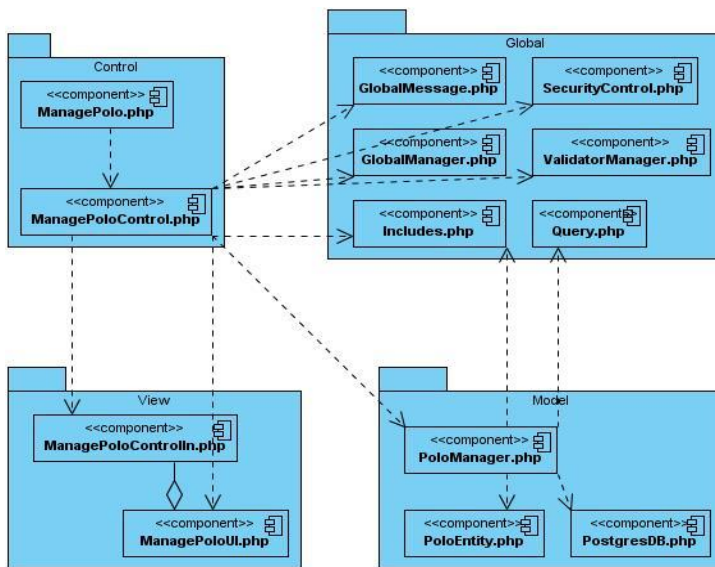


Figura 4 Diagrama de Componente Listar Polos.

### 2.3.2.2 Insertar Polo.

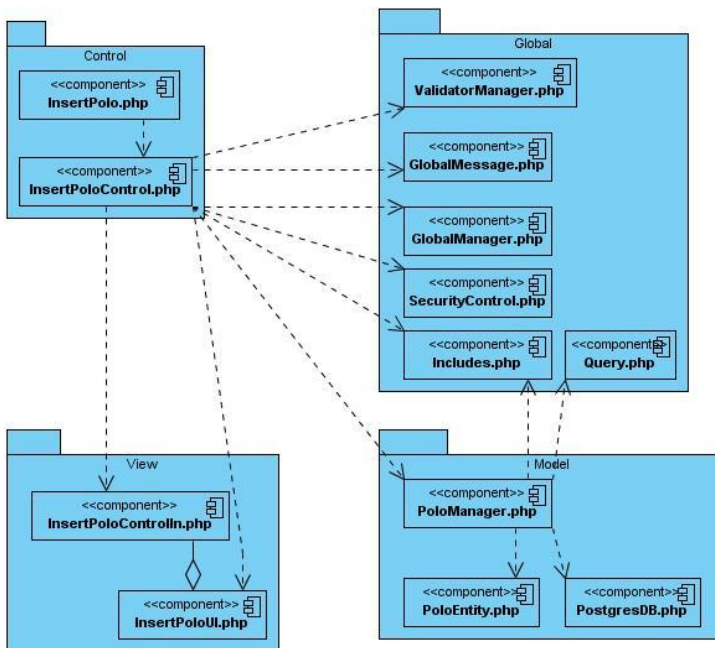


Figura 5 Diagrama de Componente Insertar Polo.

### 2.3.2.3 Modificar Polo.

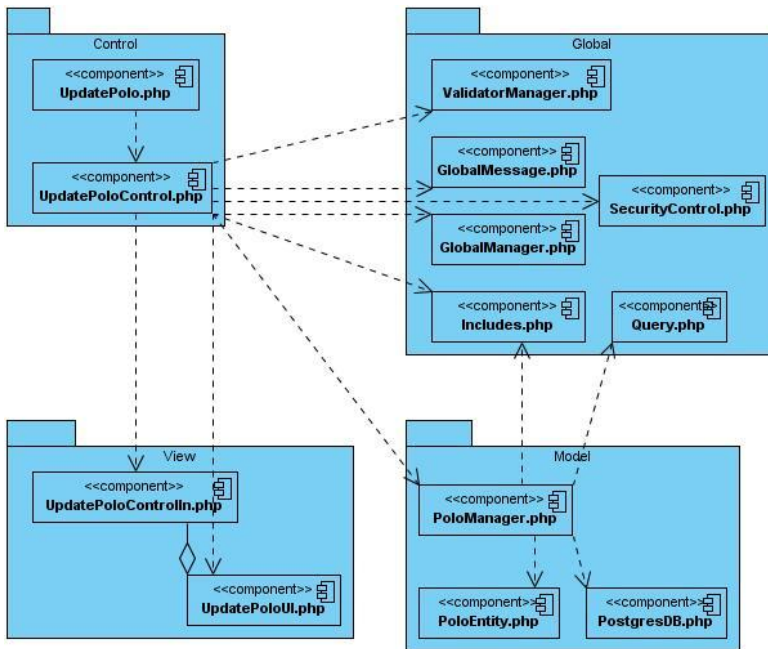


Figura 6 Diagrama de Componente Modificar Polo.

### 2.3.2.4 Mostrar Polo.

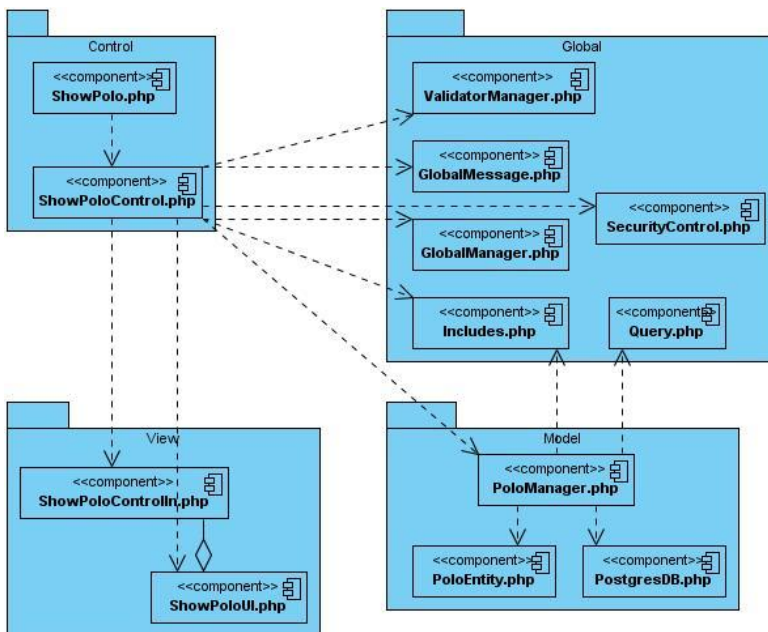


Figura 7 Diagrama de Componente Mostrar Polo.

## 2.4 Estándares de Codificación.

Los estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código.

Si bien los programadores deben implementar un estándar de forma prudente, éste debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez.

Al comenzar un proyecto de software, se debe establecer un estándar de codificación para asegurar de que todos los programadores del proyecto trabajen de forma coordinada. El mejor método para asegurarse de que un equipo de programadores mantenga un código de calidad es establecer un estándar de codificación sobre el que se efectuarán luego revisiones del código.

Usar técnicas de codificación sólidas y realizar buenas prácticas de programación con vistas a generar un código de alta calidad es de gran importancia para la calidad del software y para obtener un buen rendimiento.

La adopción de un estándar de codificación sólo es viable si se sigue desde el principio hasta el final del proyecto de software. No es práctico, ni prudente, imponer un estándar de codificación una vez iniciado el trabajo.

La utilización de un estándar de codificación trae consigo ventajas, tales como: (24)

- Asegurar la comprensión de cada una de las líneas de código.
- Facilitar el mantenimiento posterior a la creación del código.
- Facilitar la tarea de los programadores en el desarrollo del software.

Por las razones antes expuestas se decide poner en práctica el uso de estándares de codificación para la implementación del sistema automatizado GESTAPRO.

Para la definición de este estándar es primordial mencionar dos notaciones muy importantes en el momento de codificar las cuales se mencionan a continuación.

**Camel Case:** La primera letra de los identificadores y nombres de las variables así como las funciones se escriben con minúscula y si contiene más de una palabra, las mismas tienen su primera letra mayúscula. Ejemplo: notaciónCamelCase.

**Pascal Case:** La primera letra de los identificadores y nombres de las variables así como las funciones se escriben con mayúscula y si contiene más de una palabra, las mismas tienen su primera letra mayúscula. Ejemplo: NotaciónPascalCase.

Después de todo lo expresado y por su importancia, a continuación, se hace referencia a diferentes reglas que componen el estándar definido con su ejemplo ilustrativo:

#### **Declaración de variables:**

Para la declaración de las variables no se utilizarán abreviaturas y se registrará por la notación Pascal Case. Ejemplo:

```
$TareasAsignadas;
```

#### **Declaración de funciones y clases:**

Para declarar funciones o clases los nombres de las mismas no se abreviarán y serán declaradas teniendo en cuenta la notación Pascal Case. Ejemplo:

```
public function GestionarUsuario()  
{  
    //Código  
}  
class ControladoraGestionarUsuario  
{  
    //Código  
}
```

Para una mejor organización visual y entendimiento del código se hace referencia a las siguientes normas definidas por el estándar de codificación a utilizar:

- Declarar las variables en líneas diferentes.
- Para indentar el código se utilizará la indentación del IDE Zend Studio.

- Se dejará una línea en blanco antes y después de cada declaración de clase o implementación de una función.
- Se dejará un espacio en blanco entre operadores lógicos y aritméticos.  
Ejemplo: Usuario = LoginUser.
- No declarar más de una instrucción por línea.

Para más información ver la tabla 7 y 8 en la sección de Anexos.

## 2.5 Descripción de las principales clases y sus funcionalidades.

Durante la implementación del sistema se han construido clases con sus respectivas funcionalidades las cuales tienen como objetivo propiciar el buen funcionamiento de la aplicación. A continuación, se describen algunas de ellas.

Nombre: <b>ManagePoloControl</b>	
Tipo de Clase: <b>Controladora</b>	
<b>Para cada responsabilidad</b>	
Nombre:	Descripción:
Main()	Es la función principal de la clase, es la que se encarga de llamar a los métodos para la conexión con la BD y sus propios métodos para la construcción de la interfaz para la gestión de los polos.
DefaultAction()	Es una función que se encarga de la llamada al método que obtiene de la BD los polos existentes.
BuildUserInterface()	Es la función que construye la interfaz de usuario para la gestión de los polos existentes.
Validate4Delete()	Es una función para la validación a la hora de eliminar un polo determinado verificando si se ha especificado el id del

	polo.
Delete()	Es la función que hace la llamada al método para la validación y para eliminar un polo determinado.

**Tabla 1 Descripción de la clase ManagePoloControl.**

Nombre: <b>InsertPoloControl</b>	
Tipo de Clase: <b>Controladora</b>	
<b>Para cada responsabilidad</b>	
Nombre:	Descripción:
Main()	Es la función principal de la clase, es la que se encarga de llamar a los métodos para la conexión con la BD y sus propios métodos para la construcción de la interfaz para insertar polos.
DefaultAction()	Es una función que se encarga de obtener el usuario que está conectado.
BuildUserInterface()	Es la función que construye la interfaz de usuario para insertar polos.
Validate4Insert()	Es una función para la validación a la hora de insertar un polo determinado verificando los campos a llenar.
Insert()	Es la función que hace la llamada al método para la validación y para insertar un polo.
AddErrors()	Es la función que adiciona mensajes de errores durante la validación.

**Tabla 2 Descripción de la clase InsertPoloControl.**

Nombre: <b>ShowPoloControl</b>
--------------------------------

Tipo de Clase: <b>Controladora</b>	
<b>Para cada responsabilidad</b>	
Nombre:	Descripción:
Main()	Es la función principal de la clase, es la que se encarga de llamar a los métodos para la conexión con la BD y sus propios métodos para la construcción de la interfaz para mostrar un polo determinado.
DefaultAction()	Es la función que se encarga de la llamada al método que obtiene de la BD un polo determinado.
BuildUserInterface()	Es la función que construye la interfaz de usuario para mostrar un polo determinado.

**Tabla 3 Descripción de la clase ShowPoloControl.**

Nombre: <b>UpdatePoloControl</b>	
Tipo de Clase: <b>Controladora</b>	
<b>Para cada responsabilidad</b>	
Nombre:	Descripción:
Main()	Es la función principal de la clase, es la que se encarga de llamar a los métodos para la conexión con la BD y sus propios métodos para la construcción de la interfaz para la modificación de un polo determinado.
DefaultAction()	Es una función que se encarga de la llamada al método que obtiene de la BD un polo determinado.
BuildUserInterface()	Es la función que construye la interfaz de



	usuario para la modificación de un polo determinado.
Validate4Update()	Es una función para la validación a la hora de modificar un polo determinado verificando los campos a llenar.
Update()	Es la función que hace la llamada al método para la validación y para modificar un polo determinado.

**Tabla 4 Descripción de la clase UpdatePoloControl.**

Nombre: <b>PoloManager</b>	
Tipo de Clase: <b>Controladora</b>	
<b>Para cada responsabilidad</b>	
Nombre:	Descripción:
InsertPolo()	Ejecuta una consulta para la inserción de un polo en la BD.
GetAllPolos()	Ejecuta una consulta para obtener todos los polos existentes en la BD.
GetPoloById()	Ejecuta una consulta para obtener un polo determinado contenido en la BD.
UpdatePolo()	Ejecuta una consulta para la modificación de un polo contenido en la BD.
DeletePolo()	Ejecuta una consulta para la eliminación de un polo contenido en la BD.

**Tabla 5 Descripción de la clase PoloManager.**

Nombre: <b>PoloEntity</b>	
Tipo de Clase: <b>Entidad</b>	
Atributo	Tipo
Id	Interger

Name	Varchar
Description	Text
Banner	Text
<b>Para cada responsabilidad</b>	
Nombre:	Descripción:
Convert()	Es la función encargada de asignar los valores de un polo determinado en la BD a los atributos de la entidad.

**Tabla 6 Descripción de la clase PoloEntity.**

## 2.6 Conclusiones parciales.

Concluida la revisión del diseño propuesto y de todos los artefactos generados se puede concluir que dicho diseño facilita e implanta las bases para la implementación gracias a la buena estructura y organización que posee. Además, con la realización de los diagramas de componentes se facilita la comprensión de las clases del sistema así como las relaciones entre ellas. Por otra parte, se puede afirmar que los estándares de codificación son de gran importancia para la documentación y mantenimiento del código, debido a que a través de éstos se puede comprender el código sin necesidad de recurrir a la ayuda de un profesional y facilita la comunicación entre desarrolladores.

## Capítulo 3: Validación de la solución propuesta.

### 3.1 Introducción.

En este capítulo se aspira a validar la solución propuesta, a partir de la construcción de casos de prueba de caja negra encaminados a brindar un punto de partida para la ejecución del flujo de trabajo de Prueba y así de esta manera verificar el cumplimiento de los requisitos sin probar el código.

### 3.2 Descripción general de las pruebas de software.

Las pruebas de software son un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación (25). Están constituidas por técnicas, métodos y herramientas que contribuyen a que un programa tenga un correcto funcionamiento.

Las pruebas de software pueden ser clasificadas en dependencia del tipo de prueba que se le haga a la aplicación aunque todas se ejecutan para obtener el mismo resultado, encontrar errores o defectos en la aplicación. Algunas de ellas pueden ser:

**Pruebas de caja blanca:** Están dirigidas a probar las funciones internas, es decir, el código del sistema.

**Pruebas de caja negra:** Están dirigidas a ejecutar los elementos pertenecientes a la interfaz de usuario para estudiar la entrada de datos y respuestas del sistema.

**Pruebas de validación:** Están dirigidas a verificar si el sistema cumple con las especificaciones que el cliente definió, es decir, si cumple con las especificaciones del análisis.

**Pruebas de verificación:** Están encargadas de comprobar el cumplimiento de las especificaciones del diseño.

### 3.3 Diseño de los casos de prueba para la validación de la solución.

Para la validación de la aplicación se procede a diseñar los casos de pruebas de los principales casos de uso. Éstos pueden ser clasificados como pruebas de caja negra

ya que se llevan a cabo sobre la interfaz del software sin tener en cuenta el comportamiento interno y la estructura del programa.

Los casos de prueba de caja negra pretenden demostrar que:

- Las funciones del software son operativas.
- La entrada se acepta de forma adecuada.
- Se produce una salida correcta.
- La integridad de la información externa se mantiene.

Para comenzar con el diseño de los casos de prueba es válido aclarar el significado de las siguientes siglas:

**SC:** Sección.

**EC:** Escenario.

### 3.3.1 Gestionar Polo.

**Descripción General:** El caso de uso se inicializa cuando el Vicedecano de Producción de la Facultad desea insertar, modificar, eliminar o ver los datos de un polo.

**Condiciones de ejecución:** El Vicedecano de Producción de la Facultad 9 debe estar previamente autenticado en el sistema y tener los privilegios para realizar cualquiera de las acciones mencionadas anteriormente.

#### Secciones a probar en el Caso de Uso.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
----------------------	--------------------------	---------------------------------

<b>SC</b> Insertar polo.	<b>1:</b>	<b>EC 1.1:</b> Insertar un polo exitosamente.	El sistema muestra un formulario con los campos a llenar del polo, valida los datos de entrada. Guarda los datos del polo e informa al usuario que se ha insertado correctamente el polo finalizando el caso de uso.
		<b>EC 1.2:</b> Insertar un polo fallida.	El sistema muestra un formulario con los campos a llenar del polo, verifica que los datos de entrada no son correctos. Muestra mensajes informando cuales son los campos que presentan problemas finalizando el caso de uso.
<b>SC</b> Modificar polo.	<b>2:</b>	<b>EC 2.1:</b> Modificar un polo exitosamente.	El sistema actualiza la información del polo. Guarda los cambios hechos al polo seleccionado e informa al usuario de la modificación del polo finalizando el CU.
		<b>EC 2.2:</b> Modificar un polo fallida.	El sistema no actualiza la información del polo e informa al usuario los problemas encontrados, finalizando el CU.
<b>SC</b> Eliminar polo.	<b>3:</b>	<b>EC 3.1:</b> Eliminar un polo exitosamente.	El sistema verifica que exista el polo seleccionado, lo elimina, informa al usuario de su eliminación y actualiza la lista de polos, finalizando el CU.
		<b>EC 3.2:</b> Eliminar un polo fallida.	El sistema no elimina el polo e informa al usuario cuál fue el problema, finalizando el CU.
<b>SC</b> Mostrar polo.	<b>4:</b>	<b>EC 4.1:</b> Mostrar un polo	El sistema muestra los datos del polo finalizando el CU.
		<b>EC 4.2:</b> Mostrar un polo fallida.	El sistema informa al usuario cuál fue el problema finalizando el CU.

#### Descripción de variable.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
----	-----------------	---------------	------------	-------------

1	Nombre	Campo de Texto	No	Este campo es para la entrada del nombre del polo.
2	Descripción del polo	Campo de Texto	Sí	Este campo es para la entrada de la descripción del polo.

### SC 1: Insertar Polo.

Escenario	Nombre	Descripción	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
Insertar un polo exitosamente.	V	N/A	Se adiciona a la base de datos el nuevo polo.		<ul style="list-style-type: none"> <li>• Principal.</li> <li>• Panel de Control.</li> <li>• Gestionar Polo.</li> <li>• Nuevo Polo.</li> </ul>
Insertar un polo fallida.	I	N/A	No se adiciona a la base de datos el nuevo polo informando al usuario del error ocurrido.		<ul style="list-style-type: none"> <li>• Principal.</li> <li>• Panel de Control.</li> <li>• Gestionar Polo.</li> <li>• Nuevo Polo.</li> </ul>

### SC 2 Modificar Polo.

Escenario	Nombre	Descripción	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
Modificar un polo exitosamente.	V	N/A	Se actualiza en la base de datos el polo seleccionado.		<ul style="list-style-type: none"> <li>• Principal.</li> <li>• Panel de Control.</li> <li>• Gestionar Polo.</li> <li>• Editar.</li> </ul>
Modificar un polo fallida.	I	N/A	No se actualiza en la base de datos el polo seleccionado informando al usuario del error ocurrido.		<ul style="list-style-type: none"> <li>• Principal.</li> <li>• Panel de Control.</li> <li>• Gestionar Polo.</li> <li>• Editar.</li> </ul>

### SC 3 Eliminar Polo.

Escenario	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
Eliminar un polo exitosamente.	Se elimina en la base de datos el polo seleccionado.		<ul style="list-style-type: none"> <li>• Principal.</li> <li>• Panel de Control.</li> <li>• Gestionar Polo.</li> <li>• Borrar.</li> </ul>

Eliminar un polo fallida.	No se elimina en la base de datos el polo seleccionado informando al usuario del error ocurrido.		<ul style="list-style-type: none"> <li>• Principal.</li> <li>• Panel de Control.</li> <li>• Gestionar Polo.</li> <li>• Borrar.</li> </ul>
---------------------------	--	--	---

#### SC 4 Mostrar Polo.

Escenario	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
Mostrar un polo exitosamente.	Se muestra el polo seleccionado.		<ul style="list-style-type: none"> <li>• Principal.</li> <li>• Panel de Control.</li> <li>• Gestionar Polo.</li> <li>• Mostrar.</li> </ul>
Mostrar un polo fallida.	No se muestra el polo seleccionado informando al usuario del error ocurrido.		<ul style="list-style-type: none"> <li>• Principal.</li> <li>• Panel de Control.</li> <li>• Gestionar Polo.</li> <li>• Mostrar.</li> </ul>

#### 3.3.2 Buscar Polo.



**Descripción General:** El caso de uso se inicializa cuando el Vicedecano de Producción de la Facultad desea buscar según un criterio un polo o un conjunto de polos.

**Condiciones de ejecución:** El Vicedecano de Producción de la Facultad 9 debe estar previamente autenticado en el sistema y tener los privilegios para realizar cualquiera de las acciones mencionadas anteriormente.

**Secciones a probar en el Caso de Uso.**

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC 1: Buscar polo.	EC 1.1: Buscar un polo exitosamente.	El sistema muestra una lista con todos los polos que coincidieron en el criterio entrado finalizando el caso de uso.
	EC 1.2: Buscar un polo fallida.	El sistema muestra un mensaje informando cuál fue el problema encontrado finalizando el caso de uso.

**Descripción de variable.**

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Criterio	Campo de Texto	No	Este campo es para la entrada del criterio para la búsqueda del polo.

**SC 1 Buscar Polo.**

Escenario	Criterio	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
-----------	----------	-----------------------	------------------------	---------------

Buscar un polo exitosamente.	V	Se buscan en la base de datos los polos que coincidan con el criterio entrado.		<ul style="list-style-type: none"> <li>• Principal.</li> <li>• Panel de Control.</li> <li>• Gestionar Polo.</li> </ul>
Buscar un polo fallida.	I	Se muestra un mensaje informando cuál fue el problema ocurrido.		<ul style="list-style-type: none"> <li>• Principal.</li> <li>• Panel de Control.</li> <li>• Gestionar Polo.</li> </ul>

### 3.3.3 Gestionar Rol.

**Descripción General:** El caso de uso se inicializa cuando el Usuario Avanzado del Sistema desea insertar, modificar, eliminar, ver los datos de un rol así como asignar un usuario a un rol.

**Condiciones de ejecución:** El Usuario Avanzado del Sistema debe estar previamente autenticado en el sistema y tener los privilegios para realizar cualquiera de las acciones mencionadas anteriormente.

#### Secciones a probar en el Caso de Uso.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC 1: Insertar rol.	EC 1.1: Insertar un rol exitosamente.	El sistema muestra un formulario con los campos a llenar del rol, valida los datos de entrada. Guarda los datos del rol e informa al usuario que se ha insertado correctamente el rol finalizando el caso de uso.

	<b>EC 1.2:</b> Insertar un rol fallida.	El sistema muestra un formulario con los campos a llenar del rol, verifica que los datos de entrada no son correctos. Muestra mensajes informando cuales son los campos que presentan problemas finalizando el caso de uso.
<b>SC 2:</b> Modificar rol.	<b>EC 2.1:</b> Modificar un rol exitosamente.	El sistema actualiza la información del rol. Guarda los cambios hechos al rol seleccionado e informa al usuario de la modificación del rol finalizando el CU.
	<b>EC 2.2:</b> Modificar un rol fallida.	El sistema no actualiza la información del rol e informa al usuario los problemas encontrados, finalizando el CU.
<b>SC 3:</b> Eliminar rol.	<b>EC 3.1:</b> Eliminar un rol exitosamente.	El sistema verifica que exista el rol seleccionado, lo elimina, informa al usuario de su eliminación y actualiza la lista de roles, finalizando el CU.
	<b>EC 3.2:</b> Eliminar un rol fallida.	El sistema no elimina el rol e informa al usuario cuál fue el problema, finalizando el CU.
<b>SC 4:</b> Mostrar rol.	<b>EC 4.1:</b> Mostrar un rol exitosamente.	El sistema muestra los datos del rol finalizando el CU.
	<b>EC 4.2:</b> Mostrar un rol fallida.	El sistema informa al usuario cuál fue el problema finalizando el CU.
<b>SC 5:</b> Asignar rol.	<b>EC 5.1:</b> Asignar un rol exitosamente.	El sistema asigna correctamente el rol al usuario especificado finalizando así el CU.
	<b>EC 5.2:</b> Asignar un rol fallida.	El sistema informa al usuario cuál fue el problema encontrado, finalizando el CU.

#### Descripción de variable.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Nombre	Campo de Texto	No	Este campo es para la entrada del nombre del rol.
2	Descripción del rol	Campo de Texto	Sí	Este campo es para la entrada de la descripción del rol.
3	Funcionalidades	CheckBox	No	Este campo es para la selección de las funcionalidades del rol.
4	Usuario	Lista Desplegable	No	Este campo es para la selección del usuario al que se le va a asignar el
5	Rol	Lista Desplegable	No	Este campo es para la selección del rol que se le va a asignar al usuario.

#### SC 1 Insertar Rol.

Escenario	Nombre	Descripción	Funcionalidades	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
Insertar un rol exitosamente.	V	N/A	V	Se adiciona a la base de datos el nuevo rol.		<ul style="list-style-type: none"> <li>• Principal.</li> <li>• Panel de Control.</li> <li>• Gestionar Rol.</li> <li>• Nuevo Rol.</li> </ul>

Insertar un rol fallida.	I	N/A	I	No se adiciona a la base de datos el nuevo rol informando al usuario del error ocurrido.	<ul style="list-style-type: none"> <li>• Principal.</li> <li>• Panel de Control.</li> <li>• Gestionar Rol.</li> <li>• Nuevo Rol.</li> </ul>
	V	N/A	I	No se adiciona a la base de datos el nuevo rol informando al usuario del error ocurrido.	<ul style="list-style-type: none"> <li>• Principal.</li> <li>• Panel de Control.</li> <li>• Gestionar Rol.</li> <li>• Nuevo Rol.</li> </ul>
	I	N/A	V	No se adiciona a la base de datos el nuevo rol informando al usuario del error ocurrido.	<ul style="list-style-type: none"> <li>• Principal.</li> <li>• Panel de Control.</li> <li>• Gestionar Rol.</li> <li>• Nuevo Rol.</li> </ul>

## SC 2 Modificar Rol.

Escenario	Nombre	Descripción	Funcionalidades	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
Modificar un rol exitosamente.	V	N/A	V	Se actualiza en la base de datos el rol seleccionado.		<ul style="list-style-type: none"> <li>• Principal.</li> <li>• Panel de Control.</li> <li>• Gestionar Rol.</li> <li>• Editar.</li> </ul>
Modificar un rol fallida.	I	N/A	I	No se actualiza en la base de datos el rol seleccionado informando al usuario del error ocurrido.		<ul style="list-style-type: none"> <li>• Principal.</li> <li>• Panel de Control.</li> <li>• Gestionar Rol.</li> <li>• Editar.</li> </ul>
	I	N/A	V	No se actualiza en la base de datos el rol seleccionado informando al usuario del error ocurrido.		<ul style="list-style-type: none"> <li>• Principal.</li> <li>• Panel de Control.</li> <li>• Gestionar Rol.</li> <li>• Editar.</li> </ul>

	V	N/A	I	No se actualiza en la base de datos el rol seleccionado informando al usuario del error ocurrido.		<ul style="list-style-type: none"> <li>• Principal.</li> <li>• Panel de Control.</li> <li>• Gestionar Rol.</li> <li>• Editar.</li> </ul>
--	---	-----	---	---	--	--

### SC 3 Eliminar Rol.

Escenario	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
Eliminar un rol exitosamente.	Se elimina en la base de datos el rol seleccionado.		<ul style="list-style-type: none"> <li>• Principal.</li> <li>• Panel de Control.</li> <li>• Gestionar Rol.</li> <li>• Borrar.</li> </ul>
Eliminar un rol fallida.	No se elimina en la base de datos el rol seleccionado informando al usuario del error ocurrido.		<ul style="list-style-type: none"> <li>• Principal.</li> <li>• Panel de Control.</li> <li>• Gestionar Rol.</li> <li>• Borrar.</li> </ul>

### SC 4 Mostrar Rol.

Escenario	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
-----------	-----------------------	------------------------	---------------

Mostrar un rol exitosamente.	Se muestra el rol seleccionado.		<ul style="list-style-type: none"> <li>• Principal.</li> <li>• Panel de Control.</li> <li>• Gestionar Rol.</li> <li>• Mostrar.</li> </ul>
Mostrar un rol fallida.	No se muestra el rol seleccionado informando al usuario del error ocurrido.		<ul style="list-style-type: none"> <li>• Principal.</li> <li>• Panel de Control.</li> <li>• Gestionar Rol.</li> <li>• Mostrar.</li> </ul>

### SC 5 Asignar Rol.

Escenario	Usuario	Rol	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
Asignar un rol exitosamente.	V	V	Se adiciona a la base de datos la asignación realizada.		<ul style="list-style-type: none"> <li>• Principal.</li> <li>• Panel de Control.</li> <li>• Gestionar Rol.</li> <li>• Asignar Rol.</li> </ul>
Asignar un rol fallida.	I	I	No se asigna el rol al usuario seleccionado informando al usuario del error ocurrido.		<ul style="list-style-type: none"> <li>• Principal.</li> <li>• Panel de Control.</li> <li>• Gestionar Rol.</li> <li>• Asignar Rol.</li> </ul>
	V	I	No se asigna el rol al usuario seleccionado informando al usuario del error ocurrido.		<ul style="list-style-type: none"> <li>• Principal.</li> <li>• Panel de Control.</li> <li>• Gestionar Rol.</li> <li>• Asignar Rol.</li> </ul>



	I	V	No se asigna el rol al usuario seleccionado informando al usuario del error ocurrido.		<ul style="list-style-type: none"> <li>• Principal.</li> <li>• Panel de Control.</li> <li>• Gestionar Rol.</li> <li>• Asignar Rol.</li> </ul>
--	---	---	---	--	---

### 3.3.4 Buscar Rol.

**Descripción General:** El caso de uso se inicializa cuando el Usuario Avanzado del Sistema desea buscar según un criterio un rol o un conjunto de roles.

**Condiciones de ejecución:** El Usuario Avanzado del Sistema debe estar previamente autenticado en el sistema y tener los privilegios para realizar cualquiera de las acciones mencionadas anteriormente.

#### Secciones a probar en el Caso de Uso.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
<b>SC 1:</b> Buscar rol.	<b>EC 1.1:</b> Buscar un rol exitosamente.	El sistema muestra una lista con todos los roles que coincidieron en el criterio entrado finalizando el caso de uso.
	<b>EC 1.2:</b> Buscar un rol fallida.	El sistema muestra un mensaje informando cuál fue el problema encontrado finalizando el caso de uso.

#### Descripción de variable.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Criterio	Campo de Texto	No	Este campo es para la entrada del criterio para la búsqueda del rol.

### SC 1 Buscar Rol.

Escenario	Criterio	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
Buscar un rol exitosamente.	V	Se buscan en la base de datos los roles que coincidan con el criterio entrado.		<ul style="list-style-type: none"><li>• Principal.</li><li>• Panel de Control.</li><li>• Gestionar Rol.</li></ul>
Buscar un rol fallida.	I	Se muestra un mensaje informando cuál fue el problema ocurrido.		<ul style="list-style-type: none"><li>• Principal.</li><li>• Panel de Control.</li><li>• Gestionar Rol.</li></ul>

### 3.4 Conclusiones parciales.

En este capítulo se realizó el diseño de los casos de prueba de las principales funcionalidades, siendo éstos de gran importancia para poder determinar si el sistema cumple con los requerimientos definidos por el cliente.

Es importante acotar que existen otros tipos de pruebas que se le pueden realizar a un software pero el diseño construido, constituye un elemento primordial para detectar errores tempranamente, además, de simplificar y agilizar el flujo de trabajo de Pruebas.

## Conclusiones

Una vez finalizada la investigación se ha llegado a una serie de conclusiones las cuales se especifican a continuación:

- ❖ En su mayoría, las tecnologías, lenguajes de programación, herramientas así como el servidor web son software libre y de esta manera se apoya la premisa de soberanía tecnológica impulsada por el país.
- ❖ El diseño propuesto por el analista está bien estructurado, organizado y orientado al implementador facilitando así la construcción del sistema.
- ❖ Los estándares de codificación empleados potencian la legibilidad del código implementado y facilitan su mantenimiento.
- ❖ Los casos de prueba diseñados permiten la detección de errores tempranamente en el sistema así como agilizar y simplificar el flujo de trabajo de Pruebas.
- ❖ La reutilización de componentes ayudó a disminuir el tiempo dedicado a la implementación.
- ❖ Se implementaron el 91% de los requisitos funcionales previstos para la primera versión del sistema.
- ❖ Se obtuvo el sistema GESTAPRO para la gestión de los recursos humanos en los polos productivos de la Facultad 9 así como la documentación generada por el implementador en el desarrollo de su rol.

## Recomendaciones

Al finalizar la investigación se recomienda:

- ❖ Realizar un estudio de posibles mejoras para facilitar la adaptabilidad del sistema a los continuos cambios en las estructuras productivas y a las necesidades de diferentes clientes.
- ❖ Comenzar el flujo de trabajo de Pruebas con el objetivo de validar totalmente el sistema.
- ❖ Realizar una segunda iteración con el objetivo de incluir otras funcionalidades al sistema con los nuevos requerimientos que surjan por necesidades del cliente.

## Referencias Bibliográficas

1. **Alcalde, Verónica Fuente.** Mejora la gestión de Recursos Humanos. *Mailxmail.com*. [En línea] INTERCOM, 27 de 09 de 2007. [Citado el: 10 de 12 de 2009.] <http://www.mailxmail.com/curso-mejora-gestion-recursos-humanos/concepto-recursos-humanos>.
2. **Piñones, Ariel Rementería.** Gestión. Alcance de gestión. *Gestión de Contenidos Electrónicos*. [En línea] [Citado el: 10 de 12 de 2009.] <http://biblioteca.idict.villaclara.cu/UserFiles/File/CI%20Gestion%20de%20Ciencia%20e%20Innovacion%20Tecnologica/7.doc.doc..>
3. **Alcántara, José Luis Caballano.** INTRODUCCIÓN A LA GESTIÓN DE LOS RECURSOS HUMANOS. *El Prisma. Biblioteca Virtual*. [En línea] 2001. [Citado el: 10 de 12 de 2009.] [http://www.elprisma.com/apuntes/administracion\\_de\\_empresas/gestionempresarialrecursoshumanos/](http://www.elprisma.com/apuntes/administracion_de_empresas/gestionempresarialrecursoshumanos/).
4. **Dra. Sonia Fleitas Triana.** Modelando el proceso de gestión de recursos humanos. *Red Capital Humano*. [En línea] 2008. [Citado el: 10 de 12 de 2009.] [http://www.gecyt.cu/redcapitalhumano/ponencias/p\\_51.pdf](http://www.gecyt.cu/redcapitalhumano/ponencias/p_51.pdf).
5. **Aniuska Grotestán Columbié, Hanssel Vidal Pelaez.** SISTEMA GESTIÓN DE RECURSOS HUMANOS EN ASTRO. *Biblioteca UCI*. [En línea] 2007. [Citado el: 10 de 12 de 2009.] [http://bibliodoc.uci.cu/TD/TD\\_0848\\_07.pdf](http://bibliodoc.uci.cu/TD/TD_0848_07.pdf).
6. Metodologías usadas en ingeniería del software. *Universidad de Murcia*. [En línea] Universidad de Murcia, 10 de 05 de 2007. [Citado el: 15 de 01 de 2010.] <http://www.um.es/docencia/barzana/IAGP/lagp3.html>.
7. Lenguajes de programación. *Kioskea*. [En línea] 16 de 10 de 2008. [Citado el: 17 de 01 de 2010.] <http://es.kioskea.net/contents/langages/langages.php3>.
8. **Alvarez, Miguel Angel.** Qué es PHP. *Desarrollo Web*. [En línea] 09 de 05 de 2001. [Citado el: 20 de 01 de 2010.] <http://www.desarrolloweb.com/articulos/392.php>.
9. **Vallejo, Luis Díaz.** *Propuesta de arquitectura para el sistema de gestión automatizado de recursos humanos GESTAPRO*. Ciudad de la Habana : UCI, 2009.
10. **Programación, Dpto Técnicas de.** Introducción a las tecnologías del lado del servidor. *Entorno Virtual de Aprendizaje*. [En línea] 2009. [Citado el: 20 de 01 de 2010.] <http://eva.uci.cu/mod/resource/view.php?id=20017>.

11. **Dondo, Agustín.** Programación en Castellano. *¿Por qué elegir PHP?* [En línea] 01 de 03 de 2002. [Citado el: 22 de 01 de 2010.] <http://www.programacion.com/php/articulo/porquephp/>.
12. **Altavoz.** Altavoz. *¿Qué es CSS?* [En línea] 2006. [Citado el: 22 de 01 de 2010.] [http://www.altavoz.net/prontus\\_altavoz/antialone.html?page=http://www.altavoz.net/prontus\\_altavoz/site/artic/20061020/pags/20061020125319.html](http://www.altavoz.net/prontus_altavoz/antialone.html?page=http://www.altavoz.net/prontus_altavoz/site/artic/20061020/pags/20061020125319.html).
13. **Bonilla, Luis.** Cursos Multimedia. *Introducción a JavaScript.* [En línea] 2002. [Citado el: 23 de 01 de 2010.] <http://www.cursopaginasweb.net/curso-paginas-web/introduccion-javascript.htm>.
14. **CAVSI.** CAVSI. *¿Qué es un Sistema Gestor de Bases de Datos o SGBD?* . [En línea] [Citado el: 24 de 01 de 2010.] <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.
15. **Alvarez, Sara.** Desarrollo Web. *Sistemas gestores de bases de datos.* [En línea] 31 de 07 de 2007. [Citado el: 23 de 01 de 2010.] <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>.
16. **Pecos, Daniel.** danielpecos.com. *PostGreSQL vs. MySQL.* [En línea] 2008. [Citado el: 23 de 01 de 2010.] [http://www.netpecos.org/docs/mysql\\_postgres/x15.html](http://www.netpecos.org/docs/mysql_postgres/x15.html).
17. TiendaLinux.com. *Ventajas de PostgreSQL.* [En línea] 31 de 05 de 2003. [Citado el: 23 de 01 de 2010.] [http://soporte.tiendalinux.com/portal/Portfolio/postgresql\\_ventajas\\_html](http://soporte.tiendalinux.com/portal/Portfolio/postgresql_ventajas_html).
18. **Cuenca, Carlos Luis.** Desarrollo Web. *Desarrollo Web.* [En línea] 03 de Marzo de 2003. [Citado el: 24 de Enero de 2010.] <http://www.desarrolloweb.com/articulos/1112.php>.
19. **Kamthan, Pankaj.** Irt.org. *Irt.org.* [En línea] 1 de Agosto de 1999. [Citado el: 24 de Enero de 2010.] <http://www.irt.org/articles/js177/index.htm>.
20. **Valdés, Damián Pérez.** Radio Caribe. *Noticias de Informática.* [En línea] [Citado el: 23 de 01 de 2010.] <http://www.radiocaribe.co.cu/Secundaria/Informatica/342.htm>.
21. **Moreno, Raul Vasconcelos.** UOC. *Reseña sobre Dreamweaver.* [En línea] 19 de 05 de 2009. [Citado el: 23 de 01 de 2010.] [http://multimedia.uoc.edu/~usuaris/wiki/index.php/Rese%C3%B1a\\_sobre\\_Dreamweaver/\\_Raul\\_Vasconcelos\\_Moreno#Ventajas\\_y\\_Desventajas](http://multimedia.uoc.edu/~usuaris/wiki/index.php/Rese%C3%B1a_sobre_Dreamweaver/_Raul_Vasconcelos_Moreno#Ventajas_y_Desventajas).
22. **Aguila, Adrián Gracia.** *Análisis y diseño de un sistema automatizado para el control de los recursos humanos en los polos productivos de la Facultad 9.* Ciudad de la Habana : UCI, 2009.

23. **Mora, Dunier Domínguez.** *Implementación de los Subsistemas Web y Transferencia de la Plataforma de Transmisión Abierta para Radio y Televisión.* Ciudad de la Habana : UCI, 2009.
24. **Computacional, Programa de Genómica.** Centro de Ciencias Genómicas. *Estándar de Codificación.* [En línea] 2010 de 07 de 2009. [http://www.ccg.unam.mx/en/webfm\\_send/1216](http://www.ccg.unam.mx/en/webfm_send/1216).
25. **Autores, Colectivo de.** Teleformación. [En línea] 2009-2010. [Citado el: 3 de 5 de 2010.] [http://eva.uci.cu/file.php/259/Curso\\_2009-2010/Conferencia\\_7/Materiales\\_basicos/Conferencia\\_7\\_Disciplina\\_Prueba.doc](http://eva.uci.cu/file.php/259/Curso_2009-2010/Conferencia_7/Materiales_basicos/Conferencia_7_Disciplina_Prueba.doc).

## Bibliografía

1. **Alcalde, Verónica Fuente.** Mejora la gestión de Recursos Humanos. *Mailxmail.com*. [En línea] INTERCOM, 27 de 09 de 2007. [Citado el: 10 de 12 de 2009.] <http://www.mailxmail.com/curso-mejora-gestion-recursos-humanos/concepto-recursos-humanos>.
2. **Piñones, Ariel Rementeria.** Gestión. Alcance de gestión. *Gestión de Contenidos Electrónicos*. [En línea] [Citado el: 10 de 12 de 2009.] <http://biblioteca.idict.villaclara.cu/UserFiles/File/CI%20Gestion%20de%20Ciencia%20e%20Innovacion%20Tecnologica/7.doc.doc..>
3. **Alcántara, José Luis Caballano.** INTRODUCCIÓN A LA GESTIÓN DE LOS RECURSOS HUMANOS. *El Prisma. Biblioteca Virtual*. [En línea] 2001. [Citado el: 10 de 12 de 2009.] [http://www.elprisma.com/apuntes/administracion\\_de\\_empresas/gestionempresarialrecursoshumanos/](http://www.elprisma.com/apuntes/administracion_de_empresas/gestionempresarialrecursoshumanos/).
4. **Dra. Sonia Fleitas Triana.** Modelando el proceso de gestión de recursos humanos. *Red Capital Humano*. [En línea] 2008. [Citado el: 10 de 12 de 2009.] [http://www.gecycu/redcapitalhumano/ponencias/p\\_51.pdf](http://www.gecycu/redcapitalhumano/ponencias/p_51.pdf).
5. **Aniuska Grotestán Columbié, Hanssel Vidal Pelaez.** SISTEMA GESTIÓN DE RECURSOS HUMANOS EN ASTRO. *Biblioteca UCI*. [En línea] 2007. [Citado el: 10 de 12 de 2009.] [http://bibliodoc.uci.cu/TD/TD\\_0848\\_07.pdf](http://bibliodoc.uci.cu/TD/TD_0848_07.pdf).
6. Metodologías usadas en ingeniería del software. *Universidad de Murcia*. [En línea] Universidad de Murcia, 10 de 05 de 2007. [Citado el: 15 de 01 de 2010.] <http://www.um.es/docencia/barzana/LAGP/lagp3.html>.
7. Lenguajes de programación. *Kioskea*. [En línea] 16 de 10 de 2008. [Citado el: 17 de 01 de 2010.] <http://es.kioskea.net/contents/langages/langages.php3>.
8. **Rea, Héctor Zárate.** Paradigmas de Programacion. *Scribd*. [En línea] 11 de 2008. [Citado el: 15 de 01 de 2010.] <http://www.scribd.com/doc/9174723/Paradigmas-de-Programacion>.
9. Paradigmas de Programación. *Vandal Online*. [En línea] 20minutos.es, 24 de 07 de 2006. [Citado el: 17 de 01 de 2010.] <http://blogs.vandal.net/3996/vm/1750312472006>.
10. PARADIGMAS DE PROGRAMACION. *Universidad Tecnológica Nacional. Facultad Regional Tucumán*. [En línea] Universidad Tecnológica Nacional, 2008. [Citado el: 17 de 01 de 2010.] <http://www.frt.utn.edu.ar/sistemas/paradigmas/page22.html>.



11. Teoría de los Paradigmas. [En línea] 15 de 08 de 2003. [Citado el: 17 de 01 de 2010.] [http://www.wilucha.com.ar/Paradigma/A\\_Paradigma.html](http://www.wilucha.com.ar/Paradigma/A_Paradigma.html).
12. **Alvarez, Miguel Angel.** Qué es PHP. *Desarrollo Web*. [En línea] 09 de 05 de 2001. [Citado el: 20 de 01 de 2010.] <http://www.desarrolloweb.com/articulos/392.php>.
13. **Dondo, Agustín.** ¿Por qué elegir PHP? *Programación en castellano*. [En línea] 2007. [Citado el: 10 de 01 de 2010.] <http://www.programacion.com/php/articulo/porquephp/>.
14. **Programación, Dpto Técnicas de.** Introducción a las tecnologías del lado del servidor. *Entorno Virtual de Aprendizaje*. [En línea] 2009. [Citado el: 20 de 01 de 2010.] <http://eva.uci.cu/mod/resource/view.php?id=20017>.
15. **Programación, Departamento de Técnicas de.** Principales Etiquetas. *Entorno Virtual de Aprendizaje*. [En línea] 2009. [Citado el: 20 de 01 de 2010.] <http://eva.uci.cu/mod/resource/view.php?id=20077>.
16. Que es HTML. *Espaweb*. [En línea] 2008. [Citado el: 20 de 01 de 2010.] <http://www.espaweb.com/html.php>.
17. ¿Qué es HTML? *HTML.net*. [En línea] [Citado el: 20 de 01 de 2010.] <http://es.html.net/tutorials/html/lesson2.asp>.
18. ¿Qué es CSS? *Maestros de la Web*. [En línea] 08 de 11 de 2003. [Citado el: 21 de 01 de 2010.] <http://www.maestrosdelweb.com/editorial/introcss/>.
19. **Lazaro, Juliana Monteiro.** Qué es CSS. *Desarrollo Web*. [En línea] [Citado el: 21 de 01 de 2010.] <http://www.desarrolloweb.com/articulos/26.php>.
20. Guía Breve de CSS. *W3C*. [En línea] W3C, 09 de 01 de 2008. [Citado el: 21 de 01 de 2010.] <http://www.w3c.es/divulgacion/guiasbreves/HojasEstilo>.
21. ¿Qué es CSS? *Altavoz*. [En línea] [Citado el: 21 de 01 de 2010.]
22. **Alvarez, Miguel Angel.** Características y ventajas de las CSS. *Desarrollo Web*. [En línea] [Citado el: 21 de 01 de 2010.] <http://www.desarrolloweb.com/articulos/182.php>.
23. —. Qué es Javascript. *Desarrollo Web*. [En línea] [Citado el: 24 de 01 de 2010.] <http://www.desarrolloweb.com/articulos/25.php>.
24. ¿Qué es Javascript? *Maestros de la Web*. [En línea] 03 de 07 de 2007. [Citado el: 21 de 01 de 2010.] <http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/>.
25. Concepto de JavaScript. *Mozilla Developer Center*. [En línea] 05 de 02 de 2009. [Citado el: 24 de 01 de 2010.]

[https://developer.mozilla.org/index.php?title=Es/Gu%C3%ADa\\_JavaScript\\_1.5/Concepto\\_de\\_JavaScript](https://developer.mozilla.org/index.php?title=Es/Gu%C3%ADa_JavaScript_1.5/Concepto_de_JavaScript).

26. PostGreSQL vs. MySQL. *Daniel Pecos.com*. [En línea] [Citado el: 26 de 01 de 2010.] [http://www.netpecos.org/docs/mysql\\_postgres/x15.html](http://www.netpecos.org/docs/mysql_postgres/x15.html).

27. Ventajas de PostgreSQL. *TiendaLinux.com*. [En línea] 31 de 05 de 2003. [Citado el: 26 de 01 de 2010.] [http://soporte.tiendalinux.com/portal/Portfolio/postgresql\\_ventajas\\_html](http://soporte.tiendalinux.com/portal/Portfolio/postgresql_ventajas_html).

28. **Vallejo, Luis Díaz**. *Propuesta de arquitectura para el sistema de gestión automatizado de recursos humanos GESTAPRO*. Ciudad de la Habana : UCI, 2009.

29. **Alvarez, Miguel Angel**. Desarrollo Web. [En línea] 09 de 05 de 2001. [Citado el: 20 de 01 de 2010.] <http://www.desarrolloweb.com/articulos/392.php>.

30. —. Desarrollo Web. *Que es PHP*. [En línea] 09 de 05 de 2001. [Citado el: 20 de 01 de 2010.] <http://www.desarrolloweb.com/articulos/392.php>.

31. **Piñones, Ariel Rementeria**. Gestión. Alcance de gestión. *Gestión de Contenidos Electrónicos*. [En línea] [Citado el: 10 de 12 de 2009.] <http://biblioteca.idict.villaclara.cu/UserFiles/File/CI%20Gestion%20de%20Ciencia%20e%20Innovacion%20Tecnologica/7.doc.doc..>

32. **Dondo, Agustín**. Programación en Castellano. *¿Por qué elegir PHP?* [En línea] 01 de 03 de 2002. [Citado el: 22 de 01 de 2010.] <http://www.programacion.com/php/articulo/porquephp/>.

33. **Altavoz**. Altavoz. *¿Qué es CSS?* [En línea] 2006. [Citado el: 22 de 01 de 2010.] [http://www.altavoz.net/prontus\\_altavoz/antialone.html?page=http://www.altavoz.net/prontus\\_altavoz/site/artic/20061020/pags/20061020125319.html](http://www.altavoz.net/prontus_altavoz/antialone.html?page=http://www.altavoz.net/prontus_altavoz/site/artic/20061020/pags/20061020125319.html).

34. **Bonilla, Luis**. Cursos Multimedia. *Introducción a JavaScript*. [En línea] 2002. [Citado el: 23 de 01 de 2010.] <http://www.cursopaginasweb.net/curso-paginas-web/introduccion-javascript.htm>.

35. **CAVSI**. CAVSI. *¿Qué es un Sistema Gestor de Bases de Datos o SGBD?*. [En línea] [Citado el: 24 de 01 de 2010.] <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.

36. **Alvarez, Sara**. Desarrollo Web. *Sistemas gestores de bases de datos*. [En línea] 31 de 07 de 2007. [Citado el: 23 de 01 de 2010.] <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>.

37. **Pecos, Daniel**. *danielpecos.com*. *PostGreSQL vs. MySQL*. [En línea] 2008. [Citado el: 23 de 01 de 2010.] [http://www.netpecos.org/docs/mysql\\_postgres/x15.html](http://www.netpecos.org/docs/mysql_postgres/x15.html).

38. TiendaLinux.com. *Ventajas de PostgreSQL*. [En línea] 31 de 05 de 2003. [Citado el: 23 de 01 de 2010.]  
[http://soporte.tiendalinux.com/portal/Portfolio/postgresql\\_ventajas\\_.html](http://soporte.tiendalinux.com/portal/Portfolio/postgresql_ventajas_.html).
39. **Valdés, Damián Pérez**. Radio Caribe. *Noticias de Informática*. [En línea] [Citado el: 23 de 01 de 2010.] <http://www.radiocaribe.co.cu/Secundaria/Informatica/342.htm>.
40. **Moreno, Raul Vasconcelos**. UOC. *Reseña sobre Dreamweaver*. [En línea] 19 de 05 de 2009. [Citado el: 23 de 01 de 2010.]  
[http://multimedia.uoc.edu/~usuaris/wiki/index.php/Rese%C3%B1a\\_sobre\\_Dreamweaver/\\_Raul\\_Vasconcelos\\_Moreno#Ventajas\\_y\\_Desventajas](http://multimedia.uoc.edu/~usuaris/wiki/index.php/Rese%C3%B1a_sobre_Dreamweaver/_Raul_Vasconcelos_Moreno#Ventajas_y_Desventajas).
41. **Computacional, Programa de Genómica**. Centro de Ciencias Genómicas. *Estándar de Codificación*. [En línea] 2010 de 07 de 2009.  
[http://www.ccg.unam.mx/en/webfm\\_send/1216](http://www.ccg.unam.mx/en/webfm_send/1216).
42. **Autores, Colectivo de**. Teleformación. [En línea] 2009-2010. [Citado el: 3 de 5 de 2010.] [http://eva.uci.cu/file.php/259/Curso\\_2009-2010/Conferencia\\_7/Materiales\\_basicos/Conferencia\\_7\\_Disciplina\\_Prueba.doc](http://eva.uci.cu/file.php/259/Curso_2009-2010/Conferencia_7/Materiales_basicos/Conferencia_7_Disciplina_Prueba.doc).
43. **Cuenca, Carlos Luis**. Desarrollo Web. *Desarrollo Web*. [En línea] 03 de Marzo de 2003. [Citado el: 24 de Enero de 2010.]  
<http://www.desarrolloweb.com/articulos/1112.php>.
44. **Kamthan, Pankaj**. Irt.org. *Irt.org*. [En línea] 1 de Agosto de 1999. [Citado el: 24 de Enero de 2010.] <http://www.irt.org/articles/js177/index.htm>.
45. **Aguila, Adrián Gracia**. *Análisis y diseño de un sistema automatizado para el control de los recursos humanos en los polos productivos de la Facultad 9*. Ciudad de la Habana : UCI, 2009.

## Glosario de Términos

**Ajax:** Asynchronous JavaScript And XML o JavaScript asíncrono y XML, técnica de desarrollo web para crear aplicaciones interactivas.

**AOLServer:** Servidor web de código abierto de América Online.

**API:** Interfaz de programación de aplicaciones o application programming interface, conjunto de funciones, procedimientos o métodos que ofrece una librería para ser utilizado por otro software.

**Array:** Matriz o vector, zona de almacenamiento contiguo que contiene una serie de elementos del mismo tipo.

**ASCII:** American Standard Code for Information Interchange o Código Estadounidense Estándar para el Intercambio de Información.

**ASP:** Active Server Pages, tecnología de Microsoft del tipo "lado del servidor" para páginas web generadas dinámicamente.

**ColdFusion:** Lenguaje de programación usado para desarrollar aplicaciones de Internet, generalmente sitios web generados dinámicamente.

**DB2:** Motor de base de datos relacional que integra XML de manera nativa, que permite almacenar documentos completos dentro del tipo de datos XML.

**Framework:** Marco de trabajo, esquema para el desarrollo y/o la implementación de una aplicación.

**HTTP:** Hypertext Transfer Protocol o Protocolo de Transferencia de Hipertexto.

**IMAP:** Internet Message Access Protocol, protocolo de red de acceso a mensajes electrónicos almacenados en un servidor.

**Informix:** Familia de productos gestores de base de datos de IBM.

**JavaScript:** Lenguaje de programación interpretado utilizado principalmente en páginas web.

**MS SQL:** Microsoft SQL Server, sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional.

**MVCC:** Control de Concurrencia para Múltiples Versiones.

**MySQL:** Sistema de gestión de base de datos relacional, multihilo y multiusuario.

**Oracle:** Sistema de gestión de base de datos relacional.

**PDF:** Portable Document Format o Formato de Documento Portátil, formato de almacenamiento de documentos.

**pgAdmin:** Aplicación gráfica para gestionar el gestor de bases de datos.

**POP:** Post Office Protocol (Protocolo de la oficina de correo), se utiliza en clientes locales de correo para obtener los mensajes de correo electrónico almacenados en un servidor remoto.

**Rational Data Architect:** Herramienta de diseño de modelado e integración de datos.

**Roxen WebServer:** Servidor web, multiplataforma y con una base de datos MySQL integrada.

**Script:** Guión o conjunto de instrucciones.

**SCRUM:** Proceso en el que se aplican de manera regular un conjunto de mejores prácticas para trabajar en equipo y obtener el mejor resultado posible de un proyecto.

**Solaris:** Sistema operativo de tipo Unix desarrollado por Sun Microsystems.

**SQL:** Structured Query Language o lenguaje de consulta estructurado.

**THTTPD:** tiny/turbo/throttling HTTP server, servidor web de código libre disponible para la mayoría de las variantes de Unix.

**TIC's:** Tecnologías de la información y la comunicación.

**Tora:** Conjunto de herramientas multiplataforma de software libre creado para ayudar a los administradores y desarrolladores de aplicaciones de bases de datos Oracle.

**UNIX:** Sistema operativo portable, multitarea y multiusuario.

**W3C:** World Wide Web Consortium, consorcio internacional donde las organizaciones miembro, personal a tiempo completo y el público en general, trabajan conjuntamente para desarrollar estándares Web.

**XHTML:** Extensible Hypertext Markup Language (Lenguaje Extensible de Marcado de Hipertexto), lenguaje de marcado pensado para sustituir a HTML como estándar para las páginas web.

**XML:** Extensible Markup Language o Lenguaje de Marcado Extensible, permite definir la gramática de lenguajes específicos.