

**Universidad de las Ciencias Informáticas**

**Facultad 6**



**Título:** Sistema de Control y Asistencia a Actividades de la Producción.

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas.

**Autor:**

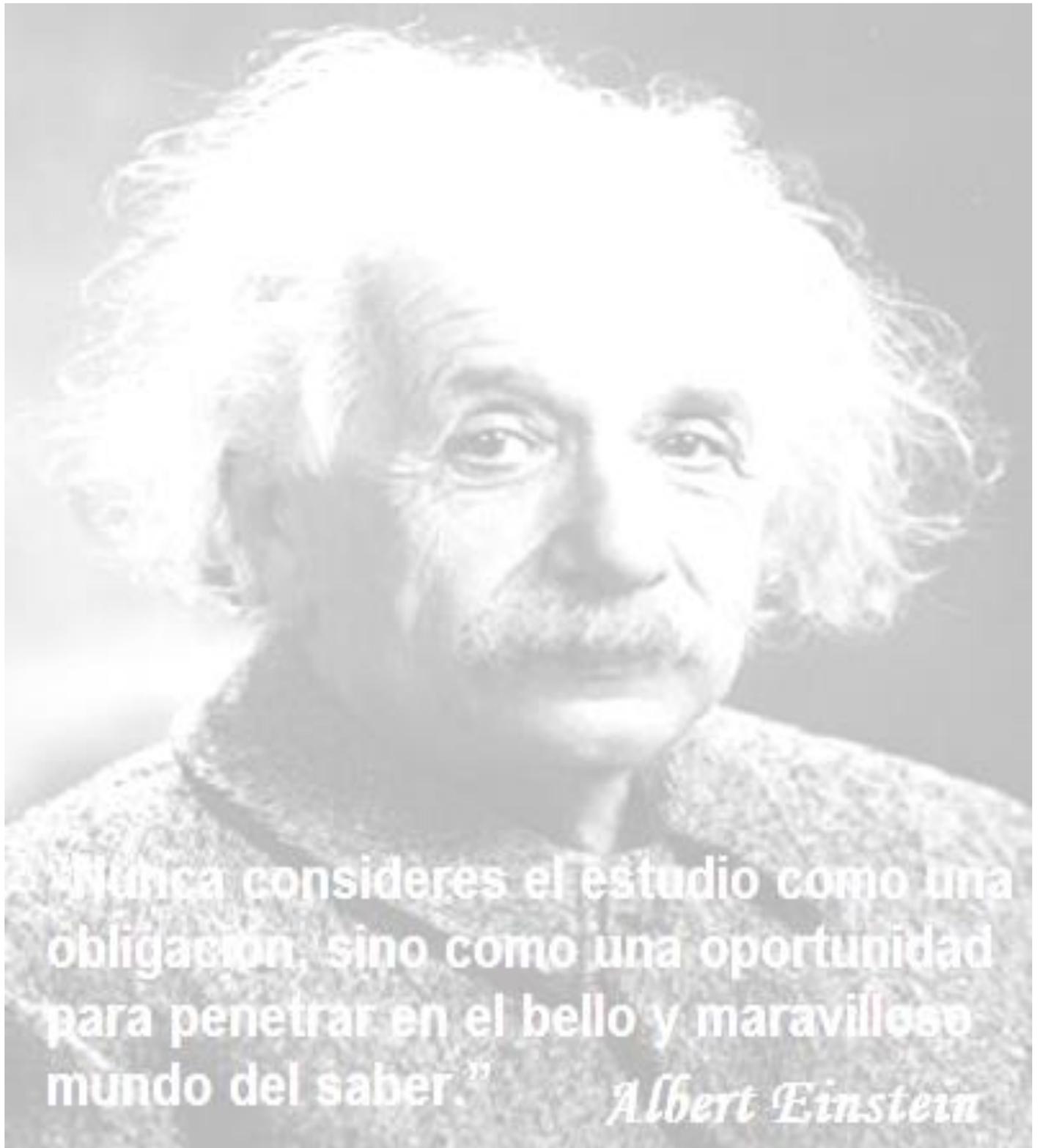
Aliú Cuesta Jiménez

**Tutor:**

M.SC. Yuniel Eliades Proenza Arias

**La Habana, julio de 2016**

**“Año 58 de la revolución”**



Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber.”  
*Albert Einstein*

**DECLARATORIA DE AUTORÍA**

Declaramos ser los autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Aliú Cuesta Jiménez

M.Sc. Yuniel Eliades Proenza Arias

\_\_\_\_\_

Firma del Autor

\_\_\_\_\_

Firma del Tutor

### **DATOS DEL CONTACTO**

**Datos del Tutor:**

**Nombre y Apellidos:** M.Sc. Yuniel Eliades Proenza Arias

**Correo electrónico:** [yproenza@uci.cu](mailto:yproenza@uci.cu)

**Institución:** Universidad de las Ciencias Informáticas.

**Síntesis del Tutor:** Graduado de Ingeniería Informática en el año 2006 de la Universidad de Holguín y CUJAE. Máster en Ingeniería de Software e Inteligencia Artificial por la Universidad de Málaga en el 2011. Se ha desempeñado como analista y desarrollador de aplicaciones Web y Desktop. Tiene experiencia en el trabajo con Sistemas de Información Geográfica y el desarrollo de Ontologías.

**DEDICATORIA**

*A mi mamá por su apoyo, por guiarme, entenderme y creer en mí en los momentos más difícil.*

### AGRADECIMIENTOS

*A todos los que han contribuido, de una forma u otra a mi formación como profesional, especialmente: A toda mi familia por el apoyo incondicional y la confianza que han depositado en mí, especialmente a mis padres, por el amor, cariño y comprensión que me han dado en todo momento. A mis amistades, por tener tanta paciencia, ayudarme en lo que he necesitado y darme toda su comprensión. A todos mis buenos amigos, por pasar tan agradables y a veces no tan agradables momentos, pero siempre juntos, y por hacerme saber que en cualquier momento puedo contar con ellos, al igual que ellos conmigo.*

*A todas estas personas especiales para mí, quiero darle las gracias por brindarme toda su ayuda, apoyo y comprensión para poder lograr hacer mi sueño realidad.*

## RESUMEN

La Universidad de las Ciencias Informáticas es una universidad productiva, cuya misión es producir software y brindar servicios informáticos a partir de la vinculación estudio-trabajo como modelo de formación. La necesidad de mejorar los procesos de control y asistencia a las actividades de formación y producción está impulsando a la adopción de sistemas automatizados para la administración de dicho proceso. Debido al hecho de que el control de la asistencia se lleve a cabo de forma manual, se realizó la presente investigación que consiste, en la implementación de un Sistema de Control y Asistencia a Actividades de la Producción para el Departamento de Practica Profesional del centro Geoinformática y Señales Digitales. Para cumplir con los objetivos trazados, el sistema brindará la posibilidad de definir sesiones de producción, establecer un horario (teniendo en cuenta las sesiones de producción), generar reportes en base a los registros de asistencia almacenados. La implementación del sistema está basada en un ambiente web, utilizando tecnologías libres.

### **Palabras claves:**

Servicios automatizados, sesiones de producción, reportes.

## **ABSTRACT**

The University of Informatics Sciences is a productive university whose mission is to produce computer software and provide services from the linkage study-work as a training model. The need to improve control processes and assistance for training and production is driving the adoption of automated systems for managing this process. Due to the fact that the control of assistance is carried out manually, this research consists on implementing a Control and Assistance to production activities System for the Department of Professional Practice of Geoinformatics and Digital Signals center was held. To fulfill the goals, the system will provide the possibility to define production sessions, set up a schedule (taking into account production sessions), to generate reports based on stored attendance records. The implementation of the system is based on a web environment, using free technologies.

Keywords :

Automated services, production sessions, reports.

## ÍNDICE DE CONTENIDO

INTRODUCCIÓN .....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA .....	5
1.1 Introducción.....	5
1.2 Conceptos Asociados al Dominio del Problema .....	5
1.3 Análisis de las Soluciones Existentes .....	6
1.4 Tecnologías a Utilizar .....	13
1.5 Conclusiones.....	19
CAPÍTULO 2: ANÁLISIS Y DISEÑO DE LA SOLUCIÓN PROPUESTA .....	20
2.1 Introducción.....	20
2.2 Propuesta de Solución .....	20
2.3 Modelo de Dominio.....	21
2.4 Requisitos Funcionales .....	22
2.5 Requisitos no Funcionales.....	24
2.6 Modelo de Casos de Uso .....	25
2.6.1 Descripción de los Actores.....	25
2.6.2 Diagrama de Caso de Uso del Sistema (DCUS) .....	26
2.6.3 Descripción Textual de los Caso de Uso del Sistema .....	27
2.7 Diagrama de Clases de Diseño .....	32
2.8 Arquitectura del Sistema.....	33
2.9 Patrones de Diseño .....	34
2.10 Conclusiones.....	35
CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS .....	36
3.1 Introducción.....	36

# Índice de Contenido

---

3.2 Modelo de Datos .....	36
3.3 Diagrama de Componente.....	37
3.4 Diagrama de Despliegue .....	38
3.5 Pruebas de la Solución.....	39
3.6 Conclusiones.....	45
CONCLUSIONES .....	46
RECOMENDACIONES .....	47
BIBLIOGRAFÍA.....	48

**ÍNDICE DE TABLAS**

Tabla 1 Actores del sistema .....25

Tabla 2 CUS Gestionar PC .....27

Tabla 3 Escenarios CUS Gestionar PC.....40

Tabla 4. Variables CUS Gestionar PC. ....41

Tabla 5 MD Buscar Consultante. ....42

Tabla 6 MD Mostar computadora.....43

Tabla 7 MD Eliminar Computadora. ....43

Tabla 8 MD Adicionar Computadora. ....44

## ÍNDICE DE FIGURAS

Fig. 1 Esquema de Websockets.....	17
Fig. 2 Modelo de Dominio .....	22
Fig. 3 Diagrama de Caso de Uso del Sistema.....	26
Fig. 4 Diagrama de Clases del Diseño del CU Gestionar PC .....	33
Fig. 5 Patrón GRASP Creador .....	34
Fig. 6 Patrón GRASP Controlador .....	35
Fig. 7 Patrón GRASP Experto.....	35
Fig. 8 Diagrama de Clases Persistentes .....	36
Fig. 9 Modelo Datos.....	37
Fig. 10 Diagrama de Componente del CU Gestionar PC .....	38
Fig. 11 Diagrama de Despliegue.....	39
Fig. 12 Interpretación gráfica de Caja Negra.....	40
Fig. 13 Pruebas de Iteración .....	44

## INTRODUCCIÓN

La utilización de las Tecnologías de la Información y la Comunicación (TIC) ha tenido un papel importante en las actividades que se realizan normalmente en la vida cotidiana. Estas han provocado un cambio positivo en la sociedad con su utilización en las diferentes esferas, porque brindan una mayor calidad y resultados más rápidos en los procesos realizados. Una de las principales tendencias del mundo actual es la informatización de todas esas esferas, ya sea en la medicina, las comunicaciones, la educación, entre otras. Especialmente en la rama de la gestión de recursos humanos la utilización de las TIC ha constituido un resultado relevante debido a la importancia de la información que se genera en ellas.

En el siglo actual, cualquier actividad laboral donde esté presente la entrega de resultados y su evaluación, las personas vinculadas deben estar reguladas en cuanto a distribución de puestos de trabajo, horarios de trabajos y otras características propias del centro.

Llevar el control de asistencia de los empleados de una empresa o centro de producción es una labor muy sencilla en algunos casos y muy compleja en otros. Sencilla, si se cuenta con una persona que registre el ingreso y salida del personal, es todo lo que podría necesitarse; compleja, si se quiere llevar un control de los días de vacaciones tomados por un empleado particular, también tener conocimiento exacto de los estímulos, certificados o descansos médicos, movimientos internos del personal de una estación de trabajo a otra, cambios en los horarios de producción o cuando la institución tiene un gran número de personal.

Pero es claro que se presentaría un gran número de dudas sobre la información que se maneja: si el encargado registra las horas y movimientos a su antojo y criterio; además de ser una labor puramente manual y sujeta a continuos errores.

En la actualidad esta tarea de identificar al trabajador, llevar y/o controlar el registro horario de entrada y salida es manual y se mantiene vigente en muchas instituciones y en empresas grandes es llevada a cabo por un empleado específico, constituyendo esto una tarea agotadora y engorrosa donde se maneja un gran número de información.

La Universidad de las Ciencias Informáticas (UCI) es una universidad productiva, cuya misión es producir software y brindar servicios informáticos a partir de la vinculación estudio-producción como modelo de formación. La UCI cuenta con 6 facultades, en el centro de desarrollo Geoinformática y Señales Digitales (GEYSED) perteneciente a la facultad 6, se encuentra el Departamento de Práctica Profesional (DPP) donde se llevan a cabo el control de asistencia a las actividades de producción por estudiantes y profesores. Actualmente el control de la asistencia en este departamento se desarrolla de forma manual o

por simple observación, lo que provoca la toma de decisiones erróneas con estudiantes y profesores a la hora de emitir una evaluación en la producción. A demás provoca la pérdida por deterioro o mala manipulación de los datos contenidos en el registro de asistencia.

A partir de la problemática planteada surge como **problema a resolver**: ¿Cómo mejorar el Control de la Asistencia de estudiantes y profesionales del Centro GEYSED al horario de formación y producción?

Como consecuencia se determina como **objeto de estudio**: los procesos de control y asistencia referente a las actividades en la producción.

Atendiendo a la situación problémica planteada, al problema a resolver y al objeto de estudio, la presente tesis se estructura y desarrolla en función del siguiente **objetivo general**: desarrollar un Sistema de Control y Asistencia a Actividades de la Producción para el DPP del centro GEYSED que permita el control de la información referente a las labores de producción.

Determinándose como **campo de acción**: la informatización de los procesos referentes al control de asistencia en el DPP del centro GEYSED.

Para guiar la investigación se definen las siguientes **preguntas de la investigación**:

- ¿Cuáles son los fundamentos teórico-metodológicos para desarrollar un Sistema de Control y Asistencia a Actividades de la Producción en el centro GEYSED?
- ¿Cuáles son las similitudes de los sistemas de control de acceso asociados al control de asistencia en las actividades productivas de GEYSED?
- ¿Cómo organizar el proceso de desarrollo del Sistema de Control y Asistencia a Actividades de la Producción?

Para el cumplimiento del objetivo general se realizarán las siguientes **tareas de la investigación**:

- Caracterización de las soluciones existentes del tema a nivel nacional e internacional.
- Caracterización de las herramientas y metodología a utilizar durante el desarrollo de la aplicación.
- Descripción de la arquitectura del sistema.
- Descripción y definición de los requisitos funcionales y no funcionales.
- Implementación del sistema con todas las funcionalidades requeridas.
- Ejecución de pruebas al sistema.

Para alcanzar un cumplimiento y adecuado desarrollo del objetivo planteado se aplicaron un conjunto de métodos científicos:

## **Métodos Teóricos:**

- **Analítico - Sintético:** Se utilizó para estudiar documentos y bibliografías de diferentes autores, para realizar la fundamentación teórica de la investigación y sustentar los conceptos fundamentales que sean necesarios para la solución del problema.
- **Inductivo - Deductivo:** Se utilizó para hacer una generalización del comportamiento de los sistemas de control y asistencia y permitió comparar soluciones informáticas, cuyos objetivos son similares al sistema a implementar. Además permitió deducir cual es la mejor solución para resolver el problema de la investigación.
- **Modelación:** Se utilizó para crear una representación o modelo de los artefactos que se generaron a lo largo de todo el proceso de desarrollo del sistema facilitando un mejor entendimiento de la solución a implementar.

## **Métodos Empíricos:**

- **Entrevista:** Se utilizó en la fase inicial para obtener información relativa al sistema que se quiere construir. Se realizó una serie de preguntas al Jefe de Departamento de Práctica Profesional para lograr el cumplimiento de los objetivos propuestos.

El presente documento está estructurado por 3 capítulos:

- **Capítulo 1 - Fundamentación Teórica:** En este capítulo se enuncian los conceptos relacionados con el sistema que se desarrolla. Se caracterizan las herramientas, la metodología a utilizar en el proceso de desarrollo del software, así como las tecnologías utilizadas. Se mostrará cuáles son las herramientas, tecnologías y tendencias actuales que podemos encontrar en el mercado, útiles para el desarrollo del sistema.
- **Capítulo 2 – Análisis y Diseño de la Solución Propuesta:** En este capítulo se realiza la propuesta del modelado de dominio del sistema. Se especifican los actores, las descripciones de los casos de uso, los requisitos funcionales y no funcionales que debe cumplir el sistema a desarrollar. Además, se muestran los diagramas de casos de uso, que permiten observar las relaciones de los actores que interactúan con el sistema, lo que conlleva a lograr que el trabajo sea guiado hacia una solución factible.

- **Capítulo 3 – Implementación y Pruebas:** En este capítulo se expondrán los artefactos fundamentales correspondientes al flujo de trabajo. Se define el Modelo de Datos, el Diagrama de Componentes y el de Despliegue, elementos que contribuyen a la construcción del sistema. Por último, se realizan pruebas al sistema para corroborar que satisface las necesidades del cliente.

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

### 1.1 Introducción

En el presente capítulo se enuncian los conceptos relacionados con el sistema que se desarrolla. Se caracterizan las herramientas, la metodología a utilizar en el proceso de desarrollo del software, así como las tecnologías utilizadas. Se muestra cuáles son las herramientas, tecnologías y tendencias actuales que se podrá encontrar en el mercado, útiles para el desarrollo del sistema.

### 1.2 Conceptos Asociados al Dominio del Problema

En esta búsqueda de información surgen conceptos y definiciones relacionadas con el objetivo, que dirige el proceso de desarrollo que se lleva a cabo. Para una mejor comprensión de la situación problemática se relacionan los principales conceptos asociados al desarrollo de la investigación.

**Sistemas informáticos:** no son más que el conjunto de elementos que trabajan hacia un fin, los elementos que se utilizan en el proceso son los de hardware y software. El software programa las computadoras para que realicen determinadas tareas o acciones y el usuario que es quien maneja y que a través del manejo del mismo logra que el hardware realice las tareas adecuadas.

Con frecuencia se tiende a confundir el concepto de sistema informático con el de sistema de información, ciertamente tanto uno como el otro, incluyen a las personas que acceden o producen información dentro del sistema. Las personas tienen que capacitarse para entender el funcionamiento y procedimientos que soporta el sistema pero un sistema informático utiliza computadoras para almacenar, procesar y/o acceder a la información y en el sistema de información no es necesario (Alegsa, 2009).

**Sistema de control:** es todo aquel con el que se supervisan los horarios de entrada y salida del personal de una entidad productiva o decente, con el fin de registrar: llegadas tardes, inasistencias, salidas anticipadas, entre otros, existen diversos tipos de control como son:

- **Sistemas de control mediante tarjetas:** estos sistemas son de los más utilizados pues los mismos se basan en la presencia de bandas magnéticas o código de barra, la información es recogida con un lector habilitado en el centro o empresa y en algunos casos en el puesto de trabajo.
- **Sistemas de control biométricos:** son sistemas muy sofisticados, en los mismos la probabilidad de fraude es mínima pues, son el resultado de la aplicación de técnicas matemáticas y estadísticas sobre los rasgos o conductas de una persona. Como por ejemplo: los patrones faciales, las huellas

dactilares, el iris y otros. Pero los mismos presentan un alto costo por las tecnologías que utilizan (Sage, 2012).

**Actividad de Producción:** se entiende como la actividad de diseñar, crear y estudiar: documentación, artefactos y sistemas informáticos con la mayor calidad posible, además la misma aumenta el desarrollo en el proceso de acomodo a las condiciones de trabajo externas luego de haber concluido los estudios universitarios facilitando una mejor inserción en las actividades laborales futuras (Msc. González, y otros, 2004).

## 1.3 Análisis de las Soluciones Existentes

Para la argumentación del epígrafe se investigó sobre los sistemas existentes a nivel nacional e internacional que se encargan del control y registro de asistencia a las actividades docentes o productivas. Los sistemas investigados serán analizados con el fin de obtener un mejor entendimiento del sistema y elementos que sirvan de base para el desarrollo de la futura aplicación. Así como para comprender su importancia y la no existencia de uno similar.

### 1.3.1 Sistemas Automatizados Utilizados a Nivel Mundial

#### **E-Time Software para Control de Tiempo y Asistencia** (Red de Software, 2009)

Calcula automáticamente las horas trabajadas de sus empleados, así como sus tiempos extras considerando horas de comida, ausentismo, vacaciones y retardos.

Características:

- Genera un archivo consolidado de nómina, preconfigurado para su exportación a cualquier programa de nómina.
- Totalmente en español.
- Puede ser modificado para necesidades específicas.

Es un sistema típico de Gestión de Tiempo y Asistencia, donde los usuarios del mismo registran sus entradas y salidas en terminales colectoras de datos (también llamados relojes electrónicos), usando tarjetas con códigos de barras, proximidad, banda magnética o dispositivos biométricos como lectores de huella digital y geometría de mano.

Cada entrada o salida de un usuario, genera un registro que contiene, entre otras:

- Fecha y hora en que se hizo el registro.

# Capítulo 1: Fundamentación Teórica

---

- Número de usuario o empleado.
- Número de la terminal colectora donde se hizo el registro.
- Información adicional, por ejemplo: el motivo de una salida (permiso sindical, permiso personal, etc.).

Estos registros son colectados a través de una red local, remota o vía internet, por una computadora ejecutando E-time, quien tiene previamente en su propia base de datos, la información de todos los empleados, como es el horario y turnos en que trabajan.

## **SICON Software Avanzado de Control de Asistencia** (Cashersoft, 2011)

Es un software avanzado con el que se lleva de una manera fácil y dinámica la gestión total de la asistencia del personal: el manejo de horas extras, tardanzas, permisos, compensaciones, vacaciones, licencias, bonificaciones, entre otros.

Características:

- **Acceso al sistema:** cada usuario ingresa al SICON mediante su contraseña, la cual le permite acceder únicamente a los módulos autorizados por el administrador del sistema.
- **Consulta de marcadores y asistencia:** consulta de asistencia de un empleado directamente en una sola pantalla, indicando la hora del mes actual o meses anteriores, desde aquí también se pueden realizar las justificaciones respectivas de un empleado.
- **Programación de vacaciones y saldo vacacional:** controla los días solicitados a cuenta de vacaciones para el año en curso. Vacacional indicando número de días disponibles de tomar.
- **Gestión de permisos y movimientos:** impresión de listados indicando múltiples criterios de selección. SICON permite llevar un control de los distintos movimientos y permisos del empleado, permitiendo toques máximos para cada uno de ellos en el año indicando si son con o sin goce.

## **Doce versión 2** (Paez, Luis, 2010)

Es un programa de control de asistencia, que incorpora un registro de datos del personal en una empresa o institución. El control de asistencia se efectúa usando el teclado de una computadora en el cual el personal registra su código individual de la manera en que se podría hacer con un reloj tarjetero. Permite utilizar una cámara web para que tome fotografías de la persona que marca su entrada y salida de manera que, posteriormente a través de una revisión manual sea posible identificar algún fraude cometido por personal que marcan a cuentas de otras.

Los reportes que el programa puede generar son:

- Varias nóminas.
- Historial.
- Reporte de atrasos y faltas.
- Tiempo de trabajo.
- Control de marcas en el reloj.
- Selección manual.
- Planilla mensual.
- Asistencia de personal por día.
- Cuadro mensual exportado a Excel.

## **SmartCik Control de Asistencia** (SecureTech, 2010)

Es un programa diseñado para asistir en la gestión de control horario y de asistencia de personal que permite evitar errores de cálculo producidos en el control de asistencia por métodos no informatizados de forma fácil, eficaz y eficiente. Además de la incorporación de registros horarios de cualquier reloj electrónico en sus bases de datos ayudando al operador en la depuración de las marcas de asistencia y el control de los horarios mediante la incorporación de herramientas efectivas y el ingreso de registros de justificación de falta por distintos conceptos. Los mismos pueden crearse en el sistema especificando el tipo de concepto para la justificación de falta. Ejemplos de estos son licencia, falta de aviso, etc.

Características y funciones del sistema:

- **Seguridad:** incorpora elementos de seguridad que permiten el registro de los operadores y todas las operaciones realizadas en el programa. Esta información se almacena en un registro de auditoría el cual puede analizarse posteriormente a los efectos de controlar la buena gestión de los datos. El ingreso a la aplicación es mediante usuario y contraseña.
- **Horarios:** permite el control de horarios fijos y rotativos fijos, de los cuales se debe especificar la hora de entrada y salida, los días o periodos que se debe controlar, horarios libres, los cuales no se especifica más que una carga horaria y los días que debe cumplir esa carga horaria, horario presencia donde el sistema controla que esas personas por lo menos cuente con una marca indicando su presencia.

# Capítulo 1: Fundamentación Teórica

---

- **Informes:** posibilita emitir informes de inconsistencia, inasistencias, flash de asistencia, diferencias de horario (entradas tardes, salidas anticipadas), cálculo de horas trabajadas y planilla resumen. Los informes permiten gran flexibilidad para la selección, filtros de registros (filas y columnas), impresión y vista previa de impresión.
- **Carga de datos externos:** proporciona la carga de los datos iniciales del sistema a partir de archivos en formato de texto externos. Adicionalmente puede leer los registros de asistencia de archivos de texto generados por cualquier reloj de control.
- **Interfaz con otros sistemas:** incluye una salida de datos en formato de texto con cuatro tablas: registro de horarios, inconsistencias, inasistencias y horas. Esta interfaz se utiliza para entregar la información de horas trabajadas a un sistema de liquidación de haberes.
- **Asignación de marcas:** permite desdoblarse un código de marca creado en un reloj de control, en dos códigos correspondientes a entrada y salida. Esta asignación se realiza teniendo en cuenta el tipo de horario asignado al funcionario y la secuencia de marcas que el mismo ha generado.

## Aplicación Web del Control Horario, Universidad de Zaragoza (Torres, 2010)

Esta aplicación se emplea en la Universidad de Zaragoza, España para lograr un adecuado cumplimiento de la normativa en materia de calendario laboral, jornada y horario de los trabajadores del centro.

Se utiliza el término fichaje como sinónimo de marcaje, ambos términos definen la incidencia de una persona en una fecha y hora determinada, realizada desde un lector o desde la propia aplicación. El personal de administración y servicios, en general, puede realizar fichajes mediante un ordenador personal, desde su puesto de trabajo, ver sus fichas y visualizar su cuadro mensual de bloques de trabajo una vez revisado por el personal de su unidad.

Los responsables de las unidades se encargarán de la supervisión del cumplimiento de la jornada y horario del personal a su cargo, así como del control de las diferentes causas de inasistencia al trabajo.

Existen dos tipos fundamentales de incidencias:

- **Incidencias dentro de la jornada laboral:** el fichaje de las entradas y salidas y demás incidencias es responsabilidad de cada personal y pueden ser realizadas desde el lector o desde la web.
- **Ausencias planificadas:** estas incidencias pueden corresponder, en algunos casos, a días completos y el responsable es el jefe de unidad quien debe introducir este tipo de incidencia en la aplicación.

# Capítulo 1: Fundamentación Teórica

---

Para el desarrollo de esta aplicación se emplearon tecnologías basadas en la web, con una interface y navegación homogénea con el resto de las aplicaciones que viene desarrollando el Servicio de Informática, se deberá contar con un navegador web para poder utilizarla.

Entre las principales funcionalidades que brinda la aplicación se encuentran:

- **Fichar hora:** esta opción permite fichar desde cada ordenador seleccionado el código correspondiente a la incidencia en un menú desplegable. La aplicación informática identifica la maquina desde la que se realiza la operación de marcaje.
- **Ver mis fichajes:** desde esta opción se consultan los fichajes realizados por cada usuario entre dos fechas determinadas. La modificación de los fichajes es función de los responsables de la unidad. Se podrán ver los fichajes realizados entre las fechas seleccionadas indicando: el día de la semana, la fecha, la hora, la duración de la jornada de ese día si es una ausencia planificada, el tipo de marcaje mediante un icono (lector o web), la procedencia (una etiqueta que identifica el lector o la dirección IP del ordenador donde se efectuó el fichaje).
- **Cuadros mensuales:** desde esta opción se consultan los horarios de entradas, salidas e incidencias en un mes determinado para cada usuario. No estará disponible si existe algún error. Se indica: día de la semana, fecha, hora de entrada y salida, el número de horas presencial real (Hp), el número de horas de incidencias (Hi), el número de horas de ausencias por motivos particulares (Haap), el número de horas teóricas justificadas ( $H_t = H_p + H_i$ ), los códigos de incidencias usados en el bloque (CI), las horas trabajadas el sábado (Hs), las horas extraordinarias (He), las horas en exceso de jornadas irregulares (Hx) y las horas de recuperación de ausencias particulares (Hrap).

## 1.3.2 Sistemas Automatizados Utilizados a Nivel Nacional

### **Sistema de Control de Asistencia de Personal del Instituto de Suelos** (Pinta, y otros, 2013)

Es una aplicación desarrollada en la Universidad de Granma, para el control de asistencia del personal mediante una aplicación web que permita un mejor manejo de la información para facilitar la gestión de la información concerniente a la asistencia del personal.

El sistema cuenta con las siguientes características:

- Registro de la jornada laboral.
- Permite mostrar la prenómina.

# Capítulo 1: Fundamentación Teórica

---

## **Sistema de Control de Asistencia del Personal Docente del Centro de Educación Básica “Dr. Néstor Mogollón López”** (Ulloa, y otros, 2011)

Es una aplicación desarrollada en la Universidad de Granma, para el control de asistencia del personal mediante el uso de un controlador biométrico, así como el cumplimiento de la jornada laboral y el horario correspondiente a cada docente. Es una aplicación web fácil de usar en cualquier plataforma y es multiusuario.

El sistema cuenta con las siguientes características:

- Gestión del personal docente.
- Horario de del personal, así como las asistencias.
- Registro de incidencias del personal.
- Inclusión de un módulo de salvos para realizar resguardos de la base de datos.

## **Sistema de Recursos Humanos (Fastos)** (Torres, 2010)

El sistema Fatos, está formado por los módulos Configuración, Personal, Capacitación y Cuadros, que permite controlar las informaciones fundamentales de los empleados de una entidad, también realizar carios procesos y operaciones que son inherentes al área de recursos humanos, tales como:

- Registro de los empleados: se guardan los datos de los empleados, así como informaciones referentes a los reportes de vacaciones, certificados médicos, licencias, resolución.
- Control de plantilla: permite establecer la estructura organizativa de las plazas de la entidad.
- Control de asistencia: incorpora el control de claves de asistencias, turnos de trabajos, horarios, tarjetas de asistencia de cada empleado. Además, posibilita acoplar relojes (RT 600) para actualizar la información de la tarjeta de asistencia de forma automática.
- Informes y modelos: proporcionan la obtención de un total de 56 informes, por ejemplo: cierre del periodo, análisis de fondo de tiempo, estadísticos, entre otros y 11 modelos entre los que se encuentran: el acta de entrega del expediente laboral, contratos de trabajo, guía de documentos archivados, convenio de trabajadores, etc.
- Control de capacitación respecto a: acciones de capacitación, estudios realizados, cursos, eventos, experiencia docente, publicaciones, conocimientos, idiomas extranjeros, plan de desarrollo, informes y otros aspectos.

# Capítulo 1: Fundamentación Teórica

---

- Control de la información de los cuadros: se establece el registro de los cuadros, dirigentes y reserva, referente a evaluaciones, inspecciones, sanciones, necesidades de capacitación, entre otros.

El sistema tiene las siguientes características generales:

- Aplicación servidor, con base de datos.
- Ayuda incorporada.
- Control y registro de acceso al sistema.
- Facilidades para la exportación de información.

## **GREHU, sistema integral para gestión de Recursos Humanos** (Torres, 2010)

Este software constituye el resultado de un profundo estudio realizado en la temática de Gestión de los Recursos Humanos, obteniéndose una herramienta de software que permite registrar, procesar y gestionar de forma integrada las principales funciones que se desarrollan en la dirección de recursos humanos de una entidad laboral entre ellas la del control de la asistencia en trabajadores.

El software desarrollado permite realizar de forma automatizada e integrada las principales funciones tales como: el inventario de personal, el control de las sanciones y amonestaciones, la selección y contratación, la evaluación del desempleo, el trabajo con los dirigentes y cuadros y el procesamiento de las nóminas.

Este sistema con sus innumerables salidas tanto estadísticas como gráficas hace posible que, en mayor o menor medida, los directivos relacionados con la gestión de personal, puedan conocer y prever las posibles promociones, necesidades de formación y capacitación, los reclutamientos futuros, el comportamiento de la disciplina laboral, el desempeño del personal, así como el desarrollo y control efectivo de la asistencia a la empresa.

Después de haber analizado las soluciones existentes en el ámbito internacional y nacional se llegó a la conclusión que:

- Estos sistemas son a la medida, y su reusabilidad está limitada por las especificaciones propias de su entidad.
- Es poco factible adquirir un sistema ya desarrollado para otro esquema, pues habría que personalizarlo totalmente a las características del centro.
- Muchos son privativos y no cumplen con la política de software libre de la universidad.

Debido a lo antes expuesto y la importancia que tiene toda la información que se almacena y controla en el departamento y la no existencia de un sistema capaz de gestionar dicha información, se decide desarrollar un sistema web para la consulta y gestión de horario, PC, estudiantes y trabajadores vinculados a las actividades de la producción para lograr una mejor centralización y control de los de los mismos.

## 1.4 Tecnologías a Utilizar

Teniendo en cuenta las necesidades planteadas y las características del entorno donde se aplicara la solución propuesta, se realizó un estudio de las tendencias y tecnologías actuales posibles a emplear, que son las especificadas y analizadas en este epígrafe.

### 1.4.1 Metodología de Desarrollo

Se define como metodología al conjunto de estrategias, procedimientos, métodos o actividades intencionadas, organizadas, secuenciadas e integradas, que permitan el logro de aprendizajes significativos y de calidad (Conceptode, 2014).

Las metodologías de desarrollo del software pretenden guiar a los de son desarrolladores a crear un nuevo software, dado que los requisitos son muy variados y cambiantes de un sistema a otro. Se da lugar a que exista una gran variedad de metodologías para la creación de software.

### Proceso Unificado Rational (por sus siglas en ingles RUP)

Es una infraestructura flexible de desarrollo de software que proporciona prácticas recomendadas probadas y una arquitectura configurable, un proceso practico que mantiene al equipo enfocado en producir incrementalmente software operativo a tiempo, con las características y calidad requeridas (GSINNOVA, 2010).

RUP es un intento encaminado a reunir los mejores rasgos y características de modelos de procesos de software, presenta res características de gran importancia:

- Guiado por casos de uso: pues reconoce la importancia de la comunicación con el cliente y los métodos encaminados a describir el punto de vista del cliente con respecto a un sistema.
- Centrado en la arquitectura: pues ayuda al arquitecto a enfocarse en las metas correctas, como el rendimiento, el ajuste a los cambios futuros y la reutilización.
- Iterativo e incremental: pues sugiere un flujo de proceso iterativo e incremental y proporciona el sentido esencial en el desarrollo del software moderno (Pressman, 2007).

# Capítulo 1: Fundamentación Teórica

---

RUP es un modelo que identifica cuatro fases diferentes en el proceso del software (inicio, elaboración, construcción y transición), dentro de estas fases se llevan a cabo un conjunto de actividades que se denominan flujo de trabajo. Existen seis principales flujos de trabajo identificados en el proceso y tres flujos de trabajo de soporte.

## **1.4.2 Lenguaje Unificado de Modelado (por sus siglas en ingles UML)**

Es importante seleccionar, además de la metodología de software a utilizar para el desarrollo del software, algún elemento que permita describir el aspecto y la conducta del producto, estos elementos se denominan lenguajes de modelado.

UML es un lenguaje para construir modelos. Es uno de los lenguajes de modelado de gran utilidad para el desarrollo, pues permite visualizar, especificar, construir y documentar los artefactos de un sistema. Su objetivo es lograr una aplicación de software robusta, flexible y escalable (OMG, 2015). La versión que se utilizará en esta investigación es la 2.5.

En cada fase del proceso de desarrollo del software se crean un conjunto de artefactos que constituyen una entrada fundamental a las fases que le suceden. Es utilizado UML en la realización del módulo pues la metodología RUP se basa en ese lenguaje de modelado para la creación de sus artefactos. Los modelos creados sirven como documentación técnica del módulo para obtener un mejor entendimiento del mismo.

## **1.4.3 Lenguajes de Programación**

Un lenguaje de programación está conformado por una serie de reglas sintéticas y semánticas que son utilizadas por el programador a través de las cuales se creara un programa o subprograma. Todas estas instrucciones escritas por el desarrollador y que forman dicho programa es a lo que se le denomina código fuente (Alegsa, 2010).

Los lenguajes de programación web se pueden clasificar en lenguajes del lado del servidor y del lado del cliente. La diferencia básica entre estos dos es que: el lenguaje del lado del cliente se ejecuta en su mismo navegador web y en cambio, el lenguaje del lado del servidor, ejecuta las aplicaciones sin tener en cuenta el navegador web del cliente (Alvarez, 2012).

## **Lenguaje de marcas de hipertexto (por sus siglas en ingles HTML)**

HTML 5 es un lenguaje utilizado para escribir y publicar documentos en la *World Wide WEB*. De esta manera informal es el lenguaje con el que se escriben las páginas web en Internet. HTML más que un

# Capítulo 1: Fundamentación Teórica

---

lenguaje se ha convertido en un estándar con el que se escriben, aceptado en todo el mundo por lo que una página HTML se puede visualizar en el navegador de cualquier Sistema Operativo (Victoria, 2009).

## **JavaScript**

Es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. Se utiliza principalmente para crear web dinámica (Eguiluz, 2010).

El código JavaScript se ejecuta del lado del cliente por lo que no se le hacen peticiones innecesarias al servidor. Además, es conveniente el uso de Java Script para la validación de los datos de los formularios del lado del cliente y para la creación de interfaces de usuario más complejas e iterativas.

## **Hojas de estilo cascada (Por sus siglas en ingles CSS)**

CSS son un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla. Se utilizan para dar estilos a documentos HTML, separando el contenido de la presentación. Los estilos definen la forma de mostrar los elementos HTML. El principal objetivo del uso de CSS es separar la estructura del documento de su presentación y de esta forma aumentar la organización del código y el riesgo de pérdida de uno u otro, además le ofrece a los desarrolladores el control total sobre el estilo y formato de sus documentos (Eguiluz, 2010). Con la utilización de CSS se pueden aplicar los diferentes estilos al sistema.

## **Procesador de hipertexto (por sus siglas en ingles PHP) v.5.5.9**

Es un lenguaje de código abierto especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. Lo que distingue a PHP del lenguaje del lado del cliente como JavaScript, es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El principal objetivo del lenguaje es permitir a los desarrolladores web codificar rápidamente páginas que se generan dinámicamente. Es un lenguaje multiplataforma que soporta gran cantidad de gestores de Base de Datos (BD), es soportado por los servidores web más conocidos y cuenta con una comunidad que se dedica al continuo desarrollo y fortalecimiento del mismo (Vázquez Mariño, 2008).

Esta versión se utilizó por las ventajas que trae consigo sobre las anteriores versiones pues:

- Los bloques try-catch ahora admiten un bloque finally para el código que debería ejecutarse pese a si se ha lanzado o no una excepción o no.
- No solo acepta variables sino que también se pueden pasar funciones como argumentos.

- Los *arrays* y los literales de tipo *string* ahora se pueden derreferenciar directamente para acceder a los elementos y características individuales (PHP, 2014).

## **1.4.4 Sistema Gestor de Base de Datos (SGBD)**

Un SGBD es un conjunto de programas que permiten crear y mantener una base de datos asegurando su confidencialidad, integridad y seguridad. Entre las características deseables de un SGBD se encuentran el control de redundancia, la restricción de los accesos y el cumplimiento de las restricciones de integridad (Avila, Katty, 2014).

### **PostgreSQL v.9.4**

Es un SBD objeto-relacional, distribuido bajo la licencia de software libre que permite el uso de código fuente en software no libre. PostgreSQL utiliza un modelo cliente-servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. En esta versión veremos características que mejoran la flexibilidad, escalabilidad y rendimiento.

Las principales ventajas que presenta son:

- Mejoras en los índices de ginebra haciéndolos hasta un 50% más pequeños y hasta 3 veces más rápido.
- “JSONB trae PostgreSQL para la comunidad de desarrollo al permitir JavaScript datos JSON para almacenar y consultar de forma nativa. Node.js y otra del lado del servidor JavaScript marcos pueden beneficiarse de la seguridad y solidez de PostgreSQL, sin dejar de almacenar los datos en el formato sin esquema que prefieren” (Postgresql, Sitio Oficial, 2014).

## **1.4.5 Plataformas, Herramientas de desarrollo y Marco de trabajo**

### **Servidor Apache v.2.4.7**

Apache es el servidor web que provee servicios del Protocolo de Transferencia de Hipertexto (por sus siglas en ingles HTTP). Al ser altamente configurable, robusto y estable hacen que cada vez millones de usuarios reiteren su confianza en este programa. Es utilizado pues puede ejecutarse en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal, es una tecnología gratuita de código fuente abierto, es altamente configurable, trabaja con gran cantidad de lenguajes y permite personalizar la respuesta ante los posibles errores que puedan dar en el servidor (Ciberaula, 2011).

### **JavaScript asíncrono y XML (por sus siglas en ingles AJAX)**

# Capítulo 1: Fundamentación Teórica

Es una técnica de desarrollo web para lograr aplicaciones interactivas mediante la combinación de tecnologías ya existentes como:

- HTML (o XHTML) y CSS para presentar la información.
- DOM para interactuar dinámicamente con la presentación.
- XML, Transformaciones XSL (por sus siglas en inglés XSLT) y Formato Ligero para el Intercambio de Datos (por sus siglas en inglés JSON) para intercambiar y manipular datos de manera asíncrona con un servidor web.
- JavaScript para unir todas las tecnologías anteriores (Eguiluz, 2010).

AJAX es soportado por la mayoría de los navegadores modernos, presenta:

- Interactividad: el usuario no tiene que esperar hasta que lleguen los datos al servidor.
- Mayor velocidad: no ahí que retornar toda la página nuevamente y la página se asemeja a una aplicación de escritorio.

## Websockets

Es una tecnología para resolver los problemas de comunicación que plantean las técnicas asíncronas, permite comunicar el cliente y el servidor a través de un canal Full Duplex bidireccional y sin tener que hacer pedidos por parte del cliente. En esencia cuando un socket está abierto o escuchando los datos son enviados de forma directa al servidor permitiendo esto que, los mismos lleguen con mayor inmediatez del servidor al navegador (Garro, 2014).

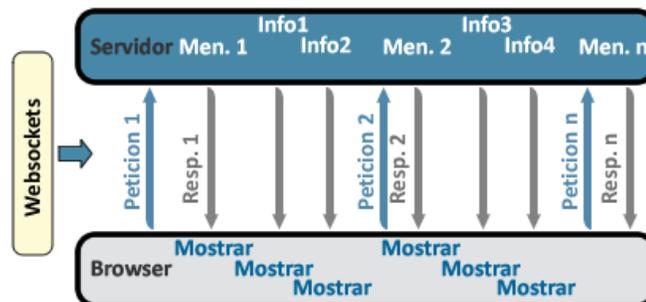


Fig. 1 Esquema de Websockets

Entre sus ventajas esta:

# Capítulo 1: Fundamentación Teórica

---

- Intercambio de información entre el cliente y el servidor cuando cualquiera de las dos partes lo deseen sin necesidad de pedir información al servidor.
- Ambas partes pueden enviar información al mismo tiempo por la conexión Full Duplex ya que los mensajes no chocaran.
- Su tecnología se extiende por la mayoría de los servidores web más conocidos como Apache, Jetty, etc. (Melnikov, A., 2011)

## Visual Paradigm v.8.0

Es una herramienta de Ingeniería de Software Asistida por Computador (por sus siglas en ingles CASE) que utiliza UML como lenguaje de modelado. La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y diseño, hasta la generación del código fuente de los programas y la documentación. Visual Paradigm ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Constituye una herramienta de software libre de probada utilidad para el análisis. Fue diseñada para una amplia gama de usuarios interesados en la construcción de un sistema de software de forma fiable a través de la utilización de un enfoque Orientado a Objetos (OO) (Menéndez, Evelyn Alonso, 2009).

## NetBeans v.7.3 como entorno de desarrollo para PHP

NetBeans es un proyecto exitoso de código abierto con una gran base de usuarios y una comunidad en constante crecimiento. NetBeans IDE es un entorno de desarrollo que contiene un número importante de módulos para extenderlo. Es un producto libre y gratuito sin restricciones de uso (Netbeans, Sitio Oficial, 2011). Es utilizado pues es multiplataforma, libre, brinda la posibilidad de desarrollar aplicaciones web utilizando PHP y ofrece soporte para Symfony.

## Symfony v.2.8 como marco de trabajo

En el desarrollo de software, un marco de trabajo simplifica el desarrollo de una aplicación mediante la automatización de algunos patrones utilizados para resolver las tareas comunes. Proporciona además, estructura al código fuente, forzando al desarrollador a crear códigos más legibles y fácil de mantener. Por último, un marco de trabajo facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas (Rodríguez, Juanda, 2011).

El uso de esta versión fue de vital importancia, pues en la misma todo es un bundle (concepto similar al de los plugins en otras aplicaciones), permitiendo la utilización de funcionalidades que fueron realizadas por

# *Capítulo 1: Fundamentación Teórica*

---

terceros. Además contiene bundles que son de vital importancia para el desarrollo del sistema pues incluyen PHP, Javascript, Websockets y otros (Librosweb, 2014).

## **1.5 Conclusiones**

Durante este capítulo se trataron los aspectos significativos de la investigación que serán el soporte teórico de la futura solución y demuestran la necesidad de la realización del presente trabajo. El estudio del arte permitió analizar las soluciones existentes en el ámbito nacional e internacional. Se realizó un análisis de las tecnologías a utilizar, herramientas y metodología, lo que aportó una serie de conocimientos esenciales para la evolución de la investigación. Luego de analizar la situación del departamento se pudo comprender, que la necesidad fundamental de desarrollar un sistema para el control y asistencia a las actividades de producción es que, no existe un mecanismo eficiente de control y asistencia en el centro.

## CAPÍTULO 2: ANÁLISIS Y DISEÑO DE LA SOLUCIÓN PROPUESTA

### 2.1 Introducción

En el presente capítulo se realiza la propuesta del modelado de dominio del sistema. Se especifican los actores, las descripciones de los casos de uso, los requisitos funcionales y no funcionales que debe cumplir el sistema a desarrollar. Además, se muestran los diagramas de casos de uso, que permiten observar las relaciones de los actores que interactúan con el sistema, lo que conlleva a lograr que el trabajo sea guiado hacia una solución factible.

### 2.2 Propuesta de Solución

La solución propuesta en la presente investigación consiste en el diseño e implementación del Sistema de Control y Asistencia a Actividades de la Producción, que permita mejorar el control de la información referente a la asistencia a la producción. Para cumplir con el objetivo trazado, el sistema brinda la posibilidad de definir las sesiones de producción del Departamento y de establecer un horario de producción semanal para cada estudiante y trabajador.

Entre las ventajas que posee el sistema se encuentran:

- Es multiplataforma, por lo que puede ser utilizado desde cualquier sistema operativo.
- Necesita requerimientos mínimos de hardware para su funcionamiento, memoria RAM de 1024 MB, almacenamiento interno de 100 GB por parte del servidor y 512 MB de RAM por parte del cliente.
- Cada usuario puede registrar la asistencia de forma automática desde su computadora de trabajo (sin necesidad de dispositivos colectores de datos).
- Cada usuario sólo puede marcar la asistencia desde una sola computadora en una misma sesión de trabajo. Lo que posibilita tener mayor control de los mismos pues solo estarán conectados desde la estación de trabajo que tendrán asignada por el sistema.

Cada entrada o salida de un usuario genera un reporte que contiene:

- Fecha.
- Llegada tarde.
- Inasistencia.

## Capítulo 2: Análisis y diseño de la solución propuesta

---

- Usuario.
- Nombre de usuario y apellidos.
- Nombre de la computadora.
- Dirección IP de la computadora.

Además, el sistema brinda la posibilidad de exportar los reportes en formato pdf, enviar el horario en formato pdf a estudiantes y profesionales que contiene las sesiones, nombre de la computadora y dirección de IP donde tienen que producir una vez adicionado o modificado su horario, todo esto de forma automática por el sistema con un correo electrónico, además del envío en formato pdf del reporte de la semana.

### 2.3 Modelo de Dominio

El modelo de dominio puede utilizarse para capturar y expresar el entendimiento ganado en un área bajo análisis como paso previo al diseño de un sistema, ya sea de software o de otro tipo. Similares a los mapas mentales utilizados en el aprendizaje (Larman, 2003). Es un modo que facilitará mostrar de manera visual al cliente o programador los principales conceptos que se manejan en el modelo de dominio del sistema a desarrollar.

A continuación, aparecen varios conceptos asociados que serán de utilidad para entender la lógica del modelado.

**Departamento de PP:** departamento dentro del centro GEYSED que organiza y atiende las actividades asistencia de estudiantes y profesores a la producción.

**Jefe de Departamento:** máxima autoridad en el departamento.

**Profesores:** profesores del centro vinculados a la producción y que tutorean a los estudiantes.

**Estudiantes:** estudiantes de 3ro, 4to y 5to vinculados a la producción en el centro.

**Horario:** horario de producción de los estudiantes y profesores del departamento, está dividido entre los días de la semana y las secciones de trabajo.

**Jefe de Proyecto:** profesor que se encarga de llevar el registro de la asistencia de los estudiantes y profesores a su cargo.

**Registro asistencia:** datos obtenidos en la jornada de producción como: fecha, hora de entrada y salida y nombre del personal.

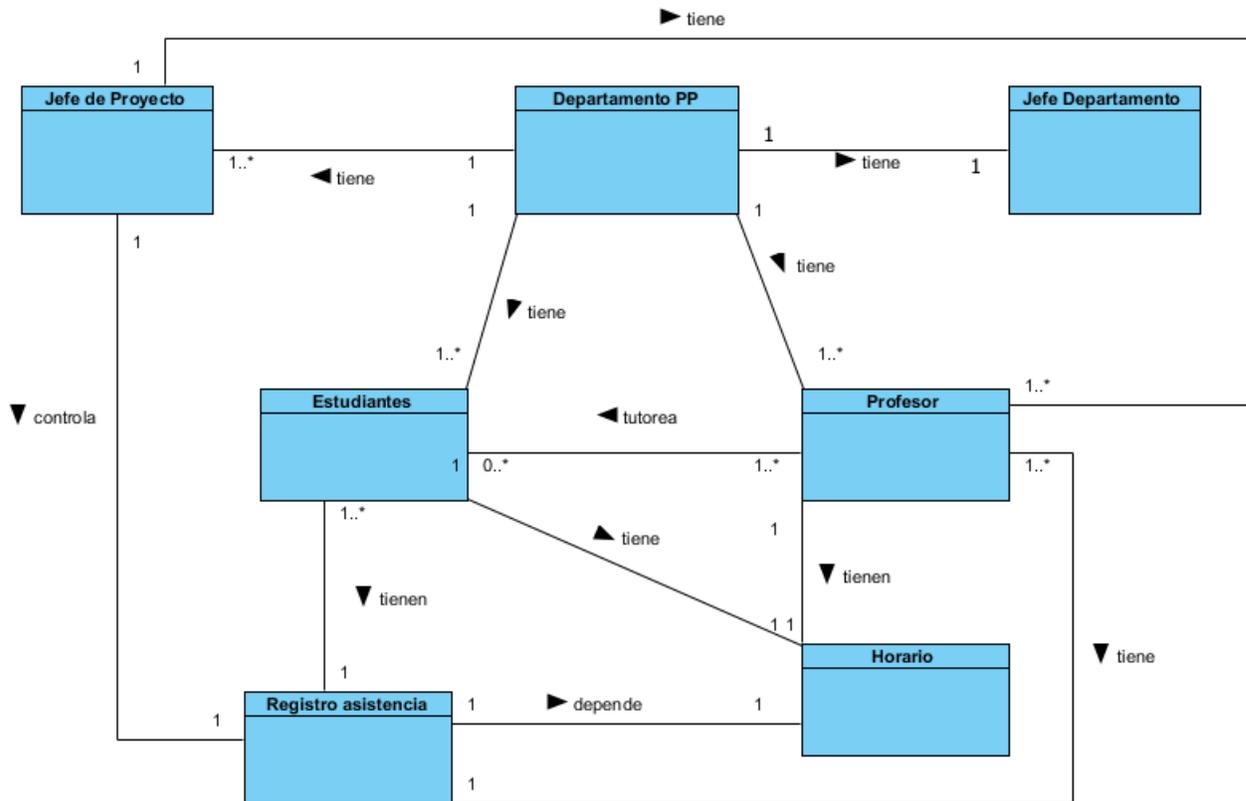


Fig. 2 Modelo de Dominio

## 2.4 Requisitos Funcionales

Los requisitos funcionales son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y cómo comportarse en situaciones particulares (Somerville, 2005). Constituyen una base para identificar lo que realmente debe hacer el sistema. Deben ser comprendidos tanto por los desarrolladores como los clientes.

A continuación, se especifican los requisitos funcionales que debe cumplir el sistema.

- **RF1 Adicionar usuario:** Esta funcionalidad permite insertar un usuario en el sistema para que luego pueda autenticarse en el mismo usando su usuario y contraseña UCI.
- **RF2 Modificar usuario:** Esta funcionalidad permite editar los datos de un usuario.
- **RF3 Eliminar usuario:** Esta funcionalidad permite eliminar un usuario.
- **RF4 Listar usuario:** Esta funcionalidad muestra un listado con todos los usuarios del sistema, de los mismos se listan su nombre y apellido, usuario, la dirección de IP de su computadora.

## Capítulo 2: Análisis y diseño de la solución propuesta

---

- **RF5 Buscar usuario:** Esta funcionalidad permite buscar al usuario que contenga alguno de sus datos en la cadena insertada, ya sea su nombre o apellidos, usuario o IP de su computadora de trabajo.
- **RF6 Autenticarse:** Esta funcionalidad permite el acceso al sistema de los usuarios, el mismo será usando el usuario y la contraseña UCI, luego de que haya sido insertado en la base de datos del sistema. Para la misma se tendrá en cuenta también la dirección de IP de la PC donde se autenticara el mismo.
- **RF7 Adicionar PC:** Esta funcionalidad permite insertar los datos de una PC, como su nombre y la dirección de IP para después poder verificar si el usuario se autenticó desde su PC.
- **RF8 Modificar PC:** Esta funcionalidad permite editar los datos de una PC.
- **RF9 Eliminar PC:** Esta funcionalidad permite eliminar una PC.
- **RF10 Listar PC:** Esta funcionalidad permite ver en un listado los datos de las PC del sistema.
- **RF11 Buscar PC:** Esta funcionalidad permite buscar la PC que contenga alguno de sus datos en la cadena insertada.
- **RF12 Crear horario:** Esta funcionalidad permite asignar el horario del usuario.
- **RF13 Modificar horario:** Esta funcionalidad permite editar el horario del usuario.
- **RF14 Eliminar horario:** Esta funcionalidad permite eliminar el horario del usuario.
- **RF15 Visualizar reporte:** Esta funcionalidad permite visualizar el reporte de llegada tarde o inasistencia, cada usuario solo podrá visualizar el reporte según el rol que tenga en el sistema.
- **RF16 Adicionar permiso:** Esta funcionalidad permite crear permisos o roles.
- **RF17 Exportar reporte a PDF:** Esta funcionalidad permite exportar a PDF el reporte, donde cada usuario solo verá el reporte según el rol que tenga en el sistema.
- **RF18 Enviar horario:** Cuando se le adiciona a un usuario su horario o se modifica, el sistema le envía el horario en formato pdf a su correo.
- **RF19 Notificación de usuario conectado:** Cuando un usuario se conecta, al resto de los usuarios de su grupo de trabajo les llega una notificación en el monitor que, el usuario se ha conectado.
- **RF20 Enviar reporte semanal:** El sistema envía los viernes el reporte semanal de asistencia a la producción.

# Capítulo 2: Análisis y diseño de la solución propuesta

---

## 2.5 Requisitos no Funcionales

Los requisitos no funcionales son restricciones de los servicios o funciones ofrecidos por el sistema. Son aquellos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de éste como fiabilidad, tiempo de respuesta y la capacidad de almacenamiento (Somerville, 2005).

A continuación, se especifican los requisitos no funcionales que debe cumplir el sistema.

### Requisitos de usabilidad

- La información estará disponible en todo momento.

### Requisito de confiabilidad

- El acceso al sistema será a través del usuario y contraseña del dominio.
- Cada usuario debe tener los permisos necesarios para realizar las operaciones que le sean permitidas en el sistema.

### Requisitos de Diseño

- El producto final debe diseñarse sobre una arquitectura modelo-vista-controlador.
- El sistema debe tener una apariencia profesional y un diseño sencillo.

### Requisitos de interfaz

- El sistema debe tener indicadores que permitan conocer al usuario las acciones que debe realizar.
- El sistema debe permitir al usuario pasar de una tarea a otra sin la necesidad de dar más de 3 clics.

### Requisitos de software

#### PC Servidor

- Instalación del servidor web Apache v.2.4.7.
- Instalación del SBD PostgreSQL v.9.4.
- Instalación de PHP v.5.5.9.
- Sistemas operativos: distribuciones de GNU/Linux.

#### PC Cliente

## Capítulo 2: Análisis y diseño de la solución propuesta

- Navegador web con soporte de HTML. Recomendable Firefox 20 o superior.
- Sistemas operativos: Microsoft Windows 7 o superior y distribuciones de GNU/Linux.

### Requisitos de hardware

#### PC Servidor

- Memoria RAM: 1024 o superior.
- Almacenamiento interno: 20 GB disponibles.
- Velocidad de procesamiento del microprocesador 1 GHz o superior.

#### PC Cliente

- Memoria RAM: 512 o superior.
- Velocidad de procesamiento del microprocesador 1GHz o superior.

### 2.6 Modelo de Casos de Uso

Los componentes primarios de un modelo de casos de uso son los casos de uso, los actores y el sistema modelado. Un caso de uso es una técnica de modelado usada para describir lo que debería hacer un sistema nuevo o lo que hace un sistema que ya existe. En el modelado de casos de uso, el sistema se observa como una caja negra que proporciona casos de uso. Cómo lo haga el sistema, cómo se implementen los casos de uso y cómo trabajen internamente no importa (Mediavilla, 2014).

#### 2.6.1 Descripción de los Actores

Después de haberse definido los requisitos funcionales y no funcionales que el sistema debe cumplir, corresponde definir los actores que intervendrán en el sistema.

**Actores del sistema:** son los trabajadores del negocio que poseen actividades a automatizar y pueden ser algún actor del negocio siempre y cuando interactúe con el sistema (Rumbaugh, 2000).

Tabla 1 Actores del sistema

Actor	Descripción
Administrador	Usuario que accede al sistema para visualizar el reporte de todos los profesores y estudiantes, consultar y gestionar el horario de todos los profesores y estudiantes en el sistema, encargado de asignar los permisos a los usuarios.

## Capítulo 2: Análisis y diseño de la solución propuesta

Trabajador	Usuario que accede al sistema para marcar su asistencia, visualizar su reporte y el de sus estudiantes asignados, consultar su horario y el de sus estudiantes asignados, además de gestionar el horario de sus estudiantes.
Usuario	Usuario que accede al sistema para registrar su asistencia, visualizar su reporte o su horario de producción.

### 2.6.2 Diagrama de Caso de Uso del Sistema (DCUS)

Un Caso de Uso (CUS) es un fragmento de una funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores. Estos se forman agrupando requisitos funcionales y especifican las acciones que tienen lugar durante la interacción actor-sistema.

Un DCUS representa gráficamente a los procesos y sus interacciones con los actores. En la fig. 3 se representa el DCUS del Sistema de Control y Asistencia a Actividades de la Producción (SCAAP).

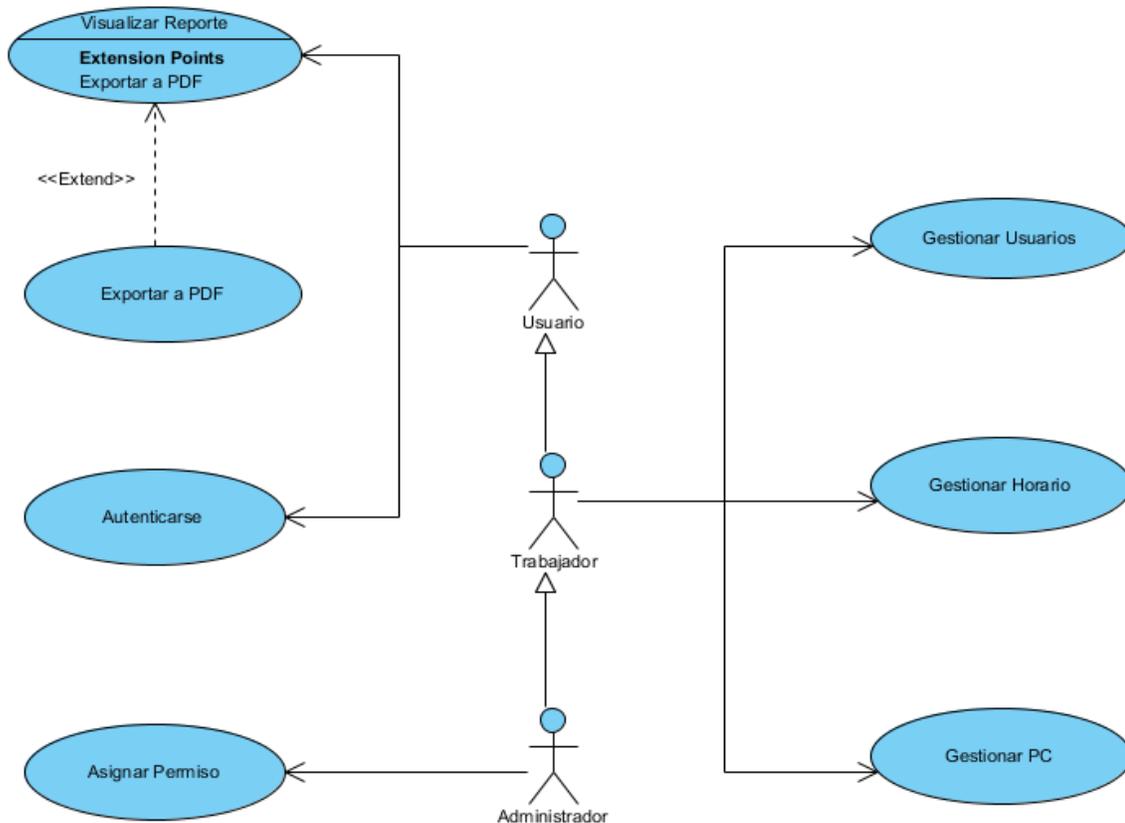


Fig. 3 Diagrama de Caso de Uso del Sistema

## Capítulo 2: Análisis y diseño de la solución propuesta

### 2.6.3 Descripción Textual de los Caso de Uso del Sistema

#### CUS Gestionar PC

Tabla 2 CUS Gestionar PC

<b>Caso de Uso</b>	Gestionar PC	
<b>Actores</b>	Trabajador	
<b>Resumen</b>	El caso de uso inicia cuando el actor accede a la opción Lista de Computadoras o Nueva Computadora en el menú.	
<b>Precondición</b>	Que el usuario este autenticado.	
<b>Referencias</b>	RF7, RF8, RF9, RF10, RF11	
<b>Prioridad</b>	Critico	
<b>Flujo Normal de Eventos</b>		
<b>Flujo básico "Gestionar PC"</b>		
	<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1.	Accede a la opción Listar Computadoras del menú Computadoras	
2.		Muestra una página que contiene la lista de las PC con los siguientes datos: Nombre y dirección de IP y las acciones de Mostrar (ir a la sección Mostrar PC) y Eliminar (ir a la sección Eliminar PC) y Buscar (ir a la sección Buscar PC). (Ver interfaz 1).
<b>Interfaz 1</b>		

# Capítulo 2: Análisis y diseño de la solución propuesta

Administración de computadoras

Computadoras

10 datos por página

Buscar:

#	Nombre	Ip	Acciones
14	LAB-1 PC-1	10.34.34.1	<ul style="list-style-type: none"> <li>Mostrar</li> <li>Eliminar</li> </ul>
15	LAB-1 PC-2	10.34.34.2	<ul style="list-style-type: none"> <li>Mostrar</li> <li>Eliminar</li> </ul>
17	LAB-2 PC-1	10.34.35.1	<ul style="list-style-type: none"> <li>Mostrar</li> <li>Eliminar</li> </ul>
18	LAB-2 PC-2	10.34.35.2	<ul style="list-style-type: none"> <li>Mostrar</li> <li>Eliminar</li> </ul>
19	LAB-2 PC-3	10.34.35.3	<ul style="list-style-type: none"> <li>Mostrar</li> <li>Eliminar</li> </ul>
5	LAB-1 PC-3	10.34.34.3	<ul style="list-style-type: none"> <li>Mostrar</li> <li>Eliminar</li> </ul>

Showing 1 to 6 of 6 entries

## Sección "Buscar Computadora"

	Acción del actor	Respuesta del sistema
1.	Introduce una cadena de texto en el campo Buscar.	
2.		Muestra las PC que contengan la cadena insertada en alguno de sus datos. (Ver interfaz 2).

**Interfaz 2**

# Capítulo 2: Análisis y diseño de la solución propuesta

## Sección “Modificar”

	Acción del Actor	Respuesta del sistema
1.	El usuario selecciona la opción mostrar de una de las computadoras existentes	
2.		Muestra una página que contiene en un formulario los datos de la PC para que puedan ser modificados. Los campos a mostrar son los mismos que los planteados en el RF7. (Ver interfaz 3).
3.	Modifica los datos que desee de la PC. <ul style="list-style-type: none"> <li>➤ IP (Obligatorio).</li> <li>➤ Nombre (Obligatorio).</li> </ul>	
4.		Verifica que no existan campos obligatorios vacíos.

## Capítulo 2: Análisis y diseño de la solución propuesta

5.		Valida que los valores introducidos sean correctos.
6.	Da clic en el botón "Modificar".	
7.		Modifica los datos y finaliza el Caso de Uso.

### Interfaz 3



#### 4ª Evento: Deja campos obligatorios vacíos

	Acción del actor	Respuesta del sistema
1.	No introduce datos en los campos obligatorios.	
2.		Señala en rojo los campos vacíos, regresa al paso 4 de la sección Modificar.

#### 5ª Evento: Introduce campos incorrectos

	Acción del actor	Respuesta del sistema
--	------------------	-----------------------

## Capítulo 2: Análisis y diseño de la solución propuesta

1.	Introduce datos incorrectos.	
2.		Señala en rojo los campos incorrectos, regresa al paso 5 de la sección Modificar.
<b>Sección “Eliminar”</b>		
	<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1.		Elimina la PC y finaliza el caso de uso.
<b>Sección “Nueva Computadora”</b>		
	<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1.	El usuario selecciona la opción nueva computadoras.	
2.		Muestra una página que contiene en un formulario los campos de la computadora a insertar. Los campos a mostrar son los mismos que los planteados en el RF7.
3.	Inserta los datos que desee de la PC. <ul style="list-style-type: none"> <li>➤ IP (Obligatorio).</li> <li>➤ Nombre (Obligatorio).</li> </ul>	
4.		Verifica que no existan campos obligatorios vacíos.
5.		Valida que los valores introducidos sean correctos.
6.	Da clic en el botón “Guardar”.	
7.		Inserta los datos y finaliza el Caso de Uso.
<b>4ª Evento: Deja campos obligatorios vacíos</b>		

## Capítulo 2: Análisis y diseño de la solución propuesta

	Acción del actor	Respuesta del sistema
1.	No introduce datos en los campos obligatorios.	
2.		Señala en rojo los campos vacíos, regresa al paso 4 de la sección Nueva Computadora.
<b>5ª Evento: Introduce campos incorrectos</b>		
	Acción del actor	Respuesta del sistema
1.	Introduce datos incorrectos.	
2.		Señala en rojo los campos incorrectos, regresa al paso 5 de la sección Nueva Computadora.

Las descripciones textuales de los CUS Autenticarse, Visualizar Reportes, Gestionar Horario, Gestionar Usuario y Asignar Permisos están recogidos en el [Anexo 1](#).

### 2.7 Diagrama de Clases de Diseño

El Diagrama de Clases del Diseño (DCD) describe gráficamente las especificaciones de las clases de software y las interfaces en una aplicación, o sea, muestra cómo quedaría implementada la aplicación en términos lógicos. Para un mejor entendimiento en la fig.4 se muestra el DCD del CU Gestionar PC.



## Capítulo 2: Análisis y diseño de la solución propuesta

- **Modelo:** es el encargado de administrar el comportamiento y los datos del dominio de aplicación.
- **Vista:** maneja la visualización de la información.
- **Controlador:** interpreta las acciones del *mouse* y el teclado, informando al modelo y/o a la vista para que cambien según resulte apropiado.

### 2.9 Patrones de Diseño

Un patrón es una solución a un problema en un contexto, codifica conocimiento específico acumulado por la experiencia en un dominio. Los desarrolladores lo usan como una forma de reutilizar la experiencia, clasificando las soluciones con términos de común denominación y van formando un amplio repertorio de principios generales y expresiones que los guían al crear un software. Se deben tener en cuenta los siguientes elementos de un patrón su nombre, el problema (¿cuándo aplicar un patrón?), la solución (descripción abstracta del problema) y las consecuencias (costos y beneficios) (Tedeschi, Nicolás, 2011). En el diseño de la aplicación se usarán los patrones Generales de Software para Asignar Responsabilidades (por sus siglas en ingles GRASP).

#### Patrones GRASP

Describen los principios fundamentales de diseño para la asignación de responsabilidades. Constituyen un apoyo para la enseñanza que ayuda a entender el diseño y el razonamiento para el diseño de una forma sistemática, racional y explicable (Grosso, Andrés, 2011).

A continuación, se muestran los patrones GRASP utilizados:

**Creador:** este patrón se utilizó para asignar responsabilidades relacionadas con la creación de objetos producidos en cualquier momento, en la siguiente figura se muestra el uso del mismo.

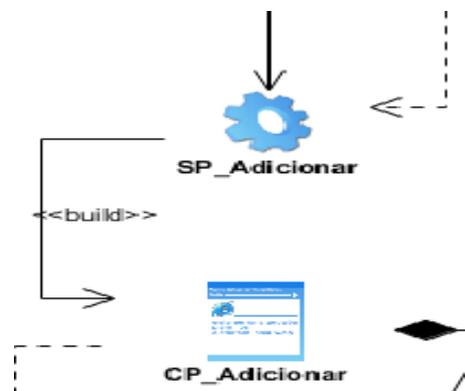


Fig. 5 Patrón GRASP Creador

## Capítulo 2: Análisis y diseño de la solución propuesta

**Controlador:** se utilizó este patrón para asignarle a las clases controladoras la responsabilidad de las operaciones del sistema, en la siguiente figura se muestra el uso del mismo.

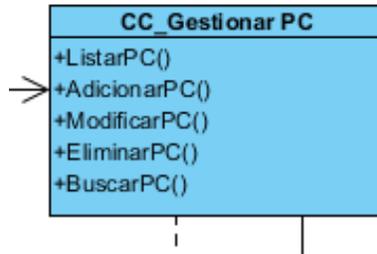


Fig. 6 Patrón GRASP Controlador

**Experto:** se utilizó este patrón para asignarle responsabilidad a la clase que cuenta con la información necesaria para cumplir la responsabilidad, en la siguiente figura se muestra el uso del mismo.

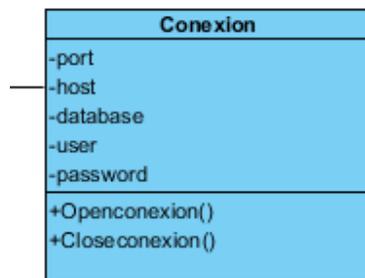


Fig. 7 Patrón GRASP Experto

### 2.10 Conclusiones

Con la realización de este capítulo se logró construir las bases para poder diseñar e implementar el sistema, consolidando el entendimiento del mismo a través de las definiciones de los casos de uso y las personas o actores involucradas en dichos procesos.

## CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS

### 3.1 Introducción

En este capítulo se expondrán los artefactos fundamentales correspondientes al flujo de trabajo. Se define el Modelo de Datos, el Diagrama de Componentes y el de Despliegue, elementos que contribuyen a la construcción del sistema. Por último, se realizan pruebas al sistema para corroborar que satisface las necesidades del cliente.

### 3.2 Modelo de Datos

Un modelo de datos es un conjunto de reglas y convenciones bien definidas que permite aplicar abstracciones con el fin de manipular los datos del mundo real que se desean almacenar en una base de datos (Somerville, 2005).

Para realizar el diseño de la base de datos primeramente se construye el diagrama de clases persistentes a partir de las clases del diseño que tienen un carácter permanente y a partir de ahí, se lleva a cabo la realización del modelo de datos del sistema.

En las fig. 8 y 9 se representan el diagrama de clases persistentes y el modelo de datos respectivamente.

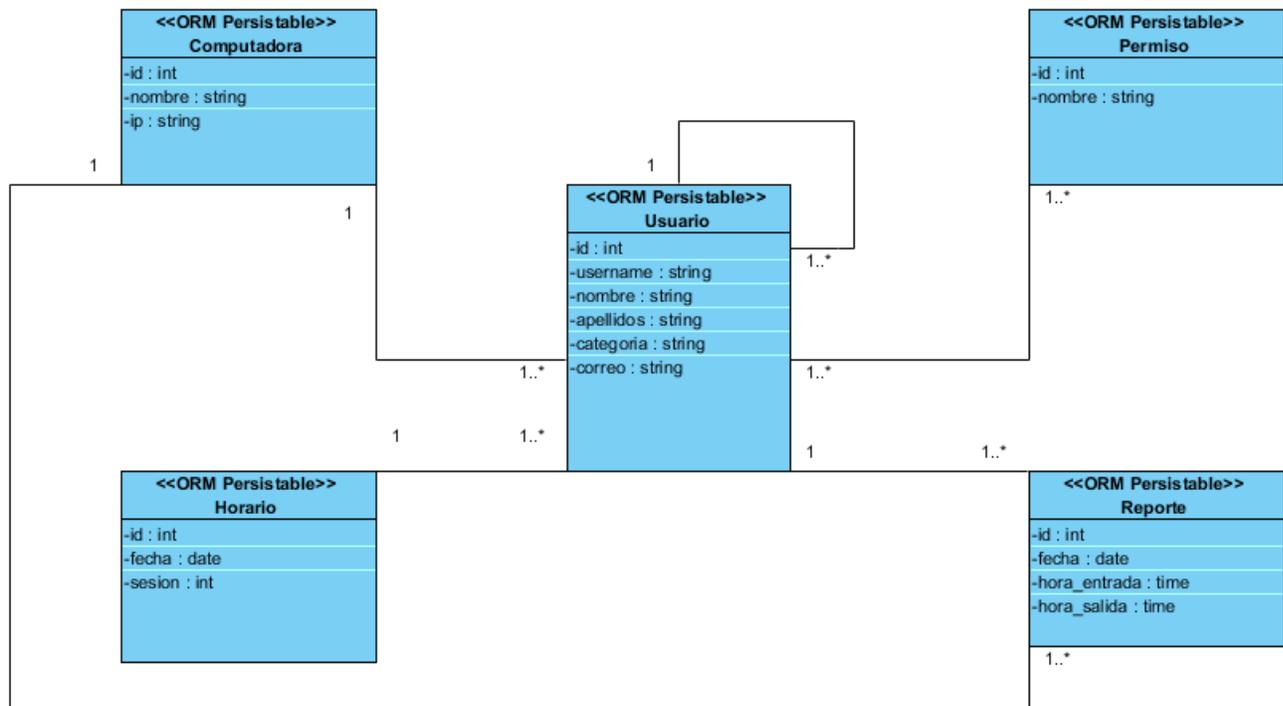


Fig. 8 Diagrama de Clases Persistentes

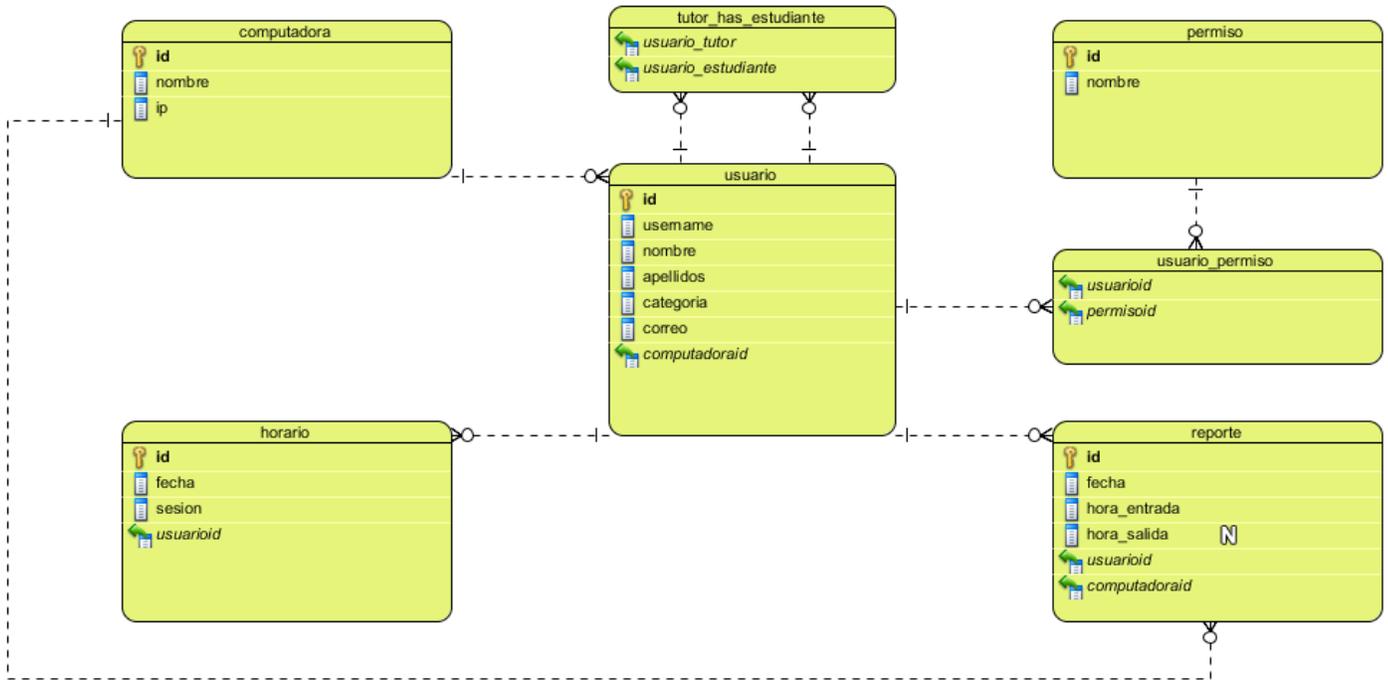


Fig. 9 Modelo Datos

### 3.3 Diagrama de Componente

Un diagrama de componentes muestra las dependencias lógicas entre componentes de software y se relacionan con los diagramas de clases ya que un componente normalmente se corresponde con una o más clases, interfaces o colaboraciones; pero un diagrama de componentes tiene un nivel más alto de abstracción que un diagrama de clases, usualmente un componente se implementa por una o más clases (u objetos) en tiempo de ejecución.

Un componente puede ser considerado como una unidad autónoma dentro de un subsistema, tiene una o más interfaces proporcionadas o requeridas, y sus interioridades permanecen ocultas o inaccesibles. Un componente se va modelando a través de todo el ciclo de desarrollo y sucesivamente se va refinando hasta llegar a su implantación y creación de su módulo ejecutable (Pressman, 2007).

En la fig. 10 se representa el diagrama de componentes del CU GestionarPC.

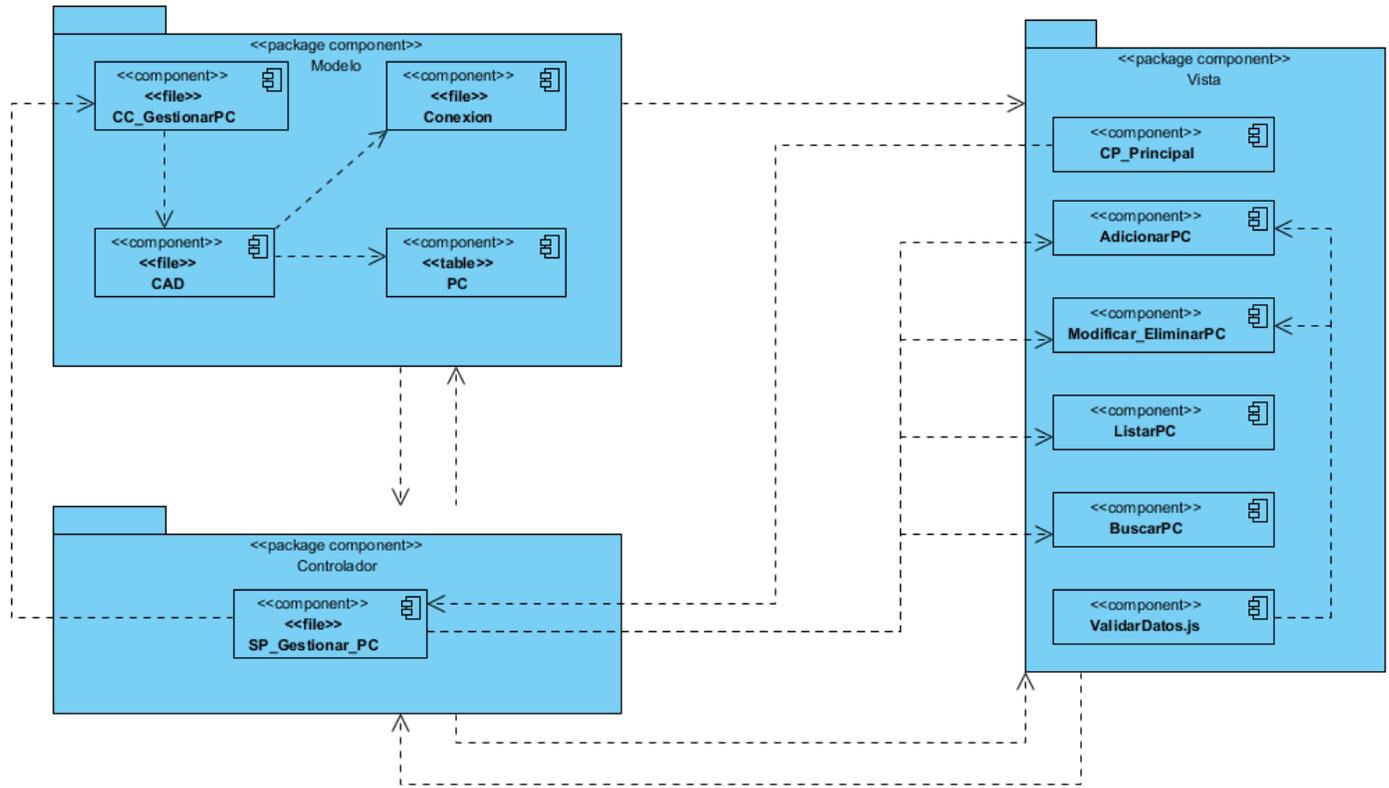


Fig. 10 Diagrama de Componente del CU Gestionar PC

La descripción de los diagramas de componente de los CU Autenticarse, Visualizar Reporte, Gestionar Usuario, Gestionar Horario y Asignar Permiso, se encuentran en el [Anexo 3](#).

### 3.4 Diagrama de Despliegue

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Se utiliza como entrada fundamental en las actividades de diseño e implementación debido a que la distribución del sistema tiene una influencia principal en su diseño (Rumbaugh, 2000). Un nodo es un objeto físico en tiempo de ejecución, es decir una máquina que se compone habitualmente de, por lo menos, memoria y capacidad de procesamiento, a su vez puede estar formada por otros componentes.

A continuación, la descripción de los nodos presentes en el diagrama:

- **PC Cliente:** representa el conjunto de computadoras a través de las cuales los usuarios pueden actualizar y consultar la información que se encuentra en el Servidor web. Para acceder al Sistema de Notificación y Asistencia a Actividades de la Producción, las PC Clientes utilizan un navegador web. la comunicación entre las PC Clientes y el Servidor web se establece utilizando el conjunto de protocolos de comunicación HTTP.

- **Servidor web:** se ubican las capas de presentación, lógica del negocio y de acceso a datos del sistema, así como los servicios que se brindan.
- **Servidor de Base de Datos:** representa un servidor PostgreSQL, en el cual se ubica toda la información persistente del sistema, almacenándose los datos que son utilizados y consultados por los usuarios del sistema.

En la fig. 11 se representa el diagrama de despliegue del sistema.

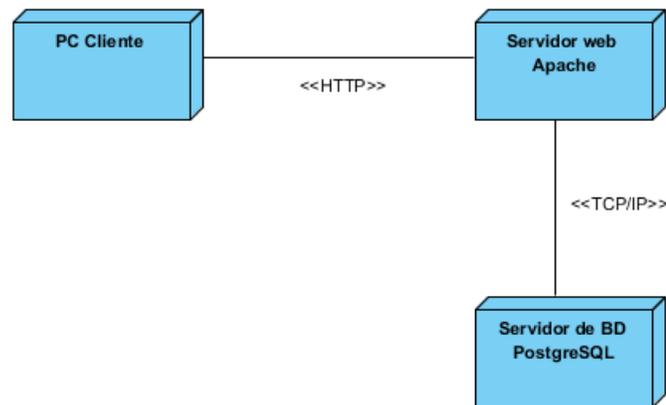


Fig. 11 Diagrama de Despliegue

### 3.5 Pruebas de la Solución

Una vez generado el código fuente es necesario comenzar las pruebas del sistema para descubrir y corregir la mayor cantidad de errores posibles antes de entregarlo al cliente. Su objetivo es diseñar una serie de casos de prueba que tengan una alta probabilidad de encontrar errores. Para ello se utilizan las técnicas de prueba que tengan una alta probabilidad de comprobar la lógica interna, las interfaces, las colaboraciones entre los componentes y los requisitos externos (Pressman, 2007).

#### 3.5.1 Pruebas de Caja Negra

Las pruebas de caja negra, también denominadas como pruebas de comportamiento, se concentran en los requisitos del software. Son las que se aplican a la interfaz del software. Se utilizan si se conoce la función específica para la que se diseñó el producto, con ella se debe demostrar que cada función es plenamente operacional, mientras se buscan los errores de cada una, teniendo poca relación con la estructura lógica del software (Pressman, 2007).

Específicamente dentro de esta prueba se utilizará la técnica de Partición Equivalente que se basa en una evaluación de las clases de equivalencia para una condición de entrada (Pressman, 2002). En esta

## Capítulo 3: Implementación y Pruebas

técnica se plantea que una clase de equivalencia representa un conjunto de estados validos o inválidos para condiciones de entrada.



Fig. 12 Interpretación gráfica de Caja Negra

A continuación, se muestran las pruebas realizadas a los CUS.

**CUS:** Gestionar PC

### Descripción general:

El caso de uso inicia cuando el actor trabajador accede a cualquiera de las opciones Listar Computadoras o Nueva Computadora en el menú Computadora, el sistema realiza la operación escogida y finaliza el caso de uso.

### Condición de ejecución:

El usuario debe estar autenticado y poseer los permisos.

### Escenarios a Probar en el CUS:

Tabla 3 Escenarios CUS Gestionar PC.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central
SC 1: Buscar Computadoras.	EC 1.1: Listar las computadoras.	El sistema muestra una lista con las PC que han sido adicionadas, mostrando de ellas el Id, Nombre, IP, además de las acciones a realizar sobre los mismos, eliminar y mostrar.	Clic en "Computadoras". Clic en "Listar Computadoras".

## Capítulo 3: Implementación y Pruebas

	EC 1.2: Buscar Computadora según texto.	Brinda la posibilidad de teclear la cadena de texto deseada para buscar las computadoras que contengan en sus datos dicha cadena.	Clic en "Computadoras". Clic en "Listar Computadoras". Escribir los caracteres deseados en el campo Buscar.
	EC 1.3: Cambiar la cantidad de elementos a mostrar.	Brinda la posibilidad de mostrar la cantidad de elementos deseados.	Clic en "Computadoras". Clic en "Listar Computadoras". Seleccionar en la opción Cantidad de elementos.
SC 2: Mostrar Computadora.	EC 2.1: Modificar datos de la computadora.	El sistema muestra una página con los datos de la computadora que pueden ser modificados y brinda la posibilidad de modificar los datos	Clic en "Computadoras". Clic en "Listar Computadoras". Clic en "Mostrar". Actualizar datos. Clic en el botón "Guardar".
SC 3: Eliminar Computadora.	EC 3.1: Eliminar computadora	El sistema elimina los datos de la computadora.	Clic en "Computadoras". Clic en "Listar Computadoras". Clic en "Eliminar". Actualizar datos.
SC 4: Adicionar Computadora.	EC 4.1: Adicionar computadora	El sistema adiciona los datos de la computadora.	Clic en "Computadoras". Clic en "Nueva Computadoras". Clic en "Adicionar". Actualizar datos.

### Descripción de las variables:

Tabla 4. Variables CUS Gestionar PC.

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

## Capítulo 3: Implementación y Pruebas

1	Buscar	Campo de texto	Si	Pueden ser letras o números.
2	Lista desplegable	Campo de texto	No	Debe seleccionar una cantidad.
3	Nombre	Campo de texto	No	Pueden ser letras y/o números.
4	Ip	Campo de texto	No	Deben ser números en el formato de Ip (ejemplo: 10.34.35.254).

### Matriz de Datos:

#### SC 1 Buscar Computadora

Tabla 5 MD Buscar Consultante.

ID del Escenario	Escenario	Buscar	Cantidad de Elementos	Respuesta del Sistema	Resultado de la Prueba
EC 1.1	Listar las computadoras.	N/A	N/A	El sistema muestra un listado con 10 (en caso de que haya menos con la cantidad existente) computadoras. Mostrando de cada una id, nombre, Ip y las acciones (eliminar y mostrar).	Satisfactorio.
EC 1.2	Buscar.	V/(10.34)	N/A	El sistema debe refrescar el listado y mostrar las computadoras que en alguno de sus campos tenga la cadena tecleada.	Satisfactorio
EC 1.3	Cambiar cantidad de elementos a mostrar.	N/A	V/(10)	El sistema debe mostrar la cantidad de computadoras que el usuario selecciona.	Satisfactorio.

## Capítulo 3: Implementación y Pruebas

### SC 2 Mostrar Computadora

Tabla 6 MD Mostar computadora.

ID del Escenario	Escenario	Nombre	IP	Respuesta del Sistema	Resultado de la Prueba
EC 2.1	Modificar datos de la computadora.	V/ (Lab 1 pc 2)	V/ (10.34.35.8)	El sistema actualiza los datos de la computadora y redirecciona a la página donde se muestra el listado de computadoras	Satisfactorio.
		I/ (vacío)	I/ (vacío)	El sistema no permite que queden campos vacíos señalándolos.	Satisfactorio.
		I/ (vacío)	V/ (10.34.35.54)		
		V/ (Lab 1 pc 12)	I/ (vacío)		
		V/ (Lab 3)	I/ (10.a)	El sistema no permite insertar datos incorrectos señalándolos.	Satisfactorio

### SC 3 Eliminar Computadora

Tabla 7 MD Eliminar Computadora.

ID del Escenario	Escenario	Buscar	Cantidad de Elementos	Respuesta del Sistema	Resultado de la Prueba
EC 3.1	Acceder a la opción de Eliminar.	N/A	N/A	El sistema elimina la computadora seleccionada.	Satisfactorio.

### SC 4 Adicionar Computadora

Tabla 8 MD Adicionar Computadora.

ID del Escenario	Escenario	Nombre	IP	Respuesta del Sistema	Resultado de la Prueba
EC 4.1	Acceder a la opción Nueva Computadora.	V/ (Lab 1 pc 2)	V/ (10.34.35.8)	El sistema adiciona la computadora.	Satisfactorio.
		I/ (vacío)	I/ (vacío)	El sistema no permite que queden campos vacíos señalándolos.	Satisfactorio.
		I/ (vacío)	V/ (10.34.35.54)		
		V/ (Lab 1 pc 12)	I/ (vacío)		

Los Diseños de Casos de Prueba correspondientes a los Casos de Uso Autenticarse, Visualizar Reporte, Gestionar Horario, Gestionar Usuarios y Asignar Permisos se encuentran en el [Anexo 3](#).

Después de realizar las pruebas al sistema en la primera iteración se encontraron una serie de no conformidades que fueron corregidas, luego se procedió a realizar una segunda iteración, la cual arrojó en algunos casos no conformidades y en otros resultados satisfactorios, luego se realizó una tercera iteración donde todos los resultados fueron satisfactorios, por lo que no se necesitó la realización de una cuarta iteración. En la Fig. 13 se representa una gráfica con una comparación de los resultados arribados en la primera, la segunda y tercera iteración de pruebas.

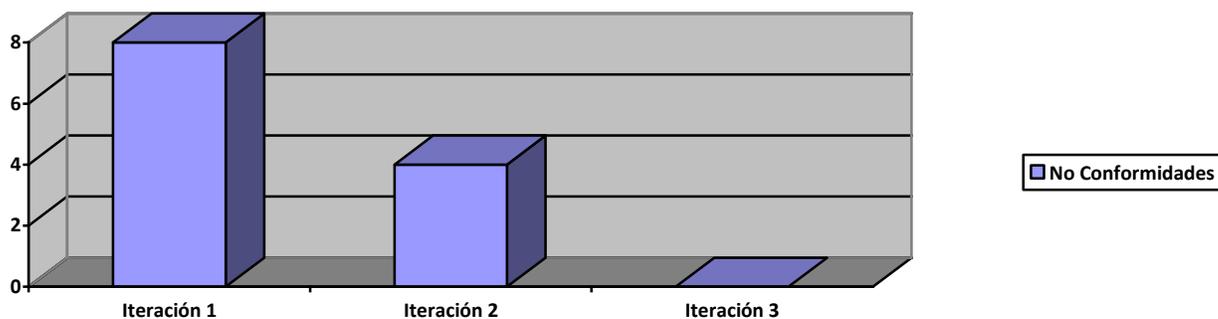


Fig. 13 Pruebas de Iteración

### 3.6 Conclusiones

Durante el desarrollo de este capítulo se generaron los artefactos correspondientes al flujo de trabajo. Con el Modelo de Datos se logró describir los elementos que intervienen en el problema y la forma en que se relacionan estos elementos entre sí. Se describió el Diagrama de Componentes donde se muestran los componentes fundamentales del CU Gestionar PC. También se elaboró el Diagrama de Despliegue con el que se muestra la distribución física del sistema. Finalmente se realizaron las pruebas pertinentes al módulo para verificar el cumplimiento de los requisitos que debe tener el sistema.

## CONCLUSIONES

Durante el desarrollo de la presente investigación se ha arribado a las siguientes conclusiones:

- Se logra una mayor organización y centralización de la información, trayendo consigo la eliminación de información redundante, pues son más reales los resultados que se obtengan a partir de su uso.
- Los artefactos generados en el proceso de desarrollo sirven para que otros desarrolladores obtengan un mejor entendimiento de su estructura, facilitando la agregación de nuevas funcionalidades y modificaciones posibles al mismo.

### RECOMENDACIONES

Luego de haber concluido el sistema y cumplido los objetivos trazados, se plantean las siguientes recomendaciones:

- Continuar agregándole funcionalidades al sistema según las necesidades que se requieran para mejorar el proceso de control y asistencia a las actividades de la producción, teniendo en cuenta los avances en las TIC, como: la incorporación de webcam.
- La implantación del Sistema de Control y Asistencia a Actividades de la Producción en otros centros o entidades con características similares en la UCI.

## BIBLIOGRAFÍA

**Alegsa, Leandro. 2010.** [En línea] 2010. [Citado el: 20 de Noviembre de 2015.] <http://www.alegsa.com.ar/Dic/lenguaje%20de%20programacion.php>.

**Alegsa, Leandro. 2009.** [En línea] 2009. [Citado el: 14 de Noviembre de 2015.] <http://www.alegsa.com.ar/Dic/sistema%20informatico.php>.

**Alvarez, Miguel Angel. 2012.** *Introducción a los lenguajes Web.* 2012.

**Avila, Katty. 2014.** Cavsi. [En línea] 2014. [Citado el: 12 de Noviembre de 2015.] <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.

**Cashersoft. 2011.** [En línea] 2011. <http://www.sicon.pe/sicon.html>.

**Ciberaula. 2011.** [En línea] 2011. [Citado el: 24 de octubre de 2015.] [http://linux.ciberaula.com/articulo/linux\\_apache\\_intro](http://linux.ciberaula.com/articulo/linux_apache_intro).

**Conceptode. 2014.** [En línea] 2014. [Citado el: 20 de Noviembre de 2015.] <http://concepto.de/metodologia/>.

**Eguiluz, Javier. 2010.** [En línea] 2010. [Citado el: 23 de noviembre de 2015.] [http://librosweb.es/libro/javascript/capitulo\\_1.html](http://librosweb.es/libro/javascript/capitulo_1.html).

**Garro, Arkaitz . 2014.** [En línea] 2014. [Citado el: 12 de octubre de 2015.] <http://www.arkaitzgarro.com/html5/capitulo-13.html>.

**Grosso, Andrés. 2011.** [En línea] 2011. [Citado el: 26 de Febrero de 2016.] <http://www.practicadesoftware.com.ar/2011/03/patrones-grasp>.

**GSINNOVA. 2010.** [En línea] 2010. [Citado el: 1 de Diciembre de 2015.] <http://www.rational.com.ar/herramientas/rup.html> .

**Larman, Craig. 2003.** [En línea] 2003. [Citado el: 12 de Marzo de 2015.] <http://is.ls.fi.upm.es/docencia/is2/documentacion/ModeloDominio.pdf...>

**Leandro Alegsa. 2009.** [En línea] 2009. [Citado el: 5 de Octubre de 2015.] <http://www.alegsa.com.ar/Dic/sistema%20informatico.php>.

**Librosweb. 2014.** [En línea] 2014. [Citado el: 6 de Diciembre de 2015.] [https://librosweb.es/libro/symfony\\_2\\_x/capitulo\\_4/el\\_sistema\\_de\\_bundles.html](https://librosweb.es/libro/symfony_2_x/capitulo_4/el_sistema_de_bundles.html).

- Mediavilla, Elena. 2014.** *UML: Modelo de casos de uso.* 2014.
- Melnikov, A. 2011.** [En línea] 2011. [Citado el: 29 de Septiembre de 2015.] <https://tools.ietf.org/html/rfc6455>.
- Menéndez, Evelyn Alonso. 2009.** [En línea] 2009. [Citado el: 12 de noviembre de 2015.] <http://www.monografias.com/trabajos73/herramientas-case-proceso-desarrollo-software/herramientas-case-proceso-desarrollo-software2.shtml>.
- Msc. González, Walfredo Hernández, Dra. Martínez, Marta Llantada y Dra Estrada, Vivian Sentí. 2004.** Monografías. [En línea] 2004. [Citado el: 13 de Noviembre de 2015.] <http://monografias.umcc.cu/monos/2003/monogWalfredoG.pdf>.
- Netbeans, Sitio Oficial. 2011.** [En línea] 2011. [Citado el: 14 de Noviembre de 2015.] [http://netbeans.org/index\\_es.html](http://netbeans.org/index_es.html).
- OMG. 2015.** [En línea] 2015. [Citado el: 24 de Noviembre de 2015.] <http://www.omg.org/spec/UML/2.5/PDF/>.
- Paez, Luis. 2010.** [En línea] 2010. <http://sistemaspaez.com/control-de-asistencia-con-webcam/>.
- PHP. 2014.** [En línea] 2014. [Citado el: 10 de Diciembre de 2015.] <https://secure.php.net/manual/es/migration55.new-features.php>.
- Pinta, Fausto Ramiro Muso y Salazar, Luis Efrain Llumistasig. 2013.** [En línea] 2013. [Citado el: 10 de Noviembre de 2015.] <http://repositorio.utc.edu.ec/bitstream/27000/1653/1/T-UTC-1527.pdf>.
- Postgresql, Sitio Oficial. 2014.** PostgreSQL. [En línea] 2014. [Citado el: 15 de octubre de 2015.] <https://www.postgresql.org/about/news/1557/>.
- Pressman, Roger. 2002.** *Ingeniería de Software: Un enfoque práctico.* 2002.
- . **2007.** *Ingeniería de Software: un enfoque práctico.* 2007.
- Red de Software, SPN. 2009.** [En línea] 2009. [Citado el: 15 de Noviembre de 2015.] [http://es.softpicks.net/software/Utilitarios/Herramientas-de-sistema/E-Time\\_es-166595.htm..](http://es.softpicks.net/software/Utilitarios/Herramientas-de-sistema/E-Time_es-166595.htm..)
- Rodríguez, Juanda. 2011.** [En línea] 2011. [Citado el: 12 de Noviembre de 2015.] <http://juandarodriguez.es/tutoriales/tutorial-de-symfony2/>.
- Rumbaugh, James. 2000.** *El Proceso Unificado de Desarrollo.* 2000.

**Sage. 2012.** [En línea] 2012. [Citado el: 1 de Noviembre de 2015.]  
[http://www.sage.com.ar/pages/informacion\\_asistencia.html](http://www.sage.com.ar/pages/informacion_asistencia.html).

**SecureTech. 2010.** [En línea] 2010. [Citado el: 4 de Noviembre de 2015.]  
[http://www.securetech.com.uy/documentos/securetech/SmartClk\\_20V316.pdf](http://www.securetech.com.uy/documentos/securetech/SmartClk_20V316.pdf).

**Somerville, Ian. 2005.** *Ingeniería de Software. Séptima Edición.* 2005.

SPN. Red Software. [En línea] [Citado el: 29 de octubre de 2015.]  
[http://es.softpicks.net/software/Utilitarios/Herramientas-de-sistema/E-Time\\_es-166595.htm..](http://es.softpicks.net/software/Utilitarios/Herramientas-de-sistema/E-Time_es-166595.htm..)

**Tedeschi, Nicolás. 2011.** [En línea] 2011. [Citado el: 21 de Febrero de 2015.]  
<http://msdn.microsoft.com/es-es/library/bb972240.aspx>.

**Torres, Mirelys Pérez. 2010.** *Sistema para el Control de Asistencia.* 2010.

**Trazar. 2010.** *BUENAS PRACTICAS EN DESARROLLO DE SOFTWARE.* 2010.

**Ulloa, Myriam Nataly Romero y Tualumbo, Mariela Cristina. 2011.** [En línea] 2011. [Citado el: 24 de Noviembre de 2015 .] <http://repositorio.utc.edu.ec/bitstream/27000/432/1/T-UTC-1010.pdf>.

**Vázquez Mariño, Carlos. 2008.** [En línea] 2008. [Citado el: 15 de Noviembre de 2015.]  
[http://administraciondesistemas.pbworks.com/f/Manual\\_PHP5\\_Basico.pdf](http://administraciondesistemas.pbworks.com/f/Manual_PHP5_Basico.pdf).

**Victoria. 2009.** [En línea] 2009. [Citado el: 20 de noviembre de 2015.]  
[www.definicionabc.com/tecnologia/html.php..](http://www.definicionabc.com/tecnologia/html.php..)