

Universidad de las ciencias Informáticas

Facultad 6



“Sistema de información para el área de residencia y extensión universitaria de la Facultad 6.”

Trabajo de Diploma para Optar por el Título de Ingeniero en Ciencias Informáticas.

Autora:

Lipsy Mariem Legrá Legrá

Tutor:

MSc. Asnay Guirola González

Ing. Lázaro Cánova Amador

Co-Tutor:

MSc. Gregorio Morales González

La Habana, junio 2011
“Año 53 de la Revolución”

Frase

“No estudio por saber más, sino por ignorar menos...”

Declaración de Autoría

DECLARACIÓN DE AUTORÍA

Declaro ser autora de la presente tesis y reconozco a la Facultad 6 de la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma.

Para que así conste firmo la presente a los ____ días del mes de Junio del año 2011.

Firma del Autor

Firma del Tutor

Firma del Co-Tutor

Datos de contacto

Datos de contacto

Msc. AsnayGuirola González:

Graduado de Ing. en Ciencias Informáticas se desempeña como profesor de Programación desde hace varios años, tiene la categoría docente de instructor y el grado científico de Máster en Informática Aplicada. El mismo ha estado vinculado a varios proyectos productivos, en los cuales ha obtenido buenos resultados, además de participar en varios eventos nacionales e internacionales.

Email: aguirola@uci.cu

Ing. Lázaro Cánova Amador

Graduado de Ing. en Informática en la CUJAE, se desempeña como profesor de Ingeniería de software. El mismo ha estado vinculado a varios proyectos productivos, en los cuales ha obtenido buenos resultados, además de participar en varios eventos nacionales e internacionales.

Email: lcanova@uci.cu

Agradecimientos

Agradecimientos

A mis padres por su constante apoyo, por confiar en mí y ser la razón de mi existir.

A mi abuelita que tantos valores me ha inculcado y es la lucecita que ilumina mi camino.

A mi familia, por su apoyo incomparable durante toda mi carrera.

A Aliandy, por confiar en mí, apoyarme, y demostrarme que en tan poco tiempo se es capaz de sentir tal como si lo vivido fueran años.

A mis amistades, por permitirme tener la familia más grande que jamás había soñado y vivir momentos inolvidables a su lado, por quererme, cuidarme, corregirme, regañarme, reír y llorar a mi lado.

A Eduardo, por ser mi impulsor, consejero, guía y amigo en el desarrollo de esta tesis.

A mis tutores Lázaro y Asnay por el tiempo dedicado, su comprensión y su apoyo incondicional en cada paso dado en el transcurso de este año.

Dedicatoria

Dedicatoria

A mis padres.

Resumen

Resumen

Con el desarrollo de las Tecnologías de Información, los Sistemas de Información han cambiado la forma en que operan las organizaciones actuales. A través de su uso se logran importantes mejoras, pues automatizan los procesos operativos y suministran una plataforma de información necesaria para la toma de decisiones, logrando con su implantación ventajas competitivas.

En la Universidad de las Ciencias Informáticas, se hace necesario el desarrollo de un sistema de información que permita medir correctamente y de forma confiable algunas variables necesarias para obtener indicadores que faciliten la toma de decisiones en el área de residencia y extensión universitaria de la facultad 6.

Este trabajo abarca el diseño e implementación de un sistema de información que posibilite la generación de reportes estadísticos y la visualización gráfica de los indicadores claves del desempeño en esta área, que faciliten la toma de decisiones a los directivos.

Palabras claves: Sistema de Información, Indicadores.

Índice

Índice

Capítulo 1 Fundamento Teórico	5
1.1 Sistemas de información.	5
1.1.1 Definición	5
1.1.2 Clasificación de los sistemas de información.	6
1.1.3 Indicadores.	9
1.2 Sistemas de información en el área de residencia estudiantil existentes en la universidad.	10
1.3 Tecnologías desarrolladas por DATEC para la gestión de la información.	11
1.4 Arquitectura de software.	13
1.4 .1 Estilos arquitectónicos	15
1.4.2 Patrones de Diseño.	17
1.5 Herramientas, tecnologías y lenguajes para el desarrollo.	18
1.5.1 Metodologías de desarrollo de software.	18
1.5.2 Lenguaje Unificado de Modelado.	19
1.5.3 Herramienta CASE	20
1.5.4 Entornos de Desarrollo (IDEs)	21
1.5.5 Lenguajes de programación empleados para el desarrollo.	21
1.5.6 Framework para el desarrollo.	23
1.5.7 Sistema Gestor de Base de Datos.	24
1.5.8 Servidor Web. Apache 2.2	25
Capítulo 2 Características del sistema	26
2.1 Modelado del negocio.	26
2.1.1 Identificación de los actores y trabajadores del negocio.	27
2.1.2 Indicadores claves del desempeño en el área de residencia y extensión universitaria.	29
2.1.3 Unidades de observación para cada proceso.	30
2.1.4 Diagrama de casos de uso del negocio.	30
2.1.5 Descripción de los casos de uso del negocio.	31
2.1.6 Modelo de objetos.	33

Índice

2.1.7 Reglas del negocio. -----	34
2.2 Levantamiento de requisitos. -----	34
2.2.1 Requisitos funcionales. -----	35
2.2.2 Requisitos no funcionales. -----	36
2.3 Propuesta de Arquitectura. -----	38
2.4 Modelado del sistema. -----	40
2.4.1 Actores del sistema. -----	40
2.4.2 Diagrama de casos de uso del sistema. -----	42
2.4.3 Descripción de los casos de uso del sistema. -----	43
2.5 Diagrama de Clases del Diseño -----	45
.Capítulo 3 Implementación del Sistema. -----	47
3.1 Modelo de Implementación -----	47
3.1.1 Diagrama de componentes -----	47
3.1.2 Diagrama de Despliegue -----	52
Capítulo 4: Pruebas del sistema. -----	54
4.1 Pruebas de Software. -----	54
4.1.1 Niveles de pruebas seleccionados. -----	54
4.1.2 Técnicas de pruebas seleccionadas. -----	55
4.1.3 Métodos de pruebas seleccionados. -----	55
Conclusiones -----	58
Recomendaciones -----	59
Referencias Bibliográficas. -----	60
Bibliografía -----	63
Glosario de Términos -----	66

Tablas y Figuras

Tablas y Figuras

<i>Figura 1: Arquitectura Cliente/Servidor</i>	14
<i>Figura 2: Modelo de Casos de Uso del negocio.</i>	30
<i>Figura 3: Modelo de Objeto del negocio.</i>	33
<i>Figura 4: Arquitectura del Sistema</i>	39
<i>Figura 5: Diagrama de Caso de Uso del Sistema</i>	42
<i>Figura 6: Diagrama de Clases del Diseño</i>	46
<i>Figura 7: Componentes de la Capa de Presentación.</i>	48
<i>Figura 8: Componentes de la Capa de Negocio</i>	49
<i>Figura 9: Componentes de la Capa de Acceso a Datos</i>	50
<i>Figura 10: Componentes de la Capa de Datos</i>	51
<i>Figura 11: Diagrama de componente del sistema</i>	52
<i>Figura 12: Diagrama de Despliegue</i>	52
<i>Tabla 1: Actores del negocio.</i>	27
<i>Tabla 2: Trabajadores del negocio.</i>	27
<i>Tabla 3: Principales procesos del negocio</i>	28
<i>Tabla 4: Indicadores del desempeño</i>	29
<i>Tabla 5 : Unidades de Observación para cada Proceso</i>	30
<i>Tabla 6: Descripción del caso de uso "Solicitar información de la residencia"</i>	31
<i>Tabla 7: Descripción del caso de uso "Procesar información de la residencia"</i>	32
<i>Tabla 8: Actores del sistema.</i>	40
<i>Tabla 9: Descripción del CUS Mostrar reporte de cuartería</i>	43
<i>Tabla 10: Descripción del CUS Llenar modelo de cuartería</i>	44
<i>Tabla 11: Descripción del CUS Visualizar Tablero Digital</i>	45
<i>Tabla 12: Caso de prueba "Visualizar Tablero Digital"</i>	56
<i>Tabla 13: Escenario de Prueba: "Visualizar Gráficos del Tablero Digital"</i>	57

Introducción

Introducción

El avance de las tecnologías en el mundo se ha convertido en una herramienta al servicio de las estrategias de cada institución. En Cuba la informatización de sus instituciones ha alcanzado gran auge, por lo que se desarrollan sistemas para la gestión de información de cada una de las mismas con el objetivo de lograr calidad y rapidez a la hora de procesar la información que se maneja en el desarrollo de sectores productivos, la salud, la actividad pública y la educación. La Universidad de las Ciencias Informáticas (UCI), fruto de la batalla de ideas, juega un papel importante en todo el proceso de informatización que se lleva a cabo en el país, desarrollándose en ella productos de software necesarios para diferentes instituciones cubanas e internacionales, y es por ello que hacer de la UCI una universidad donde todos y cada uno de los procesos estén sustentados sobre una infraestructura tecnológica capaz de responder a las necesidades de los que aquí conviven, es una tarea en la que directivos, trabajadores y estudiantes están inmersos desde sus inicios. El área de Extensión y Residencia, identificada como una de las áreas de resultados claves en el desarrollo creciente de la universidad, se sustenta sobre la base de la planificación, organización y control de todas las actividades no curriculares realizadas por los estudiantes y trabajadores que en ella conviven.

Actualmente la secretaria del vicedecano de residencia recibe toda la documentación en copia dura (papel) o en formato digital (Excel, Word etc.) de la participación en las actividades extensionistas (Juegos deportivos y Festivales de Cultura) y del cumplimiento de los deberes de los becarios, tales como: cuartería, guardia obrera estudiantil, trabajo socialmente útil y paradas de beca que le hacen llegar las instructoras de la residencia, los encargados de esta área por la FEU y los profesores asignados como responsables en estos tipos de controles. Sucede de igual modo con los reportes de estado de las afectaciones tanto de mantenimiento como tecnológicas. Estos datos llegan a la secretaria vía correo, a través de dispositivos de almacenamiento portables o en las plantillas impresas creadas con estos fines o entregadas personalmente. El procesamiento de estos datos se hace a medida que se van recibiendo, lo que provoca atraso en la disponibilidad de la información obtenida y que en ocasiones se creen ficheros duplicados por no llevar un correcto control de versiones. Esto provoca que no se cuente con la información actualizada disponible que permita medir correctamente y de forma confiable alguna variable de los procesos de gestión de la información, necesaria para obtener indicadores que permitan o faciliten la toma de decisiones en esta área.

Introducción

Analizando lo anteriormente planteado se identifica como **problema a resolver**: ¿Cómo contribuir a la gestión de la información en el área de residencia y extensión universitaria de la Facultad 6?

El problema planteado se enmarca en el **objeto de estudio**: Sistemas de información.

El objeto delimita el **campo de acción**: Sistema de información del área de extensión y residencia de la Facultad 6.

Para dar solución al problema se define como **objetivo general**: Desarrollar un sistema de información para el área de la residencia y extensión universitaria de la Facultad 6.

Objetivos específicos:

- Fundamentar los referentes teóricos de los sistemas de información existentes.
- Modelar el sistema a partir de los requerimientos identificados.
- Diseñar el sistema propuesto.
- Implementar un sistema de información con énfasis en la reutilización de activos existentes en el Centro de Tecnologías de Gestión de Datos.
- Validar el sistema mediante pruebas de integración y sistema.

Para lograr los objetivos, se realizarán las siguientes **tareas**:

- Caracterización de los sistemas de información.
- Identificación de las necesidades de información para el área de residencia y extensión universitaria de la Facultad 6.
- Identificación de posibles módulos a reutilizar teniendo como bases los sistemas desarrollados en DATEC.
- Selección de las herramientas y tecnologías para la implementación del sistema de información.
- Captura de los requisitos que debe cumplir el sistema.
- Identificación de los actores y casos de uso para el modelado del diagrama de casos de uso del sistema.

Introducción

- Modelado del diagrama de casos de uso del sistema.
- Descripción de los casos de uso del sistema.
- Identificación de las clases del diseño, sus atributos y funciones.
- Modelado del diagrama de clases del diseño.
- Identificación de los componentes y subsistemas del sistema a desarrollar.
- Modelado del diagrama de componentes.
- Implementación del sistema.
- Diseño de las pruebas.
- Realización de las pruebas.
- Valoración de los resultados de las pruebas.

Métodos Teóricos

Análisis y síntesis: Permite especificar los elementos más importantes relacionados con los procesos que tienen lugar en el área de la residencia y extensión universitaria de la Facultad 6. Se realizó un análisis de los indicadores claves de los procesos en esta área, obteniéndose los más relevantes.

Histórico Lógico Tendencial: Se realizó una amplia revisión bibliográfica en diferentes artículos digitales, sitios web y documentos actuales, con el objetivo de conocer acerca de los sistemas de información. Surgiendo así la necesidad del desarrollo de un sistema de información para el área de residencia y extensión universitaria de la Facultad 6 y los diferentes componentes para la implementación del mismo.

Inductivo deductivo: En función del problema a resolver planteado, se pone en práctica un grupo de conocimientos sobre las tendencias actuales y características generales de los procesos relacionados.

Modelación: Con el objetivo de realizar el análisis y diseño del software y posteriormente la implementación.

Métodos Empíricos

Introducción

Entrevista: Se realizó para obtener las características de los procesos que tienen lugar en el área de residencia y extensión universitaria, y capturar los requisitos del sistema de información necesarios para el desarrollo del sistema.

El documento está estructurado de la siguiente forma:

Capítulo 1: Fundamento teórico. En el presente capítulo se realizará un estudio de los sistemas de información, reflejando sus características y clasificación. Además se efectuará el análisis de la arquitectura, las herramientas, metodologías y lenguajes de programación a utilizar para el desarrollo del sistema de información a desarrollar, tomando como punto de partida las características y ventajas que ofrecen cada uno de ellos.

Capítulo 2: Características del sistema. En el presente capítulo se aborda lo referente al modelado del negocio, captura de requisitos, modelado y diseño del sistema, con la presentación de sus principales artefactos.

Capítulo 3: Implementación del sistema. Contiene la representación de los diagramas de componente y despliegue referentes al sistema a implementar.

Capítulo 4: Pruebas del sistema. Se muestran los resultados de las pruebas aplicadas y un análisis de los resultados obtenidos.

Capítulo 1 Fundamento Teórico

Introducción

En el presente capítulo se realizará un estudio de los sistemas de información, reflejando sus características y clasificación. Además se efectuará el análisis de la arquitectura, las herramientas, metodologías y lenguajes de programación a utilizar para el desarrollo del sistema de información a desarrollar, tomando como punto de partida las características y ventajas que ofrecen cada uno de ellos.

1.1 Sistemas de información.

1.1.1 Definición

Con el desarrollo de las Tecnologías de Información (TI) los Sistemas de Información (SI) han cambiado la forma en que operan las organizaciones actuales. A través de su uso se logran importantes mejoras, pues automatizan los procesos operativos, suministran una plataforma de información necesaria para la toma de decisiones y, lo más importante, su implantación logra ventajas competitivas y reduce las ventajas de los rivales. Un Sistema de Información es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una organización, al permitir operar casi todos los activos tangibles e intangibles de la institución y llegar a convertirse en la herramienta integral de gerencia más cotizada y necesaria para alcanzar con éxito los resultados propuestos por la organización.

“Sistema de información es un conjunto de personas, maquinaria y procedimientos que integrados hacen posible a los individuos trabajar con entradas y demandas que aparecen en el trabajo cotidiano.” [\[1\]](#)

“Un sistema de información es un conjunto de elementos que interactúan entre sí con el fin de apoyar a las actividades de una empresa o negocio. Tratan el desarrollo, uso y administración de la infraestructura de la tecnología de la información en una organización.” [\[2\]](#)

“Sistema de información es la combinación de recursos humanos y materiales que resultan de las operaciones de almacenar, recuperar y usar datos con el propósito de una gestión eficiente en las operaciones de las organizaciones.” [\[3\]](#)

Varios autores han emitido sus definiciones sobre dicho tema, lo que permite concluir que un Sistema de Información es un conjunto de elementos orientados al tratamiento y administración

Capítulo 1 Fundamento Teórico

de datos e información, organizados y listos para su posterior uso, generados para cubrir una necesidad.

Un Sistema de Información realiza cuatro actividades básicas: [\[4\]](#)

- ✓ **Entrada de Información:** es el proceso mediante el cual el sistema de gestión de información toma los datos que requiere para procesar la información.
- ✓ **Almacenamiento de información:** es una de las actividades o capacidades más importantes que tiene una computadora, ya que a través de esta propiedad el sistema puede recordar la información guardada en la sección o proceso anterior. Esta información suele ser almacenada en estructuras de información denominadas archivos.
- ✓ **Procesamiento de Información:** es la capacidad del sistema de gestión de información para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecida. Estos cálculos pueden efectuarse con datos introducidos recientemente en el sistema o bien con datos que están almacenados.
- ✓ **Salida de Información:** es la capacidad de un sistema de gestión de información para sacar la información procesada o bien datos de entrada al exterior. Las unidades típicas de salida son las impresoras, graficadores, cintas magnéticas, disquetes.

1.1.2 Clasificación de los sistemas de información.

Existen varias categorías de sistemas de información, orientadas cada una a necesidades específicas presentadas por distintas organizaciones. Pero a pesar de ello, estos sistemas pueden ser agrupados de acuerdo a sus características principales en tres grandes grupos: los Transaccionales, los de Apoyo a las Decisiones y los Estratégicos.

➤ **Sistemas de Información Transaccionales:**

Estos son los que logran la automatización de procesos operativos dentro de una organización, son llamados así ya que se encargan de los procesos de rutina de la organización, incluyendo aplicaciones para el mantenimiento de registros. Estos son unos de los más utilizados ya que reducen considerablemente el tiempo de realización de las actividades rutinarias de la organización.

Capítulo 1 Fundamento Teórico

Sus principales características son [5]:

- A través de éstos suelen lograrse ahorros significativos de mano de obra, debido a que automatizan tareas operativas de la organización.
- Con frecuencia son el primer tipo de sistemas de información que se implanta en las organizaciones. Se empieza apoyando las tareas a nivel operativo de la organización para continuar con los mandos intermedios y posteriormente con la alta administración conforme evolucionan.
- Son intensivos en entrada y salida de información; sus cálculos y procesos suelen ser simples y poco sofisticados. Estos sistemas requieren mucho manejo de datos para poder realizar sus operaciones y como resultado generan también grandes volúmenes de información.
- Tienen la propiedad de ser recolectores de información, es decir, a través de estos sistemas se cargan las grandes bases de información para su explotación posterior.
- Son los encargados de integrar gran cantidad de la información que se maneja en la organización, la cual será utilizada posteriormente para apoyar a los mandos intermedios y altos.
- Son fáciles de justificar ante la dirección general, ya que sus beneficios son visibles y palpables. El proceso de justificación puede realizarse enfrentando ingresos y costos. Esto se debe a que en el corto plazo se pueden evaluar los resultados y ventajas que se derivan del uso de este tipo de sistemas. Entre las ventajas que puede medirse se encuentra el ahorro de trabajo manual.
- Son fácilmente adaptables a paquetes de aplicación que se encuentran en el mercado, ya que automatizan los procesos básicos que por lo general son similares o iguales en otras organizaciones. Ejemplos de este tipo de sistemas son la facturación, nóminas, cuentas por cobrar, cuentas por pagar, contabilidad general, conciliaciones bancarias e inventarios.

➤ **Sistemas de Información de Apoyo a Decisiones:**

Son diseñados con el fin de apoyar el proceso de toma de decisiones de los directivos de las organizaciones. Identifican la información necesaria para tomar decisiones a través de procesos de decisión claramente definidos. Se encargan de apoyar en la tomas de decisiones administrativas brindando periódicamente reportes de la organización con una forma y formato normalizados, ejemplo de estos tenemos los Sistemas de Soporte a la Toma de Decisiones, los

Capítulo 1 Fundamento Teórico

Sistemas para la Toma de Decisiones de Grupos y los Sistemas de información para ejecutivos.

Las principales características de estos sistemas son las siguientes: [6]

- Suelen emplearse después de haber implantado los Sistemas Transaccionales más relevantes de la organización, ya que estos últimos constituyen su plataforma de información.
- La información que generan sirve de apoyo a los mandos intermedios y a la alta administración en el proceso de toma de decisiones.
- Suelen ser intensivos en cálculos y escasos en entradas y salidas de información. Así, por ejemplo, un modelo de planeación financiera requiere poca información de entrada, que genera poca información como resultado, pero a la vez puede realizar muchos cálculos durante su proceso.
- No suelen ahorrar mano de obra. Debido a ello, la justificación económica para el desarrollo de estos sistemas es difícil, ya que no se conocen los ingresos del proyecto de inversión.
- Suelen ser sistemas de información interactivos y amigables, con altos estándares de diseño gráfico y visual, ya que están dirigidos al usuario final.
- Apoyan la toma de decisiones que, por su misma naturaleza son repetitivas y de decisiones no estructuradas que no suelen repetirse. Por ejemplo, un Sistema de Compra de Materiales que indique cuando debe hacerse un pedido al proveedor o un Sistema de Simulación de Negocios que apoye la decisión de introducir un nuevo producto al mercado.
- Estos sistemas pueden ser desarrollados directamente por el usuario final sin la participación operativa de los analistas y programadores del área de informática.
- Este tipo de sistemas puede incluir la programación de la producción, compra de materiales, flujo de fondos, proyecciones financieras, modelos de simulación de negocios y modelos de inventarios entre otros.

➤ **Sistemas de Información Estratégicos:**

Estos sistemas son desarrollados por las organizaciones con el fin de lograr ventajas competitivas a través del uso de la tecnología de información.

Capítulo 1 Fundamento Teórico

Sus principales características son: [7]

- La función principal de estos no es la de apoyar la automatización de procesos operativos, ni la de proporcionar información para apoyar la toma de decisiones. Sin embargo, puede llevar a cabo dichas funciones.
- Normalmente son desarrollados en casa, es decir dentro de la organización, razón por la cual pueden adaptarse fácilmente a paquetes disponibles en el mercado.
- Típicamente su forma de desarrollo es con base a incrementos y a través de su evolución dentro la organización. Este comienza con un proceso o función en particular y a partir de esta se van agregando nuevas funciones o procesos.
- Su tarea es la de lograr ventajas que los, competidores no posean, tales como ventajas en costos y servicios diferenciados con clientes y proveedores. En este contexto, los Sistemas Estratégicos son creadores de barreras de entrada al negocio. Ejemplo de esto tenemos el uso de cajeros automáticos en los bancos es un Sistema Estratégico, ya que brinda ventaja sobre un banco que no posee tal servicio. Si un banco nuevo decide abrir sus puertas al público, tendrá que dar este servicio para tener un nivel similar al de sus competidores.
- También sirven de apoyo al proceso de innovación de productos y procesos dentro de la organización, debido a que buscan ventajas respecto a los competidores y una forma de hacerlo es innovando o creando productos y procesos.

De los sistemas de información analizados el que se adecua a las características del sistema que se desea proponer para el área de residencia de la facultad 6 es el Sistema de Información de Apoyo a Decisiones, pues como de su nombre se puede inferir, del procesamiento de los datos se obtendrá la información necesaria para la toma de decisiones.

1.1.3 Indicadores.

Un indicador es la medida cuantitativa o la observación cualitativa que permite identificar cambios en el tiempo y cuyo propósito es determinar qué tan bien está funcionando un sistema, dando la voz de alerta sobre la existencia de un problema y permitiendo tomar medidas para solucionarlo, una vez se tenga claridad sobre las causas que lo generaron. Los Indicadores pueden ser medidas, números, hechos, opiniones o percepciones que señalen condiciones o situaciones específicas.

Estos tienen gran importancia ya que permiten:

Capítulo 1 Fundamento Teórico

Medir cambios en esa condición o situación a través del tiempo.

- Facilitan mirar de cerca los resultados de iniciativas o acciones.
- Son instrumentos muy importantes para evaluar y dar surgimiento al proceso de desarrollo.
- Son instrumentos valiosos para orientarnos de cómo se pueden alcanzar mejores resultados en proyectos de desarrollo.

Estos se caracterizan en: [\[8\]](#)

- **Indicadores Cuantitativos:** Son los que se refieren directamente a medidas en números o cantidades.
- **Indicadores Cualitativos:** Son los que se refieren a cualidades. Se trata de aspectos que no son cuantificados directamente. Se trata de opiniones, percepciones o juicio de parte de la gente sobre algo.
- **Indicadores Directos:** Son aquellos que permiten una dirección directa del fenómeno.
- **Indicadores Indirectos:** Cuando no se puede medir de manera directa la condición económica, se recurre a indicadores sustitutivos o conjuntos de indicadores relativos al fenómeno que nos interesa medir o sistematizar.
- **Indicadores Positivos:** Son aquellos en los cuales si se incrementa su valor estarían indicando un avance hacia la equidad.
- **Indicador Negativo:** Son aquellos en los cuales si su valor se incrementa estarían indicando un retroceso hacia la inequidad.

1.2 Sistemas de información en el área de residencia estudiantil existentes en la universidad.

Existen varias aplicaciones que de una forma u otra gestionan los procesos de la Residencia y extensión universitaria, los cuales tratan el control de la guardia y la cuartelaría, o la gestión de los edificios para la ubicación de los estudiantes o bien las evaluaciones de los mismos en la beca, algunos de estos sistemas son:

Sistema de Gestión de información de la Facultad 8 (SGIF 8) y el sistema para la gestión de la información de profesores y estudiantes de la facultad 6: Desarrollo del Módulo Residencia. Ambos sistemas permiten gestionar los datos referentes a la mayoría de los procesos realizados en la facultad incluyendo los extensionistas y de residencia estudiantil como:

Capítulo 1 Fundamento Teórico

Ubicar a los estudiantes en los apartamentos de los edificios con los que dispone la facultad.

- Conocer en qué edificio, paso de escalera y apartamento de la residencia se encuentra ubicado un estudiante de la facultad o un grupo de estudiantes.
- Emitir partes de guardia.
- Conocer cómo se ha comportado la guardia estudiantil de una brigada específica hasta el momento.

Estos sistemas no poseen el soporte necesario para la generación de reportes estadísticos que visualicen indicadores claves del desempeño en esta área que faciliten la toma de decisiones a los directivos.

1.3 Tecnologías desarrolladas por DATEC para la gestión de la información.

El Centro de Tecnología de Gestión de Datos ha desarrollado varias herramientas para la gestión de información del centro, los cuales junto a otros módulos desarrollados en la universidad, por sus características, permitirían suplir las carencias de los sistemas de gestión de información del área de residencia existentes en el entorno universitario, entre los que se destacan:

SIGE

Sistema Integrado de Gestión Estadística encargado de la automatización los principales procesos que se llevan a cabo en la Oficina Nacional de Estadística (ONE). Diseñado netamente en una plataforma de código abierto. Es un sistema modular dividido en cuatro módulos fundamentales: Módulo de Gestión de la Configuración (SIGE-MGC), Módulo Diseñador de Formularios (SIGE-MDF), Módulo de Entrada de Datos (SIGE-MED) y Módulo Generador de Reportes (SIGE-MGR). [\[9\]](#)

Módulo Gestión de la Configuración (SIGE-MGC)

El Módulo Gestión de Configuración es el encargado del proceso de captación y procesamiento de la información estadística en relación con las unidades de observación. El mismo cuenta registros que identifican a las distintas unidades de observación y de clasificadores que permiten agruparlas y caracterizarlas en un lenguaje adecuado. [\[10\]](#)

Capítulo 1 Fundamento Teórico

Módulo Diseñador de Formularios

El Módulo Diseñador de Formularios (MDF) es una herramienta perteneciente al Paquete de Ayuda a la Toma de Decisiones y Soluciones Integrales (PATDSI) este da la oportunidad al usuario de diseñar los formularios y encuestas estadísticas, ofreciendo una serie de funcionalidades para definir todos los elementos que componen estos. Esta aplicación contempla el diseño de formularios de nomenclatura cerrada y de nomenclatura abierta, así como la edición de formularios previamente creados y guardados o exportados desde la misma aplicación. Además del diseño de encuestas y edición de encuestas previamente creadas y guardadas. [\[11\]](#)

Módulo de Entrada de Datos:

El módulo de Entrada de Datos es el encargado de desarrollar el proceso de digitación de los modelos existentes en cada Centro Informante. Para esto, presenta tres aplicaciones: *Gestionar Modelos*, *Digitar Modelos* y *Validar Modelos*.

La aplicación *Gestionar Modelos* se encarga de la gestión de plantillas publicadas y modelos en el sistema. Mientras *Digitar Modelos*, se enmarca en la digitación de la información a partir de plantillas o de modelos en edición. La principal salida de esta aplicación es el modelo ya digitado, el cual es validado a través de la aplicación *Validar Modelos*. [\[12\]](#)

Módulo Generador de Reportes:

Es el encargado de la creación de informes sobre diseño en una amplia variedad de formatos que no son producidos por un sistema de información como son en PDF, HTML entre otros. Extraen datos de los archivos o de las bases de datos y crean reportes de acuerdo con muchos formatos, proporcionan más control, pueden manejar datos de cálculos y lógica compleja antes de darles la salida. [\[13\]](#)

PATDSI

DATEC con el objetivo de hacerse de herramientas propias, que le permitan cumplir con la misión que se ha trazado, ha creado el proyecto Paquete de Herramientas para la Ayuda en la Toma de Decisiones (PATDSI), el mismo cuenta con herramientas tales como un generador de reportes y un sistema de gestión estadística. [\[14\]](#)

Capítulo 1 Fundamento Teórico

CEDRUX

Es un paquete de soluciones integrales de gestión de entidades adaptable a las características económicas del país y sus empresas. Este producto pone al servicio de las entidades facilidades para la integración de las diferentes áreas productivas y departamentos administrativos. Cuenta con varios subsistemas como Planificación, Seguridad, Configuración, Capital Humano, Estructura y Composición, entre otros. [\[15\]](#)

ACAXIA

ACAXIA es el sistema de gestión integral de seguridad de CEDRUX. Implementa cinco procesos (autenticación, autorización, auditoría, administración de perfiles y administración de conexiones). [\[16\]](#)

1.4 Arquitectura de software.

No basta con identificar los módulos que podrían integrarse sin hacer un análisis previo de la arquitectura de software que debería tenerse en cuenta en la futura integración. La Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución.

Generalmente, no es necesario inventar una nueva arquitectura de software para cada sistema de información. Lo habitual es adoptar una arquitectura conocida en función de sus ventajas e inconvenientes para cada caso en concreto. Así, las arquitecturas más universales son [\[17\]](#):

- Monolítica: Donde el software se estructura en grupos funcionales muy acoplados.
- Cliente- Servidor Donde el software reparte su carga de cómputo en dos partes independientes pero sin reparto claro de funciones.
- Arquitectura de tres niveles: Especialización de la arquitectura cliente-servidor donde la carga se divide en tres partes (o capas) con un reparto claro de funciones: una capa para la presentación (interfaz de usuario), otra para el cálculo (donde se encuentra modelado el negocio) y otra para el almacenamiento (persistencia). Una capa solamente tiene relación con la siguiente.

Para el desarrollo de esta aplicación se adoptará la arquitectura Cliente/Servidor.

Capítulo 1 Fundamento Teórico

Arquitectura Cliente/Servidor.

En esta arquitectura la computadora de cada uno de los usuarios, llamada cliente, produce una demanda de información a cualquiera de las computadoras que proporcionan información, conocidas como servidores, estos últimos responden a la demanda del cliente que la produjo.

Los clientes y los servidores pueden estar conectados a una red local o una red amplia, como la que se puede implementar en una empresa o a una red mundial como lo es la internet.

Bajo este modelo cada usuario tiene la libertad de obtener la información que requiera en un momento dado proveniente de una o varias fuentes locales o distantes y de procesarla como según le convenga. Los distintos servidores también pueden intercambiar información dentro de esta arquitectura.

Entre los elementos fundamentales que componen esta Arquitectura Cliente/Servidor se deben identificar los componentes que permitan articular dicha arquitectura, considerando que toda aplicación de un sistema de información está caracterizada por tres componentes básicos: [\[18\]](#)

- Presentación/Captación de Información
- Procesos
- Almacenamiento de la Información

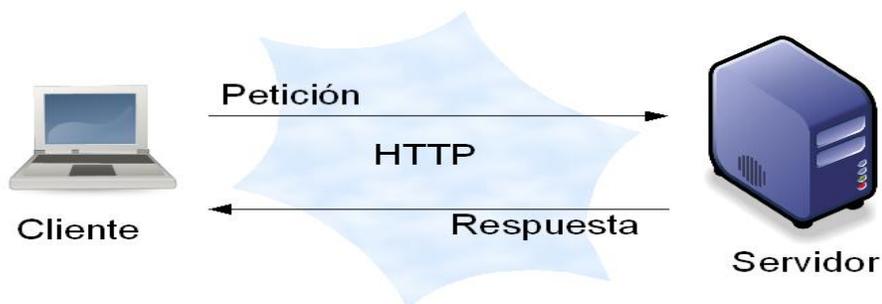


Figura 1: Arquitectura Cliente/Servidor

Capítulo 1 Fundamento Teórico

Características de la Arquitectura Cliente/Servidor.

- El Cliente y el Servidor pueden actuar como una sola entidad y como entidades separadas, realizando actividades o tareas independientes.
- Las funciones de Cliente y Servidor pueden estar en plataformas separadas, o en la misma plataforma.
- Un servidor da servicio a múltiples clientes en forma concurrente.
- Cada plataforma puede ser escalable independientemente. Los cambios realizados en las plataformas de los Clientes o de los Servidores, ya sean por actualización o por reemplazo tecnológico, se realizan de una manera transparente para el usuario final.
- La interrelación entre el hardware y el software están basados en una infraestructura poderosa, de tal forma que el acceso a los recursos de la red no muestra la complejidad de los diferentes tipos de formatos de datos y de los protocolos.
- Un sistema de servidores realiza múltiples funciones al mismo tiempo que presenta una imagen de un solo sistema a las estaciones clientes. Esto se logra combinando los recursos de cómputo que se encuentran físicamente separados en un solo sistema lógico, proporcionando de esta manera el servicio más efectivo para el usuario final.

1.4 .1 Estilos arquitectónicos

Los estilos arquitectónicos surgen en analogía al término utilizado en la arquitectura de edificios para tipificar las características distintivas de las formas básicas posibles de estructurar un software. Varios autores han definido conceptos sobre dicho término como:

“Un estilo arquitectónico encapsula decisiones esenciales sobre los elementos arquitectónicos y enfatiza restricciones importantes de los elementos y sus relaciones posibles”. [\[19\]](#)

“Conjunto de reglas de diseño que identifica clases de componentes y conectores que se pueden manejar para componer el sistema, junto con las restricciones locales o globales que determinan como se lleva a cabo la composición”. [\[20\]](#) .

Los estilos se ordenan en seis o siete clases fundamentales y unos veinte ejemplares, como máximo. A continuación la clasificación propuesta por Shaw y Garlan: [\[21\]](#)

Estilos de Flujo de datos

- Tuberías y Filtros

Capítulo 1 Fundamento Teórico

Estilos centrados en datos

- Arquitecturas de Pizarra o repositorio

Estilos de Llamada y Retorno

- Modelo – Vista – Controlador (MVC)
- Arquitectura en Capas
- Arquitectura Orientada a Objetos
- Arquitectura basada en Componentes

Estilo de Código Móvil

- Arquitectura de Máquinas Virtuales

Estilos Peer–To–Peer

- Arquitectura basada en Eventos
- Arquitecturas Orientas a Servicios (SOA)

Es común encontrar en una misma solución varios estilos combinados; cada capa o componente puede ser internamente de un estilo diferente al de la totalidad; muchos estilos se encuentran ligados a dominios específicos, o a líneas de producto particulares.

Teniendo en cuenta que se considera que la aplicación que se propone integrará módulos ya implementados que poseen una arquitectura bien definida, se impone como restricción a la nueva solución informática la adopción de los estilos y patrones presentes en estos. A continuación se describen los estilos de sumo interés para la concepción de la arquitectura del sistema en desarrollo.

Modelo Vista Controlador (MVC).

El MVC se ve frecuentemente en aplicaciones web, divide una aplicación interactiva en 3 áreas: procesamiento, salida y entrada. Utiliza las siguientes abstracciones: [\[22\]](#)

- **Modelo:** representa las estructuras de datos. Típicamente el modelo de clases contendrá funciones para consultar, insertar y actualizar información de la base de datos.
- **Vista:** Presenta el modelo con el que va a interactuar el usuario, más conocida como interfaz.
- **Controlador:** actúa como intermediario entre el Modelo, la Vista y cualquier otro recurso necesario para generar una página.

Capítulo 1 Fundamento Teórico

Arquitectura en Capas

Se caracteriza por organizar los componentes del sistema en capas con responsabilidades bien definidas y donde las capas del nivel más alto invocan los servicios de las del nivel inferior, en el sistema es posible identificar las siguientes capas: presentación, negocio, acceso a datos y datos. [\[23\]](#)

1.4.2 Patrones de Diseño.

Un patrón es una solución a un problema en un contexto, codifica conocimiento específico acumulado por la experiencia en un dominio. Un sistema bien estructurado está lleno de patrones. Cada patrón describe un problema que ocurre una y otra vez en nuestro ambiente, y luego describe el núcleo de la solución a ese problema, de tal manera que puedes usar esa solución un millón de veces más, sin hacer jamás la misma cosa dos veces.

Los patrones de diseño, al contrario de los estilos arquitectónicos, son muchos y muy variados y es casi imposible revisar todos los que existen a la hora de hacer una determinada aplicación, por eso se recomienda el uso de los patrones que estén predeterminados en cada uno de los estilos que se seleccionen para la arquitectura. En el desarrollo de aplicaciones son característico el uso de patrones específicos asociados a las diferentes capas (presentación, negocio, acceso a datos), a continuación se presentarán algunos de estos patrones fundamentalmente los utilizados en la aplicación (sobre todo asociados con el trabajo del framework Symfony que es el utilizado para el sistema en desarrollo):

Capa de Presentación

Esta capa es la que presenta el sistema al usuario, conocida como interfaz gráfica, pues en ella se representan todos los aspectos relacionados con la presentación y diseño del sistema de información. Es la encargada de establecer, mantener y gestionar la comunicación entre el cliente y la siguiente capa del sistema: la capa de negocio. [\[24\]](#)

Capa de Negocio

Es aquí donde se establecen todas las reglas que deben cumplirse en el sistema de información. Esta capa se comunica con la capa de presentación, para recibir las solicitudes ó peticiones del usuario, y presenta los resultados luego de comunicarse con la capa de acceso a datos. [\[25\]](#)

Capítulo 1 Fundamento Teórico

Capa de Acceso a Datos

Es la capa que gestiona todos los elementos de información del Sistema de Información. Se le conoce también como la capa de gestión de recursos tal como el hardware, las bases de datos, la información necesaria para que el sistema pueda funcionar de manera correcta. [26]

Capa de Datos

Es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o mas gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de acceso a datos. [27]

1.5 Herramientas, tecnologías y lenguajes para el desarrollo.

Uno de los aspectos fundamentales en la elaboración de un software es seleccionar las tecnologías y tendencias que mayores beneficios aporten. Para el desarrollo de este sistema se tuvo en cuenta las tecnologías y lenguajes empleados no solo en la construcción de los posibles módulos a integrar sino a las asumidas en la línea de producción del centro DATEC, encargada del desarrollo de la aplicación propuesta.

1.5.1 Metodologías de desarrollo de software.

El desarrollo de software no es una tarea fácil. Prueba de ello es que existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Por una parte tenemos aquellas propuestas más tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas. Estas propuestas han demostrado ser efectivas y necesarias en un gran número de proyectos sin embargo han presentado problemas en otros. Todo desarrollo de software es riesgoso y difícil de controlar, si no se lleva una metodología para guiar el proceso, lo que se obtendría como resultado clientes insatisfechos con el producto o servicio y desarrolladores aún más insatisfechos.

Se entiende por metodología de desarrollo una colección de documentación formal referente a los procesos, las políticas y los procedimientos que intervienen en el desarrollo del software. La finalidad de una metodología de desarrollo es garantizar la eficacia y eficiencia en el proceso de generación de software.

Capítulo 1 Fundamento Teórico

Existen varios procesos de desarrollo de software, cada uno con sus propias características y particularidades aunque sus objetivos o propósitos son los mismos. Ejemplos de estos procesos son: RUP, XP, FDD, Open up. Teniendo en cuenta que la línea de producción del centro encargada del desarrollo del sistema que se propone, utiliza la metodología de desarrollo Open-up, esta se asume como la principal candidata en la selección, la cual se caracteriza brevemente a continuación:

Open up

OpenUP es una metodología diseñada para el desarrollo de proyectos pequeños, pero basada en las mejores prácticas del RUP, de esta manera se podrá desarrollar artefactos ligeros apropiados utilizando el lenguaje UML, incrementando así las probabilidades de éxito en función de costo, tiempo y alcance.

OpenUP es un proceso iterativo que es: mínimo, ya que solo incluye el contenido del proceso fundamental, completo porque puede ser manifestado como proceso entero para construir un sistema y extensible pues puede ser utilizado como base para agregar o para adaptar más procesos.

Se centra en la arquitectura de forma temprana para minimizar el riesgo y organizar el desarrollo, lleva a cabo un desarrollo evolutivo para obtener retroalimentación y mejoramiento continuo. Por ser una metodología ágil tiene un enfoque centrado al cliente y con iteraciones cortas. Aporta varios beneficios como la disminución de la probabilidad de fracaso y el incrementando las probabilidades de éxito, puesto que permite detectar errores a través de un ciclo iterativo, evita la elaboración de documentación, diagramas e iteraciones innecesarias requeridas en la metodología RUP. [\[28\]](#)

1.5.2 Lenguaje Unificado de Modelado.

Para modelar el análisis y el diseño del software se utilizó el Lenguaje Unificado de Modelado (UML) ya que permite la modelación de sistemas con tecnología orientada a objetos y está implícito en la metodología utilizada.

Está consolidado como el lenguaje estándar en el análisis y diseño de sistemas de cómputo. Mediante UML es posible establecer la serie de requerimientos y estructuras necesarias para plasmar un sistema de software previo al proceso intensivo de escribir código. Se usa para

Capítulo 1 Fundamento Teórico

entender, diseñar, configurar, controlar la información sobre el sistema que se va a construir. El diagrama de clases juega un papel protagónico, este representa una parte importante del sistema, pero solo representa una vista estática, en cambio UML introduce nuevos diagramas que representan una visión dinámica del sistema. Su utilización es independiente del lenguaje de programación y de las características de los proyectos, ya que UML ha sido diseñado para modelar cualquier tipo de proyectos, tanto informáticos como de arquitectura. [\[29\]](#)

1.5.3 Herramienta CASE

Varias herramientas pudieran ser valoradas para tener en cuenta en el modelado del sistema propuesto, debido a las políticas del centro y las restricciones impuestas por la línea de desarrollo encargada de la realización de este producto se decide asumir Visual Paradigm.

Visual Paradigm 6.1

Es una potente herramienta CASE empleada para visualizar y diseñar elementos de software que proporciona a los desarrolladores una plataforma que les permite diseñar un producto con calidad de forma rápida.

Visual Paradigm es una herramienta que sirve para realizar modelado UML siguiendo el estándar UML, por lo que posee características gráficas muy cómodas que facilitan la realización de los diagramas de modelado como son:

- Facilita la interoperabilidad con otras herramientas CASE como Rational Rose.
- Se integra con diversos entornos de desarrollo como: NetBeans (de Sun), Eclipse (de IBM), JDeveloper (de Oracle), JBuilder (de Borland).
- Está disponible en varias ediciones: Enterprise, Professional, Community, Standard, Modeler y Personal.
- Genera código y realiza ingeniería inversa para diferentes lenguajes de programación como: Java, C++, CORBA IDL, PHP, XML Schema y ADA.
- En adición se genera código para C#, Visual Basic.net, Object Definition Lenguaje (ODL), Flash Action Script, Delphi, Perl y Python.

Capítulo 1 Fundamento Teórico

- Se integra con el Visio para importar imágenes del mismo para realizar los diagramas de despliegue. Además exporta e importa los diagramas en el estándar XML¹.

Visual Paradigm es la herramienta CASE que se empleará en la modelación de este sistema por su característica de ser multiplataforma, por las facilidades que brinda y por ser una herramienta que se puede utilizar legalmente. [\[30\]](#)

1.5.4 Entornos de Desarrollo (IDEs)

NetBeans 6.9.

Es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java y soporta el desarrollo de todos los tipos de aplicación Java (J2SE², web, EJB³ y aplicaciones móviles). Proporciona herramientas para la programación en el lenguaje PHP e incluye resaltado de sintaxis, auto-completado de código, *templates*,⁴ documentación de PHP integrada, entre otras funcionalidades. Provee soporte para el framework Symphony y cuenta con un módulo de *subversion*⁵ para facilitar la manipulación de las versiones del código durante la implementación. [\[31\]](#)

1.5.5 Lenguajes de programación empleados para el desarrollo.

Lenguajes del lado del servidor:

PHP:

Es un lenguaje de programación utilizado para la creación de sitio Web. PHP es un acrónimo recursivo que significa “PHP Hypertext Pre-processor”, (inicialmente se llamó Personal Home Page).

PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas Web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. PHP no necesita ser compilado para ejecutarse. Para su funcionamiento necesita tener instalado

¹ Extensible Markup Language (Lenguaje de marcado extensible)

² Java 2 Platform Standard Edition (Edición de Java)

³ Enterprise JavaBeans

⁴ Plantillas

⁵ Subversión es un sistema de control de versiones.

Capítulo 1 Fundamento Teórico

Apache o IIS con las librerías de PHP. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. [\[32\]](#)

Ventajas de PHP:

- Es un lenguaje multiplataforma.
- Completamente orientado a la web.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL
- Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables.

Lenguajes del lado del cliente:

JavaScript:

Es un lenguaje de programación del lado del cliente, ya que es el navegador el que soporta la carga de procesamiento, permitiendo el desarrollo de interfaces de usuario mejoradas y páginas web dinámicas. Es interpretado, no requiere compilación y es soportado por la mayoría de los navegadores como Internet Explorer, Netscape, Opera y Mozilla Firefox, por lo que es el lenguaje de programación del lado del cliente más utilizado.

JavaScript es un lenguaje con muchas posibilidades, permite la programación de pequeños scripts, pero también de programas más grandes, orientados a objetos, con funciones y estructuras de datos complejas. Además, pone a disposición del programador todos los elementos que forman la página web, para que este pueda acceder a ellos y modificarlos dinámicamente. [\[33\]](#)

HTML:

HTML es la abreviatura de "HyperText Mark-up Lenguaje" (Lenguaje de Marcas Hipertextuales), diseñado para estructurar textos y presentarlos en forma de hipertexto, que es

Capítulo 1 Fundamento Teórico

el formato estándar de las páginas Web. Gracias a los navegadores como: Internet Explorer, Opera, Firefox, Netscape y Safari, el HTML se ha convertido en uno de los formatos más populares y fáciles de aprender que existen para la elaboración de documentos para la Web. Este lenguaje permite la introducción de referencias a otras páginas por medio de los enlaces hipertexto.[\[34\]](#)

1.5.6 Framework para el desarrollo.

Un framework simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Además, un framework proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. Por último, un framework facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas.

Symfony:

En el desarrollo del presente trabajo de diploma se escogió a Symfony por sus características y funcionalidades, después de un estudio que realizó el grupo de arquitectura del proyecto, Symfony es un framework completo diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. Symfony está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Symfony es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas *nix (Unix, Linux, etc.) como en plataformas Windows.

Symfony se diseñó para que se ajustara a los siguientes requisitos:

- Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y *nix estándares)
- Independiente del sistema gestor de bases de datos.
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.

Capítulo 1 Fundamento Teórico

- Basado en la premisa de “convenir en vez de configurar”, en la que el desarrollador solo debe configurar aquello que no es convencional.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con las librerías de otros fabricantes.

[\[35\]](#)

Ext JS

Ext JS es una librería de Java Script para el desarrollo de aplicaciones enriquecidas para la web haciendo un uso intensivo de las tecnologías AJAX, XHTML DHTML y DOM. Originalmente fue creado como una extensión de Yahoo User Interface (YUI) otro framework similar, Ext incluye interoperabilidad con jQuery, Prototype y Script.aculo.US.

Sus principales características son:

- Alto rendimiento en ejecución debido a la optimización de código Java Script.
- Controles de usuario personalizables.
- Modelo orientado a componentes, bien diseñado y extensible.
- Posee una API intuitivo y fácil de utilizar.
- Distribuido bajo licencias Open Source y comerciales. [\[36\]](#)

Doctrine

El framework de acceso a datos Doctrine es un mapeo objeto-relacional (ORM) cuya función es gestionar el acceso a la base de datos y gestionar el modelo del sistema. Implica que el acceso y la modificación de los datos almacenados en la base de datos se realicen mediante objetos, nunca de forma explícita, permitiendo un alto nivel de abstracción y fácil portabilidad.

[\[37\]](#)

1.5.7 Sistema Gestor de Base de Datos.

PostgreSQL:

PostgreSQL posee muchas características que tradicionalmente sólo se podían ver en productos comerciales de alto calibre. Es un Sistema de Gestión de Bases de Datos Objeto-

Capítulo 1 Fundamento Teórico

Relacionales (ORDBMS) que ha sido desarrollado de varias formas desde 1977. PostgreSQL está considerado como la base de datos de código abierto más avanzada del mundo. PostgreSQL proporciona un gran número de características que normalmente sólo se encontraban en las bases de datos comerciales tales como DB2 u Oracle. La siguiente es una breve lista de algunas de esas características, a partir de PostgreSQL 7.1.x. Altamente extensible ya que soporta operadores funcionales, métodos de acceso y tipos de datos definidos por el usuario. PostgreSQL usa una arquitectura proceso-por-usuario cliente/servidor. Esta es similar al método del Apache 1.3.x para manejar procesos. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a PostgreSQL. Posee una gran escalabilidad. Es capaz de ajustarse al número de CPUs y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta. Implementa el uso de subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz. [\[38\]](#)

1.5.8 Servidor Web. Apache 2.2

Apache es un servidor web, flexible, rápido y eficiente, de código fuente abierto, continuamente actualizado y adaptado a los nuevos protocolos. Está disponible para diferentes plataformas como: FreeBSD, NetBSD, OpenBSD, GNU/Linux, Mac OS y Mac OS X Server, Netware, Solaris, Windows, entre otras. Con los diferentes módulos de apoyo que proporciona y con la API⁶ de programación de módulos, puede ser adaptado a diferentes entornos y necesidades. Gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP. [\[39\]](#)

Conclusiones

Después de un estudio detallado se puede llegar a la conclusión que para desarrollar un sistema de información para el área de la residencia y extensión universitaria de la facultad 6, se decide adoptar por la metodología de desarrollo Open Up, Visual Paradigm 6.1 como herramienta case y UML como lenguaje de modelado, NetBeans 6.9 como entorno de desarrollo, php 5 como lenguaje de programación, haciendo uso de los frameworks Symfony 1.4.6 y Ext Js. Además de contar como sistema gestor de base de datos con PostgreSQL 8.4 y como servidor web: Apache 2.2.

⁶ Interfaz de Programación de Aplicaciones.

Capítulo 2 Características del sistema

Introducción

En el presente capítulo se enuncian y describen los procesos y reglas del negocio, además de describirse los actores y trabajadores que intervienen en el mismo. Incluye la descripción de los requisitos funcionales, no funcionales del sistema, los actores que intervienen en el sistema, el diagrama de casos de uso del sistema y la descripción detallada de cada uno de los casos de uso del sistema. Además se presentará el modelado del diseño a partir de sus artefactos fundamentales.

2.1 Modelado del negocio.

El Modelado del Negocio es una visión de lo que se debe hacer para dar respuesta a las necesidades del cliente, descifrando cabalmente las exigencias que en estricto cumplimiento deben encontrar la vinculación entre el sistema y la realidad existente. Gracias al Modelo de Negocio se obtienen los requisitos más importantes del sistema y se definen los procesos, roles y responsabilidades de la organización en los modelos de casos de uso del negocio y de objetos.

Los objetivos del modelo del negocio son: [\[40\]](#)

- Comprender la estructura y la dinámica de la organización en la cual se va a implantar un sistema.
- Comprender los problemas actuales de la organización e identificar las mejoras potenciales.
- Derivar los requerimientos del sistema que va a soportar la organización.
- Lograr una comunicación efectiva entre los usuarios y el equipo de proyecto con el objetivo de llegar a un entendimiento de lo que hay que hacer, es la clave del éxito en la producción de un software.

El negocio actual se basa en gestionar parte de la información relacionada con el área de residencia y extensión universitaria de la Facultad 6. Se desea llevar el control estadístico de la guardia estudiantil, cuarterería, TSU, paradas de beca, así como de las actividades de extensión universitaria como: juegos deportivos, festival de artistas aficionados, entre otras.

Capítulo 2 Características del Sistema

Además de los datos que se generan de la resolución de las afectaciones de mantenimiento y tecnológicas presentes en el área de residencia.

2.1.1 Identificación de los actores y trabajadores del negocio.

Actores del negocio: grupo de individuos, organización, máquina o sistema de información externo al negocio y que interactúa con él. [\[41\]](#)

Tabla 1: Actores del negocio.

Actor	Descripción
Vicedecano de residencia y extensión universitaria.	Encargado de la planificación, control y desarrollo de todas las actividades referente al área residencia y extensión universitaria.
Directivos de la facultad	Directivos de la facultad, con derecho y privilegio de consultar toda la información referente a la residencia.

Trabajadores del negocio: grupo de individuos, máquina o sistema automatizado que participan directamente en los procesos que se llevan a cabo en el negocio, ya que son los que realizan las actividades y son propietarios de elementos pero no obtienen ningún beneficio con los resultados del proceso. [\[42\]](#)

Tabla 2: Trabajadores del negocio.

Trabajador	Descripción
Secretaria del vicedecano residencia y extensión universitaria.	Es la persona que asiste al Vicedecano de Residencia y tiene acceso a toda la información que se gestiona en el área de residencia.
Instructora	Encargada de organizar la cuartería en el edificio que le corresponde atender, así como controlar su cumplimiento, la limpieza de los apartamentos y el estado del edificio en general.
Profesor	Persona responsable de un grupo de estudiantes de la facultad, encargado de controlar todas las evaluaciones referentes a los mismos.
Jefe de brigada	Estudiante al frente de la brigada, responsable de

Capítulo 2 Características del Sistema

	tomar la asistencia en las guardias estudiantiles y otras actividades que se desarrollan en la beca.
--	--

Procesos fundamentales del negocio.

Tabla 3: Principales procesos del negocio

Proceso	Objetivo	Entradas / Indicadores	Salidas	Quienes participan
Cuartelería	Procesar los datos referentes al cumplimiento de la cuartelería.	Cantidad de estudiantes por año. Cantidad de asistencias por año. Total de estudiantes. Total de asistencias.	Modelo del cumplimiento de la cuartelería. Reporte del comportamiento del cumplimiento de la cuartelería.	Instructora.
Trabajo Socialmente Útil	Procesar los datos referentes al cumplimiento de el TSU	Cantidad de estudiantes por año. Cantidad de asistencias por año.	Modelo del cumplimiento del TSU. Reporte del comportamiento del cumplimiento del TSU.	Profesor. Secretaria del vicedecano.
Guardia obrera estudiantil	Procesar los datos referentes al cumplimiento de la GOE	Cantidad de estudiantes por año Cantidad de asistencias por año.	Modelo del cumplimiento de la GOE	Jefe de brigada
Parada de Beca	Procesar los datos referentes a los resultados de la parada de beca.	Cantidad de resultados obtenidos por edificios (B, R, M).	Modelo de los resultados generales obtenidos en la parada de beca.	Profesor Jefes de brigada
Actividades de	Procesar los datos referentes	Cantidad de participantes en	Modelo de la participación en las	Profesor Jefes de

Capítulo 2 Características del Sistema

extensión universitaria.	a la participación en las actividades extensionistas.	festivales de artistas aficionados. Cantidad de participantes en juegos deportivos.	actividades extensionistas. Reporte de la participación en las distintas actividades extensionistas efectuadas.	brigada Vicedecano
Afectaciones de la residencia.	Procesar la información referente a las afectaciones que existen en la residencia.	Cantidad de afectaciones tecnológicas Cantidad de afectaciones de mantenimiento Cantidad de afectaciones tecnológicas resueltas. Cantidad de afectaciones de mantenimiento resueltas	Modelo de las afectaciones en la residencia. Reporte de las afectaciones en la residencia.	Vicedecano Instructora

2.1.2 Indicadores claves del desempeño en el área de residencia y extensión universitaria.

Tabla 4: Indicadores del desempeño

Indicador	Descripción	Adecuaciones
Cumplimiento cuartelería	la Muestra el comportamiento de la cuartelería durante los meses del curso escolar.	Este indicador se obtiene mensualmente.
Resultados paradas de beca	de Muestra la evaluación obtenida por edificios en las	Este indicador puede ser calculado mensual, semestral o anual.

Capítulo 2 Características del Sistema

distintas paradas de becas efectuadas.

Afectaciones en la residencia Muestra el comportamiento de las soluciones a las afectaciones presentes en la residencia. Este indicador puede ser calculado mensual, semestral o anual.

2.1.3 Unidades de observación para cada proceso.

Tabla 5 : Unidades de Observación para cada Proceso

Procesos	Unidades de Observación
Cuartelería	Departamento de Instructoras
Trabajo Socialmente Útil.	Vicedecanato
Guardia obrera estudiantil	Vicedecanato
Parada de Beca	Vicedecanato
Actividades de extensión universitaria.	Vicedecanato
Afectaciones de la residencia.	Departamento de Instructoras

2.1.4 Diagrama de casos de uso del negocio.

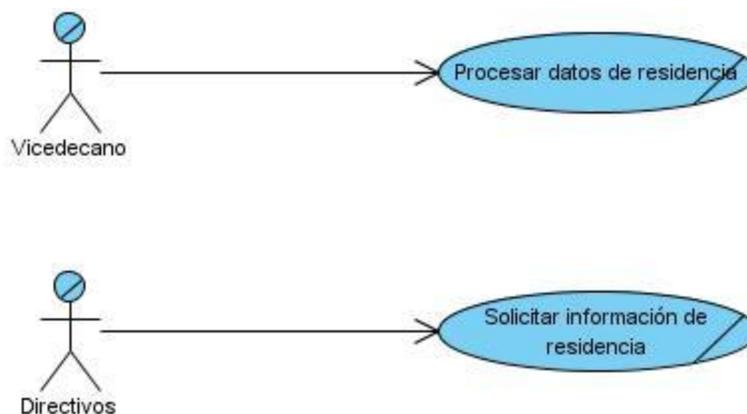


Figura 2: Modelo de Casos de Uso del negocio.

Capítulo 2 Características del Sistema

“Procesar información de la residencia”

Tabla 7: Descripción del caso de uso “Procesar información de la residencia”

Caso de Uso:	Procesar información de la residencia	
Actores:	Vicedecano de Residencia y extensión Universitaria.	
Trabajadores:	Secretaria del vicedecano Jefe de brigada Instructora Profesor	
Resumen:	El CU inicia cuando el Vicedecano gestiona los datos referentes a la residencia de la facultad.	
Precondiciones:		
Flujo Normal de Eventos		
Sección “”		
Acción del Actor	Respuesta del Negocio	
<ol style="list-style-type: none"> 1. El vicedecano planifica las actividades a realizar en la beca. 2. El vicedecano informa las actividades a realizar a toda la facultad. 	<ol style="list-style-type: none"> 3. Los profesores, jefes de brigada e instructora de edificios reciben estas informaciones y hacen posible el desarrollo de dichas actividades. 4. Los profesores, jefes de brigada e instructora de edificios informan sobre el cumplimiento de las actividades realizadas a la secretaria del vicedecano. 5. La secretaria del vicedecano 	

Capítulo 2 Características del Sistema

6. El Vicedecano recibe las informaciones concluyendo el caso de uso.	archiva los resultados obtenidos y le informa a el vicedecano.
Acción del Actor	
Respuesta del Negocio	
Poscondiciones	

2.1.6 Modelo de objetos.

El modelo de objetos del negocio, es un modelo interno a un negocio, muestra la relación entre los trabajadores y entidades del negocio dentro del flujo de trabajo de Modelamiento del negocio.

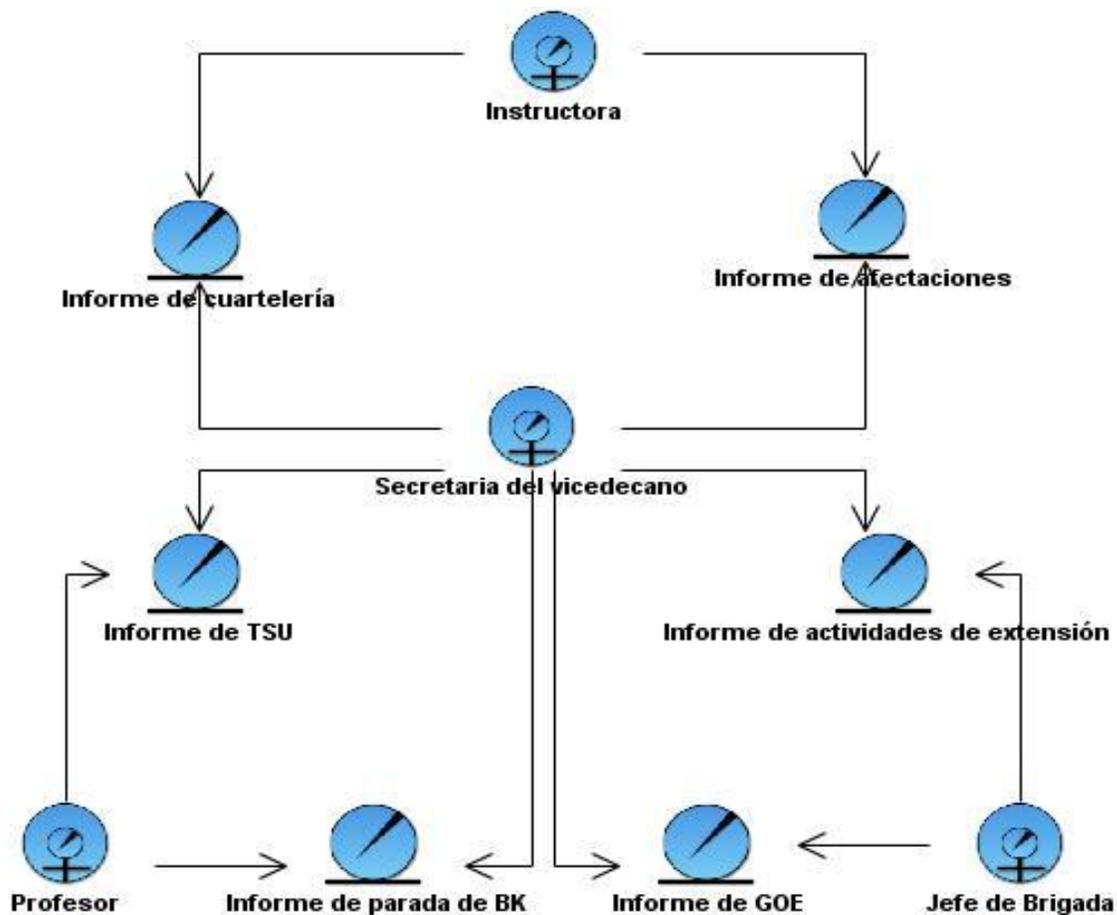


Figura 3: Modelo de Objeto del negocio.

Capítulo 2 Características del Sistema

2.1.7 Reglas del negocio.

Las reglas del negocio describen políticas que deben cumplirse y condiciones que deben ser satisfechas. [\[43\]](#)

- Diariamente, excepto los domingos, debe haber un estudiante de cuartero por cada paso de escalera de los edificios.
- Los reportes de asistencias a la cuartería se hacen diariamente a la oficina del vicedecano.
- En un curso cada brigada debe cumplir con 15 días de Trabajo Socialmente Útil.
- Los reportes de asistencia al Trabajo Socialmente Útil se hace diariamente a la oficina del vicedecano.
- Todas las noches se realiza la guardia estudiantil por brigadas, el tiempo para su realización es en dependencia de la cantidad de grupos de la facultad.
- Los reportes de asistencia a la guardia estudiantil se hacen diariamente a la oficina del vicedecano.
- Diariamente se hace el reporte de las afectaciones tecnológicas y de mantenimiento en la beca.
- El control diario de estas actividades solo se ven afectadas cuando existen actos masivos, días feriados u otras actividades que requieren de la asistencia masiva de los universitarios.

2.2 Levantamiento de requisitos.

La especificación de los requerimientos del software es la descripción de lo que debe hacer el sistema que se va a desarrollar, se divide en dos categorías: los requisitos funcionales y los requisitos no funcionales. A continuación se muestran los requisitos funcionales con los que cuenta el sistema, clasificados según su prioridad en Crítico (Esencial) y Secundario (Deseado u Opcional).

Capítulo 2 Características del Sistema

2.2.1 Requisitos funcionales.

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir, se mantienen invariables sin importar con que propiedades o cualidades se relacionen. [\[44\]](#)

En el presente trabajo se identificaron los siguientes:

RF_1 Autenticar usuario.

Prioridad: Critico.

RF_2 Adicionar usuario.

Prioridad: Critico.

RF_3 Modificar usuario.

Prioridad: Critico.

RF_4 Eliminar usuario.

Prioridad: Critico.

RF_5 Adicionar rol.

Prioridad: Critico.

RF_6 Eliminar rol.

Prioridad: Critico.

RF_7 Modificar rol.

Prioridad: Critico.

RF_8 Asignar rol a usuario.

Prioridad: Critico.

RF_9: Visualizar tablero digital.

Prioridad: Critico.

RF_10: Llenar modelo de afectaciones en la residencia.

Prioridad: Secundario.

RF_11: Llenar modelo de cuartería.

Prioridad: Secundario.

RF_12: Llenar modelo de guardia obrera estudiantil.

Prioridad: Secundario.

RF_13: Llenar modelo de Trabajo Socialmente Útil (TSU)

Prioridad: Secundario.

RF_14: Llenar modelo de Parada de Beca

Capítulo 2 Características del Sistema

Prioridad: Secundario.

RF_15 Llenar modelo de actividades de extensión universitaria.

Prioridad: Secundario.

RF_16: Mostrar reporte del comportamiento del cumplimiento de la cuarterería.

Prioridad: Secundario.

RF_17: Mostrar reporte de la participación de los becarios en las distintas actividades extensionistas efectuadas.

Prioridad: Secundario.

RF_18: Reporte del comportamiento del cumplimiento del TSU.

Prioridad: Secundario.

2.2.2 Requisitos no funcionales.

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Son esas características que posibilitan que el producto sea atractivo, usable, rápido, confiable.

[\[45\]](#)

Requisitos de Software

Para el desarrollo:

- Sistema operativo: Nova 2.1(o superior) ó Ubuntu 10.04 (o superior)
- Navegador web Mozilla Firefox 3.6.13
- Servidor web Apache 2.2 o superior, PHP 5 o superior.
- Servidor de Base de Datos PostgreSQL 8.4.

Para la explotación:

Cliente:

- PC estándar con navegador web instalado.

Servidor de Base de Datos

- Sistema Operativo: Linux Server o Windows Server
- Gestor de Base de Datos: PostgreSQL 8.4
- Cliente de Base de Datos: PgAdmin III 1.10.2

Servidor Web

- Sistema Operativo: Linux Server o Windows Server
- Aplicación Servidora: Apache 2 (o superior) con PHP5 (o superior)
- Sistema Integrado de Gestión Estadística (PATDSI - SIGE)
- Otras dependencias requeridas:

Capítulo 2 Características del Sistema

php5-pgsql (paquete de extensión PHP para PostgreSQL)

php5-xsl (paquete de extensión PHP para XSLT)

php-gd (paquete de extensión PHP para gráficos)

Requisitos de Hardware

Para el desarrollo:

- Pentium IV, con procesador Intel 1.7 GHz, o AMD similar
- 256 MB de memoria RAM o superior.
- Disco duro con capacidad de 1 GB o superior.

Para la explotación:

Cliente:

- Pentium I (superior) o AMD similar.
- 256 MB de memoria RAM o superior.
- Disco duro con capacidad de 20 MB o superior.

Servidor de Base de Datos:

- Pentium IV, con procesador Intel a 1.17 GHz o AMD similar.
- 2 GB de memoria RAM o superior.
- Disco duro con capacidad de 5 GB o superior.

Servidor Web:

- Pentium IV, con procesador Intel a 1.17 GHz o AMD similar.
- 2 GB de memoria RAM o superior.
- Disco duro con capacidad de 5 GB o superior.

Apariencia o interfaz externa

- La interfaz no debe contener muchas imágenes que demoren las respuestas al usuario.
- El diseño de la interfaz debe permitir que se obtengan los resultados en menos de 3 niveles. Debe contener elementos visibles que identifiquen cada una de sus acciones.

Capítulo 2 Características del Sistema

Seguridad

- La seguridad debe ser gestionable a través del componente de seguridad ACAXIA, el sistema de gestión integral de seguridad de CEDRUX. Implementa cinco procesos (autenticación, autorización, auditoría, administración de perfiles y administración de conexiones).

Rendimiento

- El sistema debe funcionar en redes de bajas prestaciones en velocidad y ancho de banda.
- El Sistema debe soportar conexiones simultáneas.

Portabilidad, escalabilidad, y reusabilidad

- El sistema debe ser multiplataforma (garantizado por las tecnologías que utiliza).
- Debe estar fuertemente orientado a componentes, donde se identificarán los componentes altamente reutilizables y los que son propios del negocio.

Diseño

Tanto el diseño para PHP como para Java Script se debe hacer orientado a objetos a fin de potenciar todas las ventajas que este paradigma implica.

Se utilizará la tecnología que ha venido desarrollando el centro, los frameworks de presentación, lógica de negocio y acceso a datos: Ext JS, Symfony y Doctrine respectivamente.

2.3 Propuesta de Arquitectura.

La arquitectura concebida para el sistema de información a desarrollar está compuesta por cuatro componentes fundamentales los cuales son: ACAXIA, módulo de entrada de datos, módulo generador de reportes y por último el tablero digital. Estos módulos responden a la captura y actualización de información mediante el formulario que ofrece soporte físico, la

Capítulo 2 Características del Sistema

validación, el almacenamiento en la base de datos, el procesamiento de los datos para la generación de reportes estadísticos y la visualización a través de gráficas en el tablero digital del comportamiento de los indicadores definidos.

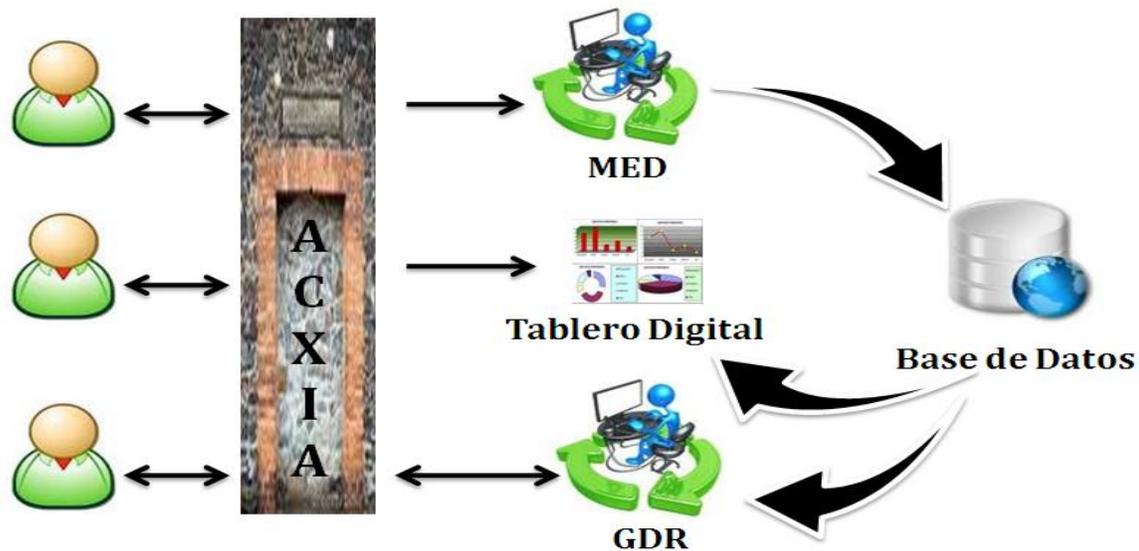


Figura 4: Arquitectura del Sistema

ACAXIA: Ofrece seguridad a los sistemas que accedan a él, incorpora cinco procesos importantes: la autenticación, autorización, auditoría, la administración de conexiones y de perfiles.

Módulo Entrada de Datos (MED): Su función principal es garantizar la actualización, captura, validación y almacenamiento en la base de datos. Ejecuta el formulario digital generado por MDF desde la base de datos o importándolo desde un XML previamente.

Módulo Generador de Reportes (MGR): Su función principal es garantizar la elaboración, generación y gestión de reportes estadísticos.

Tablero Digital o Dashboard: Es una página desarrollada en base a tecnología web mediante la cual se despliega en tiempo real información de la empresa extraída de varias fuentes o bases de datos. Su característica de tiempo real otorga a los usuarios un conocimiento completo sobre la marcha de la empresa y permite hacer análisis instantáneos e inteligencia de negocios.

Capítulo 2 Características del Sistema

2.4 Modelado del sistema.

2.4.1 Actores del sistema.

Descripción de los actores del sistema.

Tabla 8: Actores del sistema.

Actor	Descripción
Administrador	Posee acceso a ingresar, modificar y eliminar informaciones del sistema.
Usuarios.	Personas que van a tener algún rol en el sistema, es decir van a poder trabajar con la aplicación.
Secretaria del Vicedecano	Es la encargada de controlar la información de los estudiantes en la Residencia, también controlar las evaluaciones de todas las actividades extracurriculares, por lo que es la persona con acceso a todas las funcionalidades que ofrece el sistema
ACAXIA	Es el sistema encargado de implementar cinco procesos (autenticación, autorización, auditoría, administración de Perfiles y administración de Conexiones), necesarios para la seguridad del sistema.

Capítulo 2 Características del Sistema

SIGE-MGR	El Módulo Generador de Reportes extrae datos de los archivos o de las bases de datos y crean reportes de acuerdo con muchos formatos, proporcionan más control, pueden manejar datos de cálculos y lógica compleja antes de darles la salida.
SIGE- MED	El Módulo de Entrada de Datos es el encargado de desarrollar el proceso de digitación de los modelos existentes. Para esto, presenta tres aplicaciones: <i>Gestionar Modelos, Digital Modelos y Validar Modelos.</i>

2.4.2 Diagrama de casos de uso del sistema.

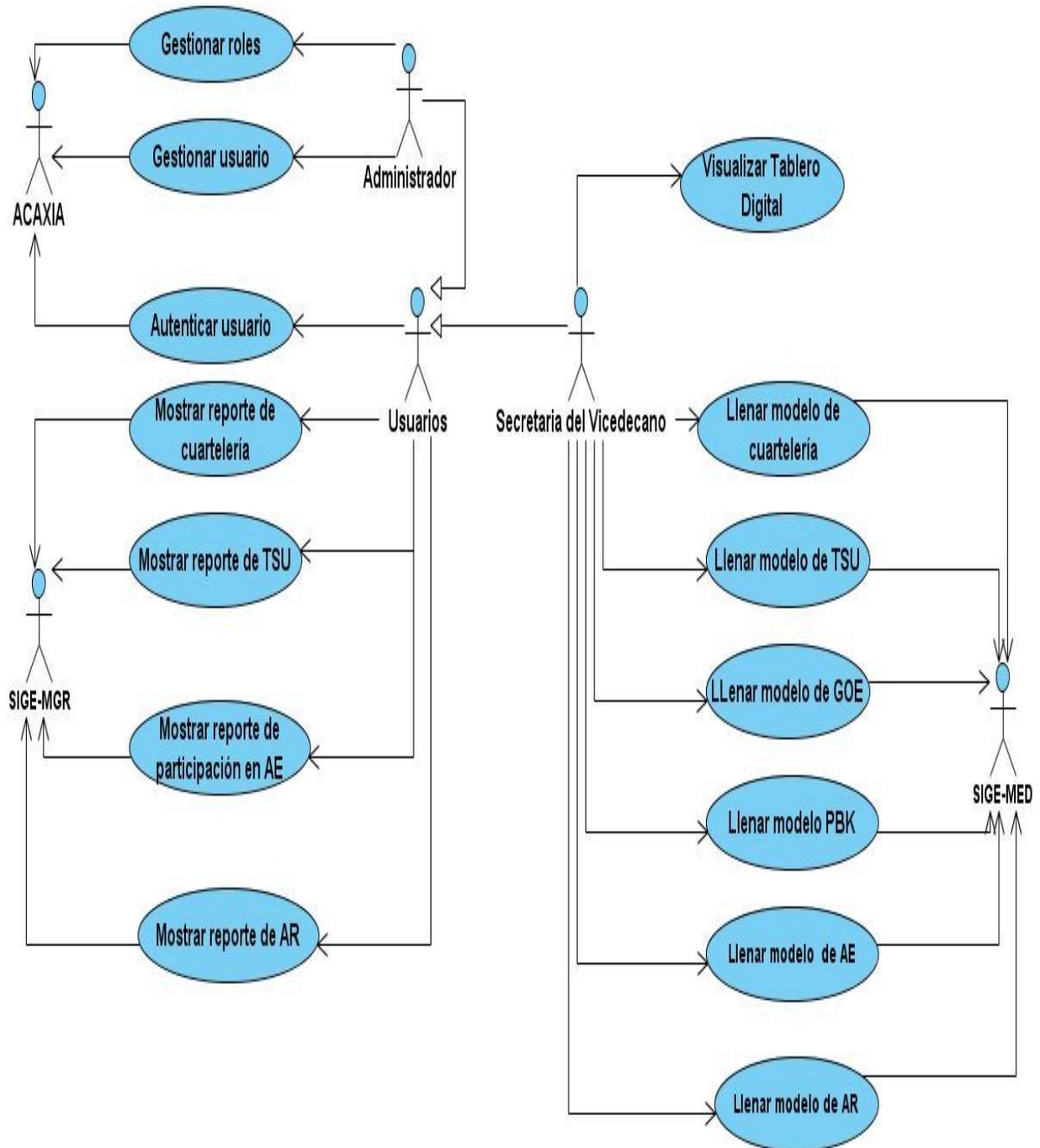


Figura 5: Diagrama de Caso de Uso del Sistema

Capítulo 2 Características del Sistema

2.4.3 Descripción de los casos de uso del sistema.

A continuación se describen dos de los casos de uso más significativos para la arquitectura del sistema, cuyos flujos de eventos responden a la misma lógica del resto de los casos de uso de su tipo. Por ejemplo, todos los casos de uso “Mostrar reportes” siguen el mismo flujo de eventos. De igual forma sucede con los casos de uso “Llenar modelo.”

“Mostrar reporte de cuartería”

Tabla 9: Descripción del CUS Mostrar reporte de cuartería

Caso de Uso:	Mostrar reporte de cuartería	
Actores:	Usuarios, Secretaria del vicedecano.	
Resumen:	El caso de uso inicia cuando el usuario o la secretaria del vicedecano desea visualizar un reporte de la cuartería, culmina una vez el sistema muestra la vista previa del reporte.	
Precondiciones:	El usuario debe estar autenticado en el sistema	
Flujo Normal de Eventos		
Sección “”		
Acción del Actor	Respuesta del Sistema	
1. El caso de uso inicia cuando el usuario desea visualizar un reporte.	1.1 El sistema muestra los reportes que se han creado.	
2. El usuario selecciona un reporte. 2.2 El usuario selecciona la opción Siguiente.	2.1 El sistema muestra una interfaz para el filtrado del reporte. 2.3 El sistema muestra una vista previa del reporte seleccionado por el Usuario y finaliza así el caso de uso.	
Flujos Alternos		
Acción del Actor	Respuesta del Sistema	
1. El usuario selecciona la opción Cancelar.	1.1. El sistema muestra un mensaje de confirmación para la acción solicitada.	
1.2 El usuario confirma que desea cerrar la aplicación.	1.3 El sistema cierra la aplicación volviendo al área de trabajo.	
Poscondiciones		

Capítulo 2 Características del Sistema

“Llenar modelo de cuartería”

Tabla 10: Descripción del CUS Llenar modelo de cuartería

Caso de Uso:	Llenar modelo de cuartería
Actores:	Secretaria del vicedecano.
Resumen:	El caso de uso inicia cuando la secretaria del vicedecano desea llenar modelo de cuartería, culmina una vez el sistema guarda los datos en la base de datos.
Precondiciones:	La secretaria del vicedecano debe estar autenticada en el sistema
Flujo Normal de Eventos	
Sección “”	
Acción del Actor	Respuesta del Sistema
1. La secretaria del vicedecano selecciona el modelo a realizar	1.1 Abre el archivo del modelo seleccionado. 1.2 Carga la información de archivo en formato XML 1.3 Crea los componentes de formulario correspondientes. 1.4 Crear formulario adicionando los componentes. 1.5 Muestra el formulario.
2. La secretaria del vicedecano digita los datos estadísticos. 2.3 La secretaria del vicedecano selecciona la opción Siguiente.	2.1 Guarda los datos estadísticos introducidos 2.2 Concluye con la entrada de datos 2.4 El sistema muestra una vista previa del modelo seleccionado y finaliza así el caso de uso.
3. La secretaria del vicedecano activa el estado listo para validar.	
Poscondiciones	

Capítulo 2 Características del Sistema

“Visualizar tablero digital”

Tabla 11: Descripción del CUS Visualizar Tablero Digital

Caso de Uso:	Visualizar tablero digital.
Actores:	Secretaria del vicedecano.
Resumen:	El caso de uso inicia cuando la secretaria del vicedecano desea visualizar los resultados de los indicadores definidos en el tablero digital, culmina cuando esta cierra el tablero visualizado.
Precondiciones:	La secretaria del vicedecano debe estar autenticada en el sistema
Flujo Normal de Eventos	
Sección “”	
Acción del Actor	Respuesta del Sistema
1. La secretaria del vicedecano selecciona la opción Tablero Digital.	1.1 Se muestra el tablero digital.
2. La secretaria del vicedecano escoge el indicador a visualizar.	2.1 El sistema muestra las graficas referentes al indicador seleccionado.
3. La secretaria del vicedecano escoge la opción salir.	
Poscondiciones	

2.5 Diagrama de Clases del Diseño

Los diagramas de clases de diseño describen gráficamente las especificaciones de las clases del software y contienen las clases, atributos, métodos, navegabilidad y dependencias existentes entre ellas.

En el siguiente diagrama de clases del diseño se muestran las clases, métodos y atributos correspondientes al Dashboard a desarrollar, donde del lado del cliente los principales elementos corresponden a las librerías “ext-base.js”, “ext-alb.css”, “ext-all.js”, “Dasboard.js”. Del lado del servidor se encuentra el módulo (visto con el estereotipo “Symfony module”). Se puede observar la relación que existe de flujo de información entre el componente caso de uso del lado del cliente y el componente módulo del lado del servidor, donde las solicitudes son

Capítulo 2 Características del Sistema

AJAX, devolviendo las respuestas específicamente en formato JSON, debido a que el sistema a desarrollar reutiliza aplicaciones existentes en el centro DATEC y en su mayoría utilizan AJAX como tecnología.

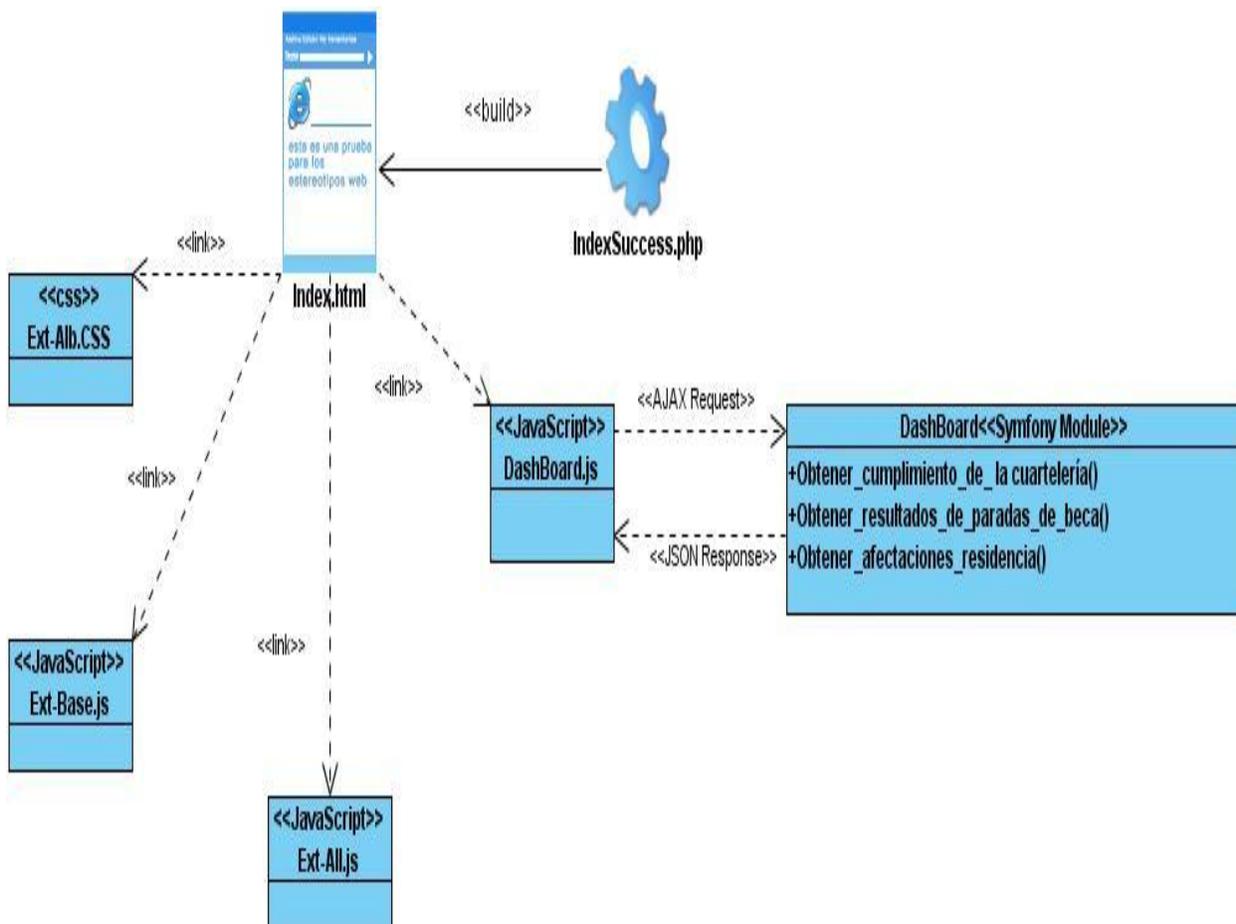


Figura 6: Diagrama de Clases del Diseño

Conclusiones

Con el desarrollo de este capítulo se realizó un profundo estudio de los procesos que tienen lugar en el área de residencia estudiantil de la facultad 6, identificándose así los indicadores claves en esta área, los actores y casos de uso del negocio, así como trabajadores y casos de uso del sistema.

Se identificaron además 18 requisitos funcionales, de ellos 9 críticos y 9 secundarios, los que permitieron llegar a un acuerdo entre desarrollador y clientes, dando una mayor comprensión de lo que debe hacer el sistema a desarrollar

.Capítulo 3 Implementación del Sistema.

Introducción

En el presente capítulo se describe la implementación del sistema a partir de los requerimientos identificados. Se muestran el diagrama de componentes del sistema, así como el diagrama de despliegue reflejando la distribución física de los nodos que intervienen en el desarrollo del mismo.

3.1 Modelo de Implementación

El Modelo de Implementación hace más entendible el trabajo a los desarrolladores, pues describe cómo los elementos del modelo de diseño, se implementan en términos de componentes, ficheros de código fuente, ejecutables, entre otros. Describe también cómo se organizan los componentes de acuerdo a los mecanismos de estructuración y modularización disponibles en el entorno de implementación y lenguaje o lenguajes de implementación empleados, y cómo dependen los componentes unos de otros. [\[47\]](#)

3.1.1 Diagrama de componentes

Los diagramas de componentes muestran como el sistema está dividido en componentes y las dependencias entre ellos. Proveen una vista arquitectónica de alto nivel del sistema, ayudando a los desarrolladores a visualizar el camino de la implementación, permitiendo tomar decisiones respecto a las tareas de implementación. 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, paquetes, bibliotecas cargadas dinámicamente, etc. [\[48\]](#)

Capítulo 3 Implementación del Sistema

Capa de Presentación

La capa de presentación está compuesta por los componentes SIGE, GRD, ACAXIA y el Dashboard que contiene las librerías Ext Js, HTML, PHP y CCS , el explorador Mozilla Firefox, el sistema operativo Linux.

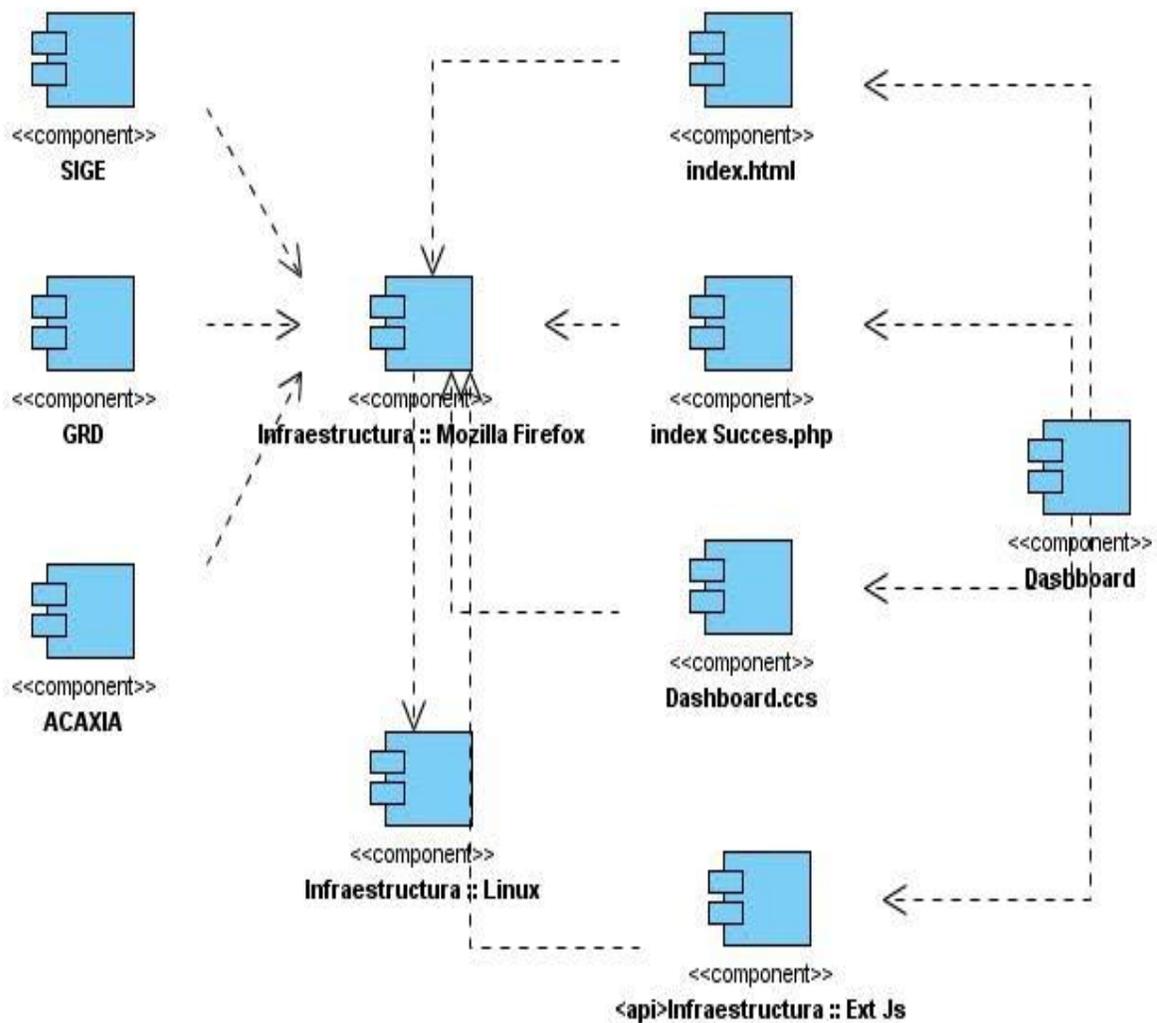


Figura 7: Componentes de la Capa de Presentación.

Capítulo 3 Implementación del Sistema

Capa de Negocio

La capa de negocio está compuesta por los componentes de negocio: SIGE, GRD, ACAXIA, el Dashboard y las librerías Symfony. Se hace uso del intérprete de PHP integrado en el servidor Apache con la extensión php5_xsl para trabajo con XSL, el sistema operativo para el servidor es Linux.

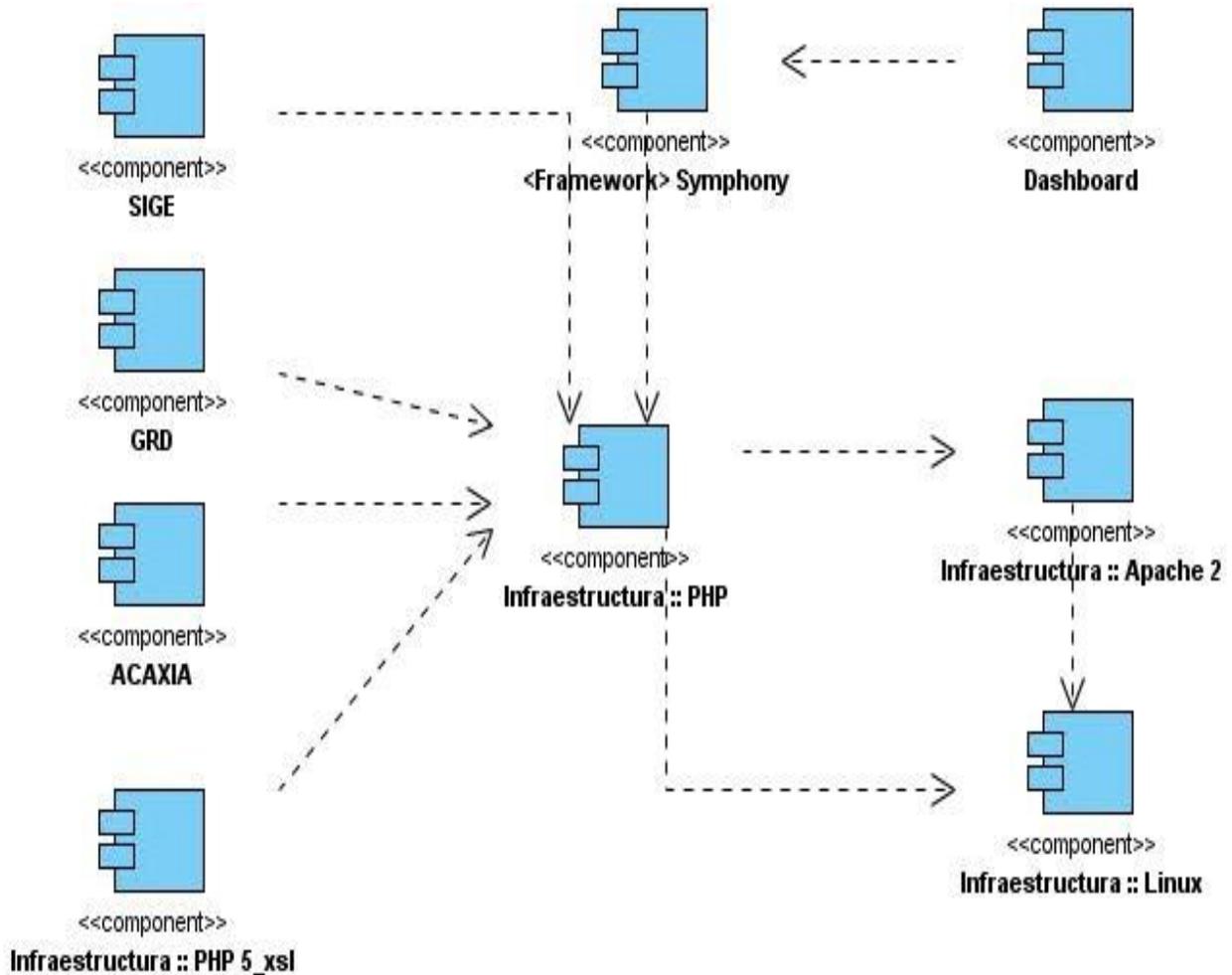


Figura 8: Componentes de la Capa de Negocio

Capítulo 3 Implementación del Sistema

Capa de Acceso a Datos

La capa de acceso a datos está compuesta por los componentes comunes del framework de acceso a datos Doctrine. Se utiliza el intérprete de PHP integrado al servidor Apache con la extensión php5_pgsql para acceder al servidor de base de datos PostgreSQL.

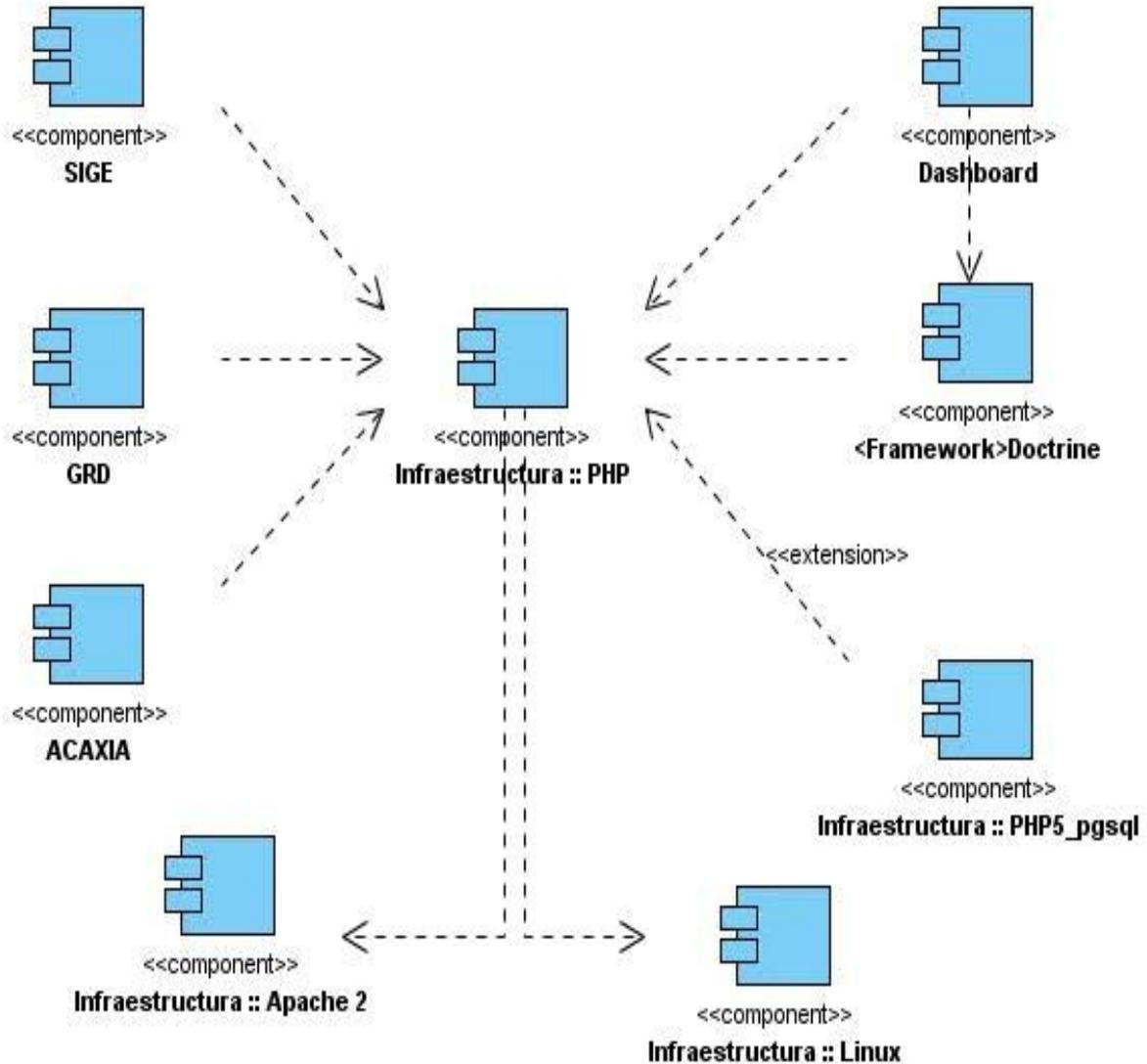


Figura 9: Componentes de la Capa de Acceso a Datos

Capítulo 3 Implementación del Sistema

Capa de Datos

La capa de datos esta compuesta por la base de datos SIGE. Para dicha base de datos se cuenta con un servidor con PostgreSQL y sistema operativo Linux.

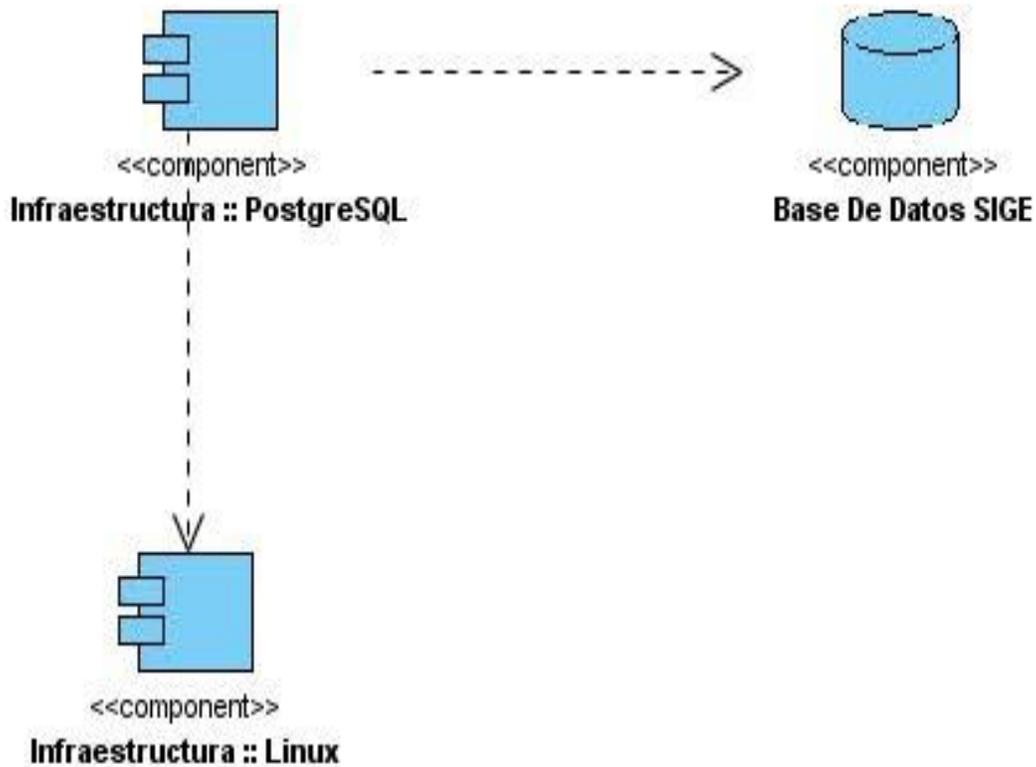


Figura 10: Componentes de la Capa de Datos

Capítulo 3 Implementación del Sistema

Diagrama de componente del sistema.

El diagrama de componente del sistema es la integración de las capas, destacándose las relaciones de dependencia que se establece entre el nivel superior y el inferior.

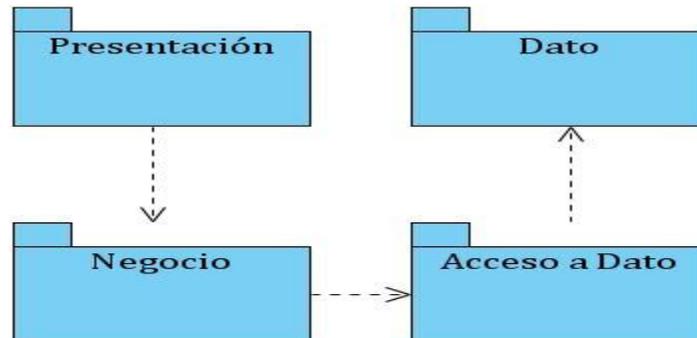


Figura 11: Diagrama de componente del sistema

3.1.2 Diagrama de Despliegue

Los Diagramas de Despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria. Los nodos poseen relaciones que representan medios de comunicación entre ellos, tales como HTTP, USB, TCP/IP, etc. La funcionalidad de un nodo se define por los componentes que se distribuyen sobre el mismo.

[49]

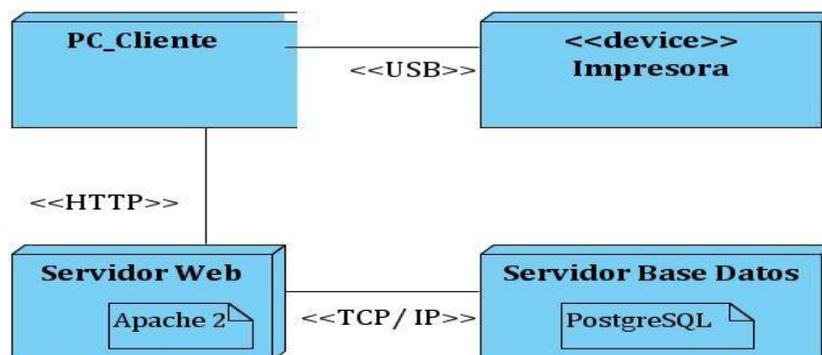


Figura 12: Diagrama de Despliegue

Capítulo 3 Implementación del Sistema

Conclusiones

En este capítulo se desarrolló la fase de Implementación, ilustrándose con los diagramas generados la relación entre los principales componentes del sistema y la distribución de los mismos. Se obtuvo como resultado final el desarrollo del sistema de gestión de la información para el área de residencia y extensión universitaria de la Facultad 6.

Capítulo 4: Pruebas del sistema.

Introducción

En el presente capítulo se aborda el tema relacionado con las pruebas realizadas al sistema. Se muestran los resultados de las mismas y el análisis de dichos resultados.

4.1 Pruebas de Software.

El desarrollo de sistemas de software implica una serie de actividades de producción en las que las posibilidades de que aparezca el fallo humano son enormes. Los errores pueden empezar a darse desde el primer momento del proceso en el que los objetivos pueden estar especificados de forma errónea o imperfecta así como en posteriores pasos de diseño y desarrollo. Debido a la imposibilidad humana de trabajar y comunicarse de forma perfecta el desarrollo del software ha de ir acompañado de una actividad que garantice la calidad.

Las pruebas del 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. Existen diferentes niveles, estrategias, técnicas y métodos de pruebas. Para realizar estas pruebas se diseñan casos de pruebas. El diseño de casos de prueba debe tener en cuenta las características propias del software. Los niveles se aplican para diferentes objetivos, entre ellos están las pruebas de desarrollador, independiente, de Unidad, de Integración, de sistema y de aceptación. Las técnicas de pruebas son varias, pueden ser de funcionalidad, usabilidad, confiabilidad, rendimiento y soporte. Los métodos pueden ser pruebas de caja negra y pruebas de caja blanca. En lo adelante se abordará todo lo relacionado con las pruebas realizadas al sistema.

4.1.1 Niveles de pruebas seleccionados.

En la etapa del ciclo de vida en que se encuentra el software y las características de reutilización que atribuyen a los componentes que lo conforman, ya se ha transitado por los primeros niveles de prueba como son: pruebas de desarrollador, pruebas independientes y pruebas de unidad, por lo que los niveles de prueba seleccionados para llevar a cabo el proceso de liberación del sistema de información para el área de residencia y extensión universitaria de la Facultad 6 son:

Capítulo 4 Pruebas del sistema

- Pruebas de integración
- Pruebas de sistema.

Las pruebas al sistema a pesar de ser un nivel superior al de Integración, se ve con este último estrechamente relacionado. Dado que las pruebas al sistema básicamente son similares a las pruebas de integración pero con un alcance mucho más amplio, a medida que se integran los módulos, se va probando el sistema como un todo, por lo que no se hace distinción específica de las técnicas en uno u otro nivel, sino que se ven de manera complementada.

Las pruebas de integración y sistema se basaron en la verificación y validación de las funcionalidades del módulo Dashboard, para comprobar que éste, como un componente integrado y combinado ejecutaba correctamente su función.

4.1.2 Técnicas de pruebas seleccionadas.

Para estos niveles de prueba la técnica de prueba seleccionada es: la prueba de funcionalidad, mediante las cuales se trata de garantizar que el software trabaje funcionalmente como se especificó inicialmente por el cliente.

Esta técnica comprobó que se permitiera la navegación por toda la aplicación a medida que se probara la correspondencia entre las funciones implementadas con los requisitos del cliente. Para ello se diseñaron casos de pruebas basados en los casos de uso y a su vez en los requerimientos funcionales, comparando cada funcionalidad implementada con la descrita, para verificar hasta qué punto cumplía con las necesidades del cliente.

4.1.3 Métodos de pruebas seleccionados.

Existen métodos de prueba independientemente de la técnica que se utilice o el nivel en que se enmarquen estas técnicas. El método seleccionado para llevar a cabo el proceso de liberación del sistema de información para el área de residencia y extensión universitaria de la Facultad 6 es: El método de caja negra o funcional, en el cual se realizaron pruebas sobre la interfaz del Dashboard o Tablero Digital, para comprobar la funcionalidad que debe realizar.

Capítulo 4 Pruebas del sistema

Mostrar Tablero Digital

Condiciones de ejecución:

El usuario debe tener los permisos necesarios para realizar esta operación.

Tabla 12: Caso de prueba "Visualizar Tablero Digital"

Nombre del requisito	Descripción general	Escenarios de prueba	Flujo de escenario
Visualizar tablero digital.	Permitir que el usuario del sistema pueda visualizar el Dashboard.	Visualizar gráficos del Dashboard correctamente.	<ul style="list-style-type: none">- Clic en botón inicio y se selecciona la opción Dashboard, luego se selecciona la opción Visualizar Dashboard.- El sistema muestra la interfaz del Dashboard.- Se escoge el indicador a graficar.- El sistema muestra la grafica del indicador seleccionado.

Capítulo 4 Pruebas del sistema

Tabla 13: Escenario de Prueba: "Visualizar Gráficos del Tablero Digital"

No	Nombre del campo	Clasificación	Puede ser nulo	Descripción
1	Cuartelería	bool	No	El campo acepta un valor de la lista.
2	Resultados de Paradas de Beca	bool	No	El campo acepta un valor de la lista.
3	Afectaciones de la residencia	bool	No	El campo acepta un valor de la lista.

Conclusiones

Tras el flujo de prueba aplicado al sistema desarrollado se obtuvieron resultados satisfactorios, logrando gran aceptación del mismo por los usuarios finales y aumentándose así la calidad final de la solución.

Conclusiones

Conclusiones

Con el desarrollo del sistema de información para el área de residencia y extensión universitaria de la Facultad 6 se llegó a las siguientes conclusiones:

- Se definió toda la problemática de la situación, lo que permitió centrarse en el problema que se planteaba, así como su objeto de estudio y campo de acción, aspectos fundamentales a la hora de limitar el trabajo a realizar y hasta qué punto consumarlo.
- Se elaboró la fundamentación teórica, donde se realizó el estudio de los sistemas de información, permitiendo reformar la visión que se poseía acerca de los mismos en Cuba y el mundo, aseverando la no existencia de un módulo capaz de aportar las funcionalidades brindadas por estos requeridas en el área de residencia y extensión universitaria de la facultad 6.
- A partir de la descripción del problema se definieron y describieron los procesos del negocio, así como su modelación, dando una vista global del sistema de información para el área de residencia y extensión universitaria de la Facultad 6.
- Posteriormente de realizarse la definición de los requisitos funcionales y no funcionales del sistema, se obtuvo una perspectiva más completa de las funcionalidades con que debía cumplir el sistema de información, así como de las cualidades o condiciones que se comprometían a estar presentes en la realización del mismo.
- La generación de artefactos relacionados con el flujo de análisis y diseño, teniendo en cuenta la arquitectura que el marco de trabajo establece, permitió obtener una mayor comprensión de la aplicación y definió los principios que guiaron la implementación del sistema.
- Como conclusión general, puede afirmarse que se le dieron cumplimiento a todos los objetivos planteados al inicio del trabajo, incluyendo la implementación del sistema de información para el área de residencia y extensión universitaria de la Facultad 6, se verificó además la validez de la idea a defender materializada en la solución obtenida.

Recomendaciones

Recomendaciones

- Definir nuevos indicadores claves que permitan al cliente facilitar la toma de decisiones en esta área de la facultad.
- Integrar esta solución con los otros sistemas de gestión de información de la facultad de manera que se cuente con un solo sistema que centralice toda información y facilite la toma de decisiones basándose en todas las áreas de la misma.
- Proponer el uso de esta aplicación en el resto de las facultades estandarizándose los procesos que automatiza en toda la universidad.

Referencias Bibliográficas

Referencias Bibliográficas.

1. Theoretical analysis of information systems. [Online] [Cited: 3 10, 2011.] <http://www.getcited.org>.
2. definicion. [Online] [Cited: 3 16, 2011.] <http://definicion.de/sistema-de-informacion>.
3. www.siiu.udg.mx. [Online] [Cited: 2 16, 2011.] http://www.siiu.udg.mx/html/pronad/doctos/introduccion_siaa.pdf.
4. [Online] [Cited: 2 10, 2011.] <http://sistemasdeinformacion-kaal.blogspot.com/2009/06/actividades-basicas.html>
5. www.sirac.info. [Online] [Cited: 2 20, 2011.] <http://www.sirac.info/Curtiembres/html/indicadores.asp>.
6. Integración de PATDSI – SIGE al marco ACAXIA.
7. Manual de Usuario PATDSI. Módulo Gestión de Configuración.
8. Manual de Usuario PATDSI. Módulo Diseñador de Formularios.
9. Manual de Usuario PATDSI. Módulo de Entrada de Datos.
10. Manual de Usuario PATDSI. Generador Dinámico de Reportes.
11. [informaticahabana.cu](http://www.informaticahabana.cu). [Online] [Cited: 3 15, 2011.] <http://www.informaticahabana.cu/node/2887>.
12. [javerianacali.edu.co](http://cic.javerianacali.edu.co). [Online] [Cited: 3 13, 2011.] <http://cic.javerianacali.edu.co/wiki/lib/exe/fetch.php?media=materias:introarq.pdf>.
13. **Casanovas, Josep.** Desarrolloweb. [Online] [Cited: 3 10, 2011.] <http://www.desarrolloweb.com/articulos/1622.php>.
14. **Wolf, Dewayne, Perry, E. y L., Alexander.** Software Engineering Notes. 2002.
15. **Shaw, Mary y Garlan, David.** Software Architecture: Perspectives on an Emerging Discipline. 2002.
16. neleste. [Online] [Cited: 3 14, 2011.] <http://www.neleste.com/modelo-vista-controlador>.

Referencias Bibliográficas

17. profitek. [Online] [Cited: 3 12, 2011.] <http://www.profittek.com.co/productos/signo.htm> 14. Profitek.
18. pbworks. [Online] [Cited: 3 13, 2011.] <http://isg3.pbworks.com/w/page/7624479/Patrones-Arquitect%C3%B3nicos>.
19. Newcomlab. [Online] [Cited: 3 14, 2011.] <http://www.Newcomlab.com>.
20. slideshare. [Online] [Cited: 2 20, 2011.] <http://www.slideshare.net/samith/metodologia-upen-up-3439131>.
21. **Cueva Lovelle, Juan manuel.** gidis. [Online] [Cited: 2 10, 2011.] <http://gidis.ing.unlpam.edu.ar/downloads/pdfs/IntroduccionUML.PDF>.
22. visual-paradigm. [Online] [Cited: 2 10, 2011.] <http://www.visual-paradigm.com/>.
23. elcodigok. [Online] [Cited: 3 2, 2011.] <http://www.elcodigok.com.ar/2010/09/7-caracteristicas-de-netbeans-6-9-1-integrado-a-php/>.
24. maestrosdelweb. [Online] [Cited: 2 9, 2011.] <http://www.maestrosdelweb.com/editorial/phpintro/>.
25. **Potencier, Fabien.** librosweb. [Online] [Cited: 2 17, 2011.] <http://www.librosweb.es/symfony/>.
26. desarrollosweb. [Online] [Cited: 2 22, 2011.] <http://www.desarrollosweb.net/tag/extjs/>.
27. doctrine-projec. [Online] [Cited: 2 22, 2011.] <http://www.doctrine-project.org/>.
28. usabilidadweb. [Online] [Cited: 2 23, 2011.] <http://www.usabilidadweb.com.ar/postgre.php>.
29. modulehosting. [Online] [Cited: 2 24, 2011.] <http://www.modulehosting.com/apache.html>.
30. eva. [Online] [Cited: 2 25, 2011.] http://eva.uci.cu/file.php/102/Curso_2010-2011/Clases/Semana_04/Conferencia_6/Materiales_complementarios/Fase_de_Inicio._Disciplina_de_Modelamiento_del_Negocio.pdf.
31. eva. [Online] [Cited: 2 23, 2011.] http://eva.uci.cu/file.php/102/Curso_2010-2011/Clases/Semana_06/Conf_7/Materiales_complementarios/Introduccion_a_la_Disciplina_de_Requisitos.pdf.

Referencias Bibliográficas

32. definicion.de. [Online] [Cited: 2 25, 2011.] <http://definicion.de/modelo-de-datos/>.
33. ilustrados. [Online] [Cited: 2 26, 2011.] <http://www.ilustrados.com/.../tesis-multimedia-derechos-ninos-070208.doc>.
34. dsi. [Online] [Cited: 2 27, 2011.] <http://www.dsi.uclm.es/asignaturas/42530/pdf/M2tema12.pdf>.

Bibliografía

1. Casanovas, Josep. Desarrolloweb. [Online] [Cited: 3 10, 2011.]
<http://www.desarrolloweb.com/articulos/1622.php>.
2. Cueva Lovelle, Juan manuel. gidis. [Online] [Cited: 2 10, 2011.]
<http://gidis.ing.unlpam.edu.ar/downloads/pdfs/IntroduccionUML.PDF>.
3. definicion. [Online] [Cited: 3 16, 2011.] <http://definicion.de/sistema-de-informacion>.
4. definicion.de. [Online] [Cited: 2 25, 2011.] <http://definicion.de/modelo-de-datos/>.
5. desarrollosweb. [Online] [Cited: 2 22, 2011.] <http://www.desarrollosweb.net/tag/extjs/>.
6. doctrine-projec. [Online] [Cited: 2 22, 2011.] <http://www.doctrine-project.org/>.
7. dsi. [Online] [Cited: 2 27, 2011.]
<http://www.dsi.uclm.es/asignaturas/42530/pdf/M2tema12.pdf>.
8. elcodigok. [Online] [Cited: 3 2, 2011.] <http://www.elcodigok.com.ar/2010/09/7-caracteristicas-de-netbeans-6-9-1-integrado-a-php/>.
9. eva. [Online] [Cited: 2 25, 2011.] http://eva.uci.cu/file.php/102/Curso_2010-2011/Clases/Semana_04/Conferencia_6/Materiales_complementarios/Fase_de_Inicio._Disciplina_de_Modelamiento_del_Negocio.pdf.
10. eva. [Online] [Cited: 2 23, 2011.] http://eva.uci.cu/file.php/102/Curso_2010-2011/Clases/Semana_06/Conf_7/Materiales_complementarios/Introduccion_a_la_Disciplina_de_Requisitos.pdf.
11. ilustrados. [Online] [Cited: 2 26, 2011.] <http://www.ilustrados.com/.../tesis-multimedia-derechos-ninos-070208.doc>.
12. informaticahabana. [Online] [Cited: 3 15, 2011.]
<http://www.informaticahabana.cu/node/2887>.
13. Integración de PATDSI – SIGE al marco ACAXIA.
14. javerianacali. [Online] [Cited: 3 13, 2011.]
<http://cic.javerianacali.edu.co/wiki/lib/exe/fetch.php?media=materias:introarq.pdf>.

Bibliografía

15. maestrosdelweb. [Online] [Cited: 2 9, 2011.]
16. <http://www.maestrosdelweb.com/editorial/phpintro/>.
17. Manual de Usuario PATDSI. Generador Dinámico de Reportes.
18. Manual de Usuario PATDSI. Módulo de Entrada de Datos.
19. Manual de Usuario PATDSI. Módulo Diseñador de Formularios.
20. Manual de Usuario PATDSI. Módulo Gestión de Configuración.
21. modulehosting. [Online] [Cited: 2 24, 2011.] <http://www.modulehosting.com/apache.html>.
22. neleste. [Online] [Cited: 3 14, 2011.] <http://www.neleste.com/modelo-vista-controlador>.
23. Newcomlab. [Online] [Cited: 3 14, 2011.] <http://www.Newcomlab.com>.
24. pbworks. [Online] [Cited: 3 13, 2011.]
<http://isg3.pbworks.com/w/page/7624479/Patrones-Arquitect%C3%B3nicos>.
25. [Online] [Cited: 2 10, 2011.] <http://sistemasdeinformacion-kal.blogspot.com/2009/06/actividades-basicas.html>
26. Potencier, Fabien. librosweb. [Online] [Cited: 2 17, 2011.]
<http://www.librosweb.es/symfony/>.
27. profitek. [Online] [Cited: 3 12, 2011.] <http://www.profittek.com.co/productos/signo.htm> 14.
Profitek.
28. Shaw, Mary y Garlan, David. 2002. Software Architecture: Perspectives on an Emerging Discipline. 2002.
29. slideshare. [Online] [Cited: 2 20, 2011.] <http://www.slideshare.net/samith/metodologia-open-up-3439131>.
30. Theoretical analysis of information systems. [Online] [Cited: 3 10, 2011.]
<http://www.getcited.org>.
31. usabilidadweb. [Online] [Cited: 2 23, 2011.]
<http://www.usabilidadweb.com.ar/postgre.php>.

Bibliografía

32. visual-paradigm. [Online] [Cited: 2 10, 2011.] <http://www.visual-paradigm.com/>.
33. Wolf, Dewayne, Perry, E. y L., Alexander. 2002. Software Engineering Notes. 2002.
34. www.siiu.udg.mx. [Online] [Cited: 2 16, 2011.]
http://www.siiu.udg.mx/html/pronad/doctos/introduccion_siia.pdf.
35. www.sirac.info. [Online] [Cited: 2 20, 2011.]
<http://www.sirac.info/Curtiembres/html/indicadores.asp>.
36. Roger S. Pressman. Ingeniería de software, un enfoque practico, chapter Técnicas de prueba del software, pages 281–299. McGraw-Hill, 2001

Glosario de Términos

AJAX: Es una técnica de desarrollo web para crear aplicaciones interactivas.

Dashboard: Es una página desarrollada en base a tecnología web mediante la cual se despliega en tiempo real información de la empresa extraída de varias fuentes o bases de datos.

Diagrama: Representación gráfica en la que se muestran las relaciones entre las diferentes partes de un conjunto o sistema.

Herramientas: Es lo que se va a usar para realizar determinado objetivo, es en lo que se apoya para llevar a cabo un producto terminado.

HTML: Acrónimo inglés de Hyper Text Markup Language (lenguaje de marcación de hipertexto), es un lenguaje de marcas diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web. Gracias a Internet y a los navegadores del tipo Explorer o Netscape, el HTML se ha convertido en uno de los formatos más populares que existen para la construcción de documentos.

Java Script: Es un lenguaje interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas Web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C. Al contrario que Java, JavaScript no es un lenguaje orientado a objetos propiamente dicho, ya que no dispone de herencia, es más bien un lenguaje basado en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad.

J2SE: Es la primera plataforma para el rápido desarrollo y despliegue de aplicaciones seguras, portátiles que se ejecutan en sistemas de servidor y escritorio que abarca la mayoría de sistemas operativos

Lenguaje del lado del servidor: Lenguajes de programación que se enfocan en funcionalidades que no tienen que ver con la interfaz de usuario, están hechas pero el cliente no es capaz de ver el código ni como este influye en lo que está haciendo.

Lenguaje del lado del cliente: Lenguajes que su función principal es mejorar la interfaz del cliente, ayudar al usuario a navegar y realizar operaciones sobre la IW sin que sea engorroso.

Glosario de Términos

Marco teórico: Todo lo que se va a usar para la confección y realización del software en cuestión.

PHP: Es un lenguaje de programación usado generalmente para la creación de contenido para sitios web.

Sistema de Información: es un conjunto de elementos orientados al tratamiento y administración de datos e información, organizados y listos para su posterior uso, generados para cubrir una necesidad.

Tecnología: Tecnología es el conjunto de conocimientos técnicos, ordenados científicamente, que permiten construir objetos y máquinas para adaptar el medio y satisfacer las necesidades de las personas.