

Documento de Arquitectura del Software
Proyecto:

Versión: 1

**Desarrollo de un Sistema de Gestión
para la Seccional de la U.E.T.D.
“Liceo Caracas”**

Historial de Revisiones

Versión	Fecha	Autor	Descripción
1.0	24/01/2012	Astrid Fernández Mailyn Hernández	<i>Entrega preliminar</i>
2.0	04/07/2012	Astrid Fernández Mailyn Hernández	<i>Entrega a la coordinación</i>

Índice de Contenido

1. Introducción	4
1.1 Alcance	4
1.2 Definiciones, Acrónimos y Abreviaturas.....	4
1.3 Documentos relacionados.....	5
2. Resumen Arquitectónico	5
2.1 Hechos más importantes.....	5
2.2 Estilo Arquitectónico	5
2.3 Objetivos de la Arquitectura.....	6
Componentes Significativos de la Arquitectura del Sistema	6
2.4 Presentación/Componentes de la Interfaz de Usuario	6
2.5 Componentes Lógicos de la Aplicación	7
2.6 Componentes de Almacenamiento de Datos	8
Vista de Casos de Uso	9
2.7 Diagrama de Clases	9
Vista de Procesos.....	10
Vista de Implementación.....	12
2.8 Visión General	12
Vista de Implantación.....	14
Vista de Datos	14
Integración.....	15
2.9 Integración de los Componentes y su Comunicación	15
3. Lista de Control de la Arquitectura	16
4. Aseguramiento de la Calidad	17
1.1.1 Esenciales.....	17
1.1.2 Esperados	18
1.1.3 Deseados	20

Documento de Arquitectura del Software

1. Introducción

1.1 Alcance

La elaboración de este documento permite considerar los factores involucrados en el desarrollo del software, no omitir detalles para su ejecución tomando en cuenta todos los requerimientos del sistema, tanto dentro como fuera del entorno del mismo, ya que es importante hacerlo con antelación para que el sistema sea estructurado de la mejor manera y resultado al utilizarlo sea óptimo. Además que Forma la columna vertebral para construir un software y es en gran medida responsable de permitir o no ciertos atributos de calidad del sistema

1.2 Definiciones, Acrónimos y Abreviaturas

- **Marco Lógico:** es un instrumento de planificación que permite estructurar los principales elementos de un proyecto, subrayando los lazos lógicos entre los insumos previstos, las actividades planeadas y los resultados esperados. [CRESPO, 2010].
- **Software:** conjunto de programas, instrucciones y reglas informáticas que permiten ejecutar distintas tareas en una computadora. [RAE]. Se considera que el software es el equipamiento lógico e intangible de una computadora.
- **U.E.T.D:** Unidad Educativa Técnico Deportiva.
- **Sistema:** conjunto de entidades caracterizadas por ciertos atributos, que tienen relaciones entre sí y están localizadas en un cierto ambiente, de acuerdo con un cierto objetivo. [PUELO, 1985].
- **Seccional:** departamento encargado de la organización de documentos y demás información importante referente a un lugar en específico. En este caso referente a los salones de clase, e información sobre los alumnos de la institución.
- **Requerimientos:** algo que se le pide o solicita a alguien, características que se desea que posea un sistema o un software.

1.3 Documentos relacionados

Título	Fecha	Organización	Identificador del documento
Marco Lógico	20/07/11	Fernández Astrid Hernández Mailyn	9.Marco_Lógico.odt
Acta Constitutiva de Proyecto	08/10/11	Fernández Astrid Hernández Mailyn	9._Plantilla_Acta_de _Constitucion_del_Pr oyecto-2.odt
Visión del Sistema	08/11/11	Fernández Astrid Hernández Mailyn	VISION_DEL_SISTE MA_REQUERIMIEN TOS.odt
Especificación de Requerimientos del Software	20/12/11	Fernández Astrid Hernández Mailyn	8.Especificación_Re querimientos_Softwa re_Grupo8.odt

2. Resumen Arquitectónico

2.1 Hechos más importantes.

Es un sistema para ser utilizado en página web, el ingreso de los usuarios a la página principal se limitará al personal de la seccional y profesores, siempre y cuando estén previamente registrados con usuario y contraseña para que sea validado al ingresar a las siguientes ventanas del sistema, de no ser así, será invitado a realizar el registro.

2.2 Estilo Arquitectónico

- La arquitectura que se empleara en el desarrollo del software será la de cliente/servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes.
- Esta arquitectura permitirá que un cliente realice peticiones a otro programa, el servidor, que le da respuesta.
- Cabe destacar que este software que se está mencionando es una aplicación web sencilla de ejecutarse en cualquier computador que cumpla con algunos requerimientos.

2.3 Objetivos de la Arquitectura

El software a desarrollar tiene los siguientes objetivos:

- Expansibilidad: Mantendrá la expansibilidad necesaria para facilitar ese proceso de desarrollar versiones posteriores
- Portabilidad: Por estar diseñado en PHP, MYSQL Y JAVASCRIPT, ocupa muy poco espacio en la memoria, lo que lo hace un sistema muy portátil
- Funcionalidad: Ha sido diseñado de manera tal que todas las funcionalidades estén integradas.
- Reutilización: Puede transferirse de un sistema a otro.

Componentes Significativos de la Arquitectura del Sistema

2.4 Presentación/Componentes de la Interfaz de Usuario

C-01 ingreso al sistema	
Descripción:	Autenticación de usuario, se exigen dos requerimientos de información para ingresar al sistema.
Requerimientos:	Ingresar usuario y contraseña, de no estar registrado acceder a registro nuevo usuario.
Interfaces Disponibles:	Ingreso al sistema y registro nuevo usuario.

C-02 Registro nuevo usuario.	
Descripción:	Formulario de registro para obtener un usuario y contraseña de manera de acceder a la aplicación.
Requerimientos:	Llenar todos los campos requeridos en el formulario aceptando el registro.
Interfaces Disponibles:	registro nuevo usuario.

C-03 Página principal administrador.	
Descripción:	Interfaz que muestra las diferentes opciones que tiene el sistema como registro de estudiantes, registro de horarios, búsqueda de listas, expedientes de estudiantes.
Requerimientos:	Autenticación de usuario.
Interfaces Disponibles:	Ingreso al sistema, registro nuevo usuario, registro de

C-03 Página principal administrador.	
	estudiantes, registro de horarios, búsqueda de listas, expedientes de estudiantes.

C-04 Página principal Usuario-Profesor.	
Descripción:	Interfaz que muestra las diferentes opciones que tiene el profesor, consultar expedientes, consultar horarios, Agregar Asistencias, Consultar Asistencias, Consultar Justificativo, Imprimir Reportes
Requerimientos:	Autenticación de usuario.
Interfaces Disponibles:	Ingreso al sistema, registro nuevo usuario, consultar expedientes, consultar horarios, Agregar Asistencias, Consultar Asistencias, Consultar Justificativo, Imprimir Reportes

2.5 Componentes Lógicos de la Aplicación

C-05 interfaces en HTML	
Descripción:	Pantallas que muestran la información que contiene la página.
Requerimientos:	Navegador web, herramientas aplicativas.
Interfaces Disponibles:	Ingreso al sistema, registro nuevo usuario, registro estudiantes, búsqueda de listas, expedientes de alumnos, entre otras.

C-06 PHP	
Descripción:	Lenguaje de programación con el que fueron creados los funcionamientos de las interfaces del software.
Requerimientos:	Interfaces ensambladas en HTML.
Interfaces Disponibles:	Ingreso al sistema, registro nuevo usuario, registro estudiantes, búsqueda de listas, expedientes de alumnos, entre otras.

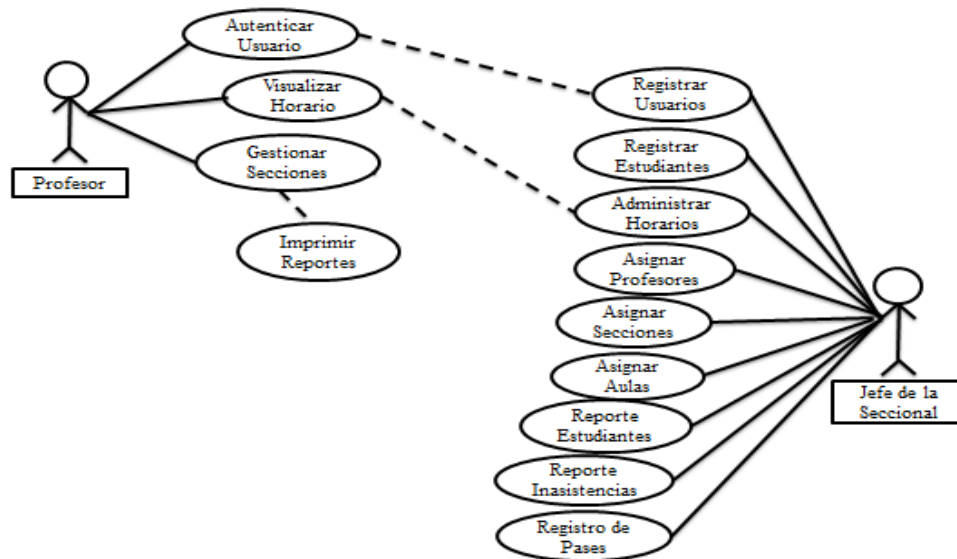
C-06 Java	
Descripción:	Lenguaje de programación para crear las validaciones del sistema..
Requerimientos:	Carpeta de extensión .js en el sistema.
Interfaces Disponibles:	Ingreso al sistema, registro nuevo usuario, registro estudiantes, búsqueda de listas, expedientes de alumnos, entre otras.

C-06 Sistema Operativo	
Descripción:	Cualquiera de los Sistemas Operativos necesarios para la instalación del sistema.
Requerimientos:	Windows, Linux. Cualquiera de sus versiones.
Interfaces Disponibles:	Se obtendrá la disponibilidad de todas las interfaces del software ya que el mismo será multiplataforma.

2.6 Componentes de Almacenamiento de Datos

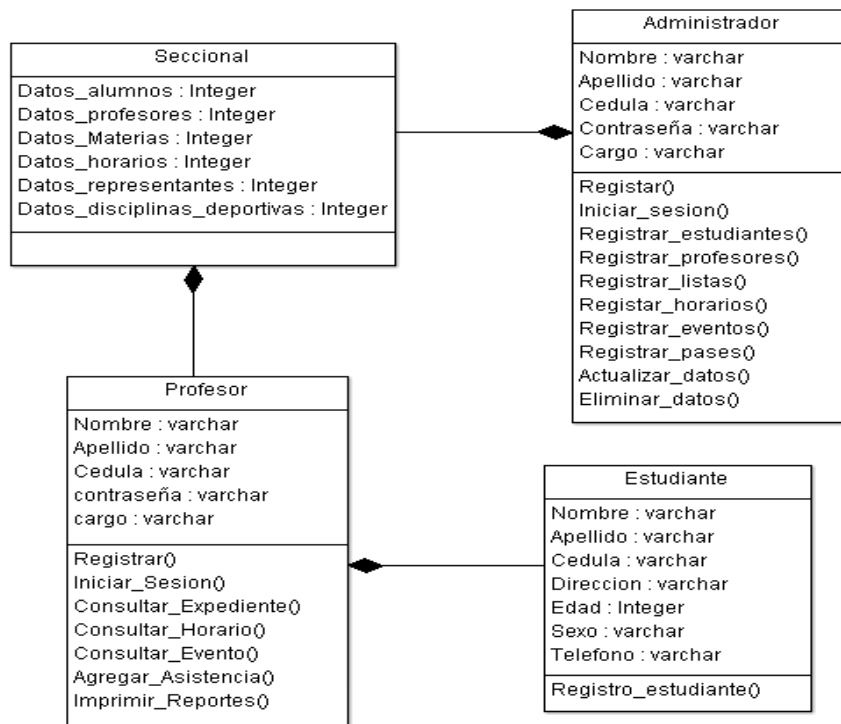
C-20: Base de Datos.	
Descripción:	Se utilizara la base de datos MySQL.
Requerimientos:	Wamp Server, XAMPP
Interfaces Disponibles:	

Vista de Casos de Uso

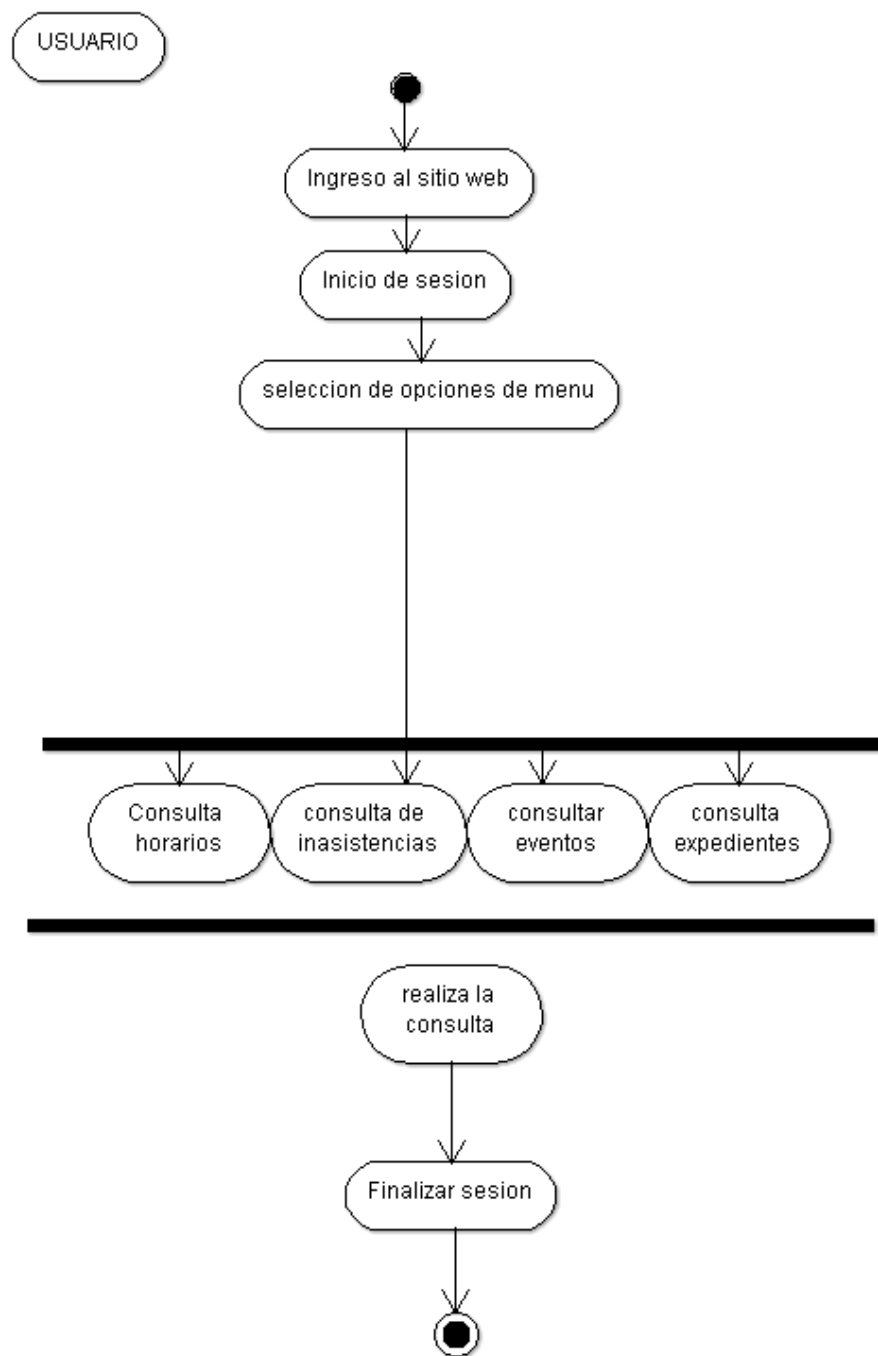


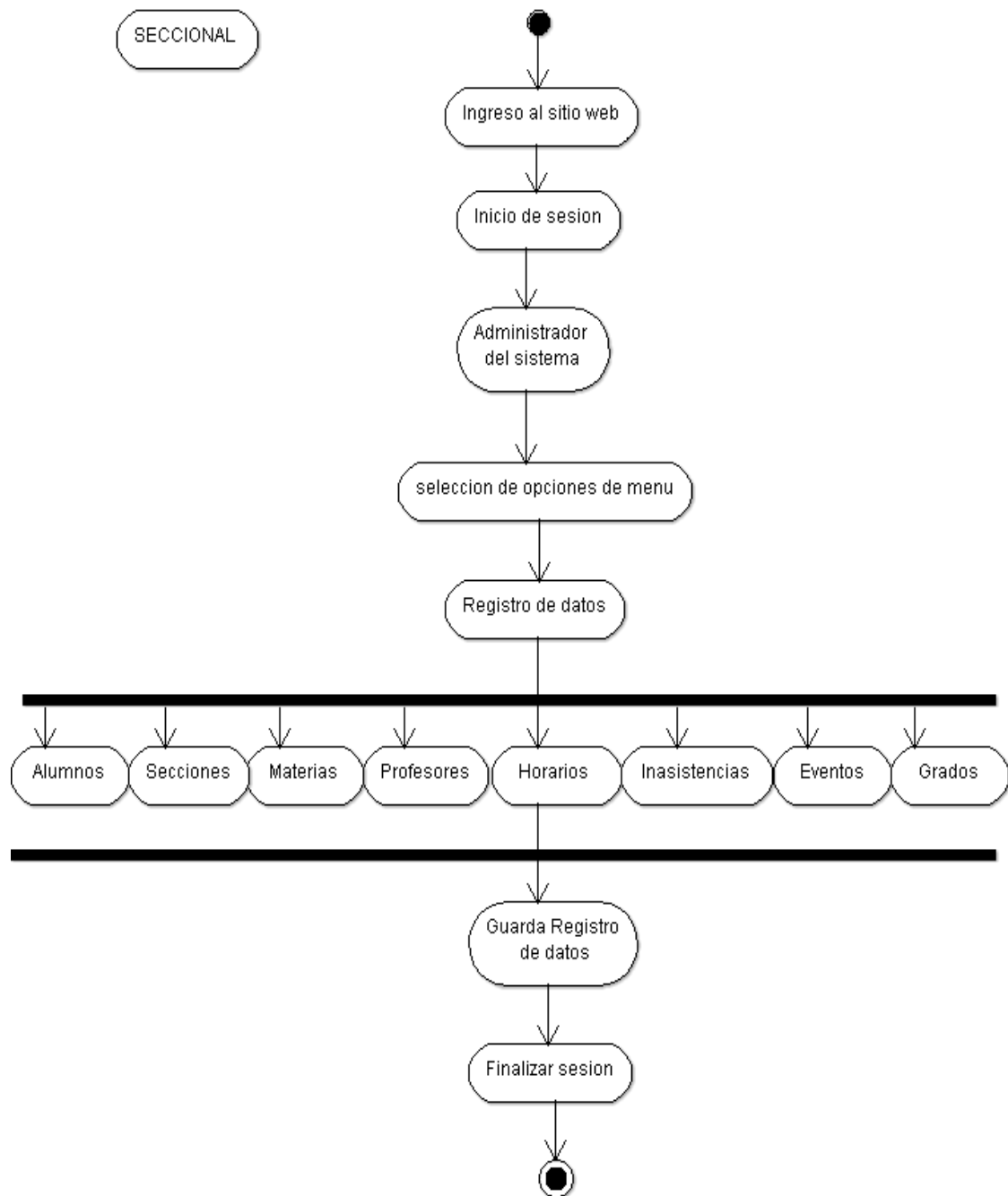
Vista Lógica

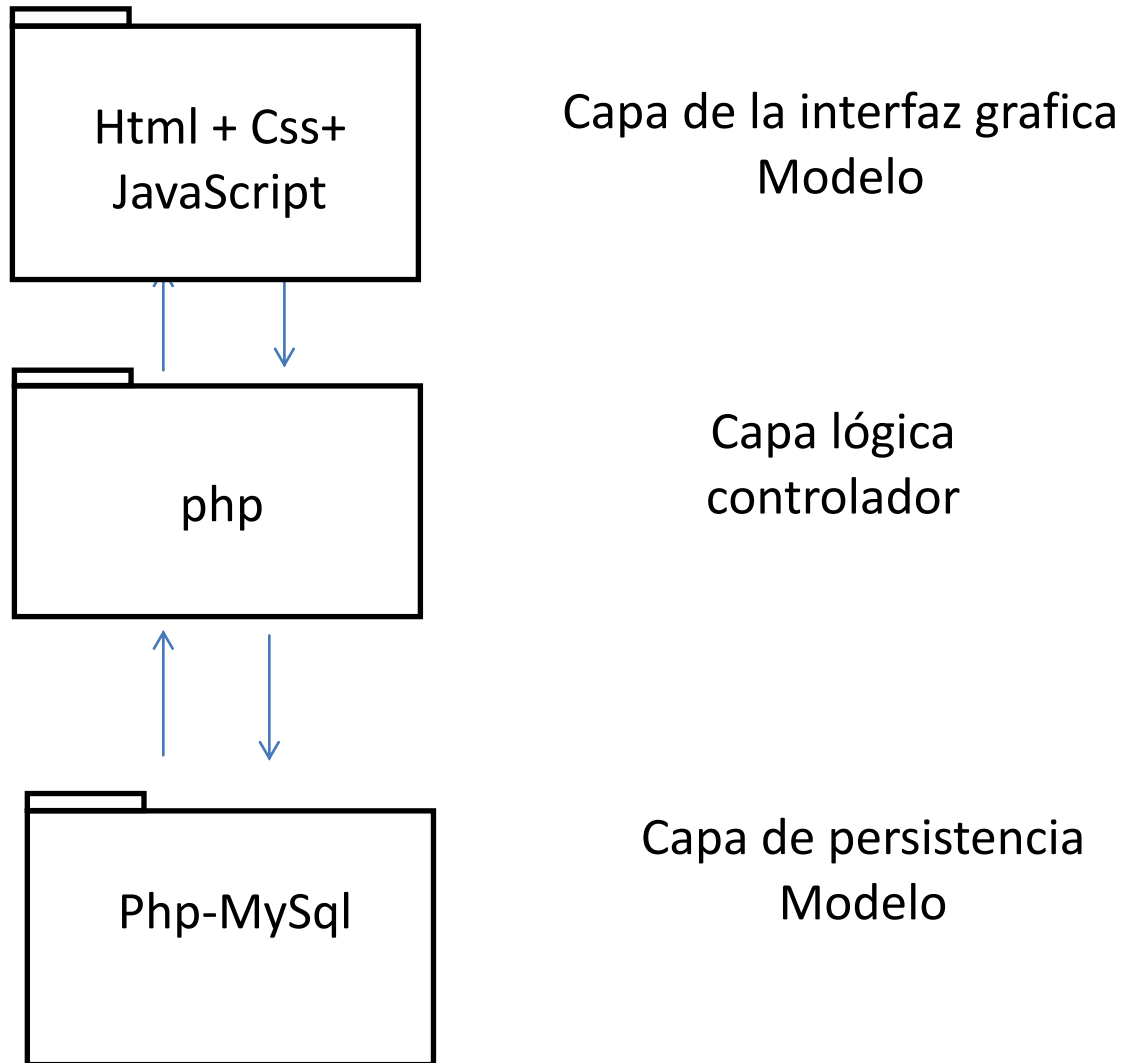
2.7 Diagrama de Clases



Vista de Procesos

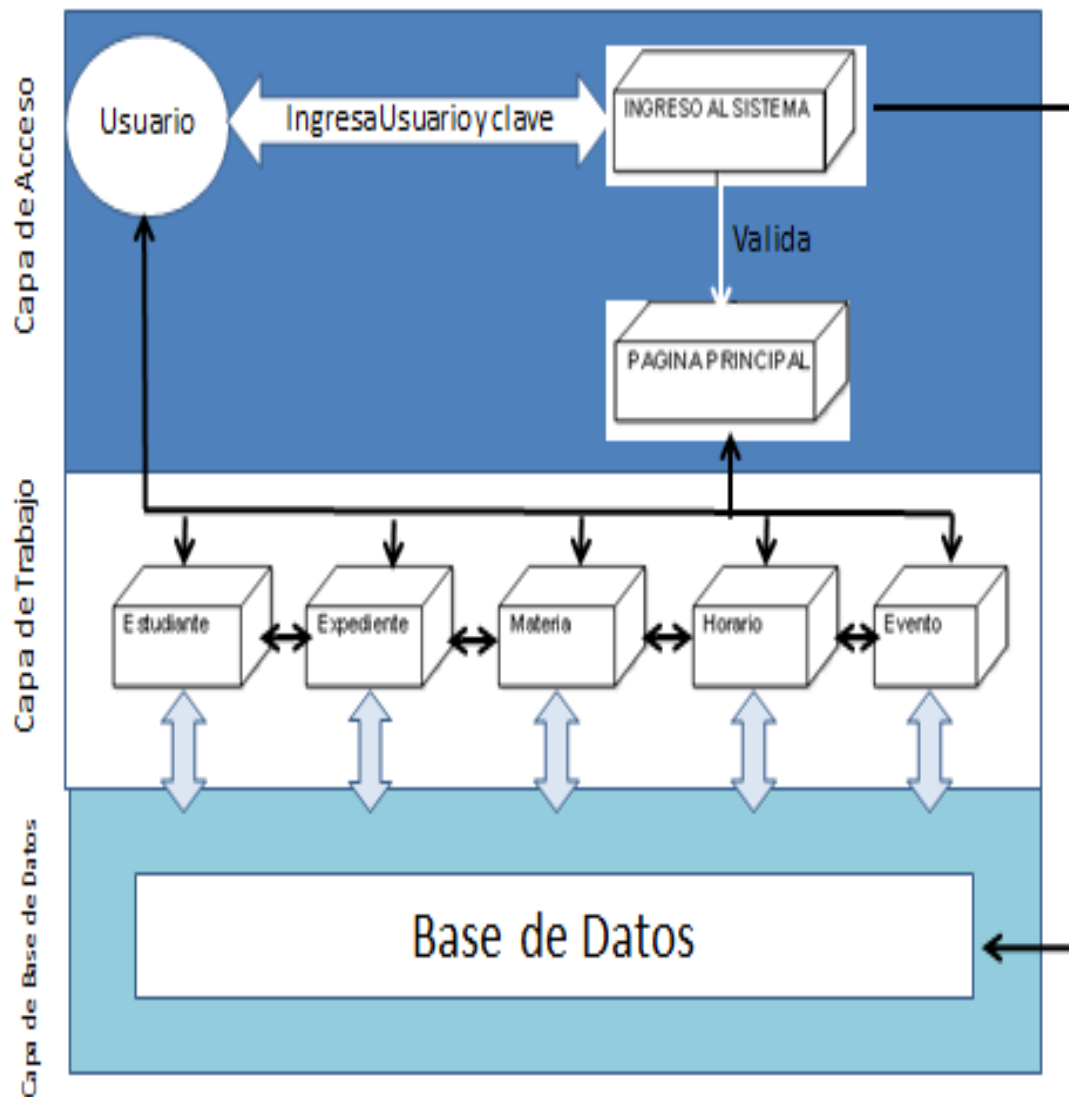




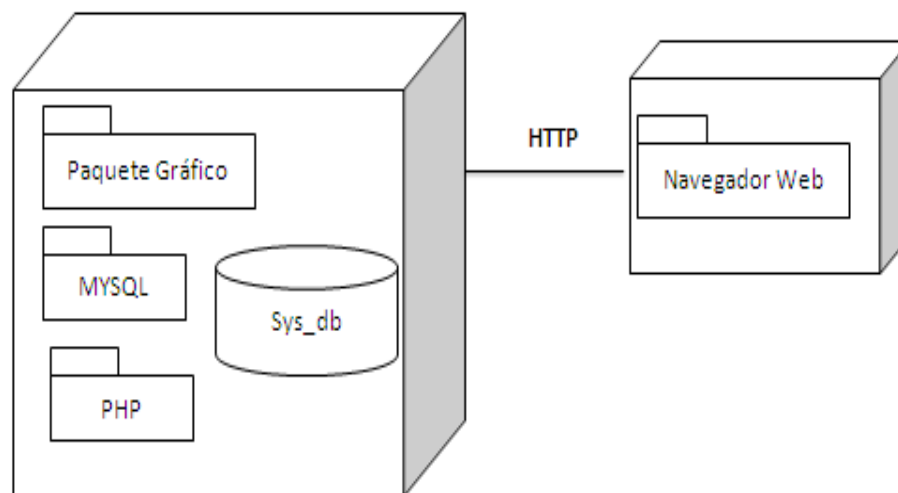
Vista de Implementación**2.8 Visión General**

- **Capa de Interfaz Gráfica:** Es la capa que presenta todo lo que el usuario puede observar, interactuar, utilizar.
- **Capa de Lógica:** En esta capa se reciben las llamadas al sistema para realizar la conexión entre la interfaz gráfica y la base de datos.
- **Capa de Persistencia:** Esta permite las conexiones de la base de datos con la interfaz gráfica, es la que ejecutan el comando de llamadas al sistema.

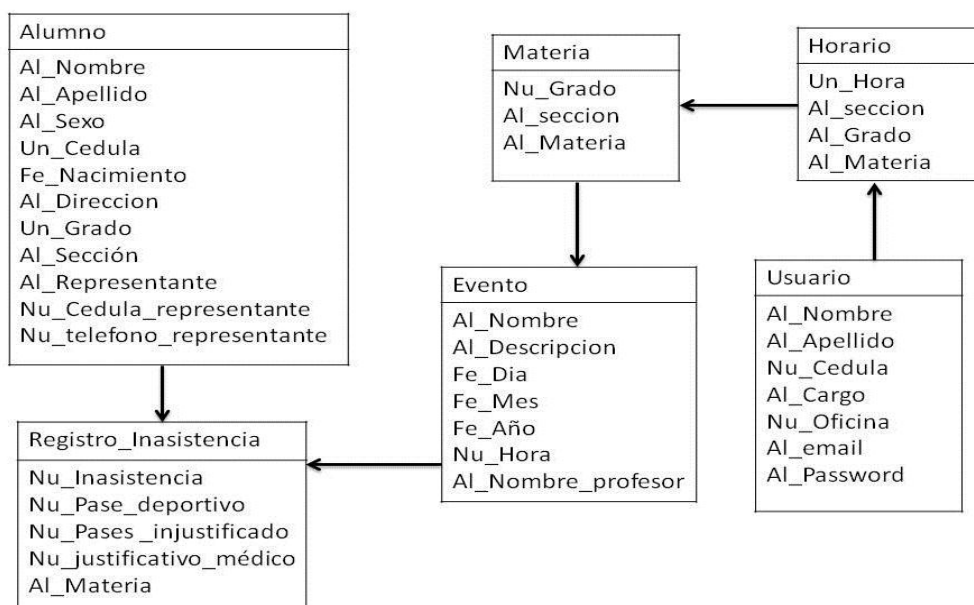
7.2. Capas



Vista de Implantación



Vista de Datos



Integración

2.9 Integración de los Componentes y su Comunicación

El código utiliza llamadas directas a procedimiento. La base de datos es accesada con un controlador. Los componentes dentro del mismo proceso usan llamadas directas a procedimientos o eventos Java.

El sistema ha sido diseñado del sistema se compone por dos paquetes de servicios principales, los cuales son: autenticación de usuario y el panel de control, del cual se descomponen los siguientes subsistemas: sistema de búsqueda, asignación de materias, registro de eventos y administrador de horarios.

- Autenticación de Usuario: página Web principal donde los usuarios previamente registrados se validan para ingresar al sistema, y desde donde los usuarios nuevos pueden entrar a la página de creación de usuario.
- Sistema de búsqueda: este paquete permite realizar una búsqueda avanzada de los estudiantes del plantel, así como de sus historias académicas.
- Asignación de materias: este paquete permite designar a los cursos escolares el listaras de las materias a cursar en el nuevo periodo escolar
- Registro de eventos: este paquete permite registrar los eventos deportivos que presentan los estudiantes, de esta manera se puede justificar la ausencia de los mismos en las materias a cursar.
- Administración de horarios: paquete permite crear y subir los

horarios para que los mismos sean asignados a profesores, aulas y alumnos.

3. Lista de Control de la Arquitectura

Evalúe su arquitectura respecto a cada uno de sus objetivos. A Continuación se colocan algunos ejemplos:

Facilidad de Integración

¿Se han previsto mecanismos para cada tipo necesario de integración?

Sí. En este sistema, todos los componentes están diseñados para trabajar de manera integrada. Y los componentes rehusados son integrados con interfaces simples, entre estos casos tenemos la herramienta FPDF que nos permite imprimir los reportes.

Expansibilidad

¿Qué tipos de componentes pueden ser añadidos después y cómo?

Pueden ser incluidos componentes desarrollados en software libre, integrando la misma en el código y en la base de datos, para comenzarla a utilizar.

Ajuste a la Capacidad

¿Cómo esta arquitectura ha ajustado las necesidades de recursos de los componentes a los equipos?

Se ha planteado integrar el sistema en un computador (gestor de base de datos, servidor de aplicaciones, servidor web) ya que será el jefe de la seccional quien insertara los datos en el sistema, el mismo ha sido diseñado de manera tal que no presente falla alguna.

Del Acuerdo entre el Equipo de Desarrollo y los Involucrados

¿La arquitectura se ha comunicado al equipo del desarrollo y a los involucrados?

Sí, cada uno entiende. La opinión de de los involucrados responde a lo que se ha propuesto y a su vez se ha cumplido con los requerimientos de los mismos.

4. Aseguramiento de la Calidad

1.1.1 Esenciales

- Funcionalidad > Corrección

El sistema de gestión de la U.E.T.D. “Liceo Caracas” ha sido desarrollado para cumplir todos los requerimientos del usuario, y a su vez efectúa con exactitud todas las tareas que se le han especificado, consiguiendo los objetivos encomendados por el cliente.

- Funcionalidad > Robustez

Con la culminación del sistema se alega que si se presentan casos o aspectos que no se contemplaron con anterioridad, como errores en el software, la robustez hará un despliegue de mensajes de error apropiados, en donde se pueda terminar la aplicación de

manera limpia y segura para los datos.

1.1.2 Esperados

- Funcionalidad > Exactitud

El sistema diseñado contempla toda la exactitud requerida ya que responde correctamente a las tareas que le han sido asignadas, cumpliendo con requerimientos de tiempo, memoria y flexibilidad, prestando todas las utilidades que el usuario requiere.

- Funcionalidad > Compatibilidad

Todas las funciones desarrolladas en el sistema son compatibles, ya que muchas funciones han sido derivadas de macro funciones, con el fin de ejecutar las diversas tareas prescritas por lo que el sistema es muy dinámico y funcional.

- Funcionalidad > Corrección medible

Se trata de producir un software libre de fallos, aunque esto sea un poco complicado ya que el software es hecho por el hombre y el hombre no es perfecto, se está trabajando para lograr producir software con un mínimo nivel de errores en cuanto en medida sea posible.

- Usabilidad > Comprensibilidad y Legibilidad

El sistema genera una interacción muy sencilla entre el usuario y el, debido a lo familiar y comprensible que ha sido diseñado, siendo visibles para el usuario las herramientas que requiere para cumplir sus tareas en dicho sistema.

- Usabilidad > Apoyo para tareas

Se han determinado una serie de momentos de interacción o tareas del usuario, para que el sistema los pueda soportar, por ello ha sido diseñado de manera tal, que sea cómodo y sencillo para el usuario navegar dentro del sistema.

- Usabilidad > Eficiencia

Se dice que el sistema trabaja de una manera eficiente ya que su tiempo de respuesta ante cada acción es de un mínimo plazo de tiempo, por lo que genera dinamismo al momento de interactuar con el sistema

- Usabilidad > Seguridad

Por cuestiones de seguridad el sistema ha sido desarrollado para que solo ingresen personas que estén registradas en el mismo, por medio de un usuario y una contraseña (login), para ingresar a nuevas ventanas del sistema, él mismo valida el login para cumplir dicha acción, por lo tanto el software es completamente seguro.

- Usabilidad > Consistencia y Familiaridad

El sistema de gestión de seccional se ha estructurado con la guía de previos sistemas de control de estudio con resultados excelentes en cuanto a los usuarios, por lo que se considera que resulta muy familiar para los usuarios este tipo de interfaz, no solo para los sistemas de este tipo, sino también con los programas que son usados cotidianamente.

- Usabilidad > Satisfacción Subjetiva

Se ha diseñado el sistema acorde a los requerimientos del usuario, por lo que al usarlo el sistema responde a todas las

exigencias del cliente, por lo que el resultado es muy satisfactorio tanto para los desarrolladores, como para el cliente.

1.1.3 Deseados

- **Confiabilidad > Consistencia en carga**
Todo sistema tiene límites de capacidad, cuando estos límites son excedidos, el sistema no pierde la información, solo se debe esperar que culmine la carga.
- **Confiabilidad > Consistencia bajo concurrencia**
El sistema está desarrollado para permitir accesos simultáneos por usuarios múltiples, sin embargo el sistema está dirigido a un límite de pocas personas por lo que su respuesta debería ser inmediata ante cada acción.
- **Confiabilidad > Disponibilidad bajo carga**
Cuando los límites del sistema son excedidos, el mismo continúa ejecutando aquellas solicitudes que es capaz de cumplir sin detener el proceso de ninguna.
- **Confiabilidad > Longevidad**
El sistema es capaz de operar tanto como lo necesite, sin terminarse progresivamente los recursos disponibles.
- **Eficiencia**
Capacidad del software para hacer buen uso de los recursos que manipula, por ello en el desarrollo del sistema se ha optimizado excesivamente para mantener un balance adecuado entre eficiencia y corrección.
- **Escalabilidad**
El sistema continuara satisfaciendo requerimientos del cliente en situaciones en que sus parámetros se han incrementado.
- **Escalabilidad > Desempeño bajo carga**
El sistema responde cuando sus servicios tienen muchas solicitudes de muchos usuarios, sin embargo el sistema solo será

manejado por la seccional por lo que los servicios serán limitadamente concurridos.

- Escalabilidad > Grandes volúmenes de datos

El software puede manejar grandes volúmenes de datos, ejecutando operaciones eficientes aunque el tamaño del volumen de datos aumente, siendo la interfaz del usuario útil aunque ésta incremente en longitud.

- Operabilidad

Las necesidades a largo plazo de los administradores del sistema son apoyadas confiablemente, ya que es fácil de instalar y la actualización no genera pérdida de datos.

- Capacidad de mantenimiento > Comprensibilidad

El sistema está diseñado de la manera más comprensible, de manera que al desarrollar una nueva versión o mejora del sistema el mismo facilita la incorporación en el diseño.

- Capacidad de mantenimiento > Capacidad de evolución

El sistema fácilmente se puede incrementar u modificar para cumplir nuevas exigencias y requerimientos.

- Capacidad de mantenimiento > Capacidad de prueba

Antes de la implantación del sistema el mismo será sometido a diversas pruebas con la finalidad de percibir los errores para proceder a depurar y reducir el margen de error a cero.

Documento de Arquitectura del Software	
Sistema de Gestión de Seccional para la U.E.T.D. “Liceo Caracas”	Versión: 2.0

Copyright (C) 2003 Jason Robbins. Todos los derechos reservados.

Copyright (C) 2006 CNTI. Todos los derechos reservados.

La redistribución y el uso de las plantillas, con o sin modificación, están permitidas siempre que se cumplan las siguientes condiciones expuestas en:

http://merinde.rinde.gob.ve/index.php?option=com_remository&Itemid=37&func=fileinfo&id=1