**MINISTERIO DE EDUCACIÓN**

**DIRECCIÓN REGIONALDE EDUCACIÓN TACNA**

**INSTITUTO DE EDUCACIÓN SUPERIOR TECNOLÓGICO PRIVADO**

**“UNITEK - TACNA”**

CARRERA PROFESIONAL TÉCNICA:

**COMPUTACIÓN E INFORMÁTICA**



PROYECTO:

**SISTEMA DE ASISTENCIA DE PERSONAL VIA WEB UTILIZANDO TECNOLOGIA PHP MVC PARA LA OFICINA DESENTRALIZADA DE ESTADISTICA E INFORMATICA.**

Para Optar el Título de:

**PROFESIONAL TÉCNICO EN COMPUTACIÓN E INFORMÁTICA**

Presentado por:

* Valdez Pacohuanaco, Salomón
* Sacari Choque, Wilson Fredy
* Quispe Serrano, Eder Jhoel

TACNA – 2018

**DEDICATORIA**

La sonrisa y el cariño son mi alegría diaria, mi energía, las ganas de vivir. Me basta con mirarlos a los ojos para sentir que el sol brilla a cada amanecer y el cielo se torne más azul, gracias a mis padres por darme tanto amor y conseguir que el mundo cada vez parezca mejor.

AGRADECIMIENTO

Expresamos nuestro agradecimiento al Instituto Superior Tecnológico Particular UNITEK – TACNA a través de su carrera computación e informática por nuestra formación académica y profesional

**INDICE**

[**DEDICATORIA 2**](#_Toc531087221)

[**AGRADECIMIENTO 3**](#_Toc531087222)

[**INTRODUCCION 11**](#_Toc531087223)

[**CAPITULO I 12**](#_Toc531087224)

[**IDENTIFICACIÓN DEL PROYECTO 12**](#_Toc531087225)

[1.1. TÍTULO DEL PROYECTO 12](#_Toc531087226)

[1.2. DESCRIPCIÓN 12](#_Toc531087227)

[1.3. AUTOR 12](#_Toc531087228)

[1.4. FECHA DE PRESENTACIÓN 12](#_Toc531087229)

[**CAPITULO II 13**](#_Toc531087230)

[**MARCO GENERAL 13**](#_Toc531087231)

[2.1. DESCRIPCIÓN GENERAL DE LA EMPRESA 13](#_Toc531087232)

[2.1. NOMBRE DE LA EMPRESA 13](#_Toc531087233)

[2.2. UBICACIÓN 13](#_Toc531087234)

[2.3. DESCRIPCIÓN 13](#_Toc531087235)

[**CAPITULO III 14**](#_Toc531087236)

[**INVESTIGACIÓN PRELIMINAR 14**](#_Toc531087237)

[3.1. PROBLEMÁTICA GENERAL 14](#_Toc531087238)

[3.2. JUSTIFICACIÓN DEL PROYECTO 14](#_Toc531087239)

[3.3. OBJETIVOS GENERAL Y OBJETIVOS ESPECÍFICOS 14](#_Toc531087240)

[3.4. ALCANCE DEL PROYECTO 15](#_Toc531087241)

[3.5. ESTUDIO DE FACTIBILIDAD 15](#_Toc531087242)

[3.5.1. FACTIBILIDAD TÉCNICA 15](#_Toc531087243)

[3.5.2. FACTIBILIDAD OPERATIVA 16](#_Toc531087244)

[3.5.3. FACTIBILIDAD ECONÓMICA 17](#_Toc531087245)

[3.5.4. FACTIBILIDAD LEGAL 18](#_Toc531087246)

[3.6. RECURSOS PRELIMINARES REQUERIDOS 18](#_Toc531087247)

[3.6.1. RECURSOS HUMANOS 18](#_Toc531087248)

[3.6.2. RECURSOS DE SOFTWARE 19](#_Toc531087249)

[3.6.3. RECURSOS DE HARDWARE 27](#_Toc531087250)

[3.7. COSTOS 28](#_Toc531087251)

[3.8. BENEFICIOS 28](#_Toc531087252)

[3.9. TÉCNICAS PARA LA OBTENCIÓN DE INFORMACIÓN 28](#_Toc531087253)

[**CAPITULO IV 30**](#_Toc531087254)

[**PLAN GENERAL DEL PROYECTO 30**](#_Toc531087255)

[4.1. DESCRIBIR LA ESTRATEGIA DE DESARROLLO PARA EL SISTEMA 30](#_Toc531087256)

[4.2. PLAN GENERAL DE DESARROLLO DEL PROYECTO 33](#_Toc531087257)

[4.2.1. DEFINIR EL PANORAMA DEL PROYECTO 33](#_Toc531087258)

[4.2.2. ESTRUCTURA DE ACTIVIDADES (Usar el MS-Project) 33](#_Toc531087259)

[4.2.2.1. DIAGRAMA DE GANTT CON ASIGNACIÓN DE RECURSOS 34](#_Toc531087260)

[**CAPITULO V 35**](#_Toc531087261)

[**ANÁLISIS DEL SISTEMA 35**](#_Toc531087262)

[5.1. ESPECIFICACIÓN DE REQUERIMIENTOS 35](#_Toc531087263)

[5.1.1. REQUERIMIENTOS BÁSICOS 35](#_Toc531087264)

[5.1.1.1. REQUERIMIENTOS FUNCIONALES 35](#_Toc531087265)

[5.1.1.2. REQUERIMIENTOS NO FUNCIONALES 36](#_Toc531087266)

[5.1.2. REQUERIMIENTOS GENERALES DE LA ORGANIZACIÓN 36](#_Toc531087268)

[5.1.3. REQUERIMIENTOS DEL USUARIO 37](#_Toc531087269)

[5.2. PROCESOS DEL NEGOCIO 38](#_Toc531087270)

[5.3. REGLAS DEL NEGOCIO 38](#_Toc531087271)

[5.4. PAQUETES 38](#_Toc531087272)

[5.4.1. DESCRIPCIÓN GENERAL DE CADA PAQUETE 38](#_Toc531087273)

[5.4.2. CONSTRUCCIÓN DEL DIAGRAMA DE PAQUETES 39](#_Toc531087274)

[5.5. DIAGRAMAS DE CASOS DE USO 39](#_Toc531087275)

[5.5.1. DEFINICIÓN DE CASOS DE USO 39](#_Toc531087276)

[5.5.1.1. DESCRIPCIÓN DE ACTORES 39](#_Toc531087277)

[5.5.1.2. DESCRIPCIÓN DE CASOS DE USO 40](#_Toc531087278)

[5.5.2. CONSTRUCCION DE DIAGRAMAS DE SECUENCIA. 46](#_Toc531087279)

[**CAPITULO VI 49**](#_Toc531087280)

[**DISEÑO DEL SISTEMA 50**](#_Toc531087281)

[6.1. IDENTIFICACIÓN DE OBJETOS 50](#_Toc531087282)

[6.2. MODELO DEL NEGOCIO 51](#_Toc531087283)

[6.2.1. DESCRIBIR CÓMO FUNCIONA O CÓMO FUNCIONARÍA EL SISTEMA CON UN TEXTO EN FORMA PRECISA 51](#_Toc531087284)

[6.2.2. IDENTIFICACIÓN DE LOS TIPOS DE RELACIONES 51](#_Toc531087285)

[6.2.3. MODELO CONCEPTUAL DEL NEGOCIO 54](#_Toc531087286)

[6.3. DISEÑO E IMPLANTACIÓN DE LA BASE DE DATOS 54](#_Toc531087287)

[6.3.1. GENERACIÓN DEL MODELO DE DATOS A PARTIR DEL DIAGRAMA DE CLASES 55](#_Toc531087288)

[6.3.2. CREACIÓN DE LA BASE DE DATOS 55](#_Toc531087289)

[6.3.3. CONSTRUIR EL DICCIONARIO DE DATOS 56](#_Toc531087290)

[**CAPITULO VII 58**](#_Toc531087291)

[**CONSTRUCCIÓN DEL SISTEMA 58**](#_Toc531087292)

[7.1. FUNCIONES DEL SISTEMA 58](#_Toc531087293)

[7.2. PROGRAMACIÓN DEL SISTEMA DE ASISTENCIA 59](#_Toc531087294)

[7.3. CONEXIÓN A BASE DE DATOS DESDE PHP. 62](#_Toc531087295)

[7.4. TIPOS DE USUARIOS 66](#_Toc531087296)

[7.5. NIVELES DE SEGURIDAD 67](#_Toc531087297)

[7.6. DISEÑO DE LA ARQUITECTURA DEL SISTEMA 67](#_Toc531087298)

[**CAPITULO VIII 71**](#_Toc531087299)

[**PRUEBA DEL SISTEMA 71**](#_Toc531087300)

[8.1. PROGRAMA DE CAPACITACIÓN AL USUARIO 71](#_Toc531087301)

[8.2. EVALUACIÓN DEL SISTEMA 71](#_Toc531087302)

[8.2.1. EVALUACIÓN OPERACIONAL 71](#_Toc531087303)

[8.2.2. IMPACTO ORGANIZACIONAL 72](#_Toc531087304)

[8.2.3. DESEMPEÑO DEL DESARROLLO 72](#_Toc531087305)

[8.3. PLAN DE PRUEBAS DEL SISTEMA 72](#_Toc531087306)

[8.4. PLAN DE MANTENIMIENTO DEL SISTEMA 72](#_Toc531087307)

[**CAPITULO IX 73**](#_Toc531087308)

[**REQUERIMIENTOS TÉCNICOS 73**](#_Toc531087309)

[9.1. HARDWARE 73](#_Toc531087310)

[9.2. SOFTWARE 73](#_Toc531087311)

[9.3. ARQUITECTURA DEL SISTEMA 74](#_Toc531087312)

[**CONCLUSIONES 75**](#_Toc531087313)

[**RRECOMENDACIONES 76**](#_Toc531087314)

[**BIBLIOGRAFIA 77**](#_Toc531087315)

[**ANEXOS 79**](#_Toc531087316)

**LISTA DE TABLAS**

[**Tabla 1 :** Recursos Informáticos 12](#_Toc531085712)

[**Tabla 2 :** Cuadro Económico para el Sistema Operativo 13](#_Toc531085713)

[**Tabla 3:** Recursos Humanos 14](#_Toc531085714)

[**Tabla 4:** Cuadro Economico para Sistema Operativo 24](#_Toc531085715)

[**Tabla 5:** Cronograma del Proyecto (Diagrama de Gantt). 30](#_Toc531085716)

[**Tabla 6:** Requerimientos Funcionales 31](#_Toc531085717)

[**Tabla 7:** Requerimientos del Usuario 33](#_Toc531085718)

[**Tabla 8:** Procesos de la Empresa 34](#_Toc531085719)

[**Tabla 9:** Descripción de Paquetes 35](#_Toc531085720)

[**Tabla 10:** Documentación CU Gestionar Registro del Empleado 36](#_Toc531085721)

[**Tabla 11:** Documentación CU Gestionar Registro Asistencia 38](#_Toc531085722)

[**Tabla 12:** Documentación CU Gestionar Registro Horario 39](#_Toc531085723)

[**Tabla 13:** Documentación CU Gestionar Registro de Permiso de asistencia 41](#_Toc531085724)

[**Tabla 14**: Diagrama de Secuencia 42](#_Toc531085725)

[**Tabla 15:** Tipo de Relación Clase Usuario 47](#_Toc531085726)

[**Tabla 16:** Tipo de Relación Clase Cargo 47](#_Toc531085727)

[**Tabla 17:** Tipo de Relación Clase Empleado 47](#_Toc531085728)

[**Tabla 18:** Tipo de Relación Clase Horario 48](#_Toc531085729)

[**Tabla 19:** Tipo de Relación Clase Registro 48](#_Toc531085730)

[**Tabla 20:** Tipo de Relación Clase Curso 49](#_Toc531085731)

[**Tabla 21:** Diccionario De Datos 52](#_Toc531085732)

[**Tabla 22:** Funciones del Sistema 54](#_Toc531085733)

[**Tabla 23:** Nombre de Formularios 64](#_Toc531085734)

[**Tabla 24:** Descripción de la Capa de Negocios (Controllers). 64](#_Toc531085735)

[**Tabla 25:** Descripción de la Capa de Acceso a Datos (ABD). 64](#_Toc531085736)

[**Tabla 26:** Descripción de la Capa de Datos 65](#_Toc531085737)

[**Tabla 27**: Elementos Técnicos (Hardware) 69](#_Toc531085738)

[**Tabla 28:** Arquitectura del Sistema 70](#_Toc531085739)

**LISTA DE FIGURAS**

[**Ilustración 1:** Tecnologías independiente de AJAX 24](#_Toc531097142)

[**Ilustración 2:** Funcionamiento Ajax 25](#_Toc531097143)

[**Ilustración 3:** Formato Json 26](#_Toc531097144)

[**Ilustración 4:**Data Tables 29](#_Toc531097145)

[**Ilustración 5:**Ejemplo de Mensaje de Alerta (Alertefy) 29](#_Toc531097146)

[**Ilustración 6:** Diseño Responsive 30](#_Toc531097147)

[**Ilustración 7:** Ejemplo Media Queries 31](#_Toc531097148)

[**Ilustración 8 :** Componentes de Modelo Vista Controlador 32](#_Toc531097149)

[**Ilustración 9:** Modelo vista Controlador en PHP Nativo 36](#_Toc531097150)

[**Ilustración 10:** Fases del método RUP 40](#_Toc531097151)

[**Ilustración 11:** Estructura de Descomposicion del Trabajo 43](#_Toc531097152)

[**Ilustración 12**: Modelo Conceptual 63](#_Toc531097153)

[**Ilustración 13:** Base de Datos 64](#_Toc531097154)

[**Ilustración 14:** Arquitectura de Modelo – Vista - Controlador 69](#_Toc531097155)

[**Ilustración 15:** Registro Asistencia 72](#_Toc531097156)

[**Ilustración 16**: Vista Usuarios Activos 72](#_Toc531097157)

[**Ilustración 17:** Arquitectura Sistema 78](#_Toc531097158)

**LISTA DE DIAGRAMAS**

[**Diagrama 1:** Diagrama de Paquetes 36](#_Toc531085941)

[**Diagrama 2:** Caso de Uso Registrar Empleado 37](#_Toc531085942)

[**Diagrama 3:** Caso de Uso Registrar Asistencia 39](#_Toc531085943)

[**Diagrama 4:** Caso de Uso Registro Horario 40](#_Toc531085944)

[**Diagrama 5:** Caso de Uso Registrar Permiso Asistencia 42](#_Toc531085945)

[**Diagrama 6**: Diagrama de Secuencia Registrar Asistencia 44](#_Toc531085946)

[**Diagrama 7:** Diagrama de Secuencia Registrar Empleado 45](#_Toc531085947)

[**Diagrama 8:** Diagrama de Secuencia Registrar Horario 46](#_Toc531085948)

[**Diagrama 9:** Diagrama de Clases 52](#_Toc531085949)

[**Diagrama 10:** Diagrama de Despliege 67](#_Toc531085950)

**INTRODUCCION**

En los últimos años las tecnologías de información han tenido un significativo desarrollo. Ello ha generado un fuerte impacto en los distintos ámbitos del que hacer de las personas y de la actividad económica en las empresas, nuestra función como ingenieros es ofrecer nuevas soluciones de arquitectura de software y ofrecer una plataforma adecuada para la integración de los sistemas en la organización, esto conlleva a una toma de decisiones correcta.

Actualmente según las investigaciones de mercado el índice de sistematización en las entidades públicas es de 85%, y es fundamental para brindar un servicio de calidad. Los sistemas son usados como estrategias para lograr la competitividad.

Para el presente proyecto me basare en la gestión de asistencia de la Institución ODEI TACNA, la cual se encarga de registrar la asistencia de los docentes que tienen curso a su cargo, los cuales deben marcar la entrada y salida, para esto se utilizara el lenguaje de programación de php y MySQL, el cual dicha herramienta me brindara un desarrollo escalable y con una performance adecuada.

CAPITULO I

IDENTIFICACIÓN DEL PROYECTO

* 1. TÍTULO DEL PROYECTO

SISTEMA DE ASISTENCIA DE PERSONAL VIA WEB UTILIZANDO TECNOLOGIA PHP MVC PARA LA OFICINA DESENTRALIZADA DE ESTADISTICA E INFORMATICA.

* 1. DESCRIPCIÓN

El presente proyecto, comprende en la realización de un sistema con tecnología moderna, el cual se encargará de manejar la asistencia del personal, primordialmente de los docente y practicantes vía web de la ODEI TACNA. Donde en el sistema registrara la asistencia de entrada y salida a los diferentes cursos que presta el servicio la institución. Este proyecto es indispensable para la solución del manejo de la información de asistencia del personal docente en los diferentes horarios.

* 1. AUTOR
* Salomón Valdez Pacohuanaco
* Wilson Fredy Sacari Choque
* Eder Jhoel Quispe Serrano
  1. FECHA DE PRESENTACIÓN

27/11/2018

1. CAPITULO II
2. MARCO GENERAL
   1. DESCRIPCIÓN GENERAL DE LA EMPRESA
   2. NOMBRE DE LA EMPRESA

ODEI TACNA.

* 1. UBICACIÓN

La ODEI Tacna, se encuentra ubicado en el centro de la ciudad, en la Av. San Martin Nº 520 (Cercado).

* 1. DESCRIPCIÓN

La capacitación que desarrolla la ENEI de ODEI TACNA, asegura la calidad educativa, la cual considera en primer lugar , la eficacia, para lo cual consideramos que los participantes, aprenden aquello que está establecido en los planes y programas curriculares, la segunda es la calidad, que permite que se reciban los conocimientos necesarios, en forma práctica, para lo cual se cuenta con docentes calificados y con experiencia, materiales de estudio, estrategias didácticas, y bases de datos actualizadas.

1. CAPITULO III

INVESTIGACIÓN PRELIMINAR

* 1. PROBLEMÁTICA GENERAL

La institución ODEI TACNA, no tiene el control adecuado de los registros de asistencia de los docentes y practicante en sus diferentes cursos que dicta, dado que el procedimiento de registro es manual, ocasionándole no tener un registro adecuado de las asistencias del personal. A su vez tiene una demora sustancial en el procesamiento de información de las asistencias para los descuentos respectivos.

* 1. JUSTIFICACIÓN DEL PROYECTO

La falta de tener un control adecuado de las asistencias de los docentes, siendo una institución que brinda cursos de tecnología, se hace indispensable contar con un sistema adecuado para automatizar el procedimiento de información.

* 1. OBJETIVOS GENERAL Y OBJETIVOS ESPECÍFICOS

**Objetivo general**

Desarrollar el Análisis y diseño e implementación del sistema de asistencia de docentes y personal practicante, utilizando tecnología Php MVC para el manejo de control del personal docente y practicante.

**Objetivo especifico**

* El Sistema web registrara la asistencia de los docentes y practicantes, mediante su código que se le proporcione.
* El Sistema web realizara el manejo de los empleados (Docentes) para realizar la gestión de su información (Guardar, Modificar, Inhabilitar y Buscar)
* El Sistema web proporcionara los reportes diarios y mensuales del personal docente, en su asistencia de sus diferentes cursos.
  1. ALCANCE DEL PROYECTO

El sistema a desarrollar manejara el procedimiento de asistencia de los docentes y practicantes. No realizará el control del personal administrativo, su funcionamiento solo será mediante la intranet de la institución. Solo se tomará la asistencia y permisos realizados por los docentes.

* 1. ESTUDIO DE FACTIBILIDAD
     1. FACTIBILIDAD TÉCNICA

La Factibilidad Técnica consistió en realizar una evaluación de la tecnología existente en la Institución ODEI TACNA, este estudio estuvo destinado a recolectar información sobre los componentes técnicos que posee la Empresa y la posibilidad de hacer uso de los mismos en el desarrollo e implementación del sistema propuesto y de ser necesario, la organización cuenta con equipos de computación los cuales son capaces de satisfacer las necesidades del sistema.

Desarrollando este análisis de hardware se vio lo siguiente:

**La Empresa cuenta con 01 servidor**:

* 01 servidor de Base de Datos.
* La organización cuenta con el siguiente software Sistema Operativo Windows, por lo que el sistema elaborado se implementara en dicha organización con la ayuda del software MySQL, Php.

**RECURSOS INFORMATICOS.**

Los Recursos Informáticos y tecnológicos existente en la Institución se muestran en el cuadro:

**Cuadro 01:** Equipos.

**Tabla 1 : Recursos Informáticos**

|  |  |
| --- | --- |
| **RECURSOS INFORMATICOS** | **CANTIDAD** |
| **Servidor** | 1 |
| **Computadoras de Escritorio** | 3 |
| *I5* | 2 |
| **Computadora Portátil** | 1 |
| **Impresoras** | 1 |
| **Equipos de Comunicaciones** | 3 |
| **Router** | 1 |
| **Firewall** | 1 |
| **Switch** | 1 |

***Fuente: ODEI TACNA***

* + 1. FACTIBILIDAD OPERATIVA
* La necesidad y deseo de un cambio en el sistema actual, expresada por los usuarios y el personal involucrado con el mismo, llevó a la aceptación de un nuevo sistema, que de una manera más sencilla y amigable, que cubra todos sus requerimientos, expectativas y proporciona la información en forma oportuna y confiable. Encontramos que algunos trabajadores tienen conocimientos en informática por lo que no se necesitara capacitarlas, pero a los demás se realizara la capacitación correspondiente para que puedan manipular el sistema con facilidad.
* En este proyecto se planificarán las diversas capacitaciones necesarias para el personal que lo necesite.
* Se realizará: **La Instalación del Sistema.**
* Se desarrollará un plan de capacitación personal y se expondrá de las bondades de este Sistema.
* Se monitoreará remotamente para proporcionar soporte técnico.
  + 1. FACTIBILIDAD ECONÓMICA

Como se mencionó anteriormente en el estudio de Factibilidad Técnica, la Empresa contaba con las herramientas necesarias para la puesta en marcha del sistema, por lo cual el desarrollo de la propuesta no requirió de una inversión inicial en Hardware.

A continuación, se presenta un cuadro de los costos que conlleva implantar el sistema:

**Tabla 2 : Cuadro Económico para el Sistema Operativo**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Componente / Actividad / Acción** | **Unidad** | **Cantidad** | **Costo Unitario** | **Total** |
| Analista de Sistemas y Testeador. Jefe Py. | Mes | 1 | 1,800.00 | 1,800.00 |
| Diseñador de Base de Datos | Mes | 1 | 1,500.00 | 1,500.00 |
| Programador 01 | Mes | 2 | 1,500.00 | 3,000.00 |
|  |  |  |  | **6,300.00** |

**Fuente:** (Elaboración Propia, 2018)

* + 1. FACTIBILIDAD LEGAL

La institución tiene en sus servidores los cuales están instalados software libre.

* 1. RECURSOS PRELIMINARES REQUERIDOS
     1. RECURSOS HUMANOS

Para el desarrollo del sistema se necesitará el siguiente personal, el cual debe cumplir un perfil de acuerdo a su cargo designado.

**Tabla 3: Recursos Humanos**

|  |
| --- |
| **Personal** |
| Analista de Sistemas y Testeador. Jefe Py. |
| Diseñador de Base de Datos |
| Programador 01 |
| Programador 02 |

**Fuente**:(Elaboración Propia, 2016)

**Analista de Sistemas y Testeador Jefe de Proyecto.**

Es la persona encargada de llevar el proyecto a buen término. A la vez cumplirá funciones de enlace con el cliente. Y analizara el proceso, en el cual utilizara diversas técnicas de recolección de información. Asimismo, modelara los diversos diagramas.

* Diagrama de Caso de uso
* Diagrama de Secuencia
* Diagrama de clases
* Diagrama de Componentes
* Diagrama de Despliegue

**Diseñador de Base de Datos.**

Es la persona encargada de diseñar la base de datos, de acuerdo con el análisis del proceso. Realizara los diseños siguientes:

* Diagrama Conceptual
* Diagrama Lógico
* Diagrama Físico

A la vez gestionara la base de datos en el servidor de SQL SERVER. Creando permisos de ingreso a la base y tomara todos los aspectos de seguridad que debe contar el sistema.

**Programador**

Personal que realizará el código fuente, el cual seguirá el estándar establecido de programación. Y seguirá los planos de software obtenido en el análisis.

* + 1. RECURSOS DE SOFTWARE

Se necesita el siguiente software:

* **PHP**

PHP, acrónimo recursivo en inglés de PHP Personal Hypertext processor (preprocesador de hipertexto), es un lenguaje de programación de propósito general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en un documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera el HTML resultante. (PHP Group, 2018)

PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. Puede ser usado en la mayoría de los servidores web al igual que en muchos sistemas operativos y plataformas sin ningún costo. (PHP Group, 2018)

Fue creado originalmente por Rasmus Lerdorf en el año 1995. Actualmente el lenguaje sigue siendo desarrollado con nuevas funciones por el grupo PHP.3​ Este lenguaje forma parte del software libre publicado bajo la licencia PHPv3\_01, es una licencia Open Source validada por Open Source Initiative. La licencia de PHP es del estilo de licencias BSD, esta licencia no tiene restricciones de copyleft" asociadas con GPL. (PHP Group, 2018)

* **MySQL**

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base datos de código abierto más popular del mundo,​ y una de las más populares en general junto a Oracle y Microsoft SQL Server, sobre todo para entornos de desarrollo web.

MySQL fue inicialmente desarrollado por MySQL AB (empresa fundada por David Axmark, Allan Larsson y Michael Widenius). MySQL AB fue adquirida por Sun Microsystems en 2008, y ésta a su vez fue comprada por Oracle Corporation en 2010, la cual ya era dueña desde 2005 de Innobase Oy, empresa finlandesa desarrolladora del motor InnoDB para MySQL.

Al contrario de proyectos como Apache, donde el software es desarrollado por una comunidad pública y los derechos de autor del código están en poder del autor individual, MySQL es patrocinado por una empresa privada, que posee el copyright de la mayor parte del código. Esto es lo que posibilita el esquema de doble licenciamiento anteriormente mencionado. La base de datos se distribuye en varias versiones, una Community, distribuida bajo la Licencia pública general de GNU, versión 2, y varias versiones Enterprise, para aquellas empresas que quieran incorporarlo en productos privativos. Las versiones Enterprise incluyen productos o servicios adicionales tales como herramientas de monitorización y asistencia técnica oficial. En 2009 se creó un fork denominado MariaDB por algunos desarrolladores (incluido algunos desarrolladores originales de MySQL) descontentos con el modelo de desarrollo y el hecho de que una misma empresa controle a la vez los productos MySQL y Oracle Database. (Wkipedia Mysql, 2017).

* **Servidor APACHE**

El servidor HTTP Apache es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual según la normativa RFC 2616. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Su nombre se debe a que alguien quería que tuviese la connotación de algo que es firme y enérgico pero no agresivo, y la tribu Apache fue la última en rendirse al que pronto se convertiría en gobierno de Estados Unidos, y en esos momentos la preocupación de su grupo era que llegasen las empresas y "civilizasen" el paisaje que habían creado los primeros ingenieros de internet. Además Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. En inglés, a patchy server (un servidor "parcheado") suena igual que Apache Server.

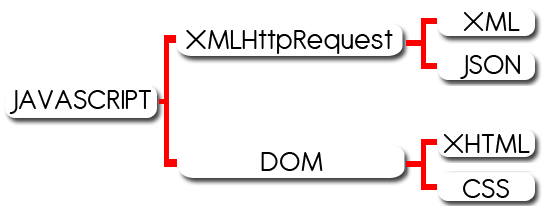
El servidor Apache es desarrollado y mantenido por una comunidad de usuarios bajo la supervisión de la Apache Software Foundation dentro del proyecto HTTP Server (httpd).

Apache presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración. (Wikipedia Apache, 2018).

* **AJAX**

AJAX (Asynchronous Javascript and XML) es una técnica de desarrollo web que, al combinar una serie de tecnologías independientes, nos permite intercambiar información entre el servidor y el cliente (un navegador web) de forma asíncrona. Como resultado, obtenemos una navegación ágil, rápida y dinámica; y también la posibilidad de realizar cambios sobre una web sin necesidad de actualizarla. Las tecnologías independientes que lo hacen posible son: (Freddy Vega, 2012)

**Ilustración 1:** Tecnologías independiente de AJAX



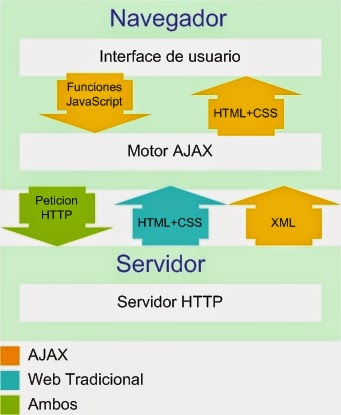
**Fuente:** (Freddy Vega, 2012). *Tecnologías independientes de AJAX*

* JAVASCRIPT: Es la base fundamental que une estas tecnologías
* XMLHttpRequest: Intercambio asíncrono
* XML: Manipulación e intercambio de información
* JSON: Alternativa de XML (Actualmente más usado que XML)
* DOM: Document Object Model
* XHTML y CSS: Estilos - Creación de una presentación de objetos

**Funcionamiento de AJAX:**

El usuario accede a la aplicación que es enviada por el servidor en formato HTML, JavaScript y CSS. Luego el código JavaScript de la aplicación pide al servidor los datos que quiere mostrar y este, ejecuta un código de lado de servidor que envía al navegador los datos en formato XML. (Concepcion, 2018).

**Ilustración 2:** Funcionamiento Ajax



**Fuente:** (Concepcion, 2018). *Funcionamiento Ajax*

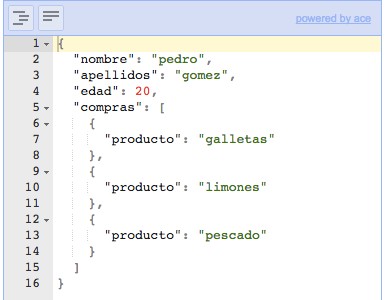
Cada vez que el usuario realiza una acción que significa mostrar unos datos, la capa JavaScript, repite la acción anterior de manera invisible al usuario y muestra los datos deseados.

* **AJAX Y JSON**

Y llegó JSON (JavaScript Object Notation), más ligero y con una notación más simple, y se convirtió en el formato más utilizado para el intercambio de datos cuándo se trabaja con Ajax. Además, con una ligera variación (JSONP) puede utilizarse en solicitudes Ajax entre diferentes dominios (cross-domain). (Juan Padial, 2016)

Formato Json

**Ilustración 3:** Formato Json



Fuente: (Álvarez, 2015). *Formato Json*

Para el Diseño Web se utilizara los siguientes lenguajes:

* **HTML**

Teniendo en cuenta a (Pergamino Virtual, 2002) el HTML, acrónimo inglés de HyperText Markup Language (lenguaje de marcas hipertextuales), lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web.

El HTML (Hyper Text Markup Language) es el lenguaje de programación con el que se escriben las páginas web. Es un lenguaje muy sencillo que permite confeccionar archivos de texto mostrándolo de forma estructurada y atractiva, con enlaces (hyperlinks) que hacen referencia a otros documentos o fuentes de información relacionada, y que permite incorporar archivos multimedia como gráficos, sonidos, imágenes, videos, etc. (La revista Tecnologica, 2015).

* **CSS**

CSS sirve para dar estilo y diseño a las páginas web — cambiar de tipo de letra, color, tamaño y espaciado del contenido, mostrar múltiples columnas, añadir animaciones y otras funcionalidades. Este módulo nos introducirá en el dominio de las reglas básicas de CSS, selectores, propiedades, reglas de escritura de CSS, aplicación de CSS a HTML, introducción de medidas, color y otras unidades en CSS, cascadas y herencias; y depuración de CSS. (Colaboradores Mozilla, 2005-2018).

Funcionamiento:

En este módulo comenzaremos con la base teórica, viendo qué es CSS y cómo el navegador transforma HTML en un [DOM](https://developer.mozilla.org/en-US/docs/Glossary/DOM), y cómo se aplica el CSS a las partes del DOM, algunos ejemplos muy básicos de sintaxis y el código necesario para incluir nuestro CSS en nuestra página web. (Colaboradores Mozilla, 2005-2018).

* **JAVA SCRIPT**

JavaScript fue creado inicialmente para "hacer que las páginas web estén vivas" .Los programas en este idioma se llaman scripts . Se pueden escribir directamente en el HTML y ejecutarse automáticamente a medida que se carga la página. Los scripts se proporcionan y ejecutan como un texto plano. No necesitan una preparación especial o una compilación para correr. (Ilya Kantor. Ilya Kantor, 2018).

* Importancia de utilizar JavaScript:

JavaScript es importante para el desarrollo web, ya que permite realizar todo tipo de aplicaciones cliente/servidor de gran uso en Internet, como salas de chat, reproductores multimedia online, galerías de imágenes, mini juegos. También, es ampliamente utilizado para crear efectos visuales que adicionan dinamismo a los sitios web y ofrecen la posibilidad de interactuar con este, para crear una sensación de comunicación entre el sitio web y el usuario. (Cespede, 2017).

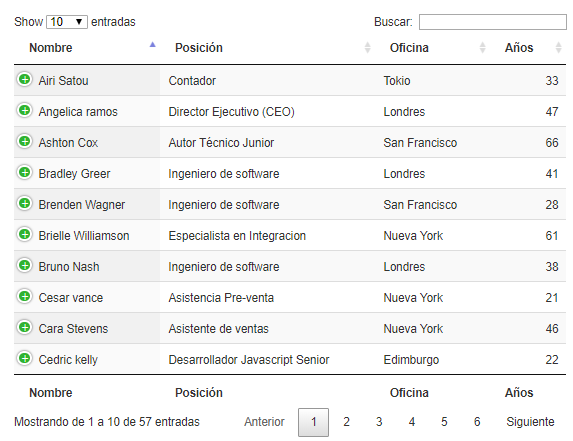
* **JQUERY**

jQuery es una biblioteca de JavaScript rápida, pequeña y con muchas funciones. Hace que cosas como la manipulación y manipulación de documentos HTML, el manejo de eventos, la animación y Ajax sean mucho más simples con una API fácil de usar que funciona en una multitud de navegadores. Con una combinación de versatilidad y extensibilidad, jQuery ha cambiado la forma en que millones de personas escriben JavaScript. Actualmente está en la versión 3.3.1. (jQuery, 2018)

* **DATATABLES**

DataTables es un complemento para la biblioteca jQuery Javascript. Es una herramienta altamente flexible, basada en los cimientos de la mejora progresiva, que agrega todas estas características avanzadas a cualquier tabla HTML. (SpryMedia Ltd, 2017)

**Ilustración 4:**Data Tables

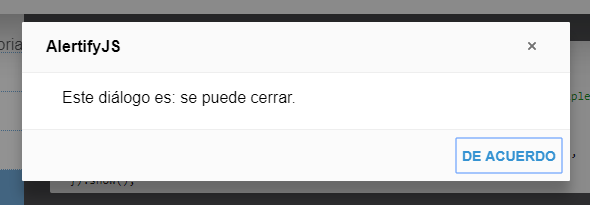


**Fuente**: (SpryMedia Ltd, 2017). *Data Tables*

* **ALERTEFY JS**

AlertifyJS es un marco javascript para desarrollar diálogos y notificaciones de navegadores bonitos. ( Mohammad Younes, 2018).

**Ilustración 5:**Ejemplo de Mensaje de Alerta (Alertefy)



**Fuente:** ( Mohammad Younes, 2018). Ejemplo de Mensaje de Alerta (Alertefy)

* **DISEÑO** **RESPONSIVE**

Se trata de redimensionar y colocar los elementos de la web de forma que se adapten al ancho de cada dispositivo permitiendo una correcta visualización y una mejor experiencia de usuario. Se caracteriza porque los layouts (contenidos) e imágenes son fluidos y se usa código media-queries de CSS3. (Academia Web, 2017).

**Ilustración 6:** Diseño Responsive



**Fuente:** (Academia Web, 2017). Diseño Responsive

**-CSS GRID**

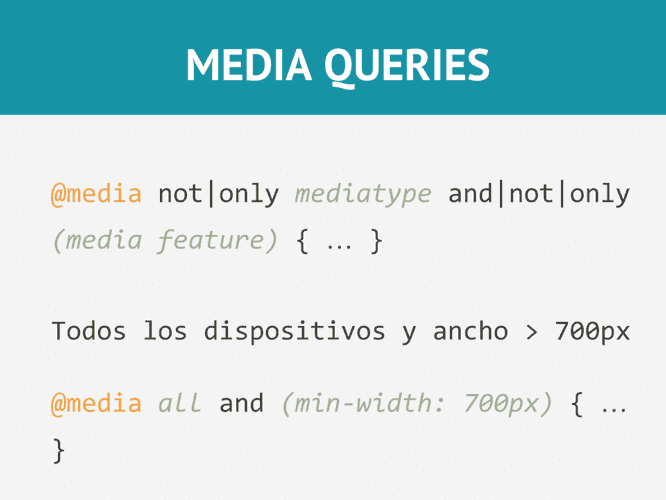
El Módulo de diseño de cuadrícula CSS GRID ofrece un sistema de diseño basado en cuadrícula, con filas y columnas, que facilita el diseño de páginas web sin tener que usar flotadores y posicionamiento. (W3Schools, 2017).

Método de uso de un concepto de cuadrícula para diseñar el contenido, proporcionando un mecanismo para que los autores dividan el espacio disponible para el diseño en columnas y filas utilizando un conjunto de comportamientos de tamaño predecibles. Incluye soporte para todas las grid-\*propiedades y la unidad. (Fridd Lensco, 2018).

**-CSS3 MEDIA QUERIES**

Método de aplicación de estilos basados ​​en la información de los medios. Incluye cosas como dimensiones de página y dispositivo. (Fridd Lensco, 2018).

**Ilustración 7:** Ejemplo Media Queries



**Fuente:** (Ricardo Prieto, 2017). Ejemplo Media Queries

Se necesita siguiente patrón de arquitectura de software.

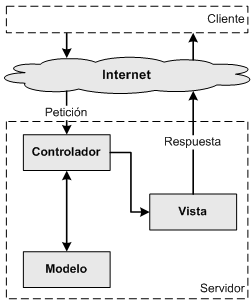
**Patrón Modelo Vista controlador**

El patrón Modelo – Vista – Controlador fue inventado en el contexto de Smalltak para realizar una separación entre la interfaz gráfica y el código del funcionamiento de una aplicación. Esta idea teórica afectó, de forma importante, a gran parte del código de Smalltalk y fue posteriormente aplicada a los lenguajes que están basados en objetos. (Junta de Anadalucia, 2016)

En el paradigma MVC, las entradas del usuario, los modelos del mundo exterior y la retroalimentación visual son explícitamente separados y manejados por tres tipos de objetos, cada uno especializado para un conjunto de tareas específicas. (Junta de Anadalucia, 2016)

El objetivo primordial del patrón es dar soporte a los modelos funcionales y mapas mentales de la información relevante para los usuarios, permitiendo un modelo que facilite la consulta y manejo de los mismos. La única manera de construir artefactos manejables es ayudar al usuario a construir modelos del sistema. Pero esto es imposible si el modelo mental no ha sido diseñado dentro del artefacto desde el principio. Intentar adicionar los modelos mentales del usuario cuando ya se ha avanzado en el desarrollo puede ser imposible. A continuación un gráfico que resume el patrón. (Junta de Anadalucia, 2016)

**Ilustración 8 : Componentes de Modelo Vista Controlador**



Fuente: (Wikipedia MVC, 2010)*Una típica colaboración entre los componentes de un MVC*

**Ventajas y desventajas del uso del patrón.**

*Se tienen muchas ventajas como:*

* La implementación se realiza de forma modular.
* Sus vistas muestran información actualizada siempre. El programador no debe preocuparse de solicitar que las vistas se actualicen, ya que este proceso es realizado automáticamente por el modelo de la aplicación.
* Cualquier modificación que afecte al dominio, como aumentar métodos o datos contenidos, implica una modificación sólo en el modelo y las interfaces del mismo con las vistas, no todo el mecanismo de comunicación y de actualización entre modelos.
* Las modificaciones a las vistas no afectan al modelo de dominio, simplemente se modifica la representación de la información, no su tratamiento.
* MVC está demostrando ser un patrón de diseño bien elaborado pues las aplicaciones que lo implementan presentan una extensibilidad y una mantenibilidad únicas comparadas con otras aplicaciones basadas en otros patrones.

*Como desventajas tenemos:*

* Para desarrollar una aplicación bajo el patrón de diseño MVC es necesario una mayor dedicación en los tiempos iniciales del desarrollo. Normalmente el patrón exige al programador desarrollar un mayor número de clases que, en otros entornos de desarrollo, no son necesarias. Sin embargo, esta desventaja es muy relativa ya que posteriormente, en la etapa de mantenimiento de la aplicación, una aplicación MVC es mucho más mantenible, extensible y modificable que una aplicación que no lo implementa.
* MVC requiere la existencia de una arquitectura inicial sobre la que se deben construir clases e interfaces para modificar y comunicar los módulos de una aplicación. Esta arquitectura inicial debe incluir, por lo menos, un mecanismo de eventos para poder proporcionar las notificaciones que genera el modelo de aplicación; una clase Modelo, otra clase Vista y una clase Controlador genéricas que realicen todas las tareas de comunicación, notificación y actualización que serán luego transparentes para el desarrollo de la aplicación.
* MVC es un patrón de diseño orientado a objetos por lo que su implementación es sumamente costosa y difícil en lenguajes que no siguen este paradigma.

**Componentes de Modelo vista controlador**

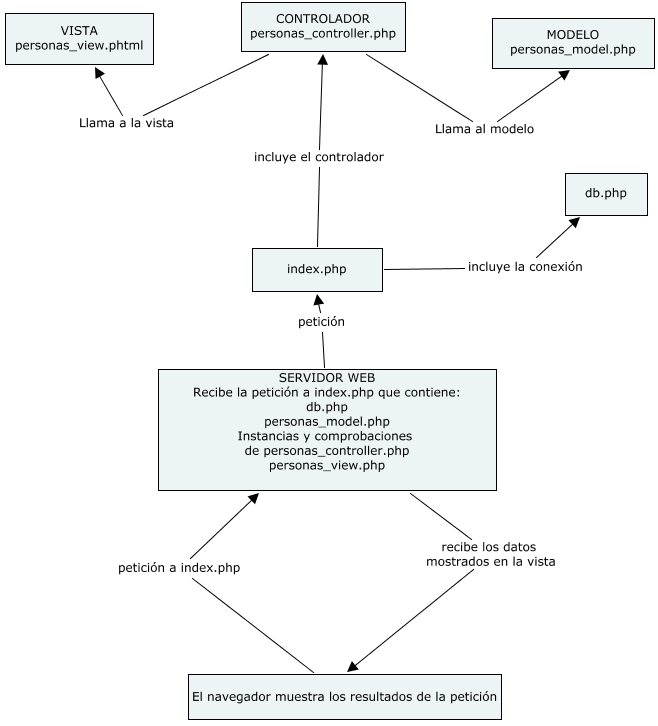
De manera genérica, los componentes de MVC se podrían definir como sigue: (Wikipedia L. e., 2018)

**El Modelo:** Es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio). Envía a la 'vista' aquella parte de la información que en cada momento se le solicita para que sea mostrada (típicamente a un usuario). Las peticiones de acceso o manipulación de información llegan al 'modelo' a través del 'controlador'.

**El Controlador:** Responde a eventos (usualmente acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la información (por ejemplo, editar un documento o un registro en una base de datos). También puede enviar comandos a su 'vista' asociada si se solicita un cambio en la forma en que se presenta el 'modelo' (por ejemplo, desplazamiento o scroll por un documento o por los diferentes registros de una base de datos), por tanto se podría decir que el 'controlador' hace de intermediario entre la 'vista' y el 'modelo'.

**La Vista:** Presenta el 'modelo' (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la interfaz de usuario), por tanto requiere de dicho 'modelo' la información que debe representar como salida.

**Ilustración 9:** Modelo vista Controlador en PHP Nativo



*Fuente:* (Victor Robles, 2015) *MVC en PHP nativo*

* + 1. RECURSOS DE HARDWARE

Se necesita equipos de cómputo para la implantación del software los cuales se describen a continuación:

* + - 02 equipos de cómputo de procesador I5 con memoria de 4 Gb, Disco Duro de 1 Tera.
    - El cual un computador será para un servidor de desarrollo.
    - 01 impresora.
    - Switch de 8 puertos.
    - Cableado estructurado
  1. COSTOS

**Tabla 4:** Cuadro Economico para Sistema Operativo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Componente / Actividad / Acción** | **Unidad** | **Cantidad** | **Costo Unitario** | **Total** |
| Analista de Sistemas y Testeador. Jefe Py. | Mes | 1 | 1,800.00 | 1,800.00 |
| Diseñador de Base de Datos | Mes | 1 | 1,500.00 | 1,500.00 |
| Programador 01 | Mes | 2 | 1,500.00 | 3,000.00 |
|  |  |  |  | **6,300.00** |

Fuente: (Elaboración Propia, 2018)

* 1. BENEFICIOS
* Procesar la información de las asistencias de forma rápida y poder realizar los diferentes reportes para la toma de decisiones.
* Gestionar el registro de asistencia de docentes y practicantes.
* Contar con un sistema fiable, de alta escalabilidad y gratuito.
  1. TÉCNICAS PARA LA OBTENCIÓN DE INFORMACIÓN

Se ha utilizado para este trabajo las técnicas de Recolección de Datos las cuales fueron:

* Las Entrevistas.
* La Observación Directa.
* Documentación.

**Entrevistas. -** Se ha conversado para la obtención de información con el jefe de la Oficina de Informática.

**DESCRIBIREMOS ALGUNAS PREGUNTAS QUE SE HICIERON EN LA ENTREVISTA AL PERSONAL DE LA EMPRESA.**

Preguntas de la Entrevista

1. ¿Desearía UD. ¿Con un Software que agilice más el trabajo?

2. ¿Cuánto tiempo demora en realizar el registro diario de asistencia?

3. ¿Cuenta con la infraestructura en hardware para la implementación del sistema?

4. ¿El registro de asistencia mensual en cuanto tiempo se demora?

5. ¿El personal cuenta con conocimientos de Computación?

**La Observación Directa**

Se desarrolló esta técnica para seguir el procedimiento de registro de asistencia de entrada y salida de los diferentes cursos.

Se desarrolló esta técnica previamente coordinando los horarios y pidiendo permiso a las autoridades pertinentes.

Esto conllevo a precisar y tener nuevos requerimientos y se diseñó un esquema del proceso,

**Análisis** **Documentación**

Se pidió la documentación pertinente como registro de asistencia, listado de cursos, listado de docentes, etc. los cuales eran de mucha importancia para el desarrollo del sistema y ver aspectos del desarrollo siguiendo la norma de la empresa.

1. CAPITULO IV
2. PLAN GENERAL DEL PROYECTO
   1. DESCRIBIR LA ESTRATEGIA DE DESARROLLO PARA EL SISTEMA

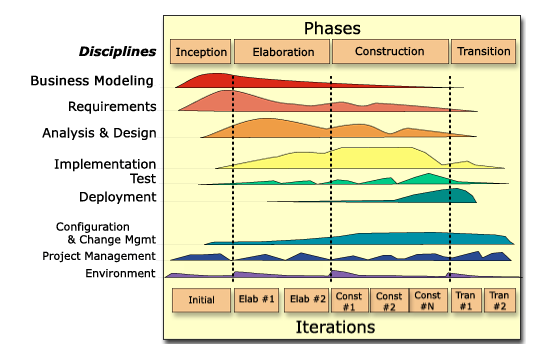
El estudio que se realizó para llevar a cabo este trabajo, se puede definir como investigación y desarrollo. Este tipo de estudios se llevan a cabo para mejorar una situación dada en un contexto, en el que se realizan actividades sistemáticas de investigación de campo, que sirven para el desarrollo de nuevo conocimiento.

Este proyecto siguió esta metodología para diseñar una herramienta adecuada para un adecuado control de asistencias de los empleados del ODEI.

También se ha empleado la metodología para desarrollar el software el cual es el RUP. Proceso Unificado de Rational es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

**Ilustración 10:** Fases del método RUP



**Fuente:** (Process, 2002) Rational Unified Process Overview

El ciclo de vida consiste en una serie de ciclos, cada uno de los cuales produce una nueva versión del producto, cada ciclo está compuesto por fases y cada una de estas fases está compuesta por un número de iteraciones, estas fases son: (CHACÓN, 2006)

Concepción, Inicio o Estudio de oportunidad

* Define el ámbito y objetivos del proyecto
* Se define la funcionalidad y capacidades del producto

Elaboración

* Tanto la funcionalidad como el dominio del problema se estudian en profundidad
* Se define una arquitectura básica
* Se planifica el proyecto considerando recursos disponibles

Construcción

* El producto se desarrolla a través de iteraciones donde cada iteración involucra tareas de análisis, diseño e implementación
* Las fases de estudio y análisis sólo dieron una arquitectura básica que es aquí refinada de manera incremental conforme se construye (se permiten cambios en la estructura)
* Gran parte del trabajo es programación y pruebas
* Se documenta tanto el sistema construido como el manejo del mismo.
* Esta fase proporciona un producto construido junto con la documentación.

Transición

* Se libera el producto y se entrega al usuario para un uso real
* Se incluyen tareas de marketing, empaquetado atractivo, instalación, configuración, entrenamiento, soporte, mantenimiento, etc.
* Los manuales de usuario se completan y refinan con la información anterior
* Estas tareas se realizan también en iteraciones
  1. PLAN GENERAL DE DESARROLLO DEL PROYECTO
     1. DEFINIR EL PANORAMA DEL PROYECTO

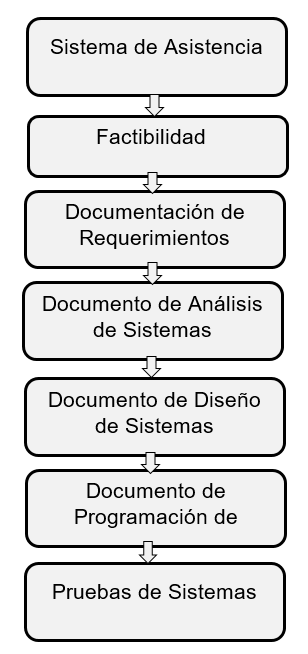
El sistema de Asistencia de la Institución ODEI TACNA, está desarrollado con herramientas de última generación, y con una arquitectura escalable con un alto rendimiento, utiliza una capa de servicio el cual puede conectarse a cualquier otro sistema. Este sistema servirá a la ODEI TACNA. Ha poder guardar su Información de ventas de productos y realizar diversos reportes para la toma de decisiones de los productos y ventas.

* + 1. ESTRUCTURA DE ACTIVIDADES (Usar el MS-Project)

Una Estructura de Descomposición del Trabajo o EDT, también conocida por su nombre en inglés Work Breakdown Structure o WBS, es en gestión de proyectos una descomposición jerárquica orientada al entregable, del trabajo a ser ejecutado por el equipo de proyecto, para cumplir con los objetivos de éste y crear los entregables requeridos, con cada nivel descendente de la EDT representando una definición con un detalle incrementado del trabajo del proyecto. La EDT es una herramienta fundamental en la gestión de proyectos.

El propósito de una EDT es organizar y definir el alcance total aprobado del proyecto según lo declarado en la documentación vigente. Su forma jerárquica permite una fácil identificación de los elementos finales, llamados "Paquetes de Trabajo". Se trata de un elemento exhaustivo en cuanto al alcance del proyecto, la EDT sirve como la base para la planificación del proyecto. Todo trabajo a ser hecho en el proyecto debe poder rastrear su origen en una o más entradas de la EDT.2

**Ilustración 11:** Estructura de Descomposicion del Trabajo



**Fuente**:(Elaboración Propia, 2018)

* + - 1. DIAGRAMA DE GANTT CON ASIGNACIÓN DE RECURSOS

**Tabla 5:** Cronograma del Proyecto (Diagrama de Gantt).

|  |  |  |  |
| --- | --- | --- | --- |
| **NOMBRE DE LA TAREA** | **COMIENZO** | **FIN** | **DURACION** |
| **Investigación Preliminar** | **01/09/2018** | **13/09/2018** | **13 días** |
| Planificación | 01/09/2018 | 05/09/2018 | 5 días |
| Estudios de factibilidad | 06/09/2018 | 08/09/2018 | 3 días |
| Desarrollo del presupuesto | 09/09/2018 | 11/09/2018 | 3 días |
| Evaluación | 12/09/2018 | 13/09/2018 | 2 días |
| **Determinación de Requerimientos** | **14/09/2018** | **26/09/2018** | **13 días** |
| Revisión de documentos | 14/09/2018 | 20/09/2018 | 7 días |
| Definición de requerimientos | 21/09/2018 | 26/09/2018 | 6 días |
| **Análisis del sistema** | **27/09/2018** | **04/10/2018** | **9 días** |
| Desarrollo del proyecto | 27/09/2018 | 04/10/2018 | 9 días |

**Fuente:** (Elaboración Propia, 2018). Cronograma del Proyecto

CAPITULO V

1. ANÁLISIS DEL SISTEMA
   1. ESPECIFICACIÓN DE REQUERIMIENTOS
      1. REQUERIMIENTOS BÁSICOS
         1. REQUERIMIENTOS FUNCIONALES

**Tabla 6:** Requerimientos Funcionales

|  |  |
| --- | --- |
| **REQ.** | **DESCRIPCIÓN** |
| **REQ-01** | Gestionar la Asistencia de entrada y salida de docente y Practicantes. |
| **REQ-02** | Gestionar a los empleados en este caso solamente gestionara al personal docente y practicante. |
| **REQ-03** | Gestionar los horarios establecidos por la institución de los cursos para brinda. |
| **REQ-04** | Gestionar los cursos. |
| **REQ-05** | Gestionar los reportes de asistencia. |
| **REQ-06** | Realizar validaciones de ingreso de información. |

**Fuente:** (Elaboración Propia, 2018) Requerimientos Funcionales

* + - 1. REQUERIMIENTOS NO FUNCIONALES

Contribuir a las capacitaciones permanentes de los usuarios a través de la base de conocimiento.

Se desarrollará mediante un plan de capacitación que constará de sesiones personales y por grupo de acuerdo al nivel de manejo del software.

A la vez se le motivara al personal de la empresa en el manejo del sistema.

Se le debe otorgar al Personal Capacitado:

* El manual de Usuario
* El manual de seguridad de cuenta de Usuario y las políticas del manejo.
* Certificado de Capacitación.
  + 1. REQUERIMIENTOS GENERALES DE LA ORGANIZACIÓN

Que el sistema tenga una interfaz amigable, fácil de manejar. Asimismo, debe registrar los datos de los diferentes análisis.

* + 1. REQUERIMIENTOS DEL USUARIO

**Tabla 7:** Requerimientos del Usuario

|  |  |
| --- | --- |
| **REQ.** | **DESCRIPCIÓN** |
| REQ-01 | Manejo fácil del sistema web en base al registro de asistencia. |
| REQ-02 | Consultas variadas en base de registro de asistencia. |
| REQ-03 | Diseño de formularios web amigables con colores a nivel de la institución. |

**Fuente:** (Elaboración Propia, 2018). *Requerimientos del Usuario*

Usuario de Línea (NIVEL 1):

* El usuario podrá acceder a las interfaces, que le permitan realizar gestión sobre:
* Registrar personal Docente y practicante.
* Registrar horario de cursos.
* Registrar los permisos a diversos cursos.
* Podrá realizar las siguientes transacciones:
* Registrar datos nuevos.
* Eliminar un registro existente.
* Modificar un registro existente.
* Consultas a los datos: por pantallas y/o impresas, de acuerdo a requerimiento.
* El usuario podrá acceder a los reportes de gestión inherentes a las actividades que realiza, de acuerdo a los requerimientos definidos, tales como:
* Reporte de asistencia por meses, trimestres, semestres, anual.
* Reporte de consolidados de asistencia anuales.
* Reporte de Entradas y Salidas por mes trimestral, semestral, anual.
  1. PROCESOS DEL NEGOCIO

Se obtiene la información respectiva a fin de conocer nuestro negocio y establecer las reglas de negocio obteniendo así los distintos requerimientos del usuario.

A la vez se obtuvo los procesos de dicha empresa, los cuales son los siguientes:

**Tabla 8:** Procesos de la Empresa

|  |  |
| --- | --- |
| **CÓDIGO** | **NOMBRE DEL PROCESO** |
| **GE-01** | Registrar Asistencia de Docentes |
| **GE-02** | Registrar Asistencia de Practicante |
| **GE-03** | Otorgar permiso de curso. |

**Fuente**:(ODEI Tacna, 2018). *Procesos de la Empresa*

* 1. REGLAS DEL NEGOCIO
* El sistema registrara la asistencia del docente, teniendo en cuanta las tolerancias del caso.
* El sistema manejara el control de asistencia mediante el horario establecido.
* El Sistema debe tener toda la seguridad respectiva sobre el ingreso de información.
  1. PAQUETES
     1. DESCRIPCIÓN GENERAL DE CADA PAQUETE

El diagrama de paquetes se muestra los contenedores de las clases. Los cuales se distribuirán y se relacionarán para el funcionamiento del sistema.

Se ha Dividido este Sistema en dos paquetes los cuales son Control de Asistencia y Gestión de Usuarios. Los cuales contendrán las diversas clases que controlarán los procesos de entrada y salida.

**Tabla 9:** Descripción de Paquetes

|  |
| --- |
| DESCRIPCION DE PAQUETES. |
| Paquete: Control de Asistencia. Este paquete se encargará de ver el control de asistencia de entrada y salida de los diferentes cursos. |
| Paquete: Gestión de Usuarios. Este paquete se encargará administrar los usuarios, desde su creación y sus permisos respectivos. |

**Fuente:**(Elaboración Propia, 2018)

* + 1. CONSTRUCCIÓN DEL DIAGRAMA DE PAQUETES

**Diagrama 1:** Diagrama de Paquetes



**Fuente**: (Elaboración Propia, 2018). *Diagrama de Paquetes*

* 1. DIAGRAMAS DE CASOS DE USO
     1. DEFINICIÓN DE CASOS DE USO
        1. DESCRIPCIÓN DE ACTORES

Administrador. - Es la persona encargada de registrar los horarios, cursos y todo lo que necesite el sistema de información para funcionar.

Docente. - Es la persona que se registrara las entradas y salidas mediante el sistema de asistencia.

* + - 1. DESCRIPCIÓN DE CASOS DE USO

**CU-001:** Gestionar registro del Empleado.

Registrar los datos del empleado.

**Diagrama 2:** Caso de Uso Registrar Empleado



**Fuente: (**Elaboración Propia, 2018**)**. Caso de Uso Registrar Empleado

Documentación del **CU-001:** Gestionar Registro del Empleado.

**Tabla 10:** Documentación CU Gestionar Registro del Empleado

| **GESTIONAR REGISTRO DEL EMPLEADO** | | | |
| --- | --- | --- | --- |
| **Tipo** | Obligatorio | | |
| **Versión** | v.1.0 | | |
| **Autor** | Administrador | | |
| **Actores** | Administrador | | |
| **Iteración** | Análisis | | |
| **Descripción** | **El caso de uso se inicia cuando** el administrador decide registrar al Empleado. | | |
| **Referencias** | Entrevista de Levantamiento de Requerimientos | | |
| **Precondiciones** | Sesión Iniciado del Usuario. | | |
| **Flujo normal de eventos “GESTIONAR REGISTRO”** | | | |
| **FLUJO DE EVENTOS** | | | |
| **Usuario** | | **Sistema** | |
| 1. Ingresar al Formulario Web de Gestión de registro. | | 1. El sistema nos muestra el formulario registrar Empleado con los siguientes campos deshabilitados.  * Nombre. * Apellido * Dni. * Dirección. * Una lista desplegable para seleccionar Cargo * Botones de Guardar, Modificar, Cancelar. * Una Grilla para mostrar Datos. | |
| 1. Presiona botón “**Guardar**” | | 1. El formulario guardará los campos ingresados si son correctos le mostrará el siguiente mensaje: “**Sistema** **Registrado Satisfactoriamente”.** | |
| 1. Presiona Cancelar | | 1. El sistema cancelara la acción que esté realizando y deshabilitada los campos. | |
| 1. El usuario modifica algún registro y Presiona el Botón Modificar | | 1. El sistema le permitirá modificar todos los campos a excepción del código, si la modificación es correcta el sistema muestra el siguiente mensaje: **Sistema, Modificado Satisfactoriamente**. | |
| 1. Presiona en la opción de Salir. | | 1. El sistema cierra el formulario. | |
| **Flujo de Excepción –FE1** | | | |
| 1. Presiona guardar con el campo Dni vacío. | | | 1. El sistema le muestra el mensaje de **"Introduzca Dni del Empleado"** |

**Fuente:(**Elaboración Propia, 2019)

**CU-002:** Gestionar registro del Asistencia.

Registrar Asistencia

**Diagrama 3:** Caso de Uso Registrar Asistencia



**Fuente:** (Elaboración Propia, 2018). *Caso de Uso Registrar Asistencia*

Documentación del **CU-002:** Gestionar Registro de Asistencia.

**Tabla 11:** Documentación CU Gestionar Registro Asistencia

| **GESTIONAR REGISTRO DE ASISTENCIA** | | |
| --- | --- | --- |
| **Tipo** | Obligatorio | |
| **Versión** | v.1.0 | |
| **Autor** | Administrador | |
| **Actores** | Administrador | |
| **Iteración** | Análisis | |
| **Descripción** | **El caso de uso se inicia cuando** el Empelado decide registrar asistencia. | |
| **Referencias** | Entrevista de Levantamiento de Requerimientos | |
| **Precondiciones** |  | |
| **Flujo normal de eventos “GESTIONAR REGISTRO”** | | |
| **FLUJO DE EVENTOS** | | |
| **Usuario** | | **Sistema** |
| 1. Ingresar al Formulario de Gestión de registro. | | 1. El sistema nos muestra el formulario registrar Asistencia con los siguientes campos deshabilitados.  * Fecha. * Horario * Fecha Entrada. * Fecha salida * Estado. * Curso. * Botones de Guardar, Modificar, Cancelar. * Una Grilla para mostrar Datos. |
| 1. Presiona botón “**Registrar Asistencia**” | | 1. El formulario guardará la fecha y hora de ingreso mensaje: “**Docente Registrado su asistencia satisfactoriamente”.** |

**Fuente: (**Elaboración Propia, 2018**).** *Documentación CU Gestionar Registro Asistencia*

**CU-003:** Gestionar registro de Horario.

Registrar Horario

**Diagrama 4:** Caso de Uso Registro Horario



**Fuente:** (Elaboración Propia, 2018)

Documentación del **CU-002:** Gestionar Registro de Horario.

**Tabla 12:** Documentación CU Gestionar Registro Horario

| **GESTIONAR REGISTRO DE HORARIO** | | |
| --- | --- | --- |
| **Tipo** | Obligatorio | |
| **Versión** | v.1.0 | |
| **Autor** | Administrador | |
| **Actores** | Administrador | |
| **Iteración** | Análisis | |
| **Descripción** | **El caso de uso se inicia cuando** el Administrador decide registrar horario. | |
| **Referencias** | Entrevista de Levantamiento de Requerimientos | |
| **Precondiciones** | Debe haberse registrado los cursos a dictar y establecido los grupos. | |
| **Flujo normal de eventos “GESTIONAR REGISTRO HORARIO”** | | |
| **FLUJO DE EVENTOS** | | |
| **Usuario** | | **Sistema** |
| 1. Ingresar al Formulario de Gestión de registro de Horario. | | 1. El sistema nos muestra el formulario registrar Horario con los siguientes campos deshabilitados.  * Cursos * Hora entrada * Hora Salida. * Una lista desplegable para seleccionar Nombre del Empleado. * Botones de Guardar, Modificar, Cancelar. * Una Grilla para mostrar Datos. |
| 1. Presiona botón “**Guardar**” | | 1. El formulario guardará los campos ingresados si son correctos le mostrará el siguiente mensaje: “**Sistema** **Registrado Satisfactoriamente”.** |
| 1. Presiona Cancelar | | 1. El sistema cancelara la acción que esté realizando y deshabilitada los campos. |
| 1. El usuario modifica algún registro y Presiona el Botón Modificar | | 1. El sistema le permitirá modificar todos los campos a excepción del código, si la modificación es correcta el sistema muestra el siguiente mensaje: **Sistema, Modificado Satisfactoriamente**. |
| 1. Presiona en la opción de Salir. | | 1. El sistema cierra el formulario. |

**Fuente:** (Elaboración Propia). *Documentación CU Gestionar Registro Horario*

**CU-004:** Gestionar Registro de permiso de asistencia.

Registrar permiso de asistencia.

**Diagrama 5:** Caso de Uso Registrar Permiso Asistencia



**Fuente:** (Elaboración Propia, 2018). *Caso de Uso Registrar Permiso Asistencia*

Documentación del **CU-004:** Gestionar Registro de Permiso de asistencia.

**Tabla 13:** Documentación CU Gestionar Registro de Permiso de asistencia

| **GESTIONAR REGISTRO DE PERMISO DE ASISTENCIA** | | | |
| --- | --- | --- | --- |
| **Tipo** | Obligatorio | | |
| **Versión** | v.1.0 | | |
| **Autor** | Administrador | | |
| **Actores** | Empleado | | |
| **Iteración** | Análisis | | |
| **Descripción** | **El caso de uso se inicia cuando** el Empelado decide otorgar permiso. | | |
| **Referencias** | Entrevista de Levantamiento de Requerimientos | | |
| **Precondiciones** |  | | |
| **Flujo normal de eventos “GESTIONAR PERMISO DE ASISTENCIA”** | | | |
| **FLUJO DE EVENTOS** | | | |
| **Usuario** | | **Sistema** | |
| 1. Ingresar al Formulario de Gestión de registro. | | 1. El sistema nos muestra el formulario registrar Permisos con los siguientes campos deshabilitados.  * Fecha inicial. * Fecha final * Dia. * Nombre * Una lista desplegable para seleccionar Empleado. * Botones de Guardar, Modificar, Cancelar. * Una Grilla para mostrar Datos. | |
| 1. Presiona botón “**Guardar**” | | 1. El formulario guardará los campos ingresados si son correctos le mostrará el siguiente mensaje: “**Sistema** **Registrado Satisfactoriamente”.** | |
| 1. Presiona Cancelar | | 1. El sistema cancelara la acción que esté realizando y deshabilitada los campos. | |
| 1. El usuario modifica algún registro y Presiona el Botón Modificar | | 1. El sistema le permitirá modificar todos los campos a excepción del código, si la modificación es correcta el sistema muestra el siguiente mensaje: **Sistema, Modificado Satisfactoriamente**. | |
| 1. Presiona en la opción de Salir. | | 1. El sistema cierra el formulario. | |
| **Flujo de Excepción –FE1** | | | |
| 1. Presiona guardar con el campo Permiso vacío. | | | 1. El sistema le muestra el mensaje de **"Introduzca Todos los datos"** |

Fuente: (Elaboración Propia, 2018). *Documentación CU Gestionar Registro de Permiso de asistencia*

* + 1. CONSTRUCCION DE DIAGRAMAS DE SECUENCIA.

En este proyecto se ha desarrollado los diferentes diagramas de Secuencia para ver las iteraciones entre los objetos del sistema.

**Tabla 14**: Diagrama de Secuencia

|  |  |
| --- | --- |
| **N°** | **Diagrama de Secuencia** |
| **01** | Registrar Asistencia. |
| **02** | Registrar Empleado. |
| **03** | Registrar Horario. |

Fuente: (Elaboración Propia, 2018). *Diagrama de Secuencia*

REGISTRAR ASISTENCIA

**Diagrama 6**: Diagrama de Secuencia Registrar Asistencia



Fuente: (Elaboración Propia, 2016). *Diagrama de Secuencia Registrar Asistencia*

Este diagrama de secuencia se visualiza como se va a registrar la asistencia. Como se muestra en el diagrama de secuencia tienen un conjunto de objetos como son VAsistencia.php, AsistenciaController.php, asistenciaModel.Php y BD.

REGISTRAR EMPLEADO

**Diagrama 7:** Diagrama de Secuencia Registrar Empleado



Fuente: (Elaboración Propia, 2018). *Diagrama de Secuencia Registrar Empleado*

Este diagrama de secuencia se encarga de registrar los datos del empleado. Tiene los siguientes objetos VEmpleado.php, EmpleadoController.php, EmpleadoModel.php, yBD.

REGISTRAR HORARIO.

**Diagrama 8:** Diagrama de Secuencia Registrar Horario



Fuente: (Elaboracion Propia, 2018). Diagrama de Secuencia Registrar Horario

Este diagrama de secuencia se encarga de registrar el Horario.

CAPITULO VI

DISEÑO DEL SISTEMA

* 1. IDENTIFICACIÓN DE OBJETOS

**Clase**

**ClsCargo.-**

Esta clase es la encargada de realizar las operaciones de registro del cargo del empleado.

**ClsEmpleado.-**

Esta clase es la encargada de realizar las operaciones de registro de los empleados.

**ClsHorario.-**

Esta clase es la encargada de realizar las operaciones transaccionales de los ingresos de las diferentes compras.

**ClsCurso.-**

Esta clase es la encargada de realizar las operaciones transaccionales de los cursos.

**ClsRegistro.-**

Esta clase es la encargada de realizar las operaciones transaccionales de registro.

**ClsUsuario.-**

Esta clase es la encargada de realizar las operaciones transaccionales de los diferentes usuarios.

* 1. MODELO DEL NEGOCIO
     1. DESCRIBIR CÓMO FUNCIONA O CÓMO FUNCIONARÍA EL SISTEMA CON UN TEXTO EN FORMA PRECISA

El sistema de asistencia, controla el procedimiento de registrar la asistencia de entrada y salida de los cursos que dicta los docentes. EL sistema tiene un formulario web que permita registrar las entradas y salidas de asistencia.

* + 1. IDENTIFICACIÓN DE LOS TIPOS DE RELACIONES

**Tabla 15:** Tipo de Relación Clase Usuario

|  |  |
| --- | --- |
| **ClsUsuario** | |
| **Responsabilidades** | **Colaboradores** |
| * **Registrar los Usuario.** * **Generar listado de búsqueda de los Usuarios.** | * Administrador |

**Fuente:** (Elaboración Propia, 2018)

**Tabla 16:** Tipo de Relación Clase Cargo

|  |  |
| --- | --- |
| **ClsCargo** | |
| **Responsabilidades** | **Colaboradores** |
| * **Consultar mediante parámetros el cargo.** * **Listado de cargos.** | * Administrador |

**Fuente:** (Elaboración Propia, 2018)

**Tabla 17:** Tipo de Relación Clase Empleado

|  |  |
| --- | --- |
| **ClsEmpleado** | |
| **Responsabilidades** | **Colaboradores** |
| * **Registrar los empleados.** * **Generar listado de búsqueda de los Empleados.** * **Cumple funciones como modificar, Activar y desactivar registros.** * **Lista de empleados.** | * Administrador del Sistema. |

**Fuente:** (Elaboración Propia, 2018)

**Tabla 18:** Tipo de Relación Clase Horario

|  |  |
| --- | --- |
| **ClsHorario** | |
| **Responsabilidades** | **Colaboradores** |
| * **Encargada de guardar los datos del Horario.** * **Activa y Desactiva registro de Horario.** * **Cumple funciones como modificar, Activar y desactivar registros.** * **Lista el horario.** | * Administrador |

**Fuente:** (Elaboración Propia, 2018)

**Tabla 19:** Tipo de Relación Clase Registro

|  |  |
| --- | --- |
| **ClsRegistro** | |
| **Responsabilidades** | **Colaboradores** |
| * **Registra los datos, fecha, horas de salidas-entradas y estados.** * **Cumple funciones como modificar, Activar y desactivar registros.** | * Administrador |

**Fuente:** (Elaboración Propia, 2018)

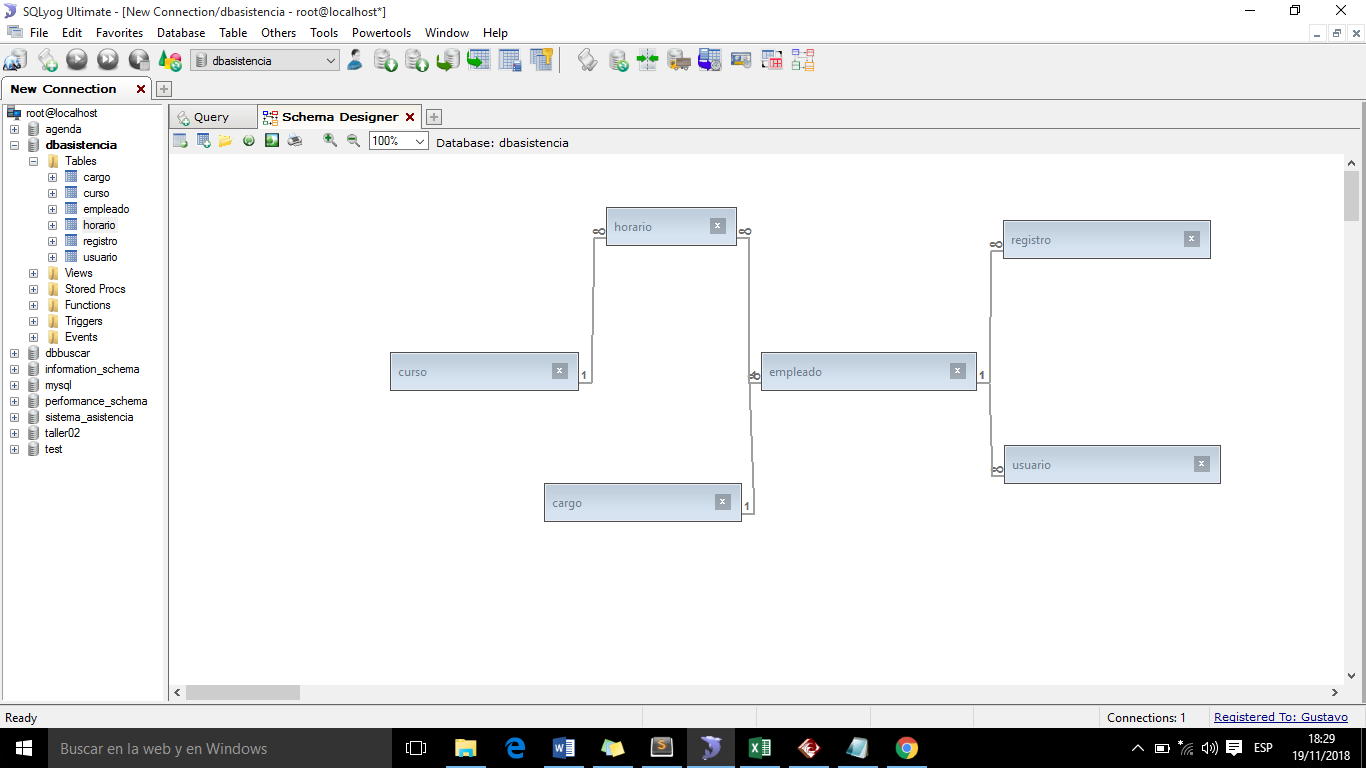
**Tabla 20:** Tipo de Relación Clase Curso

|  |  |
| --- | --- |
| **ClsCurso** | |
| **Responsabilidades** | **Colaboradores** |
| * **Registra los diferentes cursos** * **Cumple funciones como modificar, Activar y desactivar registros.**   **Listado de cursos.** | * Administrador |

**Fuente:** (Elaboración Propia, 2018)

* + 1. MODELO CONCEPTUAL DEL NEGOCIO

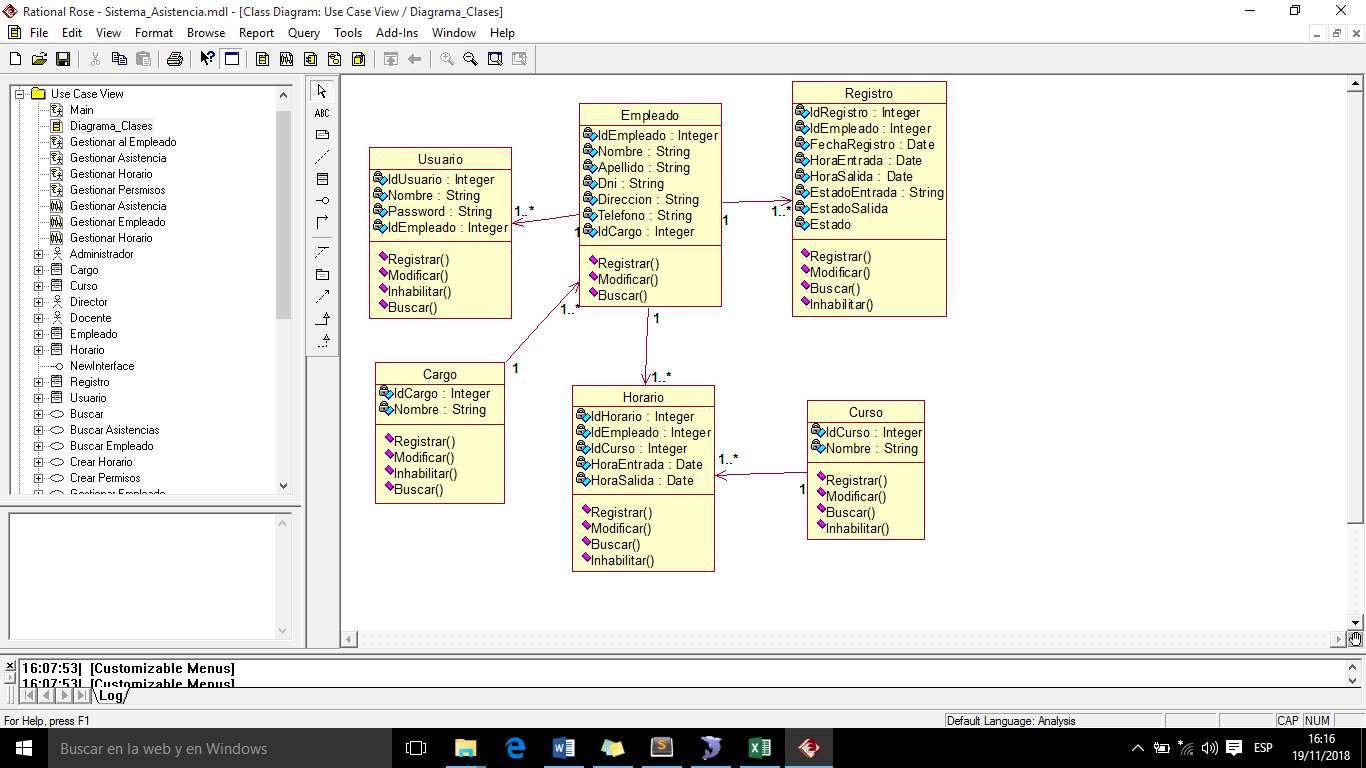
**Ilustración 12**: Modelo Conceptual



Fuente: (Elaboración Propia, 2018). *Modelo Conceptual*

* 1. DISEÑO E IMPLANTACIÓN DE LA BASE DE DATOS
     1. GENERACIÓN DEL MODELO DE DATOS A PARTIR DEL DIAGRAMA DE CLASES

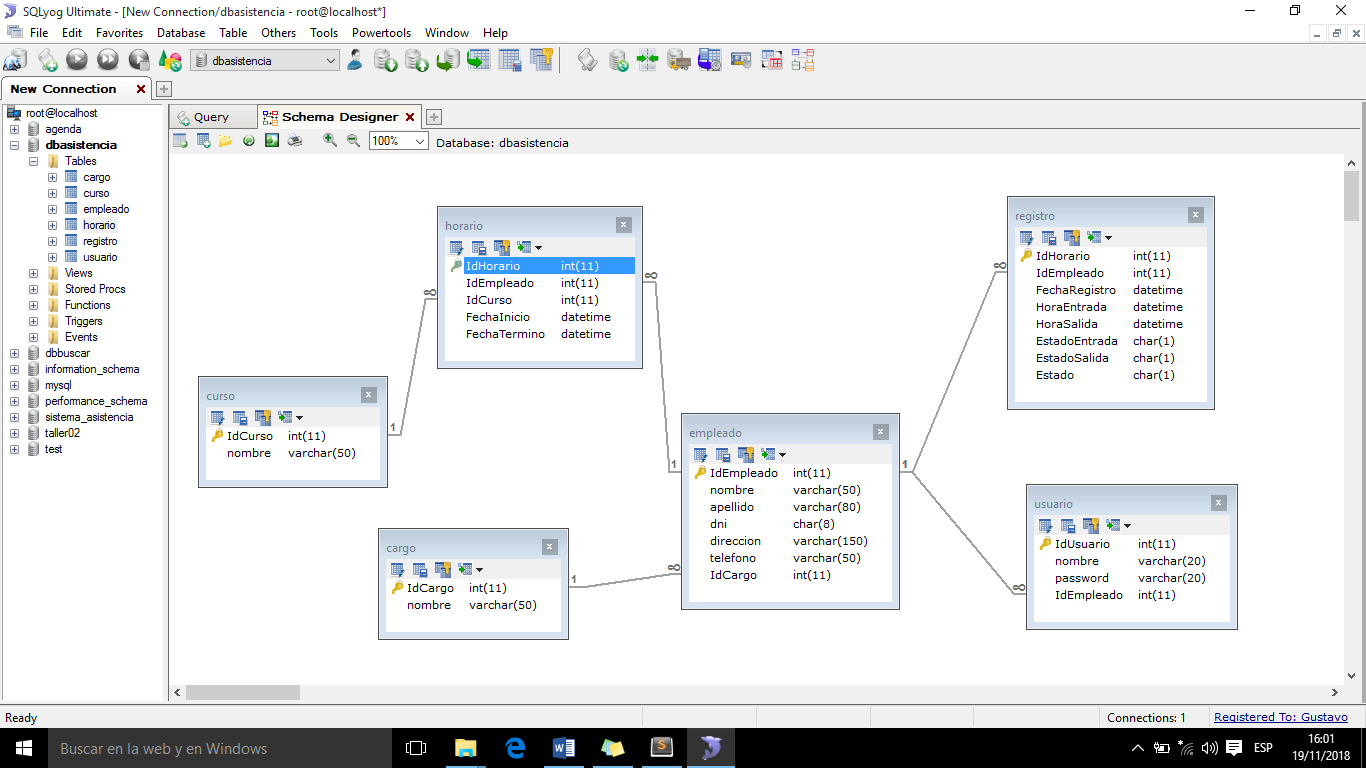
**Diagrama 9:** Diagrama de Clases



Fuente:(Elaboración propia, 2018). Diagrama de Clases

* + 1. CREACIÓN DE LA BASE DE DATOS

**Ilustración 13:** Base de Datos



Fuente: (Elaboración Propia, 2018)

* + 1. CONSTRUIR EL DICCIONARIO DE DATOS

**Tabla 21:** Diccionario De Datos

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TABLA** | **CAMPOS** | **TIPO DE DATOS** | **DESCRIPCION** | **DESCRIPCION DE LAS TABLAS** |
| Cargo | Idcargo | int | Primary key autogenerado | Esta tabla guarda los cargos que está dando la empresa. |
| nombre | varchar(20) | Se ingresa nombre. Ejm: Gerente |
| Usuario | IdUsuario | int | Primary key autogenerado | Esta tabla guarda los datos de los usuarios que forman parte de la empresa. |
| nombre | varchar(20) | Se ingresa nombre. Ejm: Carlos |
| password | varchar(20) | Se ingresa contraseña.Ejem 1234 |
| IdEmpleado | int | Primary key autogenerado |
| Empleado | IdEmpleado | int | Primary key autogenerado | Esta tabla guarda los datos de los empleados que forman parte de la empresa. |
| nombre | varchar(20) | Se ingresa nombre. Ejm: Carlos |
| apellido | varchar(20) | Se ingresa apellido. Ejm: Berrios |
| dni | char(8) | Se ingresa dni. Ejm: 00456231 |
| direccion | varchar(150) | Se ingresa direccion. Ejm: Av. San Martin 456 |
| telefono | varchar(50) | Se ingresa telefono. Ejm: 8956234 |
| Idcargo | int | Primary key autogenerado |
| Registro | IdHorario | int | Primary key autogenerado | Esta tabla guarda los registros de asistencia de los empleados de la empresa. |
| IdEmpleado | int | Primary key autogenerado |
| FechaRegistro | datetime | Se ingresa fecha. Ejm: 2018-11-24 |
| Hora Entrada | datetime | Se ingresa hora de entrada. Ejm:05:00 am |
| Hora Salida | datetime | Se ingresa hora de salida. Ejm:05:00 am |
| EstadoEntrada | char(1) | Se ingresa Estado de entrada. Ejm: A de aceptado |
| EstadoSalida | char(1) | Se ingresa Estado de salida. Ejm: A de aceptado |
| Estado | char(1) | Se ingresa Estado. Ejm: A de aceptado |
| Horario | IdHorario | int | Primary key autogenerado | Esta tabla guarda los horarios de la empresa. |
| IdEmpleado | int | Primary key autogenerado |
| IdCurso | int | Primary key autogenerado |
| FechaInicio | datetime | Se ingresa fecha de inicio.Ejm: 2018-10-05 |
| FechaTermino | datetime | Se ingresa fecha de inicio. Ejm: 2018-04-25 |
| Curso | IdCurso | int | Primary key autogenerado | Esta tabla guarda los cursos que forman parte de la empresa. |
| nombre | varchar(50) | Se ingresa nombre del curso. Ejm: Arquitectura de Entorno Web |
| Cargo | IdCargo | int | Primary key autogenerado | Esta tabla guarda los cargos de los empleados de la empresa. |
| nombre | varchar(50) | Se ingresa nombre del cargo. Ejm: secretaria. |

Fuente:(Elaboración Propia, 2018)

1. CAPITULO VII
2. CONSTRUCCIÓN DEL SISTEMA
   1. FUNCIONES DEL SISTEMA

Se debe identificar todas las funciones que el sistema va a realizar e identificar el tipo de función que cumple de acuerdo a la siguiente tabla.

**Tabla 22:** Funciones del Sistema

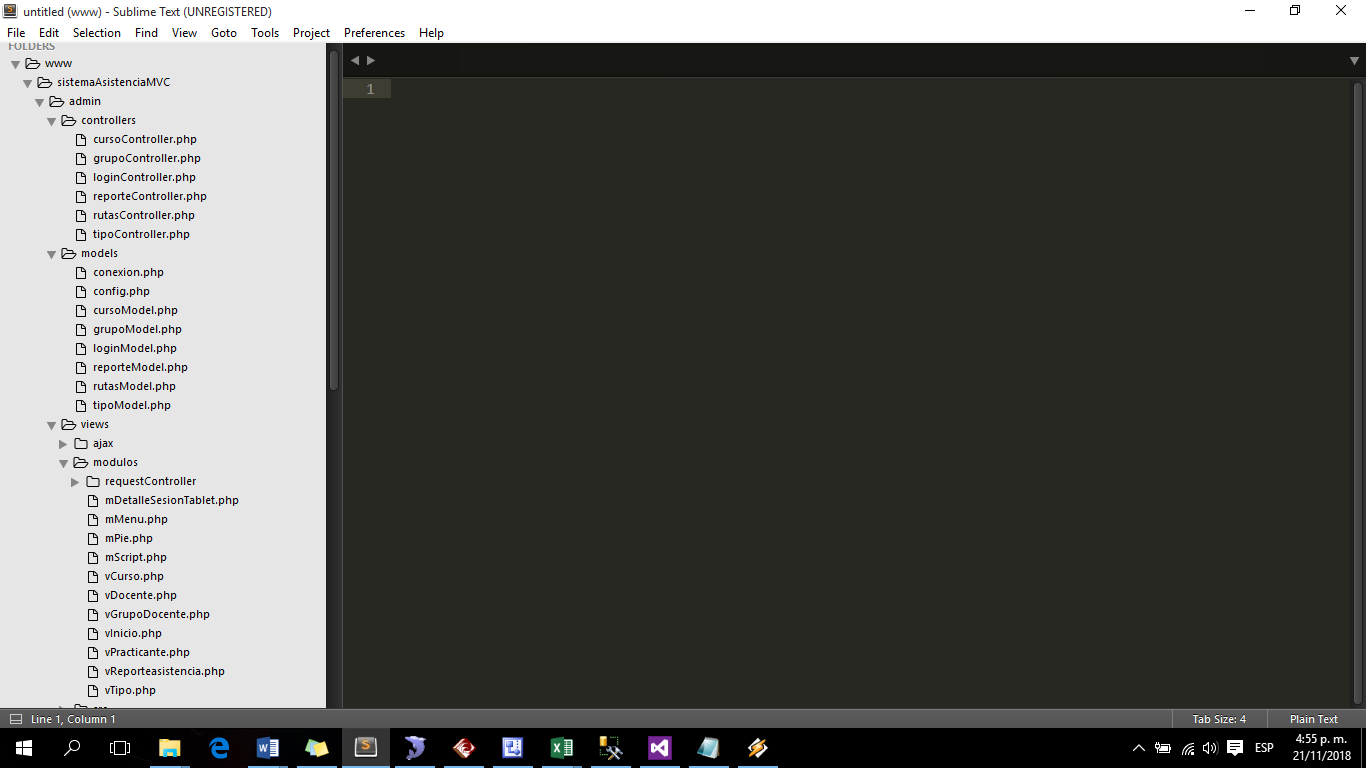
|  |
| --- |
| **Función** |
| Registrar las asistencias (ingreso -salida) de los docentes de los diferentes horarios con los respectivos requerimientos (min. Tarde – días falta) |
| Registrar las asistencias (ingreso -salida) de los practicantes de los diferentes horarios con los respectivos requerimientos (min. Tarde – días falta) |
| Registrar los horarios de los distintos cursos que se apertura.  Registrar nuevos cursos que se apertura en la escuela. |
| Asignar horario (grupo y frecuencia) a los docentes para su futuro registro de asistencia. |
| Generar Diferentes Reportes de registro de asistencia para docentes y personal practicante. |

Fuente: (Elaboración Propia, 2018). Funciones del Sistema.

* 1. PROGRAMACIÓN DEL SISTEMA DE ASISTENCIA

La arquitectura desarrollada en el Sistema de Asistencia, está dada por la arquitectura de 3 Capas Modelo – Vista – Controlador.

**Ilustración 14:** Arquitectura de Modelo – Vista - Controlador



***Fuente:*** *(Elaboración Propia, 2018).*

Modelo-vista-controlador (MVC) es un patrón de arquitectura de software, que separa los datos y la lógica de negocio de una aplicación de su representación y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado, define componentes para la representación de la información, y por otro lado para la interacción del usuario.

De manera genérica, los componentes de MVC se podrían definir como sigue:

**El Modelo:** Es la representación de la información con la cual el sistema opera, por lo tanto, gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio). Envía a la 'vista' aquella parte de la información que en cada momento se le solicita para que sea mostrada (típicamente a un usuario). Las peticiones de acceso o manipulación de información llegan al 'modelo' a través del 'controlador'.12​

**El Controlador:** Responde a eventos (usualmente acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la información (por ejemplo, editar un documento o un registro en una base de datos). También puede enviar comandos a su 'vista' asociada si se solicita un cambio en la forma en que se presenta el 'modelo' (por ejemplo, desplazamiento o scroll por un documento o por los diferentes registros de una base de datos), por tanto, se podría decir que el 'controlador' hace de intermediario entre la 'vista' y el 'modelo'.

**La Vista:** Presenta el 'modelo' (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la interfaz de usuario), por tanto, requiere de dicho 'modelo' la información que debe representar como salida.

* 1. CONEXIÓN A BASE DE DATOS DESDE PHP.

Para la conexión de base de datos utilizamos php orientado a objetos. Esta programación me permitirá desarrollar una aplicación de alta performance.

<?php

class Conexion{

public function cnx() {

try {

$options = [

PDO::ATTR\_ERRMODE =>PDO::ERRMODE\_EXCEPTION,#para menjo de errores

PDO::ATTR\_EMULATE\_PREPARES => false,

];

$link = new PDO("mysql:host=localhost;dbname=sistema\_asistencia","root","",$options);

return $link;

} catch (Exception $e) {

echo "la linea de error es: " . $e->getLine();

}

}

}

$obj = new Conexion();

$sas = $obj->cnx();

var\_dump($sas);

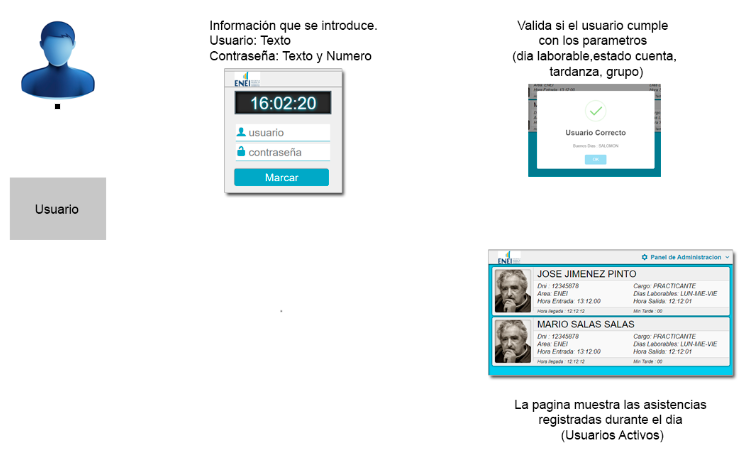
?>

**Funcionamiento Del Sistema de Asistencia.**

El sistema de asistencia el usuario se loguea e ingresa su asistencia en el turno indicado.

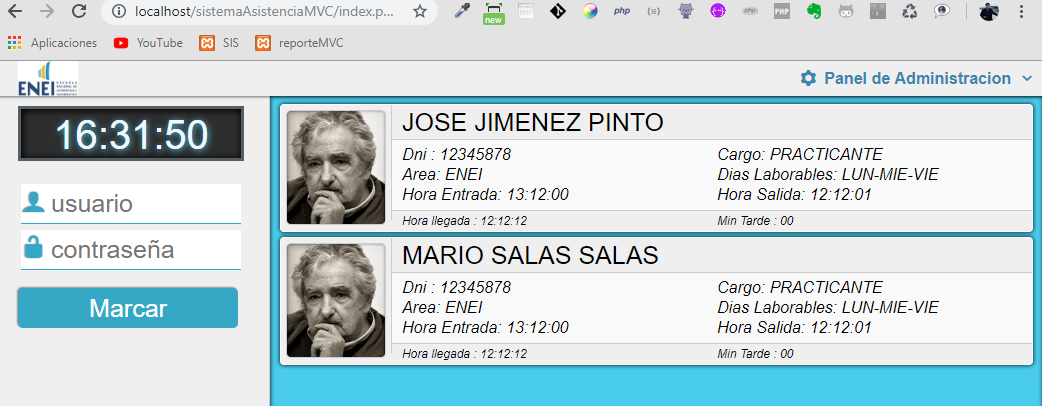
**REGISTRO DE ASISTENCIA**

**Ilustración 15:** Registro Asistencia



**Fuente:** (Elaboracion Propia, 2018).

**Ilustración 16**: Vista Usuarios Activos



**Fuente:** (Elaboración Propia, 2018)

El sistema de asistencia funcionara en un servidor web apache, con un servidor de base de datos MySQL, como se muestra en la figura.



**Código de muestra de la capa de Modelo.**

<?php

require\_once "conexion.php";

class CursoModel extends Conexion{

## INSERTAR CURSO

public function insertarCursoModel($datosModel){

try {

$stmt = Conexion::cnx()->prepare("INSERT INTO curso (nombre\_curso,inicial\_curso,tipo\_curso)

VALUES(:nombreCurso,:inicialCurso,:tipoCurso)");

#bindParam() vinvula parametros con campos de SQL

$stmt->bindParam(":nombreCurso", $datosModel['nombreCurso'], PDO::PARAM\_STR);

$stmt->bindParam(":inicialCurso", $datosModel['inicialCurso'], PDO::PARAM\_STR);

$stmt->bindParam(":tipoCurso", $datosModel['tipoCurso'], PDO::PARAM\_STR);

if ($stmt->execute()) {

return array('resp' => 1);

}else {

return array('resp'=> 0);

}

$stmt->close();

} catch (Exception $e) {

return array("resp"=>0);

// return array("reps"=>0,"<br>","menssajeError"=>$e->getMessage(),"<br>","lineaDeError"=>$e->getLine());

exit();

}

}

## LISTAR CURSO

public function listarCursoModel(){

$stmt = Conexion::cnx()->prepare("SELECT \* FROM curso");

$stmt->execute();

return $stmt->fetchAll(); //fetch = obtiene una fila / fetchAll = ObtieneTodas las filas

$stmt->close();

}

## MOSTRAR POR "ID" CURSO

public function listarIdCursoModel($idCurso){

$stmt = Conexion::cnx()->prepare("SELECT \* FROM curso WHERE id\_curso = :idcurso");

$stmt ->bindParam(":idcurso",$idCurso,PDO::PARAM\_INT);

$stmt -> execute();

return $stmt->fetch();

$stmt->close();

}

## ACTUALIZAR POR "ID" CURSO

public function actualizarCursoModel($datosCurso){

try {

$stmt = Conexion::cnx()->prepare("UPDATE curso SET nombre\_curso = :nombreCurso,

inicial\_curso=:inicialcurso

WHERE id\_curso = :idcurso");

$stmt ->bindParam(":nombreCurso",$datosCurso['nombreCurso'],PDO::PARAM\_STR);

$stmt ->bindParam(":inicialcurso",$datosCurso['inicialCurso'],PDO::PARAM\_STR);

$stmt ->bindParam(":idcurso",$datosCurso['idCurso'],PDO::PARAM\_INT);

if ($stmt -> execute()) {

if ($stmt->rowCount() > 0) {

return array("resp" =>1);

}else{

return array("resp"=>0);

}

}else {

return array("resp"=>0);

}

$stmt->close();

} catch (Exception $e) {

return array("resp"=>'errorExceptionssss');

// return array("reps"=>0,"<br>","menssajeError"=>$e->getMessage(),"<br>","lineaDeError"=>$e->getLine());

exit();

}

}

## ELIMINAR POR "ID" CURSO

public function eliminarCursoModel($idCurso){

try{

$stmt = Conexion::cnx()->prepare("DELETE FROM curso WHERE id\_curso =:idCurso");

$stmt ->bindParam(":idCurso",$idCurso,PDO::PARAM\_INT);

if ($stmt->execute()) {

if ($stmt->rowCount() > 0) {

return array("resp" =>1);

}else{

return array("resp"=>0);

}

}else {

return array("resp"=>0);

}

$stmt->close();

}

catch(PDOException $e){

return array("resp"=>0);

// return array("reps"=>0,"<br>","menssajeError"=>$e->getMessage(),"<br>","lineaDeError"=>$e->getLine());

exit();

}

}

}

?>

* 1. **TIPOS DE USUARIOS**

**Usuario Administrador.-** El usuario administrador tendrá acceso para la creación de usuarios, el manejo de todas las interfaces de la aplicación.

**Usuario Docente. -** El usuario docente tiene los privilegios para poder registrar su asistencia.

* 1. NIVELES DE SEGURIDAD
* *El sistema de Asistencia, se maneja el ingreso del sistema por un usuario y una contraseña, los cuales tienes asociado por diversos permisos para poder acceder al sistema.*
* *El usuario creado tiene un campo para poderlo activar y desactivar para que no ingrese al sistema.*
* *Se administra de SQL Server mediante un usuario y contraseña y da la protección a la base de datos.*
* *La Base de datos tiene campos de auditoria e historial para ver cualquier alteración de la base de datos.*
  1. DISEÑO DE LA ARQUITECTURA DEL SISTEMA

La implementación con que se va trabajar es arquitectura 3 capas, que es una tecnología que proporciona al usuario final el acceso transparente a las aplicaciones, datos y servicios de cómputo o cualquier otro recurso dentro del grupo de trabajo y/o a través de la empresa en múltiples plataformas.

Para la elección de la arquitectura se consideró diferentes Factores, tanto a nivel de visionamiento de la Institución con a nivel de presupuesto y nivel de capacitación de personal para el mantenimiento de nuevas tecnologías.

Se tomó en cuenta la arquitectura de 3 capas a consecuencia de la futura carga que tendrá la aplicación y a la Implementación de futuras aplicaciones los cuales permitirán el uso eficiente de recursos.

Además, al crearse nuevas aplicaciones, nuestra aplicación se adaptará fácilmente ya que solo se tendrá que cambiar algunos parámetros en la capa de negocios

* 1. **Descripción de las 3 Capas**

Descripción de la Capa de Presentación.

En la capa de presentación estarán los diferentes formularios web y los reportes los cuales estarán alojados en un servidor web. A continuación, se detalla algunos formularios que contendrá el servidor para su publicación.

**Tabla 23:** Nombre de Formularios

|  |
| --- |
| **Nombre de Formularios – Sistema de Asistencia** |
| vDocente.php |
| vCurso.php |
| vPracticante |

***Fuente:*** *(Elaboración Propia, 2018)*

*Descripción de la Capa de Negocios (Controllers).*

**Tabla 24:** Descripción de la Capa de Negocios (Controllers).

|  |  |
| --- | --- |
| **Nombre** | **Descripción** |
| Controller | * cursoController.php * loginController.php * reporteController.php |

*Descripción de la Capa de Acceso a Datos (ABD).*

**Tabla 25:** Descripción de la Capa de Acceso a Datos (ABD).

|  |  |
| --- | --- |
| **Nombre** | **Descripción** |
| Modelo | * cursoModel.php * rutasModel.php * reporteModel.php * grupoModel.php |

*Descripción de la Capa de Datos*

**Tabla 26:** Descripción de la Capa de Datos

|  |  |
| --- | --- |
| **Nombre** | **Descripción** |
| bdasistencia | En la base de datos del sistema de asistencia. Que contiene las tablas referidas a la asistencia. |
| Lógica de Datos | Contiene todos los procedimientos almacenados los que hacen las transacciones con el SQL server. |

**Ilustración 17: Arquitectura Sistema**



**Diagrama de Despliegue.**

En el diagrama de despliegue se indica la situación física de los componentes lógicos desarrollados. Es decir se sitúa el software en el hardware que lo contiene. Cada Hardware se representa como un nodo.

**Diagrama 10:** Diagrama de Despliege



Fuente:(Elaboración Propia, 2018

CAPITULO VIII

PRUEBA DEL SISTEMA

* 1. PROGRAMA DE CAPACITACIÓN AL USUARIO

*Detallar la forma como se va implantar dicha capacitación y cuáles son los objetivos específicos que persigue dicha capacitación.*

* En este proyecto se planificarán las diversas capacitaciones necesarias para el personal que lo necesite.
* Se realizara: **La Instalación del Sistema.**
* Se desarrollara un plan de capacitación personal y se expondrá de las bondades de este Sistema.
* Se monitoreara remotamente para proporcionar soporte técnico.
  1. EVALUACIÓN DEL SISTEMA
     1. EVALUACIÓN OPERACIONAL

El sistema de asistencia, operacionalmente tiene el siguiente:

* Sistema con una arquitectura con alta escalabilidad y performance
* El sistema en sus interface es amigable en su fácil manejo
* EL sistema tiene ayudas en el manejo de ingreso de datos.
* El sistema a nivel de envió de transacciones es veloz de acuerdo a la velocidad de la red de datos.
  + 1. IMPACTO ORGANIZACIONAL

El sistema de asistencia de docentes y practicantes, tiene un impacto muy alto en la ODEI TACNA, por que ayudara a llevar un mejor control de las asistencias. Y se podrá contar con un sistema de que controle estas actividades.

* + 1. DESEMPEÑO DEL DESARROLLO

El desarrollo ha sido de una manera rápida con todo los estándares establecidos en la programación. Asimismo la base de datos y su lógica de datos tiene todos los estándares necesario para su fácil mantenimiento y modificación lo que conlleva a una mejora del sistema.

* 1. PLAN DE PRUEBAS DEL SISTEMA

En este punto se realizara, diferentes pruebas al vacío, desde el inicio del proceso hasta el final del proceso.

A la vez se comprobara con un testing la funcionalidad del sistema.

* 1. PLAN DE MANTENIMIENTO DEL SISTEMA

Cuando se haya instalado el sistema, la empresa mediante su área de soporte cuenta con el especialista adecuado para darle el soporte y ampliación del sistema. El cual el técnico de la empresa será el responsable de darle el soporte adecuado.

1. CAPITULO IX
2. REQUERIMIENTOS TÉCNICOS
   1. HARDWARE

Se necesita 03 equipos de cómputo con las siguientes características

**Tabla 27**: Elementos Técnicos (Hardware)

|  |  |
| --- | --- |
| Familia de procesador | Intel Core i5 |
| Modelo del procesador | i3-4150 |
| Frecuencia del procesador | 3,5 GHz |
| Memoria interna | 4 GB |
| Tipo de memoria interna | DDR3-SDRAM |
| Velocidad de memoria del reloj | 1600 MHz |
| Capacidad total de almacenaje | 500 GB |

**Fuente:** (Elaboración Propia, 2018)

* 1. SOFTWARE
* Las estaciones de trabajo deberán sistema operativo Windows y debe estar instalado framework 4.5.
* El lenguaje de programación es php y MySQL.
  1. ARQUITECTURA DEL SISTEMA

La Arquitectura que está desarrollado el sistema es a un nivel de 03 capas. Con una capa de Acceso de Base de Datos y una capa de datos en MySQL. Como se muestra en la figura.

**Tabla 28:** Arquitectura del Sistema



Fuente: (Elaboración Propia, 2018)

CONCLUSIONES

* Podemos concluir que el uso de la programación en MVC, proporciona a los sistemas una escalabilidad y performance adecuada. Brindando la reutilización de código para ser consumidas por otros programas que necesiten la funcionabilidad del sistema primario.
* Podemos concluir que los requerimientos bien formulados se basan en aplicar un conjunto de técnicas que nos permitan recopilar información verídica para minimizar futuros cambios en el sistema.
* Aplicando el conjunto de técnicas del scrum se logró el diseño del Sistema de asistencias de docentes y practicantes que refleja las necesidades de los clientes y que este diseño puede ser modificado con flexibilidad para futuros requerimientos.
* La aplicación de una nueva arquitectura en el desarrollo de software, establece que la comunicación de otros sistemas sea más factible. Como la migración de tecnología de software.

RRECOMENDACIONES

* Se le sugiere a la empresa establecer una arquitectura adecuada y bien definida que posea escalabilidad y performance para implantar sistemas que ofrezcan a los usuarios un mejor desenvolvimiento de sus funciones.
* Se recomienda que la forma de programación sea estandarizada mediante la programación orientada a objeto. Para futuras migraciones. Como es la migración de base de datos la cual teniendo la forma de codificación adecuada se podrá migrar fácilmente, esto conlleva a que la última tecnología en software libre. Haga más rápido la fluidez de la información.

BIBLIOGRAFIA

Mohammad Younes. (2018). *Alertefy JS*. Obtenido de https://alertifyjs.com/

Academia Web. (12 de 12 de 2017). *¿Qué es el diseño responsive?* Obtenido de https://www.40defiebre.com/que-es/diseno-responsive

Álvarez, C. (26 de 02 de 2015). *Ganbeta.dev*. Recuperado el 27 de 11 de 2018, de Edición de archivos JSON online: https://www.genbeta.com/desarrollo/edicion-de-archivos-json-online

Cespede, J. (2017). *SISTEMA DE INFORMACIÓN WEB PARA MEJORAR LA GESTIÓN.* Trujillo - Perú: Universidad Nacional de Trujillo. Obtenido de file:///C:/Users/SaloValdez/Downloads/Montoya%20Rodr%C3%ADguez,%20Roxana%20Katherin%3B%20Sanchez%20Diaz,%20Miguel%20Lizardo%20(1).pdf

CHACÓN, J. C. (2006). *APLICACIÓN DE LA METODOLOGÍA RUP PARA EL.* Tesis, Guatemala.

Colaboradores Mozilla, M. (2005-2018). *Introducción a CSS*. (M. w. docs, Editor) Obtenido de https://developer.mozilla.org/es/docs/Learn/CSS/Introduction\_to\_CSS

Concepcion, R. (02 de 02 de 2018). *PROGRAMACIÓN III*. Obtenido de Funcionamiento de Ajax: http://riveraconceptosprogra3.blogspot.com/p/funcionamiento-de-ajax.html

Freddy Vega. (05 de 12 de 2012). *Platzi*. Obtenido de Platzi Web: https://platzi.com/blog/ajax-con-jquery/

Fridd Lensco. (2018). *caniuse*. Obtenido de caniuse: https://caniuse.com/#search=GRID

Ilya Kantor. Ilya Kantor. (12 de 09 de 2018). *Java Scrip Info*. Obtenido de Una introducción a JavaScript: https://javascript.info/intro

jQuery, L. F. (2018). *Jquery Documentation*, 3.3.1. Obtenido de ¿Qué es jQuery?: https://jquery.com/

Juan Padial. (5 de 7 de 2016). *CYBMETA*. Obtenido de Ajax con jQuery, JSON y PHP: ejemplo paso a paso: https://cybmeta.com/ajax-con-json-y-php-ejemplo-paso-a-paso

Junta de Anadalucia, I. A. (12 de 05 de 2016). *Uso de Patrones de diseño*. Obtenido de http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/122

La revista Tecnologica, R. (2015). *HISTORIA Y OBJETIVOS DEL LENGUAJE HTML*. Obtenido de http://www.larevistainformatica.com/historia-objetivos-del-html.htm

Pergamino Virtual, T. I. (2002). *Implementacion de sistema web de Matriculas PHP.* San Andres: Universidad San Andres de Bolivia.

PHP Group, P. N. (22 de 11 de 2018). *PHP*. Obtenido de http://php.net/

Process, R. U. (25 de 05 de 2002). *Rational Unified Process: Overview*. Obtenido de Rational Unified Process: Overview: http://sce.uhcl.edu/helm/rationalunifiedprocess/

Ricardo Prieto. (11 de 12 de 2017). *Media Queries en CSS ¿Cómo funcionan?* Obtenido de https://www.silocreativo.com/media-queries-css/

SpryMedia Ltd. (02 de 02 de 2017). *Tablas de Datos*. Recuperado el 27 de 11 de 2018, de Data Tables: https://datatables.net/

Victor Robles, V. R. (2 de 12 de 2015). *MVC (Modelo Vista Controlador) en PHP nativo*. Obtenido de https://victorroblesweb.es/2013/11/18/tutorial-mvc-en-php-nativo/

W3Schools. (2017). *CSS Grid Layout Module*. Obtenido de https://www.w3schools.com/css/css\_grid.asp

Wikipedia. (24 de 12 de 2018). *Wikipedia - PHP*. Obtenido de https://es.wikipedia.org/wiki/PHP

Wikipedia Apache, W. (24 de 10 de 2018). *Servidor HTTP Apache*. Obtenido de https://es.wikipedia.org/wiki/Servidor\_HTTP\_Apache

Wikipedia MVC, W. (1 de 5 de 2010). *Modelo Vista Controlador*. Obtenido de https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93controlador#/media/File:MVC-Process.png

Wikipedia, L. e. (27 de 11 de 2018). *Modelo–vista–controlador*. Obtenido de Modelo–vista–controlador: https://es.wikipedia.org/w/index.php?title=Programaci%C3%B3n\_orientada\_a\_objetos&oldid=112295798

Wkipedia Mysql, W. (8 de 22 de 2017). *MysQl*. Obtenido de https://es.wikipedia.org/wiki/MySQL

ANEXOS

* Glosario de la Terminología.

**B**

**Base de datos:** conjunto de datos organizados que se puede editar y gestionar para buscar  información o trabajar con ella.

**C**

**Call:** llamada. Instrucción que se utiliza para desviar la secuencia de ejecución de un programa hacia un subprograma.

**Carácter:** número, letra o símbolo en la computadora, formado por un byte.

**Command:** comando o instrucción que un usuario imparte al sistema operativo para que realice una acción determinada.

**Compilador:** intérprete. Programa traductor que convierte las instrucciones de un lenguaje avanzado en secuencias de instrucciones binarias llamadas código objeto para poder ejecutarlas.

**Computadora:** dispositivo electrónico para realizar operaciones aritméticas y lógicas de alta velocidad.

**Consta de cinco componentes básicos:** la unidad aritmética lógica (ALU), la unidad de control, dispositivos de entrada y salidad de datos y memoria.

**Computer system:** sistema informático.

**Concentrador:** procesador programado para combinar datos transmitidos por varios dispositivos de baja velocidad y luego transmitirlos a una velocidad mucho mayor, generalmente a una computadora remota.

**Controlador:** unidad que controla la ejecución de las instrucciones de la computadora y su secuencia de operaciones.

**D**

**Data:** dato. Información que se facilita a la computadora

**Database**: base de datos.  Organización sistemática de archivos de datos para facilitar el acceso, búsqueda y actualización.

**Debugger:** depuración, corrección de errores o bugs.

**Decoder**: decodificador o descifrador.

**Default:** por defecto. Por omisión.

**Delete:** borrar; eliminar; anular.

**Diseño de Entrada**.- se considera esencialmente a los ingresos de datos según la forma de que el usuario lo requiera y según los métodos pueden ser por el teclado o por el Mouse.

**Diseño de Salida**.- Es realizado esencialmente por el analista, básicamente se refiere a los resultados que obtiene el usuario como por ejemplo reportes.

**Diseño del Interfaz**.- Esta etapa tiene como objetivo el diseño de los medios por el cual el sistema se conecta con el usuario cuando este al introducir información posteriormente obtiene resultados.

Una forma de interfaz puede ser la barra de menús como el inicio de la interfaz donde el usuario podrá acceder a diversas opciones según sus necesidades para esto puede hacerlo seleccionando con el Mouse o con una combinación de teclas.

**E**

**Error.**- Concepto equivocado o juicio falso.

**F**

**Formularios:** Permiten al usuario enviar la información necesaria para realizar consultas en bases de datos, comprar o solicitar un servicio. Otras marcas permiten mejorar la presentación de los documentos, por ejemplo, añadiendo fondos, tablas de contenido o textos intermitentes.

**I**

**Interfaz: E**s el medio que permite la interacción entre esos elementos. En el campo de la informática se distinguen diversos tipos de interfaces que actúan a diversos niveles, desde las interfaces claramente visibles, que permiten a las personas comunicarse con los programas, hasta las imprescindibles interfaces hardware, a menudo invisibles, que conectan entre sí los dispositivos y componentes dentro de los ordenadores o computadoras. Las interfaces de usuario cuentan con el diseño gráfico, los comandos, mensajes y otros elementos que permiten a un usuario comunicarse con un programa.

**Informática:** Conjunto de conocimientos científicos y de técnicas que hacen posible el tratamiento automático de la información por medio de computadoras.

**Implementar.-** Poner en funcionamiento, aplicar métodos, medidas, etc., para llevar algo a cabo.

**Interfaz gráfica de usuario**: En informática, tipo de entorno que permite al usuario elegir comandos, iniciar programas, ver listas de archivos y otras opciones utilizando las representaciones visuales (iconos) y las listas de elementos del menú. Las selecciones pueden activarse bien a través del teclado o con el ratón. (Véase Interfaz de usuario).

**L**

**Label:** etiqueta.

**P**

**Password:** contraseña.

**Proyecto técnico:** Documentación que se elabora para la ejecución de una iniciativa de envergadura, con previsión de todos los detalles necesarios para su logro. Resultan necesarios en diversos campos profesionales, como en la ejecución de obras públicas o edificaciones, el diseño y la organización industrial, o la planificación económica, ambiental y territorial (ordenación del territorio).

**T**

**Toolbar:** barra de herramientas.

**Tools:** herramientas.

**U**

**USB (Universal Serial Bus):** es una interfase de tipo plug & play entre una Computadora y ciertos dispositivos, por ejemplo, teclados, teléfonos, escáners e impresoras.

**Usuario:** Persona experta en computadoras, particularmente en la gestión de aplicaciones, más que en programación o en el mantenimiento de hardware.

**Un sistema informático**: Suele estar compuesto por una unidad central de proceso (CPU), dispositivos de entrada, dispositivos de almacenamiento y dispositivos de salida. La CPU incluye una unidad aritmético-lógica (ALU), registros, sección de control y bus lógico. Los registros almacenan los datos y los resultados de las operaciones. La unidad de control regula y controla diversas operaciones. En la mayoría de las computadoras, el principal dispositivo de entrada es el teclado. Dispositivos de almacenamiento son los discos duros, flexibles (disquetes) y compactos (CD). Dispositivos de salida que permiten ver los datos son los monitores e impresoras.

**V**

**Variable:** entidad simbólica que se utiliza en un programa.

**Virus:** pequeño programa que "infecta" una computadora; puede causar efectos indeseables y hasta daños irreparables.

**W**

**Windows:** Sistema operativo desarrollado por Microsoft que tiene la particularidad de trabajar por ventanas.