

DISEÑO Y DESARROLLO DE UN SISTEMA DE INFORMACION PARA EL
SEGUIMIENTO A LOS EGRESADOS EN EL AREA DE INGENIERIA DE LA
FUNDACIÓN UNIVERSITARIA LOS LIBERTADORES

OMAR DANIEL ARIAS

RAUL ELIECER LOPEZ

WILLIAM ALEJANDRO FIESCO

JAVIER DAZA PIRAGAUTA

FUNDACION UNIVERSITARIA LOS LIBERTADORES

FACULTAD DE INGENIERIA

PROGRAMA DE INGENIERIA DE SISTEMAS

BOGOTA D.C.

2017

DEDICATORIA DE OMAR DANIEL ARIAS

A Dios.

Por darme la oportunidad de vivir y por estar conmigo en cada paso que doy, por fortalecer mi corazón e iluminar mi mente y por haber puesto en mi camino a aquellas personas que han sido mi soporte y compañía durante todo el periodo de estudio.

A Mi Madre Marisol Olaya.

Por haberme apoyado en todo momento, por sus consejos, sus valores, por la motivación constante que me ha permitido ser una persona de bien, pero más que nada, por su amor.

A Mi Padre Nelson Arias.

Por los ejemplos de perseverancia y constancia que lo caracterizan y que me ha infundado siempre, por el valor mostrado para salir adelante y por su amor.

A Mis Hermanos.

Nelson Mauricio Arias y Felipe Andrés Arias, por estar conmigo y apoyarme siempre, los quiero mucho.

A Mis Sobrinos.

Juan Jose Arias y Luciana Arias Guali, para que veas en mí un ejemplo a seguir.

A Todos Mis Amigos.

Por compartir los buenos y malos momentos.

DEDICATORIA DE RAUL ELIECER LOPEZ

A Dios.

Por haberme dado salud y las fuerzas necesarias para seguir adelante y permitirme

llegar a este punto, superando las adversidades en el camino.

A mis padres que con su amor y su esfuerzo siempre fueron un apoyo constante en los momentos difíciles, y que han sido siempre un ejemplo e inspiración para continuar con mis objetivos.

DEDICATORIA DE WILLIAM ALEJANDRO FIESCO

A Dios

Por brindarme la oportunidad de forjarme profesionalmente.

A Mi Madre Maria Quecan

*Por brindarme un apoyo incondicional en mi vida y en mi formación tanto espiritual
como educativa.*

A Mi Familia

Por acompañarme en este camino arduo.

A Todos Mis Amigos

Que de una y otra manera dejaron una enseñanza en mi vida.

TITULO

DISEÑO Y DESARROLLO DE UN SISTEMA DE INFORMACION PARA EL SEGUIMIENTO A LOS EGRESADOS EN EL AREA DE INGENIERIA DE LA FUNDACIÓN UNIVERSITARIA LOS LIBERTADORES

Palabras clave: MySQL, JAVA, Egresados, Universidad los Libertadores.

DESCRIPCION

El proyecto describe la necesidad que tiene la Fundación Universitaria Los Libertadores de reconocer la importancia que tiene sus egresados a nivel socio-económico, en el ámbito nacional, es necesario crear un vínculo permanente con ellos de forma tal que facilite la interacción de los Egresados a través de un sistema de información, apoyados en el plan de mejoramiento de la interacción y seguimiento permanente con los Egresados. Así mismo mantienen a la Fundación Universitaria Los Libertadores en contacto con los Egresados, y crear en ellos el sentido de pertenencia que los vuelva solidarios con la institución que los formo. En base a lo anterior nuestro sistema de información implementa una herramienta para automatizar la recolección, recuperación y administración de datos de los Egresados tales como datos personales, actualizaciones académicas y desempeño profesional.

ABSTRACT

TITLE

DESIGN AND DEVELOPMENT OF AN INFORMATION SYSTEM FOR THE FOLLOW-UP TO THE EMERGENCIES IN THE ENGINEERING AREA OF THE LIBERTADORES UNIVERSITY FOUNDATION

Keywords: MySQL, JAVA, Alumni, Universidad los Libertadores.

DESCRIPTION

The project describes the need for the University Foundation Los Libertadores to recognize the importance that its graduates have at socio-economic level, at the national level, it is necessary to create a permanent link with them in a way that facilitates the interaction of graduates to Through an information system, supported in the plan of improvement of the interaction and permanent monitoring with the Alumni. They also maintain the University Foundation Los Libertadores in contact with the graduates, and create in them a sense of belonging that will return them in solidarity with the institution that formed them. Based on the above, our information system implements a tool to automate the collection, retrieval and management of data of graduates such as personal data, academic updates and professional performance.

INTRODUCCION

Desde sus inicios el hombre el hombre ha venido evolucionando. En tiempos milenarios inicio con la rueda, quien es base del mundo moderno y será del futuro. Del mismo modo fue evolucionando dentro del campo informático, siendo el ábaco el motor de arranque para la calculadora y el sistema binario. Todo esto sin dejar a un lado el concepto de que las guerras, así como han dejado desolación y muerte, han sido artífices de los más grandes inventos del hombre. Pues el desarrollo armamentista ha sido la punta de lanza para la evolución del pensamiento humano en todos sus campos, más en el tecnológico; ejemplo claro de este proceso ha sido el origen de lo que hoy conocemos como internet. Inicialmente llamado ARPANET y que tan solo conectaba 4 ordenadores, los cuales al cabo de dos años eran ya 40 y fue creciendo a un ritmo exponencial, hasta el punto que hoy cada individuo posee la capacidad tecnológica de poder conectarse no solo desde su ordenador, si no desde su teléfono celular.

De esta misma forma las redes sociales incursionaron en nuestras vidas en el mismo orden que creció el internet. Inicio con el envío del primer e-mail en el año 1971 entre dos ordenadores uno al lado del otro, teniendo su mayor auge en los años 90's. Donde ya cada individuo puede crear una página web e interactuar con sus similares en cualquier parte del mundo. Ya desde el año 1995 se crean lugares virtuales como Classmates, una red social para contactar con antiguos compañeros de estudio, antecesor de Facebook quien nació hacia el año 2003 como plataforma para conectar a los estudiantes de la Universidad de Harvard. A partir de este momento nacen muchas otras redes sociales como Hi5 y Netlog, entre otras.

Observando la necesidad de interconectarse con el resto del mundo y la evolución de los sistemas virtuales, se ha podido observar que estas redes sociales no solo sirven para estar en contacto con las personas que conocemos y allegados como amigos, familiares, parientes, entre otros. Son útiles para conocer ciudades, personas, culturas y en muchas ocasiones buscar empleo u ofrecer empleo, hasta el punto de que hoy en día existen empresas on line reconocidas en todo el mundo ofreciendo bienes y servicios.

Desde este punto de vista urge la necesidad de organizar e interconectar a los egresados en ingeniería de sistemas de la universidad los libertadores, con empresas reconocidas a nivel distrital, regional, nacional e internacional; para que de esta manera se mejore el campo laboral y la universidad pueda, no solo, tener un mayor campo de acción profesional si no poder hacer un seguimiento más detallado de sus ex alumnos, permitiendo conocer las necesidades del mercado laboral y realizar los ajustes curriculares a que haya lugar si es necesario, mejorando su buen nombre; graduando ingenieros de sistemas más competitivos en el campo profesional y que cumplan con los retos impuestos por el futuro, en el presente, empleando las herramientas necesarias que le permitan ser más eficaz y eficiente.

TABLA DE CONTENIDO

| | |
|--|----|
| CAPÍTULO I | |
| PLANTEAMIENTO DEL PROBLEMA | 13 |
| ANTECEDENTES | 13 |
| DEFINICION DEL PROYECTO..... | 14 |
| CAPÍTULO II | |
| OBJETIVOS | 15 |
| OBJETIVO GENERAL | 15 |
| OBJETIVOS ESPECIFICOS | 15 |
| CAPÍTULO III | |
| INFORMACION GENERAL | 17 |
| JUSTIFICACION..... | 17 |
| ALCANCE Y LIMITACIONES..... | 17 |
| CAPÍTULO IV | |
| MARCO TEORICO | 18 |
| BASE DE DATOS | 18 |
| APLICACIONES DE LOS SISTEMAS DE BASE DE DATOS..... | 19 |
| CARACTERISTICAS DE LAS BASES DE DATOS | 20 |
| MODELOS DE DATOS | 22 |
| LENGUAJES DE BASES DE DATOS | 24 |
| LENGUAJE DE DEFINICIÓN DE DATOS | 24 |
| LENGUAJE DE MANIPULACIÓN DE DATOS | 25 |
| MYSQL..... | 25 |
| CARACTERÍSTICAS DE MYSQL..... | 26 |
| VENTAJAS DE MYSQL..... | 26 |
| PHP..... | 27 |
| VENTAJAS DE PHP..... | 28 |
| DESVENTAJAS DE PHP | 29 |
| CAPÍTULO V | |
| MARCO METODOLOGICO | 30 |
| PROTOTIPADO EVOLUTIVO | 30 |
| PROCEDIMIENTO A SEGUIR PARA LA METODOLOGÍA ESCOGIDA..... | 31 |
| FASE DE DEFINICION | 32 |
| FASE DE ANALISIS | 32 |
| FASE DE DESARROLLO E IMPLEMENTACION..... | 33 |
| VENTAJAS..... | 34 |
| DESVENTAJAS | 34 |
| CAPÍTULO VI | |
| DIAGRAMAS..... | 35 |
| LENGUAJE UNIFICADO DE MODELADO..... | 35 |
| DEFINICION DE CASO DE USO | 35 |
| DIAGRAMA DE CASO DE USO | 35 |
| ACTORES..... | 36 |
| CASO DE USO | 37 |
| RELACIONES | 37 |

| | |
|--|----|
| DIAGRAMA DE LOS CASOS DE USOS DEL SISTEMA DE INFORMACION PARA EL SEGUIMIENTO A LOS EGRESADOS EN EL AREA DE INGENIERIA DE LA FUNDACIÓN UNIVERSITARIA LOS LIBERTADORES | 39 |
| DIAGRAMA DE CASO DE USO DE LOS EGRESADOS | 39 |
| DIAGRAMA DE CASO DE USO DE LAS EMPRESAS | 39 |
| DESCRIPCION DEL CASO DE USO DE INICIO DE SESION DE LOS EGRESADOS | 40 |
| DESCRIPCION DEL CASO DE USO DE ACTUALIZACION DEL CURRICULO DE LOS EGRESADOS | 41 |
| DESCRIPCION DEL CASO DE USO DE CONSULTAR OFERTAs DE LAS EMPRESAS ... | 42 |
| DESCRIPCION DEL CASO DE USO DE GENERAR REPORTES DESDE LA ADMINISTRACION..... | 43 |
| DESCRIPCION DEL CASO DE USO PARA ADMINISTRAR LAS PUBLICACIONES DESDE LA ADMINISTRACION..... | 44 |
| DESCRIPCION DEL CASO DE USO PARA APROBAR LAS EMPRESAS REGISTRADAS DESDE LA ADMINISTRACION..... | 45 |
| CAPITULO VII | |
| EXPLICACION DE LA HERRAMIENTA | 47 |
| DESARROLLO DE LA HERRAMIENTA | 47 |
| ANALISIS DE REQUISITOS | 47 |
| REQUISITOS ADMINISTRADOR..... | 47 |
| REQUISITOS DE USUARIOS | 49 |
| REQUISITOS EMPRESAS | 50 |
| CAPITULO VIII | |
| EXPLICACION DEL CODIGO | 51 |
| EXPLICACION DE LOS LENGUAJES | 51 |
| HTML | 51 |
| PHP..... | 51 |
| JAVA SCRIPT..... | 52 |
| EXPLICACION DEL CODIGO INDEX.HTML..... | 52 |
| LA CABECERA O ENCABEZADO..... | 53 |
| EL CUERPO..... | 57 |
| EXPLICACION DEL CODIGO EN LOS CSS | 61 |
| HISTORIA DE LOS CSS..... | 63 |
| EXPLICACION DEL CODIGO EN LOS JAVASCRIPT | 71 |
| ARQUITECTURA CLIENTE/SERVIDOR..... | 73 |
| CARACTERISTICAS DE LA ARQUITECTURA CLIENTE/SERVIDOR | 75 |
| VENTAJAS DEL ESQUEMA CLIENTE/SERVIDOR..... | 76 |
| DESVENTAJAS DEL ESQUEMA CLIENTE/SERVIDOR | 77 |
| TIPOS DE ARQUITECTURA CLIENTE/SERVIDOR | 78 |
| ARQUITECTURA CLIENTE/SERVIDOR DOS CAPAS | 78 |
| ARQUITECTURA CLIENTE/SERVIDOR DE TRES CAPAS..... | 79 |
| CAPITULO IX | |
| BIBLIOGRAFIA..... | 81 |

TABLA DE CONTENIDO DE LAS ILUSTRACIONES

| | |
|---|----|
| FIGURA 1. NIVELES DE ABSTRACCIÓN DE DATOS ----- | 22 |
| FIGURA 2. CICLO METODOLOGÍA PROTOTIPADO EVOLUTIVO ----- | 30 |
| FIGURA 3. ALGORITMO METODOLOGÍA DE DESARROLLO PROTOTIPADO ----- | 31 |
| FIGURA 4. ACTORES----- | 36 |
| FIGURA 5. CASO DE USO ----- | 37 |
| FIGURA 6. DIAGRAMA DE CASOS DE USO ADMINISTRADOR DEL SISTEMA----- | 38 |
| FIGURA 7. DIAGRAMA DE CASO DE USO DE LOS EGRESADOS----- | 39 |
| FIGURA 8. DIAGRAMA DE CASO DE USO DE LAS EMPRESAS ----- | 39 |
| FIGURA 9. EL ENCABEZADO DE LA PÁGINA WEB DE LOS EGRESADOS----- | 56 |
| FIGURA 10. EL CUERPO DE LA PÁGINA WEB DE LOS EGRESADOS----- | 60 |
| FIGURA 11. EL INDEX.HTML DE LA PÁGINA WEB DE LOS EGRESADOS ----- | 61 |
| FIGURA 12. EL CSS DE LA PÁGINA WEB DE LOS EGRESADOS ----- | 67 |
| FIGURA 13. EL CSS DE LA PÁGINA WEB DE LOS EGRESADOS ----- | 69 |
| FIGURA 14. EL CSS DE LA PÁGINA WEB DE LOS EGRESADOS ----- | 70 |
| FIGURA 15. MODELAMIENTO INTERNO DE LA PÁGINA WEB ----- | 72 |
| FIGURA 16. ARQUITECTURA CLIENTE/SERVIDOR----- | 74 |
| FIGURA 17. ARQUITECTURA CLIENTE/SERVIDOR DOS CAPAS----- | 79 |
| FIGURA 18. ARQUITECTURA CLIENTE/SERVIDOR TRES CAPAS ----- | 80 |

TABLA DE CONTENIDO DE LAS TABLAS

| | |
|---|----|
| TABLA 1. DESCRIPCIÓN DEL CASO DE USO DE INICIO DE SESIÓN DE LOS EGRESADOS ----- | 40 |
| TABLA 2. DESCRIPCIÓN DEL CASO DE USO DE ACTUALIZACIÓN DE CURRÍCULO DE LOS EGRESADOS ----- | 41 |
| TABLA 3. DESCRIPCIÓN DEL CASO DE USO DE CONSULTAR OFERTAS DE LAS EMPRESAS ----- | 42 |
| TABLA 4. DESCRIPCIÓN DEL CASO DE USO DE GENERAR REPORTES DESDE LA ADMINISTRACIÓN ----- | 43 |
| TABLA 5. DESCRIPCIÓN DEL CASO DE USO PARA ADMINISTRAR LAS PUBLICACIONES DESDE LA ADMINISTRACIÓN ----- | 45 |
| TABLA 6. DESCRIPCIÓN DEL CASO DE USO PARA APROBAR LAS EMPRESAS REGISTRADAS DESDE LA ADMINISTRACIÓN ----- | 46 |
| TABLA 7. VARIABLES QUE SE UTILIZA EN LA CABECERA O ENCABEZADO ----- | 54 |
| TABLA 8. VARIABLES QUE SE UTILIZA DENTRO DEL TYPE----- | 55 |

CAPÍTULO I

PLANTEAMIENTO DEL PROBLEMA

ANTECEDENTES

En la educación superior en Colombia es reglamentada por la ley 30 de 1992 y precisiones adicionales en la ley 115 de 1994, la cual establece que: “la educación superior debe ser un servicio público cultural, inherente a la finalidad social del estado”. Es impartida en instituciones de educación superior clasificadas por la ley en Instituciones Técnicas profesionales, Instituciones universitarias o Escuelas Tecnológicas y Universidades.

En cumplimiento a las disposiciones de ley surge la Fundación Universitaria los Libertadores en el año de 1982, la cual cuenta con un reconocimiento a nivel nacional por su excelencia académica y egresar profesionales competitivos altamente calificados. No obstante, la apertura económica, los Tratados de Libre Comercio (TLC), el cambio en las políticas de estado, hacen que del mismo modo las universidades adecuen sus modelos educativos para que estos sean competitivos y vanguardistas.

La Fundación Universitaria los Libertadores no es la excepción. El ritmo cambiante del mercado y las políticas internas requieren del desarrollo de procesos que impacten en el avance tecnológico de la universidad y su posicionamiento a nivel nacional; así como la interacción de las diferentes decanaturas con las todas aquellas instituciones que soliciten personal altamente calificado en diferentes áreas del conocimiento.

No obstante, es de anotar que la Fundación Universitaria los Libertadores realiza encuentros presenciales en ferias de emprendimiento y de empleo, en ayuda a sus graduandos,

brindándoles oportunidades en el campo laboral. Pero cabe anotar que no todos los egresados pueden asistir a estos eventos y por ende no pueden acceder a este beneficio, lo que limita sus posibilidades para un desempeño laboral acorde con su perfil. Por otro lado, las empresas, organizaciones, instituciones y demás entidades solicitantes de mano de obra calificada no pueden observar con detenimiento los diferentes perfiles a los que podría vincular laboralmente.

Conocedores de esta situación, se busca con la ejecución del presente proyecto suplir esta necesidad creando un espacio virtual que permita a los egresados de ingeniería de sistemas ingresar sus hojas de vida, para que de esta manera las diferentes entidades públicas o privadas puedan establecer contacto directo, no solo con la universidad, sino también con los diferentes profesionales. Por otro lado, permite que la institución pueda llevar un registro más confiable y veraz sobre sus egresados.

DEFINICION DEL PROYECTO

El proyecto describe la necesidad que tiene la Fundación Universitaria Los Libertadores de reconocer la importancia que tiene sus egresados a nivel socio-económico, en el ámbito nacional, es necesario crear un vínculo permanente con ellos de forma tal que facilite la interacción de los Egresados a través de un sistema de información, apoyados en el plan de mejoramiento de la interacción y seguimiento permanente con los Egresados.

CAPÍTULO II

OBJETIVOS

OBJETIVO GENERAL

Crear, organizar, establecer y diseñar un sistema de información que permita integrar a los Egresados en el área de Ingeniería de la Fundación Universitaria Los Libertadores, con las diferentes empresas públicas o privadas que soliciten personal altamente calificado en cualquier área de sus respectivas carreras de Ingeniería para laborar en sus dependencias a nivel regional y/o nacional.

OBJETIVOS ESPECIFICOS

- Diseñar un sistema de información donde los Egresados del área de Ingeniería inscriban sus hojas de vida para ser vistas por las diferentes empresas públicas o privadas a nivel nacional.
- Diseñar, analizar e implementar una base de datos con información de los egresados de la Fundación Universitarias Los Libertadores.
- Mantener un registro de los Egresado del área de Ingeniería y de las Empresas que publican ofertas.
- Diseñar, generar y descargar reportes de la información que los Egresados del área de Ingeniería tengan en la base de datos.

- Ampliar las oportunidades de trabajo de los Egresados del área de Ingeniería de la Fundación Universitaria Los Libertadores.
- Permitir un seguimiento más eficiente por parte de la Fundación Universitaria Los Libertadores sobre los Egresados del área de Ingeniería y su desarrollo en el campo laboral.
- Establecer mecanismos que le permitan a la Fundación Universitaria Los Libertadores observar los requerimientos del mercado laboral y ajustar sus planes de aula si así se requiere.
- Ofertar mano de obra altamente calificada en el área de Ingeniería de la Fundación Universitaria Los Libertadores.

CAPÍTULO III

INFORMACION GENERAL

JUSTIFICACION

La creación de plataformas virtuales como la propuesta en el presente escrito, ayuda no solo a la praxis de quienes intervienen en ella, sino que alimenta las bases de datos de la Fundación Universitaria Los Libertadores de manera más objetiva con información veraz, clara y precisa. Así como también permite ofertar mano de obra calificada del área de Ingeniería a las diferentes entidades que requieran personal idóneo para laborar en las diferentes dependencias de cualquier empresa pública, privada o de capital mixto. De igual manera se busca con la creación de la plataforma poner en práctica los conocimientos adquiridos por el educando durante su tiempo de estudio en las aulas en miras del mejoramiento continuo y la calidad educativa; analizando la pertinencia de los planes de estudio y su impacto dentro del mercado laboral, en concreto para la ingeniería de sistemas. Como también reconocer la importancia de las Tecnologías de la Información y Comunicación (TIC) como instrumento potencial para el crecimiento cultural y económico del pueblo colombiano, en el marco de la Ciencia, Tecnología e Innovación (CTeI).

ALCANCE Y LIMITACIONES

El desarrollo del Sistema de Información aportará un escenario de acciones constante de seguimiento a sus Egresados en el área de Ingeniería, ya sea para la reorganización interna de los programas o para la proposición de nuevos programas, a través de una formación integral para la vida, la familia, el conocimiento y la convivencia.

CAPÍTULO IV

MARCO TEORICO

BASE DE DATOS

Una base de datos consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos de datos. En otras palabras, una base de datos no es más que un conjunto de información (un conjunto de datos) relacionada que se encuentra agrupada o estructurada.

Un sistema de bases de datos es una colección de datos interrelacionados y un conjunto de programas que permiten a los usuarios tener acceso a esos datos y modificarlos.

Los sistemas de bases de datos se diseñan para gestionar grandes cantidades de información. La gestión de los datos implica tanto la definición de estructuras para almacenar la información como la provisión de mecanismos para la manipulación de la información. Además, los sistemas de bases de datos deben garantizar la fiabilidad de la información almacenada, a pesar de las caídas del sistema o de los intentos de acceso no autorizados.

APLICACIONES DE LOS SISTEMAS DE BASE DE DATOS

Las bases de datos se usan ampliamente. Algunas de sus aplicaciones representativas

son:

- **Banca:** Para información de los clientes, cuentas, préstamos y transacciones bancarias.
- **Líneas aéreas:** Para reservas e información de horarios. Las líneas aéreas fueron de las primeras en usar las bases de datos de forma distribuida geográficamente.
- **Universidades:** Para información de los estudiantes, seguimiento a los alumnos egresados, matriculas en las asignaturas y cursos.
- **Telecomunicaciones:** Para guardar un registro de las llamadas realizadas, generar las facturas mensuales, mantener el saldo de las tarjetas telefónicas de prepago y para almacenar información sobre las redes de comunicaciones.
- **Ventas:** Para información de clientes, productos y compras.

CARACTERISTICAS DE LAS BASES DE DATOS

Las características que definen a un Sistema gestor de bases de datos, son las siguientes:

- **Integración de toda la información de la organización:** La base de datos se crea para dar servicio a toda o a una parte importante de la organización y no para unos usuarios particulares; de esta forma se evita la redundancia de datos dentro del sistema de información y los problemas de inconsistencia derivados de ella.
- **Persistencia de los datos:** Los datos deben estar disponibles en todo momento, lo que significa que la base de datos debe almacenarse en un dispositivo de memoria secundaria.
- **Accesibilidad simultánea para distintos usuarios:** Debido al carácter integrador que tiene la base de datos, ésta tendrá que ser compartida por distintos grupos de usuarios, lo que significa que estos podrán acceder simultáneamente a los datos.
- **Independencia de los programas respecto a la representación física de los datos:** Las aplicaciones que se desarrollen para manipular los datos deben ser independientes de la implementación elegida para las estructuras de la base de datos. A esta característica se le conoce como independencia de datos.
- **Mecanismos para controlar la integridad y la seguridad de los datos:** Para que la base de datos refleje fielmente la realidad de la cual es una representación, el SGBD debe asegurar en todo momento la calidad de la información almacenada (integridad) evitando que ésta se deteriore por un uso incorrecto (actualizaciones

que no son válidas, accesos concurrentes no controlados, etc.). Así mismo, debe asegurarse que a la información almacenada sólo acceden las personas autorizadas y en la forma autorizada (seguridad).

Para que el sistema sea útil debe recuperar los datos eficientemente, la necesidad de eficiencia ha llevado a los diseñadores a usar estructuras de datos complejas para la representación de los datos en la base de datos. Dado que muchos de los usuarios de sistemas de bases de datos no tienen formación en informática, los desarrolladores ocultan esa complejidad a los usuarios mediante varios niveles de abstracción para simplificar la interacción de los usuarios con el sistema.

NIVEL FÍSICO

El nivel más bajo de abstracción, describe como se almacenan realmente los datos.

NIVEL LÓGICO

El nivel inmediatamente superior de abstracción describe que datos se almacenan en la base de datos y que relaciones existen entre esos datos.

NIVEL DE VISTAS

El nivel más elevado de abstracción solo describe parte de la base de datos. El nivel de abstracción de vistas existe para simplificar la interacción del usuario con el sistema. El sistema puede proporcionar muchas vistas para la misma base de datos.

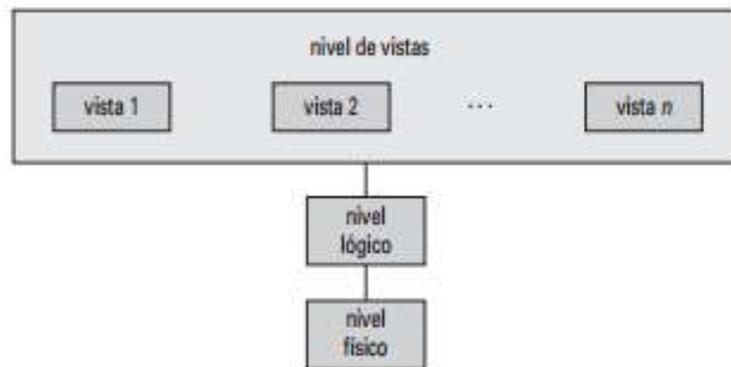


Figura 1. Niveles de abstracción de datos. Sánchez, C. (2006). Recuperado de https://mercadosolanoalan.wikispaces.com/file/view/ima3_niveles_de_abstraccion.jpg/284777424/ima3_niveles_de_abstraccion.jpg

MODELOS DE DATOS

Los modelos de datos ofrecen un modo de describir el diseño de las bases de datos en los niveles físico, lógico y de vistas. Los modelos de datos se pueden clasificar en cuatro categorías diferentes:

MODELO RELACIONAL

El modelo relacional usa una colección de tablas para representar tanto los datos como sus relaciones. Cada tabla tiene varias columnas, y cada columna tiene un nombre único. El modelo relacional es un ejemplo de un modelo basado en registros.

El modelo de datos relacional es el modelo de datos más ampliamente usado, y una amplia mayoría de sistemas de bases de datos actuales se basan en el modelo relacional.

El modelo relacional se encuentra a un nivel de abstracción inferior al modelo de datos E-R. Los diseños de bases de datos a menudo se realizan en el modelo E-R, y después se traducen al modelo relacional.

MODELO ENTIDAD-RELACIÓN

Se basa en una percepción del mundo real que consiste en una colección de objetos básicos, denominados entidades, y de las relaciones entre ellos. Una entidad es un objeto del mundo real que es distinguible de otros objetos.

La estructura lógica general de una base de datos se puede expresar gráficamente mediante un diagrama ER, que consta de los siguientes componentes:

- Rectángulos, que representan conjuntos de entidades.
- Elipses, que representan atributos.
- Rombo, que representan relaciones entre conjuntos de entidades.
- Líneas, que unen los atributos con los conjuntos de entidades y los conjuntos de entidades con las relaciones.

MODELO DE DATOS ORIENTADO A OBJETOS

Se puede considerar como una extensión del modelo entidad-relación con los conceptos de la encapsulación, los métodos y la identidad de los objetos

El modelo de datos relacional orientado a objetos combina las características del modelo de datos orientado a objetos y el modelo de datos relacional.

MODELO DE DATOS SEMIESTRUCTURADOS

El modelo de datos semiestructurados permite la especificación de datos donde los elementos de datos individuales del mismo tipo pueden tener diferentes conjuntos de atributos.

Esto lo diferencia de los modelos de datos mencionados anteriormente, en los que cada elemento de datos de un tipo particular debe tener el mismo conjunto de atributos.

LENGUAJES DE BASES DE DATOS

Un sistema de bases de datos proporciona un lenguaje de definición de datos para especificar el esquema de la base de datos y un lenguaje de manipulación de datos para expresar las consultas a la base de datos y las modificaciones. En la práctica, los lenguajes de definición y manipulación de datos no son dos lenguajes separados; en su lugar simplemente forman partes de un único lenguaje de bases de datos, tal como SQL, ampliamente usado.

LENGUAJE DE DEFINICIÓN DE DATOS

Un esquema de base de datos se especifica mediante un conjunto de definiciones expresadas mediante un lenguaje especial llamado lenguaje de definición de datos (LDD).

Por ejemplo, la siguiente instrucción en el lenguaje SQL define la tabla cuenta:

```
create table cuenta  
(número-cuenta char(10),  
saldo integer)
```

La ejecución de la instrucción LDD anterior crea la tabla cuenta. Además, actualiza un conjunto especial de tablas denominado diccionario de datos o directorio de datos.

LENGUAJE DE MANIPULACIÓN DE DATOS

La manipulación de datos es:

- La recuperación de información almacenada en la base de datos.
- La inserción de información nueva en la base de datos.
- El borrado de información de la base de datos.
- La modificación de información almacenada en la base de datos.

Un lenguaje de manipulación de datos (LMD) es un lenguaje que permite a los usuarios acceder o manipular los datos organizados mediante el modelo de datos apropiado.

MYSQL

MySQL es una base de datos relacional que utiliza el lenguaje SQL (Structured Query Language Lenguaje de consulta estructurado). Se trata de un sistema gestor de bases de datos de código abierto, lanzado en 1995, que más tarde fue adquirido por Sun Microsystems en 2008(más tarde, en 2009, Oracle compro Sun).

SQL es el nombre de un lenguaje desarrollado para la formulación de búsquedas en bases de datos. Surgió a principios de los años 70, y se basa en el idioma Ingles para nombrar sus comandos, y actualmente está estandarizado por el lenguaje ISO y ANSI. Actualmente, la mayoría de sistemas de bases de datos relacional y otros formatos tratan de seguir el estándar SQL para formalizar sus consultas, aunque cada empresa que usa el lenguaje lo adapta a sus necesidades.

MySQL es la base de datos de código abierto mas popular del mundo. Código abierto significa que todo el mundo puede acceder al código fuente, es decir, al código de programación de MySQL. Todo el mundo puede contribuir para incluir elementos, arreglar

problemas, realizar mejoras o sugerir optimizaciones. Y así ocurre. MySQL ha pasado de ser una "pequeña" base de datos a una completa herramienta y ha conseguido superar a una gran cantidad de bases de datos comerciales.

CARACTERÍSTICAS DE MYSQL

- MySQL está escrito en C y C++
- Emplea el lenguaje SQL para consultas a la base de datos.
- MySQL Server está disponible como freeware bajo licencia GPL.
- Funciona en diferentes plataformas
- Proporciona sistemas de almacenamiento transaccionales y no transaccionales
- Un sistema de reserva de memoria muy rápido basado en threads
- Un sistema de privilegios y contraseñas que es muy flexible y seguro, y que permite verificación basada en el host

VENTAJAS DE MYSQL

- **Coste:** El coste de MySQL es gratuito para la mayor parte de los usos y su servicio de asistencia resulta económico.
- **Velocidad:** MySQL es mucho más rápido que la mayor parte de sus rivales.
- **Funcionalidad:** MySQL dispone de muchas de las funciones que exigen los desarrolladores profesionales, como compatibilidad completa con ACID, compatibilidad para la mayor parte de SQL ANSI, volcados online, duplicación, funciones SSL e integración con la mayor parte de los entornos de programación. Así mismo, se desarrolla y actualiza de forma mucho más rápida que muchos de sus

rivales, por lo que prácticamente todas las funciones estándar de MySQL todavía no están en fase de desarrollo.

- **Portabilidad:** MySQL se ejecuta en la inmensa mayoría de sistemas operativos y, la mayor parte de los casos, los datos se pueden transferir de un sistema a otro sin dificultad.
- **Facilidad de uso:** MySQL resulta fácil de utilizar y de administrar. Gran parte de las viejas bases de datos presentan problemas por utilizar sistemas obsoletos, lo que complica innecesariamente las tareas de administración. Las herramientas de MySQL son potentes y flexibles, sin sacrificar su capacidad de uso.

PHP

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

PHP es un lenguaje interpretado de propósito general ampliamente usado, diseñado especialmente para desarrollo web y que puede ser incrustado dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno.

PHP es un lenguaje de secuencia de comandos de servidor cuyo código fuente es público, es multiplataforma y se puede incrustar dentro del código de las páginas HTML. Los

programadores de PHP escriben su código directamente dentro de las paginas HTML. Luego, el servidor Web ejecutara el código PHP que está incluido dentro de la página HTML, antes de devolver esta al explorador de internet del usuario.

PHP está enfocado principalmente a la programación de scripts del lado del servidor, por lo que se puede hacer cualquier cosa que pueda hacer otro programa similar, como recopilar datos de formularios, generar páginas con contenidos dinámicos, o enviar y recibir cookies. Aunque PHP puede hacer mucho más.

VENTAJAS DE PHP

- PHP puede emplearse en todos los sistemas operativos principales, incluyendo Linux, muchas variantes de Unix (incluyendo HP-UX, Solaris y OpenBSD), Microsoft Windows, Mac OS X, RISC OS y probablemente otros más. PHP admite la mayoría de servidores web de hoy en día, incluyendo Apache, IIS, y muchos otros.
- Con PHP, se tiene la libertad de elegir el sistema operativo y el servidor web. Además, se tiene la posibilidad de utilizar programación por procedimientos o programación orientada a objetos (POO), o una mezcla de ambas.
- Entre las capacidades de PHP se incluyen la creación de imágenes, ficheros PDF e incluso películas Flash (usando libswf y Ming) generadas sobre la marcha. También se puede generar fácilmente cualquier tipo de texto, como XHTML y cualquier otro tipo de fichero XML. PHP puede autogenerar estos ficheros y guardarlos en el sistema de ficheros en vez de imprimirlos en pantalla, creando una caché en el lado del servidor para contenido dinámico.

- Una de las características más potentes y destacables de PHP es su soporte para un amplio abanico de bases de datos. Escribir una página web con acceso a una base de datos es increíblemente simple utilizando una de las extensiones específicas de bases de datos (p.ej., para mysql), o utilizar una capa de abstracción como PDO, o conectarse a cualquier base de datos que admita el estándar de Conexión Abierta a Bases de Datos por medio de la extensión ODBC
- PHP tiene útiles características de procesamiento de texto, las cuales incluyen las expresiones regulares compatibles con Perl (PCRE), y muchas extensiones y herramientas para el acceso y análisis de documentos XML.
- Lo mejor de utilizar PHP es su extrema simplicidad para el principiante, pero a su vez ofrece muchas características avanzadas para los programadores profesionales.

DESVENTAJAS DE PHP

- El manejo de errores no es tan sofisticado como otros lenguajes (ASP.net)
- No existe IDE o Debugger consolidado. Aunque en la actualidad existen proyectos varios entre los más destacados esta Eclipse (gratis) o Zend Studio (de pago).

CAPÍTULO V

MARCO METODOLOGICO

PROTOTIPADO EVOLUTIVO

Para la realización de nuestro sistema de información de egresados Los Libertadores, nos propondremos trabajar la Metodología de Desarrollo de Prototipo Evolutivo.

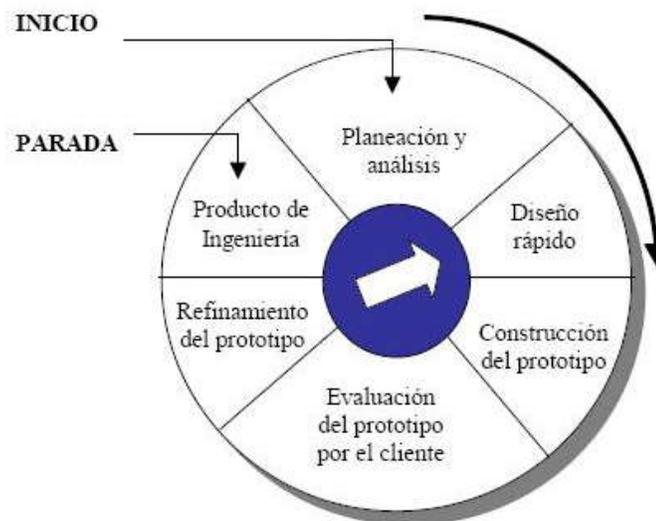


Figura 2. Ciclo Metodología Prototipado Evolutivo

Fuente: <https://requisitosdesoftware.wikispaces.com/file/view/Prototipado.JPG/73698683/Prototipado.JPG>

En la actualidad, la parte de Egresados de Ingeniería de Sistemas de la Fundación Universitaria Los Libertadores de Bogotá, presenta muchas dificultades asociadas al proceso de gestión y análisis de la información. Esto se considera particularmente crítico, toda vez que los egresados, además de hacer parte natural de la Familia Sistémica, resultan esenciales para medir el nivel de impacto en la sociedad, la calidad general de los programas ofrecidos por la Universidad, y una fuente de retroalimentación para orientar la extensión y la formación profesional.

Por lo tanto, se decidió implementar una herramienta de software para automatizar la recolección y administración de los datos de egresados, tales como: información personal, actualizaciones, vínculos laborales, publicaciones, etc.

Esta herramienta de software permite que cada egresado ingrese y actualice sus datos, diligenciando una encuesta de acceso web, que se generen informes estadísticos y consultas según sus requerimientos; y, lo más importante, les debe permitir generar publicaciones para unificar información entre los exalumnos.

Es por ese motivo que se decidió que el modelo de desarrollo que más favorece un intercambio de información con los usuarios finales es el Modelo de Prototipo Evolutivo, porque en él se puede ir retroalimentando luego de producir muchas pruebas antes de liberar un nuevo prototipo e ir mejorando cada uno de los puntos de la problemática planteada y la que pueda salir mediante la presentación de cada prototipo.

PROCEDIMIENTO A SEGUIR PARA LA METODOLOGÍA ESCOGIDA

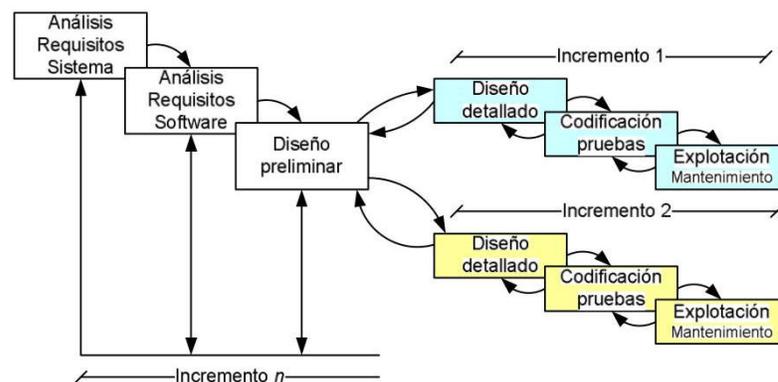


Figura 3. Algoritmo Metodología de Desarrollo Prototipado
Fuente: <http://osc.co.cr/wp-content/uploads/2011/06/incremental2.jpeg>

FASE DE DEFINICION

En esta primera etapa del proyecto se realizó la recolección y refinamiento de requisitos, se identificó la situación a mejorar, la cual dio pie a la elaboración de nuestro proyecto, apoyados en el plan de mejoramiento de la interacción y seguimiento permanente con los egresados, definimos los objetivos junto con el alcance y la metodología a seguir. Se hizo la recolección de la información a graduados registrados en las bases de datos, para la planeación de la estrategia a utilizar a la aplicación de encuestas de seguimiento a graduados de nuestra aplicación web.

El desarrollador y algunos de los usuarios, se reunieron para definir los objetivos globales para el software, identifican los requisitos conocidos y perfilan las áreas en donde serán necesarias una mayor definición.

FASE DE ANALISIS

Se realizó el diseño del sistema, analizando cada uno de los requerimientos y poniéndolos en un prototipo, se realizó el diseño UML y el diseño rápido del sistema.

Se hizo la determinación del medio adecuado de comunicación ya sea vía e-mail, alguna red social, o por envío de encuestas, con el fin de mantener contacto con los egresados, compartir información, programar eventos y actividades, para garantizar el mayor número de egresados contesten la encuesta, fomentando la integración y pertenencia.

FASE DE DESARROLLO E IMPLEMENTACION

El desarrollo de este proyecto, se implementó y desarrollo principalmente las partes más importantes de la herramienta con base a los requerimientos planteados, en los objetivos específicos.

Para la implementación de los prototipos se utilizaron las siguientes herramientas:

- Sublime Text.
- Xampp.
- Wow Slider.
- Workbench Mysql.
- Mozilla Firefox - Complementos Firefox Mozilla Firebug.
- Eclipse.
- NotePad ++.

Con estas herramientas de programación y el sistema gestor de las bases de datos, se codificará las páginas y se estructuraran los datos.

La construcción de prototipos comienza con la recolección de los requisitos.

Luego se le diseño el prototipo que se enfocó sobre la representación de los aspectos del software visibles al usuario (por ejemplos, métodos de entrada y formatos de salida) y se prosigue a su construcción.

El prototipo fue evaluado por el usuario y se utiliza para refinar los requisitos del software a desarrollar. Se produce un proceso interactivo donde el prototipo es afinado

(Refinamiento del prototipo) para que satisfaga las necesidades del usuario, al mismo tiempo que facilita al desarrollador una mejor comprensión de lo que hay que hacer.

VENTAJAS

- Este modelo fue útil porque el cliente conoce los objetivos generales para el software, pero no identifico los requisitos detallados de entrada, procesamiento o salida.
- También ofrece un mejor enfoque cuando los desarrolladores del software tienen poco conocimiento sobre la forma que debería tomar la interacción humano-máquina.

DESVENTAJAS

- El usuario tiende a crearse unas expectativas cuando ve el prototipo de cara al sistema final. A causa de la intención de crear un prototipo de forma rápida, se suelen desatender aspectos importantes, tales como la calidad y el mantenimiento a largo plazo, lo que obliga en la mayor parte de los casos a reconstruirlo una vez que el prototipo ha cumplido su función.

CAPÍTULO VI

DIAGRAMAS

LENGUAJE UNIFICADO DE MODELADO

UML es un popular lenguaje de modelado de sistemas de software. Se trata de un lenguaje gráfico para construir, documentar, visualizar y especificar un sistema de software. Entre otras palabras, UML se utiliza para definir un sistema de software.

Posee la riqueza suficiente como para crear un modelo del sistema, pudiendo modelar los procesos de negocios, funciones, esquemas de bases de datos, expresiones de lenguajes de programación, etc.

DEFINICION DE CASO DE USO

Los casos de uso se crean para refinar un conjunto de requisitos de acuerdo con una función o tarea. En lugar de la tradicional lista de requisitos que quizá no trate de forma directa el uso de la solución, los casos de uso reúnen requisitos comunes basados en el tipo de función u objetivo. Los casos de uso definen qué harán los usuarios o funciones en la solución y un proceso empresarial define cómo realizarán esas funciones.

DIAGRAMA DE CASO DE USO

Este diagrama representa la funcionalidad completa de un sistema (o una clase) mostrando su interacción con los agentes externos. Esta representación se hace a través de las relaciones entre los actores (agentes externos) y los casos de uso (acciones) dentro del sistema. Los diagramas de casos de uso definen conjuntos de funcionalidades afines que el sistema debe cumplir para satisfacer todos los requerimientos que tiene a su cargo. Esos conjuntos de

funcionalidades son representados por los casos de uso. Se pueden visualizar como las funciones más importantes que la aplicación puede realizar o como las opciones presentes en el menú de la aplicación.

ACTORES

Un actor es una clase de persona, organización, dispositivo o componente de software externo que interactúa con el sistema.

En la figura a continuación se identifican tres actores para el sistema de información de Egresados Ingeniería de sistemas



*Figura 4. Actores.
Fuente: Elaboración Propia*

CASO DE USO

Un caso de uso especifica una interacción entre un actor y el sistema de modo tal que pueda ser entendida por una persona sin conocimientos técnicos. Es importante también que capte una función visible para un actor. Generalmente se representa en forma de elipse.



*Figura 5. Caso de Uso.
Fuente: Elaboración Propia*

RELACIONES

Indican la relación existente entre un caso de uso y un actor, existen los siguientes tipos de relaciones:

- **Comunicación:** Relación (asociación) entre un actor y un caso de uso. El estereotipo de la relación de comunicación es: <<communicate>> aunque generalmente no se estipula ningún nombre.
- **Inclusión (include):** Un caso de uso base incorpora explícitamente el comportamiento de otro en algún lugar de su secuencia. La relación de inclusión sirve para enriquecer un caso de uso con otro y compartir una funcionalidad común entre varios casos de uso, también puede utilizarse para estructurar un caso de uso describiendo sus subfunciones. El caso de uso incluido existe únicamente con ese propósito, ya que no responde a un objetivo de un actor.
- **Extensión (extends):** Un caso de uso base incorpora implícitamente el comportamiento de otro caso de uso en el lugar especificado indirectamente por este otro caso de uso. En el caso de uso base, la extensión se hace en una serie de

DIAGRAMA DE LOS CASOS DE USOS DEL SISTEMA DE INFORMACION PARA EL SEGUIMIENTO A LOS EGRESADOS EN EL AREA DE INGENIERIA DE LA FUNDACIÓN UNIVERSITARIA LOS LIBERTADORES

DIAGRAMA DE CASO DE USO DE LOS EGRESADOS

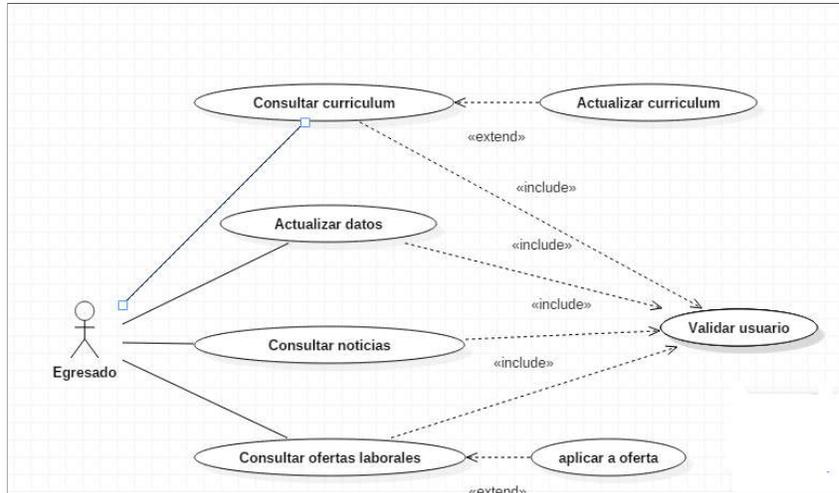


Figura 7. Diagrama de caso de uso de los Egresados.
Fuente: Elaboración propia

DIAGRAMA DE CASO DE USO DE LAS EMPRESAS

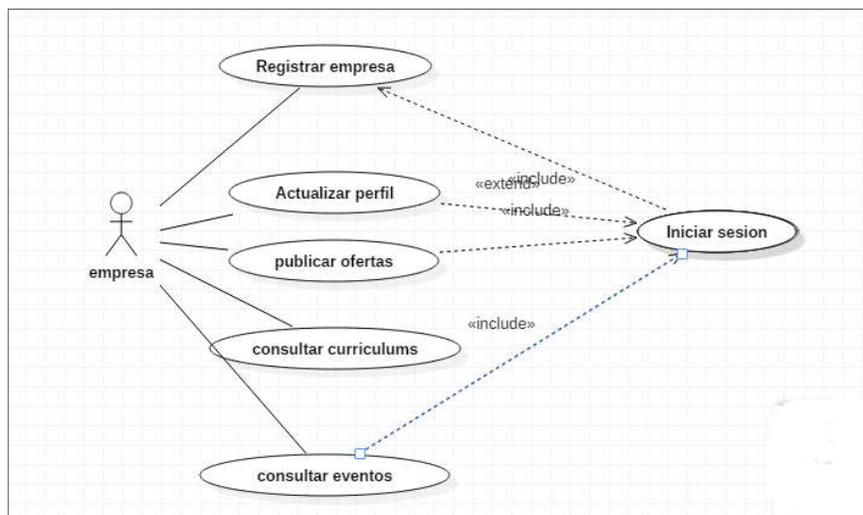


Figura 8. Diagrama de caso de uso de las Empresas.
Fuente: Elaboración Propia

**DESCRIPCION DEL CASO DE USO DE INICIO DE SESION DE LOS
EGRESADOS**

| CASO DE USO | INICIO SESION |
|-----------------------|---|
| DESCRIPCION | EL usuario realizara el ingreso a la aplicación, digitando su usuario y contraseña, previamente creados en la base. El sistema realiza la validación y posteriormente dará acceso al usuario a la página principal donde se habilitarán los menús para egresados. |
| ACTORES | Egresado |
| PRE CONDICION | El usuario debe estar creado previamente en la base de datos. |
| POST CONDICION | Después de 5 minutos de inactividad, la sesión se cerrará. |
| PROCESO | 1. Acceder a la página web |
| | 2. Mostrar ventana emergente para ingreso de usuario y contraseña |
| | 3. Ingresar usuario y contraseña. |
| | 4. Realizar validación de los datos ingresados contra los registrados en la base de datos |
| | 5. Ingresar a la página con los menús activos para egresados |

*Tabla 1. Descripción del caso de uso de inicio de sesión de los Egresados
Fuente: Elaboración Propia*

**DESCRIPCION DEL CASO DE USO DE ACTUALIZACION DEL
CURRICULO DE LOS EGRESADOS**

| CASO DE USO | ACTUALIZAR CURRICULO |
|-----------------------|--|
| DESCRIPCION | El usuario podrá actualizar sus datos personales, a excepción de los nombres, numero de documento, lugar de expedición y fecha de nacimiento |
| ACTORES | Egresado |
| PRE CONDICION | El usuario debe haber iniciado sesión |
| POST CONDICION | Se actualizan los datos del estudiante. |
| PROCESO | 1. Seleccionar la opción actualizar currículo |
| | 2. Muestra formulario solicitando la información a actualizar. |
| | 3. Usuario actualiza los datos |
| | 4. Seleccionar el botón guardar currículo |
| | 5. Mostrar mensaje de verificación de actualización. |

*Tabla 2. Descripción del caso de uso de actualización de currículo de los Egresados
Fuente: Elaboración Propia*

**DESCRIPCION DEL CASO DE USO DE CONSULTAR OFERTAS DE LAS
EMPRESAS**

| CASO DE USO | CONSULTAR OFERTAS |
|----------------------|--|
| DESCRIPCION | El usuario podrá acceder a consultar las ofertas laborales publicadas por las empresas registradas |
| ACTORES | Egresado |
| PRE CONDICION | 1. El usuario debe haber iniciado sesión |
| PROCESO | 2. Seleccionar la opción ofertas laborales |
| | 3. El sistema muestra las ofertas publicadas por las empresas |
| | 4. Usuario selecciona un rango de tiempo |
| | 5. El sistema filtra las ofertas según el rango escogido. |

*Tabla 3. Descripción del caso de uso de consultar ofertas de las Empresas
Fuente: Elaboración de Propia*

DESCRIPCION DEL CASO DE USO DE GENERAR REPORTES DESDE LA ADMINISTRACION

| CASO DE USO | GENERAR REPORTES |
|-----------------------|---|
| DESCRIPCION | Podrá consultar información sobre los egresados, empresas registradas y ofertas publicadas. |
| ACTORES | Administrador |
| PRE CONDICION | El usuario debe haber iniciado sesión |
| POST CONDICION | Después de 5 minutos de inactividad, la sesión se cerrará |
| PROCESO | 1. Seleccionar la opción generar reportes |
| | 2. El sistema muestra menú desplegable con las opciones de reporte. |
| | 3. El usuario seleccionar uno o más filtros |
| | 4. El sistema muestra en pantalla el detalle del reporte. |
| | 5. El usuario tendrá la opción de exportar el reporte a un archivo Excel |

*Tabla 4. Descripción del caso de uso de generar reportes desde la Administración
Fuente: Elaboración Propia*

**DESCRIPCION DEL CASO DE USO PARA ADMINISTRAR LAS
PUBLICACIONES DESDE LA ADMINISTRACION**

| CASO DE USO | ADMINISTRAR PUBLICACIONES |
|-----------------------|---|
| DESCRIPCION | El administrador podrá publicar y borrar eventos, noticias de interés y fotos. |
| ACTORES | Administrador |
| PRE CONDICION | El usuario debe haber iniciado sesión |
| POST CONDICION | La página se habrá actualizado con la nueva publicación |
| PROCESO | 1. El administrador selecciona el botón publicar eventos. |
| | 2. El sistema muestra el panel publicación de eventos. |
| | 3. El administrador podrá cargar imágenes para la publicación en la opción seleccionar archivo y guardar imagen |
| | 4. El administrador podrá darle un título a la publicación en el cuadro de texto Titulo Evento |
| | 5. El administrador podrá escribir una descripción del evento en el cuadro de texto Detalle evento |
| | 6. El administrador podrá seleccionar que personas pueden ver el evento publicado en la opción Nivel privacidad evento. |

| | |
|--|---|
| | 7. El administrador seleccionara la opción Activo, o inactivo cuando desee que la publicación no continúe visible en la página. |
| | 8. Seleccionar el botón Guardar título y detalle del evento. |

*Tabla 5. Descripción del caso de uso para administrar las publicaciones desde la Administración
Fuente: Elaboración Propia*

**DESCRIPCION DEL CASO DE USO PARA APROBAR LAS EMPRESAS
REGISTRADAS DESDE LA ADMINISTRACION**

| CASO DE USO | APROBAR EMPRESAS |
|-----------------------|---|
| DESCRIPCION | Cada que se registre una empresa en la página, el administrador podrá aprobar que realice publicaciones de empleo |
| ACTORES | Administrador |
| PRE CONDICION | El usuario debe haber iniciado sesión |
| POST CONDICION | Después de 5 minutos de inactividad, la sesión se cerrará |
| PROCESO | 1. Seleccionar la opción Empresas Registradas. |
| | 2. Seleccionar la opción ver empresas pendientes. |
| | 3. Dar clic sobre la opción Editar/asignar de la empresa que se quiere activar. |

| | |
|--|---|
| | 4. Digitar usuario y password |
| | 5. En el menú desplegable estado, seleccionar la opción Activo. |
| | 6. Seleccionar el botón Guardar Datos |

*Tabla 6. Descripción del caso de uso para aprobar las empresas registradas desde la Administración
Fuente: Elaboración Propia*

CAPITULO VII

EXPLICACION DE LA HERRAMIENTA

DESARROLLO DE LA HERRAMIENTA

El producto de este proyecto es una primera versión del sistema de información de los Egresados de Ingeniería de Sistemas de la Fundación Universitaria Los Libertadores, analizamos la gran importancia que planteamos desde un comienzo, y fue posible realizar mejoras para el software de entrega.

Se utilizó la metodología de preguntas y respuestas, en el que, mediante una elaboración de preguntas y pruebas, se logró hacer un correcto análisis de requerimientos y se inició la elaboración de un primer prototipo que se fue enriqueciendo a medida que fue avanzando el proyecto

ANALISIS DE REQUISITOS

REQUISITOS ADMINISTRADOR

- La interfaz debe llevar el perfil para Administrador que tenga las opciones normales de un egresado registrado dentro del sistema, además las opciones exclusivas de un Administrador del Sistema, controla y actualiza el sistema de información.
- El Sistema de información deberá contener un sistema de validación, como página principal para que el administrado pueda acceder a su perfil.

- El administrador debe encontrar entre sus opciones, la consulta de todos los usuarios y las empresas bajo varios criterios de búsquedas.
- El Sistema debe permitir Activar o Denegar el acceso a cualquier perfil ya sea Egresado o Empresa.
- El Administrador podrá borrar publicaciones de los demás Usuarios o Empresas del Sistema, y también hacer publicaciones.
- El Sistemas debe permitir al Administrador, gestionar la información de pregrado y demás temas de interés para los Egresados.
- El Sistema debe permitir mantener actualizar la base de datos de los usuarios y de las empresas por medio de una interfaz que permita subir esta información al sistema.
- El administrador debe tener una interfaz que permita eliminar y compartir fotos del sistema.
- Para las publicaciones de noticias y los eventos, el Administrador podrá escoger el nivel de visibilidad para el que desea su publicación, sea para un Egresado, para las Empresas o para el público en general.
- El Administrador podrá crear, modificar o eliminar las contraseñas de los perfiles.

- El sistema de información debe tener una interfaz que permita descargar la información de los usuarios y las empresas en un formato digital.

REQUISITOS DE USUARIOS

- En este módulo el usuario tendrá una opción de “Inicio de Sesión” y la interfaz le pedirá la cuenta y la clave para el ingreso.
- La interfaz debe llevar el perfil de Usuario, para que los registrados puedan manejar su propia información, modificar sus datos personales, sus respuestas e incluso sus publicaciones.
- El Sistema debe permitir al Egresado, actualizar sus datos.
- La interfaz permitirá tener su propio perfil personal y profesional, con la facilidad de actualizar cuando el usuario lo prefiera.
- La interfaz debe ofrecer unas opciones: Nuevas oportunidades de empleo, Actualizar la Hoja de vida, Visualizar las Noticias y los Eventos de la Universidad.
- El Sistema debe permitir al Egresado, poder postularse a cualquier oferta de trabajo.
- El sistema brindará a los usuarios la información actualizada sobre pregrado y demás temas de interés del Egresado, previamente expuestos por el Administrador.

REQUISITOS EMPRESAS

- En este módulo la empresa tendrá una opción de “Inicio de Sesión” y la interfaz le pedirá la cuenta y la clave para el ingreso.
- La interfaz debe llevar el perfil de la Empresa, para que los registrados puedan manejar su propia información, modificar sus datos personales y sus ofertas laborales.
- El Sistema debe permitir a la Empresa, actualizar sus datos.
- La interfaz permitirá tener su propio perfil personal, con la facilidad de actualizar cuando la Empresa lo prefiera.
- La interfaz debe ofrecer unas opciones: Las postulaciones de los Egresados a las ofertas de trabajo, Actualizar las ofertas laborales, Visualizar las Noticias y los Eventos de la Universidad.
- El sistema brindará a las Empresas la información la Hoja de Vida sobre los Egresados.

CAPITULO VIII

EXPLICACION DEL CODIGO

EXPLICACION DE LOS LENGUAJES

HTML

Es un lenguaje markup (de hecho, las siglas de HTML significan Hyper Text Markup Language) usado para estructurar y presentar el contenido para la web. Es uno de los aspectos fundamentales para el funcionamiento de los sitios, pero no es el primero. Es de hecho la quinta revisión del estándar que fue creado en 1990. A fines del año pasado, la W3C la recomendó para transformarse en el estándar a ser usado en el desarrollo de proyectos venideros. Por así decirlo, qué es HTML5 está relacionado también con la entrada en decadencia del viejo estándar HTML 4, que se combinaba con otros lenguajes para producir los sitios que podemos ver hoy en día. Con HTML5, tenemos otras posibilidades para explotar usando menos recursos. Con HTML5, también entra en desuso el formato XHTML, dado que ya no sería necesaria su implementación.

PHP

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. Lo que distingue a PHP de algo del lado del cliente como JavaScript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP, por lo que no hay manera de que los usuarios puedan saber qué se tiene debajo de la manga.

JAVA SCRIPT

Es un lenguaje de programación, al igual que PHP, si bien tiene diferencias importantes con éste. JavaScript se utiliza principalmente del lado del cliente (es decir, se ejecuta en nuestro ordenador, no en el servidor) permitiendo crear efectos atractivos y dinámicos en las páginas web. Los navegadores modernos interpretan el código JavaScript integrado en las páginas web. La ventaja de JavaScript es que al estar alojado en el ordenador del usuario los efectos son muy rápidos y dinámicos. Al ser un lenguaje de programación permite toda la potencia de la programación como uso de variables, condicionales, bucles, etc.

El proyecto consta de una página principal llamada index.html donde se muestra toda la configuración de la página principal, el archivo index.html se divide en dos partes fundamentales el encabezamiento o cabecera y el cuerpo del documento.

EXPLICACION DEL CODIGO INDEX.HTML

Cuando construimos una página de internet el primer paso que hay que dar es crear el archivo index.html. El archivo index.html es la página principal del dominio al cual se accede. Al acceder un dominio cualquiera el servidor donde se encuentra ese dominio buscará el archivo llamado index.html donde se incluye todo el texto, imágenes, códigos, etc. que se quieran mostrar en la página principal. O sea, el archivo index.html es la página principal.

LA CABECERA O ENCABEZADO

La funcionalidad de la cabecera es que contiene una serie de datos que el navegador necesita para interpretar correctamente el documento, pero que no se visualizan dentro del mismo. La cabecera tiene, por tanto, una función no visible pero fundamental. Los atributos que se utilizó en la cabecera del proyecto fueron los siguientes:

<DOCTYPE>: Es el primer elemento que abriera un documento. Cuya función es proporcionar información al servidor de web anfitrión además también permite la comunicación de la página entre navegador y servidor.

<HTML>: Todos los elementos y el contenido de un documento HTML están encerrados dentro de las marcas `<HTML></HTML>` cuya función es servir de apertura y cierre al archivo. Las marcas `<HTML></HTML>` indican al navegador que el documento está marcado en HTML.

<HEAD>: Los elementos `<HEAD></HEAD>` se colocan inmediatamente después de la apertura de la marca `<HTML>` y cierran el encabezamiento propiamente dicho del documento; es decir toda la información que necesita el navegador, el servidor de web y los motores de búsqueda. Se trata del primer elemento que lee el navegador y, por ello, es el mejor lugar para insertar sintaxis script. Dentro de `<HEAD></HEAD>` se introduce el título del documento y otros datos.

<META>: Se utiliza para añadir información sobre la página. Esta información puede ser utilizada por los buscadores. Los buscadores consultan la información de la etiqueta `<meta>` de las páginas, buscando coincidencias con lo que el usuario pretende encontrar. A través de

esta etiqueta pueden especificarse los atributos name y content. El atributo name indica el tipo de información, y el atributo content indica el valor de dicha información.

Para cada etiqueta <meta> solo es posible indicar un tipo de información y su valor, pero es posible insertar varias etiquetas <meta> en un mismo documento. La etiqueta <meta> ha de estar entre las etiquetas <head> y </head>.

| TIPO | DESCRIPCIÓN |
|----------------|---|
| Author | Autor de la página |
| Classification | Palabras para clasificar la página en los buscadores. |
| Description | Descripción del contenido de la página |
| Generator | Programa utilizado para crear la página. |
| Keywords | Palabras claves. |

Tabla 7. Variables que se utiliza en la cabecera o encabezado
Fuente: Admin. (2013), Variables que se utiliza dentro del TYPE. Recuperado de <http://www.aprendoencasa.com/index.php/2013/09/19/etiqueta-script-curso-prctico-html5/>

<SCRIPT>: Permite la inserción de códigos script, como puede ser JavaScript en un documento web del lado del cliente. El código se puede insertar directamente entre las etiquetas o haciendo referencia a un documento externo el cual contiene el código script a ejecutar.

- **TYPE**: Permite especificar el tipo MIME del script. El tipo MIME está compuesto de dos partes una el tipo seguido de una barra y el subtipo. Por defecto el tipo MIME en caso de no especificarse es “text/JavaScript”. Entre los posibles valores de este atributo se pueden encontrar las siguientes, siendo los más comunes.

| TIPO | DESCRIPCIÓN |
|------------------------|---|
| Text/JavaScript | Scripts en lenguaje JavaScript. |
| Text/ecmascript | Especificación de lenguaje de programación publicada por ECMA International basado en JavaScript. |
| Application/JavaScript | Igual que text/JavaScript, aunque es preferible utilizar el anterior. |
| Text/vbscript | Especificación de lenguaje de programación interpretado basado en Visual Basic. |

Tabla 8. Variables que se utiliza dentro del TYPE

Fuente: AulaClic. (2004), Variables que se utiliza en la cabecera o encabezado. Recuperado de http://www.aulacli.es/html/a_2_1_1.htm

- **SRC:** El atributo permite especificar la dirección URL del recurso al cual se quiere acceder, siendo en este caso el archivo que contiene el código de script a cargar en el documento web.

<LINK>: Permite realizar la carga y establecer la relación existente entre el documento web actual y un recurso externo.

- **HREF:** Este atributo permite establecer la URL en la que se encuentra el documento el cual se quiere enlazar, con el documento web.
- **HREFLANG:** Este atributo permite establecer el lenguaje del documento que se está enlazando, aunque no realiza ninguna función especial en los navegadores, puede ser utilizado por los motores de búsqueda para establecer el idioma.

- **REL:** Este atributo sirve para especificar la relación existente entre el documento web y el documento con el que se está realizando el enlace. El atributo es obligatorio y se ha de incluir.
- **SIZE:** El atributo se utiliza sólo en caso de haber especificado anteriormente en el atributo rel el valor icon, y permite especificar el tamaño del atributo con el cual se está enlazando.
- **TYPE:** Sirve para especificar el tipo MIME del documento con el que se está realizando el enlace.
- **MEDIA:** Este atributo sirve para indicar el tipo de dispositivo para el cual se está realizando el enlace.

<TITLE>: Sirve para dar título a la página. El título suele ser visualizado por los navegadores en la cabecera de página.

```

<!-- ***** -->
<!-- INDEX GENERAL PARA LOGIN INICIAL -->
<!-- ***** -->

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>!-- -->
    <meta name="author" content="Raúl Lopez - Omar Arias - Alejandro Fiesco" />
    <meta name="description" content="Programa de Seguimiento de los Egresados de la Fundación Universitaria Los Libertadores">
    <meta name="keywords" content="Ingeniería de Sistemas" />
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

EL CUERPO

El cuerpo de un documento HTML estará delimitado por las etiquetas `<BODY>` y `</BODY>` y en él se incluirán todas las instrucciones HTML y el texto que forman el documento, al igual que la cabecera `<HEAD>` es opcional, pero se recomienda para la buena identificación de las distintas zonas del documento. Si un documento no presenta ninguna de las etiquetas de identificación de sus distintas partes (`<HTML>`, `<HEAD>` o `<BODY>`) se considerará que todo lo que se defina pertenece al cuerpo del documento. Los atributos que se utilizó en el cuerpo del proyecto fueron los siguientes:

<BODY>: Se coloca inmediatamente después del cierre de la marca `</HEAD>` y, en cualquier caso, dentro de los elementos `<HTML></HTML>`; tiene una marca de apertura y una de cierre y delimitado por ellas figura el cuerpo del documento. Si el elemento `<HEAD>` contenía meta información (es decir, datos no visualizados materialmente por el navegador) la función de la marca `<BODY>` es la de mostrar los objetos (texto, imágenes, sonidos, apliques, etc.) de la página.

<DIV>: Se emplea para definir un bloque de contenido o sección de la página, para poder aplicarle diferentes estilos e incluso para realizar operaciones sobre ese bloque específico.

- **ID**: Permite establecer un identificador único para el bloque. Así podremos referimos al bloque de forma inequívoca.
- **CLASS**: Es similar a id, pero con la ventaja de que se puede repetir, así que podemos tener varios `<div>` diferentes con la misma clase.

- **SPAN:** Después de un `</div>` se comienza con una nueva línea. Con ella conseguimos un efecto similar, pero en el interior de un párrafo. Se emplea con frecuencia para aplicar estilos en el interior de un bloque de texto.

<HEADER>: Representa un grupo de ayudas introductorias o de navegación. Puede contener algunos elementos de encabezado, pero también otros elementos como un logo, una sección que aglutine secciones de encabezados, un formulario de búsqueda o cosas parecidas.

: El texto comprendido entre estas marcas queda formateado en negrita (la B corresponde al inglés BOLD).

<I>: El texto comprendido entre estas marcas queda formateado en cursiva (la I corresponde al inglés ITALIC).

<U>: El texto comprendido entre estas marcas queda subrayado aun sin ser un enlace (la U corresponde al inglés UNDERLINE).

<A>: Lo característico del lenguaje HTML es el poder generar vínculos de hipertexto para enlazar con ellos todos sus documentos en web.

**
:** Saltos de línea. Esta tag sirve para realizar un salto de línea, puede poner tantas como desee y realizará un salto de línea por cada una de ellas.

<FORM>: Representa una sección de un documento que contiene controles interactivos que permiten a un usuario enviar información a un servidor web.

<LABEL>: Representa una etiqueta para un elemento en una interfaz de usuario. Este puede estar asociado con un control ya sea mediante la utilización del atributo for, o ubicando el control dentro del elemento label. Tal control es llamado "el control etiquetado" del elemento label.

<INPUT>: Se utiliza para el ingreso de datos por parte del usuario. El campo del input dependerá del tipo de control que le asignemos al atributo type.

- **TEXT**: Entrada de texto de una línea.
- **PASSWORD**: Entrada de texto de una línea con caracteres ocultos.
- **CHECKBOX**: Casilla de verificación.
- **RADIO**: Radio botón.
- **SUBMIT**: Botón de envío.
- **IMAGE**: Botón de envío gráfico o mapa de imágenes.
- **RESET**: Botón de re inicialización.
- **BUTTON**: Botón pulsador.
- **HIDDEN**: Control oculto.
- **FILE**: Selección de archivo.

<H1...H6>: Se emplea para definir el estilo de los encabezados de la página, dándoles más o menos relieve según el número correspondiente. Los números van del 1 al 6, siendo más importantes los números más bajos que los altos. Visualmente, esta diversa importancia se plasma en la diferente dimensión del texto.

```

<body>
  <div class="mainWrap">
    <div id="modal" class="popupContainer" style="display:none;">!-- div emergente para login -->
    <header class="popupheader">
      <span class="header_title">Ingreso</span>
      <span class="modal_close"><i class="fa fa-times"></i></span>
    </header>
    <section class="popupBody">
      <!-- Social login -->
      <div class="social_login">
        <div class="">
          <a href="#" class="social_box fb">
            <span class="icon"><i class="fa fa-facebook"></i></span>
            <span class="icon_title">Connect with Facebook</span>
          </a>
          <a href="#" class="social_box google">
            <span class="icon"><i class="fa fa-google-plus"></i></span>
            <span class="icon_title">Connect with Google</span>
          </a>
        </div>
        <div class="centeredText">
          <span>Or use your Email address</span>
        </div>
        <div class="action_btms">
          <div class="one_half"><a href="#" id="login_form" class="btn">Login</a></div>
          <div class="one_half last"><a href="#" id="register_form" class="btn">Sign up</a></div>
        </div>
      </div>
    </div>
  </div>

```

Figura 10. El cuerpo de la página Web de los Egresados
Fuente: Elaboración Propia

En el index.html que creamos en el proyecto del Programa de Seguimiento de los Egresados de la Fundación Universitaria Los Libertadores se incluyó todo el texto, imágenes, login, conexiones a otras páginas, etc.

El index.html se inicia con la cabecera que se define todos datos para que el navegador necesita para interpretar correctamente el documento los cuales fueron los JQuery, CSS, etc. También realiza el proceso de intercambio de fotos en el Slider como también muestra un pop up (Emergente) donde el usuario ya sea el administrador, estudiante o la empresa digita las credenciales para el inicio de sesión de su perfil.



*Figura 11. El Index.html de la página Web de los Egresados
Fuente: Elaboración Propia*

EXPLICACION DEL CODIGO EN LOS CSS

El proyecto del Programa de Seguimiento de los egresados de la Fundación Universitaria Los Libertadores se crearon archivos .css en el cual se encuentran dentro de la carpeta css del proyecto.

CSS: Es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas.

Separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados "documentos semánticos"). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes.

Al crear una página web, se utiliza en primer lugar el lenguaje HTML/XHTML para marcar los contenidos, es decir, para designar la función de cada elemento dentro de la página: párrafo, titular, texto destacado, tabla, lista de elementos, etc.

Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, etc.

Los .css funcionan a base de reglas, es decir, declaraciones sobre el estilo de uno o más elementos. Las hojas de estilo están compuestas por una o más de esas reglas aplicadas a un documento HTML o XML. La regla tiene dos partes: un **selector** y la **declaración**.

h1 {color: red;}

h1 es el selector

{Color: red;} es la declaración

- **Declaración:** La declaración está compuesta por una propiedad y el valor que se le asigne.
- **Selector:** Funciona como enlace entre el documento y el estilo, especificando los elementos que se van a ver afectados por esa declaración. La declaración es la parte de la regla que establece cuál será el efecto.

En el ejemplo anterior, el selector h1 indica que todos los elementos h1 se verán afectados por la declaración donde se establece que la propiedad color va a tener el valor red (rojo) para todos los elementos h1 del documento o documentos que estén vinculados a esa hoja de estilos.

HISTORIA DE LOS CSS

Las hojas de estilos aparecieron poco después que el lenguaje de etiquetas SGML, alrededor del año 1970. Desde la creación de SGML, se observó la necesidad de definir un mecanismo que permitiera aplicar de forma consistente diferentes estilos a los documentos electrónicos.

El gran impulso de los lenguajes de hojas de estilos se produjo con el boom de Internet y el crecimiento exponencial del lenguaje HTML para la creación de documentos electrónicos. La guerra de navegadores y la falta de un estándar para la definición de los estilos dificultaban la creación de documentos con la misma apariencia en diferentes navegadores.

El organismo W3C (World Wide Web Consortium), encargado de crear todos los estándares relacionados con la web, propuso la creación de un lenguaje de hojas de estilos específico para el lenguaje HTML y se presentaron nueve propuestas. Las dos propuestas que se tuvieron en cuenta fueron la CHSS (Cascading HTML Style Sheets) y la SSP (Stream-based Style Sheet Proposal).

La propuesta CHSS fue realizada por Håkon Wium Lie y SSP fue propuesto por Bert Bos. Entre finales de 1994 y 1995 Lie y Bos se unieron para definir un nuevo lenguaje que tomaba lo mejor de cada propuesta y lo llamaron CSS (Cascading Style Sheets).

En 1995, el W3C decidió apostar por el desarrollo y estandarización de CSS y lo añadió a su grupo de trabajo de HTML. A finales de 1996, el W3C publicó la primera recomendación oficial, conocida como "CSS nivel 1".

A principios de 1997, el W3C decide separar los trabajos del grupo de HTML en tres secciones: el grupo de trabajo de HTML, el grupo de trabajo de DOM y el grupo de trabajo de CSS.

El 12 de mayo de 1998, el grupo de trabajo de CSS publica su segunda recomendación oficial, conocida como "CSS nivel 2". La versión de CSS que utilizan todos los navegadores de hoy en día es CSS 2.1, una revisión de CSS 2 que aún se está elaborando (la última actualización es del 8 de septiembre de 2009). Al mismo tiempo, la siguiente recomendación de CSS, conocida como "CSS nivel 3", continúa en desarrollo desde 1998 y hasta el momento sólo se han publicado borradores.

La adopción de CSS por parte de los navegadores ha requerido un largo periodo de tiempo. El mismo año que se publicó CSS 1, Microsoft lanzaba su navegador Internet Explorer 3.0, que disponía de un soporte bastante reducido de CSS. El primer navegador con soporte completo de CSS 1 fue la versión para Mac de Internet Explorer 5, que se publicó en el año 2000. Por el momento, ningún navegador tiene soporte completo de CSS 2.1.

<Link>: Aplica estilos a los enlaces que apuntan a páginas o recursos que aún no han sido visitados por el usuario.

- **Visited**: Aplica estilos a los enlaces que apuntan a recursos que han sido visitados anteriormente por el usuario. El historial de enlaces visitados se borra automáticamente cada cierto tiempo y el usuario también puede borrarlo manualmente.
- **Hover**: Aplica estilos al enlace sobre el que el usuario ha posicionado el puntero del ratón.
- **Active**: Aplica estilos al enlace que está pinchando el usuario. Los estilos sólo se aplican desde que el usuario pincha el botón del ratón hasta que lo suelta, por lo que suelen ser unas pocas décimas de segundo.
- **pseudo-clases**: Las pseudo-clases siempre incluyen dos puntos (:) por delante de su nombre y siempre se añaden a continuación del elemento al que se aplican, sin dejar ningún espacio en blanco por delante. Las pseudo-clases también se pueden combinar con cualquier otro selector complejo.

<Margin>: Establece la anchura de algunos o todos los márgenes de los elementos. Uno, dos, tres o cuatro de los siguientes valores medida, porcentaje, auto e inherit.

<Padding>: Establece la anchura de algunas o todas las zonas de relleno de los elementos. Uno, dos, tres o cuatro de los siguientes valores medida, porcentaje, inherit.

<Color>: Son formas de representar los colores que vemos cada día en un formato que un ordenador puede interpretar y mostrar.

<Text-Decoration>: Establece la decoración que se añade al texto de un elemento. Uno o más de los siguientes valores y en cualquier orden underline, overline, line-through, blink.

<Font-Size>: Establece el tamaño de letra. Uno y sólo uno de los siguientes valores absolute-size, relative-size, medida, porcentaje, inherit.

<Line-Height>: Establece el interlineado del texto. Uno y sólo uno de los siguientes valores normal, número, medida, porcentaje, inherit.

<Font-Family>: Establece la fuente o tipo de letra. Uno y sólo uno de los siguientes valores nombre de la fuente o tipo de letra o nombre genérico de la fuente o tipo de letra.

<Background>: Establece algunas o todas las propiedades del color y/o imagen de fondo de un elemento. Alguno o todos los siguientes valores y en cualquier orden background-color, background-image, background-repeat, background-attachment, background-positioninherit.

<Width>: Establece la anchura de los elementos de bloque. Uno y sólo uno de los siguientes valores medida, porcentaje, auto, inherit.

```

|* {
  margin: 0;
  padding: 0;
}
a {
  color: #000000;
  text-decoration: underline;
}
a:hover {
  text-decoration: none;
}
body {
  font-size: 12pt;
  line-height: 1.75em;
  font-family: Arial;
  background: #02E600;
  color: #382b31;
}
#header {
  position: relative;
  background: #34564f;
  width: 100%;
  min-height: 80px;
}
.sesionInit {
  position: absolute;
  top: -60px;
  right: 17.5%;
  line-height: 150px;
  list-style: none;
  font-size: 10pt;
  text-decoration: none;
  color: #fff;
}

```

*Figura 12. El CSS de la página Web de los Egresados
Fuente: Elaboración Propia*

<Min-Height>: Establece la altura mínima de los elementos de bloque. Uno y sólo uno de los siguientes valores medida, porcentaje, inherit.

<Position>: Posiciona los elementos de forma normal, relativa, absoluta o fija. Alguno o todos los siguientes valores y en cualquier orden static, relative, absolute, fixed, inherit.

<Top>: Indica el desplazamiento entre el borde superior del elemento y la parte superior de su elemento contenedor. Alguno o todos los siguientes valores y en cualquier orden medida, porcentaje, auto, inherit.

<Right>: Indica el desplazamiento entre el borde derecho del elemento y la parte derecha de su elemento contenedor. Alguno o todos los siguientes valores y en cualquier orden medida, porcentaje, auto, inherit.

<List-Style>: Establece algunas o todas las propiedades del marcador de los elementos de la lista. Alguno o todos los siguientes valores y en cualquier orden list-style-type, list-style-position, list-style-image, inherit.

<Max-Height>: Limita la altura máxima de los elementos de bloque. Uno y sólo uno de los siguientes valores medida, porcentaje, none, inherit.

<Max-Width>: Limita la anchura máxima de los elementos de bloque. Uno y sólo uno de los siguientes valores medida, porcentaje, none, inherit.

<Min-Height>: Establece la altura mínima de los elementos de bloque. Uno y sólo uno de los siguientes valores medida, porcentaje, inherit.

<Min-Width>: Establece la anchura mínima de los elementos de bloque. Uno y sólo uno de los siguientes valores medida, porcentaje, inherit.

```

.imageLogo {
  max-width: 40%;
  min-height: 80px;
  min-width: 350px;
}

.imageTab {
  max-width: 50%;
  margin: 0% 0% 0% 27%;
  box-shadow: 3px 3px 7px #777;
}

.imageCurriculum{
  max-width: 20%;
  margin: 0% 0% 0% 40%;
  box-shadow: 3px 3px 7px #777;
}

br.clear {
  clear: both;
}

strong {
  color: #000000;
}

p {
  margin-bottom: 1.5em;
}

ul {
  margin-bottom: 1.5em;
}

.imageList {
  list-style: none;
}

.imageList li {
  clear: both;
}

```

*Figura 13. El CSS de la página Web de los Egresados
Fuente: Elaboración Propia*

<Box-Shadow>: Establece la sombra y el efecto en 3d.

<Clear>: Controla el comportamiento de los elementos adyacentes a elementos posicionados de forma flotante. Alguno o todos los siguientes valores y en cualquier orden none, left, right, both, inherit.

<Margin-Bottom>: Establece la anchura del margen inferior de los elementos de bloque. Uno y sólo uno de los siguientes valores medida, porcentaje, auto, inherit.

<Overflow>: Controla el comportamiento de los contenidos que no caben en su elemento contenedor. Uno y sólo uno de los siguientes valores visible, hidden, scroll, auto, inherit.

<Display>: Establece el tipo de caja generada por un elemento. Uno y sólo uno de los siguientes valores inline, block, list-item, run-in, inline-block, table, inline-table, table-row-group, table-header-group, table-footer-group, table-row, table-column-group, table-column, table-cell, table-caption, none, inherit.

```
.ca-content{
  position: absolute;
  left: 70px;
  width: 370px;
  height: 60px;
  top: 5px;
}
.ca-main{
  -webkit-transition: all 300ms linear;
  -moz-transition: all 300ms linear;
  -o-transition: all 300ms linear;
  -ms-transition: all 300ms linear;
  transition: all 300ms linear;
}
.ca-sub{
  color: #666;
}
.ca-menu li:hover{
  background-color: #6B8E23;
}
.ca-menu li:hover .ca-icon{
  color: #34564f;
  font-size: 120px;
  opacity: 0.2;
  left: -20px;
  -webkit-transform: rotate(20deg);
  -moz-transform: rotate(20deg);
  -ms-transform: rotate(20deg);
  transform: rotate(20deg);
}
.ca-menu li:hover .ca-main{
  color: #f900b0;
  opacity: 0.8;
}
.ca-menu li:hover .ca-sub{
  color: #fff;
  opacity: 0.8;
}

#copyright {
  text-align: center;
  color: #819895;
}
```

*Figura 14. El CSS de la página Web de los Egresados
Fuente: Elaboración Propia*

EXPLICACION DEL CODIGO EN LOS JAVASCRIPT

JavaScript es un lenguaje de programación, al igual que PHP, se utiliza principalmente del lado del cliente (es decir, se ejecuta en nuestro ordenador, no en el servidor) permitiendo crear efectos atractivos y dinámicos en las páginas web.

La ventaja de JavaScript es que al estar alojado en el ordenador del usuario los efectos son muy rápidos y dinámicos. Al ser un lenguaje de programación permite toda la potencia de la programación como uso de variables, condicionales, bucles, etc. Algunos inconveniente por ejemplo si el usuario tiene desactivado JavaScript en su navegador, no se mostrarán los efectos. No obstante, hoy día la mayoría de los usuarios navegan por la web con JavaScript activado.

Para usar JavaScript es definir JavaScript en un archivo externo. Los archivos de JavaScript son archivos de texto guardados con la extensión .js.

El código JavaScript debe ir encerrado entre las siguientes marcas:

```
<script type="text/javascript">
```

```
</script>
```

Los documentos HTML permiten incrustar fragmentos de código JavaScript, bien dentro del propio archivo HTML o bien realizando una carga de ese código indicando el archivo donde se encuentra el código JavaScript. Dentro de un documento HTML puede haber ninguno,

uno o varios scripts de JavaScript. El uso de JavaScript es muy habitual en la programación web.

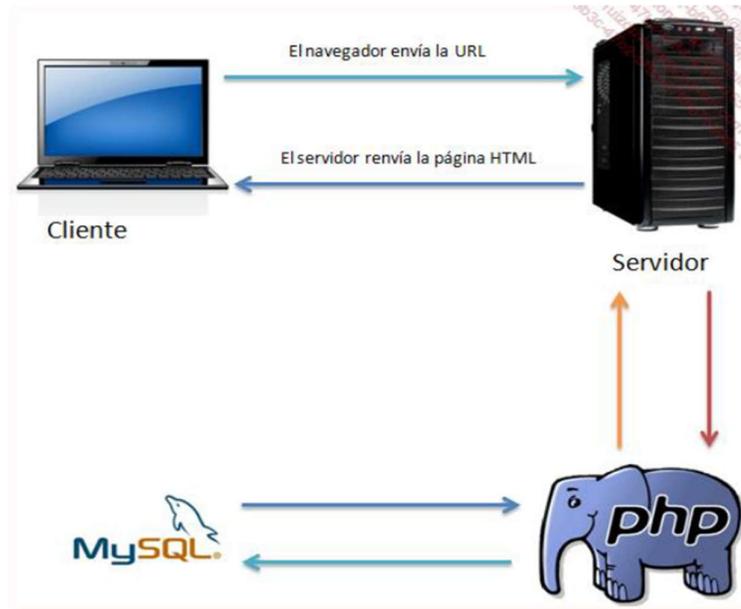


Figura 15. Modelamiento interno de la página web

Fuente: [http://4.bp.blogspot.com/-](http://4.bp.blogspot.com/-8ilzYCS0kwc/U778ZSMumrI/AAAAAAAAARQk/iTN17Y_poYY/s1600/Arquitectura-Cliente-Servidor-PHP-MySQL.PNG)

[8ilzYCS0kwc/U778ZSMumrI/AAAAAAAAARQk/iTN17Y_poYY/s1600/Arquitectura-Cliente-Servidor-PHP-MySQL.PNG](http://4.bp.blogspot.com/-8ilzYCS0kwc/U778ZSMumrI/AAAAAAAAARQk/iTN17Y_poYY/s1600/Arquitectura-Cliente-Servidor-PHP-MySQL.PNG)

CAPITULO VIII

MODELAMIENTO DE LAS ARQUITECTURAS UTILIZADAS

ARQUITECTURA CLIENTE/SERVIDOR

La arquitectura cliente/servidor es un modelo utilizado para construir Sistemas de Informacion, que basa su importancia en distribuir todos los datos y la informacion que requiere manejar, por todo el Sistemas de ordenadores por medio de una red para descentralizar el procesamiento, permitiendo mejorar el rendimiento del Sistema de Informacion.

Se denomina cliente al proceso que inicia el dialogo o solicita los recursos y servidor al proceso que responde a las solicitudes.

El concepto de cliente/servidor proporciona una forma eficiente de utilizar todos los recursos de máquina, de tal forma que la seguridad y fiabilidad que proporcionan los entornos mainframe se traspasa a la red de area local. A esto hay que añadir la ventaja de la portencia simplicidad de los ordenadores personales.

En este modelo las aplicaciones se deciden de forma que el servidor contiene la parte que debe ser compartida por varios usuarios, y en el cliente permanece sólo lo particular de cada usuario.

El servidor realiza el servicio que fue solicitado por el cliente y devuelve los resultados, aunque en algunos casos existen procesos auxiliares que reciben las solicitudes del cliente y verifican la activación del servidor para recibir la solicitud, satisfacer el cliente y enviar el resultado.

Por otro lado, los clientes realizan generalmente funciones como:

- Manejo de la interfaz de usuario.
- Captura y validación de los datos de entrada.
- Generación de consultas e informes sobre las bases de datos.

Por su parte los servidores realizan, entre otras, las siguientes funciones:

- Gestión de periféricos compartidos.
- Control de accesos concurrentes a bases de datos compartidas.
- Enlaces de comunicaciones con otras redes de área local o extensa.

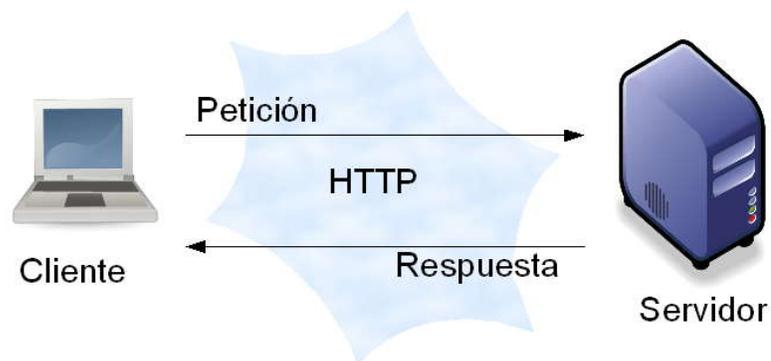


Figura 16. Arquitectura Cliente/Servidor

Fuente: <https://encrypted->

[tbn2.gstatic.com/images?q=tbn:ANd9GcRt7xnQDtFaVEsfwL1K9WvZB3sDQWyXTULS-kkZmzEdbsPsJxRT](https://encrypted-tbn2.gstatic.com/images?q=tbn:ANd9GcRt7xnQDtFaVEsfwL1K9WvZB3sDQWyXTULS-kkZmzEdbsPsJxRT)

CARACTERISTICAS DE LA ARQUITECTURA CLIENTE/SERVIDOR

- El cliente y el servidor pueden actuar como una sola entidad y también pueden actuar como entidades separadas, realizando actividades o tareas independientes.
- Las funciones de Cliente y Servidor pueden estar en plataformas separadas, o en la misma plataforma.
- Un Servidor da servicio a múltiples clientes en forma concurrente.
- Cada plataforma puede ser escalable independientemente. Los cambios realizados en las plataformas de los Clientes o de los Servidores, ya sean por actualización o por reemplazo tecnológico, se realizan de una manera transparente para el usuario final.
- Un sistema de Servidores realiza múltiples funciones al mismo tiempo que presenta una imagen de un solo sistema a las estaciones Clientes. Esto se logra combinando los recursos de cómputo que se encuentran físicamente separados en un solo sistema lógico, proporcionando de esta manera el servicio más efectivo para el usuario final. También es importante hacer notar que las funciones Cliente /Servidor pueden ser dinámicas. Ejemplo, un Servidor puede convertirse en Cliente cuando realiza la solicitud de servicios a otras plataformas dentro de la red.
- Además se constituye como el nexo de unión más adecuado para reconciliar los sistemas de información basados en mainframes o minicomputadores, con aquellos otros sustentados en entornos informáticos pequeños y estaciones de trabajo.

VENTAJAS DEL ESQUEMA CLIENTE/SERVIDOR

Entre las principales ventajas del esquema Cliente/Servidor están:

- Unos de los aspectos que más ha promovido el uso de sistemas Cliente/Servidor, es la existencia de plataformas de hardware cada vez más baratas. Esta constituye a su vez una de las más palpables ventajas de este esquema, la posibilidad de utilizar máquinas considerablemente más baratas que las requeridas por una solución centralizada, basada en sistemas grandes. Además, se pueden utilizar componentes, tanto de hardware como de software, de varios fabricantes, lo cual contribuye considerablemente a la reducción de costos y favorece la flexibilidad en la implantación y actualización de soluciones.
- El esquema Cliente/Servidor facilita la integración entre sistemas diferentes y comparte información permitiendo, por ejemplo que las máquinas ya existentes puedan ser utilizadas pero utilizando interfaces más amigables al usuario. De esta manera, podemos integrar PCs con sistemas medianos y grandes, sin necesidad de que todos tengan que utilizar el mismo sistema operacional.
- Al favorecer el uso de interfaces gráficas interactivas, los sistemas contruidos bajo este esquema tienen mayor interacción y más intuitiva con el usuario. En el uso de interfaces gráficas para el usuario, el esquema Cliente/Servidor presenta la ventaja, con respecto a uno centralizado, de que no es siempre necesario transmitir información gráfica por la red pues esta puede residir en el Cliente, lo cual permite aprovechar mejor el ancho de banda de la red.

- Una ventaja adicional de uso del esquema Cliente/Servidor es que es más rápido el Mantenimiento y el desarrollo de aplicaciones, pues se pueden emplear las herramientas existentes.
- La estructura inherentemente modular facilita además la integración de nuevas tecnologías y el crecimiento de la infraestructura computacional, favoreciendo así la escalabilidad de las soluciones.
- El esquema Cliente/Servidor contribuye además, a proporcionar, a los diferentes departamentos de una organización, soluciones locales, pero permitiendo la integración de la información relevante a nivel global.

DESVENTAJAS DEL ESQUEMA CLIENTE/SERVIDOR

Entre las principales desventajas del esquemas Cliente/Servidor están:

- El mantenimiento de los sistemas es más difícil pues implica la interacción de diferentes partes de hardware y de software, distribuidas por distintos proveedores, lo cual dificulta el diagnóstico de fallas.
- Se cuenta con muy escasas herramientas para la administración y ajuste del desempeño de los sistemas.
- Además, hay que tener estrategias para el manejo de errores y para mantener la consistencia de los datos.

- La seguridad de un esquema Cliente/Servidor es otra preocupación importante. Por ejemplo, se deben hacer verificaciones en el cliente y en el servidor.
- El desempeño es otro de los aspectos que se deben tener en cuenta en el esquema Cliente/Servidor. Problemas de este estilo pueden presentarse por congestión en la red, dificultad de tráfico de datos, etc.

TIPOS DE ARQUITECTURA CLIENTE/SERVIDOR

ARQUITECTURA CLIENTE/SERVIDOR DOS CAPAS

Consiste en una capa de presentación y lógica de la aplicación; y la otra de la base de datos.

Normalmente esta arquitectura se utiliza en las siguientes situaciones:

- Cuando se requiera poco procesamiento de datos en la organización.
- Cuando se tiene una base de datos centralizada en un solo Servidor.
- Cuando la base de datos es relativamente estática.
- Cuando se requiere un mantenimiento mínimo.

El sistemas se separa de en dos partes fijas: el Cliente y el Servidor, la lógica de las aplicaciones debe estar en el Cliente o en el Servidor y la comunicación con el Servidor es transparente para el usuario.

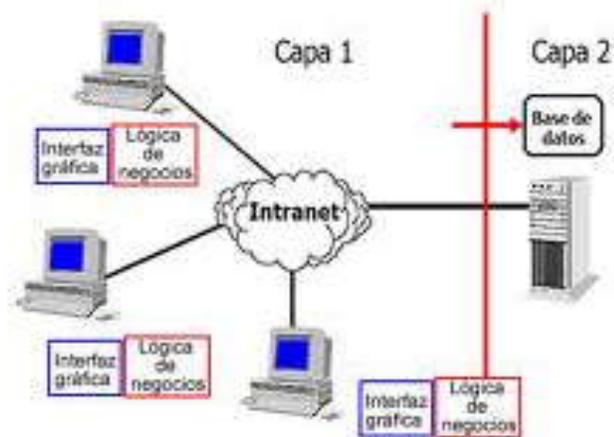


Figura 17. Arquitectura Cliente/Servidor Dos Capas

Fuente: https://laurmolina7821.files.wordpress.com/2013/03/dos_capas.jpg?w=300&h=202

ARQUITECTURA CLIENTE/SERVIDOR DE TRES CAPAS

Consiste en una capa de la Presentación, otra capa de la lógica de la aplicación y otra capa de la base de datos.

- Normalmente esta arquitectura se utiliza en las siguientes situaciones:
- Cuando se requiera mucho procesamiento de datos en la aplicación.
- En aplicaciones donde la funcionalidad esté en constante cambio.
- Cuando los procesos no están relativamente muy relacionados con los datos
- Cuando se requiera aislar la tecnología de la base de datos para que sea fácil de cambiar.
- Cuando se requiera separar el código del cliente para que se facilite el mantenimiento.
- Esta muy adecuada para utilizarla con la tecnología orientada a objetos.

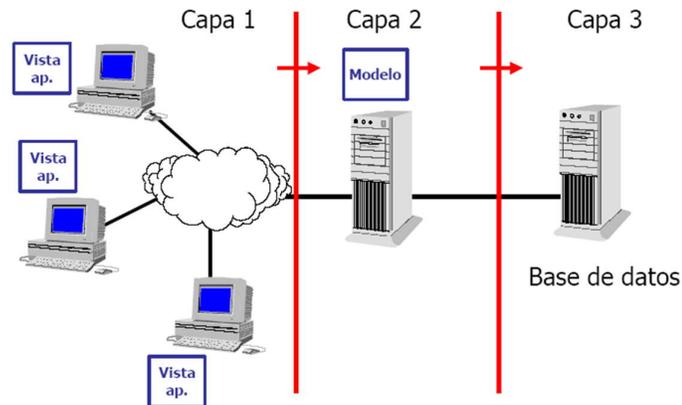


Figura 18. Arquitectura Cliente/Servidor Tres Capas

Fuente: <http://eltamiz.com/elcedazo/wp-content/uploads/2010/06/3capas-300x180.gif>

CAPA DE PRESENTACION

Esta capa es la interfaz con el usuario y consiste de hardware como un Pc o una estación de trabajo y un navegador de red.

CAPA DE LOGICA DE FUNCIONALIDAD

Esta capa proporciona la funcionalidad al extremo de los usuarios y contiene la lógica del negocio (aplicación). Proporciona el puente entre la primera y la tercera capa.

CAPA DE DATOS

Esta capa incluye la Base de Datos que contiene los datos de la organización y esta se encapsula desde los usuarios finales.

CAPITULO IX

BIBLIOGRAFIA

- Gilfillan, I, (2003). *La biblia de MySQL*. Madrid, España. Editorial Anaya Multimedia.
- <https://developer.mozilla.org/es/>
- http://www.aulaclie.es/html/a_2_1_1.htm
- Gomez, M, (2013). *Notas del curso Bases de datos*. Mexico DF, México. Universidad Autónoma Metropolitana.
- <http://aprenderaprogramar.com/>
- <http://librosweb.es/referencia/css/>
- Sánchez, C. (2006). *Fundamentos de diseño de bases de datos, quinta edición*. Editorial McGraw-Hill.
- Vásquez, C. (2008). *Programación en PHP5 Nivel Básico*. Recuperado de http://www.ticarte.com/sites/su/users/7/arch/manual_php5_basico.pdf
- <http://www.aprendoencasa.com>
- <http://www.htmlpoint.com>
- The php group. (2001-2017) Recuperado de <http://docs.php.net/manual/es/intro-what-is.php>
- Páez, T y Gómez, P.(2008). *Sistemas I*[Entrada en blog].Recuperado de <http://sistemaspyt.blogspot.com.co/2008/09/principales-caracteristicas.html>
- http://www.ibm.com/support/knowledgecenter/es/SSWSR9_11.0.0/com.ibm.pim.dev.doc/pim_tsk_arc_definingusecases.html
- <https://msdn.microsoft.com/es-es/library/bb972214.aspx>