



Universidad Autónoma del Estado de México

Centro Universitario UAEM Valle de Chalco

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA PARA EL CONTROL DE HORAS DEL
PERSONAL QUE REALIZA SERVICIO SOCIAL O PRÁCTICAS PROFESIONALES
USANDO SPRING MVC Y BOOTSTRAP**

T E S I S

QUE PARA OBTENER EL TÍTULO DE

INGENIERO EN COMPUTACIÓN

P R E S E N T A

ERNESTO RENDÓN MORALES

ASESOR:

MTRO. EN C. C. RICARDO JAVIER BUCIO LÓPEZ

VALLE DE CHALCO SOLIDARIDAD, MÉXICO

ENERO 2023.



CUVCH

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA PARA
EL CONTROL DE HORAS DEL PERSONAL QUE
REALIZA SERVICIO SOCIAL O PRÁCTICAS
PROFESIONALES USANDO SPRING MVC Y
BOOTSTRAP**

ÍNDICE

I. Resumen	14
II. Antecedentes de la temática.....	15
III. Importancia del problema	18
IV. Planteamiento del problema	19
V. Marco conceptual o teórico.....	23
5.1. Requerimientos mínimos para ejecución de aplicaciones java web	28
5.2. Base de datos	29
5.3. Terminología sobre bases de datos	30
5.4. SQL.....	31
5.5. MariaDB	31
5.6. Conexión a una base de datos.....	32
5.7. Pruebas.....	33
5.8. Comunicación de proyectos.....	34
5.9. Especificación de Requerimientos de software	35
5.10. Captura de requerimientos	36
5.11. Análisis de los requerimientos.....	37
5.12. Especificación de los requerimientos.....	38
5.13. Diseño del sistema	38
5.14. Desarrollo de alcance del sistema.....	39
5.15. Ciclo de vida del desarrollo de software	40
5.16. Modelo-Vista-Controlador	44
5.17. Beneficios de usar MVC.....	45

5.18. UML (Lenguaje Unificado de Modelo).....	46
5.18.1. UML es un lenguaje.....	46
5.19. Desarrollo de diagramas de casos de uso	47
5.20. Diagrama de clases	48
5.20.1. Las clases	49
5.20.2. Métodos	50
5.20.3. Encapsulación	50
5.20.3. El modelo relacional	51
5.20.4. Modelo Entidad-Relación.....	52
5.20.5. Elementos del modelo E-R.....	53
5.21. Conceptos básicos del servidor web.....	56
5.21.1. Servlets (Módulos java).....	57
5.21.2. Páginas de servidor ASP (Active Server Pages) y JSP (Java Server Pages)	58
5.22. Desarrollo web	59
5.22.1. Aplicaciones web.....	60
5.22.2. HTML	61
5.22.3. CSS	61
5.22.4. Lenguaje java	62
5.23. Frameworks de programación.....	63
5.23.1 Spring.....	64
5.23.2 Componentes de una aplicación Spring.....	64
5.23.3 Características de Spring.....	66
5.23.4. Spring MVC.....	66
5.24. Diseño Arquitectónico.....	67

5.25. Bootstrap.....	68
5.25.1. Como funciona Bootstrap	70
VI. Métodos y técnicas de investigación.....	72
VII. Presentación y discusión de resultados	74
7.1. Herramientas y tecnologías por usar.....	75
7.2. Desarrollo del sistema	75
7.3. Como obtener MariaDB.....	76
7.4. Listar	81
7.5. Agregar un registro.....	85
7.6. Modificar un registro.....	89
7.7. Eliminar un registro.....	90
7.8. Diagrama de casos de uso	92
7.9. Diseño del sistema	93
7.10. Arquitectura del sistema.....	93
7.11. Modelo de base de datos.....	94
7.12. Tabla alumno	96
7.13. Vista de la página principal	98
7.14. Vistas.....	99
7.15. Manual del sistema	101
7.15.1 Agregar usuario	102
7.15.2 Editar un registro	103
7.15.3 Eliminar un registro	105
VIII. Conclusiones y sugerencias.....	107
8.1. Conclusiones.....	107
8.2. Aportaciones.....	108

8.3. Sugerencias.....	109
8.4. Trabajo a futuro.....	109
IX. Referencias de consulta.....	111

ÍNDICE DE ILUSTRACIONES

Figura 1: Clasificación de modelos concretos en clases abstractas (Pressman, 2010).	24
Figura 2: Modelo general de un sistema (Fernández, 2006).	25
Figura 3: Sistema automatizado de venta de bebidas (Rendón, 2022).	26
Figura 4: Características de un objeto (Rendón, 2022).	27
Figura 5: Logo oficial (Oracle, 2022).	28
Figura 6: Muestra los atributos de la tabla Stock_code (Gilfillan, 2003).	29
Figura 7: Muestra los atributos de la tabla Invoice_code (Gilfillan, 2003).	29
Figura 8: Simbología en diagramas Base de datos (Rendón, 2022).	31
Figura 9: Conexión entre Servidor y Cliente (Rendón, 2022).	32
Figura 10: Conexión entre Servidor y Cliente (Rendón, 2022).	32
Figura 11: El sistema se actualiza constantemente durante su uso (Rendón, 2022).	33
Figura 12: Representación de comunicación para el desarrollo (Rendón, 2022).	34
Figura 13: Desarrolladores en la fase de recolección de información (Rendón, 2022).	37
Figura 14: Visión general de un sistema en cascada (Rendón, 2022).	40
Figura 15: Ciclo de vida de desarrollo de software (Booch et, al 2006).	42
Figura 16: Un sistema puede sufrir cambios hasta lograr una versión estable (Rendón, 2022).	43
Figura 17: Relación entre los módulos del patrón MVC (Bascón, 2004).	45
Figura 18: Diagrama de casos de uso que representa al sistema utilizado para planear una conferencia (Kendall, & Kendall. 2011).	48
Figura 19: Diagrama de clases para los ofrecimientos de cursos (Kendall, & Kendall. 2011).	49
Figura 20: Estructura general de una clase (Fuente propia, 2022).	50
Figura 21: Representación de una relación en forma de tabla (Ramos & Ramos, 2007).	52
Figura 22: Diagrama E-R un proveedor suministra varios artículos (Ramos & Ramos, 2007).	53

Figura 23: Representación de la entidad ALUMNOS y sus atributos (Ramos & Ramos, 2007).	54
Figura 24: Representación de una relación (Ramos & Ramos, 2007).	55
Figura 25: Simbología del modelo E-R (Rendon, 2022).	56
Figura 26: Applet se ejecuta desde un servidor (Rendón, 2022).	58
Figura 27: Funcionamiento de las páginas de servidor, Una página ASP/JSP (Berzal et al., 2005).	58
Figura 28: Representación de librerías antiguas que ofrecía HTML con solo texto (Rendón, 2022).	59
Figura 29: El contenido que se le muestra al usuario se genera dinámicamente para cada solicitud proveniente del navegador web instalado en la máquina cliente (Berzal et al., 2005).	60
Figura 30: Representación mediante cuerpos de HTML y CSS (Rendón, 2022).	61
Figura 31: Java puede usarse con las diferentes herramientas de estilos (Rendón, 2022).	62
Figura 32: Estructura física de un Framework (Rendon, 2022).	67
Figura 33: Bootstrap se caracteriza por ofrecer estilos visuales en equipos móviles y escritorio (Rendón, 2022).	69
Figura 34: Bootstrap es una herramienta fácil de aprender y ofrecer estilos a páginas web (Rendón, 2022).	70
Figura 35: Modelo del ciclo de vida en Cascada (Maida & Pacienza, 2015).	73
Figura 36: Consola inicial de MariaDB (Rendón, 2022).	76
Figura 37: Interfaz inicial de Netbeans (Rendón, 2022).	77
Figura 38: Selección de librería MySQL JDBC Driver (Rendón, 2022).	78
Figura 39: Creación del paquete llamado Index.htm (Rendón, 2022).	80
Figura 40: Página de inicio de estilos de Bootstrap (Rendón, 2022).	82
Figura 41: Ubicación de enlace para que los estilos tomen efecto en el proyecto (Rendón, 2022).	82
Figura 42: Interfaz principal del sistema (Rendón, 2022).	84
Figura 43: Diagrama de casos de usos del sistema (Rendón, 2022).	92
Figura 44: Modelo vista controlador del sistema (Rendón, 2022).	93

Figura 45: Arquitectura del sistema (Rendón, 2022).....	94
Figura 46: Modelo de base de datos (Rendón, 2022).....	95
Figura 47: Tabla alumno con atributos (Rendón, 2022).....	96
Figura 48: Tabla entrada con atributos (Rendón, 2022).	97
Figura 49: Tabla salida con atributos (Rendón, 2022).	97
Figura 50: Interfaz principal (Rendón, 2022).	98
Figura 51: Interfaz para agregar registro (Rendón, 2022).....	99
Figura 52: Interfaz para actualizar registro (Rendón, 2022).....	100
Figura 53: Interfaz principal con personal registrado (Rendón, 2022).....	101
Figura 54: Botón agregar registros (Rendón, 2022).	102
Figura 55: Interfaz agregar registros (Rendón, 2022).	102
Figura 56: Campos llenados correctamente (Rendón, 2022).	103
Figura 57: Botón Editar (Rendón, 2022).....	103
Figura 58: Interfaz actualizar registros (Rendón, 2022).....	104
Figura 59: Eliminar registros (Rendón, 2022).....	105
Figura 60: Botón para registrar entrada (Rendón, 2022).....	105
Figura 61: Botón para registrar salida (Rendón, 2022).....	106
Figura 62: Ejemplo de sistemas que puedan implementar un reloj checador (Rendón, 2022).	110

ÍNDICE DE TABLAS

Tabla 1: Clasificación de Frameworks (Ortega, 2017).	16
Tabla 2: Clasificación de Frameworks (Ortega, 2017).	63

ÍNDICE DE GRAFICAS

Grafica 1: Se muestra que la institución no contaba un sistema propio para el conteo de horas (Rendón, 2022)	19
Grafica 2: Se muestra que la institución llevaba el control de los datos (Rendón, 2022)	20
Grafica 3: Se muestra que la institución proporcionaba un formato (Rendón, 2022)	20
Grafica 4: Se muestra que los alumnos les gustaría un sistema (Rendón, 2022)	21
Grafica 5: Los alumnos consideran como mejor opción un sistema (Rendón, 2022)	21

I. Resumen

En el presente trabajo de tesis se enfocará en diseñar un sistema de información para los alumnos y pasantes de diferentes instituciones, el cual ellos tienen como función cumplir una cierta cantidad de horas necesarias y así poder obtener el documento que acrediten las horas realizadas en las instituciones públicas y privadas.

La idea principal de implementar este sistema de información se llevó a cabo gracias a una necesidad, los alumnos y pasantes tienen que almacenar los datos de sus horas realizadas en una bitácora la cual no ofrece integridad de datos en su totalidad y los cálculos de las horas eran de manera manual, no se tenía una exactitud de los datos que se obtenían al final de cada día y por ende los alumnos no tenían un control de horas de servicio. Otras de las razones por la cual se va a desarrollar este sistema es para disminuir el uso de hojas excesivo por parte de las instituciones.

Para el desarrollo del sistema de información, se empleó el lenguaje de programación java de alto nivel haciendo uso de los FRAMEWORKS para aplicaciones web, hoy en día son herramientas que son de ayuda para las empresas, para el desarrollo de aplicaciones ya que son herramientas que ofrecen un desarrollo eficaz de aplicaciones web.

II. Antecedentes de la temática

La computadora es una herramienta capaz de llevar millones de cálculos a la vez, sin embargo, una persona con calculadora en mano tardaría mucho en llevar a cabo todas esas operaciones que realiza la computadora, además de que la persona podría equivocarse en ocasiones al realizar esas operaciones y como la computadora procesa datos bajo el control del conjunto de instrucciones llamados programas de cómputo, el marco de error que tiene la computadora sería bajo (Paul Deitel, H. D, 2008).

La programación en los años 50 era diferente a la de la actualidad ya que en aquellos años utilizaban switch eléctricos conectados a un computador de enorme tamaño, con el fin de configurar secuencias numéricas. Para los programadores de esa época no veían la programación como tal ya que no se contaban con buenos recursos de velocidad por partes de las maquinas, además otra problemática para ellos era escribir código pequeño y que fuera funcional, gracias a las empresas que se dedicaban al desarrollo de ordenadores, fueron innovando con equipos que cumplían con las necesidades de los programadores ya que tenían mejor rendimiento y dejó de ser un principal criterio la velocidad (Gacitúa, 2003).

En pocas palabras la programación fue evolucionando y se expandió ya que los ordenadores de hoy en día cumplen con las necesidades de los programadores para desarrollar software que sean de código en teoría pequeño pero eficiente.

Para el entender el desarrollo de sistemas computacionales se deriva el concepto de informática, este se encarga de estudiar la capacidad que tiene un programa, que pueden y que no pueden hacer, la complejidad de un algoritmo,

las estructuras de datos, tipos de datos, la comunicación entre sistemas y los seres humanos (Cedano, et al., 2014).

Tomando en cuenta los conceptos y un poco de historia para entender mejor el concepto de computación la siguiente información servirá para tener mejor enfoque sobre el proyecto de investigación que se desarrollará para el proyecto de tesis.

Según los autores (Cervantes & Gómez, 2012) existen diferentes tipos de modelos de desarrollo de software los cuales se mostrarán continuación en la siguiente figura:

<p>Tradicionales</p> <p>O</p> <p>Pesados</p>	En cascada	<p>Pura</p> <p>Con fases solapadas</p> <p>Con subproyectos</p> <p>Con reducción de riesgos</p>
	Evolutivos	<p>Espiral</p> <p>Entrega por etapas o incremental</p> <p>Entrega evolutiva o iterativo</p> <p>Diseño por planificación</p> <p>Cascada en V</p>
	Minimización de desarrollos	<p>Componentes reutilizables</p> <p>Diseño por herramientas</p>
	Híbridos	<p>Proceso unificado relacional</p> <p>Otros</p>
Metodologías Ágiles		<p>Programación extrema</p> <p>SCRUM</p> <p>Desarrollo dirigido por pruebas</p> <p>Desarrollo dirigido po características</p> <p>Agile, lean, crystal,....., etc.</p>

Tabla 1: Clasificación de modelos concretos en clases abstractas (Cervantes & Gómez, 2012).

En la gráfica se representan los modelos tradicionales o pesados y las metodologías ágiles las cuales se explicarán a continuación:

Modelo cascada: Se caracterizan por llevar una secuencia en serie de las siguientes actividades; análisis definición de requerimientos, diseño, codificación, validación y mantenimiento.

Evolutivos: Este tipo de modelado se caracterizan por revisar las diferentes etapas de desarrollo las veces que sean necesarias, pero con un orden específico.

Minimización de desarrollo: Se encuentran los siguientes componentes “reutilizables” y de “desarrollo por herramientas”, aquí se utilizan elementos desarrollados previamente con la finalidad de acelerar los procesos.

Híbridos: Propuesto por IMB, se basa en el uso de las mejores prácticas de proyectos exitosos. “RUP” en español proceso unificado racional, propone el uso de buenas prácticas para el desarrollo de diseños junto con el lenguaje unificado de modelo UML, con esto se crea una metodología estándar para el análisis, implementación y documentación de sistemas orientado a objetos.

Metodologías ágiles: En las metodologías ágiles, se encuentran programación extrema, SCRUM, desarrollo dirigido por pruebas de desarrollo, dirigido por características, agile, a cristal, etc. Todas ellas son que se caracterizan por un tiempo corto en su desarrollo y que producen buenos resultados a pesar del corto plazo.

Para el desarrollo de este proyecto también se utilizarán los Frameworks de programación que son los esquemas de reutilización de software conformado por componentes que contienen los siguientes elementos como son la abstracción de clases, objetos o componentes y las conexiones de base de datos como los controladores que van conectados a MySQL, SQL Server y Oracle (Ortega, Guevara & Benavides, 2016).

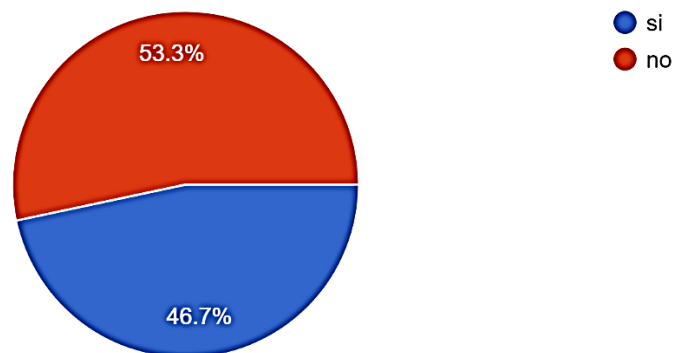
III. Importancia del problema

Recopilar información del departamento encargado del conteo de horas, por lo cual al momento de que los alumnos contabilizaran sus horas es un poco la demora al calcular dicha información con exactitud, por lo que se considera de suma importancia realizar un sistema de información automatizado que sirva como herramienta de apoyo tanto para la oficina encargada donde pueda almacenar de manera rápida los datos de las horas de los alumnos, así mismo presentar una interfaz de consulta con mejor presión del al visualizar las horas registradas en sistema.

IV. Planteamiento del problema

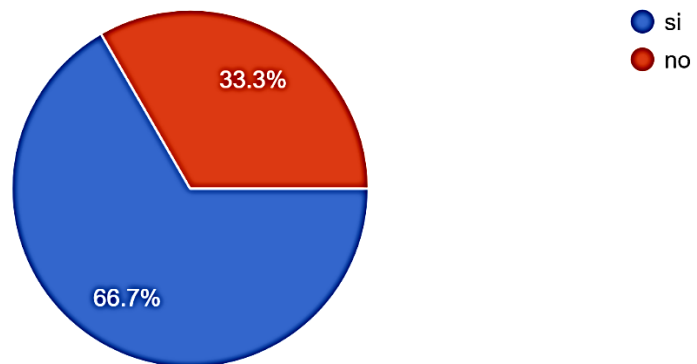
Debido a las necesidades de algunas instituciones de mantener información clara y concisa sobre las horas que llevan realizando los estudiantes en sus planteles ya que suelen haber inconformidades como al realizar cálculos erróneos para contabilizar horas por parte del alumnado suele haber un disgusto ya que la información que maneja el plantel suele ser distinta a la de los alumnos. Se utilizó la herramienta de Google formularios para aplicar 5 preguntas a 30 personas para saber si un sistema satisface sus necesidades al momento de llevar a cabo su registro de horas. A continuación, se muestran las preguntas aplicadas con una gráfica con los resultados obtenidos:

1-. ¿Cuándo usted realizó su servicio social o prácticas profesionales, la institución contaba con un sistema para llevar el control de horas?



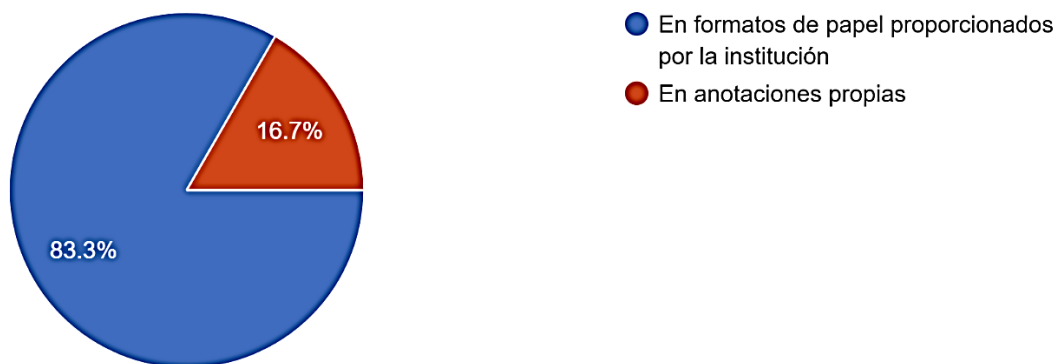
Grafica 1: Se muestra que la institución no contaba un sistema propio (Rendón, 2022).

2-. ¿La institución llevaba el control de horas del personal?



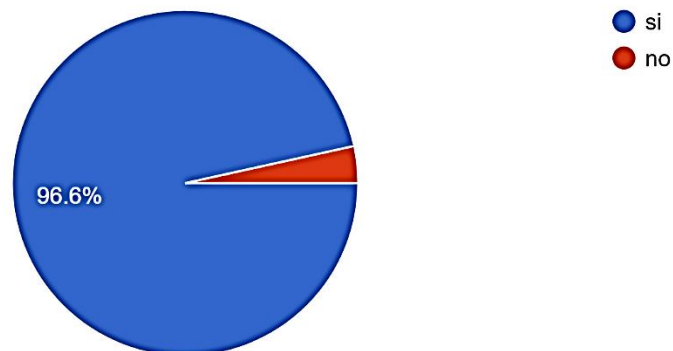
Grafica 2: Se muestra que la institución llevaba el control de los datos (Rendón, 2022).

3-. ¿Como era el registro en la institución en donde prestaba su servicio?



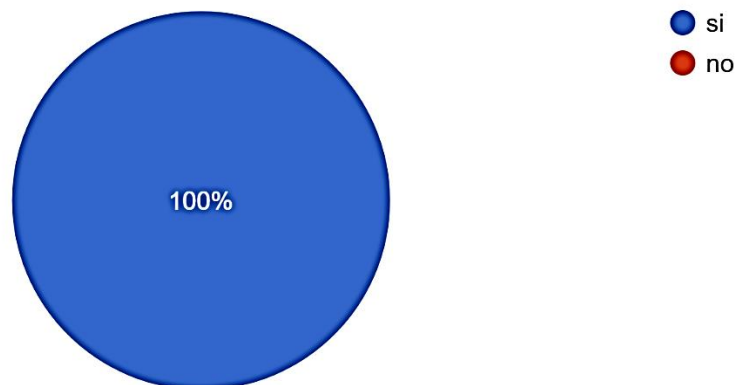
Grafica 3: Se muestra que la institución proporcionaba un formato (Rendón, 2022).

4-. ¿Le gustaría un sistema que llevará a cabo el registro?



Grafica 4: Se muestra que los alumnos les gustaría un sistema (Rendón, 2022).

5-. ¿Considera que es más funcional llevar el control de la información con un sistema?



Grafica 5: Los alumnos consideran como mejor opción un sistema (Rendón, 2022).

Con la información obtenida se generan las siguientes preguntas de investigación:

¿Cuáles son las necesidades que cubrirá el nuevo sistema de control de horas para los usuarios de servicio social y prácticas profesionales en las instituciones?

¿Cuál es el método que se debe implementar para un mejor acceso de información sobre las horas realizadas de los alumnos?

¿Qué tipo de acceso y fiabilidad tendrá el método propuesto?

V. Marco conceptual o teórico

El presente trabajo se centra como un diseño de sistema de información para agilizar y facilitar a los alumnos y al área encargada verificar sus horas realizadas, será necesario mencionar algunos conceptos de ingeniería del software que sirvan de guía para apoyar la lectura.

A continuación, como menciona (Sommerville, 2005) softwares en una computadora son programas hechos por programadores profesionales a los cuales con frecuencia se les da mantenimiento para ofrecer una mejora además de que los mantenimientos son a largo plazo. Estos programas suelen ser de distintos tamaños y arquitecturas.

En cuanto a la ingeniería de software según Sommerville, reconocido como uno de los padres de la ingeniería de software “es una disciplina de la ingeniería que comprende todos los aspectos de la producción de software” (Somerville, 2004).

La IEEE por sus siglas en inglés (*The Institute of Electrical and Electronic Engineers*) el Instituto de Ingenieros Electricistas y en electrónica), es una de las asociaciones más grandes del mundo, define a la ingeniería de software como la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software” (Gómez, Cervantes & González, 2019).

El Software de aplicación, son aquellos programas que cumplen una tarea específica en los negocios. Las aplicaciones procesan datos técnicos o comerciales que ayuda a facilitar las operaciones en negocios y apoyo para tomar decisiones administrativas o técnicas. Además de llevar a cabo el procesamiento de datos en tiempo real, por ejemplo, procesamiento en transacciones de punto de venta, también en proceso de manufactura en tiempo real (Pressman, 2010).

Según el autor Pressman (2010), el área técnica de software es la encargada del diseño y desarrollo de software, una vez que se ha analizado y modelado los requerimientos que ocupara el sistema podemos ir a la última acción fase de la ingeniería de software en donde se prepara para la etapa final de su desarrollo, que es la construcción y prueba de código. Los elementos de modelado son los requerimientos del software que dará la información para poder crear los modelos de diseño necesarios para la especificación completa del diseño, en la siguiente figura se mostrará el flujo de información durante el diseño de software.

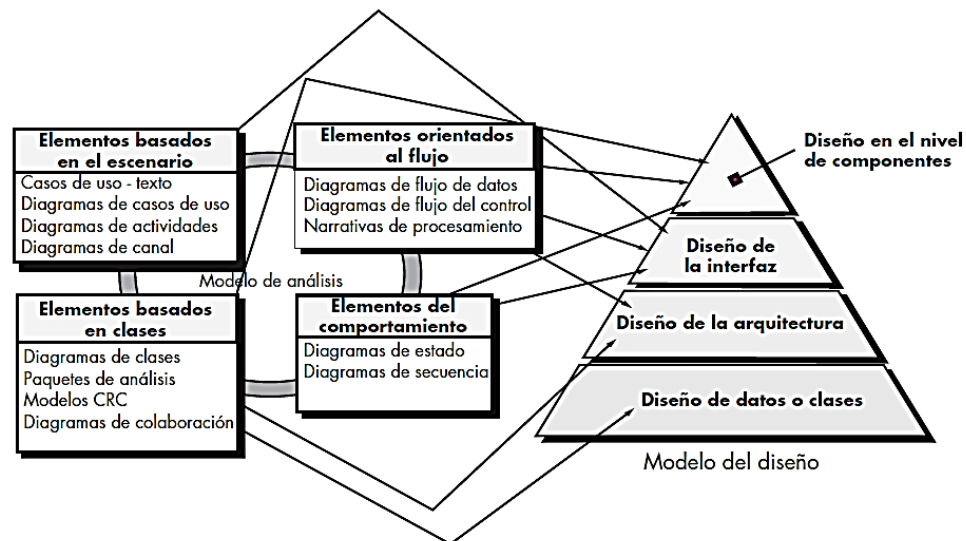


Figura 1: Clasificación de modelos concretos en clases abstractas (Pressman, 2010).

Figura 1. Clasificación de modelos concretos en clases abstractas (Pressman, 2010).

En sus inicios de las aplicaciones web solamente eran una colección de páginas simples estáticas con documentos y lo único que se podía hacer era consultar o descargar información, con el paso del tiempo (del usuario), este método era conocido como CGI (Common Gateway interface), que consiste en pasar información de servidores HTTP y programas externos (Mateu, 2004).

Según el autor Fernández (2006), un sistema de información ya es un conjunto de elementos que interactúan entre sí para llevar a cabo un objetivo, aunque un sistema sea diferente de otro es posible hacer un modelado general formado por cinco bloques básicos: elementos de entrada, elementos de salida, sección de transformación, mecanismos de control y objetivos.

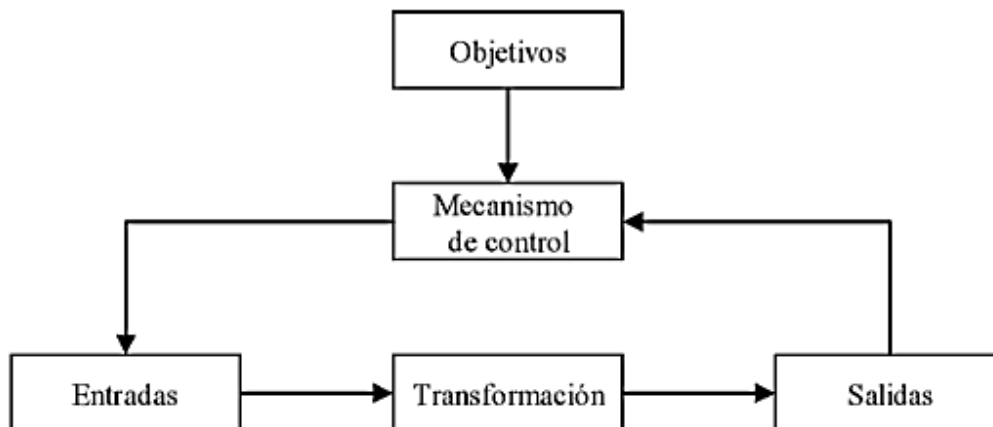


Figura 2: Modelo general de un sistema (Fernández, 2006).

En el mundo actual está lleno de sistemas desde una máquina expendedora de bebidas, una fábrica de productos, un automóvil, etc. en el caso de la máquina expendedora bueno el elemento de entrada corresponde a la ranura en donde se inserta la moneda una vez estando la moneda adentro se compara con el precio de la bebida seleccionada (objetivo del sistema) mediante la interfaz de control. Tío cuando el sistema realiza la comparación de la cantidad de dinero introducida con el costo de la bebida el mecanismo cambia las monedas por una bebida y la bebida es entregada, de manera similar con los ejemplos anteriores (Fernandez, 2006).

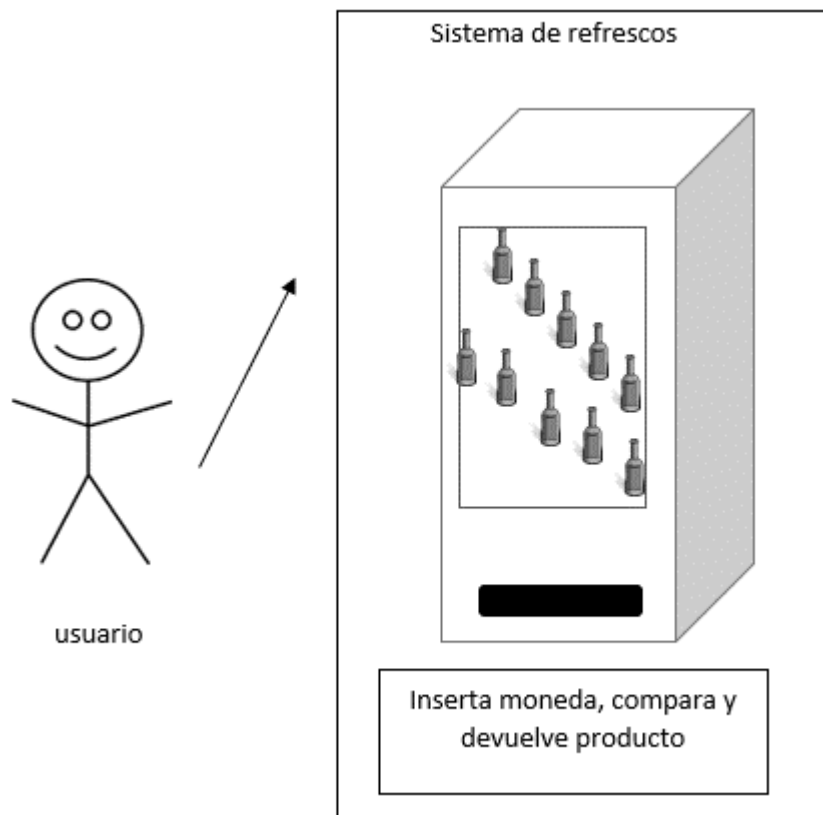


Figura 3: Sistema automatizado de venta de bebidas (Rendón, 2022).

La programación orientada a objetos es un paradigma, su intención es llevar al mundo real los objetos para la solución de problemas que se presentan en los programas informáticos. los objetos poseen propiedades (estado), por ejemplo una pelota de baloncesto tiene propiedades como radio peso y color. al igual el balón contiene acciones tales como rebotar, rodar y girar, cada una de ellas son verbos. Traducido a un lenguaje de programación orientada a objetos estos suelen llamarse atributos. son un conjunto de características y métodos.



Figura 4: Características de un objeto (Rendón, 2022).

Se pueden representar las características de los objetos, formando un modelo para la representación de objetos del mismo tipo, a esto se le llama clase. Se podría decir que una pelota de baloncesto Pertenece a la clase “pelotas de baloncesto”, con las clases se crean objetos y con ellos crear atributos (Rivera, 2020).

5.1. Requerimientos mínimos para ejecución de aplicaciones java web

Según la página oficial de Oracle para ejecutar aplicaciones compatibles con la aplicación se debe utilizar un sistema operativo WINDOWS VISTA SP2, WINDOWS 7 SP1, WINDOWS 2008, WINDOWS SERVER 2012, WINDOWS 8 y WINDOWS 10, contar con una memoria mínima RAM de 128 megabytes, un espacio en disco duro local de 124 megabytes, un procesador mínimo Pentium 2 a 266 MHz, tomar en cuenta exploradores como Firefox e Internet Explorer 9 y superiores.

Tomando en cuenta los requerimientos de la página oficial ahora Oracle se debe de considerar un navegador que cuente con los complementos compatibles de la plataforma ya que al momento de intentar compilar el programa no siempre muestra el resultado esperado ya que suele estar con los elementos movidos o incompletos, si se tiene un equipo antiguo es necesario utilizar un navegador web liviano para que este favorezca al momento de utilizar la aplicación y no se vean lento. considerar la memoria RAM del equipo ya que depende de cuánta memoria tenga disponible, al ser mayor la memoria RAM el programa se puede ejecutar normalmente y no genera que el programa sea lento.



Figura 5: Logo oficial (Oracle, 2022).

5.2. Base de datos

Una base de datos es una colección de archivos, se tiene un archivo en forma digital que cuente con datos de una tienda en la cual contenga los datos como los productos, pedidos entre otros datos conocido por sistema manejador de base de datos (DBMS, por sus siglas en ingles), ayudará al usuario a llevar a cabo consultas de datos. Bueno las bases de datos funcionales son de tipo relacional, lo que quiere decir que se utilizan tablas y que tenga un campo en común. A continuación, se muestra una tabla llamada "Product" y otra tabla llamada "Invoice", en donde se observará a través de las tablas las relaciones que existen entre ellas a partir de un campo en común.

Stock_code	Description	Prize
A416	Clavos, caja	0,14 Dolares
C923	Chincheta, caja	0,08 Dolares

Figura 6: Muestra los atributos de la tabla Stock_code (Gilfillan, 2003).

Invoice_code	Invoice_line	Stock_code	Quantity
3804	1	A416	10
3804	2	C923	15

Figura 7: Muestra los atributos de la tabla Invoice_code (Gilfillan, 2003).

Las dos tablas tienen en común el campo "Stock_code" la cual muestra el tipo de base relacional ya que la mayoría de las bases de datos grandes son de este tipo y es necesario tener un campo en común.

5.3. Terminología sobre bases de datos

A continuación, se estudiará cómo se compone las dos tablas anteriores

- Cada tabla está compuesta por una cantidad de filas y columnas.
- Cada fila tiene datos sobre una sola entidad, a este tipo de datos se le llama registro. Cuando en cuenta la tabla "producto" que describe el producto con un tipo de clave **A416** Y qué es una caja de clavos que cuestan 14 centimos de dólar.
- Las columnas tienen datos en común con el registro, se le conoce como atributos un ejemplo de ello es la cantidad de un artículo vendido o el precio sobre el producto. los atributos harán referencia a una tabla de base de datos que se les dominan campos.

Gracias al tipo de estructura de la base de datos anterior, le puede brindar a las búsquedas sobre registros y llevar a cabo cambios que se van realizando durante el desarrollo. Además, todos los sistemas manejadores de bases (DBMS, por sus siglas en inglés) de datos utilizan este tipo de lenguaje (Gilfillan, 2003).

5.4. SQL

Es un lenguaje de tipo relacional y de procedimientos que se define cómo las operaciones deben llevarse a cabo y en qué orden. Los lenguajes de no procedimiento son aquellos que ofrecen un resultado de una operación, el software determinará cómo se procesan las operaciones (Oppel & Sheldon, 2010).



Figura 8: Simbología en diagramas Base de datos (Rendón, 2022).

5.5. MariaDB

MariaDB es un motor de base de datos de código abierto popular, los usuarios disfrutan de esta herramienta ya que su objetivo principal de los desarrolladores originales de MySQL es mantener un MySQL “de código abierto”. Esta herramienta ofrece rendimiento, estabilidad y apertura. Actualmente cuenta con la compatibilidad con Galera Cluster 4, una característica agregada con Oracle Database quitar las temporales y en su mayoría esta herramienta la utilizan los equipos que tienen Linux (MariaDB.org, 2022).

5.6. Conexión a una base de datos

El equipo en el que se va a instalar el gestor de base de datos (MariaDB), en donde se almacenará la información se le llamará servidor, para lograr la conexión con este equipo se debe disponer de un equipo llamado cliente en otra computadora, este último puede ser cualquier equipo que contenga gestor de base de datos, como ejemplo, tenemos la siguiente figura:

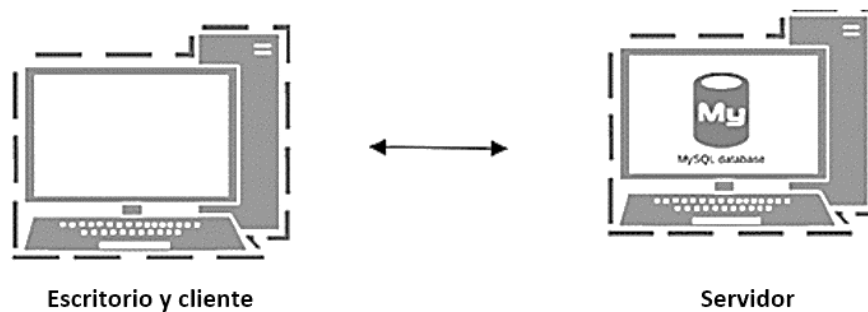


Figura 9: Conexión entre Servidor y Cliente (Rendón, 2022).

En el equipo cliente no está instalado el gestor de base de datos (MariaDB), se requiere conectar un segundo equipo se tendrá que utilizar un protocolo llamado telnet (Es un protocolo para establecer conexiones remotas con otros ordenadores), el cual nos ayuda a facilitar la conexión entre dos equipos. Para acceder desde consola a una base de datos sólo se requiere la siguiente línea de código para poder acceder a ella (Gilfillan, 2003).

```
% mysql -h nombre del anfitrión -u nombre de usuario -p contraseña nombre de la base de datos
```

Figura 10: Conexión entre Servidor y Cliente (Rendón, 2022).

5.7. Pruebas

Según los autores Bruegge & Dutoit (2002), es el proceso en donde los desarrolladores observan el comportamiento que tiene el sistema en su primera ejecución, basados en los modelos del sistema que se tenían contemplados. Las pruebas estructurales encuentran diferencias entre el modelo del diseño del sistema y un subconjunto de sus sistemas integrados. Las pruebas funcionales encuentran diferencias entre el modelo de casos de uso y el sistema. por último, las pruebas de desempeño encuentran diferencias dentro de los requerimientos no funcionales y el desempeño real del sistema. El objetivo principal es diseñar pruebas que manifiesten los desperfectos del sistema. Estas actividades son lo contrario a todas las demás ya que estas pruebas están diseñadas para el quebramiento del sistema, por lo regular estas pruebas las llevan a cabo programadores que no tuvieron nada que ver en el desarrollo del sistema.

Basado en la información anterior podemos considerar el desarrollo de un sistema. debemos tomar en cuenta los requerimientos del usuario y el cliente para poder así modelar nuestro proyecto, el sistema podrá hacer consultas de datos como insertar, eliminar, actualizar y editar. Conectar a una base de datos y posteriormente con una interfaz gráfica que le permitirá al usuario interactuar con el sistema utilizando un controlador modelo-vista-controlador.



Software Update

Figura 11:El sistema se actualiza constantemente durante su uso (Rendón, 2022).

5.8. Comunicación de proyectos

La ingeniería de software es una actividad de colaboración, se refiere a reunir a personas con diferentes conocimientos, como analistas, diseñadores, programadores, etc, ya que ninguno de los participantes puede comprender todos los sistemas y es fundamental por que dependen del uno al otro para realizar el desarrollo del sistema y poder avanzar a la siguiente fase. Los participantes informan el avance con minutas en la cual se reflejan los puntos que indican y las alternativas de diseño y cada una de éstas es apoyada con su documentación (Bruegge & Dutoit, 2002).

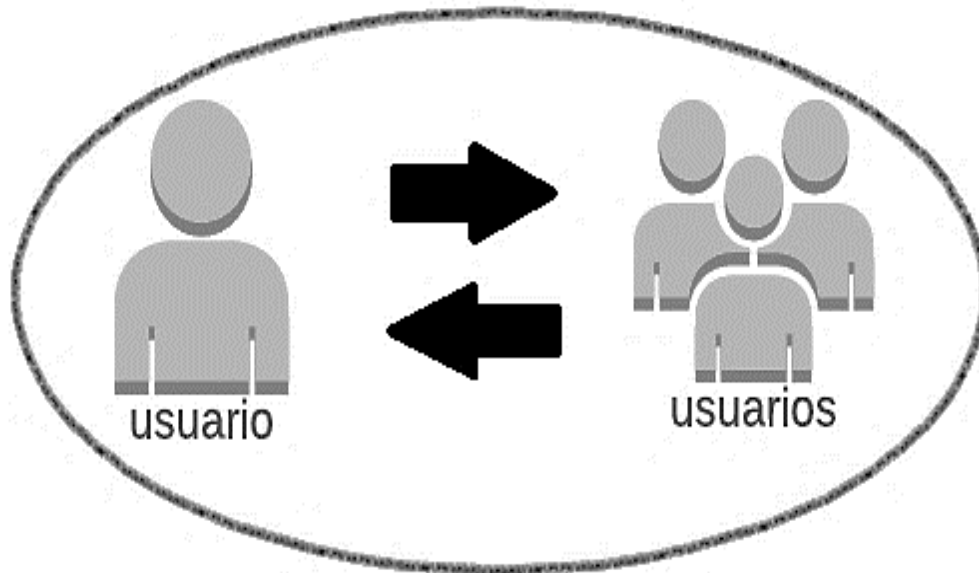


Figura 12: Representación de comunicación para el desarrollo (Rendón, 2022).

5.9. Especificación de Requerimientos de software

Según el autor Wong (2017), menciona algunos puntos a para los obtener requerimientos de software, con el fin de implementar un sistema eficaz ya que debe cubrir las necesidades y ser funcional para el usuario final.

- **Naturaleza de la especificación de requerimientos de software:** Hace referencia a Aspectos básicos como funcionalidad, interfaces, atributos, rendimiento y restricciones de diseño.
- **Medio ambiente de la especificación de requerimientos de software:** El software puede contar con la funcionalidad completa del proyecto o formar parte de un sistema más grande. Además, tomar en cuenta las dependencias que puedan existir a nivel de la aplicación, ya sea base de datos o servicios.
- **Características de la especificación de requerimientos de software:** Un requerimiento del software debe ser específico no debe de poseer ambigüedad, y ser clasificada por importancia y/o estabilidad y trazable.
- **Preparación conjunta de la especificación de requerimientos de software:** Para el desarrollo del software es necesario un acuerdo entre proveedor y cliente sobre las necesidades que el sistema debe de cumplir al estar completado. generalmente el cliente y el proveedor deben de colaborar para describir una buena especificación. por lo cual es necesario elaborar minutas de trabajo que describan los acuerdos cliente-proveedor contemplar la aprobación del usuario final.

- **Evolución de la especificación de requerimientos de software:** Conforme va avanzando el desarrollo de software será necesario que el usuario final Participe mi Revise cada avance para poder a probarla y pasar a la siguiente fase del diseño.
- **Prototipado:** La razón por la cual es necesario elaborar un prototipo del diseño del sistema es que al usuario se le facilitará la comprensión de cómo se va a desarrollar el sistema además de una retroalimentación rápida, además con un prototipo se puede visualizar imprevistos al comportamiento del sistema y así generar nuevas preguntas para el desarrollo y por último un prototipo tiende a sufrir menos cambios durante el desarrollo mejorando los tiempos que se tenían previstos para el desarrollo del sistema.
- **Diseño de inclusión en la especificación de requerimientos de software:** Gracias que se tiene un diseño se puede identificar cada uno de los componentes y sus interfaces, además un requisito específica funciones que el sistema debe de cumplir.
- **Incorporación de los requisitos del proyecto de especificación de requerimientos de software:** Las especificaciones de requerimientos deberán ir enfocados en los elementos que ocupará el sistema y llevar a cabo el desarrollo del del producto de software.

5.10. Captura de requerimientos

Según los autores Gómez at al. (2019), propone obtener mejor información acerca sobre los requerimientos de un sistema es necesario captura información

de los clientes y los usuarios sobre qué esperan que el sistema haga. Durante el proceso de la captura de información el analista identificara algunos puntos clave, existen herramientas para obtener información sobre los requerimientos, aquí mencionaremos los más comunes, la entrevista directa con el cliente y los usuarios, observación de tareas, en las que se rebelan problemas, detalles y estructuras de trabajo que son difíciles de captar con las entrevistas, obtener información hasta que se forma un modelo que cumpla las expectativas del cliente (Gómez al at., 2019).

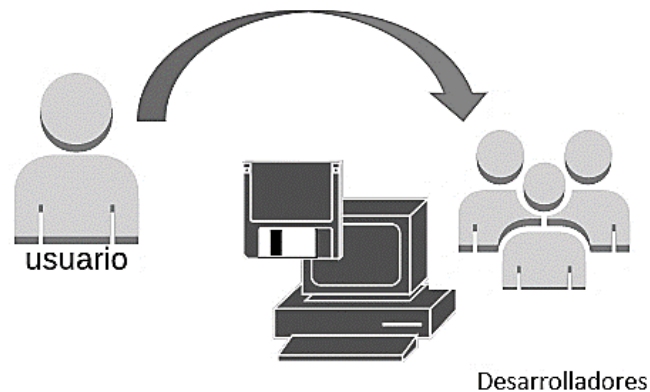


Figura 13: Desarrolladores en la fase de recolección de información (Rendón, 2022).

5.11. Análisis de los requerimientos

En esta fase del desarrollo del software se extraen las características y operaciones del software, se especificarán las interfaces entre el usuario y el sistema, en donde se establecerá el alcance que tendrá el sistema. También se traducen los requerimientos del cliente a un software. Además de identificar los conflictos que existen entre los requerimientos y se propone una solución. Con el fin que el analista pueda reconocer los elementos básicos que señala el cliente y el usuario final (Gómez et al., 2019).

5.12. Especificación de los requerimientos

Es un documento que especifica de manera completa y precisa los requisitos, el diseño y el comportamiento del sistema, mientras que el análisis de requerimientos ofrece un camino para que el cliente y los desarrolladores lleguen a un acuerdo sobre las funciones que debe realizar el sistema. Se utiliza un estándar como el de la IEEE, 830, PSS-05 de la Agencia Espacial Europea describen una estructura específica para presentar los requerimientos. Para el desarrollo de software es necesario contar con una especificación completa ya que en alguna etapa de mantenimiento era fácil. La especificación del requerimiento ofrece al técnico y al cliente los medios para valorar la calidad del sistema ya finalizado (Gómez et al., 2019).

5.13. Diseño del sistema

Según los autores Bruegge & Dutoit (2002), es la transformación que parte del modelo de análisis en un modelo de diseño del sistema. Mediante su desarrollo se definen los objetivos del diseño del proyecto y descomponen a el sistema en subsistemas para que su desarrollo pueda ser por equipos individuales. Los diseñadores también diseñan el tipo de hardware y software en donde se va a ejecutar el sistema.

El desarrollo de un sistema no es algorítmico, los desarrolladores y académicos han propuesto soluciones en forma de patrón para dar con los problemas comunes y se han definido notaciones para la arquitectura de software.

A continuación, se enlistan algunas actividades de diseño que tienen impacto en el desarrollo de diseño de software:

- La definición de los objetivos de diseño
- La descomposición del sistema en subsistemas
- La selección de componentes hechos y heredados
- La correspondencia entre los subsistemas y el hardware
- La selección de infraestructura de administración de datos persistentes
- La selección de una política de control de acceso
- La selección de un mecanismo de flujo de control global
- El manejo de condiciones de frontera

5.14. Desarrollo de alcance del sistema

El alcance de un sistema define sus límites, lo que quiere decir, identificar las funciones que tiene el sistema dentro y fuera de él. Un punto importante es el tiempo y el presupuesto ya que estos dos factores definen alcance. Los actores se situarán fuera del alcance del sistema, estará representado con líneas que definirán los límites. Como los diagramas de casos de uso se crean al inicio de las etapas de ciclo de vida de un sistema, cambiara el tiempo y el presupuesto que se tenían planeado al inicio, ya que el programa se irá desarrollando, dependiendo de las necesidades de los usuarios y esto afecta lo antes mencionado (Kendall & Kendall, 2011).

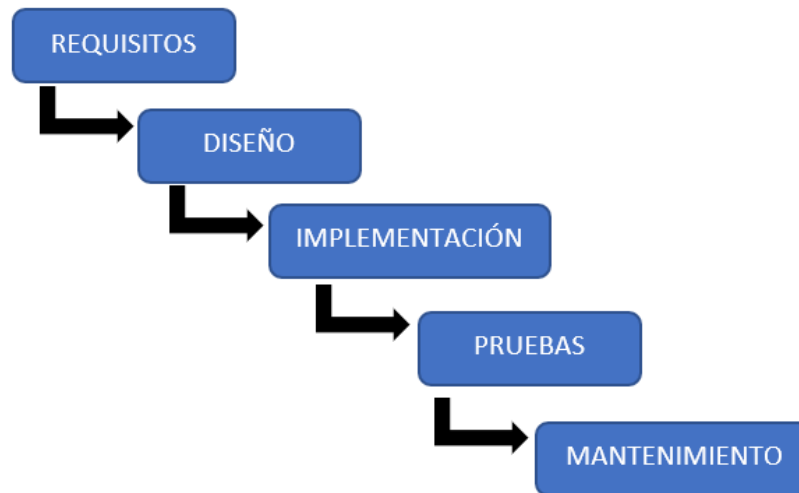


Figura 14: Visión general de un sistema en cascada (Rendón, 2022).

5.15. Ciclo de vida del desarrollo de software

UML (Lenguaje de Modelado Unificado), es independiente del proceso y por ello no está ligado a ningún ciclo de vida de desarrollo de software, pero si se desea obtener el máximo beneficio de UML se recomienda que un sistema debe considerar los siguientes procesos:

- **Dirigido por casos de uso:** Dirigido por los casos de uso significa que se utiliza para establecer el comportamiento deseado de un sistema para verificar y validar su arquitectura, ya que de esta forma se podrá visualizar la comunicación entre el usuario y el sistema.

- **Centrado en la arquitectura:** Esto quiere decir que la arquitectura del sistema se utiliza como un artefacto básico para conceptualizar construir y gestionar y llevar a cabo la evolución del sistema en desarrollo.
- **Iterativo e incremental:** Un proceso iterativo dice que gestiona las versiones ejecutables, un proceso incremental es aquel que considera la integración continua la arquitectura del sistema para producir esas versiones en la cual cada versión representa una mejora del sistema. en donde cada versión representa un riesgo o una mejora en el sistema, su objetivo es llevar al éxito al sistema.

En el proceso dirigido por casos de uso, tomando en cuenta la arquitectura, sí es iterativo e incremental, se puede representar en fase, la cual una fase es el intervalo de tiempo entre procesos, cuando se cumplen objetivos bien definidos y se completan los artefactos y se toman las decisiones para pasar a la siguiente fase. (Booch, Rumbaugh & Jacobson, 2006).

En la siguiente imagen se muestran cuatro fases del ciclo de vida de desarrollo de software en el cual consiste en concepción, elaboración, construcción y transición, mostrando los lapsos de tiempo Trabajo de una fase.

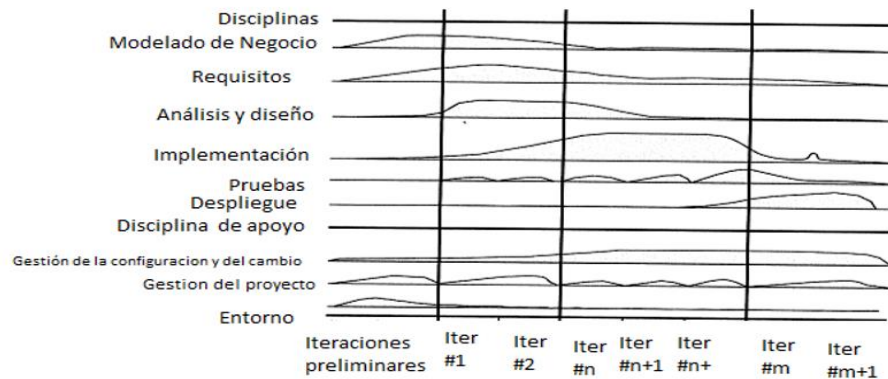


Figura 15: Ciclo de vida de desarrollo de software (Booch et, al 2006).

- **La concepción:** Es la primera fase del desarrollo del proceso, surge proponiendo una idea inicial y partir de ahí tener un punto el cual sea suficientemente fundamentada para garantizar el desarrollo en la fase de elaboración.
- **La elaboración:** Es la segunda fase del proceso, los desarrolladores definen los requisitos del sistema, proponiendo ideas claras a través de enunciados de carácter general criterios de evaluación y determinar si es funcional o no funcional, esta fase es donde se expresan con claridad los requerimientos del sistema.
- **La construcción:** Es la tercera fase del proceso, consiste en llevar el software desde una base arquitectónica ejecutable hacia la comunidad de usuarios que ocuparán el sistema, es aquí cuando el sistema es expuesto a los riesgos de esa forma poder asignar responsabilidades para la solución de ellos y poder así reducir riesgos durante el desarrollo.

- **La transición:** Es la cuarta fase del proceso, el software es llevado al usuario final, en esta fase el software tiene cambios que no estaban planeados al inicio, el usuario al usarlo tiende a descubrir más necesidades por lo cual en esta fase incluso el software se sigue mejorando ya que estas características no estaban incluidas en las versiones anteriores.

El único elemento que puede provocar un cambio durante el proceso es la interacción con un conjunto bien definido de actividades, como la interacción ofrece un producto ejecutable, después de cada iteración se pueden reevaluar los riesgos, esto quiere decir que el ciclo de vida del desarrollo del software tiende a sufrir un cambio continuo en sus versiones que parte desde la arquitectura del software (Booch, Rumbaugh & Jacobson, 2006).

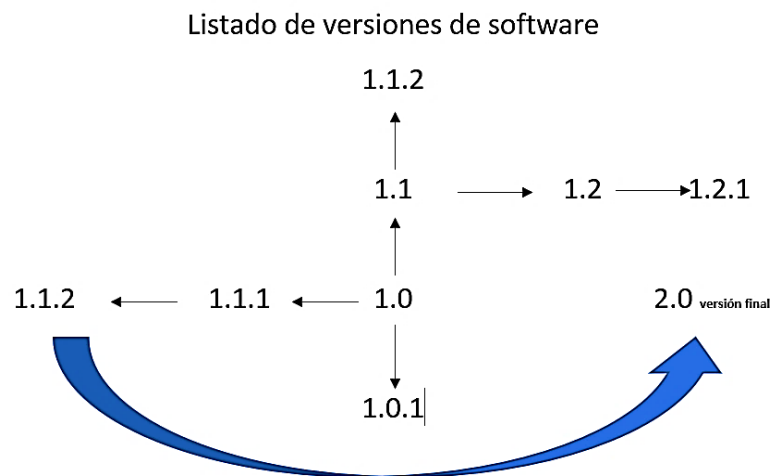


Figura 16: Un sistema puede sufrir cambios hasta lograr una versión estable (Rendón, 2022).

5.16. Modelo-Vista-Controlador

Según el autor Bascón (2004), el modelo MVC (Modelo Vista-Controlador), es un diseño de aplicaciones dividido en módulos los cuales se pueden identificar y cada una de ellas se define como el modelo, las vistas y el controlador.

- **Modelo:** Son clases que representan información de la vida real para que el sistema pueda procesar. Un ejemplo de ello es un sistema climatológico el cual va a tener información sobre temperatura, la humedad entre otros datos. No se tomará en cuenta la forma en la que el sistema va a procesar esa información ni los mecanismos que hacen que esa información esté dentro del modelo, en otras palabras, sin tener relación con otra entidad dentro de la aplicación.
- **Vistas:** Son agrupaciones de clases que llevarán a cabo la tarea de mostrar al usuario la información contenida en el modelo. Cada vista está asociada a un modelo sin embargo también es posible que puedan existir varias vistas a un modelo, un ejemplo de ello es tener la vista de un reloj analógico y que está a la vez tenga también la vista de un reloj digital.
- **Controlador:** Es un objeto encargado de dirigir la aplicación, así como los datos introducidos por el usuario y las opciones del menú seleccionadas por el usuario, a partir de los mensajes que manda el usuario la tarea del controlador será modificar el modelo de abrir y cerrar vistas.

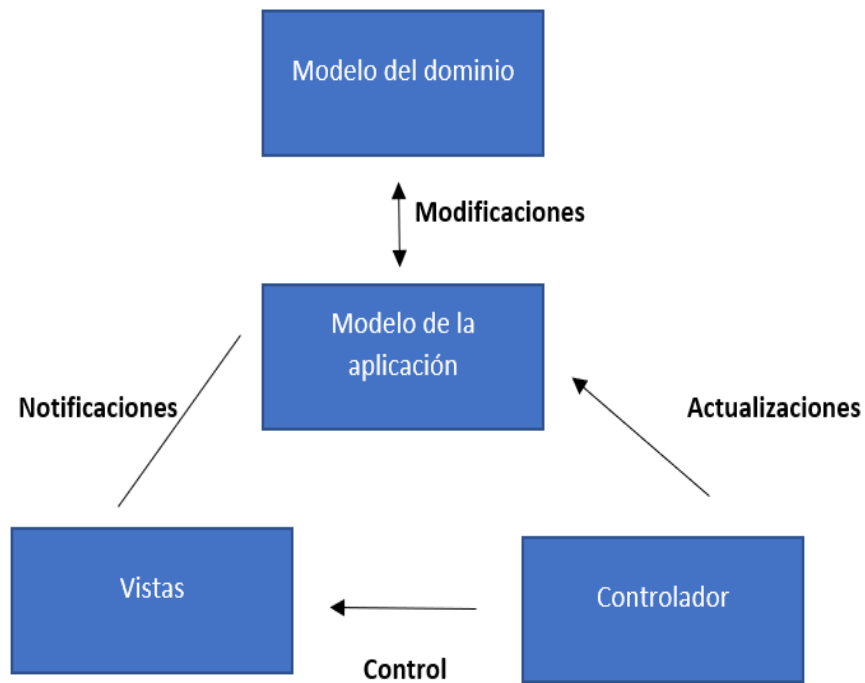


Figura 17: Relación entre los módulos del patrón MVC (Bascón, 2004).

En la actualidad estos conceptos se utilizan para el desarrollo de software y son la base de lenguajes modernos tales como C++, java o C# (Bascón, 2004).

5.17. Beneficios de usar MVC

Uno de los beneficios de utilizar esta herramienta es que se puede desarrollar un sistema con rapidez de forma organizada, separa las funciones de la aplicación con módulos y los sistemas antiguos se pueden adaptar al uso de esta herramienta y en automático toman una forma nueva de desarrollo. Permite a los desarrolladores llevar un trabajo en conjunto y avanzar con rapidez en el

prototipo, dado que está separado por módulos permite realizar cambios en la aplicación sin que los demás módulos se vean afectados (CakePHP, 2022).

5.18. UML (Lenguaje Unificado de Modelo)

Según los autores Booch, Rumbaugh, & Jacobson (2006), en general es un lenguaje estándar y se usa para escribir planos de software. Los desarrolladores lo utilizan para visualizar, especificar, construir y documentar los elementos de un sistema que tenga una gran cantidad de software. Se emplea para modelar sistemas empresariales y aplicaciones basados en la web. Se caracteriza por ser un lenguaje fácil de aprender y de aplicar, además cuenta con 3 elementos principales, los cuales son: los bloques básicos de construcción de UML (Lenguaje Unificado de Modelo), las reglas que dictan cómo pueden combinarse esos bloques y mecanismos que se aplican a lo largo de todo el lenguaje.

5.18.1. UML es un lenguaje

Es un lenguaje estándar que proporciona reglas y vocabulario para combinar palabras con la finalidad de posibilitar comunicación entre el sistema. Es un modelado cuyo reglas y vocabulario se especializan en la representación física y conceptual de un sistema, en pocas palabras sirve para generar los planos de un software. Gracias al uso del modelado a los programadores se les facilita la comprensión de un sistema, la mayoría de los casos se debe de realizar más de un modelado para entender cualquier cosa, a menudo se necesitan múltiples modelos conectados entre sí. Para aquellos sistemas que tengan una complejidad grande de software es necesario un lenguaje que alcance a cubrir las diferentes vistas de la arquitectura del sistema conforme al ciclo de vida del desarrollo del software (Booch, Rumbaugh & Jacobson, 2006).

5.19. Desarrollo de diagramas de casos de uso

Se representará con una ilustración que mostrará los eventos que describen el comportamiento de un sistema. Para recolectar información sobre las necesidades del sistema es necesario preguntar a los usuarios un listado sobre las actividades que quieren que el sistema haga por ellos, una técnica simple de obtener esa información sería por medio de entrevistas. Se deberá tener en cuenta los siguientes lineamientos:

- Identificar actores involucrados y especificaciones del negocio
- Identificar eventos principales de alto nivel, esto quiere decir roles de los usuarios.
- Revisar cada caso de uso principal para determinar variaciones de flujo. Con este análisis establecer rutas debido a que cada caso de uso es distinto, identificar actividades que puedan tener éxito o fracasar.

Un diagrama de flujo de datos a nivel contexto puede ser de ayuda para el analista para que tenga la capacidad de generar un caso de uso y llevar a cabo el desarrollo del diagrama en cada uno de sus escenarios del sistema (Kendall & Kendall, 2011).

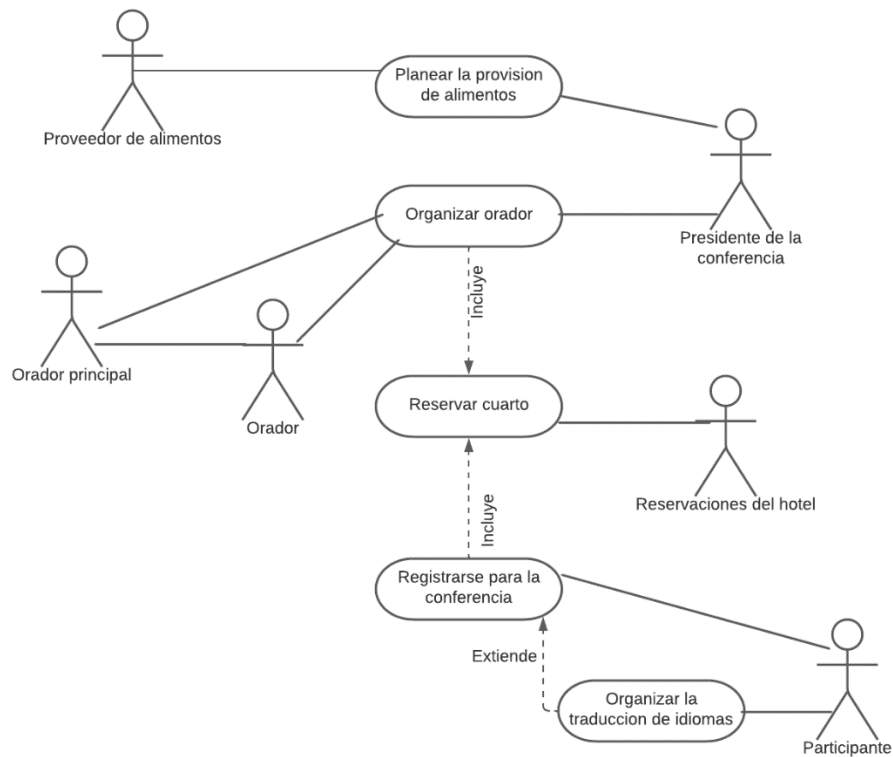


Figura 18: Diagrama de casos de uso que representa al sistema utilizado para planear una conferencia (Kendall, & Kendall. 2011).

5.20. Diagrama de clases

Los diagramas de clases muestran las características del sistema y las relaciones que hay entre clases, están representados con un rectángulo que puede incluir el nombre de la clase e incluir atributos y métodos mejor conocidos como operaciones, estos atributos son las características de los objetos, regularmente los atributos son privados y en el diagrama están representados con signo “-” al inicio del nombre del atributo, de igual manera también pueden ser protegidos e incluso se indica con un símbolo de “#” . Un diagrama clase puede mostrar solo el nombre de la clase con sus atributos y los atributos con sus métodos, en ocasiones cuando el diagrama es muy complejo es útil sólo mostrar el nombre y

si el diagrama es muy pequeño es conveniente mostrar todos los atributos y los métodos (Kendall & Kendall, 2011).

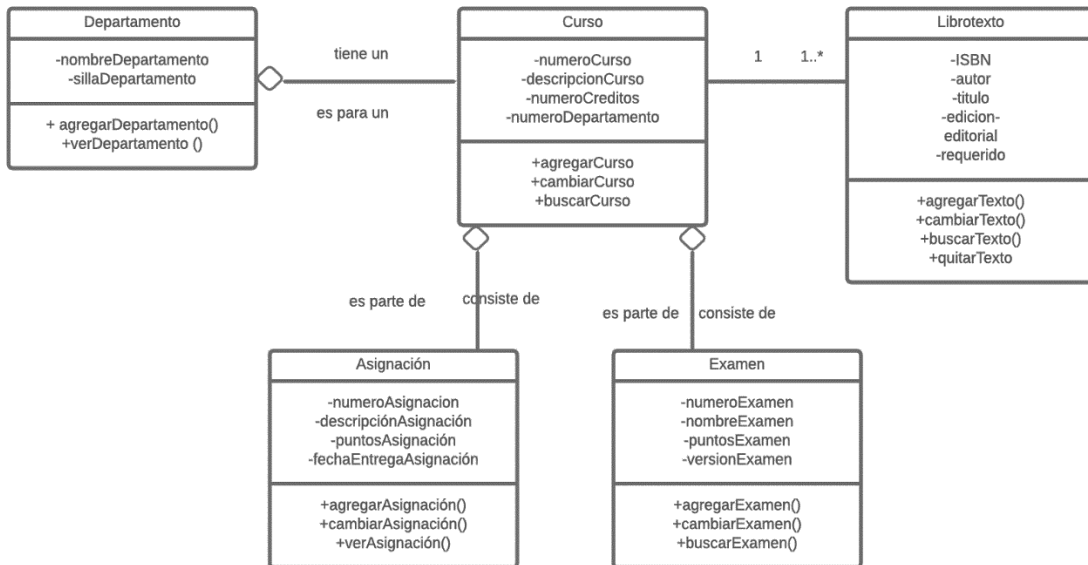


Figura 19:Diagrama de clases para los ofrecimientos de cursos (Kendall, & Kendall. 2011).

5.20.1. Las clases

Es una representación abstracta que describe un conjunto de objetos, son datos abstractos que presentan un conjunto de objetos que en común tienen una misma estructura de datos y un mismo comportamiento. En pocas palabras son la misma cosa y hacen lo mismo (López, 2013).

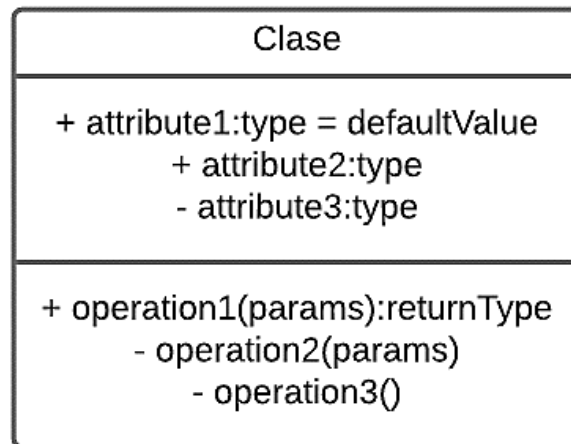


Figura 20: Estructura general de una clase (Fuente propia, 2022).

5.20.2. Métodos

Son acciones que se ejecutan por medio del uso de los objetos, un método puede leer una entrada, realizar cálculos y dar una salida de datos. Se pueden especificar mediante el nombre que se le da el método y se logra construir programas modulares los cuales nos permite ocuparlos de nuevo mandando a llamar el método y ya no escribir código repetido (López, 2013).

5.20.3. Encapsulación

Son un conjunto de datos y métodos dentro de un objeto. El desarrollador deberá pensar en el código y los datos a la vez durante el desarrollo del sistema. El código ni los datos existen en un vacío, los datos dirigen el código y el código manipula la forma de los valores de los datos (López, 2013).

5.20.3. El modelo relacional

Según los autores (Ramos & Ramos, 2007), está basado en un modelo matemático y su principal característica es mantener la independencia de la estructura lógica respecto al modo de almacenamiento y otras características de tipo físico, el objetivo del modelo relacional son los siguientes:

- **Independencia física:** Se refiere aquel modelo de almacenamiento de datos no debe influir en su manipulación lógica.
- **Independencia lógica:** Los cambios que se realicen en los objetos de la base de datos no deben de afectar a los programas y a los usuarios.
- **Flexibilidad:** Se debe explicar a los usuarios de una manera sencilla la forma de usar una aplicación.
- **Uniformidad:** Se muestra la estructura de los datos, las tablas en la manipulación de la base de datos por parte de los usuarios.
- **Sencillez:** Las características anteriores, así como un lenguaje de usuario fácil de comprender, así que los modelos sean sencillos de utilizar por el usuario.

Todos los datos de una BD (Base de Datos), se representan con relaciones dependiendo el tipo de contenido que va cambiando durante el tiempo. El modelo relacional se caracteriza porque está basado en ramas matemáticas, teoría de conjuntos y lógica de predicados, esto ayuda al modelo que sea seguro y sólido.

RELACIÓN ALUMNOS

NUM_MAT	NOMBRE	APELLIDOS	CURSO
5467	JUAN	CABELLO	1BACH-A
3421	DOLORES	GARCIA	1BACH-C
7622	JESUS	SANCHEZ	2BACH-C

← ATRIBUTOS

← TUPLAS

Figura 21: Representación de una relación en forma de tabla (Ramos & Ramos, 2007).

La figura anterior muestra los atributos y tuplas (tipo de dato que ofrece agrupar en una sola unidad diversas variables), en forma de tabla y cuenta con las siguientes características:

- No admiten filas duplicadas.
- Las filas y las columnas no cuentan con orden.
- La tabla es plana y cuentan con un solo valor no se admiten atributos multi valuados.

5.20.4. Modelo Entidad-Relación

La herramienta entidad relación ayudará a generar el diagrama para poder otorgar una estructura a los datos obtenidos para poder así darle solución a un problema o actividad real, en pocas palabras generar claves o asignar dentro de una base relacional. En el modelo se presentarán las relaciones 1:N y N:M. de las cuales existen 2 tipos de entidades, las fuertes y las débiles. Para identificar la entidad débil será requerida la clave de la entidad fuerte para identificar los registros, la cardinalidad de una relación de una entidad débil con su maestro siempre será 1:N (García, 2014).

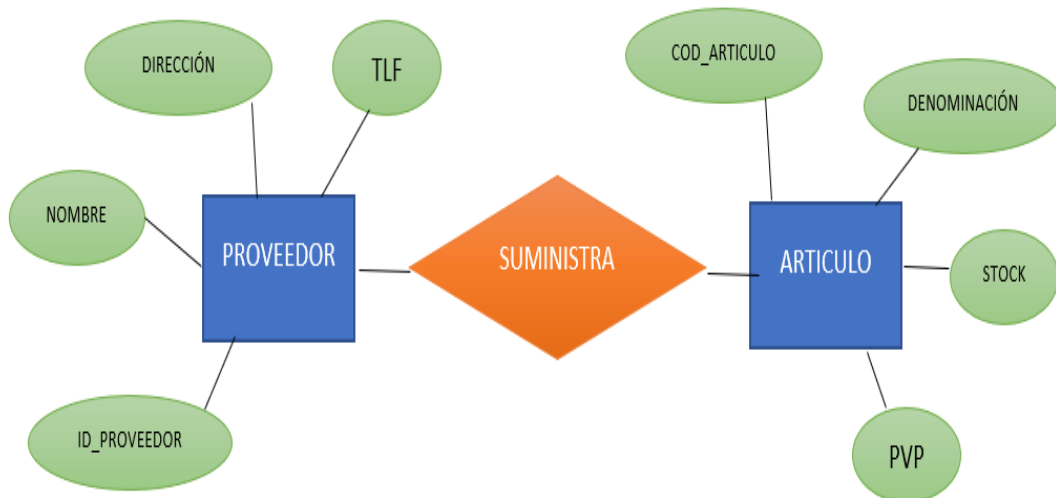


Figura 22: Diagrama E-R un proveedor suministra varios artículos (Ramos & Ramos, 2007).

5.20.5. Elementos del modelo E-R

Entidad: Es un objeto del mundo real, hace referencia al tipo de dato que va relacionado con el objeto del mundo real, un ejemplo sencillo es la entidad ALUMNO de un centro escolar, la entidad CLIENTES de un banco el cual se va a representar con la figura rectángulo. Hay dos tipos de entidades:

- **Entidad fuerte:** Se caracteriza por ser una entidad principal la cual no depende de otra entidad para existir.
- **Entidad débil:** Este tipo de entidad sí depende de otra entidad para existir.

Componentes de las entidades:

- **Atributos o campos:** Es la información que describe la propiedad de la entidad, por ejemplo, ALUMNOS los atributos serían número de matrícula, nombre del alumno, dirección, población, etc. Y cada uno de estos valores se denomina dominio, lo que quiere decir que son el conjunto de caracteres de tipo texto o número que tienen una longitud determinada, se representan de forma de eclipse.
- **Clave primaria o principal:** Es un número único que identifica a cada entidad, la característica principal de la clave primaria es no tener valores nulos y que este número no cambiará con el tiempo.

A continuación, se muestra la figura con la entidad alumnos y sus atributos, además mostrar la clave primaria.

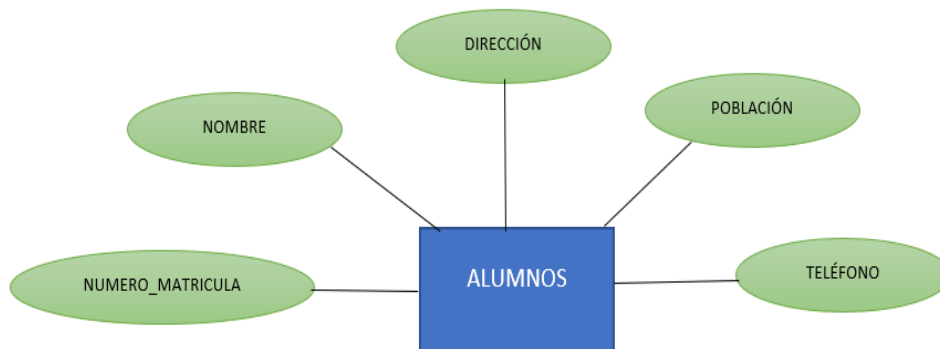


Figura 23: Representación de la entidad ALUMNOS y sus atributos (Ramos & Ramos, 2007).

- **Clave ajena o foránea (foreign key):** Es el atributo que forma la clave primaria en otra entidad, eso quiere decir que si tenemos la entidad ALUMNO la llave principal es NUMERO DE MATRICULA como se muestra en la imagen siguiente, la llave foránea es NUMERO ASIGNATURA de la entidad ASIGNATURA, servirá para la conexión de una entidad a otra.

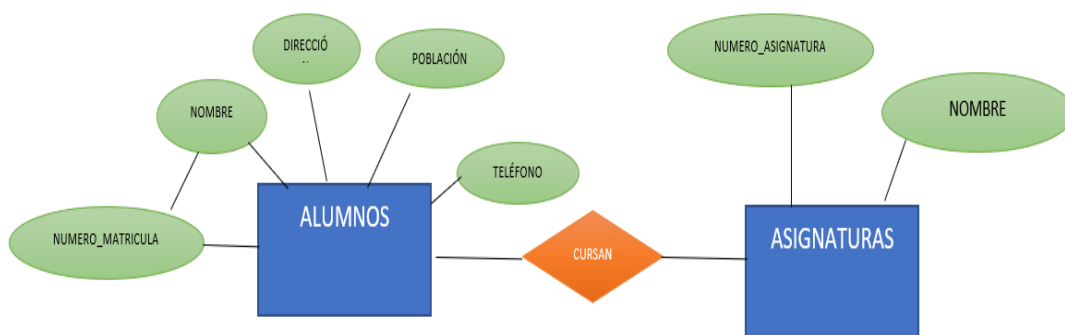


Figura 24: Representación de una relación (Ramos & Ramos, 2007).

- **Relación:** Es la Unión entre 2 o más entidades, se caracteriza por tener nombre de un verbo que se identifica con respecto a las otras relaciones. Las relaciones no suelen tener atributos, además recibe el nombre de entidad asociada y dará origen a una nueva tabla que contendrá otros atributos.

El modelo entidad-relación (E-R), creado por Peter Chen en 1976 es un modelo el cual ayuda a representar datos en el mundo real para poder visualizar los datos de las entidades y las relaciones entre ellas de una manera gráfica, se utiliza para el diseño de modelos relacionales (Ramos & Ramos, 2007).

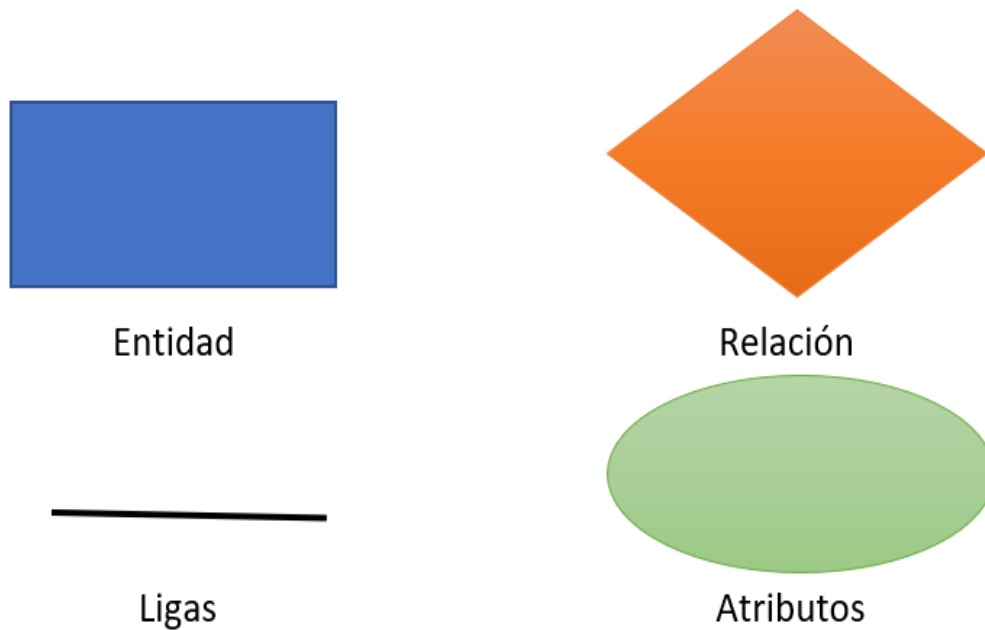


Figura 25: Simbología del modelo E-R (Rendon, 2022).

5.21. Conceptos básicos del servidor web

Según el autor Mateu (2004), un servidor web es un programa cuya función es atender y responder las peticiones de los navegadores, solicitando las herramientas necesarias mediante los protocolos HTTP (Protocolo de Transferencia de Hiper Texto) o HTTPS (Protocolo de Transferencia de Hiper Texto Seguro). Estos servidores básicos tienen un esquema muy sencillo y se ejecutan de manera infinita. A continuación se muestran los pasos de ejecución:

1. Espera peticiones en el puerto TCP (Transfer Control Protocol) con el puerto 80 asignado.

2. Recibir una petición.
3. Buscar el recurso en la cadena de peticiones.
4. Enviarle el recurso por la misma conexión de donde se ha recibido la petición.
5. Volver al punto 2.

El servidor web que sigue estos pasos Google y los requisitos básicos de servidores HTTP. Se han diseñado diversos servidores y varía el tipo de peticiones (páginas estáticas, Servlets, etc.).

5.21.1. Servlets (Módulos java)

Se le denomina servlet por el termino applet, se desarrolla en Java y se ejecuta a través de un servidor y no en el navegador del cliente como es común en los applets. Es una herramienta completa que ofrece bibliotecas de clases para que el desarrollador pueda implementar de una manera sencilla la funcionalidad de la aplicación web. Una de las ventajas que ofrece al programador esta herramienta es centrarse en la lógica de su aplicación sin tener que preocuparse por los detalles de los protocolos HTTP (Protocolo de Transferencia de Hiper Texto). Además, al ser un lenguaje que se apoya con Java orientado a objetos las aplicaciones suelen ser flexibles hacia los usuarios y elegantes (Berzal et al., 2005).

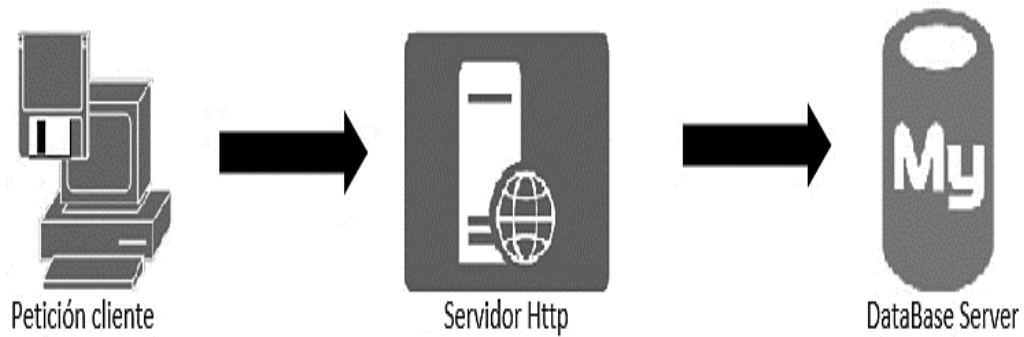


Figura 26: Applet se ejecuta desde un servidor (Rendón, 2022).

5.21.2. Páginas de servidor ASP (Active Server Pages) y JSP (Java Server Pages)

Este tipo de servidor permite preparar documentos HTML (Lenguaje de Marcas de Hipertexto), para introducir fragmentos de código que será interpretado por el servidor web cuando el usuario inserta una petición. En conclusión, en vez de crear programas que incluyan código necesario para generar el documento HTML, crearemos documentos HTML que incluyen el código de la aplicación en su interior. JSP (Java Server Pages) es un ejemplo de Microsoft del uso de esta herramienta (Berzal et al., 2005).

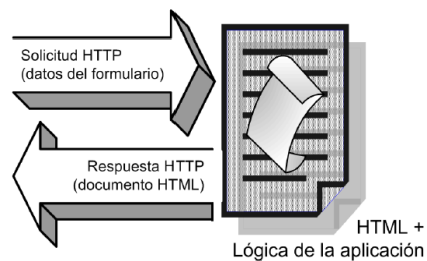


Figura 27: Funcionamiento de las páginas de servidor, Una página ASP/JSP (Berzal et al., 2005).

Servlet es una herramienta que resulta útil para el programador, ya que da por hecho que no tiene que tratar directamente con los mensajes HTTP que se muestran en el navegador del cliente, el diseño de estas aplicaciones en ocasiones no es elegante, al combinar la interfaz de usuario con la lógica de la aplicación se pierde el diseño (Berzal et al., 2005).

5.22. Desarrollo web

El desarrollo web ha avanzado en los últimos años, todo esto gracias a las nuevas herramientas de desarrollo, ya que empresas dedicadas al desarrollo lanzan nuevas librerías, pero al igual desaparecen algunos lenguajes de programación que se vuelven obsoletos. Los programadores deben de estar actualizados con las nuevas tecnologías para poder así aprovechar las nuevas herramientas de programación. Bootstrap una de las herramientas de desarrollo creada por Mark Otto y Jacob Thornton desarrolladores de Twitter han hecho que su Framework sea popular entre los programadores gracias a su facilidad de uso (Aguirre, 2021).



Figura 28: Representación de librerías antiguas que ofrecía HTML con solo texto (Rendón, 2022).

5.22.1. Aplicaciones web

Los proyectos elaborados en HTML son estáticos, lo que quiere decir que tienen una forma simple de mostrar datos, y las aplicaciones web ofrecen un mejor manejo de la información ya que y con el uso de esta herramienta dinámica podemos seleccionar, filtrar, ordenar y presentar la información de una forma más adecuada de acuerdo con las necesidades del usuario. Las aplicaciones web necesitan un software ejecutándose en el servidor o en un equipo cliente para generar ficheros HTML para que el usuario pueda visualizarlo, todo esto se lleva a cabo con el protocolo HTTP (Protocolo de Transferencia de Hiper Texto), sólo que ahora el servidor HTTPS (Protocolo de Transferencia de Hiper Texto Seguro), encarga a otros módulos de generación dinámica de las páginas HTML (Lenguaje de Marcas de Hipertexto) que se envíen al cliente. Para el para el cliente la conexión se realiza de la misma forma y se sigue generando páginas HTML ya que el navegador del cliente es independiente al servidor y desde el punto de vista del programador dispone de herramientas para generar un contenido dinámico para ofrecerle al usuario desde un programa ejecutado desde el servidor o incluso en el propio equipo del usuario.



Figura 29: El contenido que se le muestra al usuario se genera dinámicamente para cada solicitud proveniente del navegador web instalado en la máquina cliente (Berzal et al., 2005).

5.22.2. HTML

En sus inicios las páginas web no contaba con un almacenamiento de datos, solo contaba con ficheros de textos a los que se les llamaban etiquetas, estas contenían fragmentos de texto y le daban un formato, además de crear enlaces para ir de un documento a otro Inclusive dentro del mismo documento. Hoy en día para crear un HTML (Lenguaje de Marcas de Hipertexto), se necesita de los conocimientos mínimos, solo debemos de preparar la documentación HTML de tal manera como queramos que lo visualicen los usuarios de la página (Berzal et al., 2005).

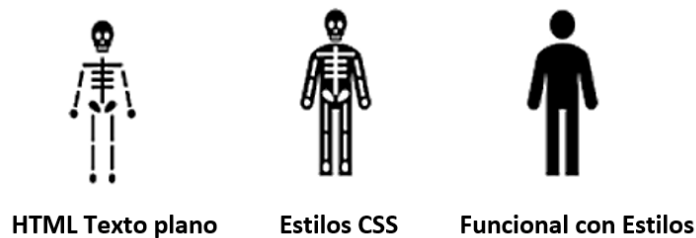


Figura 30: Representación mediante cuerpos de HTML y CSS (Rendón, 2022).

5.22.3. CSS

Se conoce como lenguaje de hojas de estilo y se utiliza para darle estilos a los elementos escritos en lenguaje de programación HTML (Lenguaje de Marcas de Hipertexto), fue desarrollada en el año 1996 por la empresa W3C (World Wide Web Consortium). Se caracteriza por utilizar etiquetas, el lenguaje HTML, esta herramienta no contaba con etiquetas, una de las principales causas del desarrollo de esta herramienta fue que HTML intento incluir este tipo de etiquetas y estilos fondos y colores, pero al ver que los sitios web tenían diferentes fuentes

era difícil para los desarrolladores utilizar esta herramienta y por eso fue creado CSS (Gustavo, B. 2022).

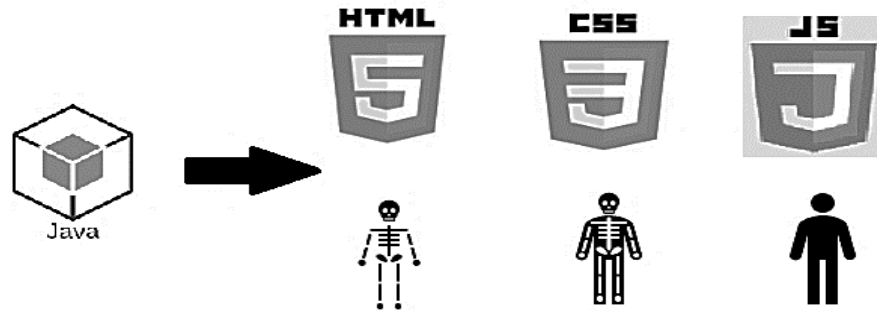


Figura 31: Java puede usarse con las diferentes herramientas de estilos (Rendón, 2022).

5.22.4. Lenguaje java

El lenguaje de programación Java contiene clases y objetos los cuales se representan por instancias de objetos de alguna clase. Además, se conservan algunos tipos de datos como son int, float y double se mantienen como opciones por eficiencia y su equivalente en forma de clase en Java. Para poder observar el código a continuación se presenta un “Hola mundo!” (Cervantes et al., 2016).

```
1 // Ejemplo simple
2 public class Ejemplo {
3 // main es el estándar principal
4 public static void main( String[] args ) {
5 // imprime algo en la salida estándar
6 System.out.println( "Hola mundo!" );
7
```

5.23. Frameworks de programación

Según Ortega (2017), se refiere a un esquema de replicación de software que lo conforman componentes y relaciones entre ellos, está conformado por la abstracción de clases, objetos y los componentes. Otro componente son las conexiones directas de base de datos como MySQL, SQL Server y Oracle. A grandes rasgos se les conoce como ODBC (Open Data Base Connectivity).

A continuación, en la siguiente tabla se mostrará la Clasificación de Frameworks según su facilidad de hacer cambios:

Tipo de framework	Descripción
Caja blanca	Poseen características de la habitación a objetos como herencia o ligadura dinámica, de modo que se extienden por herencia de sus clases base o redefiniendo sus métodos. también definen interfaces para componentes que pueden integrarse mediante composición de objetos. requieren de un conocimiento y comprensión de las clases a extender, así mismo, la dependencia de entre métodos por causa de la herencia en el caso de redefinir una operación puede necesitar redefinir otra y así sucesivamente
Caja negra o dirigidos por datos	Se estructuran utilizando comprensión de objetos y delegación en lugar de herencia, comprendiendo objetos existentes de diferentes formas para reflejar el comportamiento de la aplicación.
Caja gris	Es una mezcla de los tipos anteriores, permite la extensión mediante la herencia la ligadura dinámica, además de ocultar información innecesaria a los desarrolladores.

Tabla 2: Clasificación de Frameworks (Ortega, 2017).

5.23.1 Spring

Según el autor Pérez (2016) es un framework y un contenedor. Desde su creación con su versión 1.0 rápidamente se fue haciendo popular entre los programadores hasta que en su versión 1.2.6. alcanzó el reconocimiento de Jolt Awards y Jax Innovation Awards en el año del 2006. Las aplicaciones javas son objetos que trabajan entre sí para conseguir un propósito, esto quiere decir que un elemento depende de otro. Las plataformas javas carecen de organización durante la creación de distintas partes de una aplicación, para esto sirve el componente de inversión de control framework spring ya que con la ayuda de esta herramienta facilita y soluciona esta problemática ya que propone de manera formal componer y combinar los distintos componentes de una aplicación.

5.23.2 Componentes de una aplicación Spring

A continuación, se muestran los componentes de un framework Spring:

- **Contenedor de inversión de control:** Se encarga de la configuración de los componentes de la aplicación y gestiona las instancias de las clases java.
- **Programación de orientada a aspectos:** Facilita la implementación del código transversal.
- **Acceso de base de datos:** Ayuda a interactuar con los sistemas gestores bases de datos con ayuda de los conectores como es el JDBC que se utiliza en este proyecto.

- **Modelo vista controlador (MVC):** Es un framework basado en controladores que contiene vistas JSP (Java Server Pages), ayuda en la personalización de aplicaciones web.
- **Acceso remoto:** Con el uso de tecnologías como RMI (invocación Remota de Métodos), COBRA y protocolos basados en HTTP (Protocolo de Transferencia de Hiper Texto), ayuda a la importación, exportación de estilos y objetos java.
- **Procesamiento por lotes:** Consiste en el procesamiento de tareas, el cual se divide en lotes como son la gestión de transacciones, estadísticas de procesamiento de trabajos, inicio de tareas, etc.
- **Autenticación y autorización:** Esta herramienta es un proceso de seguridad, el cual cuenta con un amplio estándar de protocolos que ayudarán a la seguridad del sistema.
- **Administración remota:** Con la ayuda de la herramienta JMX (Java Management Extension), Se puede llevar a cabo la configuración a través de un modo remoto.
- **Testing:** Este módulo sirve para el soporte de clases y el desarrollo de pruebas.

5.23.3 Características de Spring

Gracias a que es una herramienta que es única versión básica que pesa 2 megabytes ofrece beneficios ya que es un Spring Open Source. A continuación, se menciona los beneficios al utilizar spring:

- Ayuda a los desarrolladores a utilizar la herramienta POJO (Plain Old Java Object), ya que no necesita de tener un contenedor EJB (Enterprise JavaBeans), sino que sólo se puede utilizar Servlets.
- Están conformados por módulos, en la fase de pruebas es fácil gracias a su mecanismo de inyección de dependencias.
- Consta de un framework MVC (Modelo Vista-Controlador), para el desarrollo de aplicaciones web.
- Ofrece API (Interfaz de programación de Aplicaciones), para introducir excepciones específicas lanzadas por JDBC, Hibernate, etc.

5.23.4. Spring MVC

Es uno de los frameworks más aprovechados por los programadores en el mundo gracias al uso de esta tecnología permite crear aplicaciones utilizando el modelo MVC que genere una página HTML De manera sencilla. Empleando esta herramienta se trabaja un controlador frontal (FrontController), Su tarea es llevar a cabo las peticiones web para direccionarlas a los componentes que sean necesarios, a esto se le domina ServletDispatcher (Álvarez, 2021).

5.24. Diseño Arquitectónico

Un framework necesita tener una estructura física la cual está compuesta por carpetas y archivos, se muestra la figura que describe estructura de un framework.

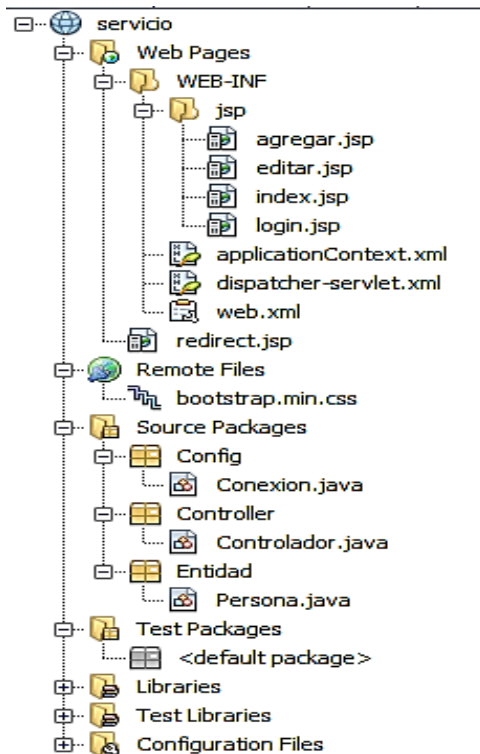


Figura 32: Estructura física de un Framework (Rendon, 2022).

A continuación, se describe cada uno de los componentes que forman la estructura básica de un framework, los cuales son los siguientes:

- **Configuración:** Es una carpeta que cuenta con los archivos necesarios para generar el funcionamiento de un framework, los cuales pueden ser la

configuración de base de datos. Este usa usuario, contraseña y el puerto a utilizar. Otro ejemplo de un inicio de sesión, cache y configuración para el funcionamiento de un sistema.

- **Content:** Esta carpeta contiene las imágenes, los estilos y los archivos de idiomas y cada una de ellas contendrá subcarpetas.
- **Kernel:** Se caracteriza por ser el núcleo de un framework, contiene todos los archivos de interfaces y está conformado por subcarpetas las cuales contienen librerías para en funcionamiento adecuado.
- **Layers:** Ya en esa carpeta se encuentran las clases y cada una de ellas contiene a sus carpetas las cuales guardan archivos de la capa generadora, se refiere al archivo que sirve para navegar en el sistema por medio de una página el cual puede ser <http://dominio/home>, se encarga de acceder al generador cuyo nombre de archivo es home y el número de la clase será HomeGenerator.

El conjunto de archivos de configuración de idiomas pueden contener la siguiente extensión .json ya que PHP (Hypertext Pre-Processor), tiene un analizador sintáctico para entender este tipo de archivos además es fácil de entender para el desarrollador (Ortega, 2017).

5.25. Bootstrap

Es un framework front-end utilizado para la creación de sitio web y aplicaciones móviles en el cual consiste en un diseño adaptado a la pantalla utilizado por el usuario. Inicialmente fue creado por Twitter en el año 2010 y era una herramienta

que utilizaba dicha compañía, al poco tiempo lo lanzaron como código abierto para así convertirse como hoy se conoce como “Bootstrap”. Este framework se encarga de combinar herramientas CSS y JavaScript Para utilizar en páginas HTML que nos permite crear un diseño más que el color y en botones. Por ello la página creada con estas herramientas ofrece al usuario interactividad por lo cual proporcionó la facilidad de comunicación con el usuario ya que contiene menús, controles de página, barras de proceso y otras herramientas que nos permiten controlar la página (Rockcontent, 2020).

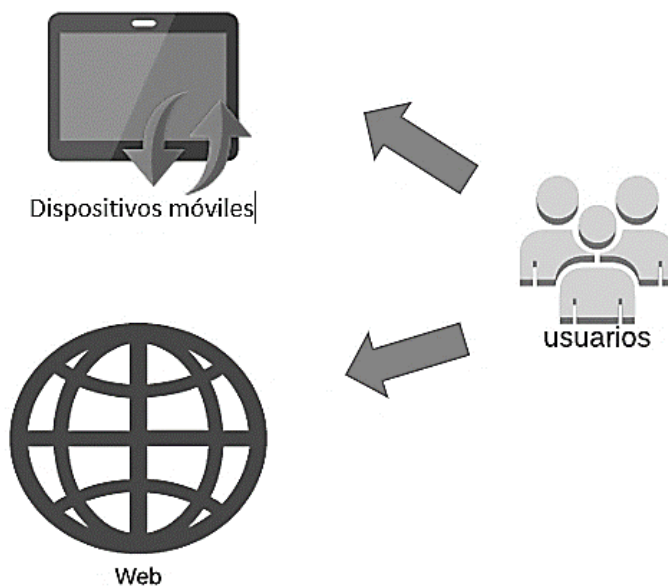


Figura 33: Bootstrap se caracteriza por ofrecer estilos visuales en equipos móviles y escritorio (Rendón, 2022).

Esta herramienta de programación es fácil de aprender, ya que cuenta con varias formas de trabajo, así los programadores pueden hacer uso de ella a su gusto dependiendo de las necesidades de los proyectos que tengan en mente. La

principal característica de Bootstrap es que se puede adaptar a los requerimientos del desarrollador. Al hacer uso de las librerías Bootstrap en un documento HTML limpio se puede ver la diferencia del cambio que nos otorgará al aplicar los estilos que nos ofrece (Aguirre, 2021).

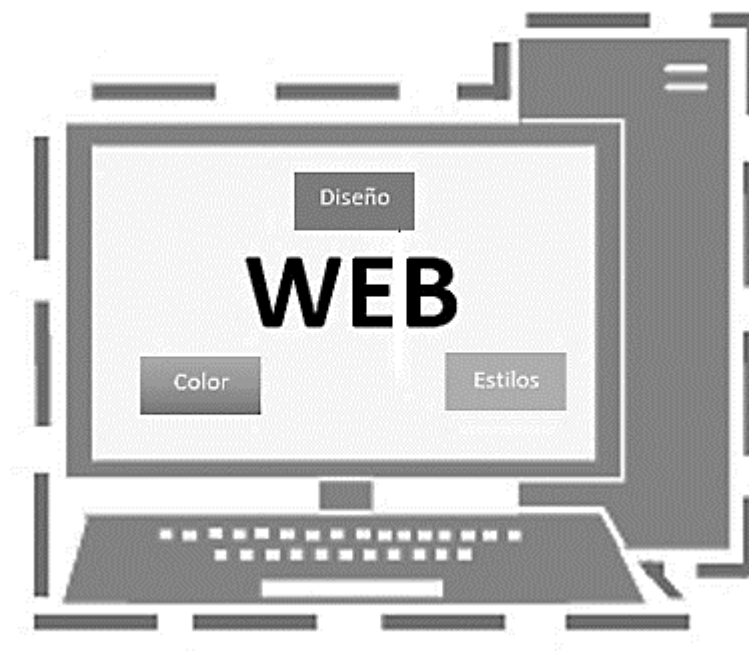


Figura 34: Bootstrap es una herramienta fácil de aprender y ofrecer estilos a páginas web (Rendón, 2022).

5.25.1. Como funciona Bootstrap

Esta herramienta está formada por archivos CSS y JavaScript que ayudan a crear características a los elementos de una página. Se crea un archivo con la extensión “.css” que tiene una definición para los estilos, prácticamente es la estructura que se compone de dos directorios los cuales son CSS y JavaScript.

- **CSS:** Incluye los archivos para agregar los estilos de los elementos y un tema único original.
- **JavaScript:** Incluye la ejecución de las aplicaciones para que el usuario pueda interactuar con la página.

La manera de asignar características a un elemento se debe tomar la clase correspondiente en la propiedad “class” del elemento que será utilizado, de esta manera se agrega los estilos a las clases y no a los elementos (Rockcontent, 2020).

VI. Métodos y técnicas de investigación

Se utilizará el método de investigación documental y experimental para el desarrollo de este sistema, ya que se hará una selección y recopilación de información que se obtendrán gracias a las entrevistas y a la información de materiales bibliográficos y repositorios digitales.

A continuación, se explicará de forma breve las dos metodologías que se utilizaran para el desarrollo de este sistema de información:

1. Investigación documental es una técnica de investigación básica que ayuda a recopilar datos, las herramientas que servirán para la recolección de datos serán desde libros, expedientes, informes de laboratorio o trabajos de campo con relación al tema a estudiar.
2. Metodología experimental son procedimientos científicos que permite el enfoque en relaciones empíricas entre variables o comprobar la veracidad de una hipótesis, ley como modelo a través de un experimento controlado.

Para el desarrollo del diseño del sistema de información se llevará a cabo el método analítico, debemos analizar las partes que conforman el sistema para que así podemos ver algunas problemáticas que se pueden generar durante el diseño y desarrollo de este.

Se apoyará con la metodología cascada ya que se adapta a nuestra necesidad del desarrollo del sistema la cual consiste en dividir el proyecto en etapas y no se puede avanzar si no es que se finaliza la etapa anterior del desarrollo.

En la ingeniería de software el desarrollo en cascada es conocido por la posición de las fases en las que el sistema está en su desarrollo, cada una de las fases parecieran caer en cascada esto nos ayuda a llevar un orden de las etapas de los procesos del desarrollo del software ya que el inicio de una etapa se debe esperar a la finalización de la etapa anterior para poder avanzar a la siguiente fase, el modelo en cascada fue uno de los primeros en utilizarse en base a todos los modelos de ciclo de vida (Maida & Pacienza, 2015).

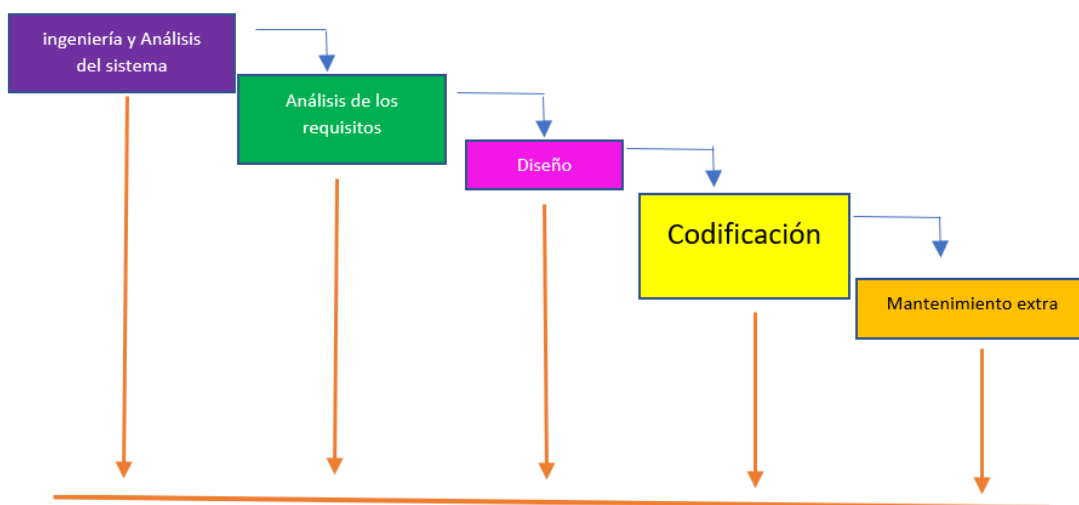


Figura 35: Modelo del ciclo de vida en Cascada (Maida & Pacienza, 2015).

VII. Presentación y discusión de resultados

En este capítulo se explicará todos los procesos que se llevaran a cabo para el diseño y desarrollo de sistema de visualización de horas.

El Análisis de requerimientos para este análisis del presente trabajo, se realizó una investigación previa y se les dio seguimiento, se estudió el comportamiento entre los alumnos y su bitácora de anotaciones con la cual día a día se va generando un registro. La información obtenida será útil para el diseño y desarrollo del sistema y determinar un objetivo que ayude a obtener un control adecuado de la información y que sea de manera instantánea.

En base a los requerimientos funcionales de lo que el sistema debe hacer, se encuentra la entrada de datos en el sistema, la operación de datos, y el resultado de datos. No obstante, se debe de tomar la interacción del usuario final y el sistema.

Después de la información obtenida se hará un análisis y se encontró los siguientes puntos:

- El sistema contará con una base de datos.
- Sistema tendrá sólo el usuario administrador.
- Los permisos asignados que tienen administrador son los siguientes: Insertar, eliminar, actualizar y editar. Con el fin de que administrador tenga un control adecuado del uso de la información.

7.1. Herramientas y tecnologías por usar

Tomando en cuenta los requerimientos funcionales, se llegó a la conclusión que se deben de utilizar las siguientes herramientas para el desarrollo del sistema:

- Lenguaje Java
- Gestor de bases de datos MariaDB
- Mysql Workbench
- Net beans 8.2
- Bootstrap (framework)
- Servidor

7.2. Desarrollo del sistema

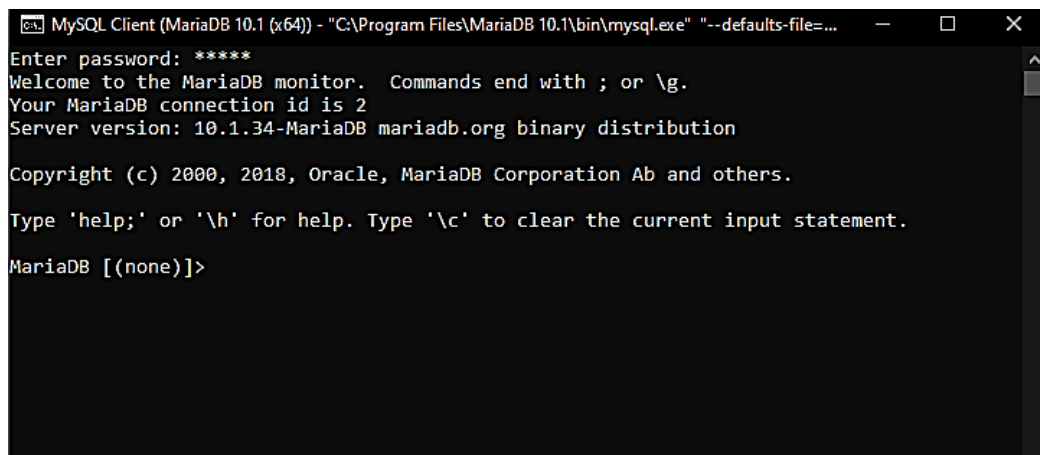
En este capítulo se escribe paso a paso sobre el diseño e implementación del sistema para el control de horas del personal que realiza servicio social o prácticas profesionales.

Se describe paso a paso cómo fue el desarrollo del sistema implementando algunas herramientas de páginas web que como objetivo principal será proporcionar una página web con estilos y con una interfaz amigable para el usuario. Se explica mediante diagramas cómo está compuesto el sistema internamente, cada diagrama se explica cuál es su función dentro del sistema. Por último, se muestran unas imágenes en donde el problema está en ejecución y se explica cada uno de sus apartados del sistema.

7.3. Como obtener MariaDB

Consultar la página oficial de MariaDB (<https://mariadb.org/download/>), en donde el usuario encontrará información sobre las versiones recientes y con los requerimientos del sistema para ver la compatibilidad sobre su computador además encontrará instrucciones que ayudan al usuario a su instalación.

El primer paso será desarrollar la base de datos utilizando MariaDB, para ello descargamos el software de la página oficial. Una vez descargado realizamos la instalación y se configurarán los puertos que vienen predeterminados, y se asignará una contraseña para poder ingresar, en la siguiente figura se observará cómo se ejecuta dicho proceso.



```
MySQL Client (MariaDB 10.1 (x64)) - "C:\Program Files\MariaDB 10.1\bin\mysql.exe" "--defaults-file=...
Enter password: *****
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 2
Server version: 10.1.34-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

Figura 36: Consola inicial de MariaDB (Rendón, 2022).

Como primer paso seleccionaremos el código que tenemos y copiamos en la consola para poder registrar una nueva base de datos principal.

Creada la base de datos realizaremos un nuevo proyecto utilizando Netbeans, eligiendo la opción Java web, seleccionar “siguiente” y elegir un nombre para el proyecto en este caso será “servicio”. Además seleccionaremos la ubicación que tendrá el proyecto, una vez más seleccionando “siguiente” seleccionaremos la opción Server y seleccionaremos “GlassFish Server 4.1.1” y la versión de “Java EE 7 Web” y seleccionar “siguiente”. En la siguiente interfaz seleccionaremos el framework que se va a utilizar “Spring Web MVC” e incluir las librerías “JSTL” y seleccionar “finalizar”. Una vez realizado estos pasos tenemos listo nuestra estructura que nos presenta Spring MVC con las carpetas necesarias y los archivos importables para la configuración XML para poder llevar a cabo el desarrollo del sistema.

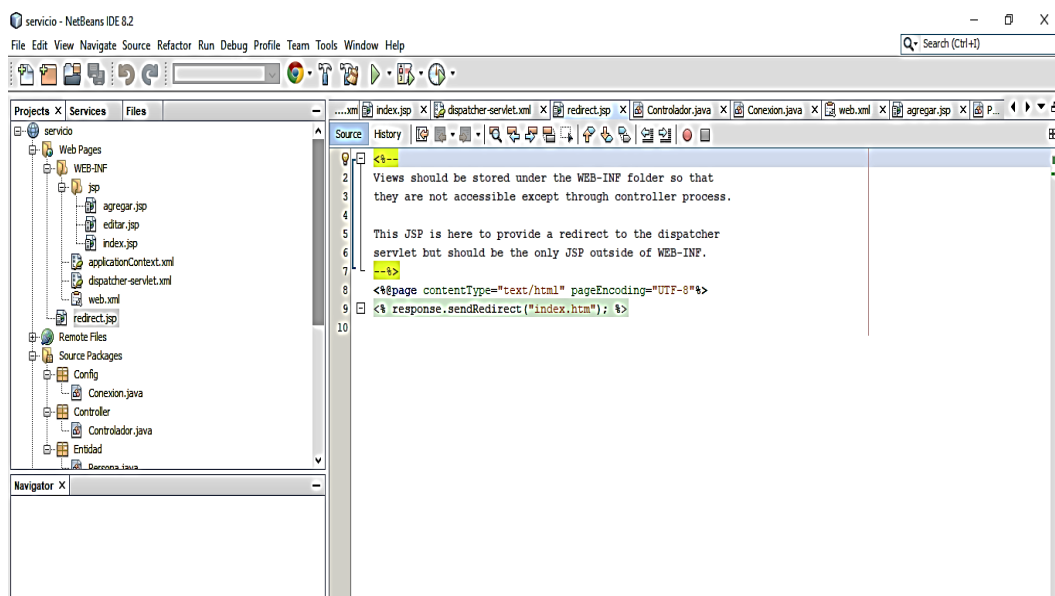


Figura 37: Interfaz inicial de Netbeans (Rendón, 2022).

El sistema integra cuatro funciones principales llamadas CRUD (Create, Read, Update y Delete). Para ello vamos a incluir la siguiente librería si buscamos el

siguiente archivo llamado “Mysql JDBC Driver” y agregamos como se muestra en la siguiente imagen:

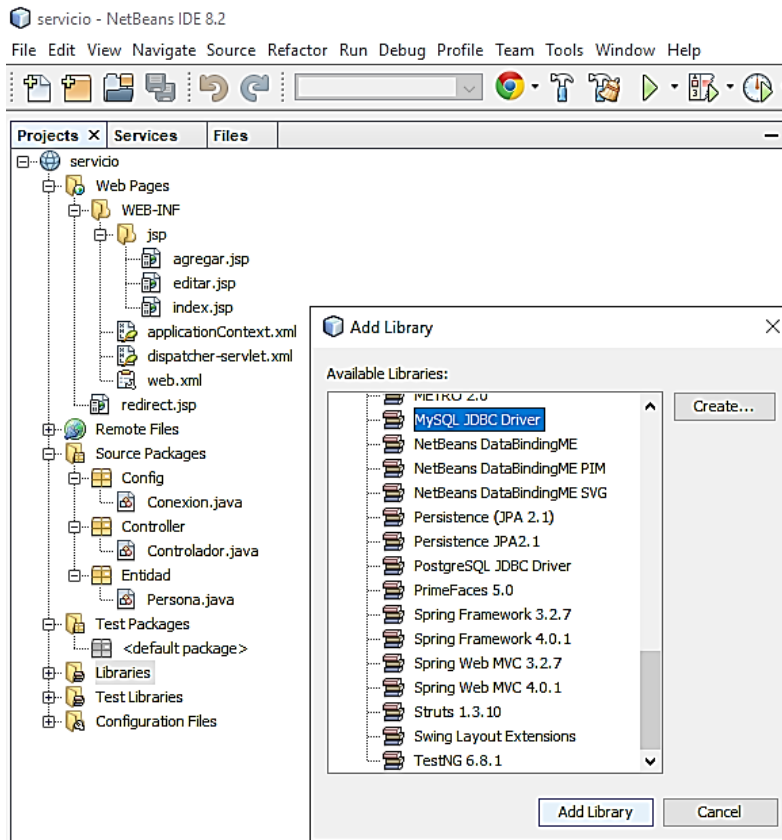


Figura 38: Selección de librería MySQL JDBC Driver (Rendón, 2022).

Una vez agregado la librería MySQL driver estará lista nuestra conexión para poder establecer comunicación con la base de datos y poder almacenar la información obtenida en el sistema que vayan generando los alumnos al transcurso de su instancia en las escuelas o empresas.

Agregada la librería procedemos a realizar la conexión con la base de datos creando una clase llamada “Conexión” ahí llevará el código con el usuario

y la contraseña para poder acceder a la base de datos llamada “alumno”, se hará referencia que será un sistema de manera local en donde se especificará el servidor, los puertos y el nombre de la base de datos, el Super Usuario que tenemos por defecto que será “root”. Así estará lista nuestra clase llamada “Conexión” a continuación, se mostrará el código que tiene nuestra clase:

```
1  * @author ernesto
2  */
3  public class Conexion {
4      public DriverManagerDataSource Conectar() {
5          DriverManagerDataSource dataSource=new
6  DriverManagerDataSource ();
7          dataSource.setDriverClassName ("com.mysql.jdbc.Driver");
8          dataSource.setUrl ("jdbc:mysql://localhost:3306/alumno");
9          dataSource.setUsername ("root");
10         dataSource.setPassword ("12345");
11         return dataSource;
12     }
```

Creado nuestro código fuente el sistema podrá hacer conexión con la base de datos y podrá almacenar los datos de los usuarios que van a ocupar el sistema, como punto principal se debe de agregar tal cual la dirección del host de la computadora local, ya que es la que va a realizar la conexión con la base de datos y de lo contrario este no hará una buena inserción de datos.

Creada la conexión de base de datos procedemos a crear una nueva clase llamada “controlador” y un paquete llamado “controller” cada que se agregará un controlador se deberá configurar el archivo llamado “dispatcher-sevlet.xml” al crear el proyecto en automático hace referencia a “IndexController”, el cual se va a sustituir por el un archivo llamado “index.htm”.

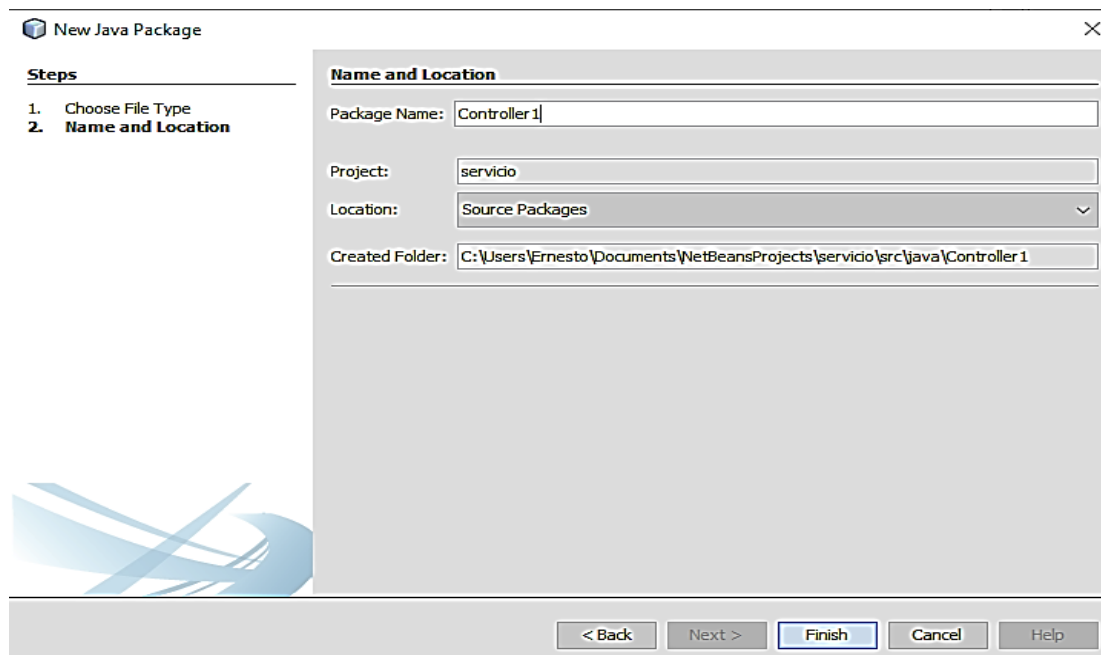


Figura 39: Creación del paquete llamado Index.htm (Rendón, 2022).

Código general proporcionado por la plataforma Java NetBeans, que servirá como página principal del sistema.

```
1 < bean id = "urlMapping" class =  
2 "org.springframework.web.servlet.handler.SimpleUrlHandlerMapping" >  
3     < nombre de la propiedad = "mappings" >  
4     < props >  
5         < clave prop = "index.htm" > controlador </ prop >  
6     </ props >  
7 </ propiedad >  
8 </ bean >
```

7.4. Listar

Agregado el archivo “Index” volvemos a nuestra clase de llamada controlador para listar los datos de nuestra tabla llamada “alumno” la cual servirá para visualizar la información que tiene la base de datos y así más adelante establecerla como interfaz principal del sistema.

```
1 @Controller
2 public class Controlador {
3     Conexion con=new Conexion();
4     JdbcTemplate jdbcTemplate=new JdbcTemplate(con.Conectar());
5     ModelAndView mav=new ModelAndView();
6     int id;
7     List datos;
8
9     @RequestMapping("index.htm")
10    public ModelAndView Listar(){
11        String sql="select * from alumno";
12        datos=this.jdbcTemplate.queryForList(sql);
13        mav.addObject("lista",datos);
14        mav.setViewName("index");
15        return mav;
16    }
```

Con esto estaría lista nuestra clase para poder así visualizar los datos de nuestra tabla, ahora vamos al archivo llamado “Index.jsp” en donde se van a utilizar los estilos de “Bootstrap” con la finalidad de poder visualizar mejor presentación a nuestra página principal del sistema.

Se utilizará el estilo de Bootstrap para tener mejor presentación, para ello vamos a ir a la página oficial de Bootstrap y vamos a copiar el código “CSS” que viene como ejemplo en la página y copiaremos el código y lo pegamos a nuestro archivo llamado “index.jsp”

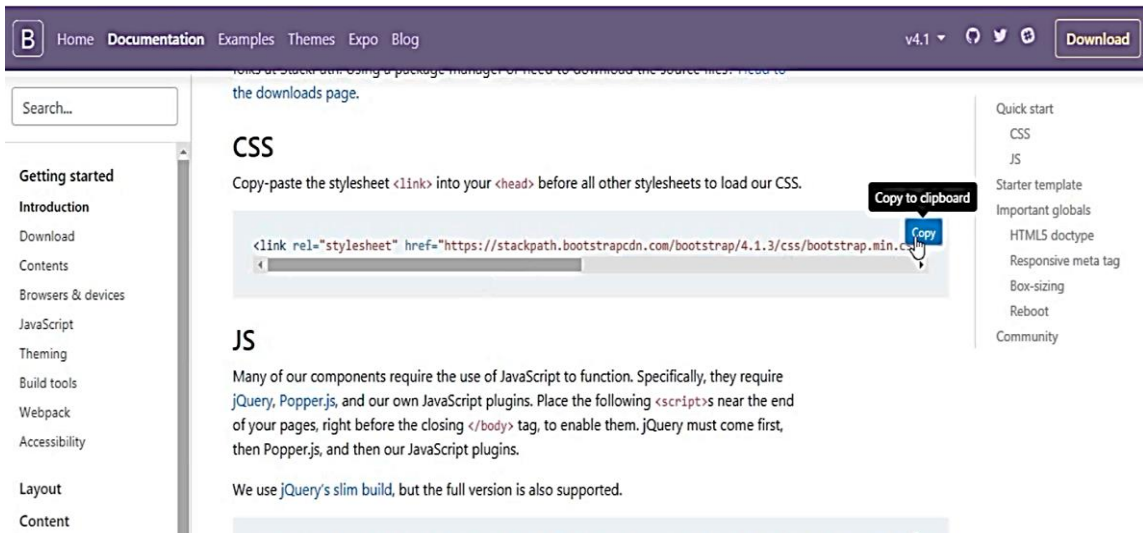


Figura 40: Página de inicio de estilos de Bootstrap (Rendón, 2022).

Copiado el enlace y pegado en el código principal “index.jsp” se visualizará el estilo de Bootstrap, por esta ocasión se utilizará este ejemplo, pero en la página oficial hay otros códigos en los cuales los programadores se pueden apoyar para poder darle un estilo único a sus programas.

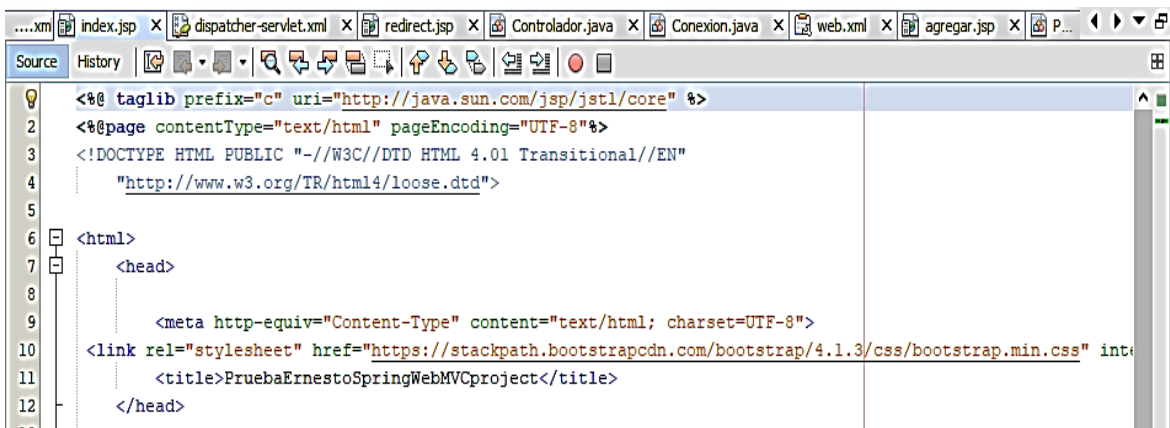


Figura 41: Ubicación de enlace para que los estilos tomen efecto en el proyecto (Rendón, 2022).

Los campos correspondientes se crean para visualizar la información, el texto quedará dentro de cajas de texto y al momento de llamar la base de datos se almacenarán ahí. Como ejemplo tenemos la siguiente imagen en donde se describe el código y las etiquetas:

```
1
2 <body>
3     <div class="container mt-4">
4         <div class="card border-info">
5             <div class="card-header bg-info text-white">
6                 <a class="btn btn-danger" href="agregar.htm">Nuevo Registro </a>
7
8             </div>
9             <div class="card-body">
10                <table class="table table-hover ">
11                    <thead>
12                        <tr>
13                            <th>#ALUMNO</th>
14                            <th>NOMBRE</th>
15                            <th>ESCUELA</th>
16                            <th>FECHA INICIO</th>
17                            <th>TOTAL HORAS</th>
18                            <th>ACCIONES</th>
19                        </tr>
20                    </thead>
21                    <tbody>
22                        <tr>
23                            <td>
24                                <c:forEach var="dato" items="{lista}">
25                                    <td>${dato.idalumno}</td>
26                                    <td>${dato.nombre}</td>
27                                    <td>${dato.escuela}</td>
28                                    <td>${dato.fechal}</td>
29                                    <td>${dato.total}</td>
30                                </td>
31
```

Dentro de las cajas de texto se almacenará la información que viene dentro de la tabla llamada “persona” las columnas deben de ser como está establecido en la base de datos, previamente se creará un botón llamado “agregar” este botón se utilizará para agregar registros en la pantalla principal del sistema. Se ejecutará

el programa y nos mostrará el enlace web que será nuestra página principal, en ella se visualizarán las cajas de texto que contiene las columnas de la base de datos en este caso será la tabla llamada “persona”, la cual se visualizará enseguida:



Nuevo Registro

#ALUMNO	NOMBRE	ESCUELA	FECHA INICIO	TOTAL HORAS	ACCIONES
1	ernesto	UAEM	1995-01-29	100:00:00	Editar Delete
6	NANCY	UAEM	1995-01-29	100:00:00	Editar Delete

Figura 42: Interfaz principal del sistema (Rendón, 2022).

Completa la interfaz principal en donde se visualizan los datos introducidos por el usuario se podrá observar que el sistema cumple su función además de que es amigable con el usuario, además cuenta con un menú principal de botones fácil de manejar.

7.5. Agregar un registro

Como siguiente paso se crea el código necesario para agregar un registro en la base de datos y ésta se puede visualizar en el sistema, crearemos una nueva entidad llamada “persona.java” y se agregara un constructor vacío la cual almacenara la información del personal, crearemos los métodos “set” y “get” como se muestra de la siguiente manera:

```
1 package Entidad;
2 /*
3  * @author Ernesto
4  */
5 public class Persona {
6     int idalumno;
7     String nombre;
8     String escuela;
9     String fecha1;
10    String fecha2;
11    String hora1;
12    String hora2;
13    String total;
14
15    public Persona () {
16    }
17
18    public int getIdalumno () {
19        return idalumno;
20    }
21
22    public void setIdalumno(int idalumno) {
23        this.idalumno = idalumno;
24    }
25
26    public String getNombre () {
27        return nombre;
28    }
29
30    public void setNombre(String nombre) {
31        this.nombre = nombre;
32    }
33
34    public String getEscuela () {
35        return escuela;
```

```
36     }
37
38     public void setEscuela(String escuela) {
39         this.escuela = escuela;
40     }
41
42     public String getFechal() {
43         return fechal;
44     }
45
46     public void setFechal(String fechal) {
47         this.fechal = fechal;
48     }
49
50     public String getFecha2() {
51         return fecha2;
52     }
53
54     public void setFecha2(String fecha2) {
55         this.fecha2 = fecha2;
56     }
57
58     public String getHoral() {
59         return horal;
60     }
61
62     public void setHoral(String horal) {
63         this.horal = horal;
64     }
65
66     public String getHora2() {
67         return hora2;
68     }
69
70     public void setHora2(String hora2) {
71         this.hora2 = hora2;
72     }
73
74     public String getTotal() {
75         return total;
76     }
77
78     public void setTotal(String total) {
79         this.total = total;
80     }
81
82
83 }
```

Dentro de la carpeta “jsp” llevara un archivo llamado “agregar.jsp” el cual va a servir para para visualizar las etiquetas. Una vez ejecutado el programa se visualizará el formulario para que permita agregar registros nuevos de los usuarios que harán uso del sistema. La interfaz contará con 2 botones, agregar y eliminar, posteriormente se agregará un nuevo botón para poder editar algún registro de un usuario, se podrá visualizar en el siguiente código:

```
1
2
3     <body>
4         <div class="container mt-4 col-lg-4">
5             <div class="card border-info">
6                 <div class="card-header bg-info">
7                     <h4>Agregar Nuevo Registro</h4>
8                 </div>
9                 <div class="card-body">
10                    <form method="POST">
11                        <label>Nombres</label>
12                        <input type="text" name="nombre" class="form-control">
13                    <label >Escuela</label>
14                        <input type="text" name="escuela" class="form-control">
15                    <label >Fecha de ingreso</label>
16                        <input type="text" name="fecha1" class="form-control">
17                    <label >Fecha de fin</label>
18                        <input type="text" name="fecha2" class="form-control">
19                    <label >Hora entrada</label>
20                        <input type="text" name="hora1" class="form-control">
21                    <label >Hora salida</label>
22                        <input type="text" name="hora2" class="form-control">
23                    <label >Total de horas</label>
24                        <input type="text" name="total" class="form-control">
25
26                        <input type="submit" value="Agregar" class="btn btn-success">
27
28                    <a href="index.htm">Regresar</a>
29                </form>
30            </div>
31        </div>
32    </div>
33 </body>
34
35
```

A cada archivo “jsp” llevará el estilo de BOOTSTRAP para conservar la visualización de la herramienta, una vez ejecutado el programa como se explicó anteriormente se declara al inicio de un archivo. Después se trabajará el código dentro del controlador creando un método llamado “agregar”, el cual va a tener el código necesario para hacer un “insert” en SQL. Cada vez que se agrega una herramienta se debe de ir al archivo llamado “dispatcher” y agregar el método que se acabó de crear.

A continuación, se describe el código del método el cual ayudara agregar un registro en el sistema:

```
1 @RequestMapping (value ="agregar.htm", method = RequestMethod.GET)
2     public ModelAndView Agregar () {
3         mav.addObject (new Persona ());
4         mav.setViewName ("agregar");
5         return mav;
6     }
```

Esta parte del código explica cada uno de los atributos que tiene la tabla alumno, como se puede observar este código es la parte de MySQL para agregar un registro:

```
1 @RequestMapping (value ="agregar.htm", method = RequestMethod.POST)
2     public ModelAndView Agregar (Persona p) {
3     String sql="insert into alumno(nombre, escuela, fecha1, fecha2, hora1,
4     hora2, total) values (?, ?, ?, ?, ?, ?, ?)";
5     this.jdbcTemplate.update (sql,
6     p.getNombre (), p.getEscuela (), p.getFecha1 (),
7     p.getFecha2 (),
8     p.getHoral (), p.getHora2 (), p.getTotal ());
9     return new ModelAndView ("redirect:/index.htm");
```

7.6. Modificar un registro

Se trabajará en otro archivo llamado “editar.jsp” que servirá para editar un dato de un usuario y de esta manera el formulario editar estará llevando a cabo esa tarea, ya que trabaja con la información de la base de datos que previamente se contaba con un registro. Para ello se reutilizará el código de agregar ya que cuenta con la misma estructura sólo que en este caso se agregará el botón de actualizar y se mandará a llamar un dato ya existente. El código quedaría de la siguiente manera:

```
1 <form method="POST">
2 <label >Nombre</label>
3 <input type="text" name="nombre" class="form-control"
4 value="\${lista[0].Nombre}">
5
6 <label >Escuela</label>
7 <input type="text" name="escuela" class="form-control"
8 value="\${lista[0].Escuela}">
9
10 <label >Fecha de Ingreso</label>
11 <input type="text" name="fecha1" class="form-control"
12 value="\${lista[0].fecha1}">
13
14 <label >Fecha de salida</label>
15 <input type="text" name="fecha2" class="form-control"
16 value="\${lista[0].fecha2}">
17
18 <label >Hora de entrada</label>
19 <input type="text" name="hora1" class="form-control"
20 value="\${lista[0].hora1}">
21
22 <label >Hora de salida</label>
23 <input type="text" name="hora2" class="form-control"
24 value="\${lista[0].hora2}">
25
26 <label >total de horas</label>
27
28 <input type="text" name="total" class="form-control"
29 value="\${lista[0].total}">
30 <br>
31 <input type="submit" value="Actualizar" class="btn btn-success">
```

```
32 <a href="index.htm">Regresar</a>
33
34 </form>
```

Se trabajará en el archivo controlador para agregar el código para editar datos y se hace una consulta basándose en el “ID” aplicando un “select” haciendo referencia “ID” de cada registro, de esta manera ya estará completo nuestro formulario para editar datos.

7.7. Eliminar un registro

Para eliminar un registro de la base de datos sólo se concentrará en el controlador, ya no será necesario crear un formulario. Para ello debemos recibir los datos de la fila seleccionada en este caso el “ID”, se hará referencia a “delete.htm” para que capture el dato “idalumno” del registro el “index.jsp” se declarará el botón “borrar” de la siguiente manera:

```
<td>
1 <a href="editar.htm?id=${dato.idalumno}" class="btn btn-
2 warning">Editar</a>
3 <a href="delete.htm?id=${dato.idalumno}" class="btn btn-
4 danger">Delete</a>
</td>
```

El código para generar un botón “borrar” y otro llamado “editar” registros del sistema. Una vez ejecutado el programa se podrá visualizar el botón llamado “Editar” y “Delete”.

En el controlador se va a recibir el “ID” que se tiene declarada en la parte superior, en donde se ejecutará una consulta “MySQL” al ejecutar la consulta

“eliminar” se actualizará y al mismo tiempo se redireccionará a la página principal del proyecto. Para que funcione correctamente se debe de agregar un “prop delete” en el archivo “dispatcher.xml”.

```
1 @RequestMapping("delete.htm")
2     public ModelAndView Delete(HttpServletRequest request){
3         id=Integer.parseInt(request.getParameter("id"));
4         String sql="delete from alumno where idalumno="+id;
5         this.jdbcTemplate.update(sql);
6         return new ModelAndView("redirect:/index.htm");
```

Tomando en cuenta el archivo “dispatch.xml” el proyecto estará completo y se ejecutaran las cuatro funciones “CRUD” (Create, Read, Update y Delete) de la base de datos, por último, se guaran los cambios y se ejecutara el proyecto para comprobar su funcionalidad en su primer versión, la cual será útil para los alumno que realizaran su servicio social o prácticas profesionales. A continuacion se muestran los “index ” que utiliza el sistema:

```
1 <props>
2
3     <prop key="index.htm">controlador</prop>
4     <prop key="agregar.htm">controlador</prop>
5     <prop key="editar.htm">controlador</prop>
6     <prop key="delete.htm">controlador</prop>
7
8 </props>
```

Se agrega controladores para cada tipo de función que el sistema va a llevar a cabo, como el sistema tiene cuatro funciones principales que ayudarán al usuario a interactuar con el sistema fácilmente.

7.8. Diagrama de casos de uso

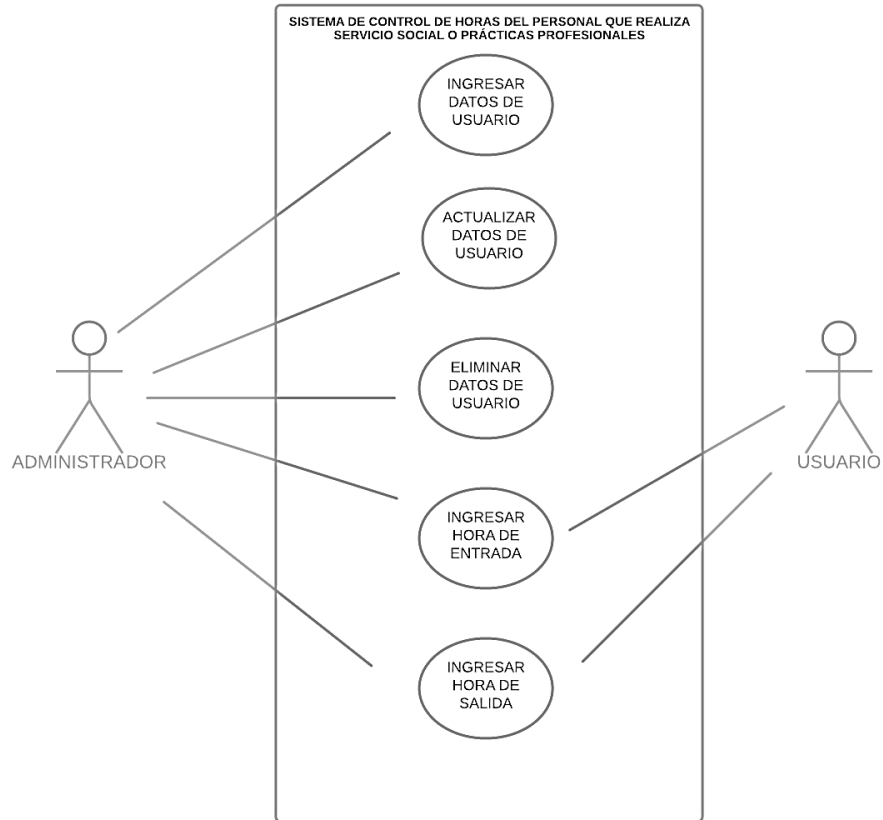


Figura 43: Diagrama de casos de usos del sistema (Rendón, 2022).

En la imagen anterior se representa con el diagrama casos de uso las interacciones que tendrá el sistema con el usuario y con el administrador, cada una de las tareas que llevará a cabo el sistema estarán representados por un círculo, el cual se señala la interacción con los actores, cabe mencionar que el sistema estará encerrado en un rectángulo que define el alcance del sistema. Este diagrama es una forma de representar gráficamente los roles que maneja cada uno de los objetos.

7.9. Diseño del sistema

El diseño del sistema se llevará a cabo cómo la representa la siguiente imagen, a través de un usuario, el sistema interpretará si es una consulta, una inserción de datos, una actualización y la eliminación un dato. En cada una de las tareas al finalizar el proceso deberá mostrar la interfaz con el dato modificado.

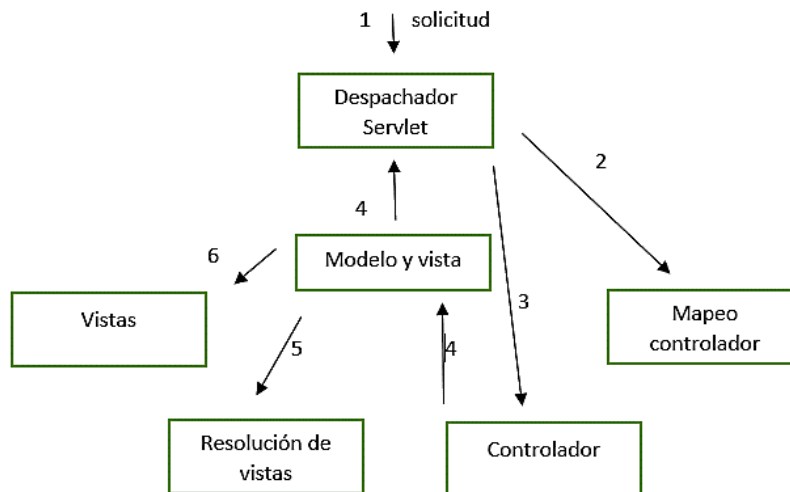


Figura 44: Modelo vista controlador del sistema (Rendón, 2022).

7.10. Arquitectura del sistema

Para el desarrollo del presente sistema se utilizará la arquitectura multicapa, en este caso aplica de 3 capas en donde se distribuirán los elementos en una capa de acceso de datos, en una capa de gestión remota de datos y servidor de aplicaciones. Dentro del entorno de desarrollo esta es una de las más utilizadas

en la actualidad. Existen diferentes arquitecturas basadas en capas ya que pueden ser abiertas o cerradas, el sistema a desarrollar será de capa cerrada ya que se reduce la dependencia entre las capas y ofrece un acoplamiento útil. Lo que se busca es afectar lo menos posible una capa al a efectuar un cambio entre capas.

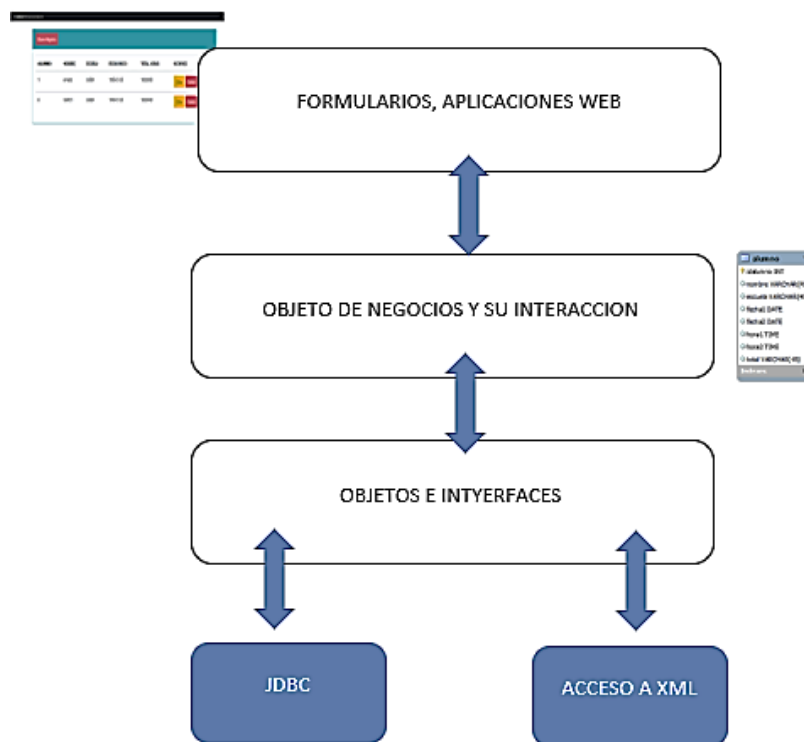


Figura 45: Arquitectura del sistema (Rendón, 2022).

7.11. Modelo de base de datos

La siguiente imagen muestra los atributos que tendrá el sistema para obtener información acerca de los usuarios, cada uno de ellos será útil para mantener un registro específico del usuario que realizará sus prácticas profesionales o

estancia de servicio social. Por lo anterior se realizó un estudio el cual se indica por parte del usuario algunos datos necesarios que ellos necesitaban para llevar el registro correcto.

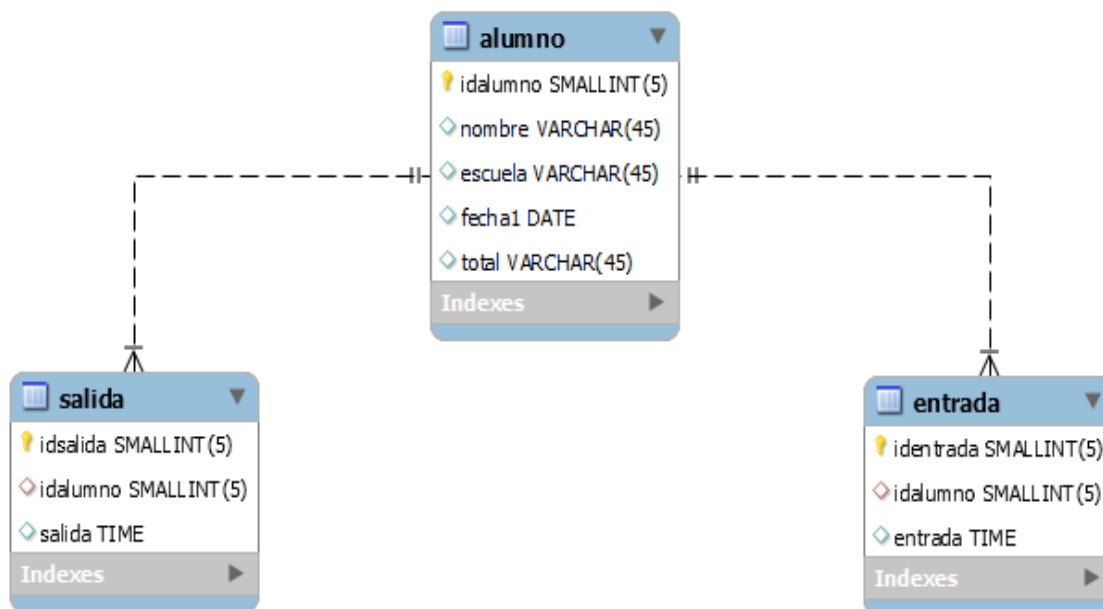


Figura 46: Modelo de base de datos (Rendón, 2022).

La tabla principal está conformada por una llave única la cual será útil para identificar a los usuarios, sus datos personales, como el nombre, la escuela de donde provienen, fecha de entrada y por último las tablas de la hora en la que ingresan y salida de los usuarios.

7.12. Tabla alumno

En la siguiente imagen se describe el tipo de dato, el "ID" será de tipo entero el cual servirá como un dato único que identificará a un usuario, en nombre y escuela estará definido por carácter de cualquier tipo de no más de cuarenta y cinco caracteres, fecha1 se especificará como "date" y su formato será el siguiente "DD/MM/AAAA" y admitirá otros datos que no respeten el formato y por último la hora de igual manera tendrá un formato único el cual sino es insertado respetando el formato el sistema mandará un mensaje de error, el formato sería el siguiente "00:00:00" estará definido por hora, minutos y segundos. Por ultimo se muestra el total, aquí se almacenará el total de horas de los usuarios.

```
MariaDB [prueba]> describe alumno;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idalumno | smallint(5) | NO   | PRI | NULL    |       |
| nombre   | varchar(45) | YES  |     | NULL    |       |
| escuela  | varchar(45) | YES  |     | NULL    |       |
| fecha1   | date       | YES  |     | NULL    |       |
| total    | varchar(45) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.29 sec)
```

Figura 47: Tabla alumno con atributos (Rendón, 2022).

7.12.1 Tabla entrada

En la tabla entrada se muestran los atributos en donde se va a almacenar el dato tipo "Time" el cual hace referencia a la hora de entrada del usuario.

```
MariaDB [prueba]> describe entrada;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| identrada  | smallint(5)   | NO   | PRI | NULL    |      |
| idalumno   | smallint(5)   | YES  | MUL | NULL    |      |
| entrada    | time          | YES  |     | NULL    |      |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)
```

Figura 48: Tabla entrada con atributos (Rendón, 2022).

7.12.2 Tabla salida

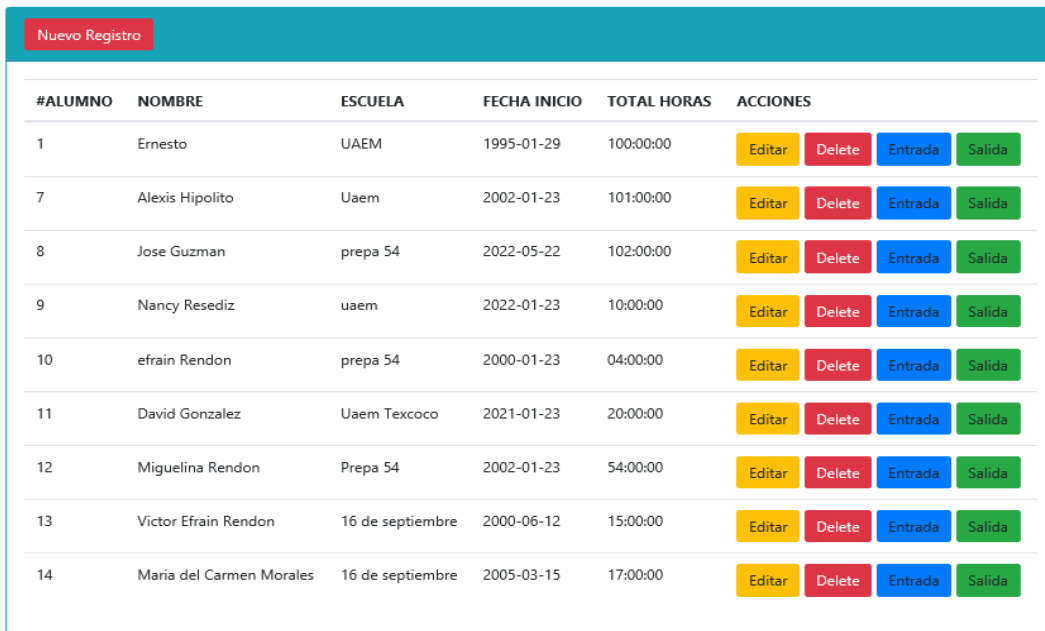
En la tabla salida se muestran los atributos en donde se va a almacenar el dato tipo "Time" el cual hace referencia a la hora de salida del usuario, ambas tablas están relacionadas con la tabla alumno.

```
MariaDB [prueba]> describe salida;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idsalida   | smallint(5)   | NO   | PRI | NULL    |      |
| idalumno   | smallint(5)   | YES  | MUL | NULL    |      |
| salida     | time          | YES  |     | NULL    |      |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)
```

Figura 49: Tabla salida con atributos (Rendón, 2022).

7.13. Vista de la página principal

En la siguiente imagen se muestra la interfaz principal del proyecto, la cual cuenta con un botón en donde se puede agregar nuevos alumnos. Además de visualizar la información del contacto desde su nombre, la escuela de procedencia y la fecha de inicio en la que el alumno inicia su estadía en la institución, el sistema tiene la función demostrar la fecha que el alumno lleva desde su inicio, se cuenta con un apartado llamado acciones que tiene 2 botones editar y eliminar, cada uno de los botones cumple una función dentro del sistema, acceder a algún dato y poder modificarlo y el botón borrar en caso de que el administrador cometa un error o que la información se requiera eliminar sólo basta con seleccionar dicho botón para que el registro en la base de datos sea eliminado.

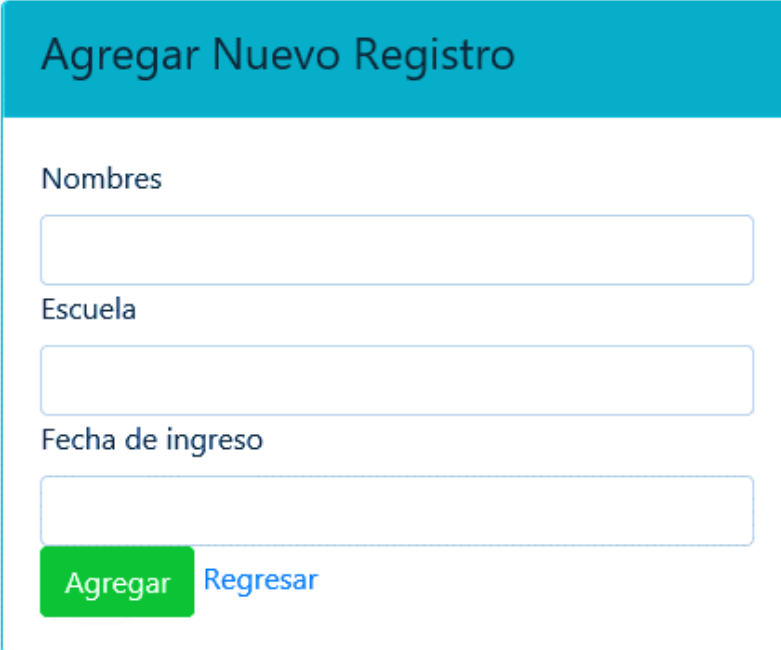


#ALUMNO	NOMBRE	ESCUELA	FECHA INICIO	TOTAL HORAS	ACCIONES
1	Ernesto	UAEM	1995-01-29	100:00:00	Editar Delete Entrada Salida
7	Alexis Hipolito	Uaem	2002-01-23	101:00:00	Editar Delete Entrada Salida
8	Jose Guzman	prepa 54	2022-05-22	102:00:00	Editar Delete Entrada Salida
9	Nancy Resediz	uaem	2022-01-23	10:00:00	Editar Delete Entrada Salida
10	efrain Rendon	prepa 54	2000-01-23	04:00:00	Editar Delete Entrada Salida
11	David Gonzalez	Uaem Texcoco	2021-01-23	20:00:00	Editar Delete Entrada Salida
12	Miguelina Rendon	Prepa 54	2002-01-23	54:00:00	Editar Delete Entrada Salida
13	Victor Efrain Rendon	16 de septiembre	2000-06-12	15:00:00	Editar Delete Entrada Salida
14	Maria del Carmen Morales	16 de septiembre	2005-03-15	17:00:00	Editar Delete Entrada Salida

Figura 50: Interfaz principal (Rendón, 2022).

7.14. Vistas

En las siguientes imágenes el sistema muestra cómo son las interfaces de agregar registros a la base de datos, es importante respetar los atributos del sistema ya que si se coloca un dato de tipo diferente el sistema marcará error y no podrá guardar la información, se especifica qué carácter reconoce el sistema para que el usuario no cometa error alguno y la inserción de datos se puede llevar con éxito al almacenamiento al servidor para poder visualizarlo en alguna otra consulta que el administrador desee.



Interfaz de usuario para agregar un nuevo registro. El formulario tiene un encabezado azul con el título "Agregar Nuevo Registro". Contiene tres campos de entrada de texto etiquetados "Nombres", "Escuela" y "Fecha de ingreso". Debajo de los campos hay dos botones: "Agregar" (verde) y "Regresar" (azul).

Figura 51: Interfaz para agregar registro (Rendón, 2022).

The image shows a web interface for updating a record. It features a teal header with the title 'Actualizar Registro'. Below the header, there are three input fields: 'Nombre' with the value 'Ernesto', 'Escuela' with the value 'UAEM', and 'Fecha de Ingreso' with the value '1995-01-29'. At the bottom, there is a green button labeled 'Actualizar' and a blue text link labeled 'Regresar'.

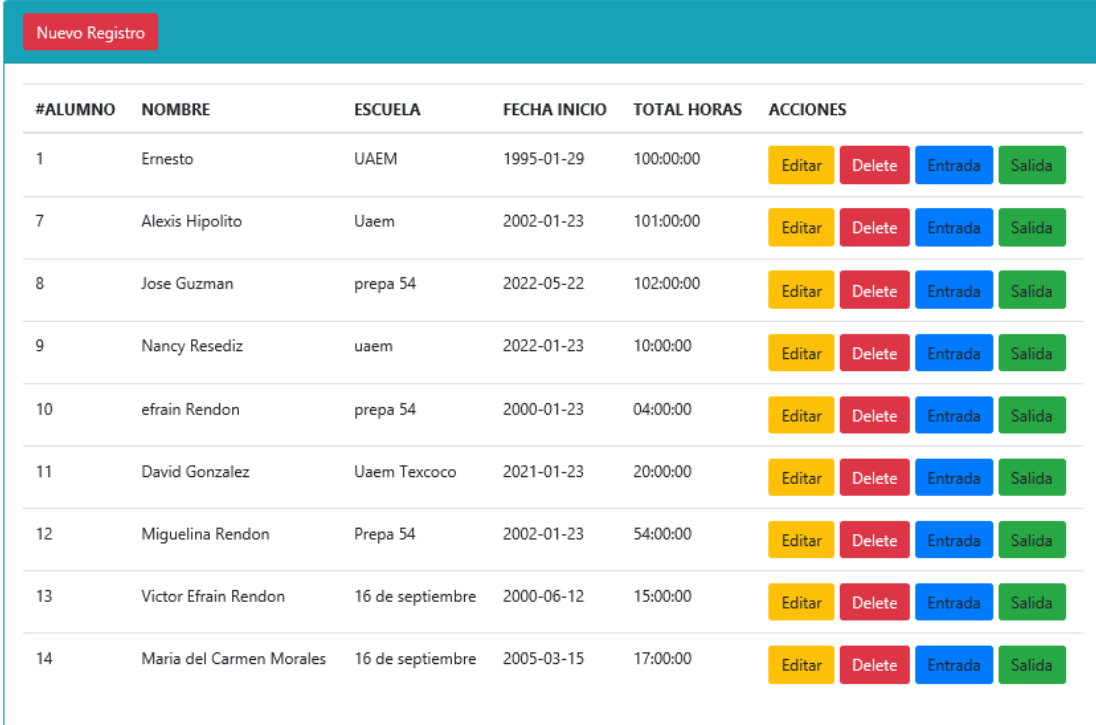
Figura 52: Interfaz para actualizar registro (Rendón, 2022).

Como dato adicional se muestra un ejemplo de los datos que son introducidos por el administrador y son aceptados dentro del sistema y se pueda guardar correctamente la información.

7.15. Manual del sistema

Por último, se agrega un manual de usuario, en dónde se describen las funciones de los botones, la visualización de las interfaces y la forma correcta de cómo agregar, editar y eliminar un registro.

En la siguiente figura se muestra la interfaz principal con usuarios en el sistema, como el sistema aún no está instalado en un servidor web, se tendrá que ejecutar desde la aplicación net beans.



#ALUMNO	NOMBRE	ESCUELA	FECHA INICIO	TOTAL HORAS	ACCIONES
1	Ernesto	UAEM	1995-01-29	100:00:00	Editar Delete Entrada Salida
7	Alexis Hipolito	Uaem	2002-01-23	101:00:00	Editar Delete Entrada Salida
8	Jose Guzman	prepa 54	2022-05-22	102:00:00	Editar Delete Entrada Salida
9	Nancy Resediz	uaem	2022-01-23	10:00:00	Editar Delete Entrada Salida
10	efrain Rendon	prepa 54	2000-01-23	04:00:00	Editar Delete Entrada Salida
11	David Gonzalez	Uaem Texcoco	2021-01-23	20:00:00	Editar Delete Entrada Salida
12	Miguelina Rendon	Prepa 54	2002-01-23	54:00:00	Editar Delete Entrada Salida
13	Victor Efrain Rendon	16 de septiembre	2000-06-12	15:00:00	Editar Delete Entrada Salida
14	Maria del Carmen Morales	16 de septiembre	2005-03-15	17:00:00	Editar Delete Entrada Salida

Figura 53: Interfaz principal con personal registrado (Rendón, 2022).

7.15.1 Agregar usuario

Para agregar un registro nuevo en el sistema hay que aplicar el botón que se muestra en la siguiente imagen:



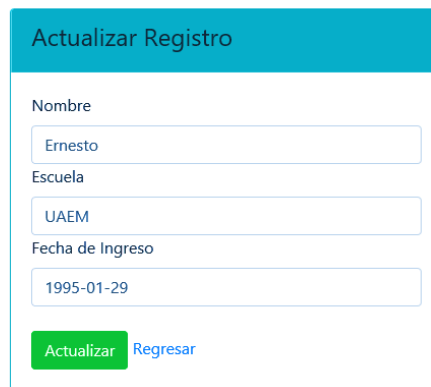
Figura 54: Botón agregar registros (Rendón, 2022).

Una vez seleccionado el botón, el sistema enviará la interfaz para agregar un registro nuevo, los siguientes campos son nombres, escuela, fecha de ingreso, fecha de fin, hora de entrada, hora de salida y por último el campo total de horas. En el campo total de horas se registra el total de horas del personal.

A screenshot of a web form titled "Agregar Nuevo Registro" in a teal header. The form contains three input fields: "Nombres", "Escuela", and "Fecha de ingreso". At the bottom, there are two buttons: a green "Agregar" button and a blue "Regresar" button.

Figura 55: Interfaz agregar registros (Rendón, 2022).

A continuación, se muestra la imagen con los datos insertados correctamente:



Actualizar Registro

Nombre
Ernesto

Escuela
UAEM

Fecha de Ingreso
1995-01-29

Actualizar Regresar

Figura 56: Campos llenados correctamente (Rendón, 2022).

7.15.2 Editar un registro

El sistema cuenta con un apartado "acciones", después se visualiza el botón con la leyenda "editar", este permite corregir un registro en caso de que haya alguna corrección de algún dato de usuario.



Figura 57: Botón Editar (Rendón, 2022).

Para actualizar un registro es necesario ubicarse en el renglón de el usuario que se va a editar para que el administrador pueda hacer la corrección de los datos, aplicando este botón el sistema redireccionará la siguiente interfaz como lo muestra la imagen:



Actualizar Registro

Nombre
Ernesto

Escuela
UAEM

Fecha de Ingreso
1995-01-29

Actualizar [Regresar](#)

Figura 58: Interfaz actualizar registros (Rendón, 2022).

El administrador tendrá que ubicar el cursor en la caja de texto que desea actualizar y basta con apretar el botón verde con la leyenda “actualizar”, el sistema guardará automáticamente el registro actualizado y redireccionará al menú principal del sistema con el dato modificado.

7.15.3 Eliminar un registro

Por último, eliminar un registro de igual manera ir al apartado de acciones y seleccionar el botón llamado “Delete”, esta acción permite eliminar un dato de manera permanente a menos que se haya llevado a cabo un respaldo de la base de datos, de lo contrario el sistema eliminará de forma permanente el dato y no habrá forma de recuperarlo, por ello es recomendable ser cuidadoso y llevar respaldos planeados. A continuación, se muestra el botón:

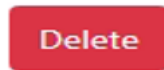


Figura 59: Eliminar registros (Rendón, 2022).

Una vez presionado el botón “Delete” el sistema borrará de forma permanente el registro y redireccionará en automático a la interfaz principal en donde ya no estará el registro que se eliminó.

7.15.4 Registrar entrada

Con este botón el usuario puede registrar su hora de entrada al sistema, solo es necesario presionar el botón y el sistema hará su registro para que sea reflejado en la interfaz.

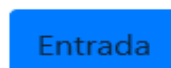


Figura 60: Botón para registrar entrada (Rendón, 2022).

7.15.5 Registrar salida

Este botón registra la salida del usuario en el sistema, de igual manera lo registra en automático como en el botón de entrada, solo que este almacenará la hora de salida del personal.

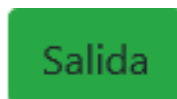


Figura 61: Botón para registrar salida (Rendón, 2022).

VIII. Conclusiones y sugerencias

8.1. Conclusiones

- Contar con un sistema de información que verifique las horas de los alumnos que realizaron mediante su servicio social o estancias profesionales es muy importante, se puede ahorrar tiempo, material y mejor organización.
- Gracias a la tecnología es posible optimizar, administrar la información de alguna institución.
- Los alumnos tendrán un mejor control al momento de visualizar sus horas sin necesidad de recurrir a las listas de papel.
- El único personal autorizado que tiene acceso a la información será el administrador que lleva a cabo ese rol de registro de alumnos, ya que por ningún motivo el alumno puede modificar sus registros.
- Considerar el uso de la tecnología para mejorar algún proceso, en el cual pueda ser tardado ya que este proporcionará una manera óptima y eficaz de controlar la información.

8.2. Aportaciones

- Otorgar la información de manera adecuada a los alumnos para un mejor servicio por parte de la institución.
- El sistema cumple su función principal y genera una satisfacción hacia las personas que la utilicen.
- A través de un sistema se puede realizar una presentación digital de datos y asimismo facilitar el manejo de la información. Anteriormente se utilizaban otros medios administrativos.

8.3. Sugerencias

- Se recomienda al usuario tener actualizado el navegador web, ya que la aplicación depende de ella y su arquitectura del sistema que así lo demanda para que pueda tener un mejor desempeño al navegar o al interactuar con el sistema.
- La aplicación se ejecuta a través de un navegador por el cual no será necesario tener algún sistema operativo necesario ya que esta aplicación se maneja en la web.

8.4. Trabajo a futuro

Este sistema se puede adaptar a otros sistemas como un módulo aparte, hay empresas que necesitan que sus empleados lleguen puntualmente a la hora de entrada del trabajo, ya sean en fábricas, restaurantes, entre otros tipos de trabajo. Este sistema apoya a visualizar las horas de entrada y salida, haciendo unas modificaciones al sistema. Originalmente el sistema está adaptado para contabilizar horas de los usuarios.

Como ejemplo tenemos la siguiente imagen, en donde los usuarios acceden a un sistema con usuario y contraseña, pero antes de entrar al sistema deberán realizar su hora entrada y en la parte de abajo viene la opción de reloj checador. Al momento que el usuario seleccione “reloj checador” se redireccionará a la página de inicio del sistema que se desarrolló anteriormente.

CONTRASEÑA

7	8	9	Enter
4	5	6	
1	2	3	CLR
0			

RELOJ CHECADOR

Figura 62: Ejemplo de sistemas que puedan implementar un reloj checador (Rendón, 2022).

Se plantea como futuro nuevos módulos, los cuales servirán para tener un mejor control de información de usuarios. A continuación, se enlistan algunas mejoras que podría tener el sistema a futuro:

- Botón en donde pueda emitir reportes de los usuarios.
- Botón para hacer búsquedas directas de los usuarios.
- Botón para salir del sistema.
- Botón para bloquear sistema.
- Agregar una interfaz para ingresar al sistema Con usuario y contraseña.

IX. Referencias de consulta

Aguirre, S. (2021). FRAMEWORK TOTAL - Vol.1: Crea APPs desde Cero con Laravel + Booststrap + MySQL. Buenos Aires: Plandos.

Álvarez, C. (2021). Spring MVC Configuración. Junio 08,2022, de Arquitecturajava Sitio web: <https://www.arquitecturajava.com/spring-mvc-configuracion/#:~:text=Spring%20MVC%20es%20quiz%C3%A1s%20el,o%20Vue%20a%20la%20hora.>

Amaya, L. (2017). ARQUITECTURA SISTEMA INFORMACION. Mayo 28,2022, de luisamayateacher Sitio web: <https://sites.google.com/site/luisamayateacher/arquitectura-sistema-informacion.>

Bascón E. (2004). El patrón de diseño Modelo-Vista-Controlador (MVC) y su implementación en Java Swing. marzo 12, 2022, de SCIELO Sitio web: http://www.scielo.org.bo/scielo.php?script=sci_arttext&pid=S1683-0789200400010.

Berzal, F., Cortijo, F., & Cubero, J. (2005). Desarrollo Profesional de Aplicaciones Web con ASP.NET. abril 24, 2022, de .net Sitio web: [http://elvex.ugr.es/decsai/csharp/aspnet.html.](http://elvex.ugr.es/decsai/csharp/aspnet.html)

Booch, G., Rumbaugh, J., & jacobson, I. (2006). El lenguaje unificado de modelado. Madrid: PEARSON EDUCACION.

CakePHP. (2022). Entendiendo Modelo-Vista-Controlador. Mayo 08, 2022, de Cake Software Foundation, Inc Sitio web: <https://book.cakephp.org/1.3/es/The-Manual/Beginning-With-CakePHP/Understanding-Model-View-Controller.html>.

Cedano, M., Cedano, A., Rubio, J., Vega, A. (2014). Fundamentos de computación para ingenieros. México: Grupo Editorial Patria.

Cervantes Ojeda, J., & Gómez Fuentes, María del Carmen (2012). Taxonomía de los modelos y metodologías de desarrollo de software más utilizados. Universidades, (52),37-47.[fecha de Consulta 11 de Septiembre de 2020]. ISSN: 0041-8935. Disponible en: <https://www.redalyc.org/articulo.oa?id=373/37326902005>.

Cervantes, J., Gómez, M., González, P., & García, A. (2016). Introducción a la programación orientada a objetos. México: UAM, Unidad Cuajimalpa.

Fernández, V. (2006). Desarrollo de sistemas de información: Una metodología basada en el modelado. Universidad Politécnica de Catalunya: Edicions upc.

Gacitúa Bustos, Ricardo A. (2003). Métodos de desarrollo de software: El desafío pendiente de la estandarización. Metodologías de desarrollo de software: un duelo pendiente de estandarización. Theoria, 12 (1), 23-42. [Fecha de Consulta 9 de agosto de 2020].

García C. (2014). Sql Facil. España: Alfaomega Grupo Editor.

Gilfillan I. (2003). La Biblia MySQL. España: Anaya Multimedia.

Gómez, C., Cervantes, J., & González, P. (2019). Fundamentos de Ingeniería de Software. MEXICO: Universidad Autónoma Metropolitana.

Gustavo., B. (2022). ¿Qué es CSS?. Mayo 07,2022, de Hostinger Tutoriales Sitio web: <https://www.hostinger.mx/tutoriales/que-es-css>.

Kendall, K., & Kendall, J. (2011). Análisis y diseño de sistemas. México: PEARSON EDUCACIÓN.

Maida, EG, Pacienza, J. (2015). Metodologías de desarrollo de software. Argentina: Universidad Católica Argentina.

MariaDB. (2022). MariaDB Server: The open source relational database. Junio 06,2022, de MariaDB Foundation Sitio web: <https://mariadb.org/>.

Mateu, C. (2004). DESARROLLO DE APLICACIONES WEB. Barcelona: Fundación para la Universitat Oberta de Catalunya.

Oppel A., & Sheldon, R. (2010). Fundamentos de SQL. México: MCGRAW-HILL.

Oracle Corporation. (2022). Manual de referencia de mysql 8.0. Enero 15 2021, de MySQL Sitio web: <https://dev.mysql.com/doc/refman/8.0/en/>.

Ortega, Dinarle, & Guevara, María, & Benavides, John (2016). ELEMENTARY: UN FRAMEWORK DE PROGRAMACIÓN WEB. *Télématique*, 15(2),144-171.[fecha de Consulta 11 de Septiembre de 2020]. ISSN: 1856-4194. Disponible en: <https://www.redalyc.org/articulo.oa?id=784/78457627004>.

Paul Deitel, H. D. (2008). *Cómo programar en C++*. México: Pearson educación.

Pérez, E. (2016). *Desarrollo de aplicaciones mediante el framework de spring*. Bogotá: Editorial Ra-Ma.

Pressman, R. S. (2002). *INGENIERÍA DEL SOFTWARE UN ENFOQUE PRÁCTICO*. España: McGrawHill.

Ramos, A., & Ramos,J. (2007). *Operaciones con bases de datos ofimáticas y corporativas*. Madrid, España.: Paraninfo.

Rivera, D. (2020). *Introducción a la Programación Orientada a Objetos*. abril 22, 2022, de Pleets Blog Sitio web: <https://blog.pleets.org/article/introducci%C3%B3n-a-la-programaci%C3%B3n-orientada-a-objetos>.

Rockcontent. (2020). *Bootstrap: guía para principiantes de qué es, por qué y cómo usarlo*. Marzo 11, 2022, de Rock Content Sitio web: <https://rockcontent.com/es/blog/bootstrap/>.

Sommerville, I. (2005). Ingeniería de Software Séptima edición. Madrid: Pearson Addison Wesley.