



Universidad Autónoma del Estado de México

Centro Universitario UAEM Valle de Chalco

**DESARROLLO DE UN SISTEMA PARA EL CONTROL DE
ENTRADAS, SALIDAS E INVENTARIO DEL TALLER DE DISEÑO
INDUSTRIAL DEL CU UAEM VALLE DE CHALCO
IMPLEMENTANDO CÓDIGO QR, 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

ROSENDO VEGA MATEOS

ASESOR:

MTRO. EN C.C. RICARDO JAVIER BUCIO LÓPEZ

REVISORA: DRA. EN C. DE LA COMP. MARÍA DE LOURDES LÓPEZ GARCÍA

REVISOR: MTRO. EN C.C. JOSÉ ERNESTO LUNA DOMÍNGUEZ

VALLE DE CHALCO SOLIDARIDAD, MÉXICO MARZO 2023



CUVCH

**DESARROLLO DE UN SISTEMA PARA EL CONTROL DE
ENTRADAS, SALIDAS E INVENTARIO DEL TALLER DE
DISEÑO INDUSTRIAL DEL CU UAEM VALLE DE
CHALCO IMPLEMENTANDO CÓDIGO QR, SPRING MVC
Y BOOTSTRAP**

ÍNDICE

I. RESUMEN	11
II. ANTECEDENTES	13
2.1 Justificación	16
III. IMPORTANCIA DEL PROBLEMA	17
IV. PLANTEAMIENTO DEL PROBLEMA	18
4.1. Pregunta de investigación.....	19
V. MARCO TEÓRICO	20
5.1. Ingeniería de Software.....	20
5.2. Programación	20
5.3. Paradigmas de programación	21
5.3.1. Programación imperativa	21
5.3.2. Programación estructurada.....	22
5.3.3. Programación procedimental	22
5.3.4. Programación modular.....	22
5.3.5. Programación declarativa	22
5.3.6. Programación lógica	23
5.3.7. Programación funcional	23
5.4. Lenguaje de programación	23
5.4.1. Bajo nivel.....	24
5.4.2. Medio nivel	24
5.4.3. Alto nivel.....	24
5.5. Programación orientada a objetos	25
5.5.1. Abstractar.....	25
5.5.2. Encapsular	25
5.5.3. Modularizar	26
5.5.4. Jerarquizar	26
5.5.5. Tipo	26
5.5.6. Concurrencia.....	27
5.5.7. Persistencia.....	27
5.6. Aplicaciones web	27
5.6.1. Página web	28
5.6.2. Sistema web.....	28
5.7. Desarrollo web.....	28
5.7.1. Front-end.....	29

5.7.2. Back -end	29
5.7.3. HTML	29
5.7.4. JavaScript	30
5.7.5. CSS.....	30
5.7.6. PHP	32
5.7.7. Java.....	32
5.7.8. Python	33
5.8. Base de datos.....	33
5.8.1. Estáticas.....	34
5.8.2. Dinámicas	34
5.9. Modelo de bases de datos	34
5.9.1. Jerárquicas.....	35
5.9.2. De red	35
5.9.3. Relacionales.....	36
5.9.4. Transaccionales	37
5.9.5. Multidimensionales.....	37
5.9.6. Orientadas a objetos	38
5.10. Sistema gestor de base de datos.....	38
5.10.1. Funciones.....	39
5.10.1.1. Definición de los datos	39
5.10.1.2. Manipulación de los datos.....	39
5.10.1.3. Seguridad e integración de los datos	39
5.10.1.4. Recuperación y restauración de los datos	40
5.10.2. Relacionales.....	40
5.10.2.1. MySQL.....	41
5.10.2.2. MariaDB.....	41
5.10.2.3. PostgreSQL.....	42
5.10.3. No relacionales	42
5.10.3.1. MongoDB.....	43
5.10.3.2. Redis	43
5.10.3.3. Cassandra	44
5.11. Servidor web.....	44
5.11.1. Servidor local	44
5.12. Modelo cliente-servidor	45
5.13. Modelo-Vista-Controlador	45
5.14. Framework.....	47
5.14.1. Framework Spring MVC.....	47

5.15. Bootstrap	47
5.16. Código QR.....	48
5.17. Modelado de los requerimientos	49
5.18. Lenguaje Unificado de Modelado.....	49
5.18.1. Diagrama de caso de uso	49
5.18.2. Diagrama de secuencia	50
5.18.3. Modelo entidad-relación.....	51
5.18.4. Diagrama de actividades	53
5.18.5. Diagrama de flujo de datos	54
VI. MÉTODOS Y TÉCNICAS DE INVESTIGACIÓN	55
VII. PRESENTACIÓN Y DISCUSIÓN DE RESULTADOS	59
7.1. Análisis de los requerimientos	59
7.1.1. Requerimientos funcionales.....	59
7.1.2. Requerimientos no funcionales.....	60
7.2. Herramientas y tecnologías a utilizar	60
7.3. Caso de uso	61
7.3.1. Registrar usuario.....	62
7.3.2. Registrar entrada	64
7.3.3. Registrar salida	66
7.3.4. Escanear código QR	68
7.3.5. Visualizar registros.....	69
7.4. Diseño del sistema	71
7.4.1. Modelo de la base de datos.....	71
7.4.2. Descripción de las entidades	73
7.4.2.1. Entidad Usuario.....	73
7.4.2.2. Entidad RegEntradaSalida	74
7.4.2.3. Entidad Taller	75
7.4.2.4. Entidad Inventario	75
7.4.3. Diagrama de secuencia	76
7.4.3.1. Registro	76
7.4.3.2. Registrar entrada.....	77
7.4.3.3. Registrar salida	78
7.4.3.4. Ver registros	79
7.4.3.5. Escanear código QR	80
7.4.4. Diagrama de actividades	81
7.4.4.1. Descargar registros	81
7.4.4.2. Agregar al inventario	82

7.4.4.3. Alerta de registro	83
7.4.5. Diagrama de flujo de datos	84
7.4.5.1. Diagrama padre.....	84
7.4.5.2. Diagrama superior.....	84
7.4.6. Diseño de las vistas	85
7.4.6.1. Vista principal	85
7.4.6.2. Registrar usuario	86
7.4.6.3. Registrar entrada.....	87
7.4.6.4. Registrar salida	87
7.4.6.5. Inicio de sesión del administrador	88
7.4.6.6. Vista administrador.....	89
7.5. Implementación (Desarrollo del sistema)	90
7.5.1. Modelo	91
7.5.1.1. Base de datos	91
7.5.1.2. Clases.....	92
7.5.1.3. Conexión con la base de datos	95
7.5.2. Vista	96
7.5.2.1. Principal.....	96
7.5.2.2. ¿Qué somos?	97
7.5.2.3. Agregar usuario.....	97
7.5.2.4. Registra tu entrada.....	98
7.5.2.5. Registra tu salida.....	99
7.5.2.6. Iniciar sesión como administrador.....	100
7.5.2.7. Administrador	101
7.5.2.8. Descargar registros	101
7.5.2.9. Teléfono celular.....	102
7.5.2.10. Código QR.....	103
7.5.3. Controlador	104
7.5.3.1. JSP.....	104
7.5.3.2. Paquete controlador	105
7.6. Verificación	110
7.6.1. Lanzamiento.....	111
7.7. Mantenimiento	116
VIII. CONCLUSIONES Y SUGERENCIAS.	117
8.1. Conclusiones	117
8.2. Sugerencias.....	117
IX. REFERENCIAS	118

ÍNDICE DE FIGURAS

1. Bitácora de ingreso a los talleres	18
2. Paradigmas de programación	21
3. Base de datos jerárquica	35
4. Base de datos de red	36
5. Base de datos relacional.....	36
6. Base de datos transaccional.....	37
7. Base de datos multidimensional	37
8. Base de datos orientada a objetos.....	38
9. Logo de MYSQL	41
10. Logo de MariaDB.....	41
11. Logo de PostgreSQL	42
12. Logo de MongoDB.....	43
13. Logo de Redis.....	43
14. Logo de Cassandra.....	44
15. Modelo cliente-servidor.....	45
16. Funcionamiento de MVC.....	46
17. Código QR	48
18. Diagrama de caso de uso	50
19. Diagrama de secuencia	51
20. Simbología modelo entidad-relación.....	52
21. Modelo entidad-relación.....	52
22. Diagrama de actividades.....	53
23. Diagrama de flujo de datos	54
24. Metodología de cascada	57
25. Diagrama de caso de uso	62
26. Diagrama de caso de uso (registro).....	64
27. Diagrama de caso de uso (registrar entrada).....	66
28. Diagrama de caso de uso (registrar salida)	67
29. Diagrama de caso de uso (escanear código QR)	69
30. Diagrama de caso de uso (ver registros).....	71
31. Diagrama entidad relación	72

32. Entidad Usuario	73
33. Entidad RegEntradaSalida.....	74
34. Entidad Taller.....	75
35. Entidad Inventario	76
36. Diagrama de secuencia (registro).....	76
37. Diagrama de secuencia (registrar entrada).....	77
38. Diagrama de secuencia (registrar salida).....	78
39. Diagrama de secuencia (ver registros)	79
40. Diagrama de secuencia (escanear código QR)	80
41. Diagrama de actividades (descargar registros).....	81
42. Diagrama de actividades (registrar inventario).....	82
43. Diagrama de actividades (alerta de registro)	83
44. Diagrama de flujo de datos (padre).....	84
45. Diagrama de flujo de datos superior	85
46. Maquetación de la vista principal	86
47. Maquetación de la vista registrar alumno	86
48. Maquetación de la vista registrar entrada	87
49. Maquetación de la vista registrar salida.....	87
50. Maquetación de la vista registrar observaciones	88
51. Maquetación de la vista iniciar sesión.....	88
52. Maquetación de la vista sesión iniciada.....	89
53. Maquetación de la vista ver lista de registros	89
54. Maquetación de la vista editar registro	90
55. Maria DB 10.6.4	91
56. Código SQL para la creación de la base de datos.....	92
57. Creación de entidades	93
58. Métodos getter y setter	94
59. Conexión con la base de datos.....	95
60. Vista principal.....	96
61. Vista ¿qué somos?	97
62. Vistas al agregar un usuario	98
63. Vistas del registro de entrada	99
64. Vistas del registro de salida	100

65. Vistas al Iniciar sesión como administrador	100
66. Vistas al agregar, editar y eliminar registros	101
67. Vistas y formatos para descargar	102
68. Vistas del funcionamiento desde un teléfono celular	103
69. Vista al escanear el código QR	103
70. Código para mostrar QR en la lista del inventario.....	104
71. Código de alerta al eliminar un registro del inventario	105
72. Creación del paquete controlador	106
73. Código del controlador para mostrar la vista.....	106
74. Código del controlador para registrar una entrada.....	107
75. Código del controlador para listar registros	108
76. Código del controlador para ver y editar registros	108
77. Código del controlador para eliminar registros.....	109
78. Código del controlador para la vista y registro de salida.....	109
79. Verificación de las sugerencias al sistema.....	111
80. Funcionamiento del sistema web en los talleres de diseño industrial ...	112
81. Alumnos y profesores haciendo uso del sistema web	112
82. Escaneo de código QR de una maquina	113
83. Sistema web en teléfono celular	114
84. Registros de usuarios en el sistema web.....	114
85. Registros de entradas y salidas en el sistema web	115
86. Registros del inventario en el sistema web	115

ÍNDICE DE TABLAS

1. Niveles de CSS.....	31
2. Comandos SQL	40
3. Historieta de caso de uso (registro)	63
4. Historieta de caso de uso (registrar entrada)	65
5. Historieta de caso de uso (registrar salida).....	67
6. Historieta de caso de uso (Escanear código QR)	68
7. Historieta de caso de uso (ver registros)	70

I. RESUMEN

Los talleres de diseño industrial de la Universidad Autónoma del Estado de México Centro Universitario Valle de Chalco no cuentan con un registro de entrada de profesores y alumnos, así como de actividades e inventario, causando conflicto al momento de requerir esta información.

En el presente trabajo se diseñó e implementó un sistema web que permite controlar el acceso de los profesores, alumnos, de igual forma el inventario. Se implementó dentro del funcionamiento del sistema web el uso del código QR (Quick Response Code) para poder acceder al sistema desde algún dispositivo móvil dentro de la red de área local, con la finalidad de verificar las características de cada herramienta o máquina del inventario y facilitar el acceso a la información.

El sistema web es adaptable lo que permite que pueda ser utilizado en distintos dispositivos como computadoras de escritorio, computadoras portátiles, teléfonos celulares, tabletas, así como cualquier tipo de resolución de pantalla conservando una interfaz amigable para el usuario y su funcionalidad.

El sistema web se desarrolló haciendo uso del IDE (Integrated Development Environment) NetBeans en su versión 8.2, utilizando el framework de desarrollo web Spring MVC (Model-View-Controller).

Como lenguajes de programación para el desarrollo del BackEnd se utilizó java y JavaScript. En la parte del FrontEnd se utilizó el framework Bootstrap haciendo uso del lenguaje CSS, el lenguaje de hipertexto HTML, así como la librería de JavaScript JQuery logrando que el diseño del sistema sea amigable para el usuario.

El sistema gestor de base de datos utilizado para la creación de este sistema es MariaDB en su versión 10.6.4 utilizando el lenguaje SQL para la creación de la base de datos, ideal para aplicaciones web, el modelo de base de datos que se utilizó es un modelo relacional evitando así la duplicidad de información.

Se utilizó la herramienta SweetAlert, un plugin de JQuery que permite mostrar las alertas de manera más profesional y amigable para el usuario. También se utilizó la página flaticon, esta proporciona algunos iconos gratuitos para agregar al sistema (flaticon, s.f.).

Al finalizar la creación del sistema, se implementó en los talleres de diseño industrial del Centro Universitario Valle de Chalco, para verificar que el funcionamiento sea óptimo y solucione los problemas para los que fue diseñado.

II. ANTECEDENTES

A continuación, se presentan los datos más relevantes que existen de acuerdo con los trabajos de investigación encontrados, relacionados con el proyecto de investigación presentado en esta tesis.

Una computadora es un dispositivo informático que es capaz de recibir, procesar, almacenar y devolver información, estos procesos los realiza mediante una serie de instrucciones que se encuentran almacenadas en ella, a esta serie de instrucciones se les denomina software.

El software puede definirse como las instrucciones y procedimientos que dan como resultado programas para un sistema de computación. Son las instrucciones que han sido establecidas con anterioridad por un programador para ejecutar una tarea en específico, estas instrucciones están compuestas por un conjunto de números binarios conocidos como bits, que son almacenados en un soporte físico conocido como hardware para que la computadora pueda leer y ejecutar dichas instrucciones (Martínez, 2022).

La mayoría de los lenguajes de programación se caracteriza por ser interpretados o compilados, lo que determina la manera en cómo serán ejecutados en una computadora, java tiene la característica de ser al mismo tiempo compilado e interpretado (Ptolomeo, s.f., p.1).

Un framework sirve como una estructura de trabajo para poder desarrollar software, ya que es una especie de plantilla de la cual los desarrolladores pueden hacer uso para crear su aplicación, compiladores, librerías y Apis, entre otras herramientas, para conectar los diversos componentes de un proyecto y que cumpla con los requerimientos del cliente (Profile,2021, párr.5).

En la actualidad Spring es el framework de Java más popular para soluciones empresariales. Permite desarrollar aplicaciones web complejas con un alto rendimiento, eliminando gran parte del código repetitivo y la configuración asociados al desarrollo web (Profile,2021, párr.9).

Spring es un marco de trabajo ligero y potente que soporta lenguajes para la JVM (Java Virtual Machine) como Java, Kotlin o Groovy. Dispone de diversos subframeworks entre los que se encuentran Spring MVC, Spring Core, Spring Security o Spring ORM (Profile,2021, párr.10).

Spring Web MVC es un subproyecto Spring que está dirigido a facilitar y optimizar el proceso de creación de aplicaciones web utilizando el patrón MVC (Modelo-Vista-Controlador), donde el modelo representa los datos que manejará la aplicación web, la vista son todos los elementos de la interfaz del usuario (UI), con ellos el usuario interactúa con la aplicación, ejemplo: botones, campos de texto, etc., finalmente el controlador será el encargado de manipular los datos en base a la interacción del usuario (Acodigo , 2017) .

Se han creado gran variedad de sistemas utilizando el framework de Spring Web MVC, Federico Julián Gutiérrez Faraoni en su proyecto de fin de curso creo una aplicación web utilizando Spring Framework, para un gestor de recetario en la universidad politécnica de Madrid (escuela técnica superior de ingeniería de sistemas informáticos), teniendo como objetivo el estudio de Spring Framework y la implementación completa. Dependiendo de los usuarios la aplicación tiene un rango distinto de acciones disponibles, creación de recetas, modificación o eliminación de recetas (Faraoni, 2015).

Para el diseño de las vistas se utilizó Bootstrap 3.3.2 Framework gratuito, HTML, CSS Y JS, jQuery 1.11.2 librería de JavaScript que ofrece Ajax, como administrador de base de datos MySQL Workbench 6.3. Se utilizo la aplicación

web con registro y autenticación de usuarios que son capaces de crear, visualizar y modificar recetas. La aplicación está contenida en un servidor web Apache Tomcat. Se concluye que Spring Framework ofrece facilidad en cualquier implementación, es uno de los mejores frameworks para desarrollo web (Faraoni, 2015).

De igual manera se han desarrollado sistemas para el control de salidas como el caso de la Universidad Nacional Autónoma de México (UNAM) donde se desarrolló un artículo en el cual se creó un sistema de control de salidas de alumnos de escuelas (TACS), el sistema provee la función de autorización que permite o prohíbe la salida de un alumno de una escuela, basando el permiso en la presencia de una persona autorizada para recogerlo, la población donde se aplicó son guarderías, colegios y otras instituciones de tipo escolar (Ayala-Hernández C.C., 2011).

Se concluyó que el uso del framework de Spring Web MVC facilita el uso de los datos tanto para sus usuarios, que deben actualizarlos, como para los programadores que tuvieron que hacer los programas correspondientes (Ayala-Hernández C.C., 2011).

El sistema de control de entradas y salidas permitirá visualizar a los usuarios o personal académico que ingresan a los talleres de diseño industrial. El uso del software para el control del inventario facilitará el proceso para conocer con que máquinas, herramientas y materiales a utilizar cuentan los talleres de diseño industrial, se implementarán de igual manera los códigos QR (código de respuesta rápida) para poder tener acceso al sistema. Se facilitará el proceso para registrar las actividades que realicen los alumnos y profesores que ingresan a los talleres, brindando un servicio de atención rápida, eficiente y oportuna para cubrir las necesidades tanto de alumnos como profesores, logrando un acceso a la información de manera específica y detallada.

2.1 Justificación

Realizando la investigación se dieron a conocer las causas por las cuales se origina la falta de información y el control inadecuado del inventario en los talleres de diseño industrial del Centro Universitario Valle de Chalco, para resolver esta problemática se creará un sistema de control utilizando el framework Spring MVC.

Se implementará el uso del framework Spring MVC, Bootstrap, código QR, así como los lenguajes de programación para llevar un control adecuado de las entradas y salidas al taller, de los alumnos, así como el registro del inventario, notando como por medio de este servicio de tecnología se facilita el acceso a la información que se requiera en un determinado momento.

El desarrollo del sistema radica en que los alumnos de licenciatura en diseño industrial y profesores hacen uso de manera recurrente de los talleres, estos llevan el control de las personas que ingresan y salen de los mismos en un documento de excel que es funcional, pero no es formalmente un sistema de gestión de bases de datos. El control del inventario con el que cuentan (máquinas y herramientas) se lleva de manera física corriendo algunos riesgos como, la pérdida de información en caso de perder dicho archivo y el fácil acceso a la información por parte de cualquier usuario.

Por esto se pretende realizar un sistema el cual registre la entrada y salida tanto de alumnos como profesores, así como que taller van a utilizar y para que lo utilizarán, se llevará el inventario de manera más ordenada haciendo fácil el acceso a la información y características de los objetos del inventario, la finalidad del sistema es optimizar los procesos y hacer la información accesible.

III. IMPORTANCIA DEL PROBLEMA

Los talleres de diseño industrial del Centro Universitario Valle de Chalco, no cuentan con un sistema que permita saber que alumnos o profesores ingresan a los talleres, de igual manera no se sabe las actividades que realizan, la materia por la cual asisten y el tiempo que permanecen ahí, el Licenciado en diseño industrial Cesar Adolfo Muñoz Herrera encargado de los talleres, menciona que un sistema de control que permita registrar las entradas, salidas y a su vez llevar el control del inventario de manera adecuada seria de mucha utilidad y a su vez facilitaría la consulta de esta información.

IV. PLANTEAMIENTO DEL PROBLEMA

Una de las prioridades de los talleres de diseño industrial es conocer la información respecto a los alumnos y profesores que acuden a ellos a realizar sus actividades, así como tener el control adecuado de las máquinas y herramientas con las que se cuenta en el inventario, para su posterior consulta en caso de requerirlo en algún determinado momento.

En base a una entrevista directa con el encargado de los talleres de diseño industrial, el Licenciado en diseño industrial Cesar Adolfo Muñoz Herrera, donde una de sus labores es llevar el control del inventario, así como de las actividades que se realizan dentro de los talleres, menciona que los registros tanto de entrada como de salida de los talleres se llevan en un documento de excel, que si bien es funcional no es lo adecuado ni lo óptimo, ver figura 1.

Bitácora de ingreso a talleres de Diseño Industrial										
Responsable: LDI Cesar Adolfo Muñoz Herrera										
Nº	Fecha	Hora de ingreso	Hora de salida	Taller a solicitar	Docente	Actividad	Unidad de Aprendizaje	No. De Alumnos	Grupo	Observaciones
1										
2										
3										
4										
5										
6										
7										

Figura 1. Bitácora de ingreso a los talleres.

Fuente: Propia (2023).

El uso de este documento es funcional al momento de registrar la información, pero al ser un documento público cualquier persona que ingrese puede hacer modificaciones a los registros ya hechos, colocar datos erróneos respecto a la hora que ingresan y salen de los talleres haciendo que la información carezca de validez.

De igual manera mencionó que el control del inventario se lleva de manera física, sin ningún archivo digital que almacene esta información. El llevar el registro de manera física tiene desventajas en caso de pérdida es imposible de recuperar esta información, además del tiempo que demora buscar una información en específico.

4.1 Pregunta de investigación

Por lo anterior planteado en este trabajo se proponen las siguientes preguntas de investigación.

¿Se cubrirán las necesidades de la administración de talleres de diseño industrial mediante un sistema de control?

¿Cuál es el método que se debe implementar para llevar el registro correcto de entradas y salidas de los talleres?

¿La implementación del código QR facilitará el acceso a la información del sistema?

V. MARCO TEÓRICO

En el presente trabajo se realiza el diseño e implementación de un sistema web para registrar entradas y salidas, así como llevar el inventario de los talleres de diseño industrial del Centro Universitario Valle de Chalco. Por lo tanto, es necesario mencionar algunos conceptos de ingeniería de software para comprender la lectura.

5.1 Ingeniería de Software

Ian Sommerville (2005) en su libro titulado "Ingeniería de software" indicó que la ingeniería de software fue propuesta inicialmente en 1968 en una conferencia para discutir lo que en ese entonces se llamó la crisis del software. La crisis del software fue resultado de nuevas computadoras basadas en circuitos integrados. El poder de cómputo hizo que las aplicaciones hasta ese entonces irrealizables fueran una propuesta factible (p.5).

Esta disciplina se encarga del desarrollo de software en todas sus etapas que van desde el análisis, cuando se tiene la primera interacción con el cliente hasta la etapa final cuando el software se encuentre en uso. La ingeniería de software se apoya de técnicas, metodologías y diferentes herramientas para obtener un software eficiente y de calidad, de igual manera se crea la documentación requerida para desarrollar operar y dar mantenimiento al software.

5.2 Programación

Es el proceso mediante el cual se le da a la computadora una serie de instrucciones para que esta las interprete y realice dicha acción, estas instrucciones se da haciendo uso de un paradigma de programación, así como

de lenguajes de programación para obtener como resultado programas, aplicaciones o en el caso de esta tesis un sistema web.

5.3 Paradigmas de programación

Digital Guide (2020) en su análisis de los paradigmas de programación los describe como los principios fundamentales de la programación de software. Estilos de programación diferenciados que, en consecuencia, generan códigos software estructurados de manera distinta (párr.1).

En la siguiente imagen se muestran los tipos de paradigmas de programación, ver figura 2.



Figura 2. Paradigmas de programación.

Fuente: <https://www.ionos.mx/digitalguide/> (2020).

5.3.1 Programación imperativa

Digital Guide (2020) describió la programación Imperativa como el paradigma clásico, ya que los primeros lenguajes de programación, así como los primeros programas informáticos se basan en este paradigma que proporciona una secuencia regularizada de órdenes determinadas. Por lo tanto, el código de programación resultante es fácil de comprender y muy abarcable (párr.3).

5.3.2 Programación estructurada

En cuanto a la programación estructurada Digital Guide (2020) explica que es una forma simplificada de la programación imperativa. La diferencia radica que este paradigma provee el uso de bucles y estructuras de control para realizar una instrucción de forma automática siempre que se dé una determinada condición (párr.4).

5.3.3 Programación procedimental

De igual manera Digital Guide (2020) describió que sentido y el propósito de la programación procedimental es hacer que el código del programa sea más claro y evitar las repeticiones innecesarias de código. Mediante la abstracción de los algoritmos, el paradigma de software procedimental representa un paso decisivo desde los lenguajes ensambladores simples hasta los lenguajes estándar complejos (párr.5).

5.3.4 Programación modular

Con relación a la programación modular Digital Guide (2020) indica que es una adaptación del enfoque procedimental para requerimientos de proyectos de software más amplios. El código fuente se divide específicamente en bloques parciales lógicos independientes los unos de los otros para proporcionar más transparencia y facilitar el proceso de debugging (resolución de errores) (párr.6).

5.3.5 Programación declarativa

Digital Guide (2020) explica que la programación declarativa se centra en el resultado que se busca. Por lo tanto, en primera línea se encuentra el “qué” del resultado y no el “cómo” de los pasos que llevan a la solución, para poder

comprender el código es complicado ya que se requiere un alto grado de abstracción, aunque resulta muy corto y preciso (párr.7).

5.3.6 Programación lógica

Respecto a la programación lógica Digital Guide (2020) explica que también es conocida como programación predictiva, así mismo se basa en la lógica matemática. Un software programado según este principio contiene un conjunto de principios que se pueden entender como una recopilación de hechos y suposiciones, en lugar de una sucesión de instrucciones. Todas las solicitudes al programa se procesan de forma que el intérprete recurre a estos principios para posteriormente aplicar reglas definidas previamente para alcanzar el resultado deseado (párr.13).

5.3.7 Programación funcional

Para terminar con los paradigmas de programación Digital Guide (2020) explica que un programa de programación funcional consta principalmente de llamadas de función concatenadas en las que cada parte del programa se interpreta como una función.

En este sentido, las funciones dentro de la programación funcional pueden adoptar distintas estructuras (párr.10).

5.4 Lenguaje de programación

Los lenguajes de programación son lenguajes con una serie de símbolos y reglas de sintaxis y semántica que definen la estructura principal del lenguaje. así mismo les dan significado a sus elementos. Permiten a los programadores indicar una serie de instrucciones o algoritmos con el fin de controlar el funcionamiento físico y lógico de una computadora.

Los lenguajes de programación se clasifican en niveles dependiendo su grado de complejidad:

5.4.1 Bajo nivel

Marker (2022) define los lenguajes de programación de bajo nivel como aquellos que interactúan directamente con el hardware del equipo, es lo más cerca a trabajar con el procesador. Por lo tanto, es un poco complicado de entender para el programador, sin embargo, para la computadora es sencillo realizar la compilación de estos programas que regularmente en cuanto a funcionamiento suelen ser rápidos, como ejemplo de estos lenguajes tenemos a Cobol, Pascal y Assembler (párr.17).

5.4.2 Medio nivel

De igual manera Marker (2022) explica que los lenguajes de nivel medio se utilizan en el desarrollo de sistemas operativos, debido a su nivel de abstracción. Este permite acercarse al lenguaje máquina utilizando herramientas complejas, pero conservando la eficiencia de los lenguajes de bajo nivel, como ejemplo de estos lenguajes tenemos a Java, C# (párr.18).

5.4.3 Alto nivel

Marker (2022) explica que los lenguajes de programación de alto nivel son aquellos que mediante una interfaz gráfica permiten crear, modificar e implementar los algoritmos necesarios para que el programa funcione. Esto en una serie de sentencias y órdenes estandarizadas, fáciles de aprender si se obtiene la educación adecuada. Con un lenguaje de alto nivel, se puede crear una aplicación y administrarla de la manera más sencilla y rápida (párr.20).

5.5 Programación orientada a objetos

Serrano (2010) define la programación orientada a objetos como un método de desarrollo en el cual los programas se organizan como colecciones de objetos que cooperan para resolver un problema. Enuncia que los objetos pueden corresponderse a entidades del mundo real (como un coche o un gato), a acciones (como saltar o realizar una transacción bancaria) o a procesos (como el vuelo o el aprendizaje), tienen la característica de poseer variables internas, así mismo cuentan con un estado, un comportamiento definido y una identidad (p. 5).

La programación orientada a objetos se basa en el modelo de objetos. Este modelo se basa en el uso de 7 capacidades, 4 de las cuales se consideran principales y 3 secundarias. Serrano (2010) define las capacidades principales de los objetos como las siguientes:

5.5.1 Abstraer

Consiste en obtener características o funciones esenciales de un objeto que lo hacen diferente del resto ocultando la información innecesaria. Serrano (2010) indica que los humanos hemos desarrollado esta característica de abstracción, nos dice como ejemplo que al estudiar algo ignoramos los detalles y tratamos con ideas muy generales lo que estemos estudiando, generando así un modelo simplificado (p. 6).

5.5.2 Encapsular

El objetivo principal de encapsular es ocultar la implementación de una abstracción, como lo indicó Serrano (2010) sirve para que ninguna parte del sistema dependa de como se ha implementado en otra parte. La ventaja es que facilita cambiar el funcionamiento de una abstracción (p. 9).

De igual manera Serrano (2010) indicó que la abstracción y la encapsulación se complementan entre sí. Mientras que la abstracción se centra en el comportamiento observable y define la interfaz, la encapsulación se basa en cómo se define ese comportamiento y los detalles de la implementación (p. 9).

5.5.3 Modularizar

En cuanto a modularizar Serrano (2010) dijo que es la capacidad que nos permite poder dividir un programa en diferentes agrupaciones llamadas módulos, Esto con la finalidad de facilitar el mantenimiento, diseño y revisión, además de brindar un aumento en la velocidad de compilación (p. 14).

5.5.4 Jerarquizar

De igual manera Serrano (2010) enunció que jerarquizar es la capacidad que nos permite ordenar las abstracciones de un sistema en una jerarquía, esto nos permite detectar estructuras y comportamientos comunes para así facilitar el desarrollo, gracias a la jerarquización clases o interfaces nuevas pueden heredar características o métodos de otras ya existentes (p. 9).

Así mismo Serrano (2010) define las capacidades secundarias de los objetos como las siguientes:

5.5.5 Tipo

Serrano (2010) define que un tipo es una característica precisa asociada a un conjunto de objetos, a la asociación del tipo a un objeto se le conoce como tipado. De esta manera existen lenguajes de programación con tipado fuerte, en los cuales no es posible mezclar variables de tipos diferentes y lenguajes de

programación con tipado débil, en los cuales es posible realizar esta mezcla de variables (p. 16).

5.5.6 Concurrency

En cuanto a la concurrencia Serrano (2010) la define como la capacidad que nos permite una ejecución paralela de varias sentencias de instrucciones, sirve en casos cuando se cuenta con múltiples procesadores y se desea hacer uso de todos para resolver el problema. Indicó que los lenguajes de programación no dan soporte para la concurrencia, generalmente esta es dada por los sistemas operativos (p. 18).

5.5.7 Persistencia

Con respecto a la persistencia Serrano (2010) define que es la capacidad que permite que los datos existentes trasciendan en el tiempo y espacio. Permite grabar objetos que ya existen, así como la definición de sus clases para poder cargarse más adelante sin ambigüedad (p. 19).

5.6 Aplicaciones web

Las aplicaciones web son básicamente programas que se encuentran disponibles dentro de la red de internet, los archivos y documentos que se encargan del funcionamiento regularmente se encuentran almacenados en servidores. Por lo que para hacer uso de estas aplicaciones no es necesario instalarlas en nuestra computadora, solo es necesario tener el enlace de dicha aplicación web y acceder por medio de algún dispositivo electrónico, a algún navegador para posteriormente abrir dicho enlace.

En el presente trabajo se realizó un sistema web para dar solución a la problemática con la que se contaba en los talleres del Centro Universitario Valle

de Chalco. Es importante identificar las diferencias entre un sistema web y una página web ya que suelen confundirse, aunque ambas son aplicaciones web el funcionamiento de cada una de ellas es diferente.

5.6.1 Página web

Las páginas web se crean principalmente para servicios de marketing donde la principal función es mostrar al usuario algunos productos o servicios, son de carácter informativo y contienen material llamativo como lo son imágenes, sonidos, videos, gifs, entre otros.

5.6.2 Sistema web

Un sistema web al igual que las páginas web se encuentra alojado en un servidor en el caso de este trabajo el servidor es local, la principal diferencia con una página web es que un sistema web es creado principalmente para ofrecer una solución a un problema en específico, como en este caso a realizar los registros de entradas y salidas de los talleres de diseño industrial, así como el inventario.

Regularmente contiene una base de datos que nos permite almacenar la información del sistema para posteriormente consultarla o hacer cambios en tiempo real.

5.7 Desarrollo web

El desarrollo web es el proceso mediante el cual se realiza la creación de una página o sistema web haciendo uso de algunos de los lenguajes de programación, el desarrollo web se divide principalmente en dos partes principales:

5.7.1 Front-end

Es la parte que interactúa con el usuario, tanto en diseño como en funcionamiento. Por ello está íntimamente relacionada con la experiencia del usuario y la interfaz de usuario (HubSpot, 2022).

5.7.2 Back-end

Se refiere a la parte que entra en contacto directo con el servidor, es donde se aplica el código de programación para crear la estructura. Permanece en un segundo plano a cargo de la accesibilidad, actualización, bases de datos y cambios del sitio (HubSpot, 2022).

Entre los lenguajes de programación más comunes para el desarrollo del back-end y el front-end en el ámbito del desarrollo web tenemos:

5.7.3 HTML

En investigaciones de Rodríguez (2022) menciona que el origen de HTML se remonta a 1980, cuando el físico Tim Berners-Lee, trabajador del CERN (Organización Europea para la Investigación Nuclear) propuso un nuevo sistema de "hipertexto" para compartir documentos más fácilmente (párr.10).

HTML es el código utilizado para creación de las páginas web y su contenido, su nombre son las siglas en inglés de HyperText Markup Language, traducándose al español como Lenguaje de Marcado de Hipertexto, se considera el estándar en el despliegue y estructuración de páginas web a nivel mundial, por lo que la mayoría de los sitios en internet se elaboran con él (Rodríguez, 2022, párr.6).

No es un lenguaje de programación sino de marcado el cual se encarga de otorgarle la estructura correspondiente al contenido de la página en cuestión, este lenguaje consiste en las marcas que sirven de guía a los servidores en el estilo de los documentos que desean incorporar (Rodríguez, 2022, párr.7).

5.7.4 JavaScript

Mohedano et al. (2012) en su libro “iniciación a JavaScript” indican que es un lenguaje de programación que se utiliza principalmente para crear páginas web capaces de interactuar con el usuario. Menciona que las páginas web se consideran estáticas cuando se limitan a mostrar un contenido establecido por el programador sin proporcionar más opciones al usuario que elegir entre los enlaces disponibles para seguir navegando. Cuando un programador incorpora JavaScript a su página, proporciona al usuario la capacidad de interacción con la página web al añadir procesos en respuesta a las acciones que realiza el usuario. Es importante señalar que estos procesos se ejecutan en la máquina cliente y por tanto no implican intercambio de datos con el servidor (p.9).

De igual manera mencionan que JavaScript es un lenguaje interpretado eso significa que las instrucciones son analizadas en secuencia por el intérprete de JavaScript del navegador web, de manera que su ejecución es inmediata a la interpretación Mohedano et al (2012).

5.7.5 CSS

Eguíluz (2008) en su libro “Introducción a CSS” explica que es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. El lenguaje apareció poco después que el lenguaje de etiquetas SGML en el año 1970. CSS es la mejor forma de separar los contenidos, se utiliza para definir el aspecto de cada

elemento: color, tamaño, tipo de letra del texto, separación entre elementos, posición de cada elemento dentro de la página web, entre otros aspectos (p. 5).

CSS se divide en niveles que han ido evolucionando a lo largo de la historia, ver tabla 1.

Nivel	Año	Descripción
CSS1	1996	Crea propiedades en las fuentes, color de texto, alineación, bordes, imágenes.
CSS2	1998	Propiedades de posicionamiento de los contenedores.
CSS2.1	2005	corrige algunos errores, elimina funcionalidades poco soportadas por los navegadores y añade nuevas.
CSS3	2011	Se comienzan a crear características de CSS como módulos independientes.

Tabla 1. Niveles de CSS.

Fuente: Propia (2023).

El gran impulso del lenguaje de hojas de estilos se produjo con el boom de Internet y el crecimiento exponencial del lenguaje HTML. Para lograr que la página se muestre de la misma manera en diferentes navegadores un programador web está limitado por las posibilidades de los navegadores que utilizan los usuarios para acceder a sus páginas. Los navegadores internamente están divididos en varios componentes. La parte del navegador que se encarga de interpretar el código HTML y CSS para mostrar las páginas se denomina motor, esto hace que desde el punto de vista del programador CSS, la versión de un motor es mucho más importante que la versión del propio navegador (Eguíluz, 2008, p. 5).

5.7.6 PHP

Es el lenguaje de programación web que se utiliza para el desarrollo del back-end y puede ser agregado en HTML, es considerado uno de los lenguajes más usados.

PHP en su página oficial explica que fue creado en 1994 por Rasmus Lerdorf, la primera encarnación de PHP era un conjunto simple de ficheros binarios (Common Gateway Interface) escritos en el lenguaje de programación C. Este modelo fue capaz de interactuar con bases de datos y mucho más, proporcionando un entorno de trabajo sobre el cual los usuarios pueden desarrollar aplicaciones web dinámicas sencillas tales como libros de visitas (PHP, s. f.).

“El motivo del éxito de PHP no es solo la facilidad con la que permite desarrollar páginas, sino también formar parte del stack de tecnologías más habitual en Internet (Linux, Apache, MySQL)” (PHP, s. f.).

5.7.7 Java

En su investigación Martínez (2022) explicó que java es un lenguaje de programación desarrollado por Sun Microsystems y fue presentado en la segunda mitad del año 1995, se ha convertido en un lenguaje de programación muy popular. Java es un lenguaje muy valorado porque los programas Java se pueden ejecutar en diversas plataformas con sistemas operativos como Windows, Mac Os, Linux o Solaris (p. 2).

De igual manera Lenovo México (2022) en su página oficial describe a Java como un lenguaje de programación orientado a objetos diseñado específicamente para permitir a los desarrolladores una plataforma de

continuidad. Dice que Java difiere de otros paradigmas de programación, esto debido a que los desarrolladores pueden continuar o actualizar algo que ya han terminado, en lugar de empezar desde cero, se esta manera los objetos mantienen el código ordenado haciendo que sea más sencillo de modificar cuando es necesario (párr.1).

5.7.8 Python

Challenger et al. (2014) en un artículo publicado en la revista Ciencias Holguín, nos hablan sobre la historia de Python, donde indican que Python fue creado por Guido van Rossum, un programador holandés a finales de los 80 y principio de los 90, mientras que se encontraba trabajando en el sistema operativo Amoeba (p. 3).

El mismo artículo nos indica que Python es un lenguaje de programación muy utilizado debido a sus diferentes aplicaciones y lo entendible que resulta para los programadores ya que su sintaxis es muy sencilla y fácil de utilizar, tanto que en algunas ocasiones parece pseudocódigo, esto facilita el trabajo a los desarrolladores.

En el ámbito del desarrollo web, Python es uno de los lenguajes de programación más utilizados para desarrollar el back-end de las aplicaciones web.

5.8 Base de datos

Las bases de datos se utilizan para almacenar grandes cantidades de información o datos. Se han utilizado desde la antigüedad donde se tenía la información sobre las cosechas, los alimentos, entre otras cosas que se contabilizaban en aquella época.

Respecto a las bases de datos en informática Click-it (2022) indicó que su origen se remonta a 1884 con Herman Hollerith, que desarrolló el tabulador electromagnético de tarjetas perforadas con el fin de ayudar en el resumen de información y posteriormente a la contabilidad (párr.3).

Las bases de datos contienen entidades y estas a su vez atributos en donde se guardan diferentes tipos de datos, esto con la finalidad de evitar la mayor redundancia posible. Para posteriormente esos datos convertirlos a información que se puede consultar en algún momento determinado. Existen diferentes tipos de bases de datos cada una con características diferentes, de acuerdo con las necesidades de los usuarios.

5.8.1 Estáticas

HostingPlus México (2021) indicó que las bases de datos estáticas son aquellas cuyo diseño está enfocado a la lectura de datos, su principal función es almacenar y registrar datos con el objetivo de realizar estudios históricos y comparativos para mejorar la toma de decisiones (párr.6).

5.8.2 Dinámicas

De igual manera HostingPlus México (2021) indicó que en las bases de datos dinámicas no solo se almacena la información, sino que se va modificando a lo largo del tiempo cuando es necesario (párr.7).

5.9 Modelo de bases de datos

Se considera como modelo de datos al conjunto de ideas lógicas que se utilizan para representar los datos y la relación que existe entre ellos. Algunos de los modelos más utilizados son los siguientes:

5.9.1 Jerárquicas

HostingPlus México (2021) Describe las bases de datos jerárquicas como bases de datos donde la información se clasifica siguiendo una estructura jerárquica. La estructura jerárquica es en forma de árbol, comienza por una raíz de la que surgen distintos nodos llamados padres, y desde cada uno de ellos surgen nuevos nodos llamados hijos y así sucesivamente (párr.10), ver figura 3.

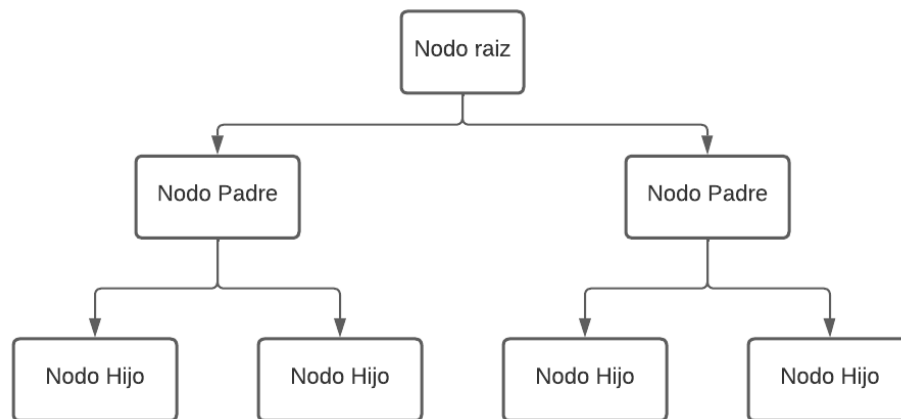


Figura 3. Ejemplo de base de datos jerárquica.

Fuente: Propia (2023).

5.9.2 De red

HostingPlus México (2021) indica que se trata de un tipo especial de base de datos jerárquica donde cada nodo puede tener distintos padres por lo cual se tiene acceso a las entidades por diferentes caminos y esto hace que la búsqueda sea más rápida. Es utilizada por programadores, aunque eso implique una dificultad a la hora de administrar la base de datos (párr.11), Ver figura 4.

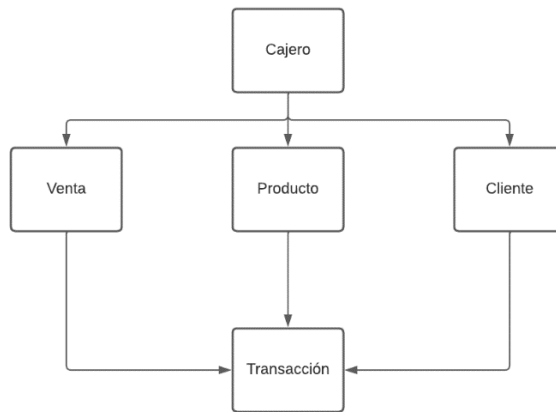


Figura 4. Ejemplo de base de datos de red.
Fuente: Propia (2023).

5.9.3 Relacionales

Oracle (s. f.) Explica que las bases de datos relacionales son un tipo de base de datos que almacena y proporciona acceso a puntos de datos relacionados entre sí. Las bases de datos relacionales se basan en el modelo relacional, una forma intuitiva y directa de representar datos en tablas ya que cada fila en una tabla es un registro con una ID o llave primaria única. Las columnas de la tabla contienen los atributos de los datos, de igual manera cada registro suele tener un valor para cada atributo (párr.1), ver figura 5.

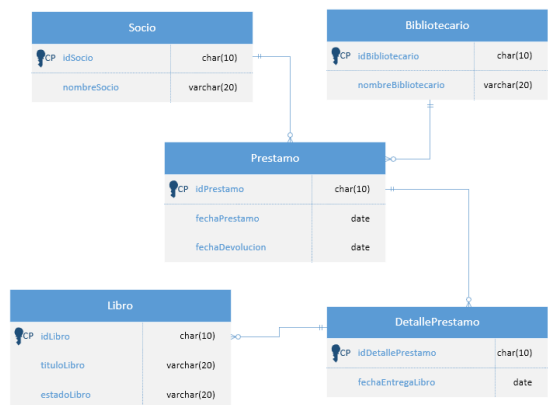


Figura 5. Ejemplo de base de datos relacional.
Fuente: Propia (2023).

5.9.4 Transaccionales

De igual manera HostingPlus México (2021) indicó que con las bases de datos transaccionales se persigue el envío y recepción de la información a una gran velocidad. Su principal uso es a nivel industrial o empresarial cuando se necesita que la información se transfiera de forma muy rápida (párr.14), ver figura 6.

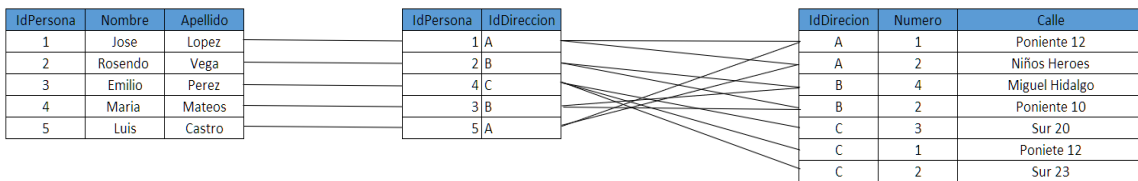


Figura 6. Ejemplo de base de datos transaccional.

Fuente: Propia (2023).

5.9.5 Multidimensionales

En cuanto a las bases de datos multidimensionales HostingPlus México (2021) explico que se utilizan para funciones específicas, pudiendo representarse dimensiones dentro de una tabla de datos (párr.15), ver figura 7.

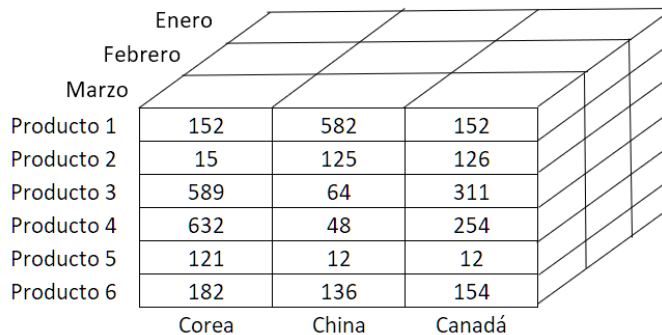


Figura 7. Ejemplo de base de datos multidimensional.

Fuente: Propia (2023).

5.9.6 Orientadas a objetos

Por último, HostingPlus México (2021) indicó que en las bases de datos orientadas a objetos no se almacenan datos sobre un objeto, sino que se almacena el objeto en sí. La ventaja de utilizar este tipo de base de datos es que admite mayor contenido que otros tipos de bases de datos facilitando al usuario el acceso a información actualizada y completa (párr.16), ver figura 8.

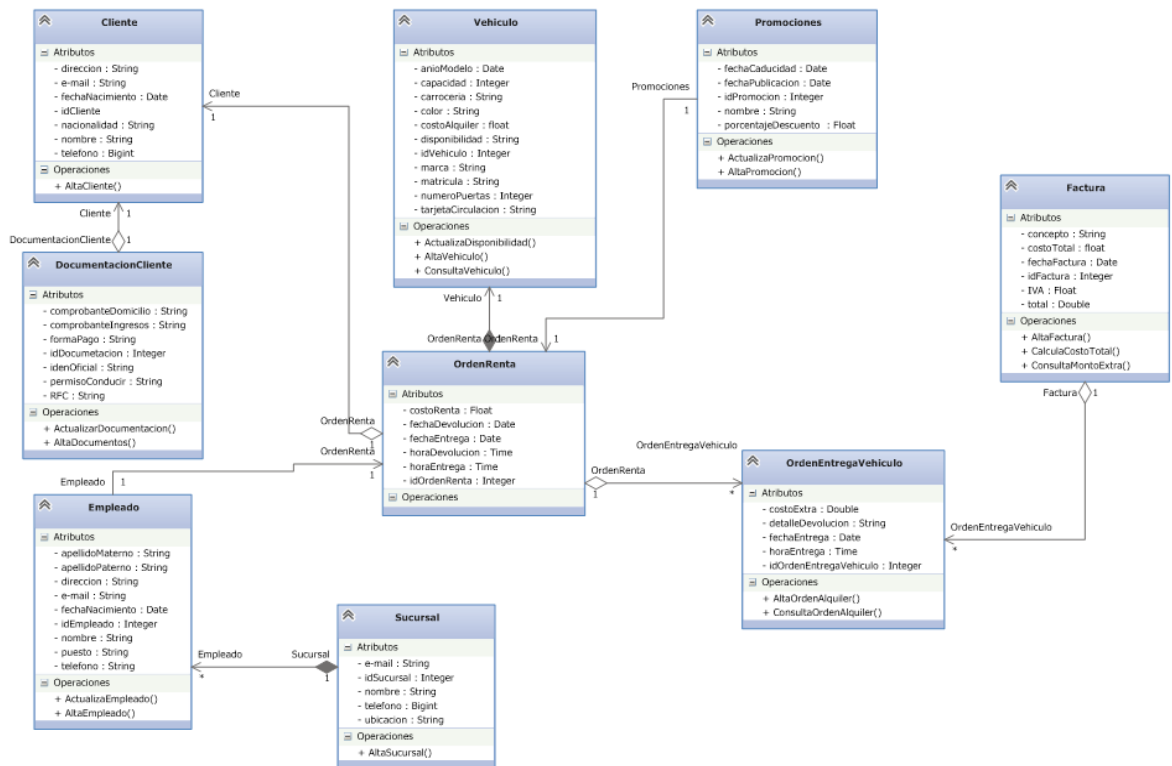


Figura 8. Ejemplo de base de datos orientada a objetos.

Fuente: Propia (2023).

5.10 Sistema gestor de base de datos

La Universidad Europea (2022) define un sistema gestor de base de datos (SGBD) como un software constituido por una serie de programas con la finalidad de crear, gestionar y administrar la información de la base de datos. Su principal objetivo es servir de interfaz entre los usuarios y las aplicaciones para facilitar la

organización de los datos garantizando su accesibilidad, calidad e integridad, brindando a su vez una manera eficaz de administrar esa información (párr. 1).

La Universidad Europea (2022) enunció que un sistema gestor de base de datos cuenta con tres componentes principales, los cuales son el lenguaje de manipulación, un lenguaje de definición de datos y un lenguaje de consulta (párr. 2).

5.10.1 Funciones

La Universidad Europea (2022) define un sistema gestor de base de datos nos permite realizar operaciones sobre la base de datos, de igual manera nos facilita la manipulación de los datos y el mantenimiento de la base de datos, para luego brindar esta información de manera segura y eficiente. Entre las principales funciones de un sistema gestor de base de datos se encuentran las siguientes: (párr. 3).

5.10.1.1 Definición de los datos: Define a los elementos contenidos dentro de la base de datos, para poderlos almacenar según su tipo de estructura (Universidad Europea,2022, párr. 5).

5.10.1.2 Manipulación de los datos: Se encarga de responder a las solicitudes de los usuarios como lo son agregar, actualizar, modificar o eliminar, permitiendo modificar la base de datos sin necesidad de modificar las aplicaciones que la utilizan (Universidad Europea,2022, párr. 6).

5.10.1.3 Preservar la seguridad e integración de los datos: Evita los ataques a los datos o accesos no autorizados (Universidad Europea ,2022, párr. 7).

5.10.1.4 Recuperación y restauración de los datos: Ante un fallo los sistemas gestores de base de datos brindan la posibilidad de recuperar y restaurar la información contenida en la base de datos (Universidad Europea ,2022, párr. 8).

Actualmente en 2023 existen multitud de sistemas gestores de base de datos, estos se clasifican según la forma en la que administran los datos.

5.10.2 Relacionales (SQL)

Los sistemas gestores de bases de datos relacionales utilizan comandos SQL para poder comunicarse con la base de datos y realizar las acciones necesarias, Para la comodidad de los desarrolladores de SQL todos los comandos del lenguaje necesarios en los correspondientes sistemas gestores de bases de datos se ejecutan a través de una interfaz específica de línea de comandos SQL, en la siguiente tabla se muestran algunos de los comandos SQL más comunes, ver tabla 2.

Comando	Función
CREATE DATABASE	se utiliza para crear una nueva base de datos vacía
DROP DATABASE	se utiliza para eliminar completamente una base de datos existente.
CREATE TABLE	se utiliza para crear una nueva tabla.
ALTER TABLE	se utiliza para modificar una tabla ya existente.
DROP TABLE	se utiliza para eliminar por completo una tabla existente.
SELECT	se utiliza cuando quieres leer los datos.
INSERT	Se utiliza para agregar datos nuevos.
UPDATE	Se utiliza para modificar un dato existente.
DELETE	Se utiliza para eliminar datos.
REPLACE	Se utiliza para cambiar o reemplazar datos existentes.
TRUNCATE	se utiliza cuando quieres borrar todos los datos de la plantilla.

Tabla 2. Comandos SQL más comunes.

Fuente: Propia (2023).

Algunos de los sistemas gestores de bases de datos relacionales que existen son:

5.10.2.1 MySQL

MySQL es un sistema gestor de bases de datos que cuenta con una doble licencia. Una parte es de código abierto sin embargo cuenta con una versión comercial gestionada por la compañía Oracle, es un gestor de bases de datos de fácil uso y gran rendimiento, el inconveniente con este gestor es la escalabilidad ya que no trabaja de manera tan eficiente con bases de datos que superen un determinado tamaño ver figura 9.



Figura 9. Logo de MySQL.

Fuente: <https://www.mysql.com/> (2023).

5.10.2.2 MariaDB

Es un sistema gestor de base de datos de código abierto derivado de MySQL, pero adicionalmente cuenta con algunas extensiones que mejoran su funcionalidad, el principal inconveniente es al momento de migrar de MariaDB a MySQL por la incompatibilidad, ver figura 10.



Figura 10. Logo de MariaDB.

Fuente: <https://mariadb.com/> (s.f.).

Este sistema gestor de bases de datos nos resulta muy útil además que cuenta con características que serán necesarias al momento de crear la base de datos para el sistema web, por lo tanto, es el gestor que se utilizó para desarrollo de la base de datos del sistema web.

5.10.2.3 PostgreSQL

Este sistema gestor de base de datos tiene gran flexibilidad en cuanto a lenguajes de programación, además que es multiplataforma está orientado a objetos, fue publicado bajo la licencia de BSD. Al igual que MySQL su gran problema es la escalabilidad ya que no soporta bases de datos muy grandes, ver figura 11.



Figura 11. Logo de PostgreSQL.

Fuente: <https://www.postgresql.org/> (s.f.).

5.10.3 No relacionales (NoSQL)

A diferencia de los sistemas gestores de bases de datos relacionales, los sistemas gestores de bases de datos no relacionales no requieren de estructuras de datos fijas como lo son las entidades ya que están diseñados para datos específicos que no necesitan estar relacionados con otros, cada una de las

entidades funcionan de manera independiente. Algunos de los sistemas gestores de base de datos no relacionales que existen son:

5.10.3.1 MongoDB

Es un sistema gestor de base de datos muy popular y utilizado actualmente en el año 2023, este sistema gestor de base de datos es orientado a ficheros y almacena la información en estructuras BSON, ver figura 12.



Figura 12. Logo de MongoDB.

Fuente: <https://www.mongodb.com/es> (2022).

5.10.3.2 Redis

Este sistema gestor de base de datos está basado en el almacenamiento clave-valor, este gestor guarda toda la información en la memoria cache, lo cual nos garantiza una respuesta rápida, aunque los datos no se encuentren estructurados, ver figura 13.



Figura 13. Logo de Redis.

Fuente: <https://redis.io/> (2022).

5.10.3.3 Cassandra

Al igual que redis este sistema gestor de base de datos utiliza almacenamiento clave-valor, es un gestor distribuido y con gran escalabilidad, este gestor dispone de su propio lenguaje para las consultas denominado CQL (Cassandra Query Language), ver figura 14.



Figura 14. Logo de Cassandra.

Fuente: https://cassandra.apache.org/_/index.html (2022).

5.11 Servidor web

Los servidores web también conocidos como servidores HTTP son un tipo de servidores utilizados para la entrega de contenido web en redes internas o en Internet, dentro de ellos se encuentran almacenados los archivos que se han programado con anterioridad de las páginas web. Utilizan el Protocolo de Transferencia de Hipertexto (HTTP) para responder a las peticiones de los usuarios, como por ejemplo transferir una página web a un navegador.

5.11.1 Servidor local

Un servidor local es un determinado ordenador dentro de la red local en el cual se encuentran almacenados los proyectos web, el administrador tiene acceso a esta máquina y no es necesario cargar los archivos de los proyectos en un servidor de internet.

De esta manera para poder acceder a alguna página o sistema web de este servidor no es necesario colocar ningún dominio, lo principal para poder acceder a los archivos es que el equipo con el que se quiere acceder se encuentre dentro de la misma red local que el servidor, posteriormente basta con colocar la dirección ip del servidor en algún navegador y la sección de archivos determinada.

5.12 Modelo cliente-servidor

El intercambio de información entre el servidor y el ordenador se realiza utilizando el modelo cliente-servidor. El cliente es quien comienza con este proceso al momento que el usuario ingresa a un sitio web desde algún ordenador, el servidor recibe esta petición y muestra como respuesta la información contenida en dicho sitio, en la siguiente figura se muestra este proceso, ver figura 15.

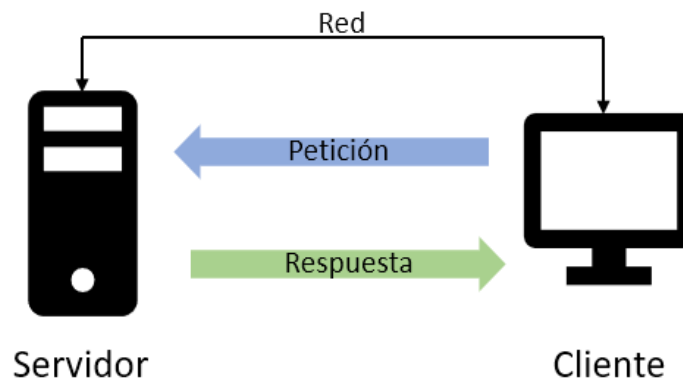


Figura 15. Modelo cliente-servidor.

Fuente: Propia (2023).

5.13 Modelo-Vista-Controlador (MVC)

MVC es una propuesta de arquitectura del software utilizada para separar el código por sus distintas responsabilidades, manteniendo distintas capas que se encargan de hacer una tarea muy concreta (desarrolloweb, 2020, párr. 1).

Su uso es principalmente en sistemas donde se requiere interfaces de usuario, aunque se puede utilizar para distintos tipos de aplicaciones. Sirve para crear un software más robusto con un ciclo de vida más adecuado, brindando facilidad al realizar el mantenimiento, reutilización del código y la separación de conceptos (desarrolloweb, 2020, párr. 2).

La siguiente figura muestra el funcionamiento de MVC, ver figura 16.

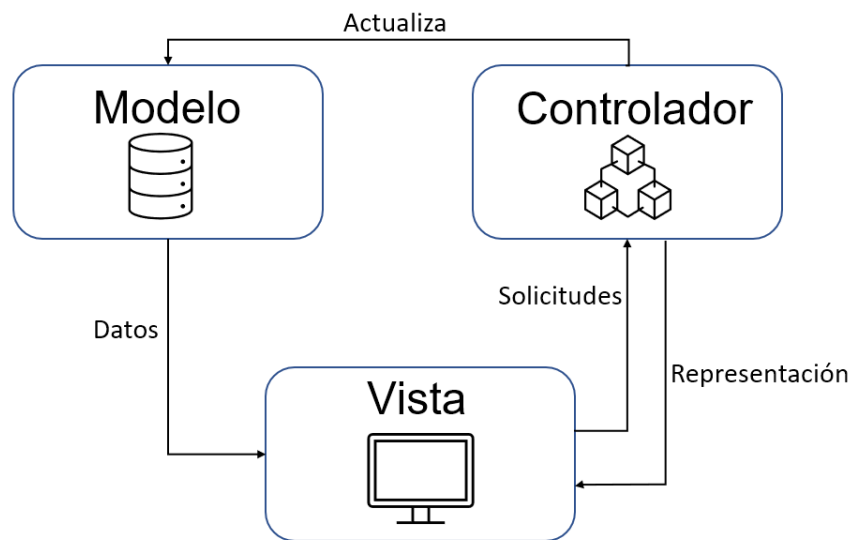


Figura 16. Funcionamiento de MVC.

Fuente: Propia (2023).

Modelo: el modelo encapsula los datos de la aplicación.

Vista: representa los datos del modelo y también genera resultados HTML que el navegador del cliente puede interpretar.

Controlador: el controlador procesa las solicitudes del usuario y las pasa a la vista para su representación (geeksforgeeks, 2022).

5.14 Framework

Un framework es un patrón o esquema que nos proporciona herramientas y características para agilizar la programación, ayuda a llevar una mejor estructura del código, así como a optimizar tiempos de programación.

5.14.1 Framework Spring MVC

Spring MVC Framework sigue el patrón de diseño Model-View-Controller (Modelo-Vista-Controlador), se utiliza para desarrollar aplicaciones web. Funciona alrededor de DispatcherServlet. DispatcherServlet maneja todas las solicitudes y respuestas HTTP y envía las solicitudes a los manejadores. Utiliza `@Controller` y `@RequestMapping` como controladores de solicitudes predeterminados (geeksforgeeks, 2022).

La anotación `@Controller` define que una clase en particular es un controlador. `@RequestMapping` anotación asigna las solicitudes web a los métodos de Spring Controller (geeksforgeeks, 2022).

5.15 Bootstrap

Para la programación del Front-end se usa el uso de Bootstrap, que es un framework de desarrollo gratuito y de código abierto el cual cuenta con algunos diseños y plantillas que nos ayuda al momento de realizar el diseño y desarrollo del sistema web.

Está diseñado para facilitar el proceso de desarrollo de los sitios web adaptables y orientados a los dispositivos móviles, proporcionando una colección de sintaxis para diseños de plantillas del proyecto (HOSTTINGERTUTORIALES, 2022).

Bootstrap ayuda a construir sitios web más rápidamente, ya que consta de scripts basados en HTML, CSS y JS que se utilizan para diversas funciones y componentes relacionados con el diseño web (HOSTTINGERTUTORIALES, 2022).

5.16 Código QR (quick responsive code)

Es un tipo de código de barras bidimensional que consiste en módulos negros cuadrados sobre un fondo blanco. Los códigos QR están diseñados para ser leídos por dispositivos inteligentes. Debido a que pueden transportar información tanto vertical como horizontalmente, pueden proporcionar acceso a una gran cantidad de información, incluidos enlaces, texto u otros datos. Se estandarizaron en 1998 y su estándar internacional (ISO) en el año 2000 (techopedia, 2022).

Los códigos QR son fácilmente identificables por su forma cuadrada y por los tres cuadros ubicados en las esquinas superior e inferior izquierda ver figura 17 (codigos qr, s.f.).

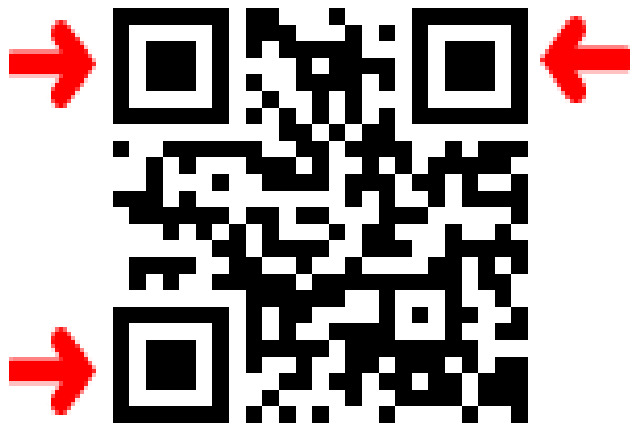


Figura 17. Código QR.

Fuente: Propia (s.f.).

5.17 Modelado de los requerimientos

Es el proceso por el cual se evalúan los requerimientos y necesidades del cliente para así poder brindarle una solución adecuada y concisa a su problema.

Tanto el desarrollador como el cliente tienen un papel activo en el modelado de los requerimientos, el cliente intenta replantear un sistema confuso, a nivel de descripción de datos, funciones y comportamiento, en detalles concretos. El desarrollador actúa como interrogador, como consultor, como persona que resuelve problemas y como negociador (GoogleSites, s.f.).

Es importante realizar un modelado de los requerimientos de forma adecuada ya que las fallas que pueden surgir y ocasionar el fracaso del sistema se derivan comúnmente de un manejo inadecuado de los requerimientos. Sirve para obtener una base sólida sobre la cual se desarrollará el sistema, así como del funcionamiento de este para poder dar solución a los requerimientos del cliente.

5.18 Lenguaje Unificado de Modelado (UML)

Es un lenguaje estandarizado de modelado comúnmente utilizado en el desarrollo de software, ya que consiste en varios tipos de diagramas que ayudan a especificar, construir, visualizar y documentar el modelado y desarrollo de software. Algunos de los diagramas con los que cuenta este lenguaje son:

5.18.1 Diagrama de caso de uso

Los diagramas de casos de uso son un documento narrativo que describe la secuencia de eventos de un actor (Persona que interactúa con el sistema) para completar un proceso. Son narraciones que representan el funcionamiento del

sistema, desde el punto de vista del usuario en un momento determinado. En la siguiente figura se muestra un ejemplo de un diagrama de caso de uso al momento de ingresar a una página web y resolver unas actividades, ver figura 18.

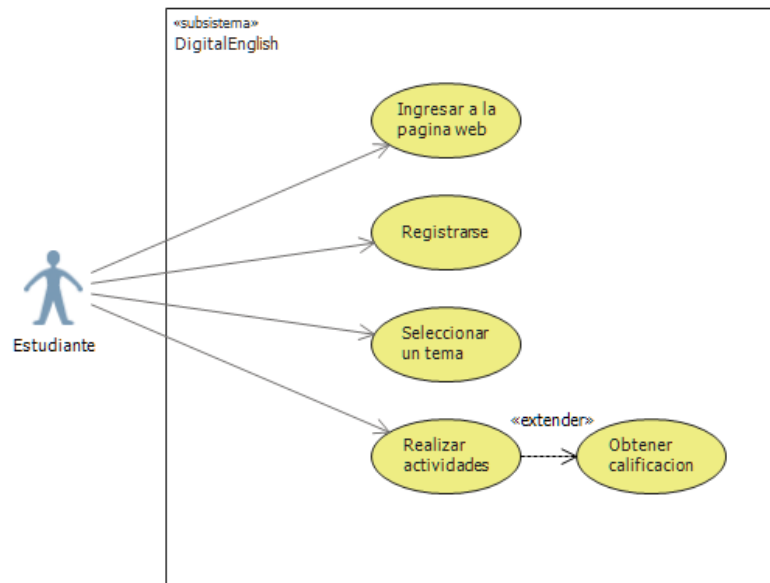


Figura 18. Diagrama de caso de uso.

Fuente: Propia (2023)

5.18.2 Secuencia

Los diagramas de secuencia muestran cómo los objetos interactúan entre sí y el orden de la ocurrencia. Representan de manera gráfica los eventos que pasan de los actores al sistema, en la siguiente figura se muestra un ejemplo de un diagrama de secuencia en el cual un usuario busca un vehículo dentro de una página web, ver figura19.

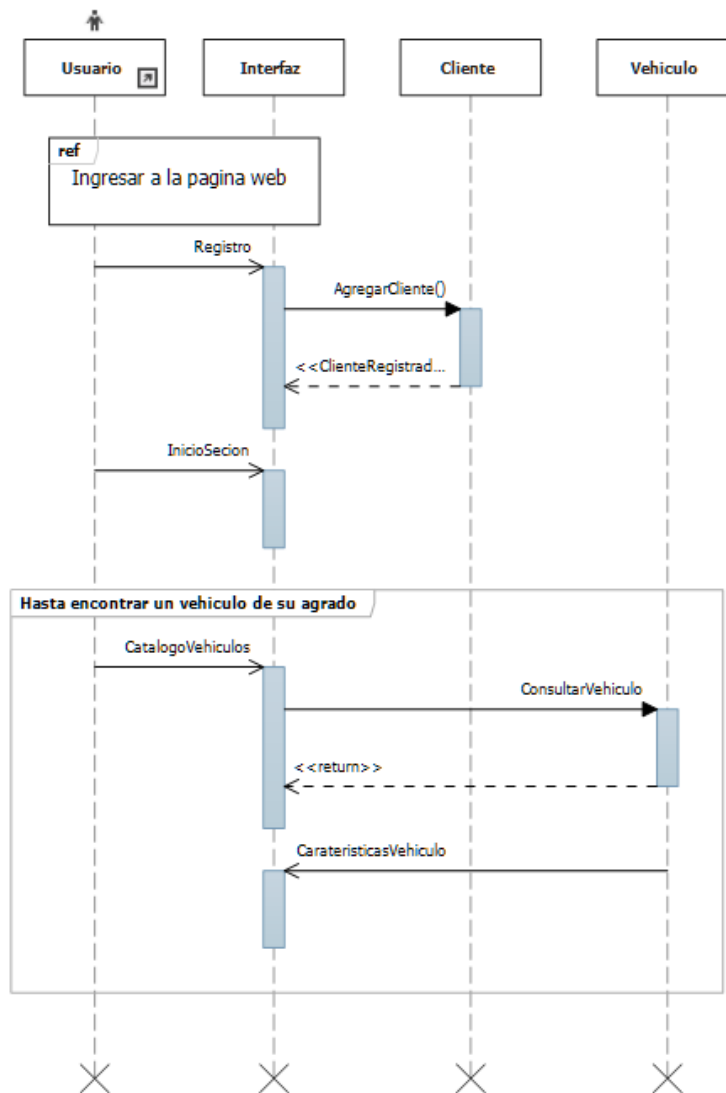


Figura 19. Diagrama de secuencia.

Fuente: Propia (2023).

5.18.3 Modelo entidad-relación

Dentro del modelo entidad-relación se representan las entidades mediante rectángulos que llevan el nombre de la información y este solo debe aparecer una vez dentro del esquema, en cambio las relaciones representan una asociación entre dos entidades mediante un rombo.

En este modelo se pueden relacionar las entidades de tres maneras diferentes, describiendo el número de instancias de una entidad asociadas a otra uno a varios (1: N), uno a uno (1: 1), varios a varios (N: N).

En la siguiente figura se observan los símbolos utilizados para la creación del modelo entidad-relación, ver figura 20.

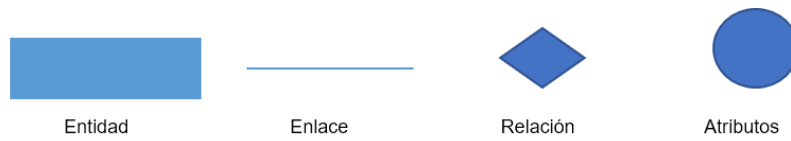


Figura 20. Simbología modelo entidad-relación.

Fuente: Propia (2023).

En la siguiente figura se muestra un ejemplo de un modelo entidad relación de una venta de casas y terrenos, en este diagrama podemos ver el tipo de relación que tienen las entidades, ver figura 21.

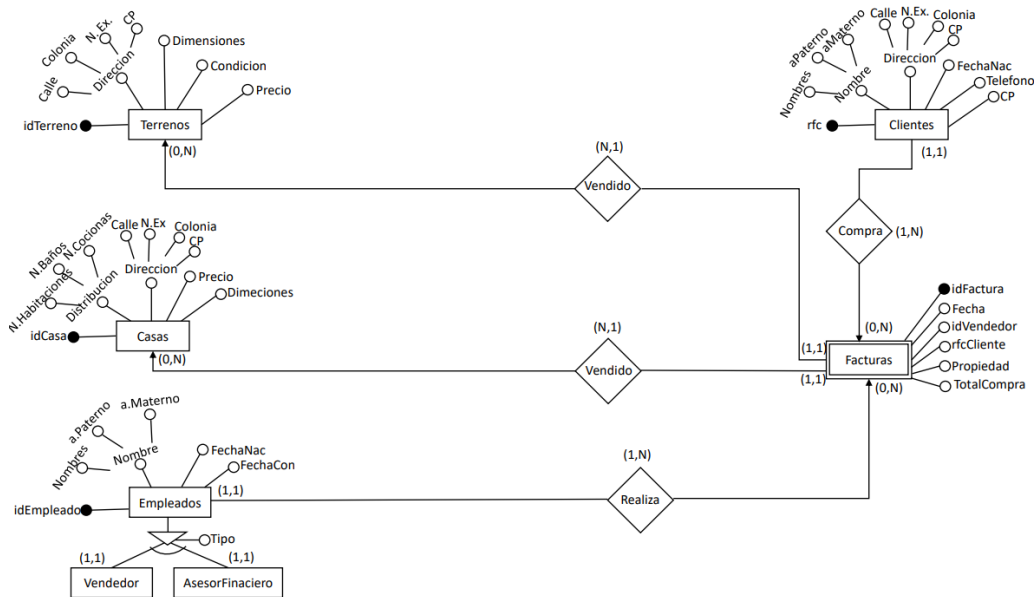


Figura 21. Modelo entidad-relación.

Fuente: Propia (2023).

5.18.4 Diagrama de actividades

Dentro del diagrama de actividades se muestran los procesos del software como un flujo de control o de los objetos haciendo énfasis en la secuencia. Estos procesos los puede llevar a cabo una persona, el propio software o un equipo, en la siguiente figura se muestra un ejemplo de un diagrama de actividades para descargar una canción de una página web, en el cual intervienen tanto el usuario como el sistema, ver figura 22.

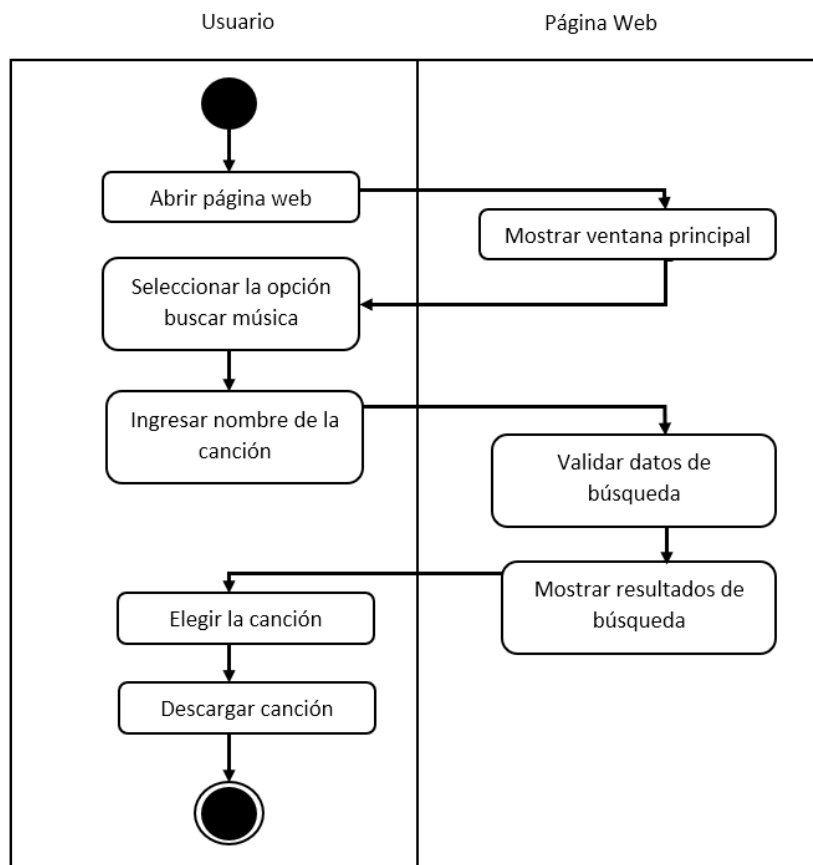


Figura 22 Diagrama de actividades.

Fuente: Propia (2023).

5.18.5 Diagrama de flujo de datos

Dentro del diagrama de flujo de datos se hace una representación gráfica destacando los movimientos de la información, así como los pasos para completar un evento en específico, en la siguiente figura se muestra un ejemplo de un diagrama de flujo de datos para realizar el registro de un nuevo usuario dentro de una página web, ver figura 23.

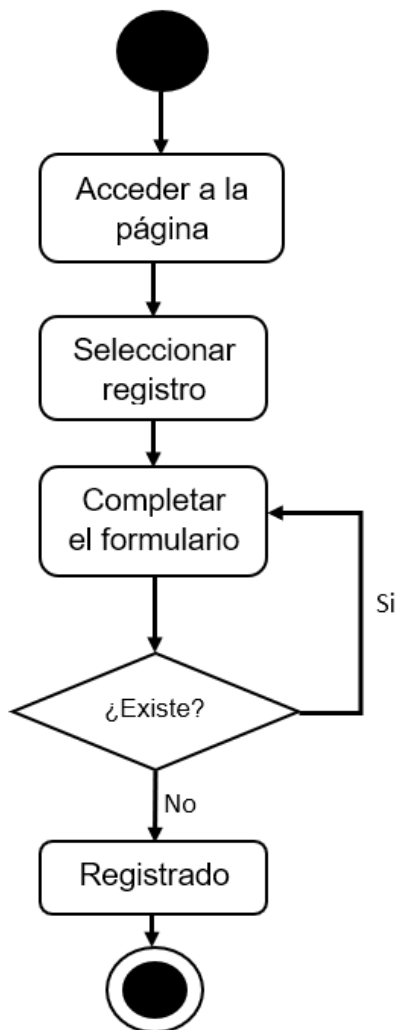


Figura 23 Diagrama de flujo de datos.

Fuente: Propia (2023).

VI. MÉTODOS Y TÉCNICAS DE INVESTIGACIÓN

En este trabajo se aplicará el método de investigación experimental y documental para el diseño del sistema de registro entradas y salidas e inventario, ya que se va a realizar una recopilación de datos que se emplean en los talleres de diseño industrial, así mismo como los tipos de software que emplearemos para desarrollar el sistema.

El motivo por el que se utilizará la investigación documental es porque nos permite obtener información a partir de la recolección y uso de documentos existentes para analizar los datos y ofrecer resultados lógicos de forma ordenada, con una lista de objetivos específicos con el fin de construir nuevos conocimientos, a sí mismo poner a prueba todos los contenidos teóricos fundamentales y conocidos para el correcto diseño del sistema. A continuación, se presenta una definición breve de cada tipo de investigación que se usa en este trabajo:

- La investigación experimental es un proceso que consiste en someter a un objeto o grupo de individuos, a determinadas condiciones para observar los efectos o reacciones que se producen (Arias, 2012, p.34).
- La investigación documental es un proceso basado principalmente en la búsqueda, recuperación, análisis, crítica e interpretación de datos secundarios, es decir, los obtenidos y registrados por otros investigadores en fuentes documentales: impresas, audiovisuales o electrónicas. El propósito de este diseño de investigación es aportar nuevos conocimientos (Arias, 2012, p.27).

En base al diseño del sistema que se desarrollará se considera importante llevar acabo un método analítico de los datos obtenidos, esto con el fin de analizar

los diferentes módulos que pueda haber o partes del sistema que se comprenderá de la causa y los efectos tras el transcurso del diseño del sistema.

Una vez diseñado y estructurado de manera correcta el sistema que se implementará, se utilizará la metodología de cascada (waterfall model) que es un procedimiento lineal que divide los procesos de desarrollo en sucesivas fases de proyecto. Cada una de estas fases se ejecuta tan solo una vez, así mismo los resultados de cada una de las fases sirven como hipótesis de partida para la siguiente (Digital Guide, 2019, párr.2).

Esta metodología no puede cambiarse sobre la marcha. La principal razón es que los requisitos y presupuestos no se pueden cambiar ya que tiene sus fases de desarrollo de manera secuencial, así que es conveniente hacer números antes. Por lo tanto, es una opción válida si se tiene muy claro lo que se quiere desde el principio (Carlemany, 2021, párr.10).

El desarrollo de esta metodología se atribuye al teórico de la informática Winston W. Royce, sin embargo, no es el inventor de este modelo. En su ensayo de 1970 titulado “Managing the Development of Large Software Systems” presenta una crítica acerca de los procedimientos lineales. A modo de alternativa, Royce presenta un modelo iterativo incremental en el que cada una de las fases se basa en la anterior (Digital Guide, 2019, párr.3).

“El modelo en cascada se popularizó a través de la norma estadounidense DoD-STD-2167. Esta norma se basa en una versión extremadamente simplificada del procedimiento desarrollado por Royce “(Digital Guide, 2019, párr.5).

La metodología tiene bien definidas las fases secuenciales, cada fase está debidamente monitorizada como se muestra en la figura 24.

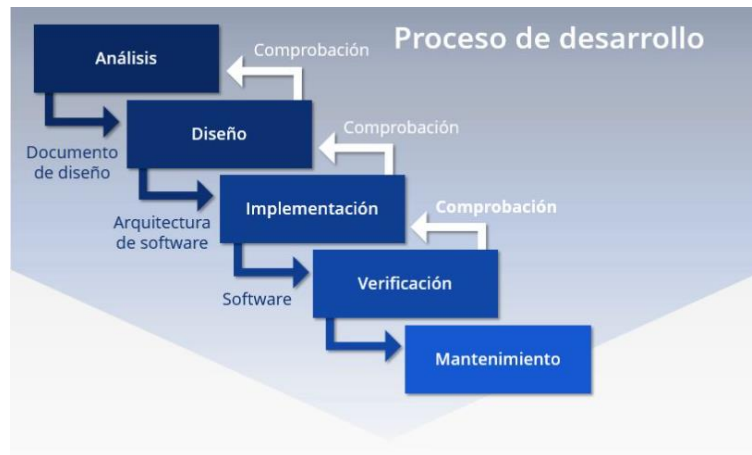


Figura 24. Metodología de cascada.

Fuente: <https://www.ionos.mx/digitalguide/> (2019).

Cada uno de esos procesos tiene un objetivo específico, de manera que el avance será efectivo.

Análisis: Cuando se comienza el desarrollo de software la primera fase es la de análisis que incluye un estudio de viabilidad y una definición de los requisitos. El estudio de viabilidad evalúa los costes, la rentabilidad y la factibilidad del proyecto de software, así como una descripción general de los requisitos (Digital Guide, 2019, párr.9).

Diseño: los desarrolladores de software se encargan de diseñar la arquitectura de software, así como crear un plan de diseño detallado del mismo, centrándose en componentes concretos como lo son, interfaces, entornos de trabajo o bibliotecas (Digital Guide, 2019, párr.12).

Implementación: La arquitectura de software concebida en la fase de diseño se ejecuta en la fase de implementación, en esta fase se incluye la programación del software, la búsqueda de errores y las pruebas unitarias. En la

fase de implementación, el proyecto de software se traduce al correspondiente lenguaje de programación (Digital Guide, 2019, párr.13).

Verificación: Se realiza la integración del software. Los productos de software se envían en primer lugar a los usuarios seleccionados en versión beta (pruebas beta). Aquellos productos de software que superan las pruebas beta están listos para su lanzamiento (Digital Guide, 2019, párr.14).

Mantenimiento: La última fase incluye la entrega del software desarrollado, así como el mantenimiento y la mejora del software (Digital Guide, 2019, párr.15).

VII. PRESENTACIÓN Y DISCUSIÓN DE RESULTADOS

En este capítulo se explican todos los procedimientos que se llevan a cabo, así como las herramientas que se utilizan para el diseño y desarrollo del sistema de control de entradas, salidas e inventario de los talleres de diseño industrial del Centro Universitario Valle de Chalco. Lo primero que se realizó fue obtener los requerimientos del sistema.

7.1 Análisis de los requerimientos

En base a una entrevista con el Licenciado en diseño industrial Cesar Adolfo Muñoz Herrera encargado de los talleres, se planteó como debe de ser el funcionamiento del sistema para que pueda dar solución al problema y al mismo tiempo se obtuvieron los siguientes requerimientos funcionales del sistema web.

7.1.1 Requerimientos funcionales

- Tener una base de datos relacional.
- Ser entendible y amigable para los usuarios finales.
- Al momento de realizar un registro, debe validar que se encuentren llenos los campos necesarios.
- Mostrar alertas para notificar al usuario el estado de la acción que realice.
- Almacenar los datos de los usuarios, permitiendo realizar un registro por cada usuario, identificándolos por su número de cuenta.
- Almacenar los registros de entrada y salida de los usuarios, obteniendo la fecha y hora de manera automática.
- No permitir realizar nuevo registro de entrada si el usuario ya tiene un registro de entrada activo.
- Almacenar la información del inventario de cada taller.

- Generar un código QR para cada herramienta del inventario, al escanear este código mostrar las características de dicha herramienta.
- Ser adaptable, permitiendo que pueda ser funcional en cualquier tipo de pantalla, incluso en dispositivos móviles.
- Contar con una interfaz de administrador, para acceder a esta interfaz se debe disponer de un correo y contraseña específicos.
- Por medio de la interfaz del administrador, debe mostrar en forma de listado los diferentes registros almacenados en la base de datos, permitiendo de igual forma al administrador modificar o eliminar registros.
- Dentro de la interfaz de administrador, permitir realizar la búsqueda de un registro en específico de la base de datos mediante palabras clave.
- Dentro de la interfaz de administrador, permitir exportar los registros en documentos excel, pdf o directamente mandarlos a imprimir.

7.1.2 Requerimientos no funcionales

- No es necesario cifrar la información.
- No es necesario cargar el sistema a un servidor de internet, ya que su funcionamiento será de manera local.
- No requiere un tiempo específico de respuesta.
- No requiere de un navegador en específico para utilizar el sistema.

7.2 Herramientas y tecnologías a utilizar

Considerando los requerimientos funcionales para el desarrollo del sistema de control de entradas, salidas e inventario de los talleres de diseño, se hará uso de las siguientes herramientas y tecnologías:

- NetBeans IDE 8.2.
- Framework SpringMVC.

- Framework Bootstrap.
- Framework jQuery.
- Lenguaje de programación Java.
- Lenguaje de programación HTML.
- Lenguaje de programación Java Script.
- Lenguaje de programación CSS.
- Sweet Alert.
- Código QR.
- Gestor de base de datos MYSQL (MariaDB 10.6.4).
- Lenguaje de consulta estructurado (SQL).
- Servidor local.

7.3 Caso de uso

El diagrama de caso de uso se utiliza para poder visualizar de manera grafica el proceso que se realiza mediante la interacción del usuario con el sistema para poder realizar determinada acción. Dentro del sistema web se identifican a tres tipos de actores(usuarios), los cuales son:

Administrador: El administrador es el actor que puede realizar todas las acciones que el mismo sistema, tiene acceso a toda la información permitiéndole crear, actualizar o eliminar algún registro. Siendo este el actor con todos los permisos del sistema.

Alumno: Este actor puede hacer uso solo de algunas funciones del sistema, entre las cuales tenemos el registro como usuario del sistema, así como los registros cuando ingresa y sale de los talleres.

Profesor: Este actor solo puede hacer uso de las mismas funciones del sistema que el actor anterior (Alumno). En la siguiente figura se muestra el diagrama de caso de uso general del sistema web, ver figura 25.

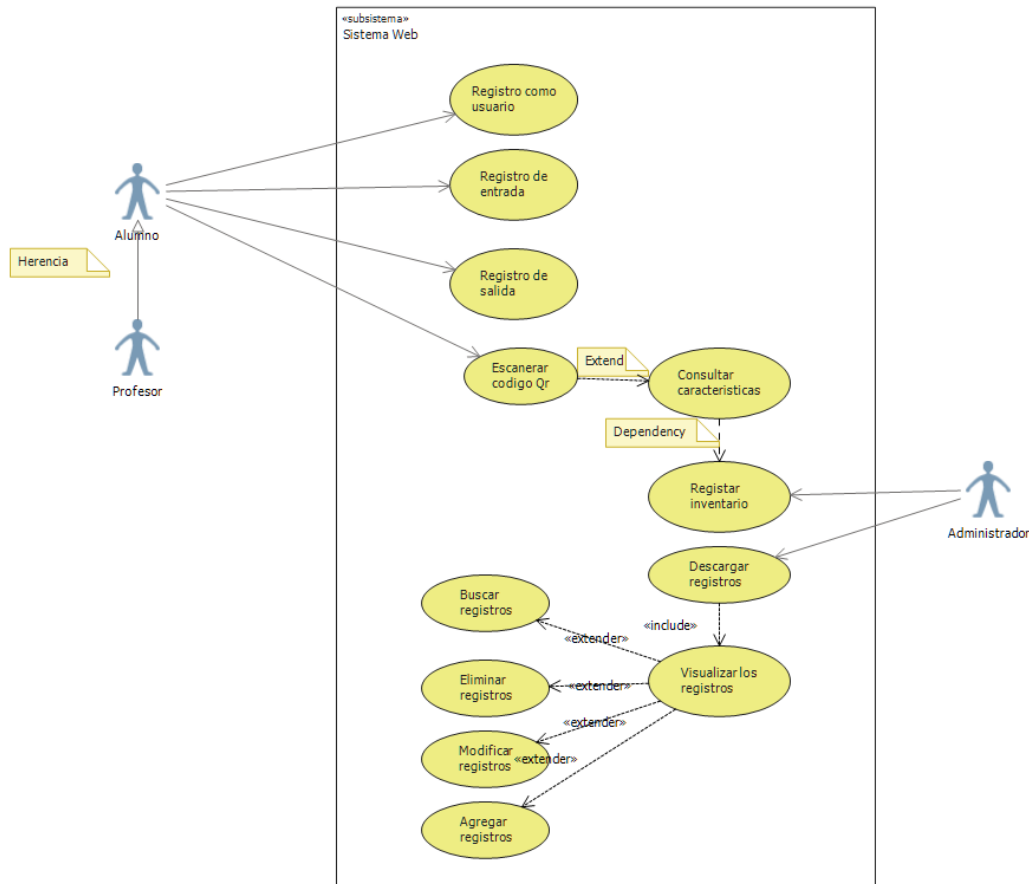


Figura 25. Diagrama de caso de uso.

Fuente: Propia (2023).

Del diagrama de caso de uso general, tenemos diferentes historietas que se derivan del propio caso de uso:

7.3.1 Registrar usuario

En la siguiente tabla se muestra la historietas del caso de uso al momento que un alumno realiza su registro dentro del sistema, ver tabla 3.

Caso de uso: Registro	Analista: Rosendo Vega Mateos
versión: 1.0	Fecha: 23/01/2023
<p>Actor principal: Alumno: requiere estar registrado como usuario para poder registrar su entrada y salida de los talleres.</p> <p>Personal involucrado e intereses: Encargado de los talleres: Requiere los datos generales de los usuarios que ingresen a los talleres.</p> <p>Precondiciones: El alumno accede al sistema.</p> <p>garantías de éxito (Postcondiciones): Se registra el alumno, el registro de sus datos se realiza una sola vez para posteriormente registrar las entradas y salidas que realice a los talleres.</p> <p>Escenario principal de éxito (o Flujo Básico):</p> <ol style="list-style-type: none"> 1. El alumno ingresa a los talleres y accede al sistema. 2. El alumno se dirige al apartado del sistema "Agregar usuario". 3. El alumno agrega sus datos y se registra. 4. El sistema le muestra al alumno que se ha registrado. 	<p>Extensiones (o Flujos Alternativos):</p> <p>a) El alumno no rellena los datos solicitados: Cuando el alumno no rellena los datos solicitados o los rellene con datos erróneos.</p> <ol style="list-style-type: none"> 1. El sistema le mostrará en color rojo el campo que falte por rellenar. 2. El alumno debe rellenar de manera correcta los campos para poder registrarse. <p>b) El alumno se intenta registrar dos o más veces: Cuando un alumno ya se haya registrado anteriormente e intenta hacerlo de nuevo.</p> <ol style="list-style-type: none"> 1. El sistema le mostrará un mensaje indicándole que ya se encuentra registrado ya que este registro se realiza solo una vez. <p>c) El sistema muestra una alerta de usuario ya registrado: Si es la primera vez que el alumno intenta registrarse.</p> <ol style="list-style-type: none"> 1. El alumno debe verificar que su número de cuenta y datos sean correctos. 2. En caso de que sean correctos y el sistema no le permita realizar su registro, el alumno debe acudir con el encargado de los talleres para verificar que alumno se registró utilizando su mismo número de cuenta y modificar o eliminar dicho registro.

Tabla 3. Historieta de caso de uso (registro).

Fuente: Propia (2023).

En la siguiente figura se observa el diagrama de caso de uso en donde se muestra de manera grafica el proceso al momento que un alumno se registra en el sistema, ver figura 26.

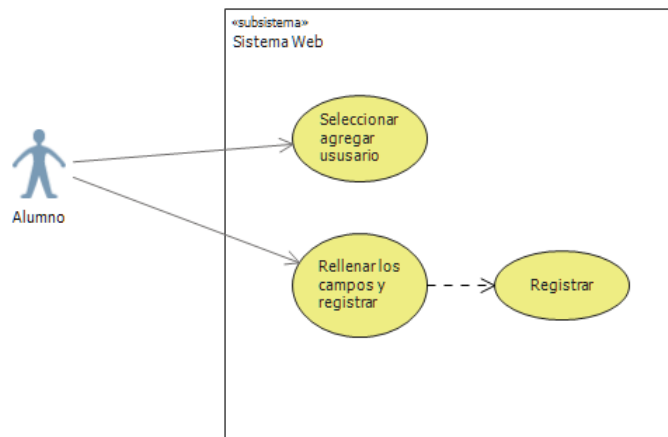


Figura 26. Diagrama de caso de uso (registro).

Fuente: Propia (2023).

7.3.2 Registrar entrada

En la siguiente tabla se muestra la historietta del caso de uso al momento que un alumno registra su entrada a los talleres, ver tabla 4.

Caso de uso: Registrar entrada	Analista: Rosendo Vega Mateos
versión: 1.0	Fecha: 23/01/2023

<p>Actor principal: Alumno: requiere registrar su entrada a los talleres en horarios ajenos a clases.</p> <p>Personal involucrado e intereses: Encargado de los talleres: Requiere saber quiénes ingresan a los talleres, así como a que talleres y que actividades realizan.</p> <p>Precondiciones: El alumno debe acceder y estar previamente registrado en el sistema.</p> <p>garantías de éxito (Postcondiciones): Se registra la entrada del alumno a los talleres.</p> <p>Escenario principal de éxito (o Flujo Básico):</p> <ol style="list-style-type: none"> 1. El alumno ingresa a los talleres y accede al sistema. 2. El alumno se dirige al apartado del sistema “Registra tu entrada”. 3. El alumno agrega los datos que el sistema le solicita y se registra. 4. El sistema muestra mensaje de éxito. 	<p>Extensiones (o Flujos Alternativos):</p> <p>a) El alumno no rellena los datos solicitados: Cuando el alumno no rellena los datos solicitados o los rellene con datos erróneos.</p> <ol style="list-style-type: none"> 1. El sistema le mostrará en color rojo el campo que falte por rellenar. 2. El alumno debe rellenar de manera correcta los campos para poder registrar su entrada. <p>b) El alumno intenta registrar dos o más veces su entrada: Cuando un alumno ya haya registrado su entrada anteriormente e intenta hacerlo de nuevo.</p> <ol style="list-style-type: none"> 1. El sistema le mostrará un mensaje indicándole que se encuentra registrada su entrada y no puede registrar una nueva hasta haber registrado su salida: 2. El alumno debe ir al apartado” Registrar salida”. 3. El alumno debe rellenar los campos y registrar su salida. 4. El alumno ya puede registrar una nueva entrada.
---	--

Tabla 4. Historieta de caso de uso (registrar entrada).

Fuente: Propia (2023).

En la siguiente figura se observa el diagrama de caso de uso en donde se muestra el proceso al momento que un alumno registra su entrada a los talleres, ver figura 27.

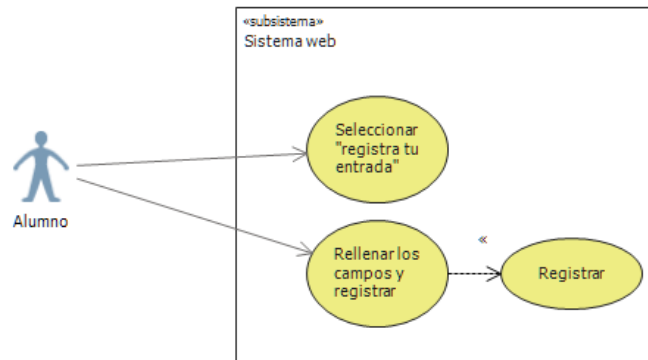


Figura 27. Diagrama de caso de uso (registrar entrada).

Figura: Propia (2023).

7.3.3 Registrar salida

En la siguiente tabla se muestra la historietta del caso de uso al momento que un alumno registra su salida de los talleres, ver tabla 5.

Caso de uso: Registrar salida	Analista: Rosendo Vega Mateos
versión: 1.0	Fecha: 23/01/2023
<p>Actor principal: Alumno: requiere registrar su salida a los talleres en el momento que se retira.</p> <p>Personal involucrado e intereses: Encargado de los talleres: Requiere saber la hora de salida de los usuarios que ingresan a los talleres.</p> <p>Precondiciones: El alumno debe acceder y debe haber registrado previamente su entrada a los talleres.</p> <p>garantías de éxito (Postcondiciones): Se registra la salida del alumno a los talleres.</p>	<p>Extensiones (o Flujos Alternativos):</p> <p>a) El alumno no rellena los datos solicitados: Cuando el alumno no rellena los datos solicitados o los rellene con datos erróneos:</p> <ol style="list-style-type: none"> 1. El sistema le mostrará en color rojo el campo que falte por rellenar. 2. El alumno debe rellenar de manera correcta los campos para poder registrar su salida. <p>b) El alumno intenta registrar dos o más veces su salida:</p> <p>Cuando un alumno ya haya registrado su salida anteriormente e intenta hacerlo de nuevo.</p> <ol style="list-style-type: none"> 1. El sistema le mostrará un mensaje indicándole que no se encuentra registrada su

<p>Escenario principal de éxito (o Flujo Básico):</p> <ol style="list-style-type: none"> 1. El alumno accede al sistema. 2. El alumno selecciona el apartado del sistema "Registra tu salida". 3. El alumno agrega los datos que el sistema le solicita y da clic en salir. 4. El sistema le muestra al alumno que se ha registrado su salida. 	<p>entrada por lo cual no puede registrar su salida.</p> <p>c) El sistema le muestra un mensaje de que no está registrada su entrada:</p> <ol style="list-style-type: none"> 1.El alumno debe verificar que su número de cuenta este escrito correctamente. 2. Si el número es correcto, El alumno debe registrar su entrada para poder registrar su salida.
---	--

Tabla 5. Historieta de caso de uso (registrar salida).

Fuente: Propia (2023).

En la siguiente figura se observa el diagrama de caso de uso en donde se muestra el proceso al momento que un alumno registra su salida de los talleres, ver figura 28.

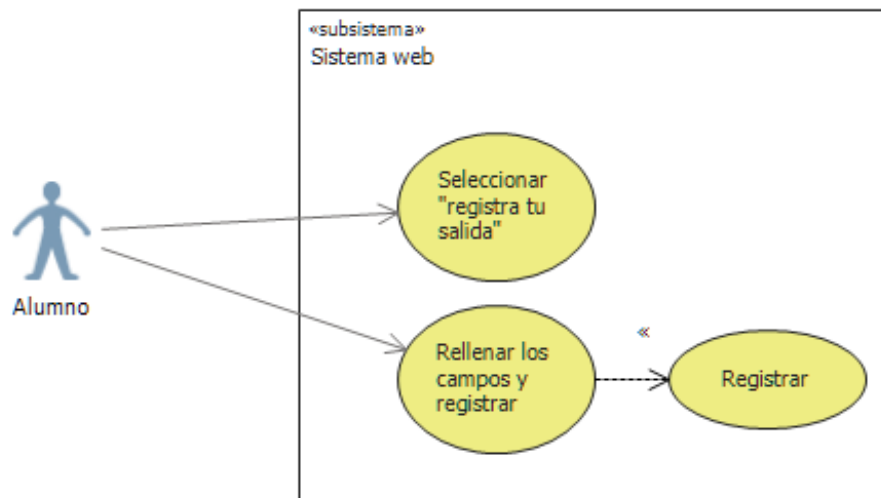


Figura 28. Diagrama de caso de uso (registrar salida).

Figura: Propia (2023).

7.3.4 Escanear código QR

En la siguiente tabla se muestra la historieta del caso de uso al momento que un alumno escanea el código QR, ver tabla 6.

Caso de uso: Escanear código QR	Analista: Rosendo Vega Mateos
versión: 1.0	Fecha: 23/01/2023
<p>Actor principal: Alumno: escanea código QR.</p> <p>Personal involucrado e intereses: Alumno: Quiere ver las características de una maquina en específico.</p> <p>Precondiciones: El dispositivo con el cual el alumno va a realizar el escaneo del código QR debe de estar en la misma red que la computadora que se utiliza como servidor.</p> <p>garantías de éxito (Postcondiciones): Se visualizan las características de cada herramienta registrada en el inventario.</p> <p>Escenario principal de éxito (o Flujo Básico):</p> <ol style="list-style-type: none"> 1. El alumno escanea el código QR con su dispositivo inteligente. 2. El alumno se dirige al enlace proporcionado por el código QR. 3. El alumno visualiza las características de esa máquina. 	<p>Extensiones (o Flujos Alternativos):</p> <p>a) El alumno no puede acceder al enlace que proporciona el código QR:</p> <ol style="list-style-type: none"> 1. El alumno debe verificar que su dispositivo móvil se encuentre en la misma red que el servidor. 2. de no ser así, el alumno no podrá visualizar el contenido del código QR.

Tabla 6. Historieta de caso de uso (Escanear código QR).

Fuente: Propia (2023).

En la siguiente figura se observa el diagrama de caso de uso, en donde se muestra el proceso al momento que el alumno escanea el código QR, ver figura 29.

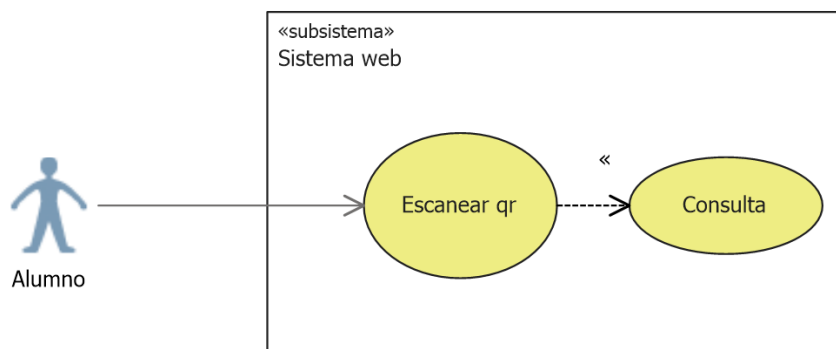


Figura 29. Diagrama de caso de uso (escanear código QR).

Fuente: Propia (2023).

Es importante mencionar que el profesor es un actor que hereda del alumno, por lo tanto, las historietas para realizar su registro, registrar su entrada y registrar su salida son las mismas y llevan los mismos procedimientos.

7.3.5 Visualizar registros

En la siguiente tabla se muestra la historietas del caso de uso al momento que el administrador revisa los registros, ver tabla 7.

Caso de uso: Ver registros	Analista: Rosendo Vega Mateos
versión: 1.0	Fecha: 23/01/2023

<p>Actor principal: Encargado de los talleres: requiere ver los registros de quien ingresa y sale de los talleres.</p> <p>Personal involucrado e intereses: Encargado de los talleres: Requiere ver que personas ingresaron a los talleres, así como la información del inventario.</p> <p>Precondiciones: El Encargado de los talleres debe acceder al sistema.</p> <p>garantías de éxito (Postcondiciones): Se visualizan los registros almacenados en el sistema.</p> <p>Escenario principal de éxito (o Flujo Básico):</p> <ol style="list-style-type: none"> 1. El Encargado accede al sistema. 2. El encargado se dirige al apartado del sistema "Administrar". 3. El encargado debe colocar el usuario y contraseña correctas para poder acceder al modo administrador. 4. El encargado visualiza los registros, permitiéndole agregar nuevos, modificar los ya existentes o eliminarlos, así como descargar dichos registros. 	<p>Extensiones (o Flujos Alternativos):</p> <p>a) El encargado coloca de manera incorrecta el correo o contraseña:</p> <ol style="list-style-type: none"> 1. El sistema le deniega el acceso. 2. El encargado debe escribir de manera correcta el correo y la contraseña. <p>b) El encargado olvida la contraseña o correo:</p> <ol style="list-style-type: none"> 1. El encargado debe revisar la información proporcionada, para revisar el correo y contraseña con los que el sistema permite el acceso. <p>c) El sistema falla:</p> <ol style="list-style-type: none"> 1. El encargado debe cerrar las ventanas que tenga abiertas del sistema 2. El encargado debe correr nuevamente el sistema.
--	---

Tabla 7. Historieta de caso de uso (ver registros).

Fuente: Propia (2023).

En la siguiente figura se observa el diagrama de caso de uso, en donde se muestra el proceso al momento que el administrador visualiza los registros dentro del sistema, ver figura 30.

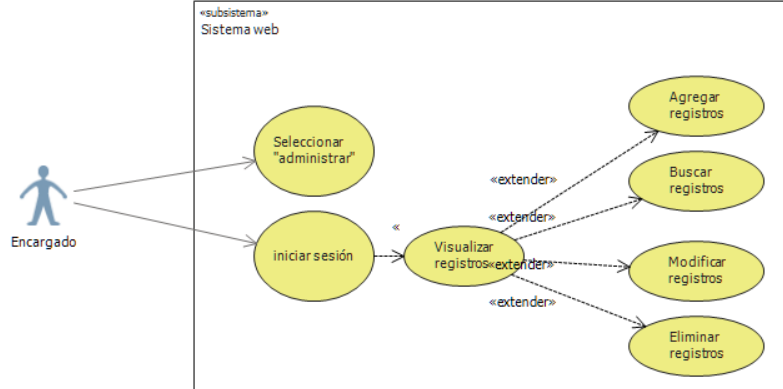


Figura 30. Diagrama de caso de uso (ver registros).

Fuente: Propia (2023).

7.4 Diseño del sistema

La primera parte para desarrollar de nuestro sistema web es el diseño lógico de la base de datos, ya que requiere un análisis bien definido para que no haya problemas al momento de realizar los registros, consultas o modificaciones. De igual manera evitar la redundancia de datos para que el sistema funcione de manera correcta y la base de datos de solución al problema planteado.

7.4.1 Modelo de la base de datos

En la siguiente figura se muestra el diseño lógico de la base de datos relacional, que se implementará en el sistema gestor de base de datos MariaDB 10.6.4 para el desarrollo del sistema web, ver figura 31.

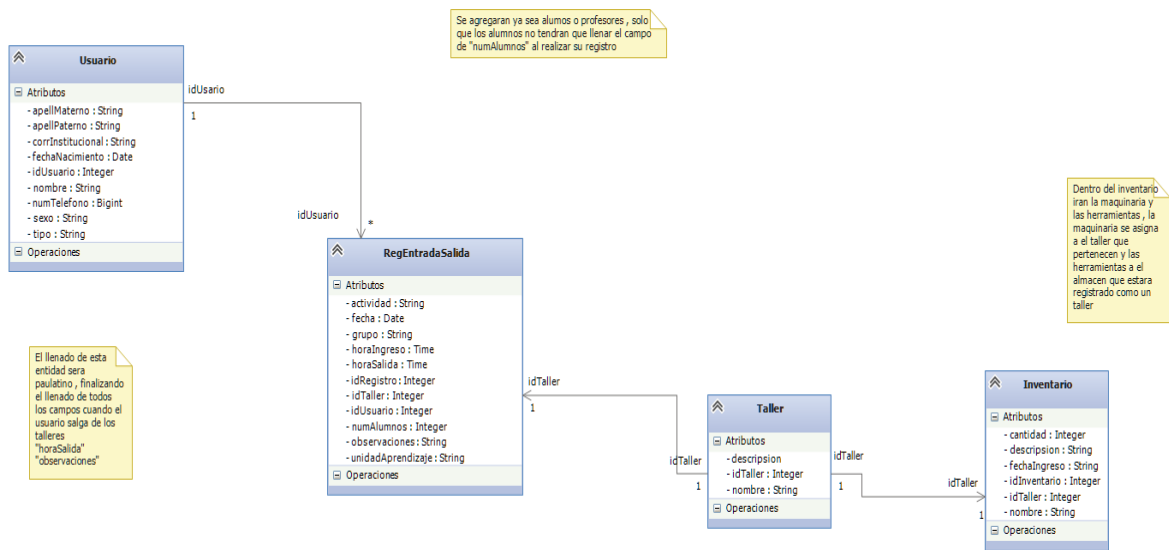


Figura 31. Diagrama entidad-relación.

Fuente: Propia (2023).

Dentro del modelo entidad relación, se pueden observar las relaciones que tienen las entidades, por ejemplo:

La entidad Usuario con la entidad RegEntradaSalida tienen una relación uno a muchos, lo que nos indica que un usuario va a poder realizar muchos registros de entradas y salidas, de igual forma que muchos registros de entrada y salida le pueden pertenecer a un solo usuario.

La entidad RegEntradaSalida con la entidad Taller tienen una relación uno a uno, ya que cada registro de entrada y salida se hará a un taller en específico.

Por último, la entidad inventario con la entidad Taller tienen una relación uno a uno, ya que cada herramienta o máquina del inventario se encuentra dentro de un solo taller, de igual forma cada taller puede tener un número específico de herramientas o máquinas dentro de él.

7.4.2 Descripción de las entidades

A continuación, se realiza la descripción de las entidades utilizadas en el modelo entidad relación que se utiliza en este proyecto, así como de la funcionalidad e importancia de estas:

7.4.2.1 Entidad Usuario

La entidad usuario es de utilidad para que dentro de ella se guarden los datos del usuario que realice su registro, este usuario puede ser tanto profesores como alumnos.

El tipo de usuario se especifica al momento de realizar el registro, el identificador o llave primaria de esta entidad es el idUsuario que en la vista para registrar se muestra como número de cuenta o trabajador. En la siguiente figura se muestra la entidad con sus atributos y el tipo de dato correspondiente a cada uno de ellos, ver figura 32.

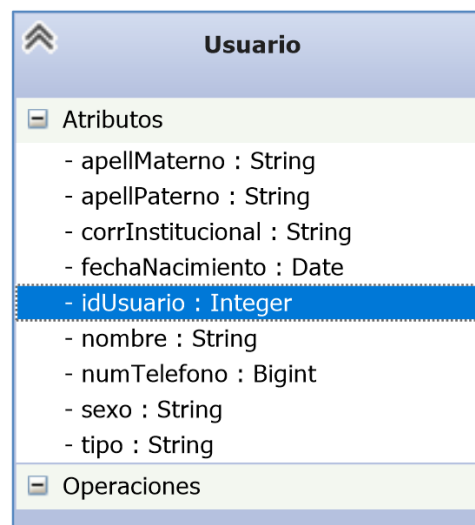


Figura 32. Entidad Usuario.

Fuente: Propia (2023).

7.4.2.2 Entidad RegEntradaSalida

La entidad RegEntradaSalida es de utilidad para que dentro de ella se guarden los registros de entradas y salidas que se realicen a los talleres, esta entidad cuenta con una llave primaria que es el atributo nombrado idRegistro y a su vez cuenta con dos llaves foráneas, una nombrada idUsuario con la cual se relaciona con la entidad Usuario y otra nombrada idTaller con la cual se relaciona con la entidad Taller.

El llenado de esta entidad es paulatino, comenzando cuando el usuario registre su entrada y finalizando cuando registre su salida. Los atributos fecha, horaIngreso, horaSalida los llena el sistema de manera automática, tomando en cuenta los datos de la computadora que realiza la función del servidor.

Esta entidad funciona para registrar la entrada tanto de alumnos como profesores, solo que, a diferencia de los profesores, los alumnos solo deben rellenar con un 1 el atributo numAlumnos. En la siguiente figura se observa la entidad RegEntradaSalida así como sus atributos y los tipos de datos correspondientes a cada uno de estos, ver figura 33.

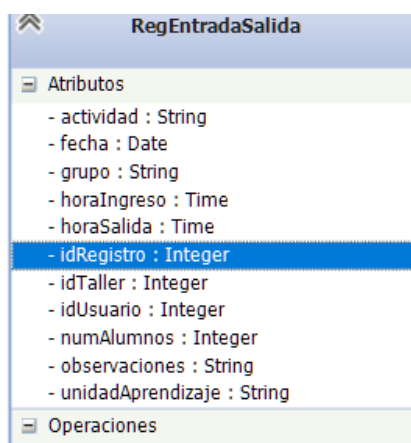


Figura 33. Entidad RegEntradaSalida.

Fuente: Propia (2023).

7.4.2.3 Entidad Taller

La entidad Taller es de utilidad para guardar los diferentes talleres que existen, esta entidad tiene como llave primaria el atributo llamado idTaller con el cual se relaciona con la entidad Inventario y RegEntradaSalida, en la siguiente figura se muestra la entidad Taller con sus atributos y el tipo de dato correspondiente a cada uno de ellos, ver figura 34.

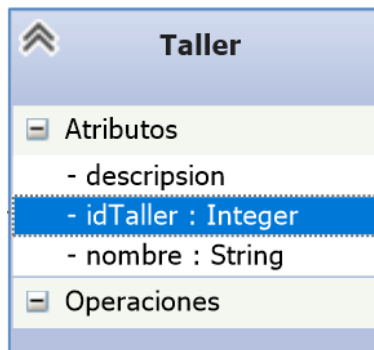


Figura 34. Entidad Taller.
Fuente: Propia (2023).

7.4.2.4 Entidad Inventario

La entidad Inventario es de utilidad para almacenar los datos respecto a las herramientas y máquinas con las que cuentan los talleres, la llave primaria es el atributo nombrado como idInventario, la llave foránea es el atributo nombrado idTaller que sirve para conectar las herramientas o máquinas con el taller al que pertenecen. En la siguiente figura se muestra la entidad Inventario con sus atributos y el tipo de dato correspondiente a cada uno de ellos, ver figura 35.

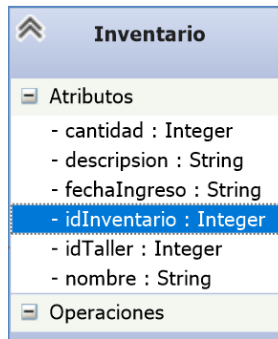


Figura 35. Entidad Inventario.

Fuente: Propia (2023).

7.4.3 Diagrama de secuencia

El diagrama de secuencia nos muestra una idea general de como fluye la información entre los diferentes actores o clases en escenarios específicos, ya sean externos o internos del sistema.

7.4.3.1 Registro

En la siguiente figura se muestra el diagrama de secuencia al momento de que un alumno realiza su registro, ver figura 36.

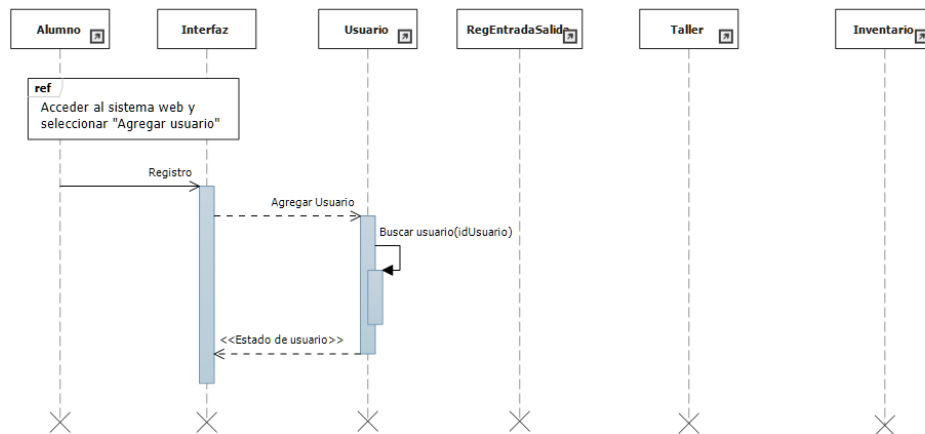


Figura 36. Diagrama de secuencia (registro).

Fuente: Propia (2023).

Se observa como por medio de la interfaz se ejecuta un método para registrar un nuevo usuario, este método antes de agregarlo realiza una validación donde busca en los registros de la entidad usuario si existe un usuario con el mismo idUsuario que el que se intenta agregar, si ya existe un usuario regresara el mensaje que el usuario ya existe, de lo contrario lo registrara. Este mismo principio se utiliza cuando un profesor realiza su registro.

7.4.3.2 Registrar entrada

En la siguiente figura se muestra el diagrama de secuencia al momento de que un alumno realiza su registro de entrada, ver figura 37.

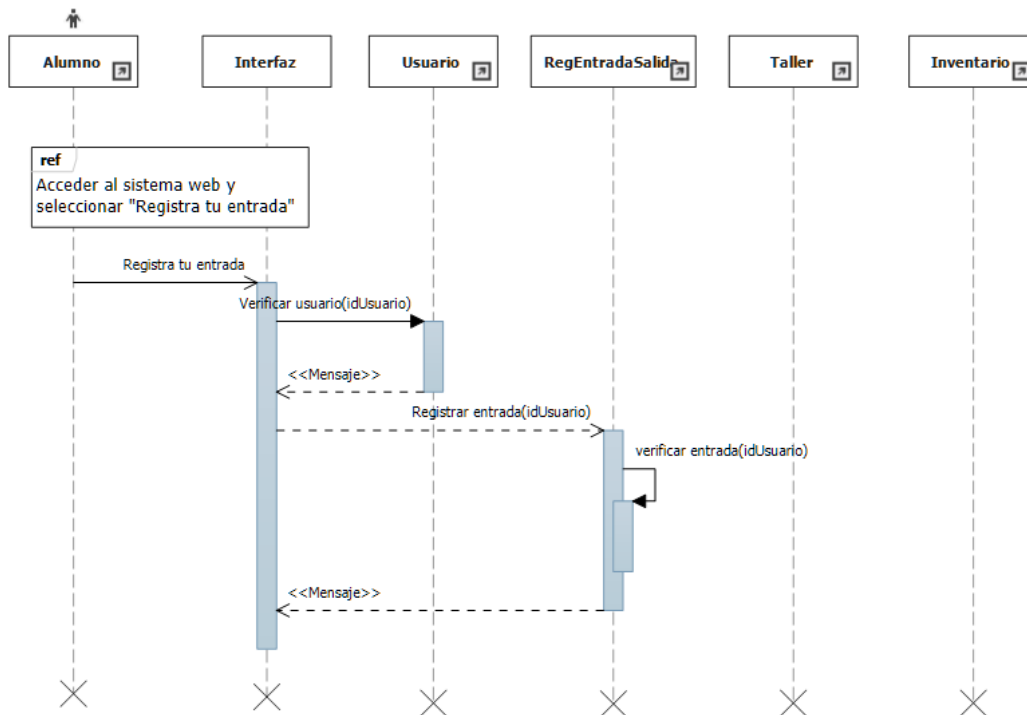


Figura 37. Diagrama de secuencia (registrar entrada).

Fuente: Propia (2023).

Se observa como para que un alumno pueda registrar su entrada primero debe de estar registrado en el sistema, ya que se hace una verificación y si no se encuentra el alumno el sistema regresa un mensaje de error, de lo contrario si el alumno si se encuentra registrado, el sistema realiza otra verificación ahora para saber si el alumno ha registrado una entrada, si es así el sistema regresa un mensaje que el alumno está dentro de los talleres, de lo contrario regresa un mensaje de éxito y registra la entrada a los talleres. Este mismo principio se utiliza cuando un profesor realiza su registro de entrada.

7.4.3.3 Registrar salida

En la siguiente figura se muestra el diagrama de secuencia al momento de que un alumno realiza su registro de salida, ver figura 38.

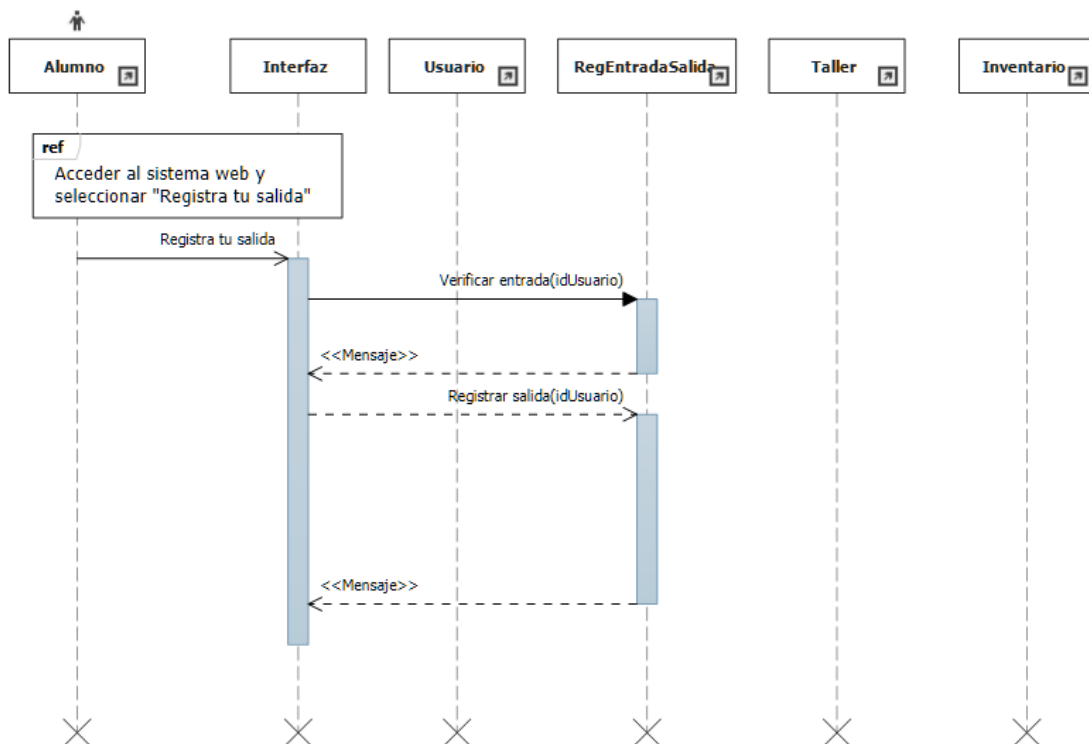


Figura 38. Diagrama de secuencia (registrar salida).

Fuente: Propia (2023).

Se observa como para poder registrar la salida de los talleres, el sistema verifica que anteriormente se haya registrado la entrada del alumno. Si la entrada se encuentra registrada, el sistema registra la salida y manda un mensaje de éxito, de lo contrario el sistema le indica al alumno que debe registrar su entrada primero. Este mismo principio se utiliza cuando un profesor realiza su registro de salida.

7.4.3.4 Ver registros

En la siguiente figura se muestra el diagrama de secuencia al momento de que el administrador visualiza los registros en el sistema, ver figura 39.

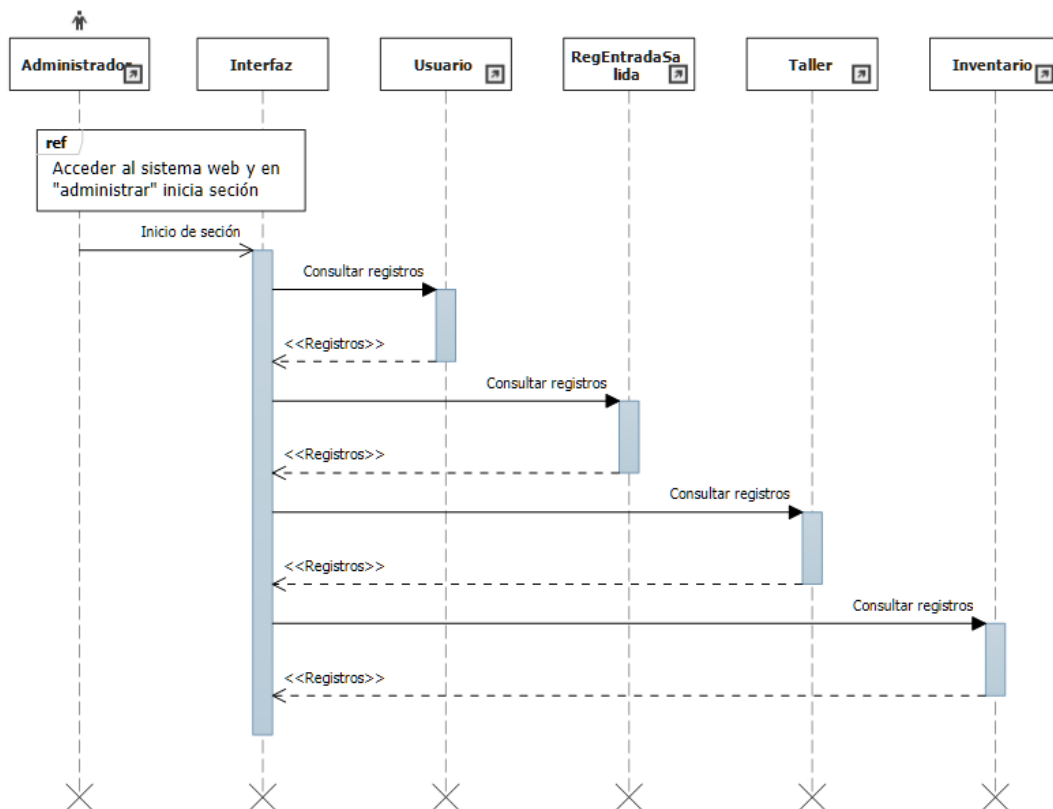


Figura 39. Diagrama de secuencia (ver registros).

Fuente: Propia (2023).

Se observa que mientras el administrador inicie sesión tiene la posibilidad de consultar todos los registros almacenados en el sistema, de igual forma puede agregar nuevos, modificar o eliminar los ya existentes.

7.4.3.5 Escanear código QR

En la siguiente figura se muestra el diagrama de secuencia al momento de que el alumno escanea un código QR de una herramienta o máquina de los talleres, ver figura 40.

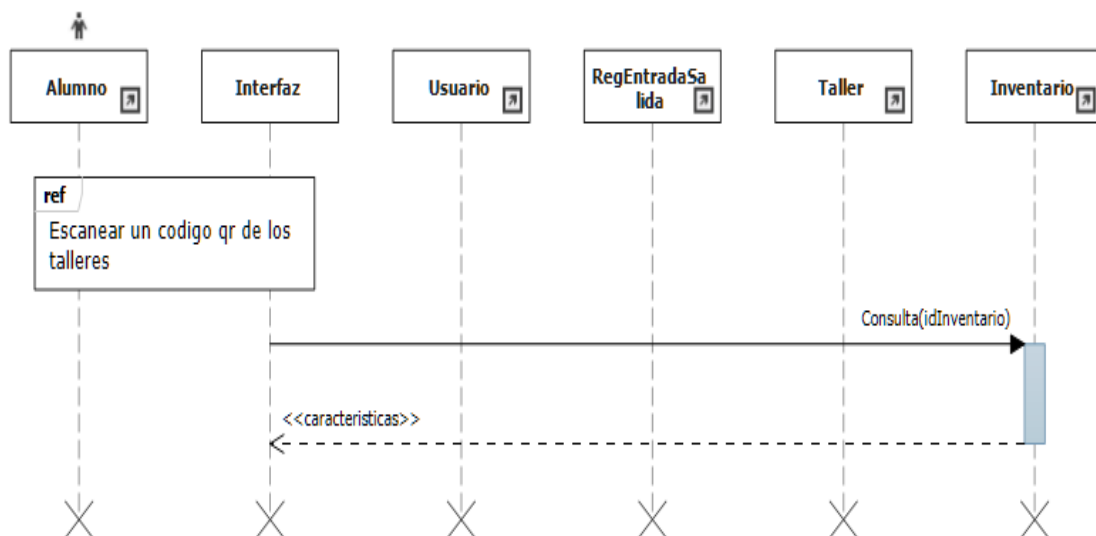


Figura 40. Diagrama de secuencia (escanear código QR).

Fuente: Propia (2023).

Se observa que al momento que el alumno escanea el código QR por medio de algún dispositivo inteligente este realiza una consulta hacia la entidad inventario llevando como dato el idInventario que se encuentra en el QR, regresando las características de dicha herramienta que se haya escaneado.

7.4.4 Diagrama de actividades

Sirve para representar el flujo de las actividades de los usuarios que interactúen con el sistema.

7.4.4.1 Descargar registros

En la siguiente figura se muestran las actividades realizadas por el administrador al momento que descarga los registros almacenados en el sistema. Primero debe de iniciar sesión para posteriormente seleccionar los registros de los cuales quiere descargar la información, a continuación, tiene la posibilidad de buscar ciertos registros o campos o de igual forma descargar todos, para finalizar debe de seleccionar el formato del documento a descargar, de esta forma comenzará la descarga, ver figura 41.

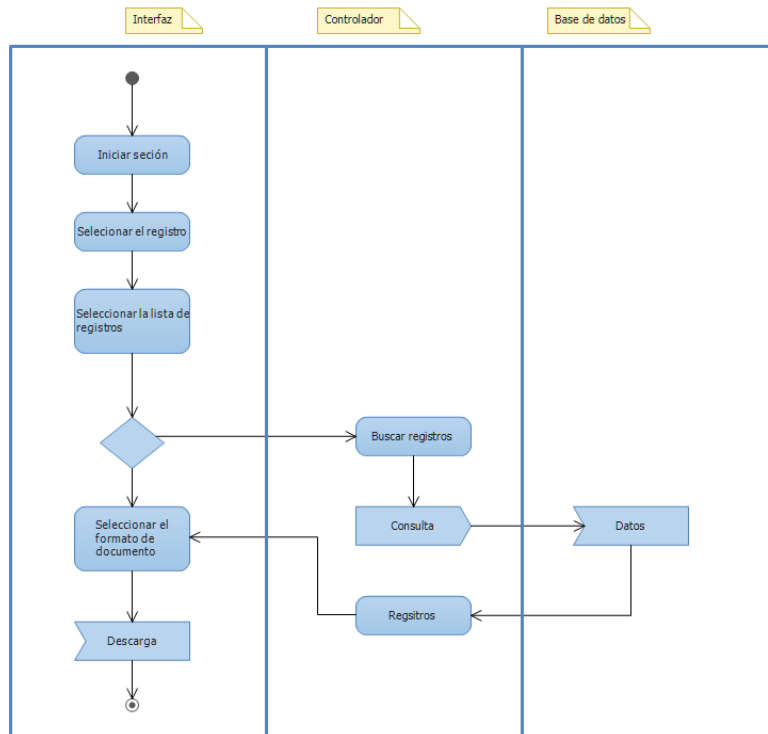


Figura 41. Diagrama de actividades (descargar registros).

Fuente: Propia (2023).

7.4.4.2 Agregar al inventario

En la siguiente figura se muestran las actividades que el administrador realiza para poder agregar un nuevo registro al inventario. Comenzando por iniciar sesión para posteriormente seleccionar el apartado de inventario, en donde debe de rellenar los campos con las características de la herramienta o maquina a registrar, posteriormente el sistema verifica por medio del controlador que no esté registrada esa herramienta o máquina en la base de datos para poder registrarla, ver figura 42.

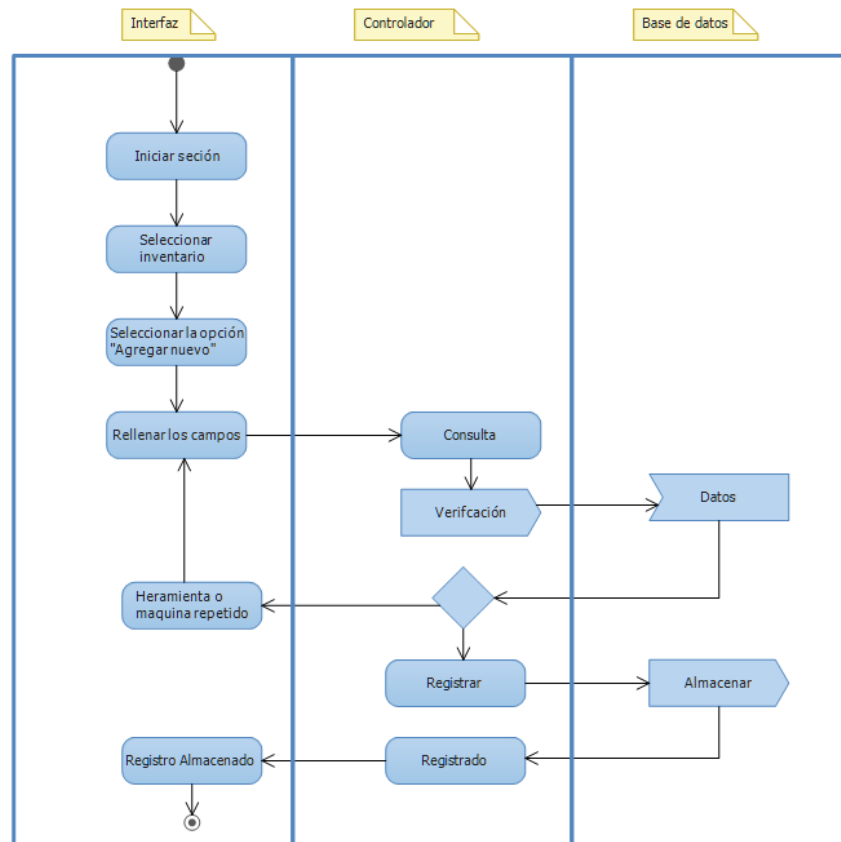


Figura 42. Diagrama de actividades (registrar inventario).

Fuente: Propia (2023).

7.4.4.3 Alerta de registro

En la siguiente figura se muestra el diagrama de actividades al momento que se ejecuta una alerta para realizar un nuevo registro. Se observa como por medio de la interfaz se tiene comunicación con el controlador y este a su vez con la base de datos, para poder verificar que no se repitan los registros y en base a eso mandar la alerta correspondiente, ver figura 43.

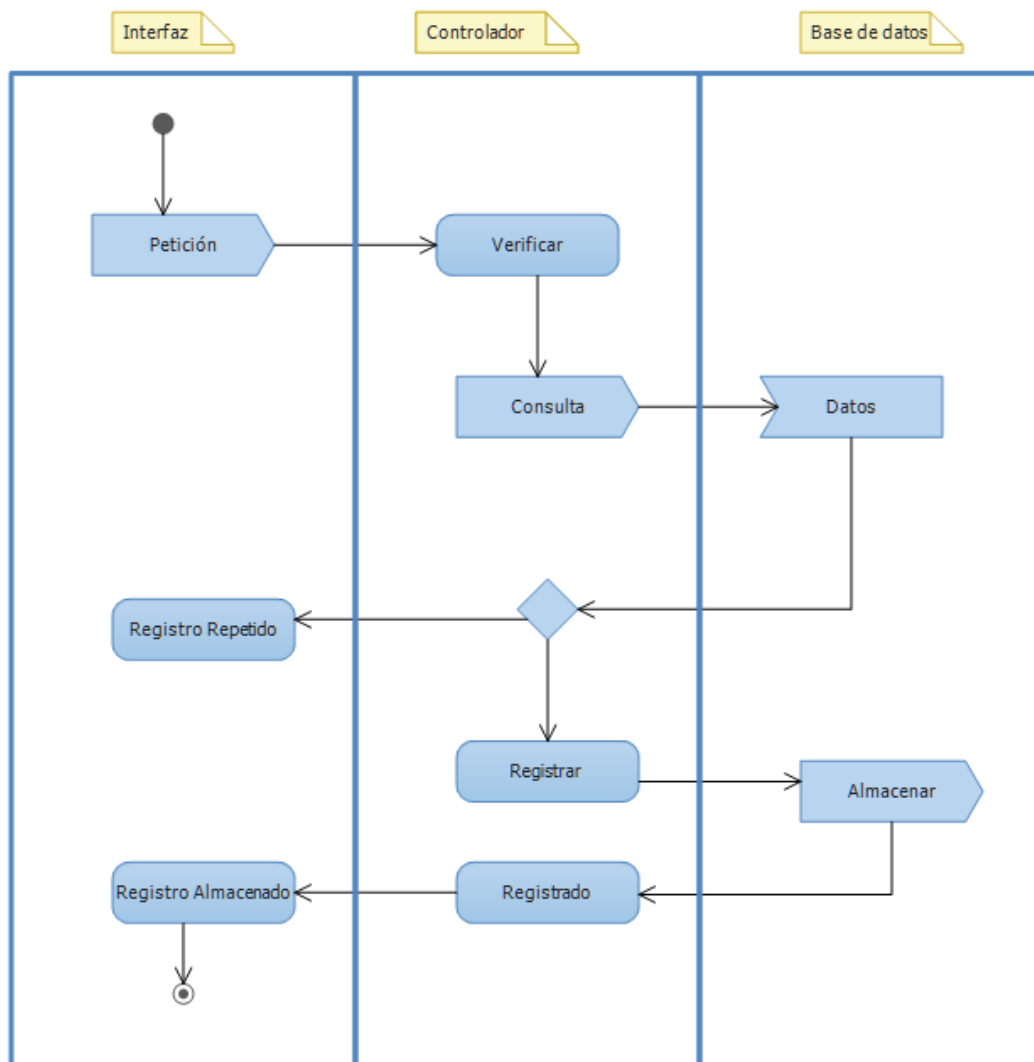


Figura 43. Diagrama de actividades (alerta de registro).

Fuente: Propia (2023).

7.4.5 Diagrama de flujo de datos

El diagrama de flujo de datos nos sirve para identificar las entradas y salidas de los datos del sistema web respecto a los usuarios que lo utilizan, así como para definir la secuencia de procesos en un orden particular.

7.4.5.1 Diagrama padre

En la siguiente figura se muestra el diagrama padre también conocido como diagrama de contexto, donde se pueden observar a los diferentes usuarios que utilizan el sistema web, ver figura 44.

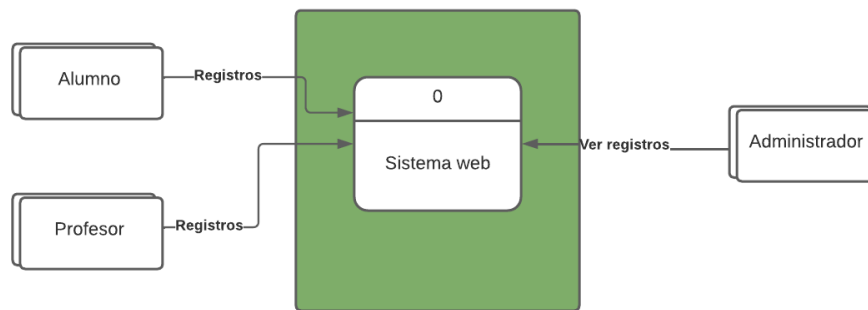


Figura 44. Diagrama de flujo de datos (padre).

Fuente: Propia (2023).

7.4.5.2 Diagrama superior

En la siguiente figura se observa el diagrama de actividades de nivel superior, en el cual podemos observar el flujo que van a tener los datos respecto a las acciones realizadas por los usuarios, este flujo se observa de manera más específica, ver figura 45.

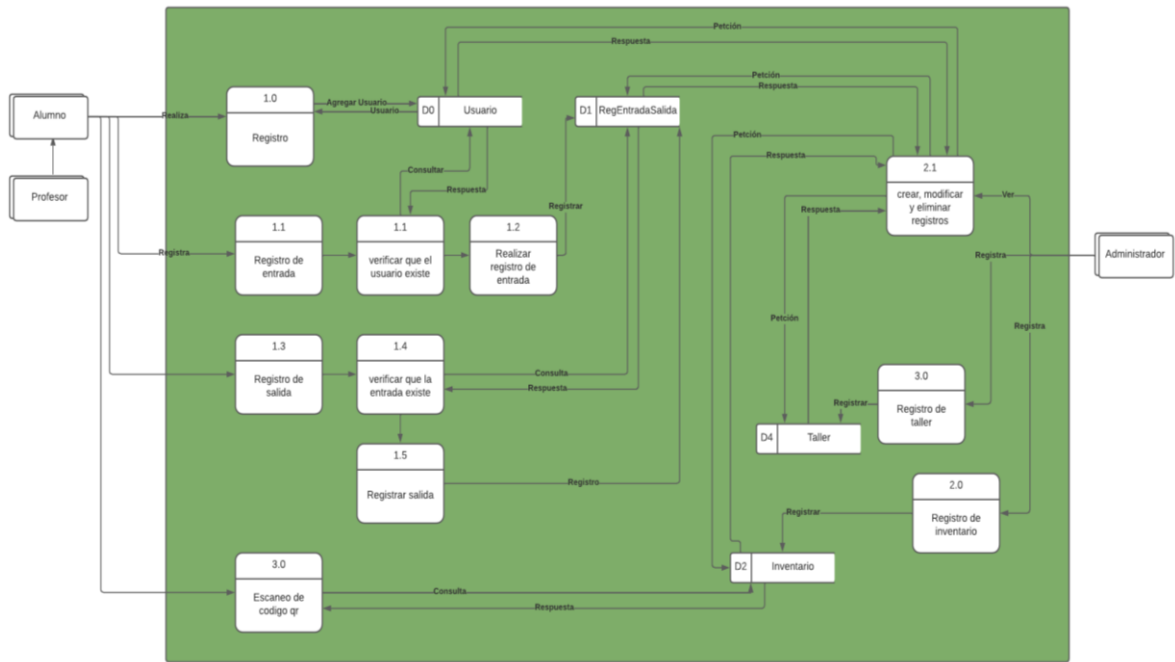


Figura 45. Diagrama de flujo de datos superior.

Fuente: Propia (2023).

7.4.6 Diseño de las vistas

Por último, se realizan una serie de maquetaciones de las vistas, estas maquetaciones se crean utilizando la herramienta Balsamiq Mockup 4.6.5, de esta manera creando las vistas de las principales funciones del sistema (Balsamiq, s. f.).

7.4.6.1 Vista principal

En la siguiente figura se muestra la propuesta diseñada para la vista principal, es decir la ventana que se mostrará al cargar el sistema, ver figura 46.



Figura 46. Maquetación de la vista principal.

Fuente: Propia (2023).

7.4.6.2 Registrar usuario

En la siguiente figura se muestra la propuesta diseñada para la vista al momento de registrar un usuario, ver figura 47.

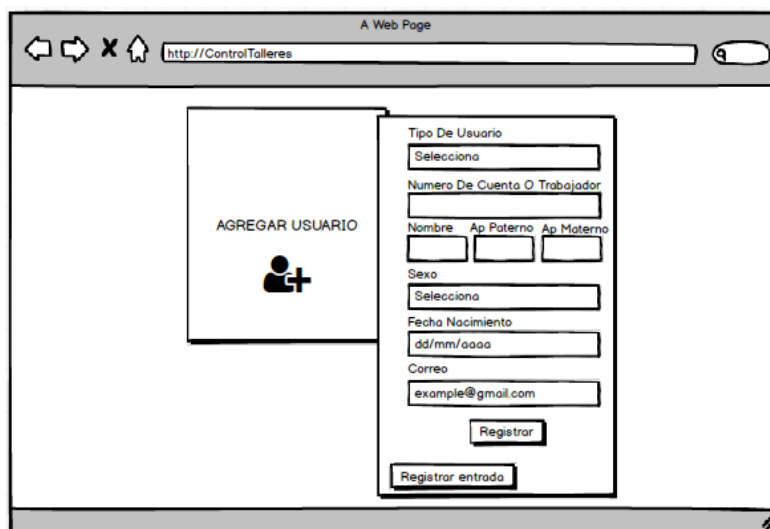
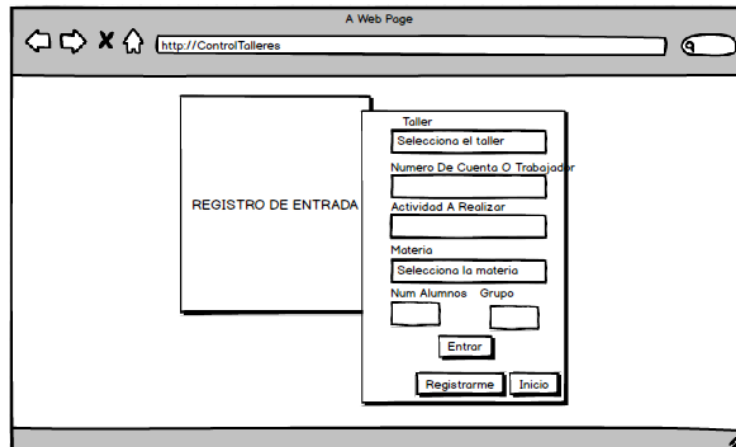


Figura 47. Maquetación de la vista registrar alumno.

Fuente: Propia (2023).

7.4.6.3 Registrar entrada

En la siguiente figura se muestra la propuesta diseñada para la vista al momento de registrar una entrada a los talleres, ver figura 48.



La imagen muestra una maqueta de una interfaz web para registrar una entrada. El navegador muestra la URL 'http://ControlTalleres'. El formulario principal, titulado 'REGISTRO DE ENTRADA', contiene los siguientes campos y botones:

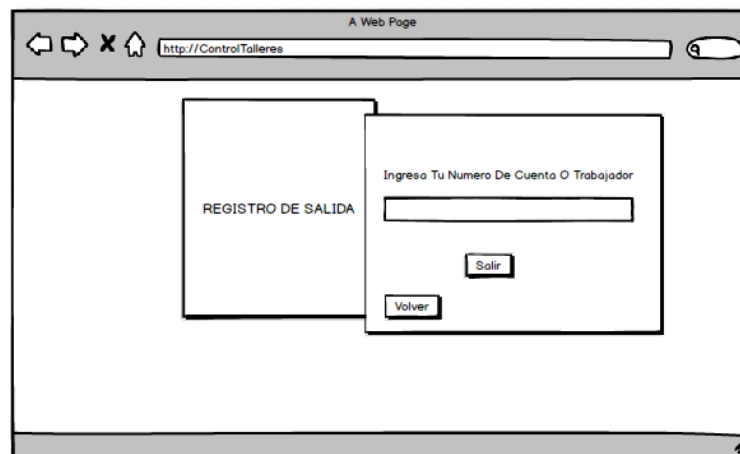
- Taller: Selección el taller (botón)
- Numero De Cuenta O Trabajador (campo de texto)
- Actividad A Realizar (campo de texto)
- Materia: Selección la materia (botón)
- Num Alumnos (campo de texto)
- Grupo (campo de texto)
- Entrar (botón)
- Registrar (botón)
- Inicio (botón)

Figura 48. Maquetación de la vista registrar entrada.

Fuente: Propia (2023).

7.4.6.4 Registrar salida

En la siguiente figura se muestra la propuesta diseñada para la vista al momento de registrar una salida de los talleres, ver figura 49.



La imagen muestra una maqueta de una interfaz web para registrar una salida. El navegador muestra la URL 'http://ControlTalleres'. El formulario principal, titulado 'REGISTRO DE SALIDA', contiene los siguientes campos y botones:

- Ingreso Tu Numero De Cuenta O Trabajador (campo de texto)
- Salir (botón)
- Volver (botón)

Figura 49. Maquetación de la vista registrar salida.

Fuente: Propia (2023).

Una vez que el usuario ingrese su número de cuenta o trabajador, el sistema le solicita que ingrese sus observaciones a los talleres para posteriormente registrar su salida, este proceso se observa en la siguiente maquetación, ver figura 50.

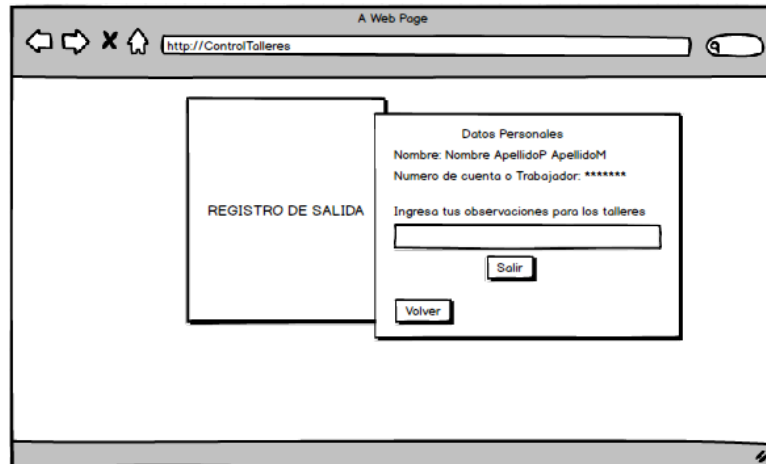


Figura 50. Maquetación de la vista registrar observaciones.

Fuente: Propia (2023).

7.4.6.5 Inicio de sesión del administrador

En la siguiente figura se muestra la propuesta diseñada para la vista al momento de iniciar sesión como administrador, ver figura 51.

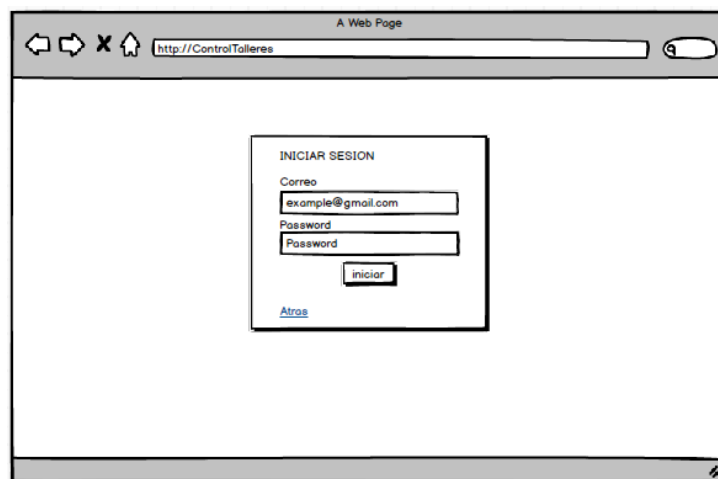


Figura 51. Maquetación de la vista iniciar sesión.

Fuente: Propia (2023).

7.4.6.6 Vista administrador

En la siguiente figura se muestra la propuesta diseñada para la vista al momento de tener la sesión iniciada como administrador, ver figura 52.

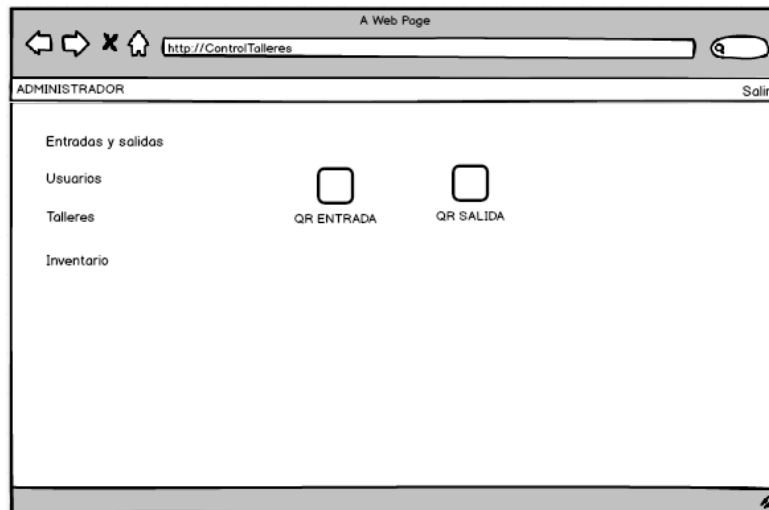


Figura 52. Maquetación de la vista sesión iniciada.

Fuente: Propia (2023).

De igual forma se plantea la vista al momento de que el administrador selecciona un registro en específico para poder ver la lista, ver figura 53.

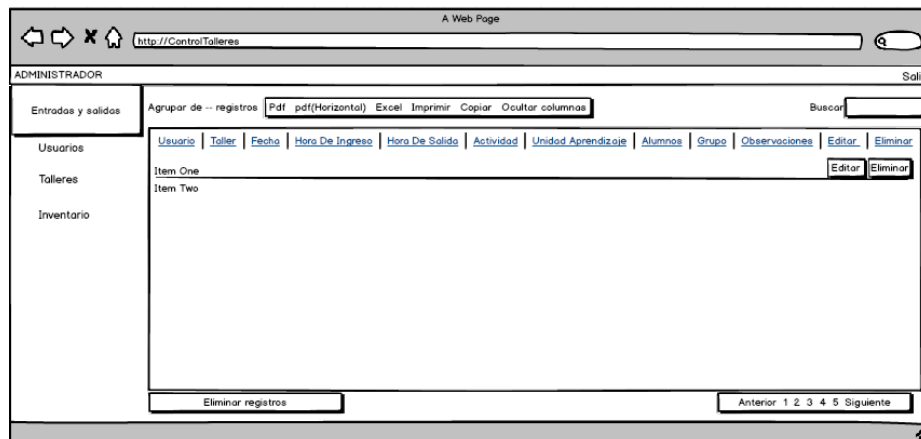


Figura 53. Maquetación de la vista ver lista de registros.

Fuente: Propia (2023).

Asimismo, se plantea la vista cuando el administrador selecciona la opción editar de algún registro en específico de la lista de registros, ver figura 54.

ADMINISTRADOR

Salir

Entradas y salidas

Usuarios

Talleres

Inventario

Editar Registro

Taller

Taller

Numero De Cuenta O Trabajador

11111

Actividad A Realizar

Tarea

Materia

Selecciona la materia

Num Alumnos

1

Grupo

13

Actualizar

Regresar

Figura 54. Maquetación de la vista editar registro.

Fuente: Propia (2023).

7.5 Implementación (Desarrollo del sistema)

En cuanto al desarrollo del proyecto se hace uso del IDE (entorno de desarrollo integrado) NetBeans IDE 8.2, de igual forma se utiliza el patrón MVC (modelo-vista-controlador) que nos permite desarrollar nuestro sistema por componentes separados para facilitar su desarrollo.

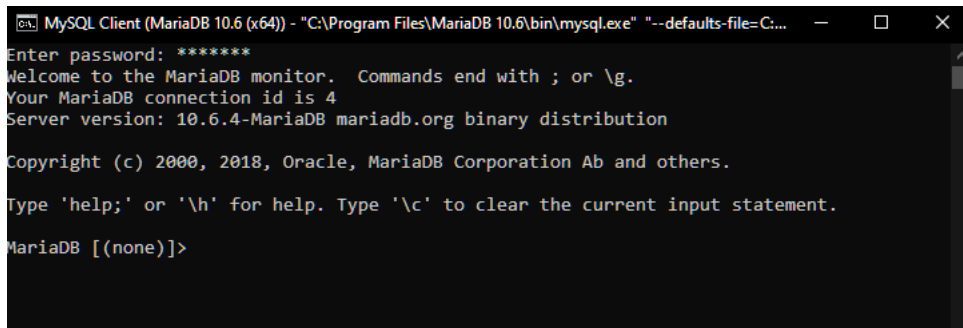
En cuanto a el servidor, se utilizó glassfish Server que viene integrado dentro del IDE NetBeans 8.2, esto para poder visualizar los resultados de nuestro proyecto.

7.5.1 Modelo

El modelo es el encargado de encapsular los datos del sistema web, por lo tanto, dentro del modelo se realiza la creación de la base de datos

7.5.1.1 Base de datos

La siguiente figura muestra el sistema gestor de base de datos maríaDB 10.6.4 que junto con el lenguaje SQL (Lenguaje de consulta estructurado) se utilizan para la implementación del diseño lógico de la base de datos de este proyecto, ver figura 55.



```
MySQL Client (MariaDB 10.6 (x64)) - "C:\Program Files\MariaDB 10.6\bin\mysql.exe" "--defaults-file=C:...
Enter password: *****
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 4
Server version: 10.6.4-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 55. Maria DB 10.6.4.

Fuente: Propia (2023).

El código en lenguaje SQL se utiliza para la creación de la base de datos. En la siguiente figura se muestra el código utilizado para la creación de la base de datos, ver figura 56.

```

1  Create database talleresDiseno;
2
3  use talleresDiseno;
4
5  create table Usuario(
6  idUsuario int primary key auto_increment,
7  nombre varchar(25),
8  apellPaterno varchar(21),
9  apellMaterno varchar(21),
10 fechaNacimiento date,
11 corrInstitucional varchar(45),
12 sexo char(1),
13 tipo varchar(8));
14
15 Create table Taller(
16 idTaller int primary key auto_increment,
17 nombre varchar(20),
18 descripcion varchar(60));
19
20 Create table Inventario(
21 idInventario int primary key auto_increment,
22 idTaller int,
23 nombre varchar(45),
24 marca varchar(30),
25 cantidad int,
26 fechaIngreso date,
27 descripcion varchar(120),
28 imagen varchar(60),
29 foreign key(idTaller)references Taller(idTaller)
30 on update cascade on delete cascade);
31
32 create table RegEntradaSalida(
33 idRegistro int primary key auto_increment,
34 fecha date,
35 horaIngreso time,
36 horaSalida time,
37 idTaller int,
38 idUsuario int,
39 actividad varchar(45),
40 unidadAprendizaje varchar(70),
41 numAlumnos int,
42 grupo varchar(5),
43 observaciones varchar(70),
44 foreign key(idUsuario)references Usuario(idUsuario)
45 on update cascade on delete cascade,
46 foreign key(idTaller)references Taller(idTaller)
47 on update cascade on delete cascade);

```

Figura 56. Código SQL para la creación de la base de datos.

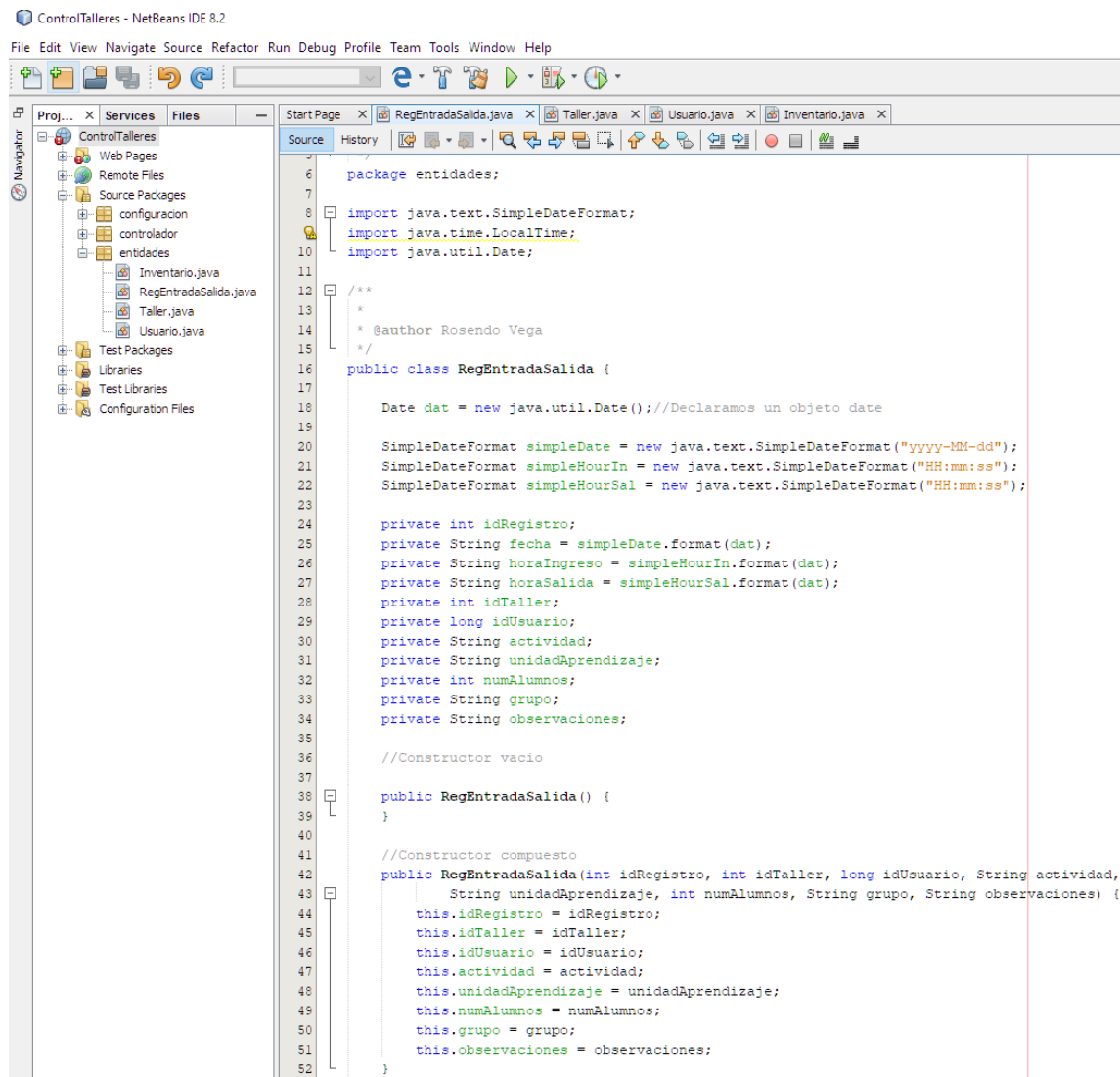
Fuente: Propia (2023).

7.5.1.2 Clases

Dentro del IDE NetBeans 8.2 se crea el proyecto, posteriormente se programan las clases respectivas de la base de datos, estas clases llevan los mismos

nombres y tipos de datos que las entidades de la base de datos para poder realizar acciones dentro de ella.

En la siguiente figura se muestra el código para la creación de la entidad RegEntradaSalida que es la que almacena todos los registros tanto de entrada como salida de los talleres de diseño, ver figura 57.

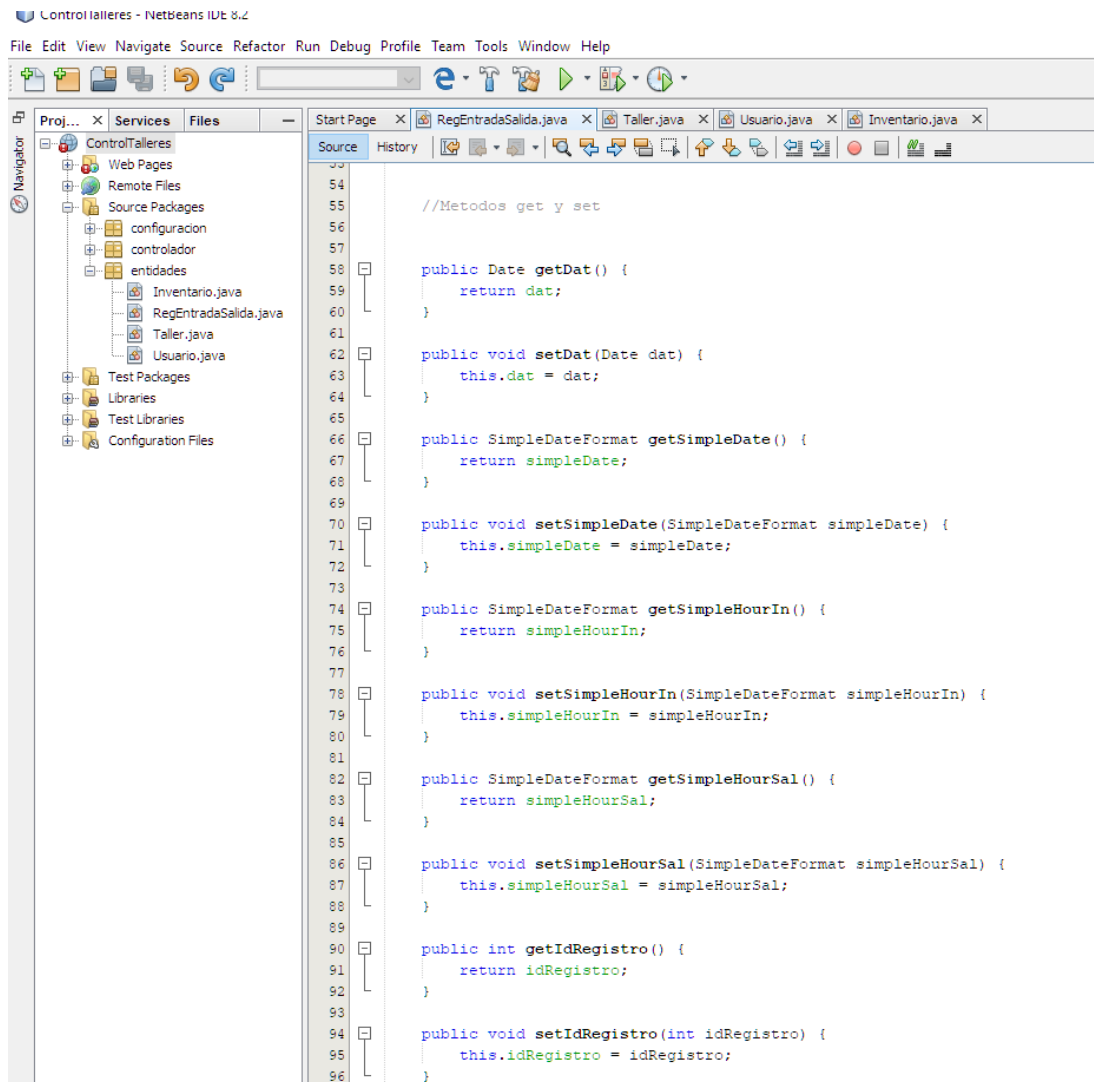


```
ControlTalleres - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
StartPage x RegEntradaSalida.java x Taller.java x Usuario.java x Inventario.java x
Source History
6 package entidades;
7
8 import java.text.SimpleDateFormat;
9 import java.time.LocalDateTime;
10 import java.util.Date;
11
12 /**
13  *
14  * @author Rosendo Vega
15  */
16 public class RegEntradaSalida {
17
18     Date dat = new java.util.Date();//Declaramos un objeto date
19
20     SimpleDateFormat simpleDate = new java.text.SimpleDateFormat("yyyy-MM-dd");
21     SimpleDateFormat simpleHourIn = new java.text.SimpleDateFormat("HH:mm:ss");
22     SimpleDateFormat simpleHourSal = new java.text.SimpleDateFormat("HH:mm:ss");
23
24     private int idRegistro;
25     private String fecha = simpleDate.format(dat);
26     private String horaIngreso = simpleHourIn.format(dat);
27     private String horaSalida = simpleHourSal.format(dat);
28     private int idTaller;
29     private long idUsuario;
30     private String actividad;
31     private String unidadAprendizaje;
32     private int numAlumnos;
33     private String grupo;
34     private String observaciones;
35
36     //Constructor vacio
37
38     public RegEntradaSalida() {
39     }
40
41     //Constructor compuesto
42     public RegEntradaSalida(int idRegistro, int idTaller, long idUsuario, String actividad,
43         String unidadAprendizaje, int numAlumnos, String grupo, String observaciones) {
44         this.idRegistro = idRegistro;
45         this.idTaller = idTaller;
46         this.idUsuario = idUsuario;
47         this.actividad = actividad;
48         this.unidadAprendizaje = unidadAprendizaje;
49         this.numAlumnos = numAlumnos;
50         this.grupo = grupo;
51         this.observaciones = observaciones;
52     }
}
```

Figura 57. Creación de entidades.

Fuente: Propia (2023).

De igual forma se crean los métodos getter y setter, estos métodos sirven para mostrar o modificar el valor de los atributos de la clase, por lo tanto, debe haber un método getter y setter por cada atributo de nuestra entidad, ver figura 58.



```
54
55 //Metodos get y set
56
57
58 public Date getDat() {
59     return dat;
60 }
61
62 public void setDat(Date dat) {
63     this.dat = dat;
64 }
65
66 public SimpleDateFormat getSimpleDate() {
67     return simpleDate;
68 }
69
70 public void setSimpleDate(SimpleDateFormat simpleDate) {
71     this.simpleDate = simpleDate;
72 }
73
74 public SimpleDateFormat getSimpleHourIn() {
75     return simpleHourIn;
76 }
77
78 public void setSimpleHourIn(SimpleDateFormat simpleHourIn) {
79     this.simpleHourIn = simpleHourIn;
80 }
81
82 public SimpleDateFormat getSimpleHourSal() {
83     return simpleHourSal;
84 }
85
86 public void setSimpleHourSal(SimpleDateFormat simpleHourSal) {
87     this.simpleHourSal = simpleHourSal;
88 }
89
90 public int getIdRegistro() {
91     return idRegistro;
92 }
93
94 public void setIdRegistro(int idRegistro) {
95     this.idRegistro = idRegistro;
96 }
```

Figura 58. Métodos getter y setter.

Fuente: Propia (2023).

La creación de las clases se realiza por cada entidad dentro de nuestra base de datos, así como sus respectivos métodos getter y setter.

7.5.1.3 Conexión con la base de datos

Por otra parte, se realiza la conexión del proyecto con el sistema gestor de base de datos, esto con la finalidad de poder acceder a la base de datos y realizar acciones sobre ella, como lo son consultar, insertar, modificar y eliminar sus registros.

Se agregó la librería “MySQL JDBC Driver” a el proyecto y posteriormente se programó la clase que permita la comunicación con el gestor y la base de datos, ver figura 59.

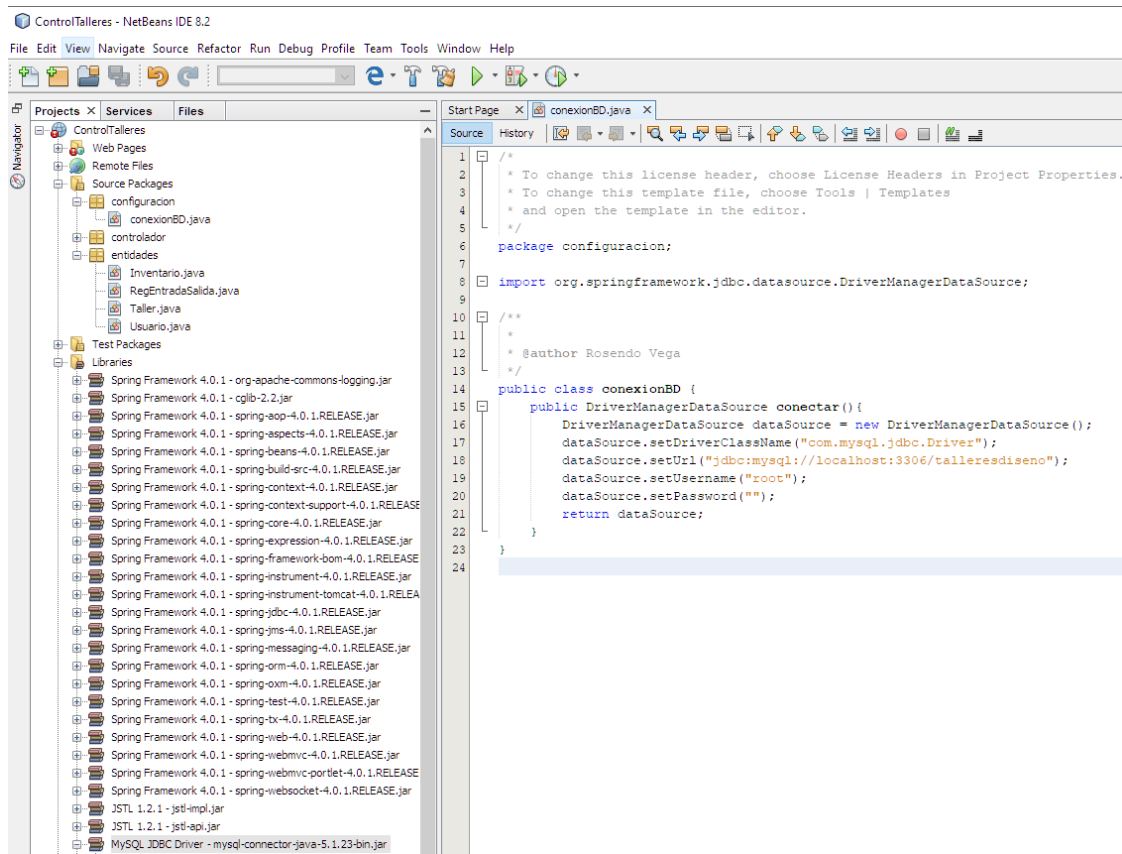


Figura 59. Conexión con la base de datos.

Fuente: Propia (2023).

7.5.2 Vista

Son todas las ventanas planteadas en la maquetación, que sirven para representar los datos del modelo al usuario, de esta manera el usuario puede realizar acciones sobre la base de datos de forma más sencilla y entendible.

Haciendo uso del framework Bootstrap, del lenguaje de estilo en cascada (CSS) y del lenguaje de programación JavaScript se programaron las siguientes vistas que sirven para la interacción del usuario con el sistema web y la base de datos.

7.5.2.1 Principal

La vista principal es la que se muestra al momento que se carga el sistema, a partir de aquí el usuario tiene diferentes acciones a realizar, ver figura 60.

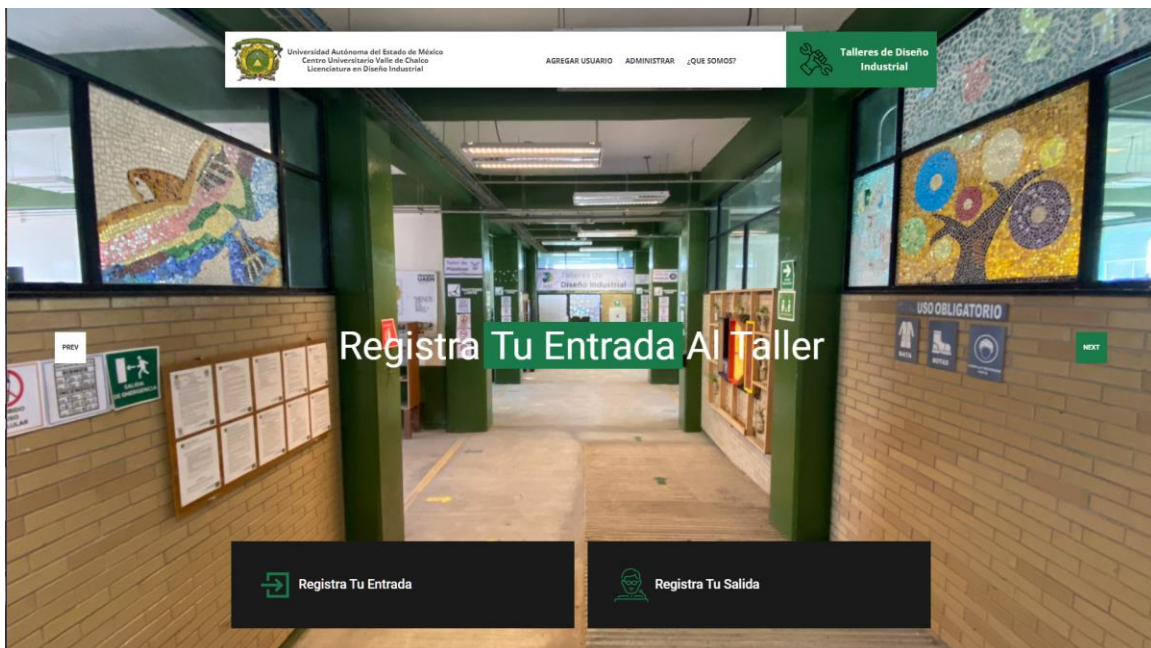


Figura 60. Vista principal.

Fuente: Propia (2023).

7.5.2.2 ¿Qué somos?

La vista ¿qué somos? muestra el objetivo y la misión de la licenciatura en diseño industrial, que es la licenciatura a la cual va dirigido el sistema, ver figura 61.



Figura 61. Vista ¿qué somos? .

Fuente: Propia (2023).

7.5.2.3 Agregar usuario

Al momento que el usuario de clic en el apartado de agregar usuario se le muestra la siguiente vista, en la cual debe registrar sus datos personales para quedar registrado dentro del sistema.

Si el usuario no rellena un campo y quiere realizar su registro, el sistema le muestra el campo que falta por llenar o en su defecto el que este completado de manera incorrecta para que sea completado de manera adecuada y así poder realizar su registro al sistema.

Asimismo, si el usuario intenta realizar su registro dos o más veces el sistema le mostrará una alerta indicándole que ya se encuentra registrado, ver figura 62.

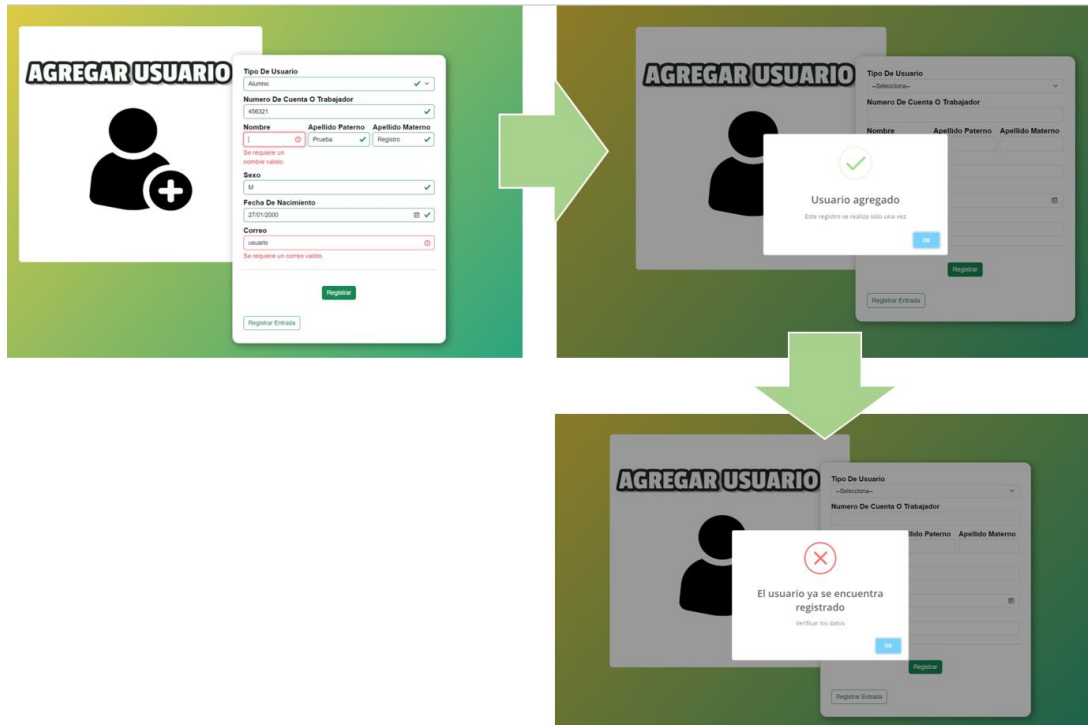


Figura 62. Vistas al agregar un usuario.

Fuente: Propia (2023).

7.5.2.4 Registra tu entrada

Cuando el usuario seleccione la opción de registrar su entrada se le muestra la siguiente vista, en la cual debe registrar sus datos, al igual que la vista anterior si no se completan los campos de manera correcta el sistema no deja continuar hasta que se coloquen de manera adecuada, de igual manera no deja que se registre una entrada sino esta registrado el usuario dentro del sistema, o si ya se ha registrado una entrada anteriormente, ver figura 63.



Figura 63. Vistas del registro de entrada.

Fuente: Propia (2023).

7.5.2.5 Registra tu salida

Al momento que el usuario seleccione la opción de registrar su salida se muestra la vista donde debe colocar su número de cuenta, para poder registrar su salida, debe de estar previamente registrada su entrada, de lo contrario el sistema no permitirá dicho registro.

Si la entrada se encuentra registrada, el sistema solicitará que el usuario realice algunas observaciones a cerca de su experiencia dentro de los talleres, ver figura 64.



Figura 64. Vistas del registro de salida.

Fuente: Propia (2023).

7.5.2.6 Iniciar sesión como administrador

La siguiente figura muestra la vista que solicita el correo y la contraseña para poder iniciar sesión como administrador, si el correo o la contraseña son incorrectos el sistema le denegará el acceso, por el contrario, cargará la vista del administrador, ver figura 65.

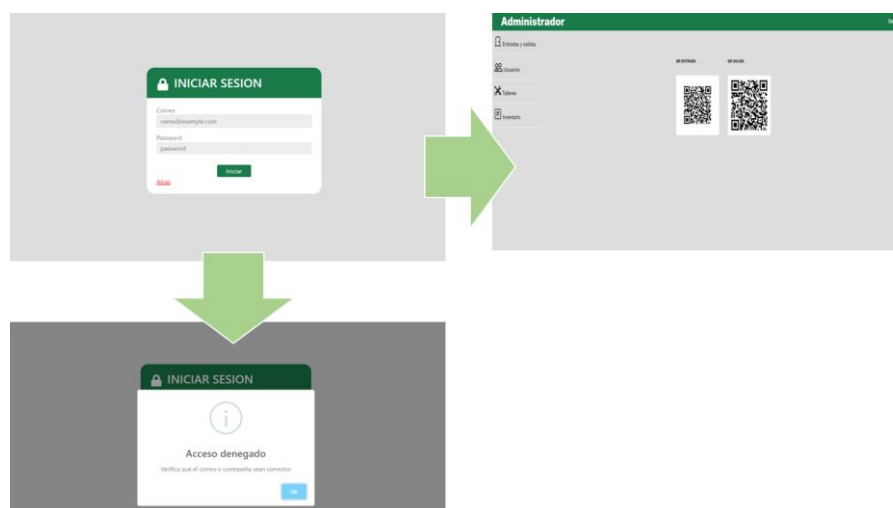


Figura 65. Vistas al Iniciar sesión como administrador.

Fuente: Propia (2023).

7.5.2.7 Administrador

En la parte del administrador se encuentran las vistas mediante las cuales se pueden hacer modificaciones a los registros, como agregar, modificar o eliminarlos.

En la siguiente figura se muestran estas operaciones sobre la entidad Taller, siendo posible realizarlas sobre las demás entidades, ver figura 66.

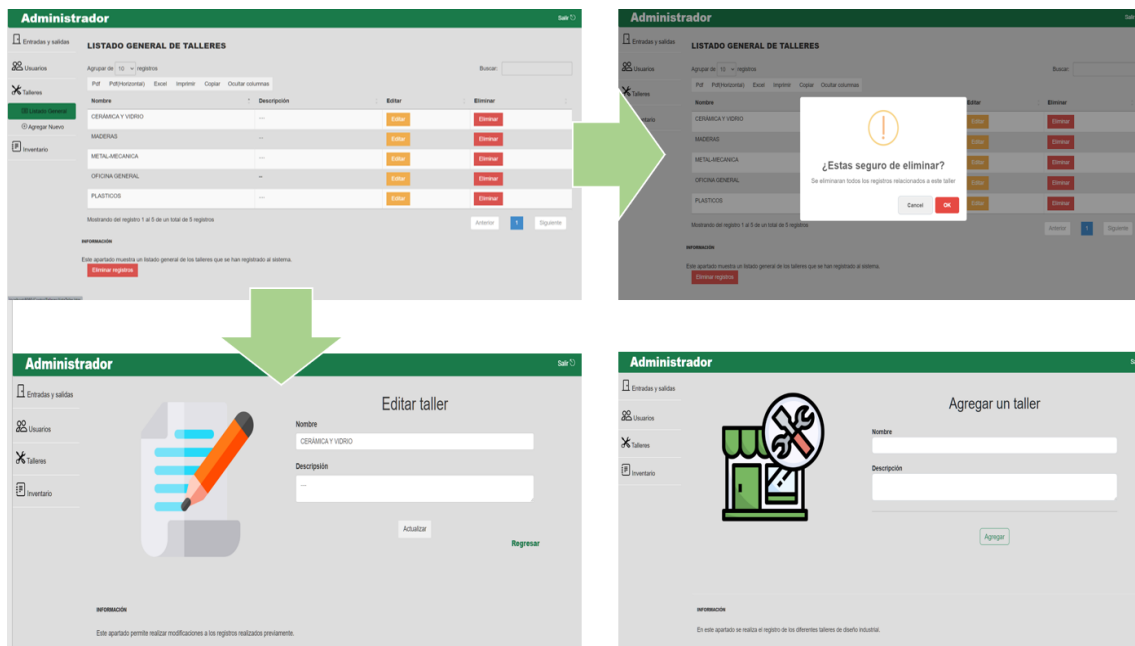


Figura 66. Vistas al agregar, editar y eliminar registros.

Fuente: Propia (2023).

7.5.2.8 Descargar registros

Para poder descargar un registro, desde el administrador tenemos que ir al apartado de lista, esta lista puede ser de cualquier entidad ya que sobre cualquier entidad se puede realizar la descarga de dichos registros. Los diferentes formatos en los cuales se pueden descargar dichos registros son excel, pdf, copiarlos o

directamente mandarlos a imprimir como se muestra en la siguiente figura, ver figura 67.

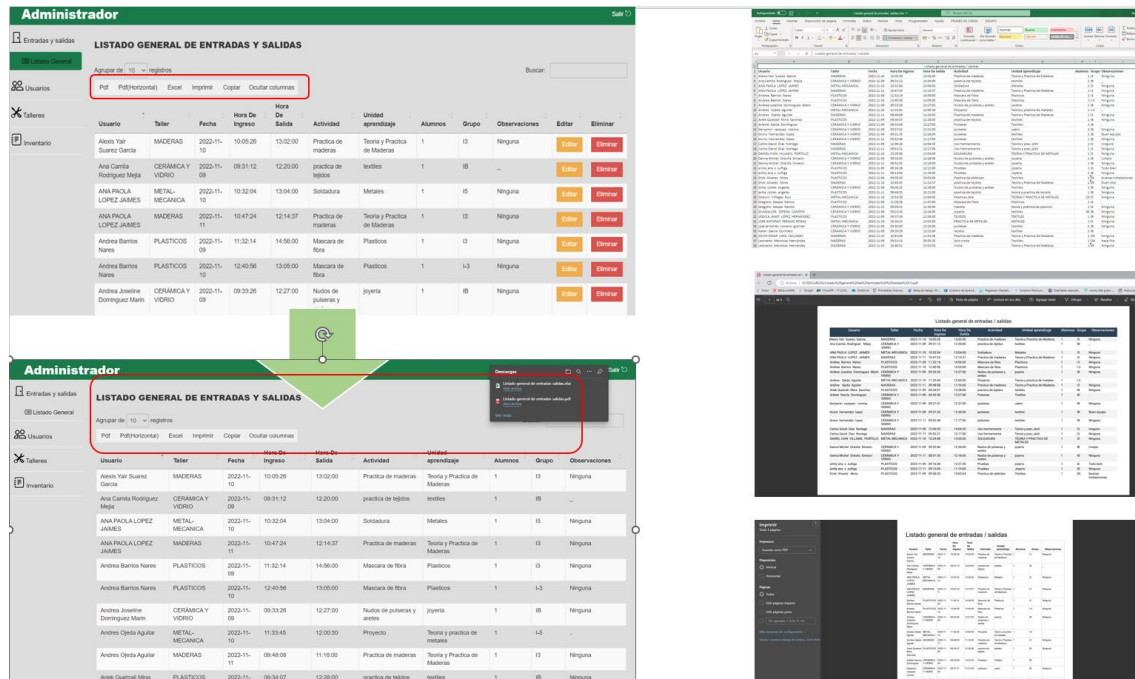


Figura 67. Vistas y formatos para descargar.

Fuente: Propia (2023).

De igual forma en el apartado de lista se puede realizar la búsqueda de cualquier registro, colocando palabras clave que identifiquen al mismo.

7.5.2.9 teléfono celular

Una de las ventajas de utilizar Bootstrap en el desarrollo del sistema, es que podemos hacer que el sistema sea adaptable, es decir que el sistema web funcione en cualquier tipo de pantalla. Si bien su diseño y funcionamiento se plantean para ser utilizado en una computadora, también es funcional en cualquier otro dispositivo móvil.

En la siguiente figura se muestran las vistas del sistema web desde un teléfono inteligente, en este caso el Samsung Galaxy a51, ver figura 68.

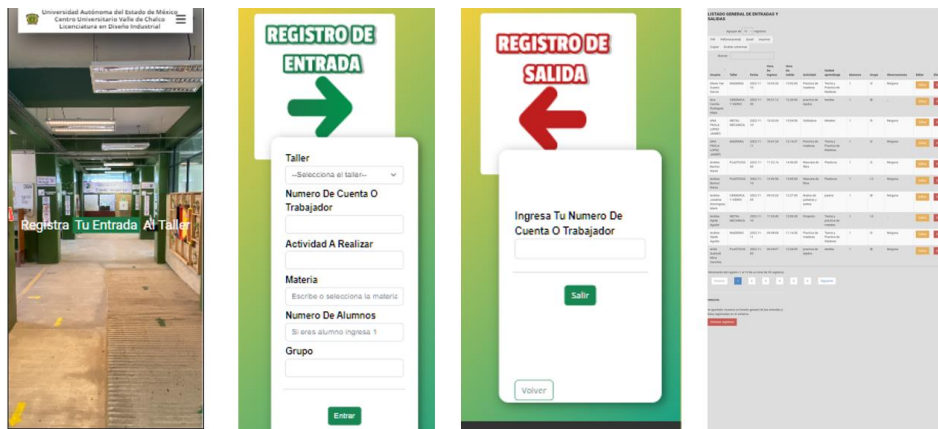


Figura 68. Vistas del funcionamiento desde un teléfono celular.

Fuente: Propia (2023).

7.5.2.10 Código QR

Al momento que el administrador registra una herramienta o maquina en el inventario, el sistema le asigna un código QR único, el cual se puede escanear con cualquier dispositivo móvil para poder ver las características de dicha herramienta, en la siguiente figura se muestra la vista que tiene el usuario al escanear el código QR, ver figura 69.



Figura 69. Vista al escanear el código QR.

Fuente: Propia (2023).

7.5.3 Controlador

Dentro del controlador se procesan las solicitudes que el usuario realice por medio de las vistas, permite la comunicación de las vistas con el modelo, para poder realizar consultas, actualizaciones, modificaciones o directamente eliminar registros, regresando un resultado al usuario por medio de las vistas.

7.5.3.1 JSP

Dentro de los archivos JSP se realiza la programación de las vistas, así como de las funciones que llevarán las mismas. Respecto a la lista del inventario, se agrega el código correspondiente para que se muestre el código QR asignado a cada registro de esa entidad dentro de la base de datos, esto mediante el uso de una API de Google, ver figura 70.

```
165 <tbody>
166 <:forEach var="datoInv" items="{listaInv}">
167 <tr>
168 <td>${datoInv.nombre}</td>
169 <td>${datoInv.marca}</td>
170 <td>${datoInv.Cantidad}</td>
171 <td>${datoInv.fechaIngreso}</td>
172 <td>
173 <!-- ${datoReg.idTaller} El ciclo recorre la lista creada para los usuarios-->
174 <:forEach var="datoT" items="{listaTall}">
175 <!-- Por cada recorrido se hace una condicion para verificar que el
176 id del usuario coincida con el id del usuario de la tabla-->
177 <:if test="{datoInv.idTaller == datoT.idTaller }">
178 <!-- Mostramos los datos de ese usuario-->
179 <td>${datoT.nombre}
180 </:if>
181 </:forEach>
182 </td>
183 <td>${datoInv.descripcion}</td>
184 <td>
185 <div class="media">
186 
187 </div>
188 </td>
189 <td>
190 
193 </td>
194 <td>
195 <a href="editarInventario.htm?idInventario=${datoInv.idInventario}"
196 class="btn btn-warning">Editar</a>
197 </td>
198 <td>
199 <a onclick="eliminar(${datoInv.idInventario})" class="btn btn-danger" > Eliminar </a>
200 </td>
201 </tr>
202 </:forEach>
203 </tbody>
204 </tbody>
```

Figura 70. Código para mostrar QR en la lista del inventario.

Fuente: Propia (2023).

De igual forma se crea el código para mostrar las alertas correspondientes a el resultado de la acción que el usuario realice, esto se hace utilizando SweetAlert que es un plugin de JQuery, así como el lenguaje de programación JavaScript.

Estas alertas tienen interacción con la base de datos, para posteriormente mostrar un mensaje por medio de las vistas al usuario, en la siguiente figura se muestra el código de la alerta correspondiente al momento que se desea eliminar un registro del inventario, ver figura 71.

```
<script src="js/sweetalertOr.min.js"></script>
<script>
    function eliminar(id) {
        swal({
            title: "¿Estas seguro de eliminar?",
            text: "El registro que intentas borrar contiene datos relacionados",
            icon: "warning",
            buttons: true,
            dangerMode: true,
        })
            .then((OK) => {
                if (OK) {
                    $.ajax({
                        url: "eliminarInventario.htm?idInventario=" + id, success: function (res) {
                            console.log(res);
                        }
                    });
                    swal("Registro eliminado correctamente", {
                        icon: "success",
                    }).then((ok) => {
                        if (ok) {
                            location.href = "listaInventario.htm"
                        }
                    });
                }
            });
    }
</script>
```

Figura 71. Código de alerta al eliminar un registro del inventario.

Fuente: Propia (2023).

7.5.3.2 Paquete controlador

Dentro de los paquetes de nuestro proyecto, se crea uno en específico para el controlador y dentro se crean los controladores específicos para cada entidad dentro de nuestra base de datos, ver figura 72.

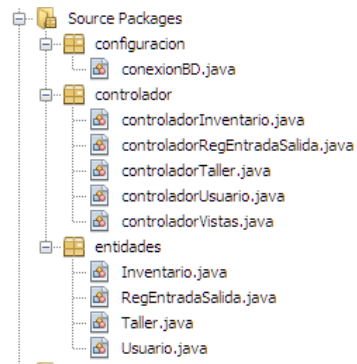


Figura 72. Creación del paquete controlador.

Fuente: Propia (2023).

Cada controlador tiene programadas las acciones que se ejecutarán por medio de las vistas a la base de datos, esto utilizando el lenguaje de programación java.

En la siguiente figura se muestra el controlador correspondiente a la entidad RegEntradaSalida, dentro del controlador se realiza la importación de los paquetes necesarios para su funcionamiento, lo primero que se crea es el método para mostrar la vista del registro de entrada y salida, ver figura 73.

```

23  @Controller
24  public class controladorRegEntradaSalida {
25
26      //Para obtener las FK
27      int idRegEntradaSalida;
28      List datos, datos2, datos3, datosL;
29
30      //Realizar una instancia para la conexion
31      conexionBD con = new conexionBD();
32      JdbcTemplate jdbcTemplate = new JdbcTemplate(con.conectar());
33      ModelAndView mav = new ModelAndView();
34
35      //Metodo para la vista
36      @RequestMapping(value = "altaRegEntradaSalida.htm", method = RequestMethod.GET)
37      public ModelAndView VistaAgregarRegEntradaSalida() {
38          String sql = "select * from taller";
39          datos2 = this.jdbcTemplate.queryForList(sql);
40          mav.addObject("lista2", datos2);
41          String sql2 = "select * from usuario";
42          datos3 = this.jdbcTemplate.queryForList(sql2);
43          mav.addObject("lista3", datos3);
44          mav.addObject(new RegEntradaSalida());
45          mav.setViewName("altaRegEntradaSalida");
46          return mav;

```

Figura 73. Código del controlador para mostrar la vista.

Fuente: Propia (2023).

Adicionalmente se crea el método para registrar una nueva entrada, dentro del cual se realizan las validaciones, en el caso que el usuario no se encuentre registrado dentro del sistema o ya se encuentre registrada su entrada, en la siguiente figura se observa el código para registrar una nueva entrada, ver figura 74.

```

49 //lista validacion fk
50 List Usuario;
51 //Metodo para agregar
52 @RequestMapping(value = "altaRegEntradaSalida.htm", method = RequestMethod.POST)
53 public ModelAndView AgregarRegEntradaSalida(ReqEntradaSalida req) {
54     //operacion opcor = correcto
55     //operacion opinc = incorrecto
56     String operacion = "";
57     //METODO PARA OBTENER LA FECHA ACTUAL
58     Date dat = new java.util.Date();//Declaramos un objeto date
59     //Damos formato de fecha
60     SimpleDateFormat simpleDateFecha = new java.text.SimpleDateFormat("yyyy-MM-dd");
61     //Asignamos ese formato a la fecha
62     String formattedString = simpleDateFecha.format(dat);
63
64     //METODO PARA OBTENER LA HORA ACTUAL
65     //Damos formato de fecha
66     SimpleDateFormat simpleDateHora = new java.text.SimpleDateFormat("HH:mm:ss");
67     //Asignamos ese formato a la fecha
68     String formattedString2 = simpleDateHora.format(dat);
69     idUsuario = (int) req.getIdUsuario();//Nombre de la BD color naranja
70     String sqlValidacion = "select *from regentradasalida where horaSalida is null and idUsuario=" + idUsuario;
71     datos = this.jdbcTemplate.queryForList(sqlValidacion);
72
73     //validamos que el id este registrado
74     String sqlValidacioFK = "select * from usuario where idUsuario=" + idUsuario;
75     Usuario = this.jdbcTemplate.queryForList(sqlValidacioFK);
76     if (Usuario.isEmpty()) {
77         //mandamos el valor de la operacion
78         operacion = "opNoExiste";
79         //si no existe
80         return new ModelAndView("redirect:/altaRegEntradaSalida.htm?operacion=" + operacion);
81     }
82     // System.out.println(datos);
83     //Comparacion
84     if (datos.isEmpty()) {
85         String sql = "insert into regentradasalida(fecha,horaIngreso,idTaller,idUsuario,actividad,unidadAprendizaje,numAlumnos,grupo) values(?, ?, ?, ?, ?, ?, ?, ?)";
86         this.jdbcTemplate.update(sql, formattedString, formattedString2, req.getIdTaller(), req.getIdUsuario(), req.getActividad(), req.getUnidadAprendizaje(),
87             req.getNumAlumnos(), req.getGrupo());
88         //mandamos el valor de la operacion
89         operacion = "opcor";
90         return new ModelAndView("redirect:/altaRegEntradaSalida.htm?operacion=" + operacion);
91     } else {
92         //mandamos el valor de la operacion
93         operacion = "opinc";
94         return new ModelAndView("redirect:/altaRegEntradaSalida.htm?operacion=" + operacion);
95     }

```

Figura 74. Código del controlador para registrar una entrada.

Fuente: Propia (2023).

De igual forma se crea el método para listar los registros, obteniendo datos de otras entidades de la base de datos, esto gracias a que se utiliza una base de datos relacional, ver figura 75.

```

97
98 //Metodo para listar
99 @RequestMapping("listaRegEntradaSalida.htm")
100 public ModelAndView ListarRegEntradaSalida() {
101     String sql = "select * from RegEntradaSalida";
102     datos = this.jdbcTemplate.queryForList(sql);
103     mav.addObject("listaReg", datos);
104     mav.setViewName("listaRegEntradaSalida");
105     //Listas para poder ver las FK
106     String sql1 = "select * from taller";
107     datos2 = this.jdbcTemplate.queryForList(sql1);
108     mav.addObject("lista2", datos2);
109     String sql2 = "select * from usuario";
110     datos3 = this.jdbcTemplate.queryForList(sql2);
111     mav.addObject("lista3", datos3);
112     return mav;
113 }

```

Figura 75. Código del controlador para listar registros.

Fuente: Propia (2023).

Así mismo en la siguiente figura se observa el método por el cual se obtienen los datos de un registro específico, en caso de querer realizar su actualización y el método correspondiente para poder realizar dicha actualización, ver figura 76.

```

115 //Metodo para la vista del valor a editar
116 @RequestMapping(value = "editarRegEntradaSalida", method = RequestMethod.GET)
117 public ModelAndView vistaEditarRegEntradaSalida(HttpServletRequest request) {
118     idRegEntradaSalida = Integer.parseInt(request.getParameter("idRegistro")); //Nombre de la BD color naranja
119     String sql = "select * from regentradasalida where idRegistro= " + idRegEntradaSalida;
120     datos = this.jdbcTemplate.queryForList(sql);
121     mav.addObject("lista", datos);
122     //Listas para poder ver las FK
123     String sql1 = "select * from taller";
124     datos2 = this.jdbcTemplate.queryForList(sql1);
125     mav.addObject("lista2", datos2);
126     String sql2 = "select * from usuario";
127     datos3 = this.jdbcTemplate.queryForList(sql2);
128     mav.addObject("lista3", datos3);
129     mav.addObject(new RegEntradaSalida());
130     mav.setViewName("editarRegEntradaSalida");
131     return mav;
132 }
133
134 //Metodo para editar
135 @RequestMapping(value = "editarRegEntradaSalida.htm", method = RequestMethod.POST)
136 public ModelAndView EditarRegEntradaSalida(RegEntradaSalida reg) {
137     String sql = "update regentradasalida set fecha=?,horaIngreso=?,horaSalida=?,idTaller=?,idUsuario=?,actividad=?,unidadAprendizaje=?,numAlumnos=?,grupo=?,
138     + "observaciones=? where idRegistro=";
139     this.jdbcTemplate.update(sql, reg.getFecha(), reg.getHoraIngreso(), reg.getHoraSalida(), reg.getIdTaller(), reg.getIdUsuario(),
140     reg.getActividad(), reg.getUnidadAprendizaje(), reg.getNumAlumnos(), reg.getGrupo(), reg.getObservaciones(), idRegEntradaSalida);
141     String operacion = "opoor";
142     return new ModelAndView("redirect:/listaRegEntradaSalida.htm?operacion="+operacion);
143 }

```

Figura 76. Código del controlador para ver y editar registros.

Fuente: Propia (2023).

En la siguiente figura se observan los métodos para eliminar uno o todos los registros, ver figura 77.

```

145 //Metodo para eliminar
146 @RequestMapping(value = "eliminarRegEntradaSalida.htm")
147 public ModelAndView EliminarRegEntradaSalida(HttpServletRequest request) {
148     idRegEntradaSalida = Integer.parseInt(request.getParameter("idRegistro")); //Obtener el id del seleccionado
149     String sql = "delete from regentradasalida where idRegistro=" + idRegEntradaSalida;
150     this.jdbcTemplate.update(sql);
151     return new ModelAndView("redirect:/listaRegEntradaSalida.htm");
152 }
153 //Metodo para eliminar todos
154 @RequestMapping(value = "eliminarRegistros.htm")
155 public ModelAndView EliminarRegistros(HttpServletRequest request) {
156     String sql = "delete from regentradasalida";
157     this.jdbcTemplate.execute(sql);
158     return new ModelAndView("redirect:/listaRegEntradaSalida.htm");
159 }

```

Figura 77. Código del controlador para eliminar registros.

Fuente: Propia (2023).

Por último, en la figura 78 se observan los métodos para mostrar la vista de registrar la salida, así como el método para registrar la salida dentro de la base de datos, ver figura 78.

```

163 //Metodo para la vista del valor a editar SALIDA
164 @RequestMapping(value = "altaSalida", method = RequestMethod.GET)
165 public ModelAndView vistaSalida(HttpServletRequest request) {
166     //operacion opcor = correcto
167     //operacion opinc = incorrecto
168     String operacion = "";
169     idUsuario = Integer.parseInt(request.getParameter("idUsuario")); //Nombre de la BD color naranja
170     System.out.println(idUsuario);
171     String sql = "select *from regentradasalida where horaSalida is null and idUsuario=" + idUsuario;
172     datos = this.jdbcTemplate.queryForList(sql);
173     mav.addObject("lista", datos);
174     //listamos los usuarios para la llave foranea
175     String sql2 = "select * from usuario";
176     datos3 = this.jdbcTemplate.queryForList(sql2);
177     mav.addObject("lista3", datos3);
178     if (datos.isEmpty()) {
179         //si no existe el registro el registro, es porque no ha entrado
180         operacion = "opinc";
181         return new ModelAndView("redirect:/altaRegSalida.htm?operacion=" + operacion);
182         //mav.setViewName("index");
183     } else {
184         //si existe continua con la salida
185         mav.setViewName("altaSalida");
186     }
187     return mav;
188 }
189
190 //Metodo para editar
191 @RequestMapping(value = "altaSalida.htm", method = RequestMethod.POST)
192 public ModelAndView AltaSalida(RegEntradaSalida reg) {
193     //operacion opcor = correcto
194     //operacion opinc = incorrecto
195     String operacion = "";
196     //METODO PARA OBTENER LA HORA ACTUAL
197     Date dat = new java.util.Date(); //Declaramos un objeto date
198     //Damos formato de fecha
199     SimpleDateFormat simpleDateHora = new java.text.SimpleDateFormat("HH:mm:ss");
200     //Asignamos ese formato a la fecha
201     String formattedString2 = simpleDateHora.format(dat);
202     operacion = "opcor";
203     String sql = "update regentradasalida set horaSalida=?,observaciones=? where idUsuario=? and horaSalida is null";
204     this.jdbcTemplate.update(sql, formattedString2, reg.getObservaciones(), idUsuario);
205     return new ModelAndView("redirect:/altaRegSalida.htm?operacion=" + operacion);
206 }
207

```

Figura 78. Código del controlador para la vista y registro de salida.

Fuente: Propia (2023).

Una vez que se concluye la fase de la implementación con éxito se realizan pruebas para verificar que el funcionamiento y la lógica del sistema sean los correctos, los errores que surgen en la etapa de pruebas se corrigen en la misma, para posteriormente pasar a la etapa de verificación.

7.6 Verificación

El día 6 de octubre del 2022 se tuvo una reunión con el Licenciado en diseño industrial Cesar Adolfo Muñoz Herrera encargado de los talleres, para mostrarle el sistema terminado en una versión beta, así mismo se le solicitó verificar que el sistema cumpla con los requerimientos planteados en la fase del análisis.

En caso de existir alguna modificación o sugerencia, es la etapa en donde se realizan los cambios para poder tener un sistema que funcione de manera correcta. Respecto a las modificaciones propuestas por el encargado son respecto al diseño y no al funcionamiento del sistema, ya que nos menciona que el sistema funciona de manera correcta. La siguiente figura muestra una fotografía verificando los cambios realizados al sistema respecto al diseño, ver figura 79.

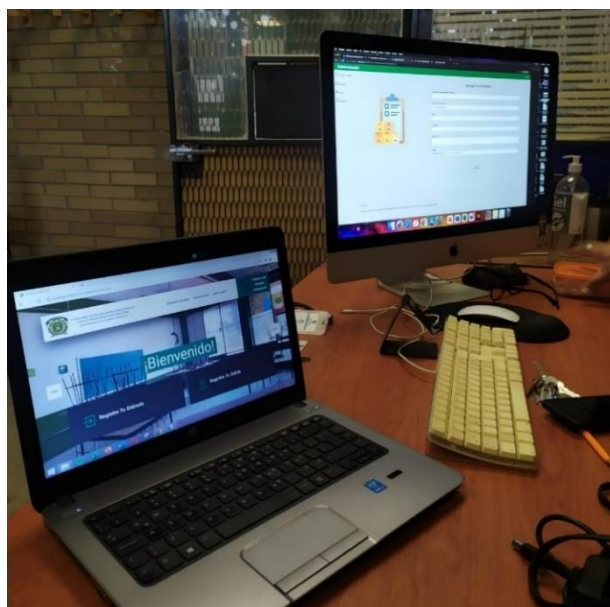


Figura 79. Verificación de las sugerencias al sistema.

Fuente: Propia (2022).

Posteriormente a que las sugerencias al sistema estén resueltas, el sistema está listo para su lanzamiento.

7.6.1 Lanzamiento

El sistema web en su versión final, se utilizó dentro de los talleres de diseño en la semana del 8 de noviembre al 11 del mismo mes, del año 2022.

Se colocó en el área de registro de los talleres una laptop con el sistema web funcionando, de igual forma se colocó una infografía para mostrarles a los alumnos y profesores como deben realizar su registro, ver figura 80.



Figura 80. Funcionamiento del sistema web en los talleres de diseño industrial.
Fuente: Propia (2022).

Dentro de este lapso, alumnos y profesores hicieron uso del sistema registrando tanto sus entradas como salidas de los talleres, ver figura 81.

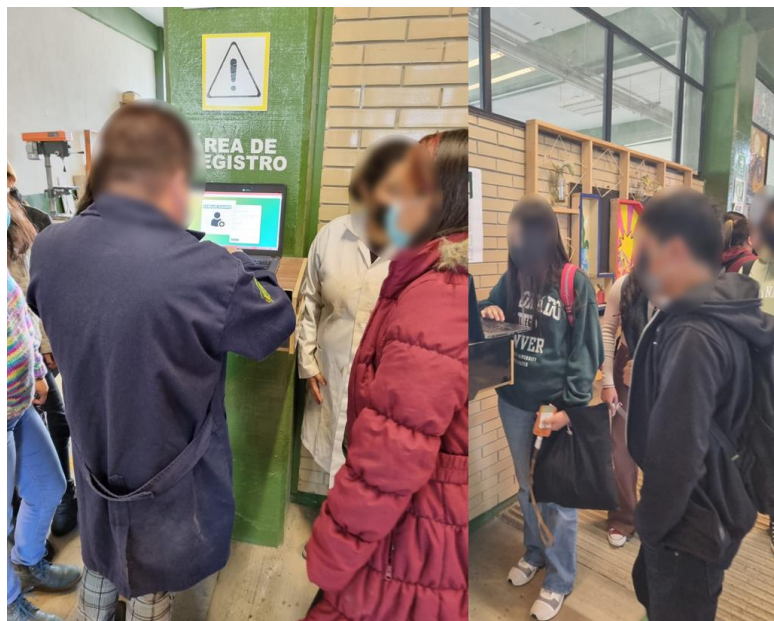


Figura 81. Alumnos y profesores haciendo uso del sistema web.
Fuente: Propia (2022).

Adicionalmente algunos profesores comentaban que este sistema sería de gran utilidad ya que registra la hora y fecha exacta del ingreso y salida de los talleres.

De igual manera se registró en el sistema web el inventario con el que cuentan los talleres y se colocaron los códigos QR generados por el sistema a las herramientas y máquinas correspondientes. En la siguiente figura se muestra el proceso al escanear el código QR de alguna herramienta o máquina dentro del taller, ver figura 82.



Figura 82. Escaneo de código QR de una máquina.

Fuente: Propia (2022).

Como se menciona en el análisis, el sistema web está montado en un servidor local, por lo tanto, para poder ver la información del código QR debemos estar en la misma red local que el servidor. Siguiendo el principio de la red local, podemos acceder al sistema desde nuestro teléfono celular, ver figura 83.



Figura 83. Sistema web en teléfono celular.

Fuente: Propia (2022).

Durante la semana que el sistema web estuvo en funcionamiento se obtuvieron un total de 45 registros de usuarios, de los cuales 4 son profesores y los 41 registros restantes son de alumnos, ver figura 84.

Administrador Salir

Entradas y salidas **LISTADO GENERAL DE USUARIOS**

Usuarios Agrupar de 10 registros Pdf Pdf(horizental) Excel Imprimir Copiar Ocultar columnas Buscar:

Num cuenta o trabajador	Tipo	Nombre	Apellido Paterno	Apellido Materno	Sexo	Fecha De Nacimiento	Correo	Editar	Eliminar
...	Editar	Eliminar
...	Editar	Eliminar
...	Editar	Eliminar
...	Editar	Eliminar
...	Editar	Eliminar
...	Editar	Eliminar
...	Editar	Eliminar
...	Editar	Eliminar
...	Editar	Eliminar
...	Editar	Eliminar

Mostrando del registro 1 al 10 de un total de 45 registros

Mostrando del registro 1 al 10 de un total de 45 registros

Anterior 1 2 3 4 5 Siguiente

INFORMACIÓN
Este apartado muestra un listado general de todos los usuarios que se han registrado al sistema.
[Eliminar registros](#)

Figura 84. Registros de usuarios en el sistema web.

Fuente: Propia (2023).

Con respecto a los registros de entrada y salida realizados, se obtuvieron un total de 58 registros, ver figura 85.

Administrador

Entradas y salidas

LISTADO GENERAL DE ENTRADAS Y SALIDAS

Agrupar de 10 registros PDF PDF(horizantal) Excel Imprimir Copiar Ocultar columnas

Buscar:

Usuario	Taller	Fecha	Hora De Ingreso	Hora De Salida	Actividad	Unidad aprendizaje	Alumnos	Grupo	Observaciones	Editar	Eliminar
										Editar	Eliminar
										Editar	Eliminar
										Editar	Eliminar
										Editar	Eliminar
										Editar	Eliminar
										Editar	Eliminar
										Editar	Eliminar
										Editar	Eliminar
										Editar	Eliminar
										Editar	Eliminar

Mostrando del registro 1 al 10 de un total de 58 registros

Mostrando del registro 1 al 10 de un total de 58 registros

Anterior 1 2 3 4 5 6 Siguiente

Figura 85. Registros de entradas y salidas en el sistema web.

Fuente: Propia (2023).

Así mismo se registraron 38 herramientas y máquinas en el inventario, en la siguiente figura se muestran los registros correspondientes al inventario, ver figura 86.

Administrador

Entradas y salidas

LISTADO GENERAL DEL INVENTARIO

Agrupar de 10 registros PDF PDF(horizantal) Excel Imprimir Copiar Ocultar columnas

Buscar:

Nombre	Marca	Cantidad	Fecha De Ingreso	Taller	Descripción	Imagen	QR	Editar	Eliminar
Estufa de madera	No Aplica	1	2023-11-10	PLASTICOS	Estufa de madera para guardar cosas en general			Editar	Eliminar
Receptor	No Aplica	1	2023-11-10	PLASTICOS	Instrumento para recibir fuerza			Editar	Eliminar
Máquina Resabido	Fipaph	1	2023-11-10	METALMECANICA	Se utiliza principalmente para mecanizar grandes piezas.			Editar	Eliminar
Rotador de tubos	Hobbit	1	2023-11-10	METALMECANICA	Máquina llamada rotador de tubos			Editar	Eliminar
Máquina para soldar	Infra	1	2023-11-10	METALMECANICA	Máquina para soldar			Editar	Eliminar
Lentira para doblar	igapa	1	2023-11-10	METALMECANICA	Máquina llamada Lentira para doblar			Editar	Eliminar
Armario de banco	No Aplica	1	2023-11-10	METALMECANICA	Alfabeto de herramientas			Editar	Eliminar

Figura 86. Registros del inventario en el sistema web.

Fuente: Propia (2023).

7.7 Mantenimiento

Posterior a la fase del lanzamiento del sistema, surgen algunas observaciones por parte de los usuarios, algunos de estos usuarios en su mayoría profesores, comentaban que el sistema es de gran utilidad para los talleres y esperaban que se implementara en los mismos.

Por otra parte, también se recibieron algunas observaciones respecto al diseño del sistema, estas observaciones fueron mínimas y se cubrieron en la fase de mantenimiento, algunas de las observaciones fueron:

- El color de los botones, indican que el color de los botones debe ser llamativo para poder identificarlos de manera correcta.
- Al momento de registrarse, el sistema ayude autocompletando la materia por la cual asisten, ya que el nombre de estas es demasiado largo.
- Poder utilizar algunas funciones del teclado, como lo es la tecla Enter para accionar el botón de registro.

Así mismo se encontraron algunas correcciones a realizar dentro de la base de datos, como lo fue la longitud de los caracteres de algunos formularios de registro, posteriormente se analizaron las observaciones y se le realizaron los ajustes pertinentes al sistema, para que sea mas amigable a los usuarios.

Concluida la etapa del mantenimiento del sistema, este se encuentra listo y es entregado al encargado de los talleres de diseño industrial, quien verifica las correcciones realizadas y indica que el funcionamiento del sistema es el adecuado.

VIII. CONCLUSIONES Y SUGERENCIAS.

8.1 Conclusiones

- Se concluyó que el uso de las herramientas tecnológicas como los lenguajes de programación y los framework Bootstrap e Spring MVC, facilitan el desarrollo de los sistemas y ayudan a dar solución a algún problema en específico.
- El uso del sistema web facilita algunas acciones, como lo es la búsqueda de información, que de no contar con el sistema web nos llevaría mucho tiempo.
- El uso del sistema facilita a los profesores al momento de realizar su registro ya que el hecho de estar modificando el archivo excel que se usaba anteriormente era tardado, sin embargo, con el sistema se agiliza este proceso.
- El acceso a la información ya sea de registros de entrada y salida, así como herramientas o máquinas del inventario, o cualquier otro registro es de manera más amigable y rápida.
- El hecho de poder descargar la información en diferentes formatos facilita el manejo de esta.
- El uso del sistema web permite llevar la administración adecuada de los talleres.

8.2 Sugerencias

- Si bien no es necesario, el uso de navegadores actualizados puede brindar mejor funcionalidad al sistema web.
- Como trabajo futuro se podría implementar en un servidor con un dominio en específico.

IX. REFERENCIAS

- Acodigo. (2017). Spring MVC Creación de aplicaciones web. Recuperado de <http://acodigo.blogspot.com/2017/03/spring-mvc-creacion-de-aplicaciones-web.html>
- Arias, F. G. (2012). *EL PROYECTO DE INVESTIGACION* .(27-36). EPISTEME, C.A.
- Ayala-Hernández C.C., B.-M. J. (2011). Un sistema de control de salidas . *Ingeniería. Investigación y Tecnología*, 63-71.
- Balsamiq Wireframes - Industry Standard Low-Fidelity Wireframing Software | Balsamiq. (s. f.). <https://balsamiq.com/wireframes/>
- Carlemany, U. (s.f.). *UNIVERSITAT Carlemany*. Recuperado de : <https://www.universitatcarlemany.com/actualidad/metodologias-de-desarrollo-de-software>
- Challenger-Pérez, I., Díaz-Ricardo, Y., & Becerra-García, R. A. (2014). El lenguaje de programación Python. *Ciencias Holguín*, XX(2), 1-13.
- Click-it (2022) Breve historia del nacimiento de las bases de datos. Recuperado de: <https://click-it.es/breve-historia-del-nacimiento-de-las-bases-de-datos/#:~:text=Pero%20si%20hablamos%20de%20bases,y%20posteriormente%20a%20la%20contabilidad.>
- codigos qr. (s.f.). Recuperado de: <https://www.codigos-qr.com/>
- desarrolloweb. (28 de 07 de 2020). Modelo-Vista-Controlador. Recuperado de: <https://desarrolloweb.com/articulos/que-es-mvc.html>
- Digital Guide . (21 de Marzo de 2019). Recuperado de: <https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/el-modelo-en-cascada/>
- Digital Guide. (2020, 22 mayo). Recuperado de IONOS Digital Guide. <https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/paradigmas-de-programacion/>.
- Eguíluz J. (2008). Introducción a CSS. Recuperado de https://www.jesusda.com/docs/ebooks/introduccion_css.pdf

Faraoni, F. J. (2015). DESARROLLO DE UNA APLICACION WEB CON SPRING FRAMEWORK PARA UN GESTOR DE UN RECETRARIO. 105.(Proyecto de fin de curso).Universidad Politécnica de Madrid.

flaticon. (s.f.). Recuperado de flaticon: <https://www.flaticon.es/>

geeksforgeeks. (31 de 05 de 2022). Spring – MVC Framework. Recuperado de: geeksforgeeks: <https://www.geeksforgeeks.org/spring-mvc-framework/>

GoogleSites. (s.f.). Recuperado de GoogleSites: <https://sites.google.com/site/modelamientosa/>

HostingPlus Mexico. (2021, 23 abril). Clasificación de las bases de datos: conoce todos los tipos | Blog | Hosting Plus Mexico. Hosting Plus. Recuperado de: <https://www.hostingplus.mx/blog/clasificacion-de-las-bases-de-datos-conoce-todos-los-tipos/>

HOSTTINGERTUTORIALES. (03 de 06 de 2022). Recuperado de HOSTTINGERTUTORIALES: <https://www.hostinger.mx/tutoriales/que-es-bootstrap>

HubSpot. (20 de 07 de 2022). Obtenido de HubSpot: <https://blog.hubspot.es/website/que-es-desarrollo-web>

Lenovo México. (2022). Java. Recuperado de: <https://www.lenovo.com/mx/es/faqs/pc-vida-faqs/que-es-java/?orgRef=https%3A%2F%2Fwww.bing.com%2F>

Marín, R. (2022, 26 septiembre). Los gestores de bases de datos más usados en la actualidad. Canal Informática y TICS. Recuperado de: <https://www.inesem.es/revistadigital/informatica-y-tics/los-gestores-de-bases-de-datos-mas-usados/>

Marker, G. (2022, 12 noviembre). Lenguaje de alto nivel. Recuperado de: Tecnología + Informática. <https://www.tecnologia-informatica.com/lenguaje-de-alto-nivel-que-es-ejemplos/>

Martínez Ladrón de Guevara, J. (s. f.). Recuperado de Repositori Digital Instituto Superior
Japón:<http://190.57.147.202:90/jspui/bitstream/123456789/1401/1/Fundamentos%20de%20programacion%20en%20Java.pdf>

Martínez, A. (28 de Abril de 2022).En conceptodefinicion.de. Recuperado de : <https://conceptodefinicion.de/software/>

- Mohedano, J., Saiz, J. M., & Román, P. S. (2012). *Iniciación a JavaScript*. Ministerio de Educación.
- Oracle (s.f) Base de datos relacional. Recuperado de: <https://www.oracle.com/mx/database/what-is-a-relational-database/>
- PHP (s. f.). *Historia de PHP - Manual*. Recuperado de: <https://www.php.net/manual/es/history.php.php>
- Profile. (2021). *En Profile*. Recuperado de: <https://profile.es/blog/frameworks-java-desarrollo-web/>
- ptolomeo. (s.f.). *En unam*. Recuperado de : http://www.ptolomeo.unam.mx:8080/jspui/bitstream/132.248.52.100/14061/1/decd_4525.pdf
- Rodríguez, M. (2022, 17 agosto). *HTML*. Recuperado de Agencia SEO en México: <https://www.seoenmexico.com.mx/blog/html/>
- Serrano, J. F. V., Abril, A. P., Bellas, F. G., & Calle, Á. S. (2010). *Diseñar y programar, todo es empezar.: Una introducción a la programación orientada a objetos usando UML y Java (Vol. 25)*. Librería-Editorial Dykinson.
- Software de diagramación en línea y solución visual. (s. f.). Lucidchart. <https://www.lucidchart.com/pages/es>
- Sommerville, I. (2005). *Ingeniería del software*.(5). Pearson educación.
- techopedia(27 de 4 de 2022). Obtenido de techopedia: <https://www.techopedia.com/definition/2915/quick-response-code-qr-code>
- Universidad Europea. (2022, 22 marzo). *¿Para qué sirve un gestor de base de datos?*. Recuperado de :<https://universidadeuropea.com/blog/para-que-sirve-gestor-base-datos/>