

EVIDENCIA DE APRENDIZAJE – DESARROLLAR SOFTWARE A PARTIR DE LA
INTEGRACIÓN DE SUS MÓDULOS COMPONENTES

GA8-220501096-AA1-EV01

FASE – EJECUCIÓN

Por:

GEFFRY SANTIAGO LEON JIMENEZ

CENTRO DE LA TECNOLOGÍA DEL DISEÑO Y LA PRODUCTIVIDAD EMPRESARIAL

REGIONAL CUNDINAMARCA – SENA – GIRARDOT

ANÁLISIS Y DESARROLLO DE SOFTWARE

FICHA 2977481

INSTRUCTOR: MILTON IVAN BARBOSA GAONA

01/10/2025

Tabla de Contenido

1. INTRODUCCIÓN	5
2. OBJETIVOS	6
2.1 Objetivo general	6
2.2 Objetivos específicos	6
3. INFORMACIÓN DE NUESTRO PROYECTO	7
3.1 requisitos funcionales y no funcionales	7
3.2 historias de usuario y casos de uso	15
3.3 IDE de desarrollo	25
3.3.1 Características de NetBeans	25
3.3.2 Ventajas de Usar NetBeans	26
3.4 Diagramas	27
3.5 Mecanismos de Seguridad	28
3.6 Repositorio en GitHub	30
3.7 Metodología de Software	30
3.8 Capas de los componentes	30
3.9 Mapa de navegación	32
3.10 Codificación de los módulos	39
3.11 Bibliotecas usadas en los módulos	41
3.11.1 Modulo Inicio de Sesión	42
3.11.2 Modulo de Registro	43
3.12 Framework utilizados	44
3.13 Fragmentos de código reutilizables	44
3.13 Organización en paquetes	47
3.14 Patrones de diseño	48
3.15 Pruebas unitarias de los módulos	49
3.16 Configuración de servidores y de base de datos	51
3.17 ambientes de desarrollo y pruebas	55
4. CONCLUSIONES	56
5. REFERENCIAS BIBLIOGRÁFICAS	57

Índice de tablas

Tabla 1 Requisito funcional	7
Tabla 2 Requisito no funcional	7
Tabla 3 Requisito Funcional	7
Tabla 4 Requisito no funcional	8
Tabla 5 Requisito Funcional	8
Tabla 6 Requisito no funcional	8
Tabla 7 Requisito funcional	9
Tabla 8 Requisito no funcional	9
Tabla 9 Requisito Funcional	9
Tabla 10 Requisito no Funcional	11
Tabla 11 Requisito Funcional	11
Tabla 12 Requisito funcional	12
Tabla 13 Requisito no funcional	12
Tabla 14 Requisito Funcional	12
Tabla 15 Requisito no funcional	13
Tabla 16 Requisito funcional	13
Tabla 17 Requisito no Funcional	13
Tabla 18 Requisito funcional	15
Tabla 19 Requisito no Funcional	15
Tabla 20 Historia de Usuario	15
Tabla 21 Historia de usuario	16
Tabla 22 Historia de usuario	16
Tabla 23 Historia de usuario	17
Tabla 24 Historia de usuario	17
Tabla 25 Historia de usuario	18
Tabla 26 Historia de usuario	19
Tabla 27 Historia de usuario	19
Tabla 28 Historia de usuario	20
Tabla 29 Historia de usuario	20
Tabla 30 Caso de uso	21
Tabla 31 Caso de uso	21
Tabla 32 Caso de uso	22
Tabla 33 Caso de uso	22
Tabla 34 Caso de uso	23
Tabla 35 Caso de uso	23
Tabla 36 Caso de uso	24
Tabla 37 Caso de uso	24
Tabla 38 Caso de uso	24
Tabla 39 Caso de uso	25

Tabla de Ilustraciones

Ilustración 1 Diagrama de Clases	27
Ilustración 2 Diagrama de Paquetes	27
Ilustración 3 Diagrama de Componentes	28
Ilustración 4 Sistema de Seguridad API	29
Ilustración 5 Sistema de Seguridad API	29
Ilustración 6 Tabla kanvan	31
Ilustración 7 Mapa de Navegación Registrarse	32
Ilustración 8 Mapa de Navegación Inicio Sesión	33
Ilustración 9 Mapa de Navegación Recuperar Contraseña	34
Ilustración 10 Mapa de Navegación Añadir Material	35
Ilustración 11 Mapa de Navegación Eliminar Material	36
Ilustración 12 Mapa de Navegación Modificar Material	37
Ilustración 13 Mapa de Navegación Buscar Material	38
Ilustración 14 Codificación Modulo Inicio De Sesión	40
Ilustración 15 Codificación Modulo Registrarse	40
Ilustración 16 Codificación Modulo Índice	41
Ilustración 17 Bibliotecas Usadas	42
Ilustración 18 Import y @WebServlet	42
Ilustración 19 Conexión Base de Datos y Consulta de Credenciales	43
Ilustración 20 Import y @webServlet	43
Ilustración 21 Conexión a Base de Datos MySQL	43
Ilustración 22 Implementación de Datos de Cuestionario	44
Ilustración 23 Confirmación de Registro	44
Ilustración 24 Cabeza de página web	44
Ilustración 25 Formulario	45
Ilustración 26 Mensaje de Error	45
Ilustración 27 Links	46
Ilustración 28 Captura de Datos	46
Ilustración 29 Conexión a Base de Datos	46
Ilustración 30 Insertar Datos en Tabla	46
Ilustración 31 Confirmación	47
Ilustración 32 Organización en Paquetes	47
Ilustración 33 Prueba de index	49
Ilustración 34 Prueba Registro.jsp	50
Ilustración 35 Prueba Inicio.jsp	50
Ilustración 36 Primer Paso	51
Ilustración 37 Segundo Paso	52
Ilustración 38 Tercer Paso	53
Ilustración 39 Mi Base de Datos	54
Ilustración 40 Mi Base de Datos	55

1. INTRODUCCIÓN

En esta evidencia adjuntaremos toda nuestra información de nuestro proyecto de software con el fin de ver nuestro avance y poder mostrar a mas gente que es lo que estamos haciendo, así que espero sea de su agrado este documento.

2. OBJETIVOS

2.1 Objetivo general

Mi objetivo para esta evidencia es poder entender como están funcionando mis módulos de software y ver que puedo cambiar

2.2 Objetivos específicos

- Revisar mis módulos
- Realizar una documentación del agrado del instructor
- Aprender a autocriticarme

3. INFORMACIÓN DE NUESTRO PROYECTO

3.1 requisitos funcionales y no funcionales

Tabla 1 Requisito funcional

cod	RF-01		
título	Añadir usuario - Registrarse		
prioridad	Alta	Fuente	HU-01
necesidad	Esencial		
Estabilidad	Si	Verificable	Si
Postcondición	El usuario ha sido creado		
Roles involucrados	Administrador del sistema		
Descripción	Se permite al administrador añadir un usuario con los datos obtenidos de este		

Nota: tabla hecha por Geffry Santiago Leon Jimenez

Tabla 2 Requisito no funcional

cod	RNF-01		
título	Uso del sistema para añadir usuarios		
prioridad	Alta	Fuente	RR-01
necesidad	Esencial		
Estabilidad	Si	Verificable	Si
Postcondición	El usuario ha sido creado		
Roles involucrados	Administrador del sistema		
Descripción	Dependiendo de que rango se le dará al usuario, este podrá tener mas opciones que usar en el sistema		

Nota: tabla hecha por Geffry Santiago Leon Jimenez

Tabla 3 Requisito Funcional

cod	RF-02		
título	Eliminar usuarios		
prioridad	alta	Fuente	HU-02
necesidad	Esencial		
Estabilidad	si	Verificable	si

Postcondición	El usuario ha sido eliminado
Roles involucrados	Administrador de la pagina
Descripción	El sistema debe darle al administrador la opción para eliminar los usuarios que no se hayan conectado en un tiempo

Nota: tabla hecha por Geffry Santiago Leon Jimenez

Tabla 4 Requisito no funcional

cod	RNF-02		
titulo	Velocidad de eliminación de usuarios		
prioridad	alta	Fuente	RR-02
necesidad	esencial		
Estabilidad	si	Verificable	si
Postcondición	El usuario ha sido eliminado		
Roles involucrados	Administrador de la pagina		
Descripción	Dependiendo que tantas vinculaciones tenga el usuario con el sistema, se va a tardar más ya que tienen que encontrar a otro usuario que cubra esas vinculaciones		

Nota: tabla hecha por Geffry Santiago Leon Jimenez

Tabla 5 Requisito Funcional

cod	RF-03		
titulo	Añadir material		
prioridad	alta	Fuente	HU-03
necesidad	Esencial		
Estabilidad	si	Verificable	si
Postcondición	El material está listo para el préstamo		
Roles involucrados	Administrador de la pagina		
Descripción	Se permite al administrador añadir un nuevo material a base de la información obtenida de este		

Nota: tabla hecha por Geffry Santiago Leon Jimenez

Tabla 6 Requisito no funcional

cod	RNF-03		
titulo	Añadir material dependiendo del tipo de información		
prioridad	alta	Fuente	RR-01
necesidad	esencial		

Estabilidad	si	Verificable	si
Postcondición	El libro está disponible para préstamo		
Roles involucrados	Administrador de la pagina		
Descripción	Al administrador le dan los datos del material, pero a este le toca crear un código en el cual pueda identificar los libros en donde están ubicados etc..		

Nota: tabla hecha por Geffry Santiago Leon Jimenez

Tabla 7 Requisito funcional

cod	RF-04		
título	Eliminar material		
prioridad	Alta	Fuente	HU-04
necesidad	Esencial		
Estabilidad	si	Verificable	si
Postcondición	El material ha sido eliminado		
Roles involucrados	Administrador del sistema		
Descripción	Al administrador se le da la orden de eliminar los materiales que ya no están disponibles por una razón u otra		

Nota: tabla hecha por Geffry Santiago Leon Jimenez

Tabla 8 Requisito no funcional

cod	RNF-04		
título	Problemas con la eliminación del material		
prioridad	alta	Fuente	RR-03
necesidad	esencial		
Estabilidad	si	Verificable	si
Postcondición	El material ha sido eliminado		
Roles involucrados	Administrador del sistema		
Descripción	Dado el caso en el que alguien pida el libro y este ya haya sido eliminado se le notificara que el material ya no está disponible		

Nota: tabla hecha por Geffry Santiago Leon Jimenez

Tabla 9 Requisito Funcional

cod	RF-05		
título	Modificar material		
prioridad	media	Fuente	HU-05
necesidad	esencial		

Estabilidad	si	Verificable	si
Postcondición	La modificación del material ha sido aceptada		

Roles involucrados	Administrador del sistema
Descripción	El material debe ser modificado ya sea por un error a la hora de añadirlo o por el simple hecho de que cambiaron algún dato

Nota: tabla hecha por Geffry Santiago Leon Jimenez

Tabla 10 Requisito no Funcional

cod	RNF-05		
titulo	Velocidad al modificar el material		
prioridad	media	Fuente	RR-03
necesidad	esencial		
Estabilidad	si	Verificable	si
Postcondición	La modificación del material ha sido aceptada		
Roles involucrados	Administrador del sistema		
Descripción	El administrador debería tardar mínimo 2 minutos para modificar los datos del material		

Nota: tabla hecha por Geffry Santiago Leon Jimenez

Tabla 11 Requisito Funcional

cod	RF-06		
titulo	Consultar el material de formación		
prioridad	baja	Fuente	HU-06
necesidad	media		
Estabilidad	si	Verificable	si
Postcondición	El material se ha encontrado		
Roles involucrados	Administrador del sistema - cliente		
Descripción	El administrador como el cliente buscan en un apartado en donde están los materiales subidos		

Nota: tabla hecha por Geffry Santiago Leon Jimenez

cod	RNF-06		
titulo	Velocidad para encontrar el material de formación		
prioridad	baja	Fuente	RR-06
necesidad	media		
Estabilidad	si	Verificable	si

Postcondición	El material se ha encontrado
Roles involucrados	Administrador del sistema – cliente
Descripción	El sistema como mínimo debería encontrar los materiales en menos de 5 segundos para tener un servicio más rápido

Nota: tabla hecha por Geffry Santiago Leon Jimenez

Tabla 12 Requisito funcional

cod	RF-07		
título	prestamos		
prioridad	alta	Fuente	HU-07
necesidad	esencial		
Estabilidad	si	Verificable	si
Postcondición	Los prestamos se han realizado correctamente		
Roles involucrados	Administrador del sistema - cliente		
Descripción	El sistema se hace responsable de informar al administrador de la página que un cliente ha pedido un préstamo		

Nota: tabla hecha por Geffry Santiago Leon Jimenez

Tabla 13 Requisito no funcional

cod	RNF-07		
título	Capacidad de los prestamos		
prioridad	alta	Fuente	RR-07
necesidad	esencial		
Estabilidad	si	Verificable	si
Postcondición	Los prestamos se han realizado correctamente		
Roles involucrados	Administrador del sistema - cliente		
Descripción	El sistema hace una capacidad máxima de material para que llevar un mejor orden a la hora de sacar prestamos		

Nota: tabla hecha por Geffry Santiago Leon Jimenez

Tabla 14 Requisito Funcional

cod	RF-08		
título	Iniciar sesión		
prioridad	alta	Fuente	HU-08
necesidad	esencial		
Estabilidad	si	Verificable	si
Postcondición	Mantener abierta la sesión por un tiempo determinado		
Roles involucrados	Administrador del sistema – cliente		

Descripción	El sistema permite ingresar al administrador y al cliente mediante un sistema de seguridad el cual te pide un usuario y contraseña
-------------	--

Nota: tabla hecha por Geffry Santiago Leon Jimenez

Tabla 15 Requisito no funcional

cod	RNF-08		
titulo	Seguridad a la hora de iniciar sesión		
prioridad	alta	Fuente	RR-08
necesidad	esencial		
Estabilidad	si	Verificable	si
Postcondición	Mantener abierta la sección por un tiempo determinado		
Roles involucrados	Administrador del sistema – cliente		
Descripción	A la hora de iniciar sesión se le enviar un código al Gmail de esa persona para verificar si es él y así prevenir que no sea alguien mas		

Nota: ilustración hecha por Geffry Santiago Leon Jimenez

Tabla 16 Requisito funcional

cod	RF-09		
titulo	Registrar usuario		
prioridad	alta	Fuente	HU-09
necesidad	esencial		
Estabilidad	si	Verificable	si
Postcondición	Tener una cuenta para poder ingresar a la pagina		
Roles involucrados	Administrador del sistema – cliente		
Descripción	El sistema permite registrar mis datos con información básica como nombre, apellido, número de teléfono, correo electrónico y fecha de nacimiento		

Nota: tabla hecha por Geffry Santiago Leon Jimenez

Tabla 17 Requisito no Funcional

cod	RNF-09		
titulo	Verificación de los datos puestos para el registro		
prioridad	alta	Fuente	RR-09
necesidad	esencial		
Estabilidad	si	Verificable	si
Postcondición	Tener una cuenta para poder ingresar a la pagina		
Roles involucrados	Administrador del sistema – cliente		

Descripción	A la hora de que la persona se registre habrá 2 opciones de confirmar el registro las cuales son: código a número de teléfono el cual como dice se enviara un código al teléfono que haya sido puesto. 2. Código de verificación al correo electrónico, se le enviará un correo a la persona con un código que tendrá que poner en la plataforma
-------------	--

Nota: tabla hecha por Geffry Santiago Leon Jimenez

Tabla 18 Requisito funcional

cod	RF-10		
titulo	Recuperar contraseña		
prioridad	alta	Fuente	HU-10
necesidad	alta		
Estabilidad	si	Verificable	si
Postcondición	La contraseña se ha cambiado correctamente		
Roles involucrados	Administrador del sistema - cliente		
Descripción	El cliente con su correo electrónico puede cambiar la contraseña de la cuenta		

Nota: tabla hecha por Geffry Santiago Leon Jimenez

Tabla 19 Requisito no Funcional

cod	RF-10		
titulo	Eficacia para recuperar la contraseña		
prioridad	alta	Fuente	RR-10
necesidad	alta		
Estabilidad	si	Verificable	si
Postcondición	La contraseña se ha cambiado correctamente		
Roles involucrados	Administrador del sistema - cliente		
Descripción	A la hora de que la persona quiera cambiar la contraseña tiene que darle a “recuperar contraseña” en la cual aparecerá que tiene que poner el correo para enviar un correo y así aparecer una interfaz donde aparezca donde poner la contraseña nueva y confirmarla		

Nota: tabla hecha por Geffry Santiago Leon Jimenez

3.2 historias de usuario y casos de uso

Tabla 20 Historia de Usuario

Identificador:	HU-01
Nombre:	Iniciar sesión
Actores:	Administrador - cliente
Objetivo:	Iniciar sesión en el sistema
Descripción:	Como cliente quiero que dejen el sistema de inicio de sesión como esta ya que me parece muy bueno
Caso de prueba:	El inicio de sesión es un éxito
Puntos estimados de esfuerzo	El administrador calcula que el tiempo mínimo para iniciar sesión es de 30 segundos, ya que solo es poner el

	correo y la contraseña
observaciones	Este sistema hará que la persona pueda ingresar a la plataforma
Criterios de aceptación	Para ser aceptado debe permitir que la persona ingrese a la plataforma

Nota: tabla hecha por Geffry Santiago Leon Jimenez

Tabla 21 Historia de usuario

Identificador:	HU-02
Nombre:	Añadir usuario - Registrarse
Actores:	Administrador - cliente
Objetivo:	Añadir un usuario al sistema
Descripción:	Como administrador quiero tener un sistema de un botón donde se permita crear un usuario para poder tener un perfil en el que puedan pedir sus préstamos a partir de las características (nombre, edad, celular, Gmail) y que se genere un mensaje en donde me confirme si la operación salió correctamente para un mejor sistema
Caso de prueba:	El usuario se ha añadido correctamente
Puntos estimados de esfuerzo	El administrador a calculado cuando podría demorar en agregar un usuario, el cual dice que demora 1 minuto
observaciones	Esta función ayudara a que la persona acceda al sistema
Criterios de aceptación	Para ser aceptado se necesita que al añadir el usuario se le dé la notificación al cliente de que su usuario ha sido activado correctamente

Nota: tabla hecha por Geffry Santiago Leon Jimenez

Tabla 22 Historia de usuario

Identificador:	HU-03
Nombre:	Eliminar usuario
Actores:	administrador
Objetivo:	Eliminar un usuario del sistema
Descripción:	Como administrador me gustaría tener un sistema en el cual me aparezcan los usuarios que no se han utilizado con un botón que diga eliminar usuarios lo cual sería que eliminaría todos, donde me vote un mensaje de alerta de que si estoy seguro de borrar el usuario y que me aparezca un mensaje de que el usuario fue borrado exitosamente para tener más orden

Caso de prueba:	El usuario se ha eliminado correctamente
Puntos estimados de esfuerzo	El administrador después de poner en desarrollo la eliminación de usuarios a determinado que se demora aproximadamente 10 minutos en eliminar un usuario
observaciones	También debería tener un botón en el cual le permita seleccionar cuales usuarios debería borrar
Criterios de aceptación	Para aceptarlo necesito saber si se realizo correctamente es decir que me vote un mensaje el cual me diga que los usuarios seleccionados han sido eliminados

Nota: tabla hecha por Geffry Santiago Leon Jimenez

Tabla 23 Historia de usuario

Identificador:	HU-04
Nombre:	Añadir material de formación
Actores:	Administrador
Objetivo:	Añadir material de formación al sistema
Descripción:	Como administrador quiero tener un apartado en donde pueda agregar el material de formación a partir de las siguientes características (nombre, autor, número de serie, grado del libro de material de formación) y que se genere un mensaje de confirmación en donde diga que el material de formación se ha añadido correctamente para un mejor orden
Caso de prueba:	El material se ha añadido correctamente
observaciones	Una observación sería que al añadir un material de formación se les notifique a los usuarios
Puntos estimados de esfuerzo	El administrador dice que demora 10 minutos en agregar un material, pero a la hora de hacerlo demoro 15 minutos
Criterios de aceptación	Para ser aceptado se necesita que al momento de querer agregar el libro y ya haber puesto toda la información vote un mensaje de que el material si fue agregado y fue notificado a los usuarios.

Nota: tabla hecha por Geffry Santiago Leon Jimenez

Tabla 24 Historia de usuario

Identificador:	HU-05
Nombre:	Eliminar material de formación

Actores:	Administrador
Objetivo:	Eliminar un material de formación
Descripción:	Como administrador quiero tener un sistema de un botón que me permita eliminar un material de formación que yo seleccione, y que al darle me aparezca una advertencia de que si quiero eliminar ese material acompañado de un mensaje de confirmación que me diga que el material de formación ha sido eliminado exitosamente para poder organizar más el sistema
Caso de prueba:	El material de formación se ha eliminado correctamente
observaciones	Una observación seria que el material que vayas a eliminar sea notificado a los usuarios
Puntos estimados de esfuerzo	El administrador a analizado la forma de eliminar usuarios y a determinado que entre buscar los materiales que borrar se llevaría aproximadamente 20 minutos
Criterios de aceptación	Para que esto sea aceptado al momento de eliminar el material deberá aparecer un mensaje en el que me confirme que el material ha sido eliminado

Nota: ilustración hecha por Geffry Santiago Leon Jimenez

Tabla 25 Historia de usuario

Identificador:	HU-06
Nombre:	Modificar material de formación
Actores:	Administrador
Objetivo:	Modificar un material de formación
Descripción:	Como administrador quiero un sistema en el cual me aparezca un listado en el cual estén todos los materiales de formación el cual tengan algún error, después quiero un botón que me permita modificar el material y que al darle de nuevo genere un mensaje de confirmación en la cual diga que la modificación ha sido un éxito para poder de algún modo ayudar a que el sistema fluya mejor
Caso de prueba:	La modificación ha sido un éxito
observaciones	Una observación para agregar es que hubiera un auto correcto automático en el cual se hiciera parte de este progreso automáticamente para así ahorrar más tiempo
Puntos estimados de esfuerzo	Teniendo en cuenta la observación pasada el administrador del sistema a estimado un tiempo en el cual le toma modificar todos los materiales el cual sería de 30 minutos

Criterios de aceptación	Para esta ser aceptada debe ser puesta a prueba si esta funciona correctamente debe aparecer un mensaje en el cual me diga que el proceso de modificación ha sido terminado correctamente
-------------------------	---

Nota: ilustración hecha por Geffry Santiago Leon Jimenez

Tabla 26 Historia de usuario

Identificador:	HU-07
Nombre:	Consultar material de formación
Actores:	Administrador
Objetivo:	Consultar cuales son los materiales de formación
Descripción:	Como administrador quiero un sistema más avanzado el cual con un solo código pueda encontrar el material de formación que necesite, así se puede ahorrar más tiempo en el momento de buscar un material de formación
Caso de prueba:	El material de formación ha sido encontrado
observaciones	Para sumar a la descripción seria que ese código se les den a los usuarios para que ellos también puedan tener un sistema de búsqueda más rápido
Puntos estimados de esfuerzo	El administrador a detallado la nueva forma que el quiere en encontrar el material para esto analiza cuanto él se podría demorar y cuanto los usuarios se podrían demorar, esto le da que para encontrar un material con el código se demoraría 1 minuto ya que es rápido el proceso de encontrar el material
Criterios de aceptación	Para esto ser aceptado y ya ser considerado terminado se necesita verificar si funciona bien lo cual para esto al momento de buscar con el código el material te ponga un mensaje de que el material a sido encontrado

Nota: ilustración hecha por Geffry Santiago Leon Jimenez

Tabla 27 Historia de usuario

Identificador:	HU-08
Nombre:	prestamos
Actores:	administrador
Objetivo:	Consultar los préstamos de los materiales de formación
Descripción:	Como administrador quiero poner un listado en el cual me aparezcan todos los préstamos que la gente está

	pidiendo, y poner un botón que diga aceptar para ahorrar más tiempo y poder hacer todos los prestamos más rápido
Caso de prueba:	El préstamo ha sido aprobado
observaciones	Los usuarios deberían tener un tiempo estimado de cuanto podría tardar en aceptar su préstamo
Puntos estimados de esfuerzo	El administrador ha analizado cuanto se podría demorar en aceptar todos los préstamos y a calculado que en unas 5 horas podría aceptar los prestamos ya que al tener que leerlos todos se lleva mucho tiempo
Criterios de aceptación	Para ser aceptado se necesita probar este sistema así que para esto cuando se acepten los prestamos se debe notificar con un mensaje a los usuarios de que el préstamo ha sido aceptado

Nota: ilustración hecha por Geffry Santiago Leon Jimenez

Tabla 28 Historia de usuario

Identificador:	HU-09
Nombre:	Pedir préstamo
Actores:	cliente
Objetivo:	Pedir un préstamo
Descripción:	Como cliente quiero que agreguen un sistema en el cual los prestamos sean automáticos, es decir que no lleven tanto tiempo en ser aceptados esto para poder tener el material más rápido
Observaciones	Los administradores deberían tener un estimado de tiempo en el cual acepten las solicitudes
Puntos estimados de esfuerzos	El usuario a dicho que para hacer un préstamo se deberían llevar entre 5 a 6 minutos
Criterios de aceptación	Para poder aceptar esto necesitamos que a la hora de hacer el préstamo se ejecute de forma correcta y alerte de que si se hizo

Tabla 29 Historia de usuario

Identificador:	HU-10
Nombre:	Cambiar contraseña
Actores:	administrador
Objetivo:	Cambiar la contraseña de usuario

Descripción:	Como administrador quiero agregar un sistema de seguridad el cual el usuario pueda cambiar la contraseña con solo el correo si esta se le olvido
Caso de prueba:	Su contraseña se ha cambiado correctamente
Observaciones	También se podría agregar un sistema en el cual si el usuario no se acuerda de su contraseña tiene la oportunidad de cambiarla con un botón
Puntos estimados de esfuerzo	Para cambiar una contraseña el administrador a analizado cuanto se podría demorar y a estimado que se demoraría unos 2 minutos ya que es super fácil con el sistema que tiene
Criterios de aceptación	Para ser aceptado se necesita que a la hora de cambiar la contraseña me vote un mensaje de confirmación el cual diga que la contraseña se ha cambiado correctamente

Nota: ilustración hecha por Geffry Santiago Leon Jimenez

Tabla 30 Caso de uso

Identificador:	CU-01
Nombre:	Iniciar sesión
Actores:	Administrador-usuario
Objetivo:	Añadir un usuario al sistema
Descripción:	1 el usuario administrador entra a la pagina 2 el usuario administrador ingresa el usuario y la contraseña 3 el usuario administrador le da a iniciar sesión
Caso de prueba:	El usuario se ha añadido correctamente

Nota: ilustración hecha por Geffry Santiago Leon Jimenez

Tabla 31 Caso de uso

Identificador:	CU-02
Nombre:	Añadir usuario - Registrarse
Actores:	Administrador - cliente
Objetivo:	Añadir un usuario al sistema
Descripción:	4 el usuario administrador entra a la pagina 5 el usuario administrador le da a donde dice registrarse 6 el usuario administrador pone los datos 7 el usuario administrador le da al botón que dice registrase

	8 se devuelve el mensaje de que la operación ha sido correcta
Caso de prueba:	El usuario se ha añadido correctamente

Nota: ilustración hecha por Geffry Santiago Leon Jimenez

Tabla 32 Caso de uso

Identificador:	CU-03
Nombre:	Eliminar usuario
Actores:	Administrador
Objetivo:	Eliminar usuario del sistema
Descripción:	<p>1 el usuario administrador entra a la pagina</p> <p>2 el usuario administrador revisa los usuarios registrados</p> <p>3 el usuario administrador ve un usuario que no ha usado su cuenta</p> <p>4 el usuario administrador le da al botón de eliminar usuario</p> <p>5 el usuario administrador le da en que está seguro de eliminar el usuario</p> <p>6 se devuelve un mensaje de que la operación ha sido correcta</p>
Caso de prueba:	El usuario se ha eliminado correctamente

Nota: ilustración hecha por Geffry Santiago Leon Jimenez

Tabla 33 Caso de uso

Identificador:	CU-04
Nombre:	Añadir material de formación
Actores:	Administrador
Objetivo:	Añadir un material de formación al sistema
Descripción:	<p>1 el usuario administrador entra a la pagina</p> <p>2 el usuario administrador inicia sesión</p> <p>3 el usuario administrador pulsa sobre el botón añadir material de formación</p> <p>4 el usuario administrador los datos del material que desea añadir</p> <p>5 el usuario administrador le da al botón agregar libro</p>

	6 al usuario administrador le llega un mensaje que confirma que el material se añadió al sistema
Caso de prueba:	El material se ha añadido correctamente

Nota: ilustración hecha por Geffry Santiago Leon Jimenez

Tabla 34 Caso de uso

Identificador:	CU-05
Nombre:	Eliminar material de formación
Actores:	Administrador
Objetivo:	Eliminar un material de formación
Descripción:	<ol style="list-style-type: none"> 1. el usuario administrador ingresa a el sistema 2. el usuario administrador busca el material que desea eliminar 3. el usuario administrador le da el botón de eliminar material de formación 4. el usuario administrador le da al botón de aceptar del mensaje de advertencia 5. el usuario administrador recibe una notificación donde se notifica que el material ha sido eliminado correctamente
Caso de prueba:	El material ha sido eliminado correctamente

Nota: ilustración hecha por Geffry Santiago Leon Jimenez

Tabla 35 Caso de uso

Identificador:	CU-06
Nombre:	Modificar material de formación
Actores:	Administrador
Objetivo:	Modificar un material de formación
Descripción:	<ol style="list-style-type: none"> 1. el usuario administrador ingresa al sistema 2. el usuario administrador selecciona que material de formación modificara 3. el usuario administrador le da al botón de guardar cambios 4. el usuario administrador recibe un mensaje de que el material de formación a sido modificado
Caso de prueba:	La modificación ha sido un éxito

Nota: ilustración hecha por Geffry Santiago Leon Jimenez

Tabla 36 Caso de uso

Identificador:	CU-07
Nombre:	Consultar material de formación
Actores:	Administrador
Objetivo:	Consultar cuales son los materiales de formación
Descripción:	<ol style="list-style-type: none"> 1. el usuario administrador accede al sistema 2. el usuario administrador accede al listado de los materiales de formación 3. el usuario administrador usa la barra de búsqueda para buscar su material de formación 4. el usuario administrador encuentra el material de formación
Caso de prueba:	El material de formación ha sido encontrado

Nota: ilustración hecha por Geffry Santiago Leon Jimenez

Tabla 37 Caso de uso

Identificador:	CU-08
Nombre:	prestamos
Actores:	Administrador
Objetivo:	Aceptar los préstamos de los materiales de formación
Descripción:	<ol style="list-style-type: none"> 1. el usuario administrador accede al sistema 2. el usuario administrador se va al apartado de préstamos 3. el usuario administrador empieza a leer y detallar que es lo que los clientes piden 4. el usuario administrador acepta los préstamos y entrega los materiales de formación
Caso de prueba:	El préstamo ha sido aprobado

Nota: ilustración hecha por Geffry Santiago Leon Jimenez

Tabla 38 Caso de uso

Identificador:	CU-09
Nombre:	Pedir préstamo
Actores:	cliente
Objetivo:	Pedir un préstamo
Descripción:	<ol style="list-style-type: none"> 1. el usuario cliente accede al sistema

	2. el usuario cliente ingresa al apartado de prestamos 3. el usuario cliente pasa los materiales que necesita 4. el usuario cliente recibe respuesta del administrador 5. el usuario cliente recibe un mensaje el cual notifica que el préstamo fue aceptado
Caso de prueba:	El préstamo fue aceptado

Nota: ilustración hecha por Geffry Santiago Leon Jimenez

Tabla 39 Caso de uso

Identificador:	CU-10
Nombre:	Cambiar contraseña
Actores:	administrador
Objetivo:	Cambiar la contraseña del usuario
Descripción:	1. el usuario administrador accede de acceder al sistema 2. el usuario administrador le da al botón de cambiar contraseña 3. el usuario administrador pone su nueva contraseña y le da al botón de cambiar contraseña 4. el usuario administrador recibe un mensaje afirmativo en el cual dice que la contraseña se ha cambiado
Caso de prueba:	La contraseña se ha cambiado correctamente

Nota: ilustración hecha por Geffry Santiago Leon Jimenez

3.3 IDE de desarrollo

Para este proyecto estamos utilizando la plataforma de Apache NetBeans, Apache es un IDE que basa en el lenguaje de java y ejecutado en swing, este mismo ayuda en el desarrollo de cualquier tipo de software y en sus distintas fases. (IMMUNE, 2022)

3.3.1 Características de NetBeans

- proporciona una base modular y extensible para el desarrollo de aplicaciones

- soporte para múltiples lenguajes
- compatibilidad con multiplataformas
- integración con bases de datos
- soporte para aplicaciones web
- software libre y gratuito

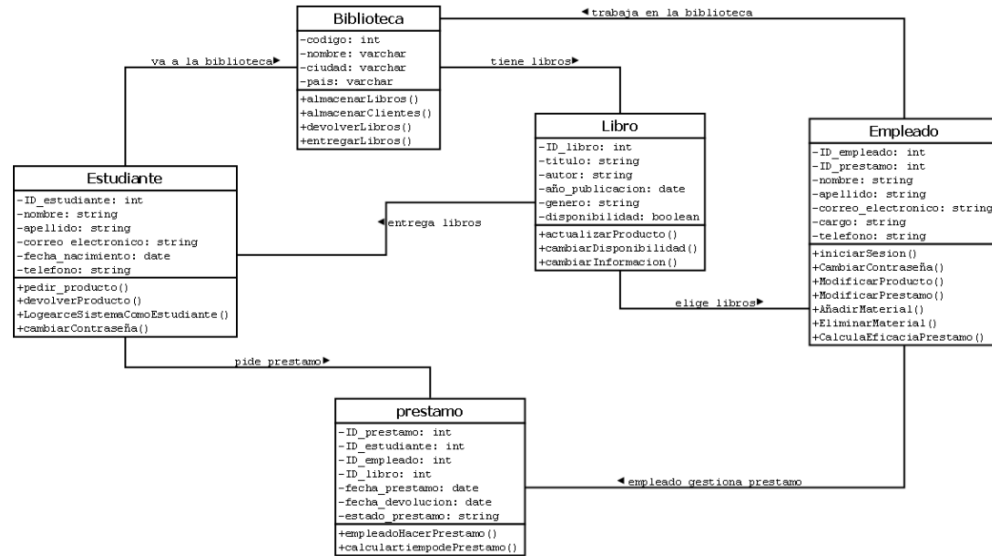
3.3.2 Ventajas de Usar NetBeans

Esta plataforma tiene la oportunidad de brindarnos varios beneficios a la hora de ser utilizada así que dejare unos a continuación:

- código abierto: NetBeans es de código abierto y gratuito, así cualquiera puede utilizarlo.
- multiplataforma: este funciona en diferentes dispositivos como también diferentes sistemas operativos como Windows, Linux etc.
- multilenguaje: aunque su principal lenguaje sea java puede operar otros lenguajes ya sea PHP, C o Ruby, entre muchos otros.

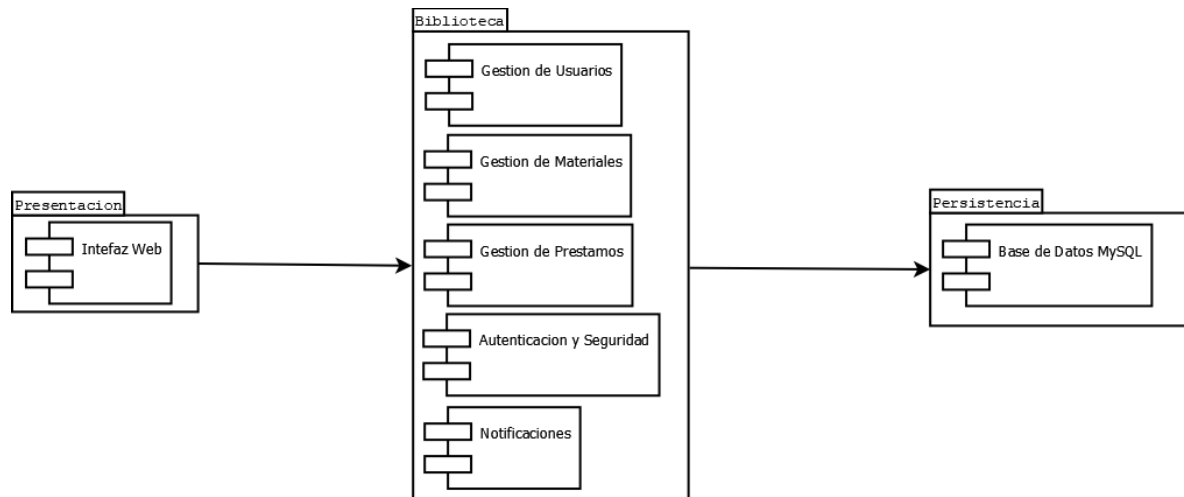
3.4 Diagramas

Ilustración 1 Diagrama de Clases



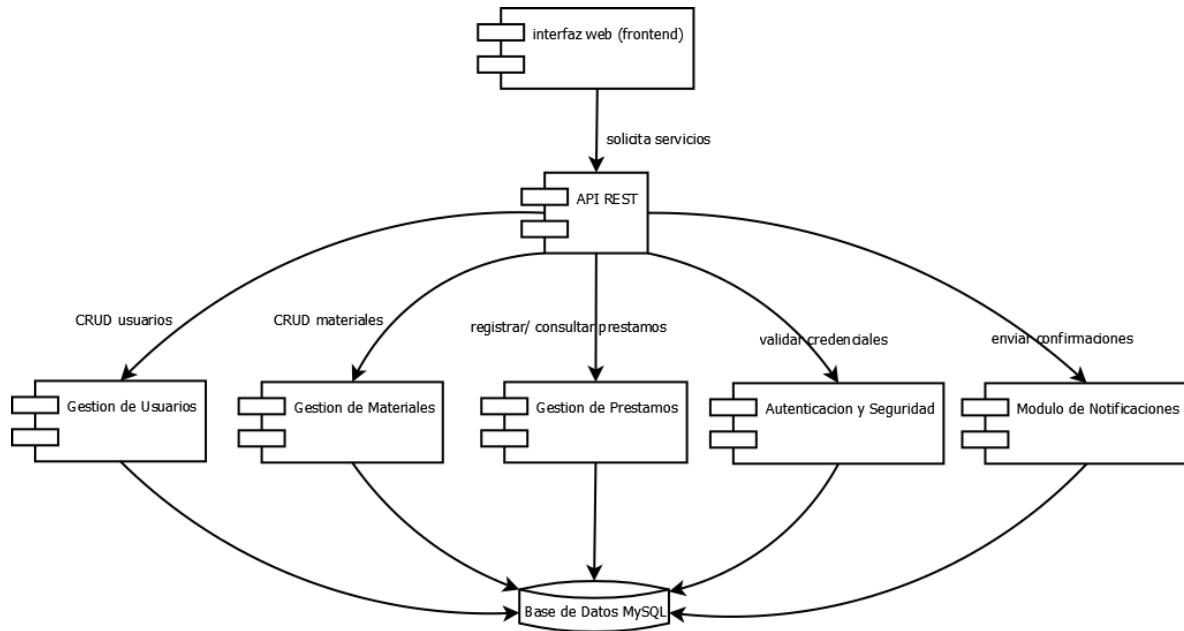
Nota: ilustración hecha por Geffry Santiago Leon Jimenez

Ilustración 2 Diagrama de Paquetes



Nota: ilustración hecha por Geffry Santiago Leon Jimenez

Ilustración 3 Diagrama de Componentes



Nota: ilustración hecha por Geffry Santiago Leon Jimenez

3.5 Mecanismos de Seguridad

Mi proyecto de software tiene un mecanismo de seguridad muy básico el cual es el API, el API que eh configurado sirve para que el cliente puede ingresar a la plataforma con las credenciales que anteriormente en el registro haya rellenado, y próximamente añadiré otra API para el cambio de contraseña que funcionara enviando código a la persona y con ese código pueda cambiar la contraseña.

A continuación, les dejare una ilustración del código de la API:

Ilustración 4 Sistema de Seguridad API

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/JSP_Servlet/Servlet.java to edit this template
 */
package Servlets;

import java.io.IOException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

//esto indica que el servlet accede con la ruta /InicioServlet//
@WebServlet("/InicioServlet")
public class InicioServlet extends HttpServlet {

    //conexion con la base de datos//
    private static final String DB_URL = "jdbc:mysql://localhost:3306/proyecto_software";
    private static final String DB_USER = "root";
    private static final String DB_PASS = "2007";

    //maneja las peticiones POST ya que en inicio.jsp se manda con method=POST//
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        //aquí se captura los datos del formulario//
        String email = request.getParameter("email");
        String contraseña = request.getParameter("contraseña");
    }
}
```

Nota: ilustración hecha por Geffry Santiago Leon Jimenez

Ilustración 5 Sistema de Seguridad API

```
try {

    Class.forName("com.mysql.cj.jdbc.Driver");

    Connection conn = DriverManager.getConnection(DB_URL, DB_USER, DB_PASS);

    //este apartado consulta si existe un usuario con el email y la contraseña que ingresen en inicio.jsp//
    String sql = "SELECT * FROM usuario WHERE email=? AND contraseña=?";
    PreparedStatement stmt = conn.prepareStatement(sql);
    stmt.setString(1, email);
    stmt.setString(2, contraseña);

    ResultSet rs = stmt.executeQuery();

    //aquí se define el resultado, si encuentra al usuario lo manda a bienvenido.jsp//
    if (rs.next()) {
        request.setAttribute("mensaje", "Inicio de sesión exitoso, bienvenido " + rs.getString("nombre_apellido"));
        request.getRequestDispatcher("bienvenido.jsp").forward(request, response);
    } //si no se encuentra, le salta un error y lo devuelve a inicio.jsp//
    else {
        request.setAttribute("mensaje", "Correo o contraseña incorrectos");
        request.getRequestDispatcher("inicio.jsp").forward(request, response);
    }

    rs.close();
    stmt.close();
    conn.close();

    //cuando ocurre un error (la base no se conecta) se muestra el error en inicio.jsp//
} catch (Exception e) {
    e.printStackTrace();
    request.setAttribute("mensaje", "Error en el login: " + e.getMessage());
    request.getRequestDispatcher("inicio.jsp").forward(request, response);
}
}
```

Nota: ilustración hecha por Geffry Santiago Leon Jimenez

3.6 Repositorio en GitHub

Para guardar mi información y actualización del proyecto lo hago en la plataforma de GitHub, esta se me hace sencilla y practica de usar, con esta también puedo mantenerme en orden y liberar espacio de mi pc al guardarlo en la nube.

<https://github.com/Geffry-2007/Mi-proyecto>

3.7 Metodología de Software

Para el desarrollo individual de este proyecto, eh optado en utilizar Kanban como método de gestión del trabajo. Aunque Kanban no se considera una metodología ágil formal como scrum, si se llega a alinear con los principios agiles, ofreciendo una forma visual, flexible y eficiente de organizar tareas.

Como el proyecto es realizado por mí, Kanban me resulta más adecuado, ya que no requiere de roles definidos ni eventos estructurados. Esto me permite avanzar de forma continua, adaptarme a los cambios y mantener una visión clara del progreso que llevo en el proyecto.

3.8 Capas de los componentes

Guiándonos por mi metologia la cual es Kanban, mostrare en que capas van mis componentes, pero para esto me basare en la tabla Kanban, así que hare una representación gráfica de cómo va mi proyecto:

Ilustración 6 Tabla kanvan

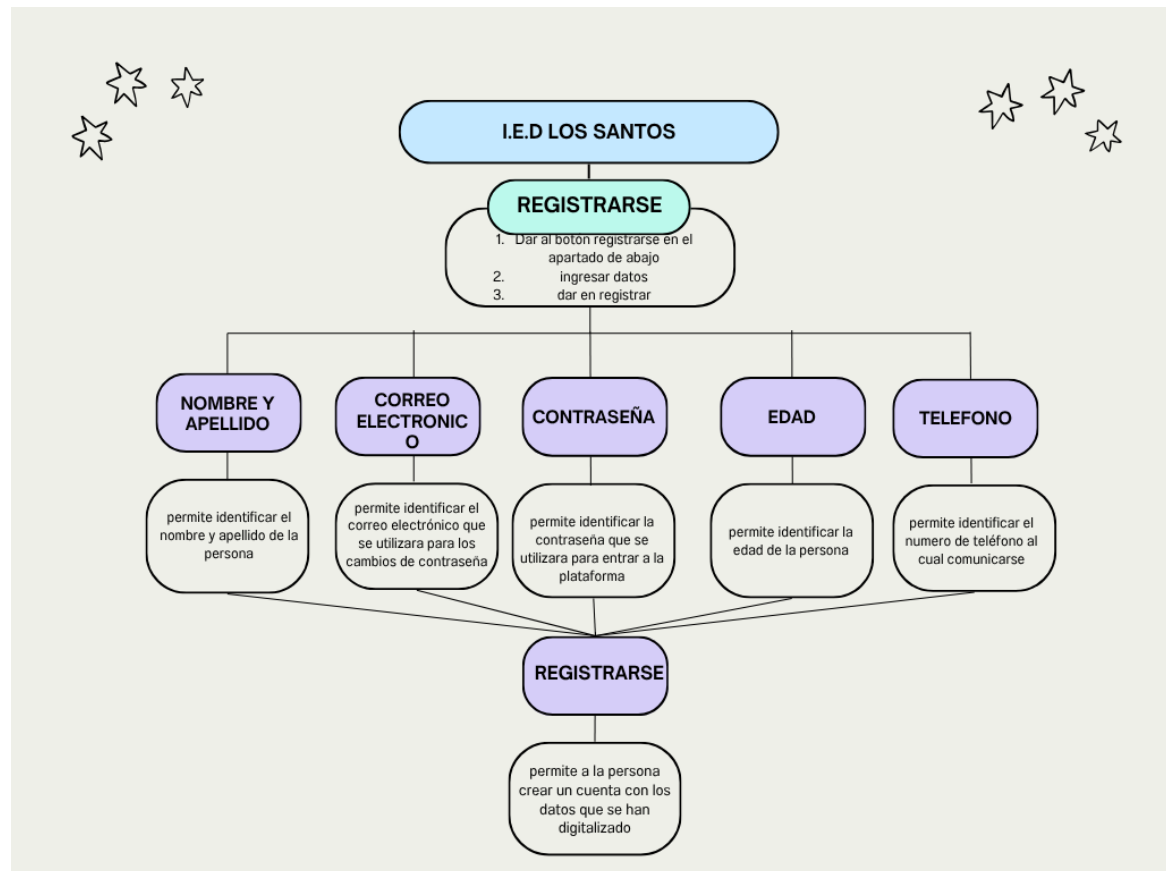


Nota: Ilustración hecha por Geffry Santiago Leon Jimenez

En mi caso yo creo que estaría en la capa de proceso, ya que estoy realizando los módulos faltantes.

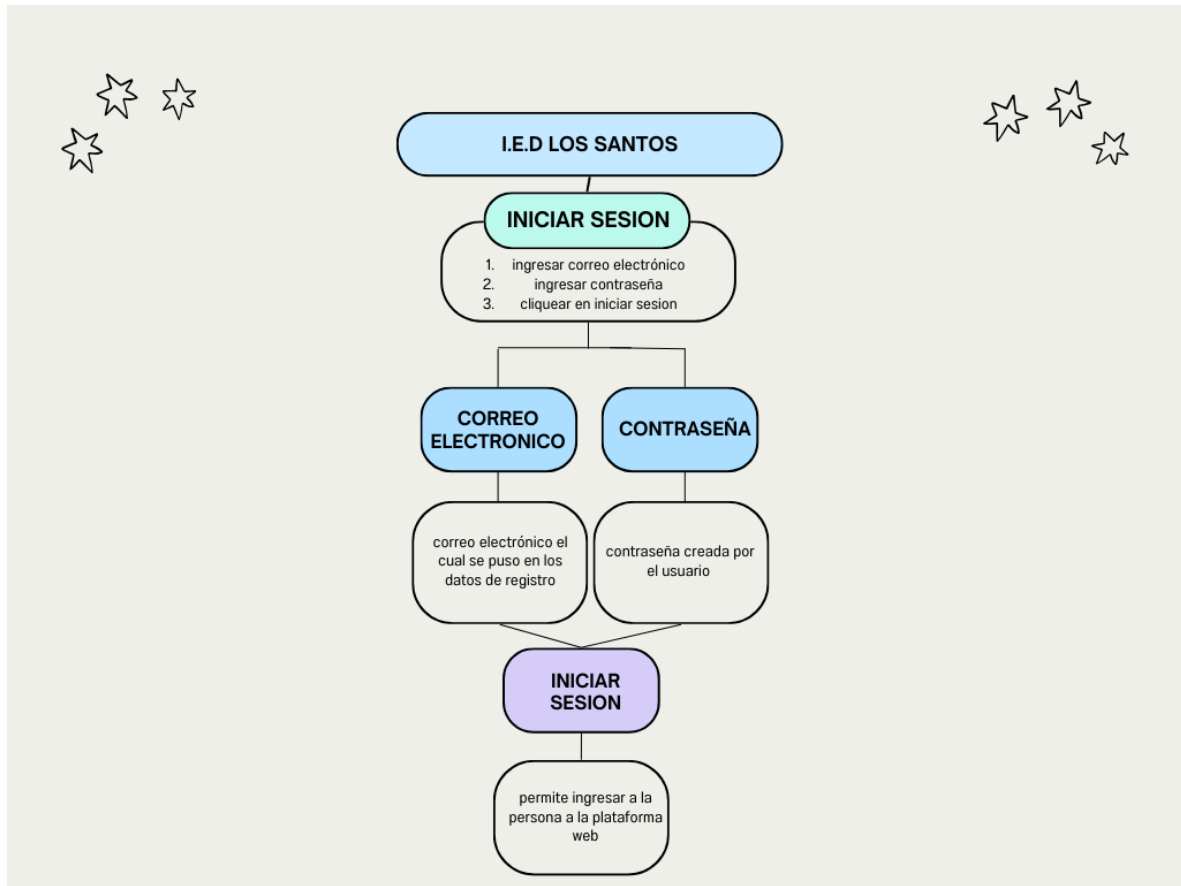
3.9 Mapa de navegación

Ilustración 7 Mapa de Navegación Registrarse



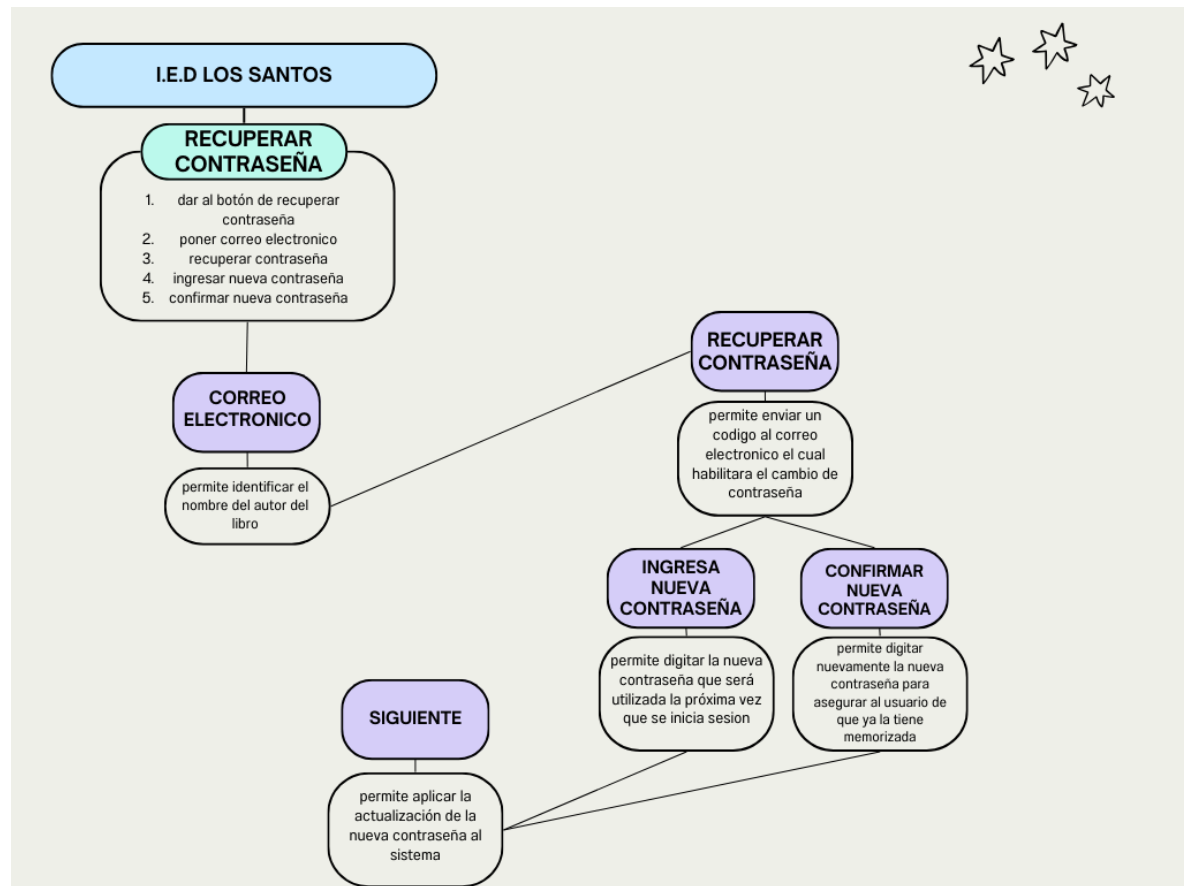
Nota: ilustración hecha por Geffry Santiago Leon Jimenez

Ilustración 8 Mapa de Navegación Inicio Sesión



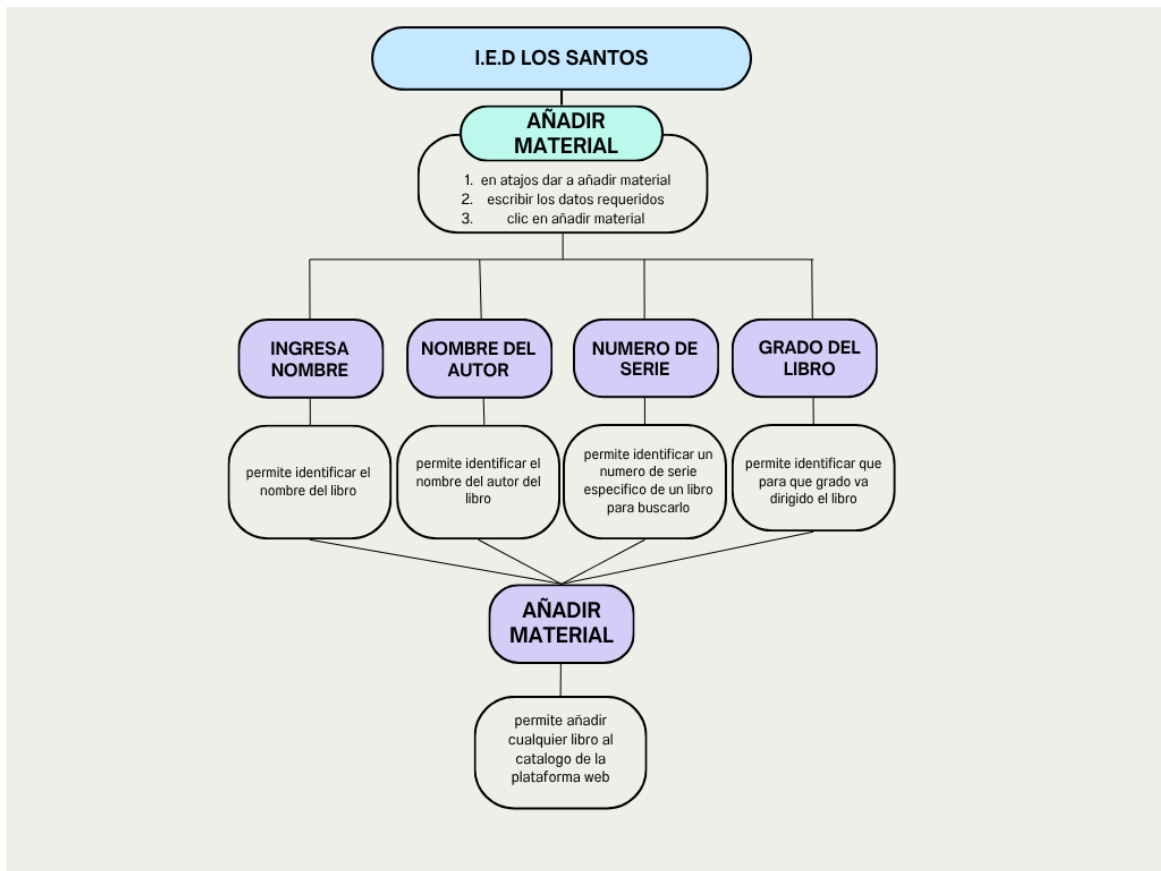
Nota: ilustración hecha por Geffry Santiago Leon Jimenez

Ilustración 9 Mapa de Navegación Recuperar Contraseña



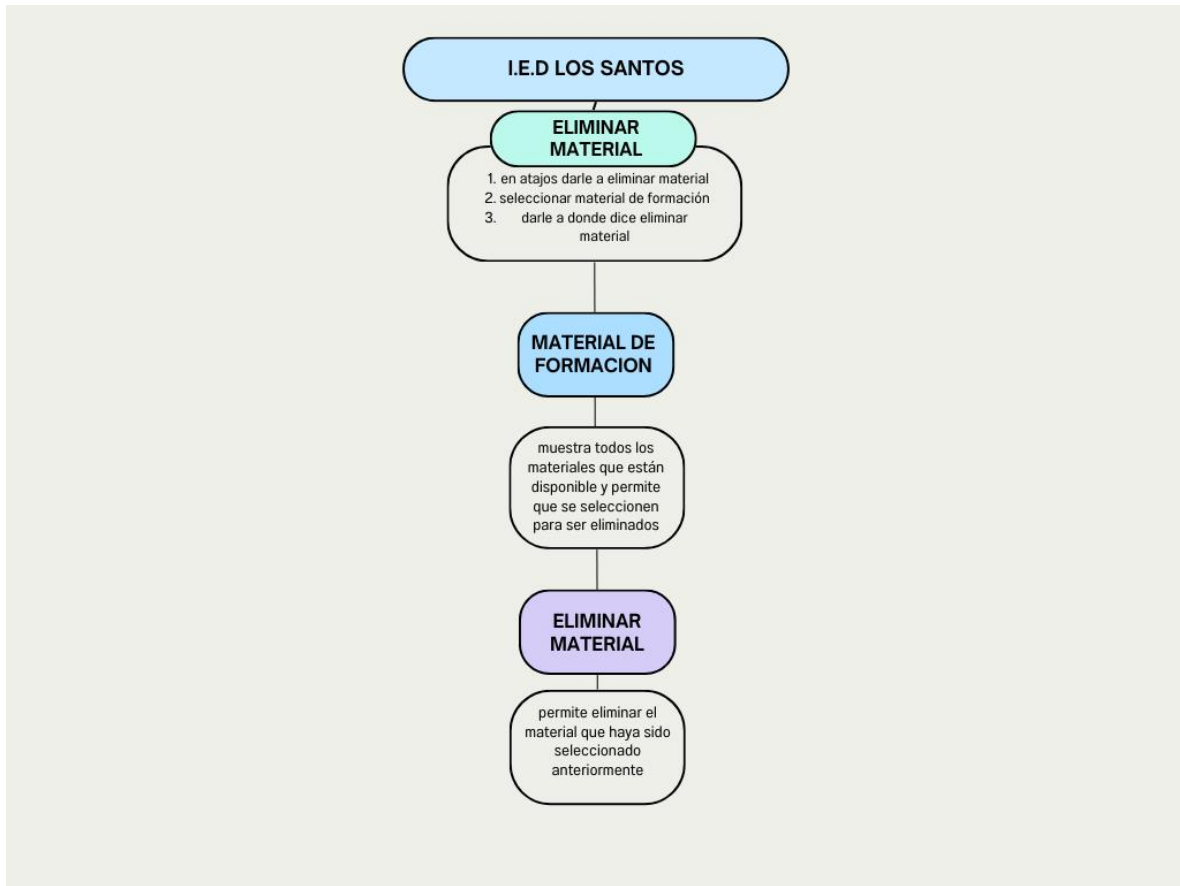
Nota: ilustración hecha por Geffry Santiago Leon Jimenez

Ilustración 10 Mapa de Navegación Añadir Material



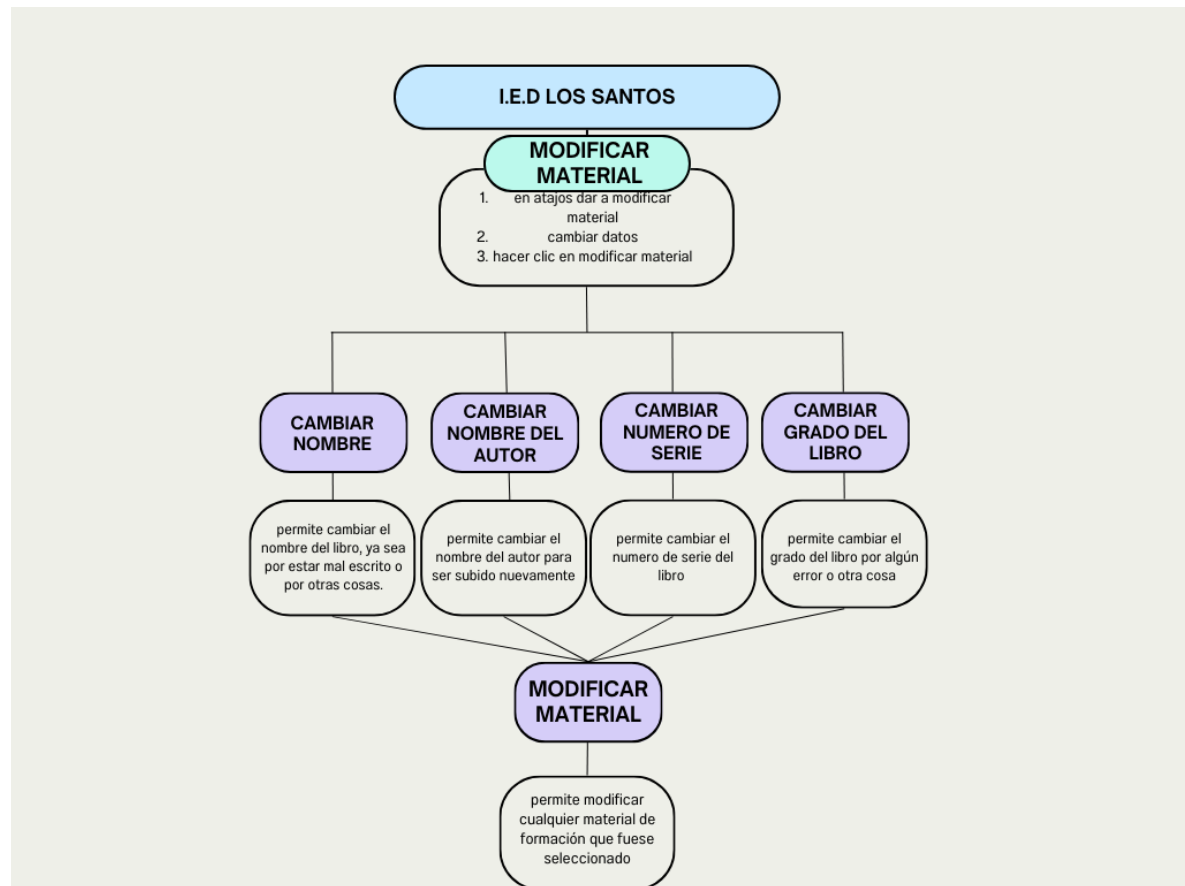
Nota: ilustración hecha por Geffry Santiago Leon Jimenez

Ilustración 11 Mapa de Navegación Eliminar Material



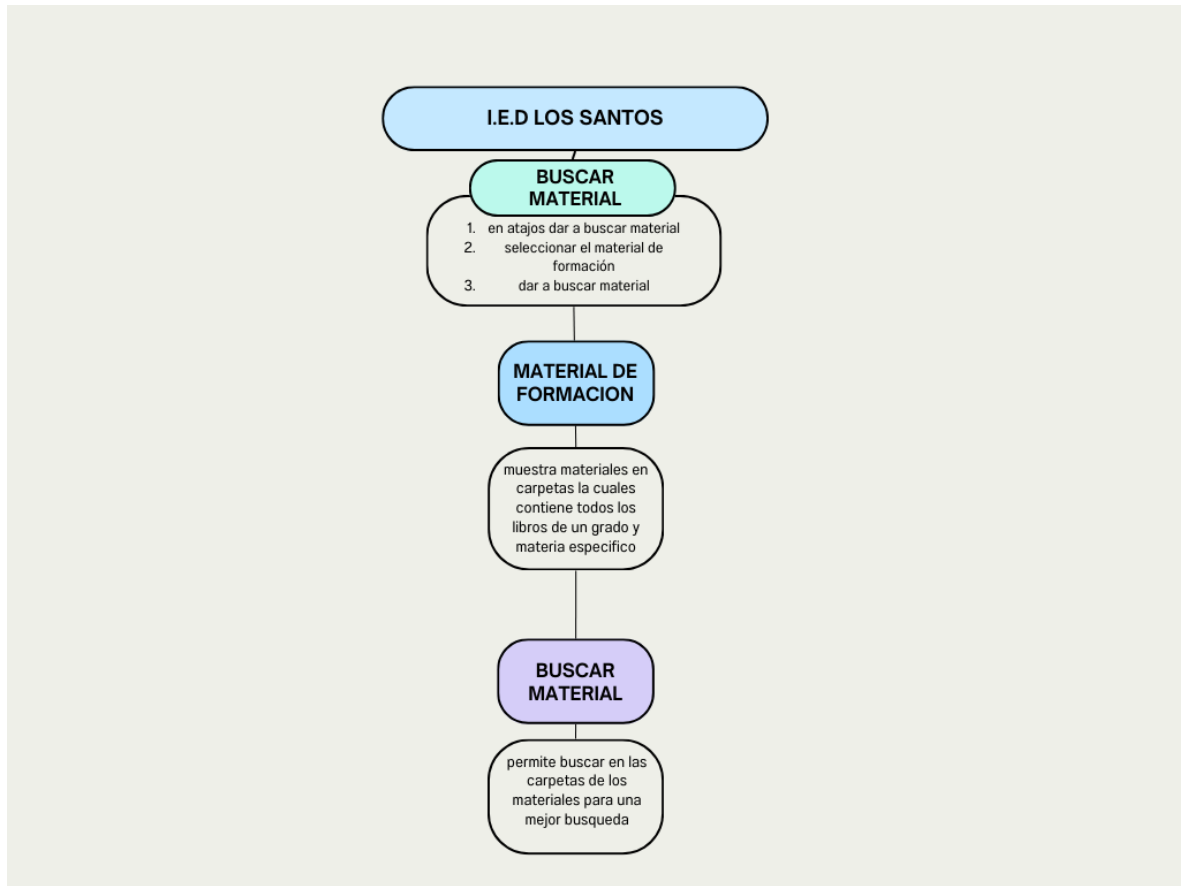
Nota: ilustración hecha por Geffry Santiago Leon Jimenez

Ilustración 12 Mapa de Navegación Modificar Material

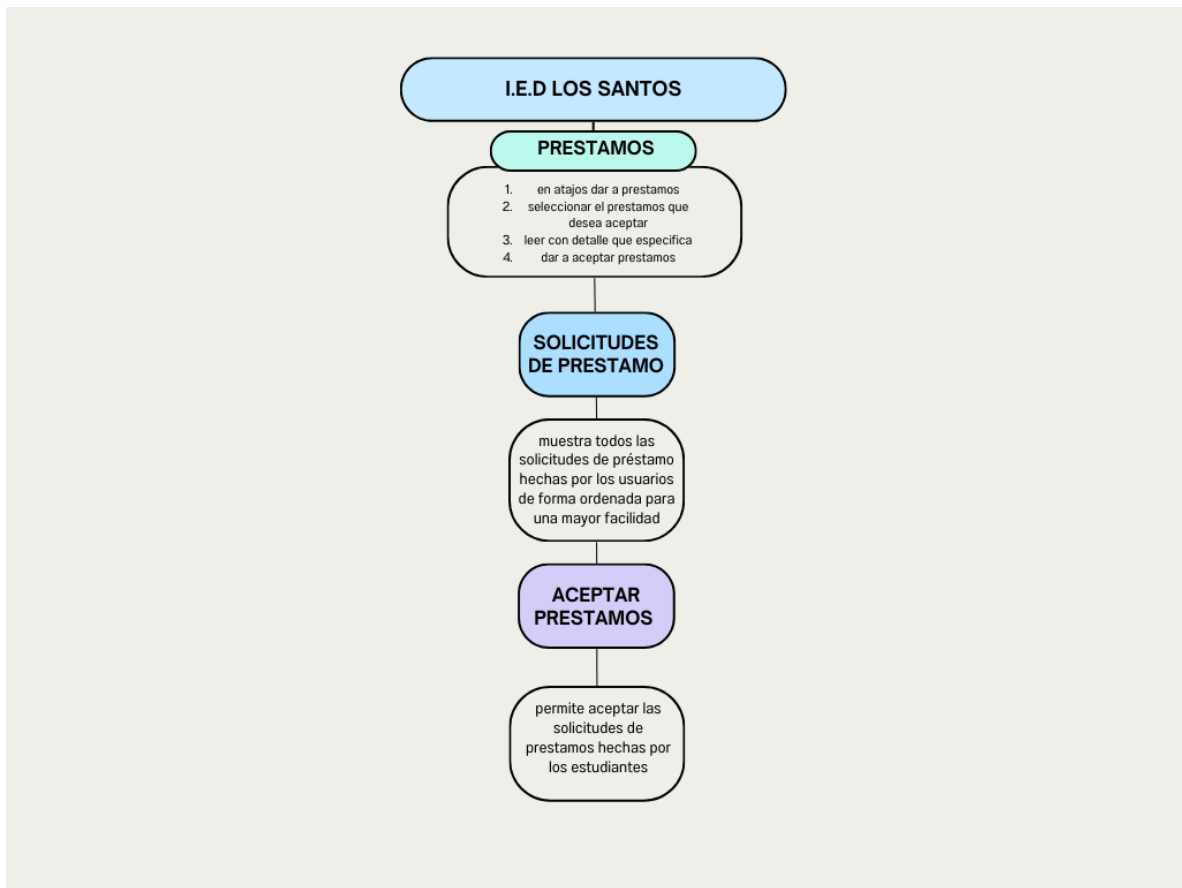


Nota: ilustración hecha por Geffry Santiago Leon Jimenez

Ilustración 13 Mapa de Navegación Buscar Material



Nota: ilustración hecha por Geffry Santiago Leon Jimenez



3.10 Codificación de los módulos

En este apartado colocare un fragmento del código como me lo indico el instructor, cabe aclarar que llevo pocos módulos codificados ya que me lo estoy tomando con calma para aprender de la mejor manera a continuación los fragmentos:

```

<!-- en esta parte de la division colocaremos donde y con que metodo se enviarian los datos, ademas de que añadiremos un formulario que r
<form action="InicioServlet" method="POST">
  <input type="email" name="email" placeholder="Correo electrónico" required>
  <input type="password" name="contraseña" placeholder="Contraseña" required>
  <button type="submit">Iniciar sesión</button>
</form>

<!-- esta parte nos da el mensaje de error que genera si no ponen los datos de manera correcta -->
<% String mensaje = (String) request.getAttribute("mensaje");
if (mensaje != null) { %>
<div style="color: white; font-weight: bold;">
  <%= mensaje %>
</div>
<% } %>

<!-- en esta division podremos colocar link que te direccionen a otras partes, esto se hace con el href -->
<div class="links">
  <p>¿Todavía no tienes cuenta?</p>
  <a href="registro.jsp">Registrarse</a><br>
  <a href="#">Recordar Contraseña</a>
</div>
</div>

<!-- esta parte termina de configurar la animacion de aparicion -->
<div class="custom-shape-divider-top-1751419049">
  <svg data-name="Layer 1" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 1200 120" preserveAspectRatio="none">
    <path d="M321.39,56.44c59-10.79,114.16-30.13,172-41.86,82.39-16.72,168.19-17.73,250.45-.39c823.78,31,906.67,72,985.66,92.
      83c70.05,18.48,146.53,26.09,214.34,3V0H0V27.35A600.21,600.21,0,0,0,321.39,56.44z" class="shape-fill"></path>
  </svg>
</div>
</body>
</html>

```

Ilustración 15 Codificación Modulo Registrarse

```

71 |         <!-- este apartado es el mensaje de error que se pone al no poner todos los datos -->
72 |         <% String mensaje = (String) request.getAttribute("mensaje");
73 |         if (mensaje != null) { %>
74 |             <p style="color: red; font-weight: bold;"><%= mensaje %></p>
75 |         <% } %>
76 |     <!-- este es el cuestionario que recolecta los datos y los manda al servlet -->
77 |     <form action="RegistroServlet" method="POST">
78 |         <input type="text" name="nombre_apellido" placeholder="Nombre completo" required>
79 |         <input type="text" name="telefono" placeholder="Teléfono">
80 |         <input type="number" name="edad" min="7" max="50" placeholder="Edad" required>
81 |         <input type="email" name="email" placeholder="Correo electrónico" required>
82 |         <input type="password" name="contraseña" placeholder="Contraseña" required>
83 |         <button type="submit">Registrarse</button>
84 |     </form>
85 | </div>
86 | </body>
87 | </html>
88 |

```

40

Ilustración 16 Codificación Modulo Índex

```
72     border: 2px solid #000;
73   }
74   /* boton rojo con un texto blanco */
75   .btn-iniciar {
76     background-color: #ff2d2d;
77     color: white;
78   }
79   /* se configura la imagen avatar.png de los botones */
80   .button img {
81     width: 30px;
82     height: 30px;
83     margin-right: 10px;
84     border-radius: 50%;
85     background: #ddd;
86     padding: 3px;
87   }
88   /* cuando el usuario pasa el raton por el boton baja la opacidad */
89   .button:hover {
90     opacity: 0.8;
91   }
92 </style>
93 </head>
94 <body>
95   <!-- en este apartado podemos poner el titulo en la franja roja -->
96   <div class="header">
97     INSTITUCIÓN EDUCATIVA LOS SANTOS<br>
98     DONDE SACAMOS A RELUCIR LAS MENTES MAESTRAS
99   </div>
100  <!-- permite colocar la imagen caminando.png -->
101  <div class="container">
102    <div class="left">
103      
104    </div>
105    <!-- en este apartado podremos colocar las imagenes de los botones -->
106    <div class="right">
```

Nota: ilustración hecha por Geffry Santiago Leon Jimenez

3.11 Bibliotecas usadas en los módulos

En este apartado mostrare las bibliotecas que utilizare para la realización de mi proyecto de software. Cabe aclarar que mi proyecto es Maven y de por si utilizo dos bibliotecas predeterminadas una de estas en jakarta EE, esta me permite crear servlets, documento JPS, conexión JDBC, validación de datos, y muchas cosas. También tengo de utilidad el MySQL conector que me permite conectarme a la base de datos que tenga en MySQL pruebas aquí:

Ilustración 17 Bibliotecas Usadas

```
<dependencies>
  <dependency>
    <groupId>jakarta.platform</groupId>
    <artifactId>jakarta.jakartaee-api</artifactId>
    <version>${jakartaee}</version>
    <scope>provided</scope>
  </dependency>
  <dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.4.0</version>
  </dependency>
</dependencies>
```

Nota: ilustración hecha por Geffry Santiago Leon Jimenez

Ahora procederé a mostrar en que parte de mis módulos son utilizados estas bibliotecas

3.11.1 Modulo Inicio de Sesión

Ilustración 18 Import y @WebServlet

```
import java.io.IOException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

//esto indica que el servlet accede con la ruta /InicioServlet//
@WebServlet("/InicioServlet")
public class InicioServlet extends HttpServlet {
```

Nota: ilustración hecha por Geffry Santiago Leon Jimenez

Ilustración 19 Conexión Base de Datos y Consulta de Credenciales

```
Connection conn = DriverManager.getConnection(DB_URL, DB_USER, DB_PASS);

//este apartado consulta si existe un usuario con el email y la contraseña que ingresen en inicio.jsp//
String sql = "SELECT * FROM usuario WHERE email=? AND contraseña=?";
PreparedStatement stmt = conn.prepareStatement(sql);

//aquí se define el resultado, si encuentra al usuario lo manda a bienvenido.jsp//
if (rs.next()) {
    request.setAttribute("mensaje", "Inicio de sesión exitoso, bienvenido " + rs.getString("nombre_apellido"));
    request.getRequestDispatcher("bienvenido.jsp").forward(request, response);
//si no se encuentra, le salta un error y lo devuelve a inicio.jsp//
} else {
    request.setAttribute("mensaje", "Correo o contraseña incorrectos");
    request.getRequestDispatcher("inicio.jsp").forward(request, response);
}
```

Nota: Ilustración hecha por Geffry Santiago Leon Jimenez

3.11.2 Modulo de Registro

Ilustración 20 Import y @webServlet

```
import java.io.IOException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

@WebServlet("/RegistroServlet")
public class RegistroServlet extends HttpServlet {
```

Nota: ilustración hecha por Geffry Santiago Leon Jimenez

Ilustración 21 Conexión a Base de Datos MySQL

```
Class.forName("com.mysql.cj.jdbc.Driver");
Connection conn = DriverManager.getConnection(DB_URL, DB_USER, DB_PASS);
```

nota: ilustración hecha por Geffry Santiago Leon Jimenez

Ilustración 22 Implementación de Datos de Cuestionario

```
//aquí se inserta los datos del formulario, cada ? se reemplazan por los datos del formulario//  
String sql = "INSERT INTO usuario (nombre_apellido, telefono, edad, email, contraseña, rol) VALUES (?, ?, ?, ?, ?, ?)";
```

Nota: ilustración hecha por Geffry Santiago Leon Jimenez

Ilustración 23 Confirmación de Registro

```
//si el numero de filas es mayor a 0, significa que el registro fue exitoso//  
if (filas > 0) {  
    request.setAttribute("mensaje", "Registro exitoso");  
    request.getRequestDispatcher("inicio.jsp").forward(request, response);  
//si no se cumple, aparece un mensaje y te devuelve a registro.jsp//  
} else {  
    request.setAttribute("mensaje", "No se pudo registrar");  
    request.getRequestDispatcher("registro.jsp").forward(request, response);  
}
```

Nota: Ilustración hecha por Geffry Santiago Leon Jimenez

3.12 Framework utilizados

en este proyecto que se encuentra en desarrollo se está utilizando jakarta EE, aunque no sea como tal un framework completo llega a proporcionar las herramientas necesarias para manejar peticiones HTTP, redirigir paginas JSP y conectar con mis bases de datos mediante JDBC. esta plataforma permite estructurar la lógica del servidor de forma clara y modular, facilitando el desarrollo de web educativas, y como dato adicional los lenguajes que se están utilizando son HTML, CSS, JavaScript y Java en el Backend, lo que me permite una integración fluida entre la interfaz de usuario y lógica del sistema.

3.13 Fragmentos de código reutilizables

Ilustración 24 Cabeza de página web

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>  
<!-- en esta parte configuraremos el idioma de la pagina, que reciba caract  
<!DOCTYPE html>  
<html lang="es">  
<head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <title>Inicio de sesión - I.E.D Los Santos</title>
```

Nota: ilustración hecha por Geffry Santiago Leon Jimenez

Este fragmento de código es la cabeza del documento donde se define el idioma, caracteres especiales, titulo, etc.

Ilustración 25 Formulario

```
<form action="RegistroServlet" method="POST">
  <input type="text" name="nombre_apellido" placeholder="Nombre completo" required>
  <input type="text" name="telefono" placeholder="Teléfono">
  <input type="number" name="edad" min="7" max="50" placeholder="Edad" required>
  <input type="email" name="email" placeholder="Correo electrónico" required>
  <input type="password" name="contraseña" placeholder="Contraseña" required>
  <button type="submit">Registrarse</button>
</form>
```

Nota: ilustración hecha por Geffry Santiago Leon Jimenez

Este fragmento es un cuestionario el cual se puede utilizar en todos los módulos que pidan datos, en mi caso lo tendré que utilizar mas adelante en otros módulos como añadir libros.

Ilustración 26 Mensaje de Error

```
<% String mensaje = (String) request.getAttribute("mensaje");
if (mensaje != null) { %>
  <p style="color: red; font-weight: bold;"><%= mensaje %></p>
<% } %>
```

nota: Ilustración hecha por Geffry Santiago león Jimenez

este fragmento es el código de error que vota al poner mal los caracteres, se puede copiar y lo único que tocaría modificarle es el nombre del código con el que lo llamas desde los servlets

Ilustración 27 Links

```
<div class="links">
    ¿Todavía no tienes cuenta?
    <a href="registro.jsp">Registrarse</a><br>
    <a href="#">Recordar Contraseña</a>
</div>
```

Nota: Ilustración hecha por Geffry Santiago Leon Jimenez

Este fragmento es el apartado de links, o los mini botones los cuales te redirigen a otros módulos, en este caso se utiliza para enviar a registro.jsp

Ilustración 28 Captura de Datos

```
String nombre_apellido = request.getParameter("nombre_apellido");
String telefono = request.getParameter("telefono");
String edad = request.getParameter("edad");
String email = request.getParameter("email");
String contraseña = request.getParameter("contraseña");
```

Nota: Ilustración hecha por Geffry Santiago Leon Jimenez

Este fragmento se encarga de capturar los datos que se piden en un formulario, en este caso este es el fragmento donde llega la información del formulario de registro.

Ilustración 29 Conexión a Base de Datos

```
Class.forName("com.mysql.cj.jdbc.Driver");
Connection conn = DriverManager.getConnection(DB_URL, DB_USER, DB_PASS);
```

Nota: Ilustración hecha por Geffry Santiago Leon Jimenez

Este fragmento es la conexión a la base de datos mediante el JDBC, este se utiliza en todos los servlets que requieran conexión a base de datos.

Ilustración 30 Insertar Datos en Tabla

```
String sql = "INSERT INTO usuario (nombre_apellido, telefono, edad, email, contraseña, rol) VALUES (?, ?, ?, ?, ?, ?)";
```

Nota: Ilustración hecha por Geffry Santiago Leon Jimenez

Este fragmento se encarga de mandar los datos a la base de datos

Ilustración 31 Confirmación

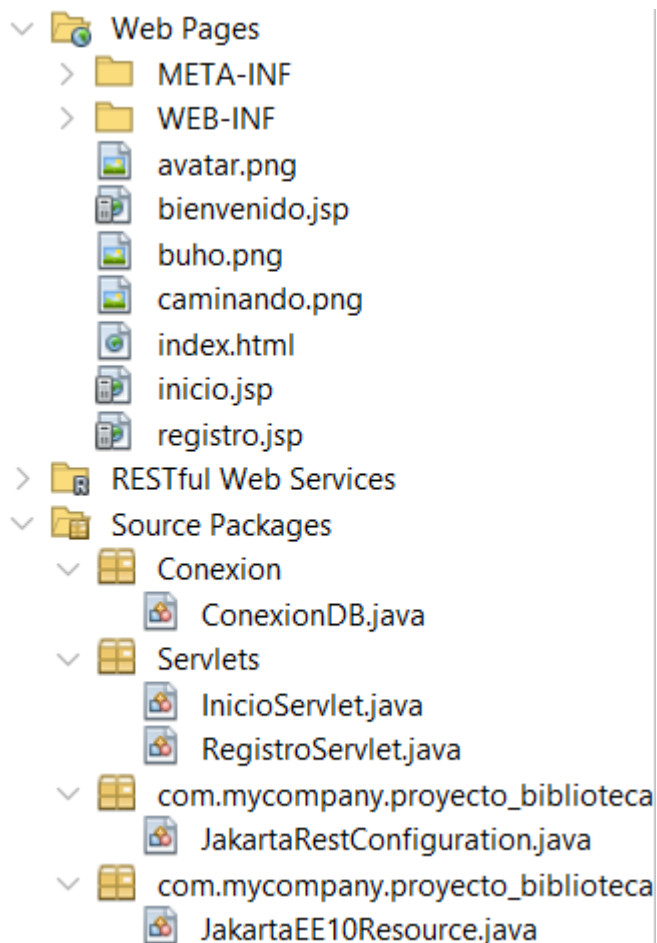
```
request.setAttribute("mensaje", "Registro exitoso");  
request.getRequestDispatcher("inicio.jsp").forward(request, response);
```

Nota: ilustración hecha por Geffry Santiago Leon Jimenez

En este fragmento se encarga de redireccionar a otro modulo con un mensaje, como podemos ver si el registro está mal te devuelve, pero si está bien te manda a inicio.

3.13 Organización en paquetes

Ilustración 32 Organización en Paquetes



Nota: Ilustración hecha por Geffry Santiago Leon Jimenez}

Podemos ver en esta ilustración como está organizado mi proyecto, en la carpeta web Pages tengo los módulos realizados como el de inicio, registro y index, y podemos ver que tenemos unas imágenes en la misma carpeta para un uso mas fácil, mas abajo en la de source packages tenemos 4 paquetes, las que yo cree fue la de conexión y servlets.

En la de conexión tengo el un archivo llamado ConexionDB.java, este me permite cargar el controlador JDBC.

En el paquete de Servlets tenemos InicioServlet.java y RegistroServlet.java, estos dos se encargan de recibir la información dada en los módulos de inicio y registro, en el caso de InicioServlet este recibe el correo y contraseña que el usuario metió y verifica si esta correctamente escrito, si esta mal escrito o no existe ese usuario mandara un error de inicio de sesión.

En el caso de RegistroServlet este recibe la información del formulario hecho en registro.jsp, y este se encarga de añadirlo a la base de datos de forma automática para que el usuario quede inscrito y pueda ingresar a la plataforma.

Podemos ver otras dos carpetas, pero esa las crea automáticamente la IDE de desarrollo.

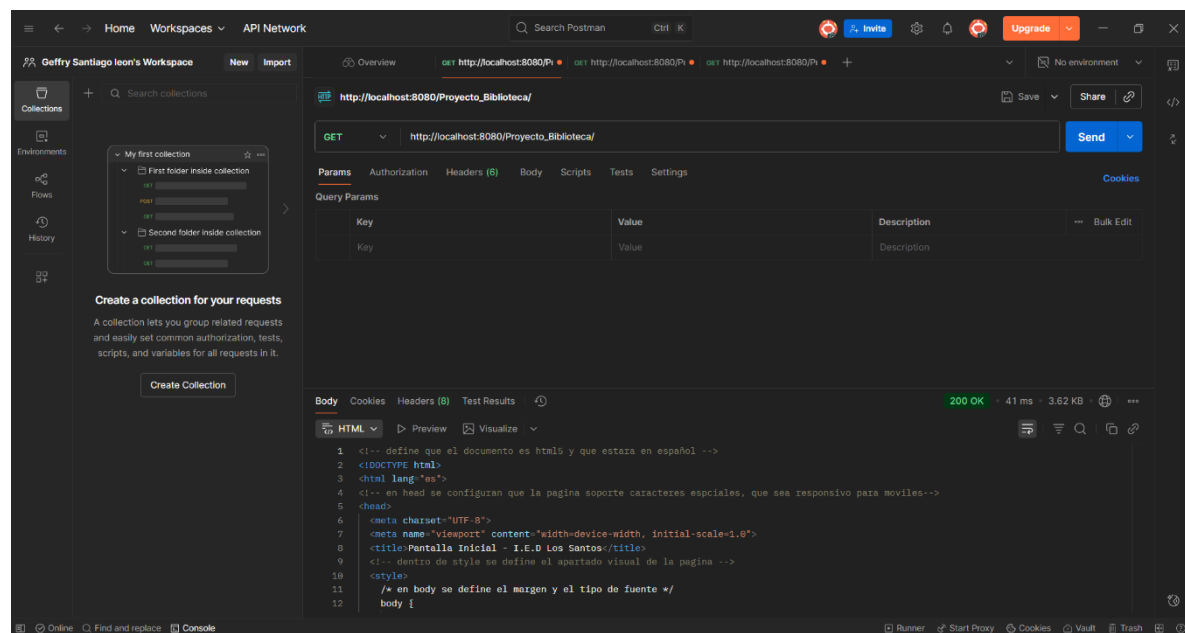
3.14 Patrones de diseño

En el mundo del software hay varios patrones de diseño que se pueden utilizar, en este proyecto se puede reflejar el MVC “modelo-vista-controlador” este separa el sistema en tres capas, modelo que es la representación de los datos y la lógica del negocio, vista que es la interfaz que ve el usuario y el controlador que es el que recibe las peticiones del usuario desde la vista.

En este momento el proyecto no ha avanzado mucho por lo tanto no tiene muchos patrones de diseño visibles, pero en un futuro se añadirían mas patrones como DAO para separar consultas SQL, Factory para manejar diferentes tipos de usuario ya que mi proyecto tiene un sistema de rangos, Observer para implementar notificaciones en el sistema.

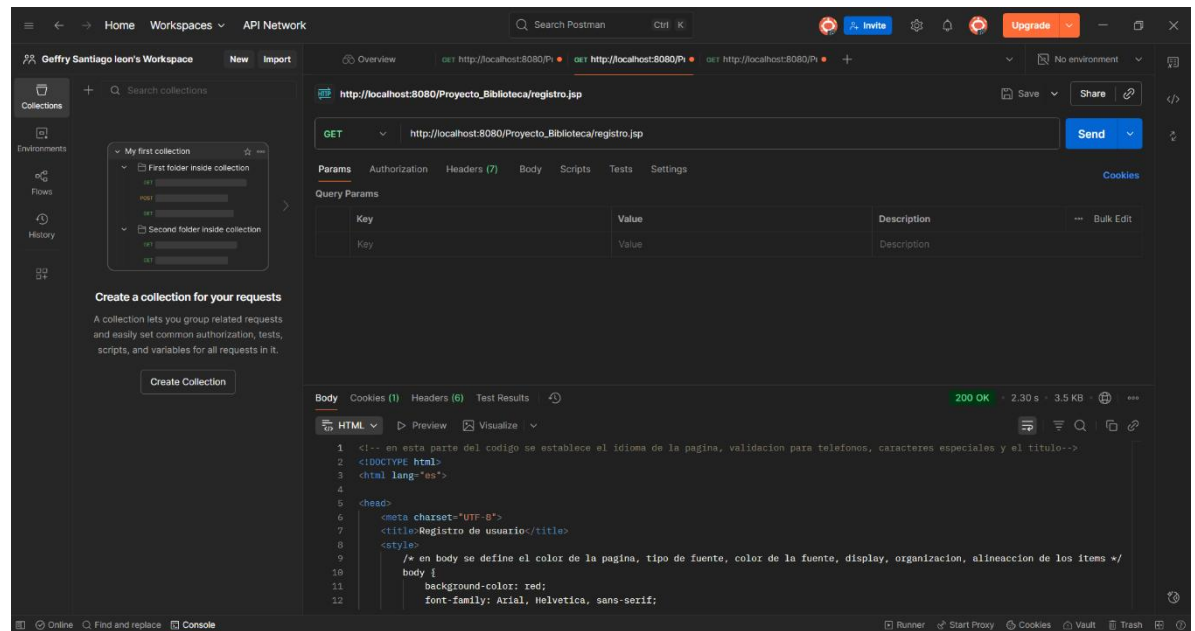
3.15 Pruebas unitarias de los módulos

Ilustración 33 Prueba de index



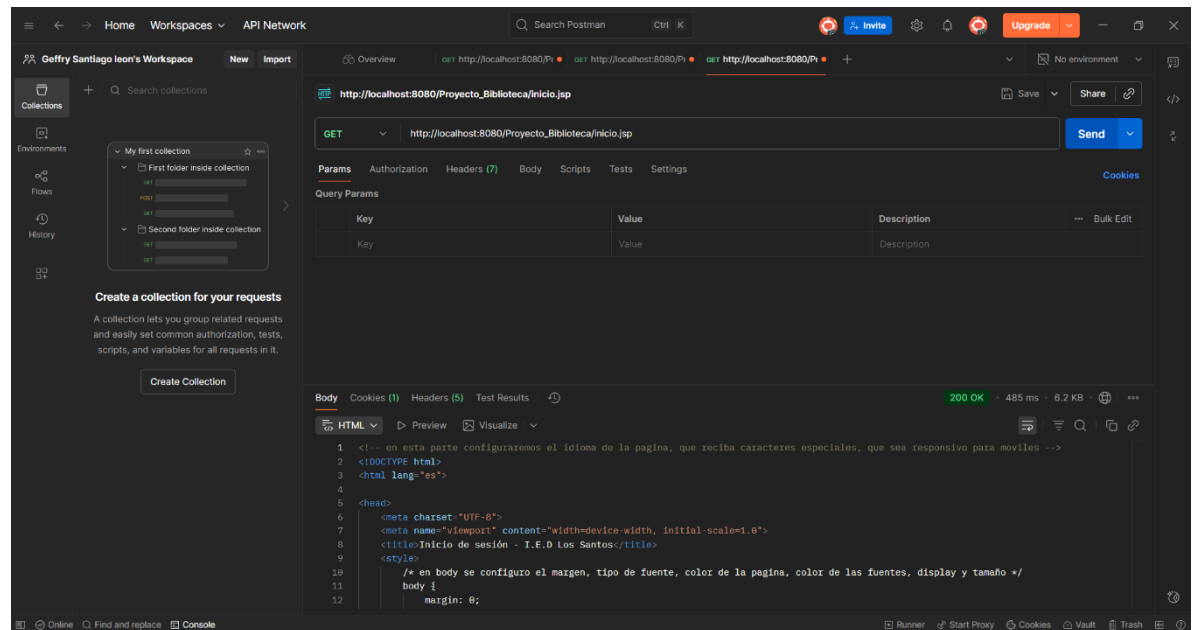
Nota: Ilustración hecha por Geffry Santiago Leon Jimenez

Ilustración 34 Prueba Registro.jsp



Nota: Ilustración hecha por Geffry Santiago Leon Jimenez

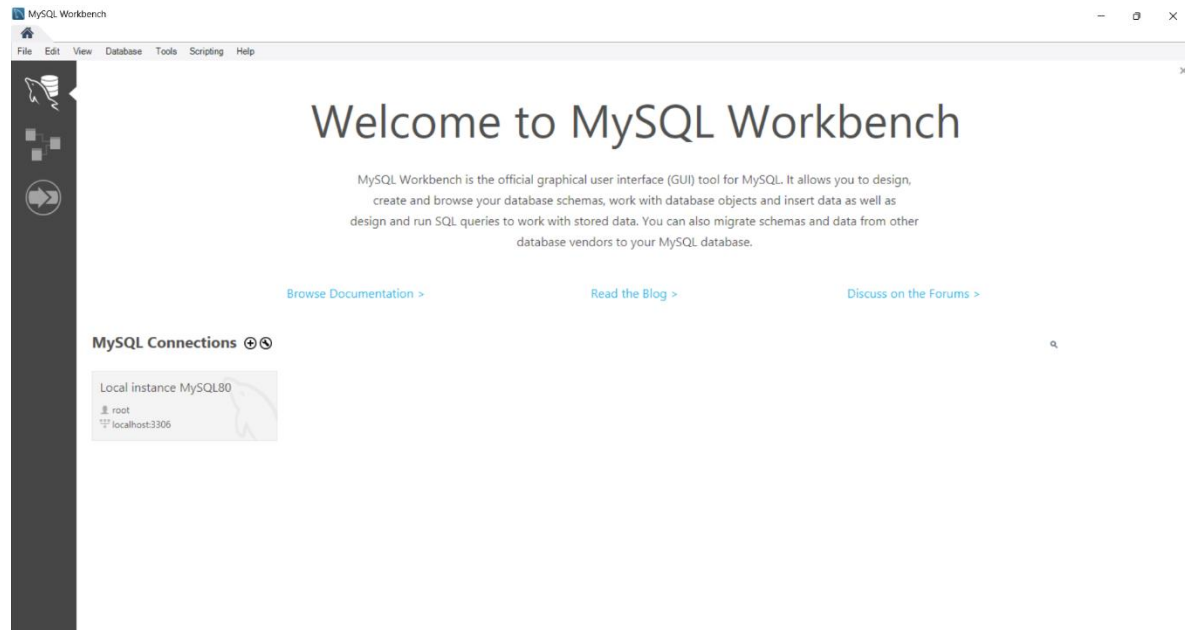
Ilustración 35 Prueba Inicio.jsp



Nota: Ilustración hecha por Geffry Santiago Leon Jimenez

3.16 Configuración de servidores y de base de datos

Ilustración 36 Primer Paso



Nota: Ilustración hecha por Geffry Santiago Leon Jimenez

Para configurar nuestra base de datos debemos darte al símbolo “+” que esta al lado de “MySQL Connections”

Ilustración 37 Segundo Paso

Setup New Connection

Connection Name: Type a name for the connection

Connection Method: Method to use to connect to the RDBMS

Parameters SSL Advanced

Hostname: Port: Name or IP address of the server host - and TCP/IP port.

Username: Name of the user to connect with.

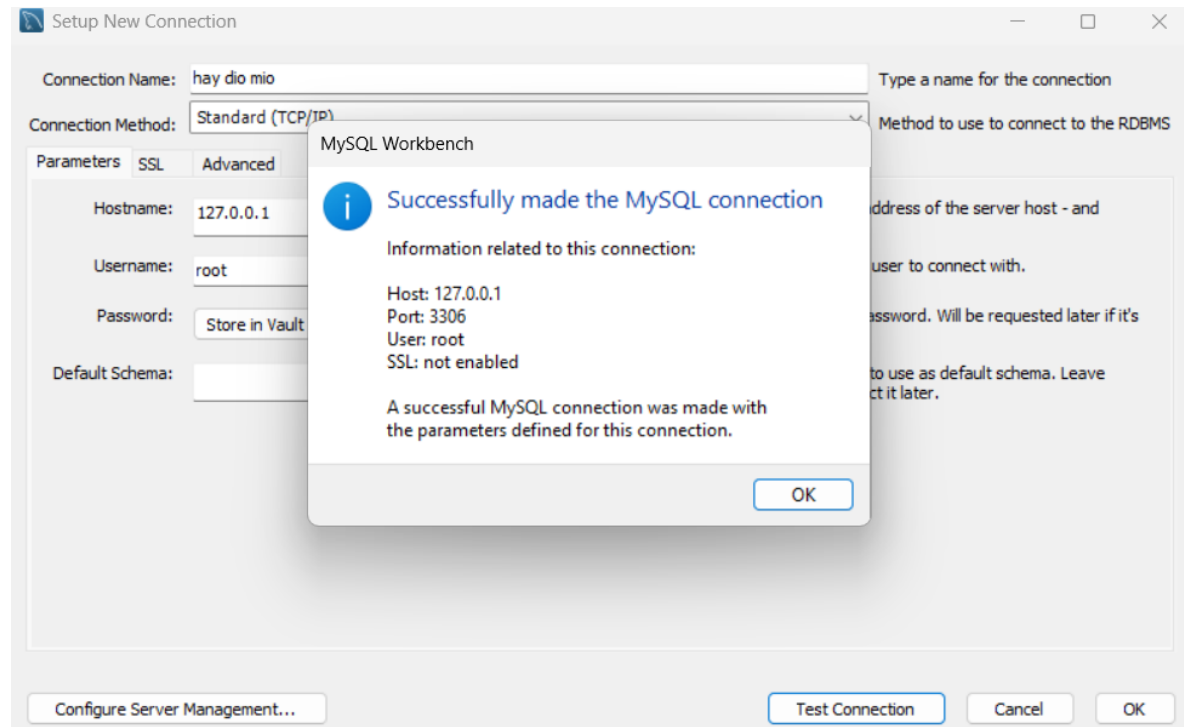
Password: The user's password. Will be requested later if it's not set.

Default Schema: The schema to use as default schema. Leave blank to select it later.

Nota: Ilustración hecha por Geffry Santiago Leon Jimenez

Luego de darle nos aparece este cuadro donde tendremos que poner esta información, el connection name es el nombre de la conexión, el hostname no se cambia y para poner la contraseña se le da a store in vault.

Ilustración 38 Tercer Paso



Nota: Ilustración hecha por Geffry Santiago Leon Jimenez

Luego de meter los datos deberemos de darle a “test connection” y nos dirá si la conexión funciona, y le damos a ok

Ilustración 39 Mi Base de Datos

```
1 • create database proyecto_software;
2 • use proyecto_software;
3   #biblioteca los santos
4 • CREATE TABLE usuario (
5   id_usuario INT AUTO_INCREMENT PRIMARY KEY,
6   nombre_apellido VARCHAR (50) NOT NULL,
7   telefono VARCHAR (15),
8   edad INT NOT NULL CHECK (edad >= 7 AND edad <=50),
9   email VARCHAR (100) UNIQUE,
10  contraseña VARCHAR (20) NOT NULL,
11  rol VARCHAR (50) NOT NULL
12 );
13 • CREATE TABLE libros (
14  id_libro INT AUTO_INCREMENT PRIMARY KEY,
15  titulo VARCHAR (50) NOT NULL,
16  autor VARCHAR (50) NOT NULL,
17  año_publicacion YEAR,
18 ✖ genero VARCHAR (50),usuariq,
19  disponibilidad BOOLEAN NOT NULL
```

Nota: Ilustración hecha por Geffry Santiago Leon Jimenez

Esta es la base de datos de mi proyecto en esta imagen se refleja las tablas de usuario y libros

Ilustración 40 Mi Base de Datos

```
20 );
21 • CREATE TABLE empleados (
22     id_empleado INT AUTO_INCREMENT PRIMARY KEY,
23     nombre_apellido VARCHAR (100) NOT NULL,
24     correo_electronico VARCHAR (50) UNIQUE NOT NULL,
25     cargo VARCHAR (50) NOT NULL,
26     telefono VARCHAR (50)
27 );
28 • CREATE TABLE prestamos (
29     id_prestamo INT AUTO_INCREMENT PRIMARY KEY,
30     id_usuario INT NOT NULL,
31     id_empleado INT NOT NULL,
32     id_libro INT NOT NULL,
33     fecha_prestamo DATE NOT NULL,
34     fecha_devolucion_limite DATE,
35     fecha_devolucion_real DATE NOT NULL,
36     estado_prestamo VARCHAR (30) NOT NULL,
37     FOREIGN KEY (id_usuario) REFERENCES usuario (id_usuario),
38     FOREIGN KEY (id_empleado) REFERENCES empleados (id_empleado),
39     FOREIGN KEY (id_libro) REFERENCES libros (id_libro)
40 );
```

Nota: Ilustración hecha por Geffry Santiago Leon Jimenez

En esta otra ilustración podemos apreciar que esta la tabla empleados y la tabla de prestamos

3.17 ambientes de desarrollo y pruebas

<https://youtu.be/JOrE25IGA3A>

4. CONCLUSIONES

Lo que eh avanzado en este proyecto me sorprende mucho ya que no imagine que llevara tantas cosas hechas, pero me eh dado de cuenta de lo que eh aprendido y lo que eh hecho, me siento orgulloso por como va mi proyecto, no va muy avanzado pero esta bien estructurado, no me quiero ni imaginar cuando este completo.

5. REFERENCIAS BIBLIOGRÁFICAS

Guia de aprendizaje 8

IMMUNE: <https://immune.institute/blog/que-es-netbeans/>

Trello:

https://trello.com/es?campaign=19322080936&adgroup=144042670505&targetid=kwd-3609071522&matchtype=e&network=g&device=c&device_model=&creative=642166749551&keyword=trello&placement=&target=&ds_eid=700000001557344&ds_e1=GOOGLE&qad_campaignid=19322080936

Canva: <https://www.canva.com/>