Logotipo

Descripción generada automáticamente

**Universidad Veracruzana**

Facultad de Informática

**“Proyecto Final Sistemas Web”**

Alumnos:

Hernández Domínguez Alan Armando

Huerta Mendoza Ameyalli

Méndez Trujillo Breayan

Pérez Sulvarán Liam Ibrahim

Catedrático:

Rojano Cáceres José Rafael

Tecnologías Computacionales, Quinto semestre

Xalapa Ver. Noviembre del 2022

Contenido

[Introducción 1](#_Toc121658918)

[Funcionamiento de la página web (Frontend) 2](#_Toc121658919)

[Index.html 2](#_Toc121658920)

[Login.html 3](#_Toc121658921)

[Register.html 5](#_Toc121658922)

[Main.html 6](#_Toc121658923)

[Funcionamiento de la página web (Backend) 9](#_Toc121658924)

[Conexión.java 9](#_Toc121658925)

[Usuario.java 10](#_Toc121658926)

[Evento.java 11](#_Toc121658927)

[App.java 13](#_Toc121658928)

[DAO.java 16](#_Toc121658929)

[DAOEvento.java 21](#_Toc121658930)

[Referencias 27](#_Toc121658932)

# Introducción

En este documento se dará una breve explicación del funcionamiento del proyecto de sistemas web, donde el Backend se desarrollo mediante el uso del lenguaje de programación Java, también Json y el framework Spark, todo esto conectado una base de datos en sql, referente al Frontend se creó en HTML, Bootstrap y JavaScript, todo esto abarcara la explicación general para entender el funcionamiento de la página.

Nuestro tema utilizado para la realización de el sistema web, fue el tema de una agenda, donde el usuario tendrá que registrarse e iniciar sesión con su cuenta, una vez dentro podrá visualizar los eventos que tiene guardado en su cuenta, también podrá crear un evento, modificar y borrar el evento que se agregó anteriormente referenciando a los que es un CRUD(Create,Read, Update and Delete).

Todo esto tiene como propósito poner en práctica los conocimientos adquiridos durante la experiencia educativa de Sistemas Web y emplearlo en un escenario real mediante este proyecto.

# Funcionamiento de la página web (Frontend)

Para nuestro proyecto se utilizaron estilos de Bootstrap y se compone de 4 archivos en formato HTML, los cuales son llamados: index.html, login.html, main.html y register.html.

Index sirve para al entrar a nuestro sitio web te redireccione hacia login, después en login puedes ir a register (en caso de no tener una cuenta) o a main (en caso de ya tener una cuenta e iniciar sesión).

## Index.html

En este archivo inicial, se creó la estructura de HTML, después se creó una función llamada toLogin dentro del cuerpo, que sirve para redireccionar la ventana del usuario hacia el login.html mediante el link del repositorio como se puede ver a continuación:

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

## Login.html

Dentro de este archivo se implementó en el script Axios, Bootstrap y Gson, para la ventana de iniciar sesión, los campos de texto, el fondo de la página y el botón iniciar, también se pide obtener los datos de la variable usuario y contraseña, si son correctos le permite al usuario pasar a main.html, si son incorrectos le marcara un error mandando un mensaje “usuario o contraseña incorrectos”.

Como ultimo dato, en caso de que usuario no tenga una cuenta, habrá un pequeño texto debajo del botón que al darle click te redireccionara a register.html donde te permitirá crear una cuenta para poder ingresar al sitio web.

Texto

Descripción generada automáticamente

Aquí se muestra login.html en la página web, con lo mencionado explicado anteriormente de la manera más breve:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

## Register.html

En este archivo utilizando los mismos estilos que en login.html, el usuario debe ingresar su nombre y la contraseña dos veces, si llena los campos indicados, se creara el usuario en la base de datos enviándolo hacia main.html, en caso de que ya exista el usuario le marcara un mensaje diciendo “usuario ya existente”, en caso de que no haya escrito bien la contraseña dos veces se le enviara un mensaje “Las contraseñas no coinciden” y si no lleno todos los campos le saldrá el mensaje “Campos obligatorios vacíos”.

Texto

Descripción generada automáticamente

Aquí se muestra como se ve en la pagina web, todo lo mencionado anteriormente implementado:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

## Main.html

En este archivo, mediante la función newEvent, le permite crear un evento tomando todos los datos que escriba dentro de los campos de texto, dando como resultado dentro de la base de datos un evento creado, solamente en caso de tener un nombre diferente a un evento existente dentro de la misma, de caso contrario marcara a un error al apretar el botón, “Titulo de evento ya existente”, también si no se llenan todos los campos “Campos obligatorios vacíos”.

Si se creo el evento aparecerá correctamente en la pagina web, con todos los datos que el usuario proporciono en los campos de texto, también es posible borrar y modificar el evento si así el usuario lo desea.

Texto

Descripción generada automáticamente

Esto es lo que aparece una vez el usuario inicia sesión o se registra en la página, todo lo explicado en el código se observa aquí, todos los campos de texto que el usuario debe llenar para poder crear un evento.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

En este botón se creará el evento, donde aparecerá en la parte derecha blanca, como si fuera un historial, si se agregan más, aparecerán más, también puedes modificar o borrar el evento que hayas creado llenando los campos de texto adecuadamente para que no marque algún error.

Interfaz de usuario gráfica

Descripción generada automáticamente

# Funcionamiento de la página web (Backend)

Para el backend se crearon las clases Usuario y Evento, para los datos, la clase conexión, para crear la conexión hacia la base de datos en sql, el App para peticiones y respuestas, DAO y DAOEvento que permiten crear tuplas a las tablas de la base de datos, manipular información dentro de las mismas con los datos que el usuario ingrese.

## Conexión.java

En la clase Conexión, como su nombre lo indica sirve para conectar la base de datos SQL, con los drivers correspondientes para el lenguaje de programación Java, donde se agrega la url de la base de datos, el usuario y la contraseña para acceder a dicha base de datos, con el método getConnection() del DriveManager se obtiene la conexión de los datos que se guarde y el try/catch para excepciones.

Texto

Descripción generada automáticamente

## Usuario.java

También se creó la clase “Usuario”, donde posee solo 3 atributos: id, nombre y contraseña,su método constructor, junto con los setters y getters, y finalmente abajo se crea el método toSring() que retorna todos los datos de la clase usuario, ya que es necesaria la creación de cuentas e inicio de las mismas que se van a utilizar en la página web.

Texto

Descripción generada automáticamente

## Evento.java

De igual manera, se creó la clase “Evento”, donde guardara datos como: id, titulo, fecha, hora de inicio, hora de fin, lugar y descripción, con su método constructor, sus setters y getters y finalmente su método toString() que retorna todos los datos de la clase Evento,es necesaria ya que se guardarán estos datos con la cuenta del usuario que los haya escrito.

Texto

Descripción generada automáticamente

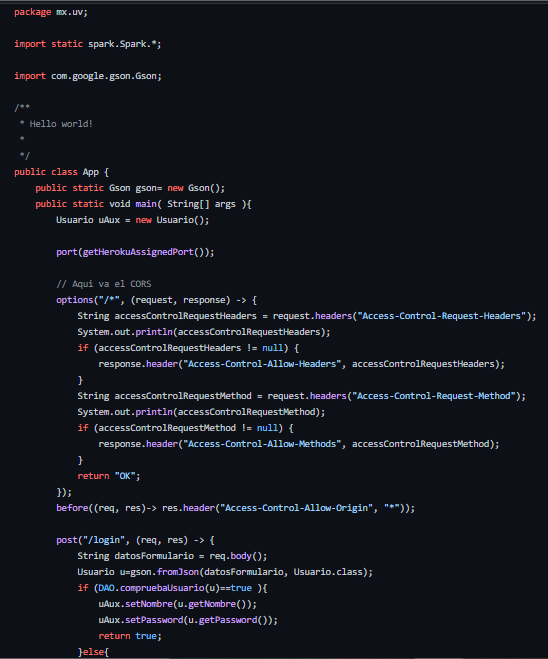
Texto

Descripción generada automáticamente

## App.java

Esta clase es responsable de responder las solicitudes del FrontEnd, el Backend le devuelve la información en Gson, primero se crea el CORS( Access-Contro-Request-Headers).

Se crea el Post a la dirección /login el cual se encarga de llamar al método compruebaUsuario de la clase DAO en el condicional if , tomando los datos del request y convirtiéndolos en un objeto de la clase Usuario, utiliza el moetodo getId() para mandarlo al método compruebaUsuario, si existe le dejara entrar, si no no podrá entrar el usuario

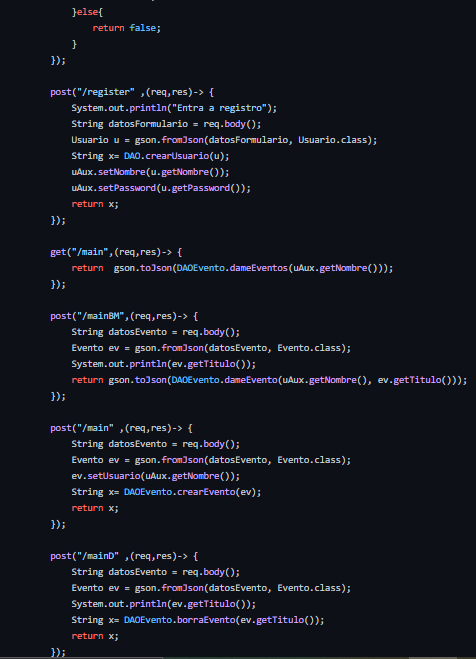


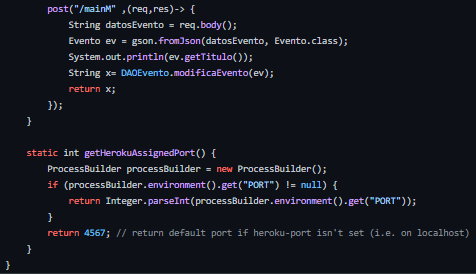
El post a la dirección /register llama al método crearUsuario de la clase DAO, toma los datos del request y los convierte en un objeto de la clase Usuario.

El post a la dirección /mainBM permite regresar el evento por el método DameEvento de la clase DAO.

El post a la dirección /main permite retornar el método CrearEvento de la clase DAO por la variable String x, parecido a este esta el post a la dirección /mainD y /mainM

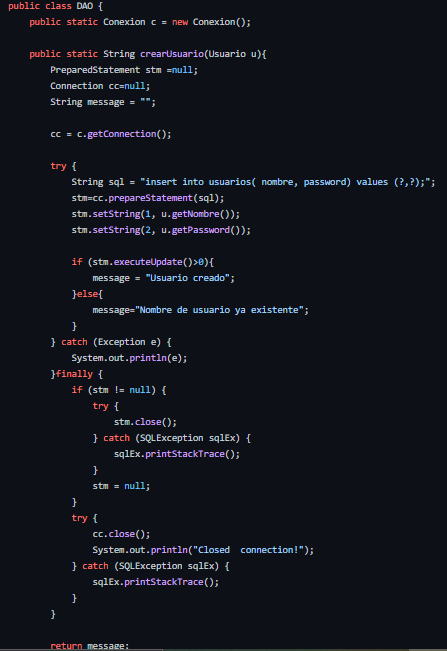
El get /main retorna el método dameEventos de la case DAO.



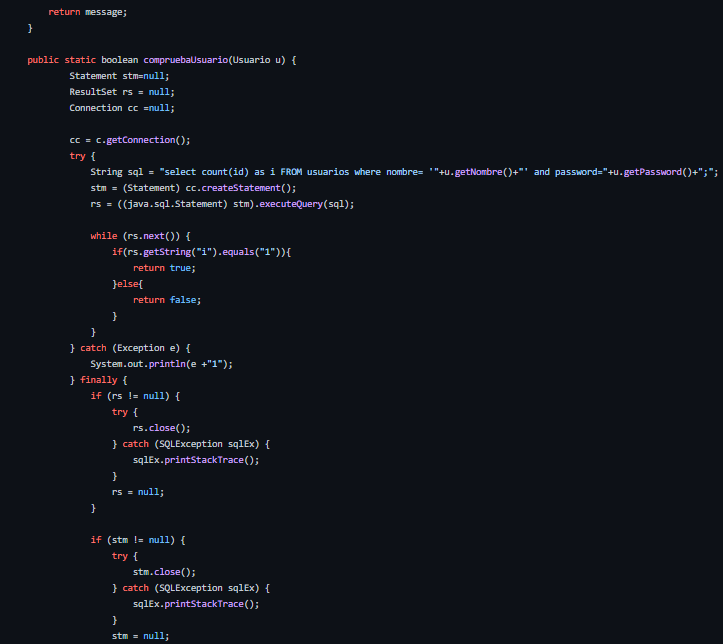


## DAO.java

En la clase DAO( Data Access Object), que sirve para almacenar los datos en la base de datos, se usa el query “Inser into” para crear una tupla en la tabla de “usuarios” recibiendo un objeto de la clase Usuario, con el Statement y la conexión devuelta con el método getConnection() de la clase Conexión y el set para los values mediante el método getNombre() del objeto que recibe el método y después el método getPassword(), con la condicional if si el valor es mayor a cero significa que se ejecuto correctamente, si no , no se realizo el update y manda un mensaje “Nombre de usuario ya existente” o marcara un error.



En este método compruebaUsuario, se hace la consulta “select count as from”, se encarga de obtener el id mediante el nombre y la contraseña, el id se compara con la posición.



Para eliminar el usuario se creó el método borrarUsuario, donde recibe el id como referencia para actualizar la base de datos, usando el query Delete From, el id que se recibe como parámetro para la instrucción hacia la base de datos, con el Statment para efectuar Update, utilizando el query definido en la variable sql y la condicional if se ejecuta la actualización con el método executeUpdate() el Statment, si es correcto se devolverá un numero mayor a cero y el mensaje “El usuario se eliminó”, en caso de no ser correcto “El usuario no se eliminó”, dentro del try/catch y finally.

Texto

Descripción generada automáticamente

Para modificar el usuario, se creó el método modificarUsuario se almacena el query en el String sql, Update eventos.db donde se concatena el nombre y password e id del objeto recibido como parámetro para la instrucción hacia la base de con el Statment para efectuar Update, utilizando el query definido en la variable sql y la condicional if se ejecuta la actualización con el método executeUpdate() el Statment, si es correcto se devolverá un número mayor a cero y el mensaje “El usuario se modificó”, en caso de no ser correcto “El usuario no se modificó”, dentro del try/catch y finally.

Texto

Descripción generada automáticamente

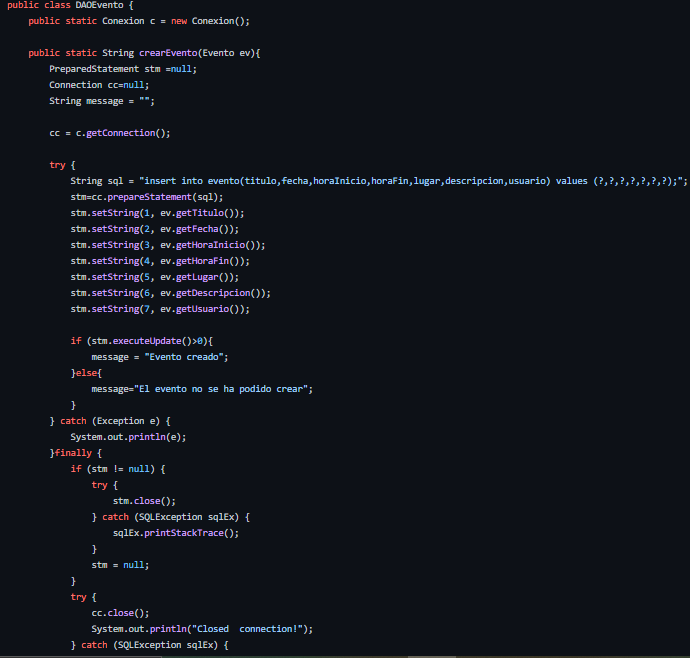
El método dameUsuarios, extrae todas las tuplas de la tabla usuarios de la base de datos, los agrega aun ArrayList y los retonar, se define el query en un String llamado sql,Select \* from usuarios, se crea el Statement con la conexión para luego ejecutar el query donde esta la variable tipo ResultSet que guarda el contenido devuelto de la consulta, con un while se itera el ResultSet para acceder a las tuplas y se crea el objeto de tipo Usuario, por cada una de las tuplas, se almacena el objeto en el array, donde es devuelto al final de método, se cierra ResultSet, el Statment y la conexión.

Texto

Descripción generada automáticamente

## DAOEvento.java

En la clase DAOEvento( Data Access Object), que sirve para almacenar los datos en la base de datos, se usa el query “Inser into” para crear una tupla en la tabla de “evento” recibiendo un objeto de la clase Evento, con el Statement y la conexión devuelta con el método getConnection() de la clase Conexión y el set para los values mediante el método getTtitulo() del objeto que recibe el método y después el método getFecha(), getHoraIncio, getHoraFin, getLugar, getDescripcion y getUsuario con la condicional if si el valor es mayor a cero significa que se ejecutó correctamente, si no , no se realizó el update y manda un mensaje “Evento creado” o marcara un error.



En este método compruebaEvento, se hace la consulta “select count as from”, se encarga de obtener el id mediante el titulo el id se compara con la posición.

# Texto Descripción generada automáticamente

Para eliminar el evento se creó el método borrarEvento, donde recibe el titulo como referencia para actualizar la base de datos, usando el query Delete From, el titulo que se recibe como parámetro para la instrucción hacia la base de datos, con el Statment para efectuar Update, utilizando el query definido en la variable sql y la condicional if se ejecuta la actualización con el método executeUpdate() el Statment, si es correcto se devolverá un número mayor a cero y el mensaje “El evento se eliminó”, en caso de no ser correcto “El evento no se eliminó”, dentro del try/catch y finally.

Texto

Descripción generada automáticamente

Para modificar el evento, se creó el método modificarEvento se almacena el query en el String sql, Update eventos.db.evento donde se concatena el título, fecha, horainicio, horafin, lugar y descripción donde es el título del objeto recibido como parámetro para la instrucción hacia la base de con el Statment para efectuar Update, utilizando el query definido en la variable sql y la condicional if se ejecuta la actualización con el método executeUpdate() el Statment, si es correcto se devolverá un número mayor a cero y el mensaje “El evento se modificó”, en caso de no ser correcto “El evento no se modificó”, dentro del try/catch y finally.

Texto

Descripción generada automáticamente

El método dameEventos, extrae todas las tuplas de la tabla eventos del usuario de la base de datos, los agrega aun ArrayList y los retonar, se define el query en un String llamado sql,Select \* from usuarios, se crea el Statement con la conexión para luego ejecutar el query donde esta la variable tipo ResultSet que guarda el contenido devuelto de la consulta, con un while se itera el ResultSet para acceder a las tuplas y se crea el objeto de tipo Evento, por cada una de las tuplas, se almacena el objeto en el array, donde es devuelto al final de método, se cierra ResultSet, el Statment y la conexión.

Texto

Descripción generada automáticamente

El método dameEvento, extrae todas las tuplas de la tabla eventos del usuario y título de la base de datos, se define el query en un String llamado sql,Select \* from evento, se crea el Statement con la conexión para luego ejecutar el query donde está la variable tipo ResultSet que guarda el contenido devuelto de la consulta, con un while se itera el ResultSet para acceder a las tuplas, se cierra ResultSet, el Statment y la conexión.

Texto

Descripción generada automáticamente

# Referencias

Otto, M. J. T. (s. f.). Bootstrap. <https://getbootstrap.com/>

W3Schools Free Online Web Tutorials. (s. f.). <https://www.w3schools.com/>

MySQL. (s. f.). <https://www.mysql.com/>

Spark Framework: An expressive web framework for Kotlin and Java. (s. f.). <https://sparkjava.com/>

Dorian Desings. (2020, 11 diciembre). *CURSO de HTML5 desde CERO 2021 - #68 - Atributos de accesibilidad*. YouTube. <https://www.youtube.com/watch?v=-4y3vb0IHgw>

*HTML: Una buena base para la accesibilidad - Aprende sobre desarrollo web | MDN*. (2022, 5 diciembre). <https://developer.mozilla.org/es/docs/Learn/Accessibility/HTML>