

PROYECTO DE DESARROLLO DE APLICACIONES WEB

WILSON SOTO
(docente)

EDUWIN ROJAS SUÁREZ
EVELIN MIRANDA BERNAL
JAVIER DARIO URREGO BLANCO
(estudiantes)

UNIVERSIDAD POLITÉCNICO GRANCOLOMBIANO
TECNOLOGÍA EN DESARROLLO DE SOFTWARE
DESARROLLO WEB

SEPTIEMBRE 06, 2022

1. Identificación de requisitos representados en historias de usuario.

Historia de usuario 1 – Crear Página de autenticación

Como: Cliente

Quiero: ingresar mi usuario y contraseña en la página inicio

Para: poder tener acceso a la información almacenada en la aplicación Web.

Criterios de aceptación

- Debe tener la opción de ingresar el usuario.
- Debe tener la opción de ingresar la contraseña.
- Debe tener la opción de iniciar sesión.
- Debe tener la opción de recuperar la contraseña.
- Solo se permitirá acceder a los usuarios registrados.
- Validar que la contraseña sea correcta.
- Se tendrá la opción de recuperación de la contraseña.
- Si la contraseña no es correcta aparece un mensaje pop-up con el texto “la contraseña es válida”.
- Si el usuario no está registrado aparece un mensaje pop-up con el texto “Usuario no encontrado”.
- Debe tener un botón llamado “Ingresar”.

Tareas

- Crear un proyecto usando angular llamado “ProyectoRutasPoligran”
- En el proyecto “ProyectoRutasPoligran” crear una página llamada “inicio”
- En la página de “inicio” crear un formulario llamado “formularioLogin” con dos componentes input para ingresar el correo y a contraseña del usuario, el primer input llevará como placeholder “correo”, el segundo input llevará como placeholder “contraseña”.
- En el archivo “inicio.page.ts” se debe crear el método ingresar() para invocar el servicio correspondiente.
- El formulario tendrá un componente button al que se le asignará un evento click() con el nombre de “ingresar()”, dicho botón se llamará “Ingresar”
- En la parte inferior del formulario debe ir un button llamado “Ingresar” al que se le asignará un evento click() con el nombre de “recordar()”, dicho botón se llamará “Olvidé mi contraseña”.
- En el método ingresar() del archivo “inicio.page.ts” se realizará la validación donde se compare los datos ingresados por el usuario con los datos que el servicio trae de la base de datos, en caso que las credenciales no

correspondan se deberá mostrar en pantalla un pop-up informando al cliente de la situación.

Historia de usuario 2 – Crear página Registro de Buses.

Como: Cliente

Quiero: ver una página llamada “Registro de Bus”

Para: gestionar los procesos realizados con la información de los buses.

Criterios de aceptación

- Debe tener la opción de ingresar número del bus.
- Solo se aceptan números enteros para el número bus.
- Debe tener la opción de ingresar la placa del bus.
- Debe tener la opción de ingresar número de sillas del bus.
- Solo se aceptan números enteros para el número de sillas.
- Debe tener un botón para el registro del bus.
- Debe tener un botón para editar el registro del bus.
- Debe tener un botón para consultar el registro del bus.
- Debe tener un botón para eliminar el registro del bus.

Tareas

- Crear una pagina llamada “buses”.
- Crear un formulario llamado “formularioBuses”.
- En el formulario insertar 3 componentes inputs.
- Los componentes inputs tendrán el formControlName idNumeroBus, idPlacaBus y idNumeroSillas respectivamente.
- Se debe insertar 4 botones en el formulario, de nombres Registrar, Editar, Consultar y Eliminar.
- Los componentes button tendrán el identificador btnRegistro, btneditar, btnConsultar y btnEliminar.

Historia de usuario 3 – Crear servicios para el Registro de Buses.

Como: Cliente

Quiero: tener servicios en la capa de negocio

Para poder: ingresar, consultar, editar y eliminar la información de los buses.

Criterios de aceptación

- Debe permitir consultar un bus(es) registrados en la base de datos.

- Debe permitir agregar un bus(es) a la base de datos.
- Debe permitir editar un bus que ya se encuentre registrado en la base de datos
- Debe permitir eliminar de la base de datos un bus que ya se encuentre registrado.
- Las peticiones deben ejecutarse en Postman.

Tareas

- Crear un proyecto en Sprint Boot usando la capa de negocio REST llamado RutaPoli, 3 package con los siguientes nombres:
 - co.edu.poli.rutaspoli.controller
 - co.edu.poli.rutaspoli.model
 - co.edu.poli.rutaspoli.repository
- Crear en MySQL un nuevo schema de nombre rutasPoli.
- Crear método GET para un bus.
- Crear método GET para consultar buses.
- Crear método POST para insertar buses.
- Crear método PUT para actualizar buses.
- Crear método DELETE para eliminar buses.

Historia de usuario 4 - Crear página Registro de Conductores.

Como: Cliente

Quiero: ver una página llamada "Registro de Conductor"

Para: gestionar los procesos realizados con la información de los conductores.

Criterios de aceptación

- Debe tener la opción de ingresar nombre del conductor donde únicamente se aceptan caracteres pertenecientes al alfabeto excluyendo caracteres especiales.
- Debe tener la opción de ingresar la identificación del conductor donde únicamente se puedan ingresar caracteres numéricos.
- Debe tener la opción de ingresar la fecha de ingreso a la universidad del conductor.
- Debe tener un botón para guardar el registro del conductor.
- Debe tener un botón para editar el registro del conductor.
- Debe tener un botón para consultar el registro del conductor.
- Debe tener un botón para eliminar el registro del conductor.

Tareas

- Crear una página llamada “conductores”.
- Crear un formulario llamado “formularioConductor”.
- En el formulario insertar 3 componentes inputs.
- Los componentes inputs tendrán el formControlName nombreConductor, idConductor y fechaIngreso respectivamente.
- Se debe insertar 4 botones en el formulario, de nombres Registrar, Editar, Consultar y Eliminar.
- Los componentes button tendrán el identificador btnRegistro, btnEditar, btnConsultar y btnEliminar.

Historia de usuario 5 – Crear servicios para el Registro de Conductor.

Como: Cliente

Quiero: tener servicios en la capa de negocio

Para poder: ingresar, consultar, editar y eliminar la información de los conductores.

Criterios de aceptación

- Debe permitir consultar un conductor(es) registrados en la base de datos.
- Debe permitir agregar un conductor(es) a la base de datos.
- Debe permitir editar un conductor que ya se encuentre registrado en la base de datos
- Debe permitir eliminar de la base de datos un conductor que ya se encuentre registrado.
- Las peticiones deben ejecutarse en Postman.

Tareas

- Crear método GET para un conductor.
- Crear método GET para consultar conductores.
- Crear método POST para insertar conductores.
- Crear método PUT para actualizar conductores.
- Crear método DELETE para eliminar conductores.

Historia de usuario 6 – Crear página Registro de Rutas

Como: Cliente

Quiero: ver una página llamada “Rutas”

Para: poder ingresar y consultar la información de la ruta realizada por los buses del Politécnico Grancolombiano.

Criterios de aceptación.

- Debe tener una lista desplegable donde se pueda seleccionar un bus del Politécnico Gracolombiano.
- Debe tener una lista desplegable donde se pueda seleccionar un conductor del Politécnico Grancolombiano
- Debe permitir el ingreso del número de sillas vacías del bus.
- Debe tener una tabla con cuatro columnas “Número de bus”, “Conductor”, “Sillas vacías” y “Fecha y hora”.
- Debe tener un botón para guardar el registro de la ruta.
- Debe tener un botón para consultar el registro de la ruta.

Tareas

- Crear una página llamada “rutas”.
- Crear un formulario llamado “formularioRuta”.
- El formulario deberá tener un componente input tendrá el formControlName sillasVacias.
- El formulario deberá tener dos listas desplegables con formControlName numeroBus y nombreConductor respectivamente.
- Se debe insertar 2 botones en el formulario, de nombres Registrar y Consultar.
- Los componentes button tendrán el identificador btnRegistro y btnConsultar.

Historia de usuario 7 – Crear servicios para el Registro de Rutas.

Como: Cliente

Quiero: tener servicios en la capa de negocio

Para poder: ingresar y consultar la información de las rutas de buses que salen desde el City campus al campus del Politécnico GranColombiano.

Criterios de aceptación

- Debe permitir consultar un bus(es) registrados en la base de datos.
- Debe permitir agregar un bus(es) a la base de datos.

- Debe permitir editar un bus que ya se encuentre registrado en la base de datos
- Debe permitir eliminar de la base de datos un bus que ya se encuentre registrado.
- Las peticiones deben ejecutarse en Postman.

Tareas

- Crear método GET para una ruta.
- Crear método GET para consultar rutas.
- Crear método POST para insertar rutas.

2. Definición de mockups.

Login



Correo Electrónico

Contraseña

INICIAR SESIÓN

Olvidaste tu contraseña?

Registro de conductores



CONDUCTORES

Nombre

Escriba aquí la palabra

Identificación

Escriba aquí la palabra

Fecha de ingreso

DIA MES AÑOS

REGISTRAR CONSULTAR

EDITAR ELIMINAR

Registro de bus

■

BUSES

Numero del bus

Escriba aquí la palabra

PLACA

Escriba aquí la palabra

Numero de sillas

REGISTRAR

CONSULTAR

EDITAR

ELIMINAR

Registro de rutas

REGISTROS

CERRAR SECCION

BUS

▼

SILLAS VACIAS

NOMBRE DEL CONDUCTOR

▼

REPORTES

ENVIAR REPORTE

NUMERO DE BUS

CONDUCTOR

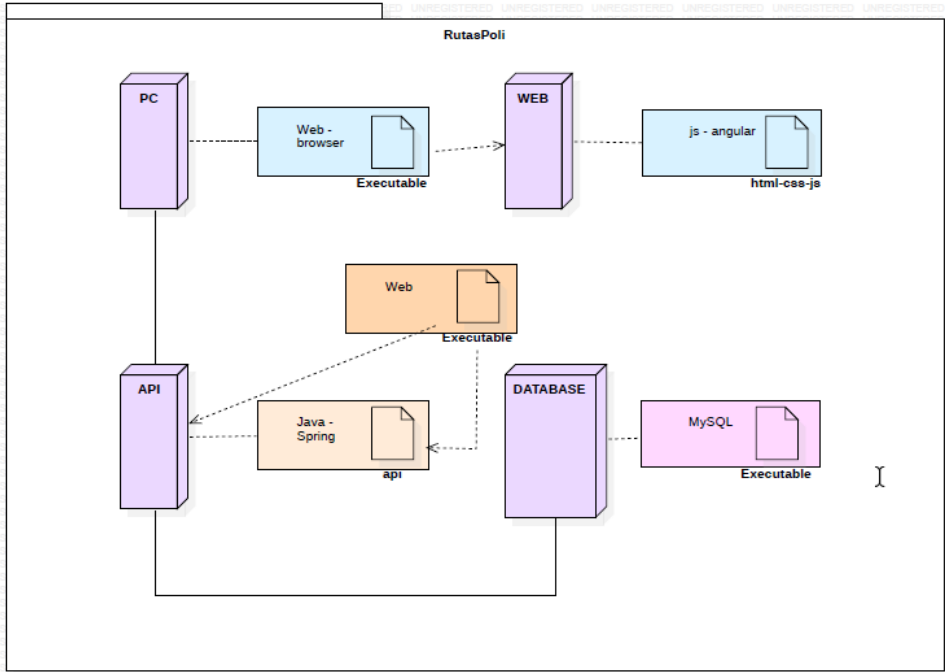
SILLAS VACIAS

FECHA Y HORA

REGISTRAR

CONSULTAR

	UNREGISTERED	UNREGISTERED	UNREGISTERED	UNREGISTERED	UNREGISTERED	UNREGISTERED



Presentación: Angular. es un framework opensource desarrollado por Google

Negocio: Spring Boot, es una tecnología que nos permite crear aplicaciones autocontenidas, con esto nos podemos olvidar de la arquitectura y enfocarnos únicamente en desarrollo, delegando a Spring Boot labores como configuración de dependencias, desplegar nuestro servicio o aplicación a un servidor de aplicaciones y enfocarnos únicamente en crear nuestro código.

Negocio: Spring Boot, es una tecnología que nos permite crear aplicaciones autocontenidas, con esto nos podemos olvidar de la arquitectura y enfocarnos únicamente en desarrollo, delegando a Spring Boot labores como configuración de dependencias, desplegar nuestro servicio o aplicación a un servidor de aplicaciones y enfocarnos únicamente en crear nuestro código.

Persistencia: MySQL, es el sistema de gestión de bases de datos relacional más extendido en la actualidad al estar basada en código abierto, trabaja con bases de datos relacionales, es decir, utiliza tablas múltiples que se interconectan entre sí para almacenar la información y organizarla correctamente. Además, basa su funcionamiento en un modelo cliente y servidor. Es decir, clientes y servidores se comunican entre sí de manera diferenciada para un mejor rendimiento. Posee la característica de no procesar las tablas directamente, sino que a través de procedimientos almacenados es posible incrementar la eficacia de nuestra implementación, también, permite automatizar ciertas tareas dentro de nuestra base de datos.

Referencias

- Qué es MySQL: Características y ventajas.* (2019, septiembre 24). Openwebinars.net.
<https://openwebinars.net/blog/que-es-mysql/>
- Gonzalez, M. V. (2021, marzo 17). *¿Qué es Spring Boot?* Codmind. <https://blog.codmind.com/que-es-spring-boot/>
- Devs, Q. (2019, septiembre 16). *¿Qué es Angular y para qué sirve?* Quality Devs.
<https://www.qualitydevs.com/2019/09/16/que-es-angular-y-para-que-sirve/>