1.Abir el programa Oracle SQL

2.una ves entrando a tu conexión guardada de ORCL\_DB (muestra 8SEMESTRE pero se cambia al que tenemos escrito) y contraseña udea2025;

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

La contraseña la que creamos en el ORCL\_DB: udea2025

Una ves conectado a la base de datos vamos a crear una base de datos llamada CALZADO lo cual va a tener los siguientes datos

CREATE TABLE ALUMNOS(

ID\_ALUMNOS NUMBER,

NOMBRE NVARCHAR2(100), //El nombre de la tipo de la calzado, NVARCHAR2(100) indica que es un texto (cadena de caracteres) con un máximo de 100 caracteres

APELLIDO NVARCHAR2(100),

APELLIDOS NVARCHAR2(100),

EDAD NUMBER),

GENERO NVARCHAR2(100),

GRADO , //El nombre de la precio del calzado, NUMBER indica que es numérico.

CONSTRAINT PK\_ALUM PRIMARY KEY (ID\_ALUMNOS)); //Esta línea define una restricción. PRIMARY KEY (ID\_ALUMNOS) indica que la columna ID\_ALUMNOS es la clave principal, asegurando que cada valor en esta columna sea único y que no se pueden tener dos computadoras con el mismo ID\_ALUMNOS

Con la información ingresada para meter los datos correspondientes que se va a meter:

INSERT INTO ALUMNOS VALUES(1, 'Ricardo Uriel' , 'Coyotzi Rodriguez' , 24, 'Hombre', 'Octavo');

INSERT INTO ALUMNOS VALUES(2, 'Luis' , 'Valerdi Asarrie ' , 22, 'Hombre', 'Octavo');

INSERT INTO ALUMNOS VALUES(3, 'Luis Enrique' , 'Lopez Mendoza' , 22, 'Hombre', 'Octavo');

INSERT INTO ALUMNOS VALUES(4, 'Lorena' , 'Cervantez' , 23, 'Mujer', 'Octavo');

INSERT INTO ALUMNOS VALUES(6, 'DIego' , 'Avila' , 25, 'Chique', 'Octavo');

SELECT \* FROM ALUMNOS; //Esta línea es una consulta SQL SELECT \* significa “seleccionar todas las columnas” FROM ALUMNOS indica que se selecciona de la tabla ALUMNOS. Esta columna mostrara todos los datos que hay en la tabla ALUMNOS en ese momento

commit; ; //Esta línea guarda permanentemente los cambios realizados en la base de datos, Antes de COMMIT, los cambios son solo transacciones y se puede deshacer.

Imagen de la base de datos en Oracle

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Una ves terminado y creado la tabla vamos a empezar a darle forma desde el programa Spring Tool

Creación del proyecto SPRING STOOL SUITE

Creamos un nuevo proyecto de Sprint. Es el menú principal, selección file > new > Spring Starter Project. Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

Pero antes de eso lo creamos desde File > File > Spring Starter Project

Configurar el proyecto: Se abrirá una ventana donde deberás configurar los detalles dl proyecto. Los campos más importantes

proyecto

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.(Crud\_ALUMNOS). Este nombre se usará

para la carpeta del proyecto

CrudAlumnos

Group y Package: El paquete base para

Com.mx.Alumnos

CrudAlumnos

el codigo (com.mx.Alumnos). Es una

buena practica usar un nombre de dominio

Com.mx.Alumnos

inverso para evitar conflictos de nombre

Descripcion: Una breve descripcion del Proyecto

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.Seleccionar dependencias

Spring Web: Esta dependencia es esencial

Incluye los componentes necesario para

manejar solicitudes HTTP.

Spring Data JPA: Esta depedencia facilita

la interaccion con la base de datos.

Spring Boot DevTools: Herramienta para

desarrolladores (recarga automatica)

Oracle Driver: Controlador JDBC para

Oracle Database

HyperSQL Database: Base de datos ligera,

en memoria, ideal para desarrollo

y pruebas

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.Una ves ya seleccionado la dependencias solo finalizar

CrudAlumnos

Buscar la carpeta src/main/resources

Encuentra application. properties

Para abrir el archivo application. properties, simplemente has doble clic sobre el

application. Properties: Configura la aplicación Spring Boot. ES donde le dice a Spring Boot cosas como donde esta tu base de datos, que puertos usar, y otras opciones de configuración. Es como un manual de instrucciones para la aplicación

//Este codigo configura una aplicación Spring Boot para conectarse a una base de datos Oracle, define el puerto que usara y activar un registro detallado de las consultas SQL que realiza

Entramos en aplication.properties y ponemos el siguiente código:

Texto

El contenido generado por IA puede ser incorrecto.

CrudAlumnos

8009

spring.application.name=CrudALUMNOS //Le da un nombre a la aplicación: “CrudALUMNOS”. Es como poner una etiqueta

server.port=8009 //Indica que la aplicación escuchara las conexiones en el puerto 8009. Es el numero de puerta por donde entraran las conexiones a mi aplicación

spring.datasource.url=jdbc:oracle:thin:@localhost:1521/xe //Esta es la direccion de mi base de datos. Oracle esta en mi laptop (localhost), en el puerto 1521, y que la instancia se llama xe

spring.datasource.username=ORCL\_DB //Aquí va el usuario de mi base de datos: ORCL\_DB

spring.datasource.password=Udea2025//Aquí es la contraseña udea2025

spring.datasource.driver-class-name=oracle.jdbc.OracleDriver

spring.jpa.database=ORACLE // le confirmo que estoy usando una base de datos Oracle. Esto es importante para Spring Boot configure correctamente JPA

spring.jpa.database-platform=org.hibernate.dialect.OracleDialect

spring.jpa.show-sql=true

spring.jpa.properties.hibernate.format.sql=true

logging.level.org.hibernate.sql=debug

logging.level.org.hibernate.type.descriptor.sql.BasicBlinder=TRACE

Navegacion en la carpeta scr/main/java



Com.mx.Alumnos

Crea el paquete: Haz clic derecho en la carpeta scr/main/java y selecciona new > Package. Introduce el nombre del paquete: com.mx.Alumnos.dominio y presiona “finish ”

Texto

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

com.mx.Alumnos

Crear la clase: Haz clic derecho a la carpeta recién creada com.mx.Alumnos.dominio y selecciona new > “Class”. Introduce el nombre de la clase Calzado y presiona “Finish”

Texto

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Calzado

CrudAlumnos/src/main/java

Com.mx.Alumnos.dominio

package com.mx.Alumnos.dominio;

*//Package: Los paquetes organizan las clases y archivos relacionados en una jerarquía de directorios*

*//com.mx.Alumnos.dominio: Este es el nombre del paquete. En este caso, parece que la clase se encuentra dentro del proyecto llamado “Calzado”, con un subpaquete “dominio*

// Importación de anotaciones para el mapeo JPA (Jakarta Persistence API)

import jakarta.persistence.Entity;

import jakarta.persistence.Id;

import jakarta.persistence.Table;

@Table(name = "CALZADO")// Anotación que indica que esta clase representa una tabla en la base de datos

@Entity// Anotación que marca esta clase como una entidad JPA

public class Calzado {

@Id// Anotación que indica que este atributo es la clave primaria

private int idAlumnos; // ID único para cada calzado

private String nombre; //

private String apellido; //

private Number String edad;

private String genero;

private String grado;

public Alumnos() { // Constructor vacío requerido por JPA

}

// Constructor con todos los parámetros, útil para crear objetos rápidamente

public Alumnos(int idAlumnos, String nombre, String apellidos, double edad, String genero, String grado) {

this.idAlumnos = idAlumnos;

this.nombre = nombre;

this.apellidos = apellidos;

this.edad = edad;

this.genero = genero;

this.grado = grado;

}

@Override // Método toString para representar el objeto como una cadena (útil para depuración)

public String toString() {

return "Alumnos [idAlumnos=" + idAlumnos + ", nombre=" + nombre + ", apellidos=" + apellidos + ", edad=" + edad

+ ", genero=" + genero + ", grado=" + grado + "]";

}

public int getIdAlumnos() {

return idAlumnos;

}

public void setIdAlumnos(int idAlumnos) {

this.idAlumnos = idAlumnos;

}

public String getNombre() {

return nombre;

}

public void setNombre(String nombre) {

this.nombre = nombre;

}

public String getApellidos() {

return apellidos;

}

public void setApellidos(String apellidos) {

this.apellidos = apellidos;

}

public double getEdad() {

return edad;

}

public void setEdad(double edad) {

this.edad = edad;

}

public String getGenero() {

return genero;

}

public void setGenero(String genero) {

this.genero = genero;

}

public String getGrado() {

return grado;

}

public void setGrado(String grado) {

this.grado = grado;

}

}

Texto

El contenido generado por IA puede ser incorrecto.

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Navega a la carpeta src/main/java



Com.mx.Alumnos

Com.mx.Alumnos

Crea el paquete: Haz clic derecho en la carpeta *scr/main/java y selecciona new > Package. Introduce el nombre del paquete: com.mx.Alumnos.dao y presiona “finish”*

AlumnosDao.java

Crear la clase: Haz clic derecho a la carpeta recién creada com.mx.Alumnos.dao y selecciona new > “Class”. Introduce el nombre de la clase AlumnosDao y presiona “Finish”

package com.mx.Alumnos.dao; */7Define la ubicación del archivo dentro del proyecto*

import org.springframework.data.repository.CrudRepository; *//Importa la interfaz para operaciones CRUD de base de datos*

import org.springframework.stereotype.Repository; *//Marca la clase como un componente de acceso a datos de Spring*

*import com.mx.Alumnos.dominio.Alumnos; //Importa la clase que representa la entidad a manejar*

*@Repository //Anota la interfaz como repositorio Spring Data*

*public interface AlumnosDao extends CrudRepository<Computadora, Integer>{ //Define la interfaz repositorio para la entidad Calzado usando entero como ID*

*}*

*Texto

El contenido generado por IA puede ser incorrecto.*

*Navega a la carpeta scr/main/java*

**

Com.mx.Alumnos.Service

AlumnosService.java

*Texto

El contenido generado por IA puede ser incorrecto.Crea el paquete: Haz clic derecho en la carpeta scr/main/java y selecciona new > Package. Introduce el nombre del paquete: com.mx.Alumnos.service y presiona “finish”*

AlumnosService.java

AlumnosServimpl.java

*Crear la clase: Haz clic derecho a la carpeta recién creada com.mx.Alumnos.service y selecciona new > “Class”. Introduce el nombre de la clase* *AlumnosService y presiona “Finish”*

*Crear la segunda clase: Haz clic derecho a la carpeta recién creada com.mx.Alumnos.service y selecciona new > “Class”. Introduce el nombre de la clase AlumnosSevimpl y presiona “Finish”*

*Para abrir el archivo AlumnosService, simplemente has doble clic sobre el*

AlummnosService.java

*El código de interface AlumnosService*

*package com.mx.Alumnos.Service; // Declaración del paquete donde se encuentra esta interfaz*

*import java.util.List; // Importación de la clase List para manejar listas de objetos*

*import com.mx.Alumnos.dominio.Alumnos; // Importación de la entidad Alumnos*

*// Esta es una interfaz que define los métodos que cualquier clase de servicio que implemente esta interfaz debe tener*

*public interface AlumnosService {*

*public List<Alumnos> listar(); // Método para obtener una lista de todos los objetos Calzado*

*public void guardar(Alumnos alum); // Método para guardar un nuevo objeto Calzado en la base de datos*

*public void editar(Alumnos alum); // Método para editar un objeto Calzado existente*

*public void eliminar(Alumnos alum); // Método para eliminar un objeto Calzado de la base de datos*

*public Calzado buscar(Alumnos alum); // Método para buscar un objeto Calzado específico (por ID u otro atributo que se implemente)*

*}*

*Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.*

*Para abrir el archivo CalzadoServimpl, simplemente has doble clic sobre el*

AlumnosServimpl.java

**

*El siguiente código del apartado AlumnosServimpl*

*package com.mx.Alumnos.Service; // Declaración del paquete donde se encuentra esta clase*

*import java.util.List; // Importación de clases necesarias*

*// Importación de Spring para autowiring y ordenamiento de listas*

*import org.springframework.beans.factory.annotation.Autowired;*

*import org.springframework.data.domain.Sort;*

*import org.springframework.stereotype.Service;*

*// Importación de la interfaz DAO y la entidad Alumnos*

*import com.mx.Alumnos.Dao.AlumnosDao;*

*import com.mx.Alumnos.Dominio.Alumnos;*

*// La anotación @Service marca esta clase como un componente de servicio en Spring,*

*// que será gestionado por el contenedor de Spring.*

*@Service*

*public class AlumServimpl implements AlumService{*

*@Autowired*

*private AlumnosDao aDao;*

*@Override*

*public Alumnos guardar(Alumnos a) {*

*Alumnos save = aDao.save(a);*

*return save;*

*}*

*@Override*

*public Alumnos editar(Alumnos a) {*

*Alumnos put = aDao.save(a);*

*return put;*

*}*

*@Override*

*public Alumnos buscar(Alumnos a) {*

*Alumnos find = aDao.findById(a.getIdAlumnos()).orElse(null);*

*return find;*

*}*

*@Override*

*public void eliminar(Alumnos a) {*

*aDao.delete(a);*

*}*

*@Override*

*public List<Alumnos> listar() {*

*List<Alumnos> lista = aDao.findAll(*

*Sort.by(Sort.Direction.****ASC****,"idAlumnos"));*

*return lista;*

*}*

*}*

*Texto

El contenido generado por IA puede ser incorrecto.*

*Texto

El contenido generado por IA puede ser incorrecto.*

*Navega a la carpeta scr/main/java*

**

Com.mx.Alumnos

*Texto

El contenido generado por IA puede ser incorrecto.Crea el paquete: Haz clic derecho en la carpeta scr/main/java y selecciona new > Package. Introduce el nombre del paquete: com.mx.Alumnos.Controller y presiona “finish”*

AlumController.java

Com.mx.Alumnos.Controller

*Crear la clase: Haz clic derecho a la carpeta recién creada com.mx.Alumnos.WsAlumnos y selecciona new > “Class”. Introduce el nombre de la clase AlumnosService y presiona “Finish”*

*// Declaración del paquete donde se encuentra esta clase*

*package com.mx.Alumnos.Controller;*

*// Importación de clases necesarias para manejar las solicitudes HTTP y las respuestas*

*import java.util.List;*

*import org.springframework.beans.factory.annotation.Autowired;*

*import org.springframework.http.HttpStatus;*

*import org.springframework.http.ResponseEntity;*

*import org.springframework.web.bind.annotation.CrossOrigin;*

*import org.springframework.web.bind.annotation.GetMapping;*

*import org.springframework.web.bind.annotation.PostMapping;*

*import org.springframework.web.bind.annotation.RequestBody;*

*import org.springframework.web.bind.annotation.RequestMapping;*

*import org.springframework.web.bind.annotation.RestController;*

*// Importación del servicio y la entidad Calzado*

*import com.mx.Alumnos.Dominio.Alumnos;*

*import com.mx.Alumnos.Service.AlumServimpl;*

*// Anotación que marca esta clase como un controlador REST*

*@RestController*

*// La ruta base para todos los endpoints de esta clase será 'Calzado'*

*@RequestMapping(path = "Alumnos")*

*// Anotación que permite solicitudes CORS desde cualquier origen (útil para solicitudes desde un frontend)*

*@CrossOrigin("\*")*

*public class WSAlumnos {*

*// Inyección de dependencia del servicio CalzadoServimpl*

*@Autowired*

*private AlumnosServimpl alumServ;*

*// Endpoint para obtener la lista de calzados*

*// URL: http://localhost:8009/Alumnos/*

*@GetMapping*

*public ResponseEntity<Object> listar() {*

*// Llama al método 'listar' del servicio para obtener la lista de calzados*

*List<Alumnos> lista = alumServ.listar();*

*// Verifica si la lista está vacía y responde con un mensaje adecuado*

*return lista.isEmpty() ?*

*ResponseEntity.status(HttpStatus.****NO\_CONTENT****).body("LISTA VACIA") :*

*ResponseEntity.status(HttpStatus.****OK****).body(lista);*

*}*

*// Endpoint para buscar un calzado específico por los datos recibidos en el cuerpo de la solicitud*

*@PostMapping*

*public ResponseEntity<?> buscar(@RequestBody Alumnos alum) {*

*// Llama al método 'buscar' del servicio para encontrar el alumnos*

*alumnos buscarTec = tecServ.buscar(alum);*

*// Si no encuentra el alumnos, responde con un estado 404 (NO\_FOUND)*

*// Si lo encuentra, responde con el objeto encontrado y estado 200 (OK)*

*return buscarTec == null || buscarTec.equals(null) ?*

*ResponseEntity.status(HttpStatus.****NOT\_FOUND****)*

*.body("Alumnos NO ENCONTRADO") :*

*ResponseEntity.status(HttpStatus.****OK****)*

*.body(buscarTec);*

*}*

*// Endpoint para guardar un calzado nuevo*

*@PostMapping("/guardar")*

*public ResponseEntity<Object> Guardar(@RequestBody Alumnos alum) {*

*// Verifica si el calzado ya está registrado en la base de datos*

*Alumnos findtec = tecServ.buscar(calz);*

*if (findtec != null) {*

*// Si ya existe, responde con un conflicto (estado 409)*

*return ResponseEntity.status(HttpStatus.****CONFLICT****).body("El alumnos ya se encuentra registrado");*

*} else {*

*// Si no existe, guarda el calzado y responde con éxito (estado 200)*

*tecServ.guardar(alum);*

*return ResponseEntity.status(HttpStatus.****OK****).body("Alumnos registrado");*

*}*

*}*

*// Endpoint para editar un calzado existente*

*@PostMapping("/editar")*

*public ResponseEntity<Object> editar(@RequestBody Alumnos alum) {*

*// Verifica si el alumnos existe antes de editarlo*

*Alumnos findtec = tecServ.buscar(calz);*

*if (findtec != null) {*

*// Si existe, lo edita y responde con éxito (estado 200)*

*tecServ.editar(calz);*

*return ResponseEntity.status(HttpStatus.****OK****).body("Datos del Alumnos actualizados");*

*} else {*

*// Si no existe, responde con un conflicto (estado 409)*

*return ResponseEntity.status(HttpStatus.****CONFLICT****).body("El Alumnos no está registrado");*

*}*

*}*

*// Endpoint para eliminar un calzado existente*

*@PostMapping("/eliminar")*

*public ResponseEntity<Object> eliminar(@RequestBody Alumnos alum) {*

*// Verifica si el calzado existe antes de eliminarlo*

*Calzado findtec = tecServ.buscar(calz);*

*if (findtec != null) {*

*// Si existe, lo elimina y responde con éxito (estado 200)*

*tecServ.eliminar(calz);*

*return ResponseEntity.status(HttpStatus.****OK****).body("Datos del Alumnos eliminados");*

*} else {*

*// Si no existe, responde con un conflicto (estado 409)*

*return ResponseEntity.status(HttpStatus.****CONFLICT****).body("El alumnos no está registrado");*

*}*

*}*

*}*

*Imagen del código*

*Texto

El contenido generado por IA puede ser incorrecto.*

*Texto

El contenido generado por IA puede ser incorrecto.*

*Texto

El contenido generado por IA puede ser incorrecto.*

[*http://localhost:8009/*](http://localhost:8009/)

*En tu navegador de confianza, puedes usar este enlace para visualizar la base de datos en formato JSON. Este endpoint es de tipo GET y, al acceder, el servidor responderá con una lista de calzados almacenadas en la base de datos. Es una manera práctica y eficiente de consultar toda la información de las computadoras.*

**

8009

CrudAlumnos

***Elementos para su funcionamiento:***

* ***"local"****: Esto indica que la aplicación está funcionando en un servidor local, es decir, en la misma computadora donde se está ejecutando la aplicación web.*
* ***Icono de encendido verde****: Este icono sugiere que el servidor local está activo y funcionando correctamente.*
* ***Flecha verde hacia arriba****: Este icono podría indicar que la aplicación está en modo de desarrollo o que se está ejecutando en un entorno de prueba.*
* ***"CrudAlumnos [devtools] [:8009]"****: Esto parece ser un título o un nombre para la aplicación. "CRUD" probablemente se refiere a las operaciones de Creación, Lectura, Actualización y Eliminación (Create, Read, Update, Delete) de datos. "Calzado" podría indicar que esta aplicación está relacionada con el manejo de datos de computadoras. "[devtools]" sugiere que esta aplicación está diseñada para ser utilizada por desarrolladores. "[:8007]" podría ser el puerto en el que se está ejecutando el servidor local (probablemente un puerto web).*

*Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.*

*Postman* Logotipo

El contenido generado por IA puede ser incorrecto.

*Postman es una herramienta muy útil que sirve para probar y desarrollar APIs (interfaces de programación de aplicaciones). Te permite enviar solicitudes HTTP (GET, POST, DELETE, etc.) a un servidor y ver las respuestas en tiempo real. Esto es ideal para verificar que los endpoints de una API funcionan correctamente y devuelven los datos esperados. Además, puedes automatizar pruebas, guardar colecciones de solicitudes y colaborar con otros desarrolladores. Es como un laboratorio para trabajar con APIs de manera sencilla y eficiente.*

*Abrir postman*

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.*“Collection”.*

*Asigna un nombre a tu nueva colección (por ejemplo “CRUD Calzado”) y guarda.*

*Dentro de esa colección, puedes*

*añadir múltiples solicitudes HTTP (como GET, POST, DELETE, etc.)*

*En la sección donde aparece el nombre de la colección, encontrarás dos símbolos: una estrella (⭐), que sirve para marcar la colección como favorita, y tres puntos horizontales (…), que despliegan un menú con opciones adicionales como editar, eliminar o exportar la colección.*

*Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.Debe hacer clic en \*Add Request\* para crear una nueva solicitud dentro de una colección.*

*En la parte donde dice New Collection, se asignar como nombre “listar”*

*En la parte de debajo de new collection, ubicamos el GET, asignamos la URL* [*http://localhost:8009/Calzado*](http://localhost:8009/Calzado) *y presionamos “SEND”*

*Texto

El contenido generado por IA puede ser incorrecto.Nos mostrara en formato JSON todos los elementos de la base de datos.*

*Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.*

*En la parte donde dice New Collection, se asignar como nombre “Guardar”*

*En la parte de debajo de new collection,*

*seleccionamos POST Y asignamos la URL* [*http://localhost:8009/guardar*](http://localhost:8009/guardar)

1. ***Localiza la sección "PARAMS":****Busca la sección "PARAMS" en la parte superior de la interfaz.*
2. ***Selecciona "BODY" y luego "RAW":****Dentro de "PARAMS", selecciona "BODY" y cambia el modo a "RAW". Esto te permitirá agregar datos en formato sin procesar.*
3. ***Agrega un id\_Alumnos único:****Asegúrate de generar un id\_Alumnos que sea único y diferente a cualquier otro id\_Alumnos existente en la base de datos. Esto es crucial para evitar conflictos.*
4. ***Ingresa los demás campos:****Agrega los valores correspondientes a todos los demás campos de la base de datos para la nuevo alumnos.*
5. ***Envía los datos:****Una vez que hayas ingresado todos los datos en el formato correcto, presiona "Send".*
6. ***Verifica el registro:****La interfaz mostrará una confirmación o una lista de los elementos agregados, incluyendo el nuevo registro de la calzado con su id\_Alumnos*

*Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.En la parte donde dice New Collection, se asignar como nombre “Editar”*

*En la parte de debajo de new collection,*

*seleccionamos POST, asignamos la URL* [*http://localhost:8009/api/Alumnos/Editar*](http://localhost:8009/api/Alumnos/Editar)

1. ***Localiza la sección "PARAMS":****Busca la sección "PARAMS" en la parte superior de la interfaz.*
2. ***Selecciona "BODY" y luego "RAW":****Dentro de "PARAMS", selecciona "BODY" y cambia el modo a "RAW". Esto te permitirá editar el código fuente de los datos.*
3. ***Copia los elementos:****Copia todos los elementos que describen la configuración de la computadora.*
4. ***Edita el campo de Nombre:****Busca el campo que especifica la cantidad de Nombre.*
5. ***Envía los cambios:****Una vez que hayas realizado el cambio, pega el código modificado en el campo "RAW" y presiona "Send".*
6. ***Verifica los cambios:****La interfaz mostrará una lista de los elementos modificados*

*Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.En la parte donde dice New Collection, se asignar como nombre “Eliminar ”*

*En la parte de debajo de new collection,*

*seleccionamos DELETE, asignamos la URL* [*http://localhost:8009/api/Alumnos/Eliminar.*](http://localhost:8009/api/Alumnos/Eliminar.)

1. ***Localiza la sección "PARAMS".****En la parte superior, busca la sección etiquetada como "PARAMS".*
2. ***Selecciona "BODY" y luego "RAW".****Dentro de la sección "PARAMS", selecciona "BODY" y cambia el modo a "RAW".*
3. ***Copia los elementos del Alumnos.****Copia todos los elementos o campos de la computadora que deseas modificar. Recuerda que solo se eliminarán los datos que selecciones.*
4. ***Pega y envía.****Pega los elementos copiados en el campo "RAW" y presiona "Send".*
5. ***Revisa los cambios.****En la sección de listado, se mostrarán los elementos cuyo valor ha cambiado.*