# Examen 2

Repositorio GIT: <https://github.com/David-GaTre/parcial_2_web>

## 1.Creacion de base de datos y tabla de alumnos

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

## 2.Conexión Base de datos

Texto

Descripción generada automáticamente

En esa screenshot hay varias cosas a notar, la parte de la conexión a la base de datos se realiza con el dataConn.js y el db.js, la parte de alumnos model es la que se encarga de hacer los métodos a la base de datos, como prueba de la conexión hicimos un select de todos los alumnos que actualmente están en la base de datos, aquí solo esta el insert de prueba que hice en la tabla de ejemplo.

OJO: LA CONTRASEÑA ESTA EN EL ARCHIVO password.txt QUE SE PONE EN EL GIT IGNORE POR PRIVACIDAD.

## 3. API Requests con su script

INSERT

Texto

Descripción generada automáticamente

sql.query("INSERT INTO Alumnos SET ?", newAlumno, (err, res) => {

        if (err) {

            console.log("error: ", err);

            result(err, null);

            return;

        }

        console.log("Created Alumno: ", { id: res.insertId, ...newAlumno });

        result(null, { id: res.insertId, ...newAlumno });

    });

UPDATE

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

sql.query(

      "UPDATE Alumnos SET Nombre = ?, Identificador = ?, Email = ?, Carrera = ? WHERE Id = ?",

      [alumno.Nombre, alumno.Identificador, alumno.Email, alumno.Carrera, id],

      (err, res) => {

        if (err) {

          console.log("error: ", err);

          result(null, err);

          return;

        }

        if (res.affectedRows == 0) {

          result({ kind: "not\_found" }, null);

          return;

        }

        console.log("Updated Alumno: ", { id: id, ...alumno });

        result(null, { id: id, ...alumno });

      }

    );

DELETE

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

sql.query("DELETE FROM Alumnos WHERE Id = ?", id, (err, res) => {

      if (err) {

        console.log("error: ", err);

        result(null, err);

        return;

      }

      if (res.affectedRows == 0) {

        result({ kind: "not\_found" }, null);

        return;

      }

      console.log("Deleted Alumno with id: ", id);

      result(null, res);

    });

SELECT

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

let query = "SELECT \* FROM Alumnos;";

    sql.query(query, (err, res) => {

        if (err) {

        console.log("error: ", err);

        result(null, err);

        return;

        }

        console.log("Alumnos: ", res);

        result(null, res);

    });