

1.4. Parte experimental

1.4.2. Diseño. ¿Cómo se hará la aplicación?

- **Estructura de la base de datos**

Para la estructura de la base de datos siguiendo la lógica de los anteriores diagramas diseñados, he decidido utilizar el formato de MySQL para la creación de esta y gestionarla a través del software de DBeaver.

DBeaver es una herramienta de administración de bases de datos. Para las bases de datos relacionales, utiliza la API de JDBC para interactuar con las bases de datos a través de un controlador JDBC.

Las tablas principales serían:

```
CREATE TABLE profesor(
    id INT CONSTRAINT profesor_pk PRIMARY KEY AUTO_INCREMENT,
    codigo_profesor VARCHAR(1) NOT NULL UNIQUE,
    nombre VARCHAR(50) NOT NULL,
    correo VARCHAR(50) NOT NULL,
    contraseña VARCHAR(50) NOT NULL
);

CREATE TABLE alumno(
    id INT CONSTRAINT alumno_pk PRIMARY KEY AUTO_INCREMENT,
    codigo_alumno VARCHAR(4) NOT NULL,
    correo VARCHAR(50) NOT NULL,
    contraseña VARCHAR(10) NOT NULL,
    icono VARCHAR(300) DEFAULT NULL,
    nombre VARCHAR(50) NOT NULL,
    puntos int DEFAULT NULL,
    CONSTRAINT alumno_profesor_id_profesor_fk FOREIGN KEY (id_profesor)
REFERENCES profesor (id_profesor) ON DELETE CASCADE ON UPDATE
CASCADE
);

CREATE TABLE mundo (
    id INT CONSTRAINT mundo_pk PRIMARY KEY AUTO_INCREMENT,
```

```
    codigo_mundo VARCHAR(2) NOT NULL,  
    dificultad INT NOT NULL,  
    puntos_desbloqueo INT NOT NULL,  
    tematica VARCHAR(50) NOT NULL  
);
```

```
CREATE TABLE nivel (  
    id INT CONSTRAINT nivel_pk PRIMARY KEY AUTO_INCREMENT,  
    codigo_nivel VARCHAR(4) NOT NULL,  
    dificultad INT NOT NULL,  
    puntos_desbloqueo INT NOT NULL,  
    puntos_minimos INT NOT NULL,  
    CONSTRAINT nivel_recompensa_id_recompensa_fk FOREIGN KEY (id_recompensa)  
        REFERENCES recompensa (id_recompensa) ON DELETE CASCADE ON UPDATE  
        CASCADE,  
    CONSTRAINT nivel_test_id_test_fk FOREIGN KEY (id_test)  
        REFERENCES test (id_test) ON DELETE CASCADE ON UPDATE CASCADE,  
    CONSTRAINT nivel_leccion_id_leccion_fk FOREIGN KEY (id_leccion)  
        REFERENCES leccion (id_leccion) ON DELETE CASCADE ON UPDATE  
        CASCADE,  
    CONSTRAINT nivel_mundo_id_mundo_fk FOREIGN KEY (id_mundo)  
        REFERENCES mundo (id_mundo) ON DELETE CASCADE ON UPDATE  
        CASCADE  
);
```

```
CREATE TABLE recompensa (  
    id INT CONSTRAINT recompensa_pk PRIMARY KEY AUTO_INCREMENT,  
    tipo VARCHAR(30) NOT NULL,  
    ruta_recompensa VARCHAR(300) NOT NULL,  
    desbloqueada BOOLEAN NOT NULL  
);
```

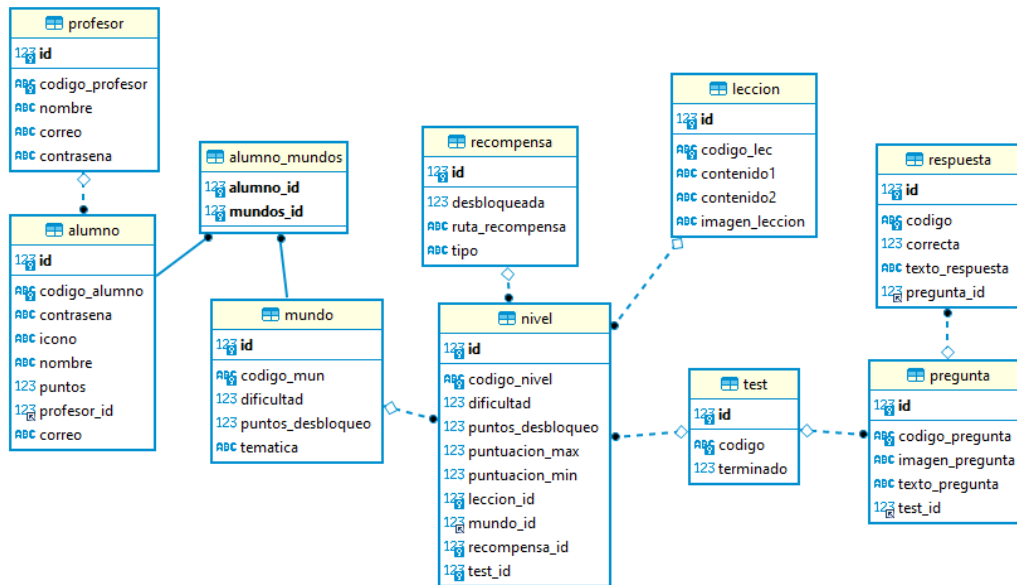
```
CREATE TABLE leccion (  
    id INT CONSTRAINT leccion_pk PRIMARY KEY AUTO_INCREMENT,
```

```
    codigo_leccion VARCHAR(4) NOT NULL,  
    contenido1 VARCHAR(500) NOT NULL,  
    contenido2 VARCHAR(500),  
    imagen_leccion VARCHAR(300)  
);
```

```
CREATE TABLE test (  
    id INT CONSTRAINT test_pk PRIMARY KEY AUTO_INCREMENT,  
    codigo_test VARCHAR(4) NOT NULL,  
    terminado BOOLEAN NOT NULL  
);
```

```
CREATE TABLE pregunta (  
    id INT CONSTRAINT pregunta_pk PRIMARY KEY AUTO_INCREMENT,  
    codigo_pregunta VARCHAR(6) NOT NULL,  
    texto_pregunta VARCHAR(100) NOT NULL,  
    imagen_pregunta VARCHAR(300),  
    CONSTRAINT pewgunta_test_id_test_fk FOREIGN KEY (id_test)  
        REFERENCES test (id_test) ON DELETE CASCADE ON UPDATE CASCADE,  
);
```

```
CREATE TABLE respuesta (  
    id INT CONSTRAINT test_pk PRIMARY KEY AUTO_INCREMENT,  
    codigo_pregunta VARCHAR(6) NOT NULL,  
    texto_respuesta VARCHAR(100) NOT NULL,  
    correcta BOOLEAN NOT NULL,  
    CONSTRAINT respuesta_pregunta_id_pregunta_fk FOREIGN KEY (id_pregunta)  
        REFERENCES pregunta (id_test) ON DELETE CASCADE ON UPDATE CASCADE,  
);
```



- **Arquitectura del proyecto: Tecnologías utilizadas en cada parte del proyecto**

Las principales tecnologías que he empleado en el desarrollo del proyecto son las siguientes:

- ➔ **Adobe XD:** Software usado para la creación de interfaces, tanto para páginas webs como para aplicaciones, perteneciente a la suite de Adobe.
Usado para el prototipado de la aplicación, el diseño de sus vistas y flujo de navegación que hay en las pantallas.
- ➔ **Visual Studio Code:** Editor de código fuente desarrollado por Microsoft.
Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código.
Usado para la parte Front de la aplicación, el código HTML, realización de las hojas de estilo CSS y documentos JavaScript.
- ➔ **SASS:** Lenguaje de hoja de estilos en cascada y preprocesador CSS, es una herramienta que genera automáticamente las hojas en base a variables definidas en el archivo SCSS de la aplicación.
Usado para la compilación de las hojas de estilo CSS, utilizado sobretodo para la definición de variables de la paleta de color.
- ➔ **JavaScript:** Lenguaje de programación que entienden los navegadores, con el que se desarrolla la parte de la funcionalidad frontend en sitios web y aplicaciones web.

Usado para la validación de formularios, acciones y diseños interactivos en la web.

- **Bootstrap:** Biblioteca multiplataforma y conjunto de herramientas de código abierto para diseño de aplicaciones web. Con clases y estilos predeterminados. Usado para la estructura y estilos responsive de las vistas de la aplicación.
- **IntelliJ Idea:** Entorno de desarrollo integrado para el desarrollo de programas informáticos. Es desarrollado por JetBrains y proporciona asistencia a la codificación inteligente para determinados lenguajes de desarrollo de código. Usado para la estructura y lógicas de la parte Back de la aplicación.
- **Jakarta:** Plataforma de Java Enterprise Edition o Java EE para desarrollar y ejecutar software de aplicaciones en el lenguaje de programación Java. Usado para la escritura del código fuente de la aplicación.
- **draw.io:** Diagrams.net, fundado por draw.io, es un software de creación de diagramas libre. Comprende todo tipo de diagramas de base de datos relacionales, UML, esquemas de tablas. etc. Usado para la realización de diagramas y esquemas para la base de datos.
- **JPA:** Java Persistence API, más conocida por sus siglas JPA, es una API de persistencia desarrollada para la plataforma Java EE. Proporciona un mecanismo para gestionar la persistencia y la correlación relacional de objetos. La persistencia de datos es un medio mediante el cual una aplicación puede recuperar información desde un sistema de almacenamiento no volátil. Usado para el manejo, creación, modificación y gestión de las tablas en la base de datos dentro de la aplicación.
- **DBEaver:** Aplicación de software cliente de SQL, a través de ella podemos gestionar las conexiones a las bases de datos. También podemos llevar su creación, modificación y gestión de datos a través de esta. Usado para la conexión entre la aplicación y la base de datos. También para la gestión y la carga de datos de una manera más concreta visualmente durante el desarrollo de la aplicación.
- **Docker:** Proyecto de código abierto que automatiza el despliegue de aplicaciones dentro de contenedores de software. Automatiza la virtualización de aplicaciones en múltiples sistemas operativos. Usado para el despliegue de la aplicación y conexión con la base de datos.