Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Estado de México

Desarrollo e implantación de sistemas de software

Quiz Class Project

Omar Rodrigo Talavera Becerra - A0752221
Juan Pablo Castañeda Serrano - A01752030
Nadia Paola Ferro Gallegos - A01752013

Docente:
Mikaela Rae Grace

29 de Mayo, 2023

General Overview

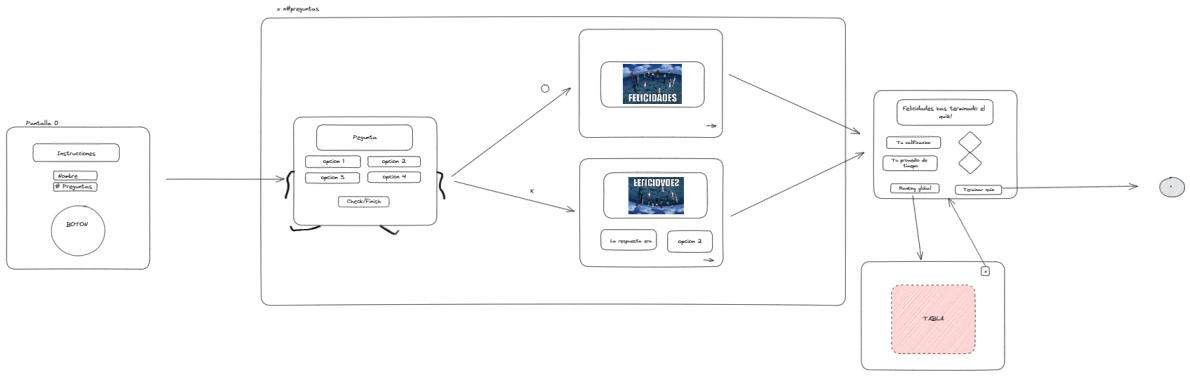
The project “quiz-app-final-proyect” is a quiz application that provides an interactive quiz. The project utilices a full-stack approach, using React on the frontend and Python on the backend. The main purpose of the application is to allow users to test their knowledge about design patterns, anti patterns, refactoring and UML.

The backend handles the storage of quiz questions, user information and ranking data. The frontend serves as the intermediary between the user and the system, facilitating communication and interaction.

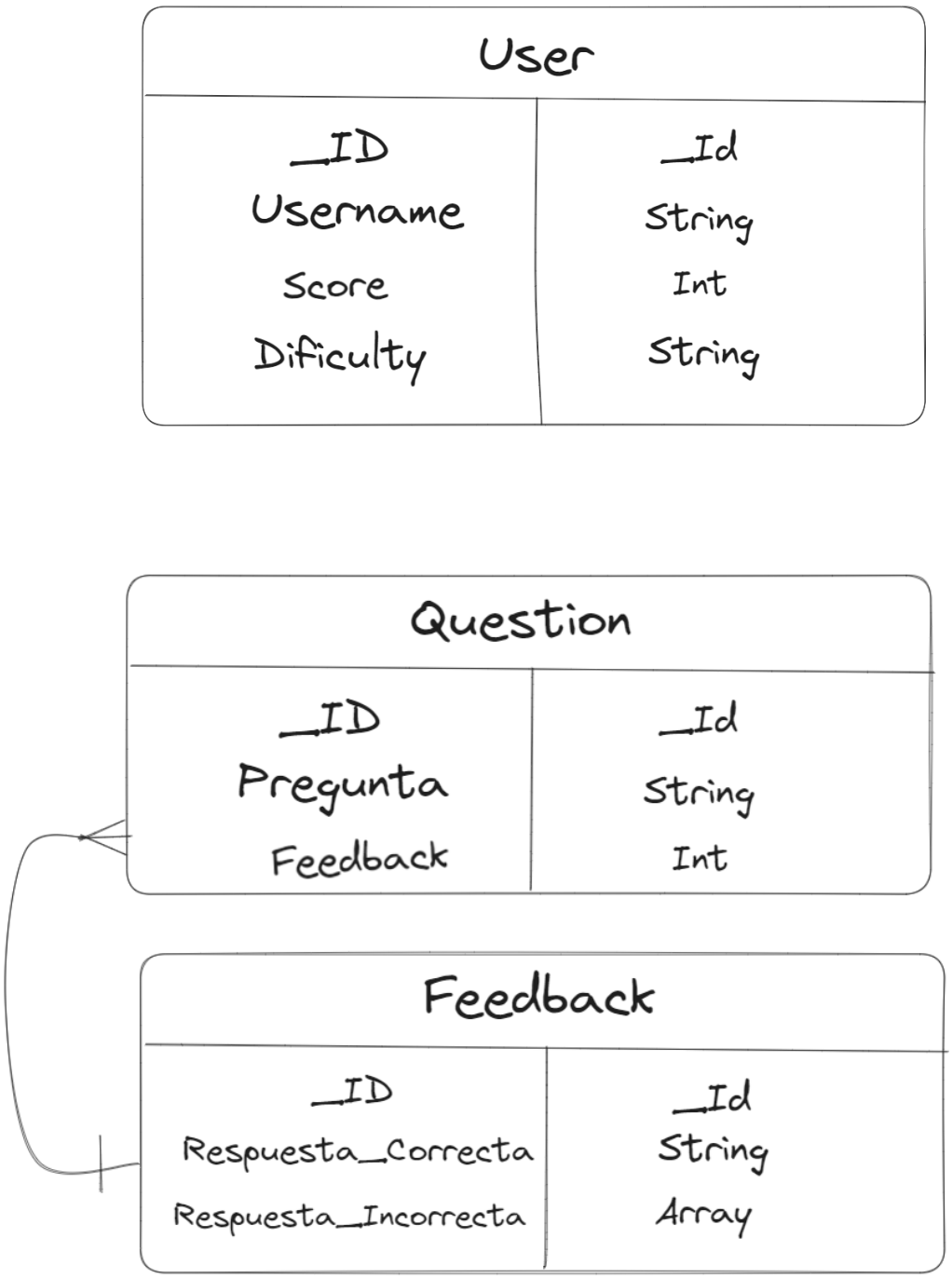
At the moment of playing the quiz you will find that questions are categorized into three difficulties: hard, medium and easy. The application tracks the user's progress and provides immediate feedback. At the end of the quiz, the user is shown their final score, and rank based on the global ranking, so the ranking system allows the user to compare their score with other participants.

System diagrams

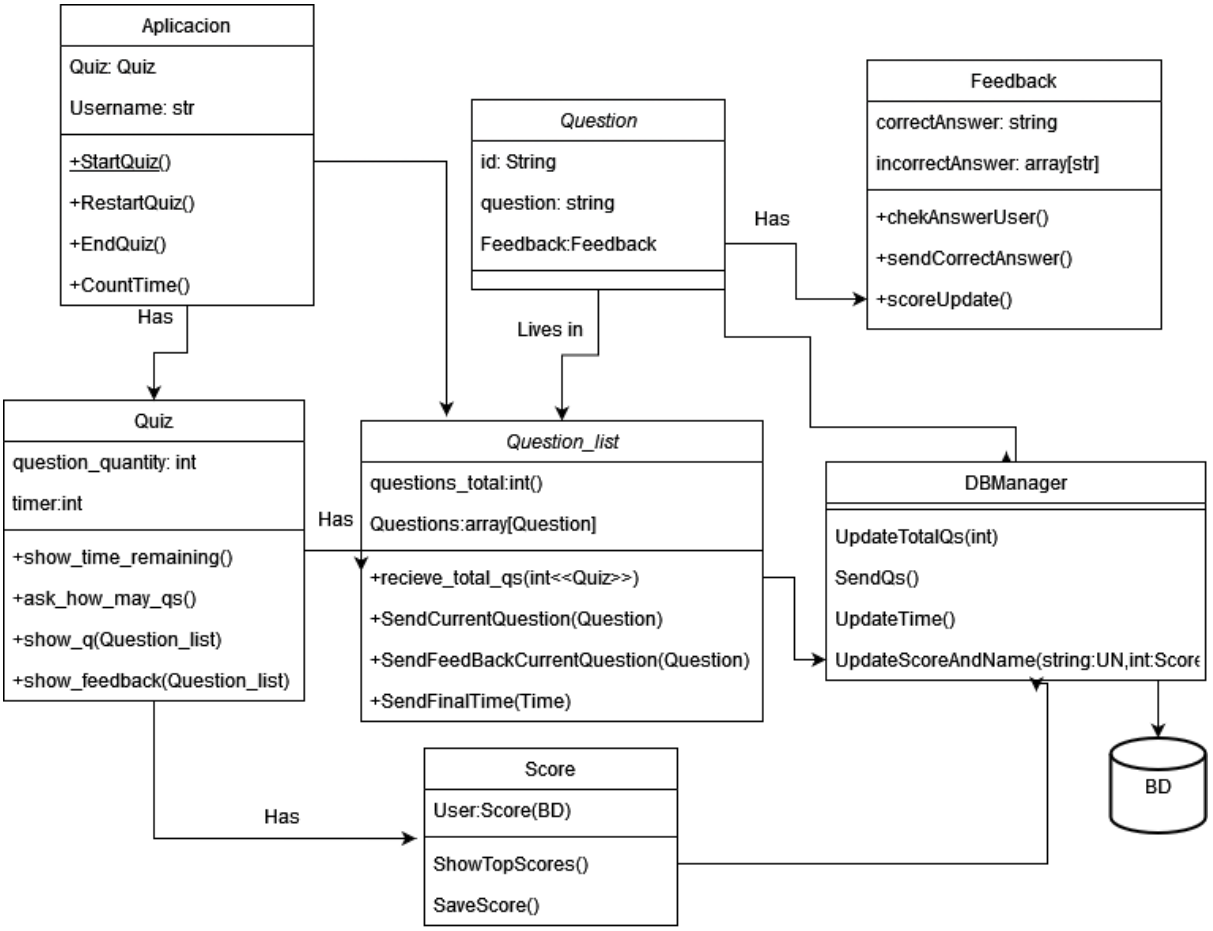
Front-end diagram



Database diagram



Architecture Diagram



Patterns used

Model-View-Controller (MVC)

The MVC pattern was chosen for the organization of the back-end. It separates the application logic into the Model, where all the data is, the View, which is the presentation layer, and the controller which handles the data flow between the Model and the View. The MVC pattern was chosen to achieve modularity, it allows development and testing of different components, and it improves the code maintainability. In the back-end it is visible in the files and directories, because they are separated by its responsibilities.

Single-page application (SPA)

The SPA pattern was implemented in the front-end. The SPA is a type of web application that runs all of its content on a single page. It allows a smooth user experience by updating the content on a single web page, eliminating the need for page refreshes. This pattern improves performances and reduces server load

Execution

This is the application executed.
https://youtu.be/sVilmjN89a8

Application:
-The user who wants to answer the quiz must make a request to the team.
-The team will enable the quiz and send the applicant a link to access the page.
-Show with ngrok

Local:

- Clone the repository of github.
- Make sure you have Node.js and Python installed in your system.
- Install boto3 if you don't have it installed.
- If you don't have an AWS account, make an account.
- Configure AWS cli with their credentials
- Run: "database", "userModel.py", "QuestionsModel.py", "InsertQuestions.py", "Instertusers.py", "controllers.py".
- Go to "FrontEnd" in the directory.
- Run "npm i " in the console.
- Once all the dependencies are installed, run "npm start" in the console
- The front-end application will run in your browser at "http://localhost:3000"
- The back-end application will be up and running at ""