

PROYECTO QUIZZ

Bases de datos

SQL

Las consultas a la base de datos de SQL se harán desde el usuario administrador con el fin de obtener, borrar u editar los datos personales, las preguntas y los quiz, así como para obtener las estadísticas asociadas a cada pregunta, quizz o usuario invitado.

Se usará la base de datos relacional para almacenar la información de los usuarios administradores y las estadísticas e identificadores de sus juegos guardados. Esos mismos identificadores se usarán en la base de datos no relacional para identificarlos y extraer las preguntas y respuestas.

Usuarios administradores – Users	Invitados– User_guests
id INT AUTO_INCREMENT, user_name VARCHAR(25) UNIQUE, name VARCHAR(25), country VARCHAR(50), email VARCHAR(50), password VARCHAR(38), type_education VARCHAR(50), institution VARCHAR(50), role VARCHAR(50), logo VARCHAR(2048), PRIMARY KEY(id)	id INT AUTO_INCREMENT, name VARCHAR(100),

Preguntas asociadas al usuario – User_questions	Quizzes asociadas al usuario – User_quizzes
id INT AUTO_INCREMENT, fk_user_name VARCHAR(15), answered_right INT, answered_wrong INT, topic VARCHAR(25), level VARCHAR(25), PRIMARY KEY(id), FOREIGN KEY (fk_user_name) REFERENCES Users(user_name)	id INT AUTO_INCREMENT, fk_user_name VARCHAR(15), name VARCHAR(100), topic VARCHAR(25), level VARCHAR(25), total_guests INT, total_successful INT, total_played INT, PRIMARY KEY(id), FOREIGN KEY (fk_user_name) REFERENCES users(user_name)

Answers
id INT AUTO_INCREMENT, fk_user_name VARCHAR(15), fk_guest_name VARCHAR(100) name VARCHAR(100), fk_quizz_id VARCHAR(15), fk_question_id VARCHAR(15), answer_right: Bool, date DATE, topic VARCHAR(25), level VARCHAR(25), time FLOAT, PRIMARY KEY(id), FOREIGN KEY (fk_user_name) REFERENCES users(user_name) FOREIGN KEY (fk_quizz_id) REFERENCES Users_quizzes(user_name) FOREIGN KEY (fk_question_id) REFERENCES Users_questions(user_name) FOREIGN KEY (fk_guest_name) REFERENCES Users_guests(user_name)

NO-SQL

En esta base de datos se almacenarán las preguntas y respuestas para poder ejecutar el juego. Igualmente estarán relacionadas mediante los correspondientes ID con su usuario administrador, y entre ellas, id de quizz y preguntas.

Además se usarán dos tablas auxiliares mientras se desarrolla el juego:

Una que contiene los quizzes activos en ese momento.

Otra que almacenara de forma temporal las respuestas de los usuarios.

quizzes_normal	Questions_normal
id id_usuario name level topic questions_id	id id_usuario level topic question right_answer wrong_answers
quizzes_video	Questions_video
id id_usuario name level topic questions_id video	id id_usuario level topic time_mark question right_answer wrong_answers

quizzes_running	questions_running
Id	Id
Id_quizz	Id_quizz
id_usuario	id_usuario
ids_guest []	id_guest
guest_ansewers	question_n
guest_times	guest_ansewers_n
date	guest_times_n
state	date
	state

Rutas - Front

Se establecerán las siguientes rutas para el cliente:

Invitado	Admin
/:quizz	/login
/:quizz/:user	/register
	/admin/
	/admin/stats/
	/admin/create-quizz
	/admin/create-video
	/admin/live

*La tura raíz(/) simplemente servirá de puente para las rutas del invitado y el admin.

Rutas – Back

POST api/login	{email: user_email, pass: user_pass}
POST api/register	{user_name: user_name, Name: name, Country: country, Email: email, Password: password, type_education: type_education, institution: institution, role: role, logo: logo_url}
Juego – desde el invitado	
POST api/:quizz/:user Pasa por parámetro el código del quizz y el nombre del usuario.	Params:quizz Params: user_name
GET api/quizz_normal/:quizz api/questions_normal/:quizz/:question_number Se recibe la información para ir sacando el juego pregunta a pregunta.	{Quizz_name: name, Level: level, Topic: topic, wrong_answers: wrong, right_answer:right}
GET api/quizz_video/:quizz api/questions_video/:quizz/:question_number Se recibe la información para ir sacando el juego pregunta a pregunta.	{Quizz_name: name, Level: level, Topic: topic, wrong_answers: wrong, right_answer:right}
POST Api/quizz_running/:quizz/:user Se envían los resultados finales a la tabla auxiliar. Se añade el usuario a Users_guest. Se añaden resultados a Answers.	{Id_quizz: id, id_guest: id, guest_ansewers: answers, guest_time: DATE}
Juego - desde el admin	
POST api/:admin_id/:quizz/	Params:admin_id Params:quizz

Se envían los datos para dar comienzo al juego e identificar el quizz. Se comprueba el token de identificación.	
GET api/quizz_normal/:quizz api/questions_normal/:quizz/:question_number Se recibe la información para ir sacando el juego pregunta a pregunta.	{Quizz_name: name, Level: level, Topic: topic, questions: questions}
GET api/quizz_video/:quizz api/questions_video/:quizz/:question_number Se recibe la información para ir sacando el juego pregunta a pregunta.	{Quizz_name: name, Level: level, Topic: topic, questions: questions}
PUT Api/quizz_running/:quizz/:user Se modifica la tabla auxiliar para cambiar su estado. Se actualizan User_questions y User_Quizzes	{state: finished}
Admin panel - cuenta	
GET api/:admin_id/ Recoge los datos del usuario	{user_name: user_name, Name: name, Country: country, Email: email, Password: password, type_education: type_education, institution: institution, role: role, logo: logo_url}
PUT api/:admin_id/update Modifica los datos del usuario	
DELETE api/:admin_id/delete Elimina los datos del usuario	
Admin panel – Quizzes normal	
GET Api/:admin_id/quizz_normal Saca listado de quizz del usuario	{id: id, Name: name, Level: level, Topic: topic, questions_id, questions_id}
POST Api/:admin_id/quizz_normal /new Añade un nuevo quizz	{Name: name, Level: level, Topic: topic}
DELETE Api/:admin_id/quizz_normal/:quizz_id/delete	Params: quizz_id

Elimina el quizz.	
GET Api/:admin_id/quizz_normal/:quizz_id/stats Saca las stats del quizz	{id:id, Name: name, Level: level, Topic: topic, questions_id, questions_id} {fk_guest_name, fk_question_id, answer_right, date DATE, topic, level, time}
GET Api/:admin_id/quizz_normal/:quizz_id Saca listado de preguntas del quizz.	Param:quizz_id {questions: [questions, questions_id]}
GET Api/:admin_id/quizz_normal/:question_id Saca la pregunta y respuestas.	{question: question, right_answer: right_answer, wrong_answers: wrong_answers}
POST Api/:admin_id/quizz_normal/:question_id/new Añade pregunta nueva en questions y quizzes	{question: question, right_answer: right_answer, wrong_answers: wrong_answers}
PUT Api/:admin_id/quizz_normal/:question_id/update Modifica la pregunta y respuestas.	{question: question, right_answer: right_answer, wrong_answers: wrong_answers}
DELETE Api/:admin_id/quizz_normal/:question_id/delete Elimina la pregunta y respuestas. Modifica el quizz y elimina la pregunta.	Params: question_id
Admin panel – quizzes video	
GET Api/:admin_id/quizz_video Saca listado de quizz del usuario	{id:id, Name: name, Level: level, Topic: topic, questions_id, questions_id}
POST Api/:admin_id/quizz_video /new Añade un nuevo quizz	{Name: name, Level: level, Topic: topic}
DELETE	Params: quizz_id

Api/: admin_id/quizz_video/:quizz_id/delete Elimina el quizz.	
GET Api/: admin_id/quizz_video/:quizz_id/stats Saca las stats del quizz	{id:id, Name: name, Level: level, Topic: topic, questions_id, questions_id} {fk_guest_name, fk_question_id, answer_right, date DATE, topic, level, time}
GET Api/: admin_id/quizz_video/:quizz_id Saca listado de preguntas del quizz.	Param:quizz_id {questions: [questions, questions_id]}
GET Api/: admin_id/quizz_video/:question_id Saca la pregunta y respuestas.	{question: question, right_answer: right_answer, wrong_answers: wrong_answers}
POST Api/: admin_id/quizz_video/:question_id/new Añade pregunta nueva en questions y quizzes	{question: question, right_answer: right_answer, wrong_answers: wrong_answers}
PUT Api/: admin_id/quizz_video/:question_id/update Modifica la pregunta y respuestas.	{question: question, right_answer: right_answer, wrong_answers: wrong_answers}
DELETE Api/: admin_id/quizz_video/:question_id/delete Elimina la pregunta y respuestas. Modifica el quizz y elimina la pregunta.	Params: question_id