

Tarea III Bases de Datos II (Proyecto)

Diego Alvarez S. – 14/11/2024

Objetivo

Aplicar todos los conocimientos adquiridos en el curso para la confección de una aplicación web que haga uso de una API RESTful y una base de datos relacional.

Requerimientos

Implementar uno de los proyectos propuestos o un proyecto de autoría propia que cumpla con los siguientes requerimientos:

- La aplicación debe contemplar un sistema de autenticación, y autorización.
- La base de datos para la aplicación debe contar con un mínimo de 5 tablas.
- Todas las funcionalidades de la aplicación deben ser accesibles a través del frontend.

Condiciones y Evaluación

- El proyecto puede realizarse en parejas.
- El backend debe ser implementado con las tecnologías vistas en clase, es decir, Python usando SQLAlchemy y Litestar. Se evaluará el diseño de la base de datos en función de los requerimientos del proyecto (15%) y la implementación de las características solicitadas (45%). Además se evaluará la organización y calidad del código de backend (uso correcto de tipos y clases y funciones, manejo de errores, etc) (10%).
- El frontend puede ser implementado en cualquier tecnología, y sólo se evaluará que la interfaz permita realizar las funcionalidades solicitadas (30%).
- El proyecto debe inscribirse mediante correo electrónico a diego.alvarez@umag.cl, debe esperarse la confirmación de la inscripción para comenzar a trabajar en el proyecto. Si es que una propuesta ya está inscrita, se deberá escoger otra. Sólo si es que todas las propuestas están inscritas, se podrá repetir el proyecto.
- Debe crearse un repositorio en GitHub para el proyecto. Este repositorio debe ser creado y compartido con el profesor (dialvarezs) desde el momento en que el proyecto sea inscrito. Deben realizarse commits regulares, indicando mediante la descripción que funcionalidad o cambios se han implementado.

Proyectos Propuestos

P1. Plataforma de coordinación de clases privadas

Implementar un sistema de coordinación de clases privadas. En esta plataforma, cada usuario registrado puede ser profesor y/o alumno. La plataforma debe tener las siguientes funcionalidades:

- Debe permitir a los usuarios publicar temas sobre los cuales pueden dar clases, por ejemplo "Programación en Python", indicando el precio por hora. Adicionalmente, el profesor puede definir sus horarios disponibles.
- Como usuario que busca clases, se debe poder buscar profesores por tema y ver sus horarios disponibles y el precio de sus clases.
- Un usuario puede solicitar una clase a un profesor, indicando el horario que contempla la solicitud. El profesor puede aceptar o rechazar la solicitud. Para simplificar los horarios, se asume que las clases sólo pueden ser en horas completas, por ejemplo, de 9:00 a 10:00.
- Los usuarios deben poder ver un historial de clases pasadas y futuras.
- En el perfil público del profesor se debe indicar la cantidad de horas de clases que ha dado por tema. Cuando sea el profesor quien está viendo su propio perfil, se debe mostrar las ganancias que ha obtenido en el último mes.

P2. Plataforma de organización de torneos de videojuegos

Implementar una plataforma para la organización de torneos de videojuegos, donde pueden participar teams o jugadores individuales. La plataforma debe permitir:

- Con el rol de organizador se debe poder crear torneos, indicando el juego, la fecha de inicio y fin, el premio, si el torneo está dirigido a teams o a jugadores individuales y la cantidad máxima de participantes.
- Los jugadores o teams deben poder ver los torneos disponibles e inscribirse en ellos.
- Cuando el organizador inicia un torneo, se debe generar los emparejamientos aleatorios de los participantes.
- Al finalizar una partida, se debe indicar el resultado de la misma. El equipo ganador o jugador ganador debe pasar a la siguiente ronda (si es que existe).
- Al finalizar el torneo, se debe indicar el equipo o jugador ganador y el premio que recibirá.
- En el perfil de cada equipo / jugador deben indicarse los torneos en los que ha participado y los premios que ha ganado.

P3. Sistema de gestión de inventario y ventas

Implementar un sistema de gestión de inventario, compras y ventas para una pequeña tienda. El sistema debe permitir:

- Gestionar productos de la tienda. Cada producto debe tener como mínimo un nombre, una descripción, un precio, una imagen y un stock.
- Ingreso de compras de artículos de la tienda. Cada compra debe tener una fecha, un proveedor, y una lista de productos comprados con su cantidad y precio.
- Venta de productos de la tienda. Cada venta como mínimo debe tener una fecha, y la lista de productos vendidos con su cantidad y precio.
- Las compras y ventas deben actualizar el stock de los productos.

- Debe poder generarse un reporte de ventas por un rango de fechas. El reporte debe mostrar la cantidad de productos vendidos, el monto total de las ventas y el monto total de las ganancias (descontando de el precio de venta el IVA y el costo de los productos vendidos).

P4. Plataforma de organización de viajes

Implementar una plataforma para la organización de viajes entre múltiples personas. La plataforma debe permitir:

- Crear un viaje e invitar a otros usuarios a participar en él.
- Cualquier usuario puede agregar ciudades al viaje, indicando la fecha de llegada, la ciudad, un enlace a Google Maps con la ubicación y una descripción.
- Se deben poder agregar atracciones turísticas a cada ciudad, indicando la fecha y hora de visita, la atracción y la duración esperada de la actividad, y un enlace de Google Maps opcionalmente.
- Se debe poder añadir transportes, que deben indicar la ciudad de origen y destino, duración del viaje, medio de transporte y precio.
- Con la información ingresada del viaje, debe poder generarse un itinerario, en el cual se debe mostrar por cada día un cronograma que incluya las actividades y transportes.

P5. Plataforma de Crowdfunding

Crear plataforma para financiar proyectos mediante aportes de usuarios interesados. La plataforma debe contar con lo siguiente:

- Los usuarios pueden crear proyectos, indicando nombre, descripción, monto objetivo, fecha límite de inicio, fecha límite de finalización (cuando el proyecto debería estar terminado), categoría y recompensas.
- Otros usuarios pueden realizar aportes económicos a los proyectos, registrando el monto y el método de pago.
- Cada proyecto debe mostrar el porcentaje de financiamiento alcanzado y la cantidad de contribuciones.
- Si el proyecto no alcanza el monto objetivo antes de la fecha límite, el proyecto debe marcarse como cancelado.
- Si el proyecto alcanza el monto objetivo antes de la fecha límite de inicio, comienza el período de ejecución del proyecto, que dura hasta la fecha límite de finalización. Luego de la fecha de finalización, los contribuyentes pueden ingresar evaluaciones del proyecto, indicando una puntuación y un comentario. Estas evaluaciones deben ser visibles para todos los usuarios, y debe calcularse un promedio de las evaluaciones.
- En el perfil de cada usuario debe mostrarse la cantidad de proyectos que ha creado y la cantidad de proyectos en los que ha contribuido.