**POLIMET**

Fueron contratados por el enigmático museo de “Polimet” que esta buscando modernizarse un poco para adaptarse a la sociedad actual. Les cuentan que para su estilo, quieren mezclar lo clásico de las obras de arte con elementos de la vida moderna. Quieren plantear la pregunta de... que pasaría si los artistas hubiesen pintado en nuestra epoca?

La búsqueda de esta modernización de identidad es que las nuevas generaciones se interesen por conocer la gran cultura del arte y la historia por la que tranasitó. Que ocurriría si las diferentes vanguardias chocaran con los memes, los emojis o los trends de tiktok?   
  
Funcionalidades:

* Home con descripción del museo, horarios, fotos y eventos que se quieran promocionar
* Loguearse para poder sacar tickets (indicando dia, hora, cant personas) y también poder tener una membresía
* Mostrar qué hay en cada piso (tener en cuenta que hay zonas bloqueadas segun la membresia)
* Sección para aprender sobre las obras y que se pueda filtrar por: ubicacion en el museo, epoca, nombre, nivel de popularidad
* Una vez que se realiza la visita, votar aquellas figuras que mas le gustaron (Asi se actualiza el nivel de popularidad de cada obra)
* Ver nuevos eventos disponibles que hay en el museo (ejemplo: noche en el museo, excursión de experiencias paranormales, etc)

1. Con esta información se le pide al equipo:
   1. Realizar un breve benchmarking analizando competencias. Analizar cómo es la marca de la competencia (que paleta de colores, tipografías y tipo de logo), que ofrecen en sus sitios web (estructura, funcionalidades, imágenes, recursos gráficos, etc).
   2. Realizar el branding para “PoliVehiculos” y construir su manual de marca.
   3. Realizar wireframes del sitio web en figma. Deberá realizarse para Desktop, tablet y mobile. Luego, realizar testeos con 5 usuarios de sus wireframes, sacar conclusiones y realizar un breve informe respecto a los resultados y los cambios a realizarse.
   4. Realizar los mockups del sitio web en figma en las 3 pantallas. Realizar nuevamente 5 testeos, sacar conclusiones y realizar un breve informe respecto a los resultados y los cambios a realizarse.
2. Diseñar un diagrama entidad-relación físico que represente de forma adecuada el sistema y crear la base de datos. Realizar allí (en un .sql) los siguientes triggers, eventos y procedimientos:
   1. Trigger que, al calificarse una obra, actualice su nivel de popularidad.
   2. Trigger que, al realizarse una reserva, chequee la disponibilidad para la fecha elegida.
   3. Procedimiento que, dado un rango de fechas, liste un top 10 de las obras más valoradas. Mostrar el nombre de la obra y el puntaje de la misma.
   4. Procedimiento que liste el promedio de las valoraciones a lo largo del tiempo por tipo de membresía.
   5. Evento que, todos los días a la medianoche invalide los tickets cuya fecha ya pasó y no fueron utilizados, deben pasar a un estado de “vencidos”.
   6. Evento que, todos los días a las 6:00am, coloque en una tabla de notificaciones, los recordatorios para las personas que tienen reservas para ese día.

Aclaración: Deben crear los procedimientos y funciones que se crean necesarios para tener un código limpio, mantenible y reutilizable.

1. Además de modelar la base de datos y sus respectivas reglas con SQL, se solicita desarrollar una **API REST en Java usando Spring Boot**, que permita gestionar las entidades del sistema de forma dinámica y probar su funcionamiento desde el frontend (pueden hacer las pruebas iniciales utilizando herramientas como Postman) Se pide desarrollar las siguientes rutas de la API:
2. Registro de usuarios

* **Ruta:** POST /api/usuarios
* **Función:** Registrar un nuevo usuario con su perfil y nivel de membresía.

1. Consulta del contenido del museo

* **Ruta**: GET /api/home
* **Función**: Muestra la descripción del museo, horarios, fotos y eventos promocionados.

1. Consultar qué hay en cada piso

* **Ruta**: GET /api/pisos
* **Función**: Devuelve los pisos del museo, salas incluidas y las restricciones por tipo de membresía.

1. Ver eventos especiales del museo

* **Ruta**: GET /api/eventos
* **Función**: Muestra la lista de próximos eventos como “Noche en el museo”, excursiones temáticas, etc.

1. Sacar ticket para visita

* **Ruta**: POST /api/tickets/reservar
* **Función**: Permite al usuario reservar tickets indicando día, hora y cantidad de personas. Verifica disponibilidad.

1. Obtener top 10 obras más valoradas en fechas dadas

* **Ruta**: GET /api/estadisticas/top-obras?desde=YYYY-MM-DD&hasta=YYYY-MM-DD
* **Función**: Devuelve el top 10 de obras mejor calificadas en el rango de fechas dado.

1. Filtrar obras

* **Ruta**: GET /api/obras?ubicacion=x&epoca=y&nombre=z&popularidad=alta
* **Función**: Lista las obras según filtros combinables: ubicación, época, nombre o nivel de popularidad.

1. Votar obras tras la visita

* **Ruta**: POST /api/obras/{id}/votar
* **Función**: Permite a los usuarios calificar una obra que vieron durante su visita. Actualiza el nivel de popularidad.

Aclaración: además se deben crear todos los endpoints que sean necesarios desde el front-end para cumplir con las funcionalidades pedidas.