

Trabajo Práctico (parte 3)

Este trabajo tiene como objetivo afianzar la práctica de bases de datos y su interacción con Java.

Turismo en la Tierra Media

1. Planteo

La secretaría de turismo de la Tierra Media ha decidido crear un sistema para promocionar el turismo en su territorio. Éste contará con la información de las distintas atracciones de toda la Tierra Media.

El sistema deberá ser capaz de sugerir visitas a partir de la ubicación de los visitantes y también de generar itinerarios a partir de la información de preferencias disponible en el perfil de los usuarios. Se espera que los usuarios puedan descargarse una aplicación móvil que les permita interactuar con el sistema y los vaya guiando en su recorrido.

Dado el alcance del sistema y la limitación de tiempo, varios equipos han sido contratados para el desarrollo de este sistema.

En el caso de su equipo, ya se dispone de la lógica para sugerir visitas e itinerarios, utilizando bases de datos. Sin embargo el equipo encargado del incremento anterior no lo realizó utilizando una interacción dinámica, típica de la Web. En este incremento, nos centraremos en darle una interfaz amigable al sistema.

Su tarea es:

1. Crear pantallas de Alta, Baja (lógica)^[1], Listado y Modificación de **Atracciones, Tipos de Atracción, Promociones, Usuarios**. Estas pantallas sólo podrán ser accedidas por un usuario **administrador** (atributo `is_admin` en `true/1`). **No habrá registración de usuario**.
2. **Banner/carrusel/slider/listado publicitario** con ofertas de atracciones y promociones, ordenado según los criterios ya sabidos (primero promociones que le gustan, luego atracciones que le gustan, luego promociones que no, luego atracciones que no, siempre de mayor precio a menor precio y de mayor duración a menor duración), mostrando sólo lo que puede comprarse. Estas pantallas deben ser accedidas por cualquier usuario **no administrador**.
3. **Listado de atracciones y promociones**.
4. **Pantalla de detalle de la atracción-promoción**. En esta pantalla se muestran los detalles de lo que se está por comprar. Posee el botón "comprar". Cuando se elige una atracción o promoción, se agrega al itinerario y se descuenta automáticamente del dinero disponible. Esta pantalla es para usuarios **no administradores**.
5. **No se puede borrar o modificar una reserva** una vez realizada.
6. Las validaciones deben realizarse tanto del lado del **cliente como del servidor**. Ejemplo: si una atracción no tiene cupo, puede listarse pero no permitir comprarse deshabilitando el botón de compra. En el servidor, se verificará el cupo nuevamente en caso de que haya sido vendido en el lapso en que el usuario tomaba la decisión.

7. Pantalla de itinerario, donde cada usuario puede ver lo que ha comprado. **Esta pantalla es para usuarios no administradores. No es un carrito, porque ya pagué.**
8. BONUS: para puntos extra: pantalla disponible para **administradores solamente**, que muestre qué compras realizó cada usuario.
9. BONUS: Confeccionar [historias de usuario](#) y [criterios de aceptación](#) en [formato GWT](#) para todas las funcionalidades del sistema.

Tecnologías a utilizar:

1. Bootstrap 5 / Bulma / Otra / Ninguna
2. JQuery / Javascript
3. Datatables / Tablesorter
4. DAO
5. MVC

Nota: El sistema se piensa para utilizarse en una única fecha (en el futuro). Por lo tanto el cupo de las atracciones sólo se considera para esa fecha y no hay que renovarlo para otra ocasión.

Sigue siendo válido:

1. Si un usuario vuelve a ingresar, se le continuarán sugiriendo promociones y atracciones según el criterio inicial, pero respetando las nuevas restricciones (el tiempo que le queda, el dinero que le queda, y el cupo restante en la atracción).
2. No deberá permitirse la compra de atracciones duplicadas, por lo que no puede ofrecerse una promoción que contenga atracciones ya compradas. Podría observarlas pero no puedo comprarlas.

[1] Una baja lógica es un borrado por el cual no está disponible para operar, pero el registro sigue existiendo en la base de datos.