

Examen práctico de conocimiento en tecnologías Java Spring boot

La siguiente prueba tiene como objetivo soportar las necesidades de una tienda de alquiler de libros.

1. Registro de clientes
2. Autenticación de clientes (Login)
3. Permitir que una vez el cliente ingrese pueda reservar un libro.
4. Permitir que un administrador pueda agregar inventario a los libros (es decir si la tienda tiene 10 libros de una referencia, el administrador podrá subir o bajar este número).
5. Permitir configurar las tarifas de alquiler (las tarifas de alquiler depende del libro)
6. Permitir listar los libros, debe visualizarse nombre del libro, autor, categoría, cantidad disponible, cantidad reservada.

Aspectos funcionales.

- Cuando el cliente quiera realizar la reserva de un libro la aplicación debe solicitarle los siguientes datos:
 - Fecha de devolución
 - Cantidad de libros (debe validar que la cantidad no supere la cantidad disponible)
 - Se debe mostrar la tarifa a pagar según los datos ingresados por el cliente.
- Debe existir una vista donde se desplieguen los libros disponibles para el cliente. Desde esta lista el cliente eligiendo el libro puede ir a la pantalla de reserva.
- Debe existir una vista administrar que permita modificar el inventario de los libros.
- Debe existir una vista que permita agregar nuevos libros.
- Debe existir una vista que permita eliminar libros
- Debe existir una vista para administrar las tarifas.

Características de los usuarios:

- Administrador:
 - Tendrá permisos de gestionar libros (crear, eliminar, modificar).
 - Tendrá permisos de modificar y crear tarifas de alquiler.
 - El administrador no se registra dentro de la aplicación
 - Puede hacer login dentro de la aplicación.
- Cliente:
 - Tendrá permisos de ver los libros disponibles para alquiler.
 - Tendrá permisos para reservar (alquilar) libros.

- Puede hacer registro y login dentro de la aplicación.

Nota: solo debe existir un administrador, puede crearlo directamente en la BD.

Aspectos técnicos

Debe construir lo siguiente:

- Modelo de base de datos autogenerado, con datos precargados, puede utilizar ORMs para este proceso.
- Aplicación backend en tecnología Java con Spring boot
- Puede usar la tecnología de su preferencia para FrontEnd.
- El backend debe exponer un API Rest con el Crud de base de datos.
- El frontEnd debe comunicarse con el BackEnd por medio de Servicios.
- La aplicación FrontEnd debe tener un aspecto prudente (organizado limpio), no es necesario un diseño que demande mucho tiempo.

Observaciones:

- Puede manejar el tema de los permisos de usuarios con una columna en la tabla Usuarios que indique el tipo de usuario: Ejemplo: **tipoUsuario** que guardará los posibles valores “Cliente” y “Administrador”.
- Puede gestionar el tema los permisos e información de usuarios usando la **Session**. No es necesario utilizar frameworks para gestionar la seguridad.
- Usar tecnologías como jquery, Ajax, Bootstrap, Razor, Entity no es obligatorio, pero genera un plus.
- Encapsular el backend en un contenedor Docker no es obligatorio, pero genera un plus
- Puede distribuir las vistas y funcionalidades como desee, lo importante es que cumplan con el objetivo funcional.
- Puede usar internet para investigación, documentación y consulta de ejemplos.

Entregables:

- Manual instructivo de despliegue y configuración de la aplicación (lo necesario para iniciar la aplicación).
- Archivo de despliegue
- Adjuntar todo el paquete de despliegue completo con librerías.
- Código fuente. (indicar IDE donde se desarrolló)