

PROYECTO FINAL

Información general

Docente:	Carlos Andrés Florez Villarraga
-----------------	---------------------------------

Proyecto Final del espacio académico

UniEventos es una empresa dedicada a la venta de entradas para conciertos y eventos en diversas ciudades de Colombia. La empresa necesita desarrollar una plataforma que permita a sus clientes adquirir entradas para los eventos de su interés.

Para utilizar esta plataforma, los clientes deben registrarse proporcionando su cédula, nombre completo, dirección de residencia, número de teléfono, correo electrónico y contraseña. Una vez registrados, los clientes pueden acceder a la plataforma utilizando su correo electrónico y contraseña.

Dentro de la plataforma, los usuarios pueden seleccionar el evento de su interés y proceder con la compra de entradas. Para completar la compra, el cliente debe especificar la cantidad de entradas que desea adquirir, seleccionar la localidad y finalmente pagar el valor total de la compra.

La plataforma debe contar con un sistema de cupones que ofrecen descuentos al cliente en el valor de la compra, estos cupones son redimibles al momento de pagar. Los cupones se distribuyen de la siguiente manera:

- Cuando un cliente se registra por primera vez se le envía un cupón vía correo electrónico. Al redimir este cupón obtiene un 15% de descuento.
- Cuando un cliente hace una compra por primera vez por medio de la plataforma se le envía un cupón vía correo electrónico. Al redimir este cupón obtiene un 10% de descuento.

El administrador de la plataforma tiene la responsabilidad de crear los eventos. Cada evento tiene un nombre, una dirección, ciudad, una descripción, un tipo (concierto, teatro, deporte, festival, otro) dos imágenes (póster del evento y distribución de localidades), fecha, y unas localidades, cada localidad tiene un nombre, precio y una capacidad máxima.

NOTA: En adelante la acción gestionar hará referencia a crear, modificar, buscar (ver información), listar y eliminar.

En **UniEventos** se desea contar con una aplicación que maneje dos tipos de usuarios.

Administrador:

- Loguearse.
- Gestionar eventos.
- Crear nuevos cupones para fechas o eventos especiales con el porcentaje que desee. Cada cupón debe incluir un código, un nombre descriptivo, un porcentaje de descuento, y una fecha de vencimiento. Existen dos tipos de cupones: Cupón de código único (un código único es compartido por todos los usuarios) y Cupón de código individual (un código generado aleatoriamente para cada usuario).
- Generar reportes a partir de las compras hechas por los clientes a los diferentes eventos que están creados. Estos reportes deben tener datos estadísticos de cada evento: porcentaje vendido por localidad, total ganado por ventas, entre otros. Los reportes se deben mostrar en la web y en PDF.
- Gestionar su propia cuenta (editar, eliminar).
- Recuperar contraseña si la olvida.

Cliente:

- Registrarse y loguearse.
- Filtrar eventos (por nombre, tipo y/o ciudad).
- Crear orden de compra o cancelarla.
- Carrito de compras: aquí pueden revisar sus selecciones antes de proceder al pago. Este carrito debe conservar los elementos del usuario, incluso si cierra sesión o cambia de dispositivo.
- Redimir cupones al momento de hacer la compra.
- Listar sus propias compras (historial compras).
- Gestionar su propia cuenta (editar, eliminar).
- Recuperar contraseña si la olvida.

Para tener en cuenta:

- Diseñar diagrama de clases para representar cada entidad del dominio de los datos del sistema.
- El proyecto se debe implementar usando Spring Boot en el backend y Angular en el frontend. Así como MongoDB para la gestión de los datos.
- Cuando el usuario se registra no puede iniciar sesión inmediatamente. Una vez el usuario se registra se le envía un correo electrónico con un código. Y cuando ingrese la primera vez se le pedirá este código para activar su cuenta. Este código debe tener una validez de 15 minutos.
- Una vez registrado, los usuarios al olvidar su contraseña pueden cambiarla por medio de un código enviado a su correo electrónico. Este código debe tener una validez de 15 minutos.
- En el frontend debe haber una página web de inicio con los eventos del momento según la ubicación del usuario, así como un componente de búsqueda.
- Se debe validar automáticamente que al momento de hacer la compra, la localidad seleccionada para el evento tenga la capacidad adecuada para alojar a la cantidad de personas especificada en la compra.
- En la orden de compra se debe guardar el código de la orden, el del cliente, el ítem (código del evento, la localidad, unidades, precio individual), el cupón (si es que usa un cupón), el total, una fecha y un pago (este pago se construye con lo que devuelve la pasarela de pagos). Tenga en cuenta que la compra solo puede realizarse hasta dos días antes del evento.
- Cada que se realice una compra se debe generar un código QR con el código de la orden y se le debe enviar un correo electrónico al cliente indicando los detalles de la orden de compra incluyendo el QR y la información del pago.
- Se debe implementar una pasarela de pago **en modo prueba** para que los clientes puedan realizar los pagos de sus entradas. Para esto usaremos MercadoPago, pero hay más alternativas válidas: PayU, Stripe, ePayco, Wompi, PayPal, etc.
- Para el manejo de imágenes se debe hacer uso de un servicio externo, puede ser Cloudinary, Firebase, Flickr, AWS S3, Google Cloud Storage, etc.
- El código fuente del proyecto debe estar en un repositorio de Github. Todos los integrantes del grupo deben contribuir en el desarrollo del proyecto.
- Los administradores están precargados en la base de datos.
- Las eliminaciones de registros se deben hacer por medio de un campo que indique si están activos o inactivos.

IMPORTANTE:

- Piense en al menos dos necesidades asociadas a este sistema para que las incluya dentro de su modelo. Estas necesidades no pueden ser triviales como agregar un simple campo, sino que deben representar algo relevante en el sistema (backend y frontend).
- El proyecto puede realizarse en grupos de máximo 3 personas.