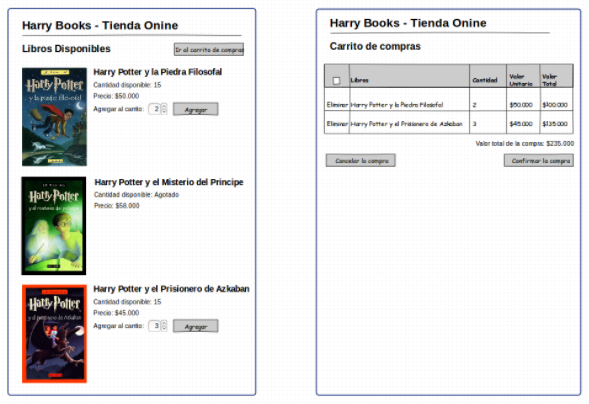
**TIENDA EL LÍNEA DE LIBROS DE HARRY POTTER.**

**Descripción del reto.**

Harry Books es una librería especializada en vender únicamente los libros de la serie de Harry Potter y lo han contratado a usted para crear su tienda en línea. Para ello, le han entregad los mockups de la figura 1 y le han dicho que la tienda en línea debe ofrecer las siguientes funcionalidades:



Ver la lista de libros disponibles:

1. Es la primera página y consiste de un listado de los libros de la serie, con su título, precio, cantidad disponible y opcionalmente, una imagen que identifique al libro.
2. Cada libro va acompañado de un control que permita indicar la cantidad e unidades que quiere comprar y agregar esas unidades de ese libro al carrito de compras.
3. En alguna parte de esta página debe haber un control que permita ir a revisar el carrito de compras.
4. Si un libro tiene cero unidades disponibles, debe mostrar el texto “Agotado”, en lugar del número cero.

Carrito de compras:

1. Es la página en donde el usuario puede ver una lista con los libros que ha seleccionado en la página “lista de libros disponibles”. Cada elemento de la lista debe tener el nombre del libro seleccionado, la cantidad indicada en la página anterior, el precio unitario, el precio total de esas unidades y el precio total de la compra, es decir la suma de los precios totales de cada libro, así:
2. En esta pantalla el usuario sólo puede cambiar la cantidad de unidades que va a comprar de los libros que ya estén en el listado.
3. También podrá eliminar un libro del carrito de compras. El usuario no puede eliminar libros del listado de libros disponibles.
4. En cualquiera de los cambios autorizados (cantidad de unidades o eliminar libro del carrito), se deben actualizar los precios totales por libro y total por la compra.
5. Esta página debe tener un botón o cualquier control para cancelar la compra. Cuando este control sea activado o pulsado, se solicita confirmación para cancelar la compra y e n caso de que el usuario acepte la solicitud de confirmación, se eliminan todos los libros del carrito de compras y queda en la misma página mostrando el mensaje “Su compra se ha cancelado”.
6. Esta página contiene un botón que permite confirmar la compra. Cuando este botón se pulsa, se debe mostrar algún tipo de solicitud de confirmación, bien sea un popup, alert, nueva página, nuevo botón o cualquier otro tipo de medio para solicitar una doble confirmación de la compra.
7. En caso de cancelar la doble confirmación, debe quedar en la página del carrito de compras.
8. En caso de aceptar la doble confirmación, el sistema debe volver a la página de listado de libros disponibles y allí mostrar el mensaje: “Hemos recibido su solicitud y será procesada por nuestros agentes. Gracias por su compra”. Una vez regrese a la página del listado de libros disponibles se deben mostrar las cantidades actualizadas de libros disponibles es decir, si al ingresar al sistema el usuario vio 10 unidades de libro 1 y confirmó la compra de 2 unidades, ahora el listado debería mostrar la cantidad de 8 unidades disponibles.
9. Dado que el sistema debe guardar en la base de datos la cantidad de libros disponibles, después de la compra, la base de datos debe quedar actualizada descontando la cantidad de libros cuya compra se confirmó.

**Parámetros**

**Tipo de tecnología:** Web

**Lenguaje de programación:** PHP (Se valorará el uso de un framework especialmente LARAVEL)

**Dificultad:** Principiante/Intermedio

**Tiempo máximo a utilizar:** 8 horas

**Resultados mínimos esperados:**

1. Construcción de un sitio web transaccional que utiliza el modelo de carrito de compras.
2. La aplicación debe utilizar una base de datos para almacenar la información básica de los productos y de las compras del usuario.
3. La aplicación puede estar construida para un único usuario, pero que sea multiusuario es un valor agregado importante porque permite analizar el uso de sesiones web. En caso de ser multiusuario, debe usar la base de datos para almacenar la información del usuario, mínimo: id y nombre.
4. Los datos de los libros, incluyendo la cantidad de libros disponibles, deben conservarse entre reinicios del servidor.
5. Cumplimiento de los retos complementarios que hayan sido indicados.

**Retos complementarios – Nivel principiante.**

Las siguientes actividades no son requeridas para determinar el reto como completado, sin embargo, si se realizan, darán puntos adicionales en la calificación.

1. Registrar la información de la compra en la base de datos, para poder usarla en informes de ventas.
2. Que la aplicación sea multiusuario utilizando sesiones web individuales por cada usuario.
3. Login con id y contraseña.
4. Pruebas unitarias automatizadas.
5. Pruebas automatizadas de aceptación.
6. Agregar módulo de administración que permita ver reportes de ventas de múltiples usuarios consolidados.

**Retos complementarios – Nivel intermedio.**

Las siguientes actividades no son requeridas para determinar el reto como completado, sólo son requeridas si está aplicando a un nivel intermedio o senior y darán puntos adicionales en la calificación:

1. Utilice patrones de diseño y de arquitectura para diseñas y construir el sistema.
2. Elabore un diagrama de despliegue en el que indique como podría ser la arquitectura del sistema en un ambiente de producción para soportar una alta transaccionalidad (miles de transacciones por minuto).
3. ¿Cuáles tecnologías utilizaría para mejorar la solución y por qué?.
4. Elabore un diagrama de componentes en el que ilustre el diseño que aplicó al sistema. ¿Qué cree que se podría mejorar a su diseño?.
5. ¿Cuáles partes de la solución son reutilizables? Porqué considera que su solución es reutilizable.
6. ¿Cuáles partes de la solución son parametrizables? Porqué considera que su solución es parametrizable.
7. Ahora usted es el dueño del producto. ¿Cuáles características cree que podría agregarse al sistema? Cuales implicaciones podrían tener sobre la arquitectura actual que ha planteado.