Trabajo Final Integrador

Taller de Tecnologías de Producción de Software - Ruby

Cursada 2024

En este Trabajo Final Integrador, buscaremos que apliques los conocimientos adquiridos en la materia para resolver un problema real, basado en un caso hipotético. El objetivo de la implementación que deberás hacer es tener un sistema que permita resolver las necesidades expuestas en este documento.

Contexto

Estás a cargo del desarrollo de una aplicación de gestión de inventario de indumentaria para una reconocida cadena de ropa deportiva. La aplicación deberá permitir al personal de la cadena administrar el stock de productos, realizar ventas y tener una página pública que permita a los potenciales clientes ver los productos que la tienda tiene en stock.

En este documento se describirán los puntos principales y la funcionalidad mínima que la aplicación debe proveer, a la cual podrás agregar soporte para más acciones o interacciones, siempre respetando lo especificado aquí como base.

Requisitos técnicos

El desarrollo debe realizarse utilizando el siguiente stack tecnológico:

- La versión estable más reciente del framework Ruby on Rails (`8.0.0` al momento de escribir este documento).
- Una versión reciente de Ruby (`3.2.0` o superior), acorde a lo requerido por el framework.
- Una base de datos SQLite para dar soporte de persistencia.

Adicionalmente, se te pide que no reinventes la rueda y cuando sea posible utilices gemas bien establecidas que provean funcionalidad habitual en una aplicación web, como puede ser el caso de la autenticación.

Alcance

Deberás desarrollar una aplicación web para gestionar el inventario de una cadena de ropa llamada *Avivas*, que brindará la siguiente funcionalidad:

- *Storefront*: un sitio web público que permita a los visitantes ver los productos disponibles y consultar su precio. No es requerido que además puedan realizarse compras, pero sí que se pueda ver el stock disponible. Ver apartado **Storefront**.

- *Administración de productos*: una interfaz de administración que permita a los empleados de la cadena gestionar los productos, su stock y su precio. Ver apartado **Administración de productos**.
- *Administración de ventas*: una interfaz de administración que permita a los empleados de la cadena registrar las ventas realizadas y los productos vendidos. Ver apartado **Administración de ventas**.
- *Gestión de usuarios*: un sistema de autenticación y autorización que permita a los empleados de la cadena acceder a las interfaces de administración. Ver apartado **Gestión de usuarios**.

Storefront

El *Storefront* es la cara visible de la cadena de ropa, y deberá permitir a los visitantes ver los productos disponibles, su precio y su stock. No es necesario que los visitantes puedan realizar compras, pero sí que puedan ver la información de los productos.

Esta interfaz deberá ser accesible por cualquier visitante, sin necesidad de autenticación. Deberá mostrar una lista de productos, con sus imágenes, nombre, descripción, precio y stock disponible. Además, deberá permitir navegar por las distintas categorías de productos y realizar búsquedas por nombre o descripción.

Administración de productos

La aplicación deberá permitir a los empleados de la cadena gestionar los productos, su stock y su precio. Para ello, se deberá implementar una interfaz de administración que permita realizar las siguientes acciones:

- **Listar productos**: presenta una lista de todos los productos existentes.
- **Agregar producto**: permitir la creación de un nuevo producto.
- **Modificar producto**: permitir la modificación de un producto existente.
- **Eliminar producto**: permitir la eliminación de un producto existente. El borrado debe ser lógico, y poner el stock del producto automáticamente en 0.
- **Cambiar stock**: permitir modificar el stock de un producto existente.

Los productos deberán tener, como mínimo, los siguientes atributos: * Nombre * Descripción * Precio unitario * Stock disponible * Imágenes (al menos una) * Categoría * Talle (opcional) * Color (opcional) * Fecha de ingreso al inventario * Fecha de última modificación * Fecha de baja (si fue eliminado)

Administración de ventas

El sistema deberá permitir a los empleados de la cadena registrar las ventas realizadas y los productos vendidos. Para ello, se deberá implementar una interfaz de administración que permita realizar las siguientes acciones:

- **Listar ventas**: presenta una lista de todas las ventas realizadas.
- **Asentar venta**: permitir la creación de una nueva venta. Cada venta deberá tener una fecha y hora de realización, y deberá permitir agregar productos vendidos a la misma. Cada producto vendido deberá tener una cantidad y un precio de venta.
- **Cancelar venta**: permitir la cancelación de una venta realizada. Al cancelarse una, se deberá devolver el stock de los productos vendidos para que queden disponibles nuevamente. Las ventas canceladas no deberán ser eliminadas de la base de datos.

Notar que las ventas deberán afectar el stock de los productos, y no se podrá vender más stock del que se tiene disponible. Por lo tanto, al registrar una venta, se deberá verificar que el stock disponible sea suficiente para la cantidad de productos vendidos.

Las operaciones de venta deberán realizarse de manera transaccional para garantizar su atomicidad a nivel de base de datos. Esto significa que, si una venta no puede completarse por falta de stock, no deberá registrarse ninguna de las operaciones de venta en la base de datos.

Datos mínimos que debe contener una venta: * Fecha y hora de realización. * Productos vendidos, con cantidad y precio de venta. * Total de la venta. * Empleado que realizó la venta. * Cliente.

Gestión de usuarios

La aplicación deberá permitir que se creen usuarios con diferentes roles desde la interfaz de administración. Los usuarios autenticados son los únicos que pueden acceder a las interfaces de administración del sistema. El Storefront sigue siendo accesible para cualquier visitante, se haya autenticado o no.

Cada usuario deberá tener, como mínimo, los siguientes atributos:

- Nombre de usuario (debe ser único)
- Correo electrónico (debe ser único)
- Teléfono
- Contraseña
- Rol
- Fecha de ingreso a la cadena

Los usuarios podrán desactivarse, siendo únicamente un usuario administrador quien podrá desactivar otros usuarios. La desactivación de usuarios funciona como un borrado lógico, y una vez desactivado, su contraseña deberá cambiarse por un valor automático (y desconocido), y el usuario no podrá acceder al sistema mientras se encuentre desactivado.

Roles de usuario

Se deben implementar los siguientes roles de usuario:

- **Administrador**: tiene acceso a todas las funcionalidades de la aplicación.
- **Gerente**: tiene acceso a la administración de productos y ventas y puede gestionar usuarios, pero no puede crear ni modificar usuarios con el rol de *administrador*.
- **Empleado**: tiene acceso a la administración de productos y ventas, pero no puede gestionar usuarios.

Además de esto, todos los roles pueden cambiar los datos de su propia cuenta (excepto su rol).

Consideraciones generales

- El diseño del modelo de datos que armes tiene que considerar no solo los modelos explícitamente mencionados en el enunciado, sino también cualquier otro que sea necesario para cumplir con los requerimientos. Por ejemplo, podrías necesitar un modelo para representar las categorías de los productos, o para representar los colores disponibles. Esa decisión queda a tu criterio.
- Se espera que todos los modelos tengan las validaciones adecuadas para asegurar la integridad de los datos.
- Se deberá incluir un conjunto de datos de prueba para poder evaluar el sistema. Estos datos deberán generarse mediante el script `db/seeds.rb` del proyecto Ruby on Rails.
- Se espera que la interfaz de usuario sea cómoda, entendible, y lo más intuitiva posible, pero no se solicitará para la entrega ningún tipo de librería ni interfaz en particular.
- Se deberá documentar en el archivo `README.md` del raiz del proyecto cualquier decisión de diseño importante que se haya tomado, así como también se deberán dejar escritos los requisitos técnicos y los pasos (comandos) para hacer funcionar la aplicación.
- La funcionalidad solicitada en este enunciado es la mínima que debe implementarse de base para poder aprobar el trabajo. En caso de desearlo, podrás agregar más funcionalidad a la aplicación, siempre y cuando respetes lo solicitado en este documento. No se permitirá el intercambio de funcionalidades solicitadas por otras que no lo estén.

Modalidad de la entrega

El trabajo deberá realizarse de manera individual. Los trabajos entregados se compararán para encontrar posibles similitudes en su estructura que pudieran provenir de copias entre distintas entregas.

Esta entrega es de caracter obligatorio, y su aprobación (en cualquiera de sus instancias) es condición necesaria para aprobar la cursada de la materia.

Tu trabajo tiene que estar versionado en un repositorio Git de GitHub, cuya URL deberás entregarnos mediante una tarea dedicada que publicaremos en el curso de la plataforma de Cátedras (Moodle). En caso que el repositorio sea privado, vamos a necesitar que nos brindes acceso al mismo para poder descargarlo y evaluarlo. Si esta es tu decisión, por favor consultá antes de la fecha límite de la entrega con el Jefe de Trabajos Prácticos para saber a qué usuarios de GitHub deberás brindarles acceso a tu repositorio.

Las fechas límite de entrega del trabajo fueron publicadas en el curso de la plataforma de Cátedras. No se admitirán entregas fuera de término.