Historias de usuario

Ingeniería de software I

Kevin David Rodriguez Riveros

Frank Sebastian Pardo Amaya

Jorge Andrés Torres Leal

Departamento de Ingeniería de Sistemas e Industrial

Universidad Nacional de Colombia - Sede Bogotá

Facultad de Ingeniería

31 de Enero de 2025

Historias de Usuario

Historia de Usuario 1

HU de Frank: Gestión de Inventario

Como administrador, **quiero** poder actualizar la cantidad de productos en el inventario, **para** mantener un control preciso del stock.

Descripción conceptual

Módulo	*Gestión de Inventarios.*
Descripción de la(s) funcionalidad(es) requerida(s):	*El administrador actualizará la cantidad de productos en el inventario para tener un control del stock disponible.* • *Agregar productos a la base de datos con información (nombre, código, precio, fecha de vencimiento, entre otros).* • *Eliminar productos cuando ya no estén disponibles para no mostrar en la lista.*

Descripción técnica

Backend

Añadir producto al inventario.

URL	**Método**	**Código html**
api/inventory	PATCH	200 (éxito), 404 (si no se encuentra el producto), 500 (error del servidor)
Caso de uso técnico: Cuando el administrador realiza una actualización al inventario, el sistema debe aceptar los datos enviados, validar que el producto existe y responder con un mensaje de éxito o error según corresponda.		

Datos de entrada
{
"productId": 123,
"quantity": 50
}
	Datos de salida
	200:
	{
	"status": "success",
	"data": {
	"productId": 123,
	"quantity": 50
	}	

	}
	404:
	{
	"status": "error",
	"message": "Producto no encontrado."
	}

	500:
	{
	"status": "error",
	"message": "Error interno del servidor."
	}

Deshabilitar un producto del inventario.

URL	**Método**	**Código HTML**
/api/inventory/disable	PATCH	200 (éxito al deshabilitar), 404 (si no se encuentra el producto), 500 (error del servidor)
Caso de uso técnico: Permite al administrador deshabilitar productos, marcándolos como no disponibles para evitar que aparezcan en el catálogo.		

Datos de entrada
200:
{
"productId": 123,
}
	Datos de salida
	200:
	{
	"status": "success",
	"message": "Producto deshabilitado correctamente",
	"data": {
	"productId": 123,
	"status": "disabled"
	}
	}	

	404:
	{
	"status": "error",
	"message": "Producto no encontrado."
	}	

	500:
	{
	"status": "error",
	"message": "Error interno del servidor."
	}	

FRONTEND

INTERACCIÓN ESPERADA:

- El administrador accede a la sección de inventario.
- Selecciona un producto de la lista y actualiza la cantidad disponible mediante un formulario.
- Al guardar los cambios, se muestra un mensaje de confirmación en pantalla.

MOCKUP/PROTOTIPO:

FLUJO VISUAL Y EVENTOS:

- Al hacer clic en "Editar", se abre un modal con los detalles del producto.
- El administrador introduce la nueva cantidad y confirma.
- Se muestra un spinner mientras se procesa la solicitud.
- Si es exitoso, se actualiza la tabla y se muestra un mensaje "Cantidad actualizada".
- En caso de error, se muestra un mensaje "Error al actualizar".

NOTAS ADICIONALES:

- La funcionalidad debe estar protegida por autenticación para evitar accesos no autorizados.
- Se deben manejar validaciones, como que la cantidad no sea negativa.

Historia de Usuario 2

HU de Kevin: Realización de Pedidos

Como usuario (Administrador o Empleado), **quiero** iniciar sesión con mi nombre de usuario y contraseña, **para** acceder al sistema según mi rol y gestionar las funciones correspondientes.

Descripción conceptual

Módulo	*Módulo de Gestión de Pedidos..*
Descripción de la(s) funcionalidad(es) requerida(s):	*El cliente podrá seleccionar productos desde un catálogo en línea y realizar un pedido. Esto incluye:* ● *Explorar productos categorizados con información relevante (nombre, precio, disponibilidad, descripción).* ● *Añadir uno o varios productos a un "carrito" temporal.* ● *Completar un formulario sencillo para registrar los datos necesarios (nombre, contacto, dirección).*

	• *Confirmar el pedido, generando un registro asociado al cliente y actualizando el inventario del sistema.*

Descripción técnica

Backend

URL	**Método**	**Código html**
/api/products	GET	200
Caso de uso técnico: Al consultar, debe retornar un código 200 y un array con los datos de los productos disponibles en el catálogo.		

Datos de entrada	**Datos de salida**
	
	{
	"status": "success",
	"data": [
	{
	"id": 1,
	"nombre": "Pulsera artesanal",
	"precio": 15000,
	"stock": 10,
	"descripcion": "Pulsera tejida a mano",
	"imagen_url":
	"https://example.com/image1.jpg"
	}
	]
	}	

Añadir producto al carrito

URL	**Método**	**Código HTML**
/api/cart	POST	200, 422
Caso de uso técnico: Permite al cliente agregar un producto al carrito de compras.		
Datos de entrada **200:** { "producto_id": 1,	**Datos de salida** **200:** { "status": "success",	

"cantidad": 2
}
	"message": "Producto agregado al carrito",
	"data": {
	"carrito_id": 123,
	"producto_id": 1,
	"cantidad": 2
	}
	}

422:
{
"producto_id": 1,
"cantidad": 2
}
	422:
	{
	"status": "error",
	"message": "Datos inválidos"
	}

Realizar Pedido

URL	**Método**	**Código html**
/api/orders	POST	201,422
Caso de uso técnico: Registra el pedido del cliente y actualiza el inventario.		

Datos de entrada
201:
{
"cliente": {
"nombre": "Juan Pérez",
"contacto": "+573001234567",
"direccion": "Carrera 123 #45-67, Bogotá"
},
"productos": [
{
"producto_id": 1,
"cantidad": 2
}
]
}
	Datos de salida
	201:
	{
	"status": "success",
	"message": "Pedido realizado con éxito",
	"data": {
	"pedido_id": 456,
	"estado": "Pendiente",
	"total": 30000,
	"fecha": "2024-12-21"
	}
	}	

422:
{
"cliente": {
"nombre": "Juan Pérez",
"contacto": "+573001234567",
"direccion": "Carrera 123 #45-67, Bogotá"
},
"productos": [
{
"producto_id": 1,
"cantidad": 2
	422:
	{
	"status": "error",
	"message": "Datos inválidos"
	}	

}] }	

FRONTEND

INTERACCIÓN ESPERADA

El cliente interactuará con el módulo de pedidos a través de un catálogo visual en la página web. El flujo funcional incluye:

Explorar productos: Los productos se muestran en un diseño de cuadrícula, cada uno con una tarjeta que incluye nombre, precio, descripción breve, imagen y botón "Añadir al carrito".

El cliente puede hacer clic en una tarjeta para obtener más detalles del producto.

Gestión del carrito: Al añadir un producto al carrito, se actualiza automáticamente un contador visible en la parte superior de la página ("Carrito: X productos").

El cliente puede hacer clic en el ícono del carrito para revisar los productos seleccionados y realizar ajustes (cantidad o eliminar).

Formulario de pedido: Al confirmar el carrito, se muestra un formulario emergente o una nueva página donde el cliente ingresa sus datos personales (nombre, contacto, dirección) y selecciona el método de pago.

Confirmación: Al completar el formulario, aparece un mensaje de éxito indicando que el pedido fue registrado, junto con el ID del pedido para futuras referencias.

Mockup:

FLUJO VISUAL Y EVENTOS

-Al cargar la página, el cliente ve el catálogo.

-Al añadir un producto, se actualiza el contador del carrito y aparece un mensaje breve.

-Al hacer clic en el carrito, se redirige a la página del carrito donde el cliente puede revisar y ajustar su pedido.

-Al confirmar, se muestra el formulario de pedido.

-Al enviar el formulario:

*Si es exitoso, el cliente ve el mensaje de confirmación.

*Si hay un error (producto fuera de stock, campos incompletos), aparece un mensaje de error contextual.

NOTAS ADICIONALES

Tipografía y estilo: Utilizar fuentes como Montserrat o Poppins, colores neutros y cálidos para alinearse con la estética artesanal del negocio.

Responsive design: Asegurarse de que la vista se ajuste perfectamente a dispositivos móviles y escritorio.

Flujo claro y directo: Limitar las acciones a un máximo de 2-3 clics para completar un pedido.

Eventos visuales clave que podrían mejorar la interacción: -Spinner al cargar el carrito o procesar el pedido.

-Animación al añadir un producto al carrito para mejorar la experiencia del usuario.

Historia de Usuario 3

HU de Jorge: Notificaciones de Pedido

Como administrador, **quiero** recibir alertas cuando un producto tenga bajo stock, **para** reponer el inventario a tiempo.

Descripción conceptual

Módulo	*Módulo de notificaciones*
Descripción de la(s) funcionalidad(es) requerida(s):	*El sistema envía una notificación al administrador cuando el stock de algún producto está bajo*

Descripción técnica

Backend

Primero se crea un código en el backend que revise al final de cada día tres días cuánto stock queda de cada producto y con ciertos disparadores y límites razonables se envía una señal de notificación al administrador.

URL	**Método**	**Código html**
localhost:8080/users	GET	200
Caso de uso técnico Al consultar, debe retornar código 200 y en data un array con los datos del stock		
Datos de entrada	**Datos de salida**	

	{
	"status": "success",
	"data": [{
	"id": x particular a cada producto,
	cantidad : x
	“nombre”: “nombre del producto”
	}]
	}

URL	**Método**	**Código html**
localhost:8080/users/1	GET	200
Caso de uso técnico Al consultar, debe retornar código 200 y en data un array con los datos de las alertas de stock		

Datos de entrada	**Datos de salida**
	200:
	{
	"status": "success",
	"data": [{
	"id": x particular a cada producto,
	cantidad : x de producto en el cual se debe enviar la alerta
	“nombre”: “nombre del producto”
	}]
	}	

Luego se comprueba por medio de un código de comparación si la cantidad en stock es menor a la cantidad designada como de alerta.

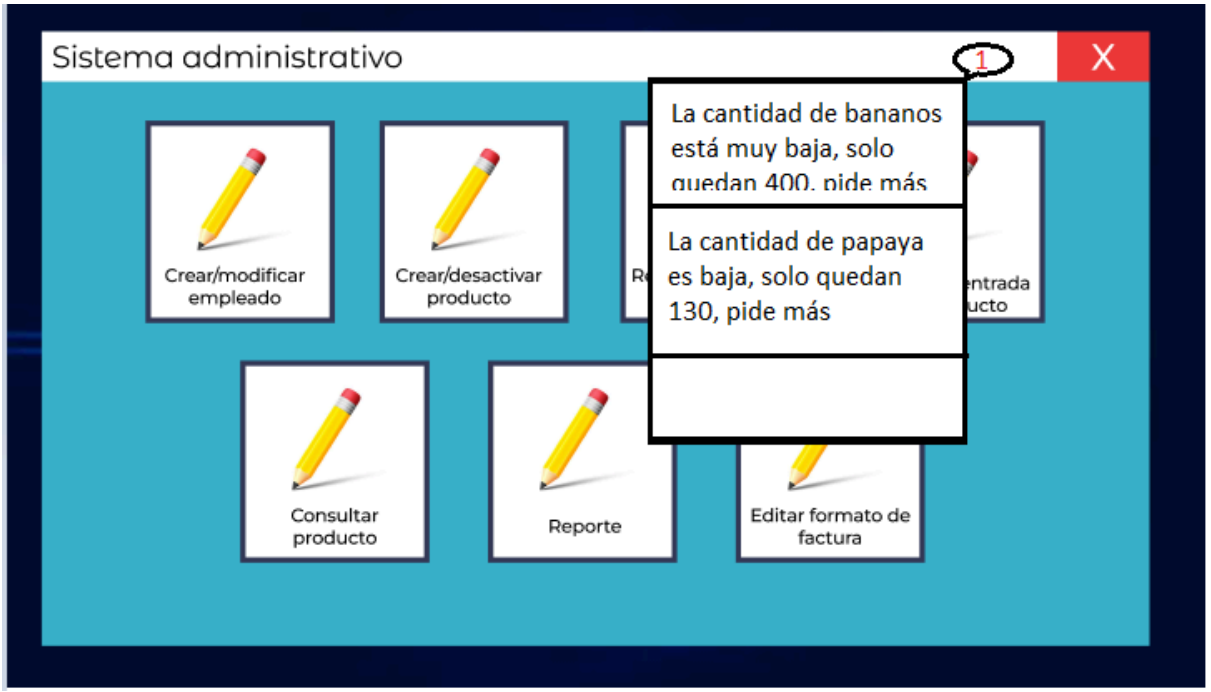
Frontend

En esta sección solo debe aparecer un pequeño aviso en una esquina que indique la alerta sobre el producto

Interacción esperada:

> *El usuario nota que el símbolo de alertas tiene un numerito en rojo que le indica que hay una alerta*

Mockups/Prototipos:



Flujo visual y eventos:

(Especificar el comportamiento esperado, por ejemplo, qué sucede al hacer clic en un botón, mostrar un spinner mientras carga, etc.)

1. *El administrador da click en el botón de alertas y se despliega un mini panel con las alertas de falta de stock*
2. *El administrador da click en esa alerta en particular para ver una descripción más detallada*
3. *El administrador procede a pedir más stock*