

## Modulo Tercero:

### Introducción

En esta sesión, asumirá el papel de un desarrollador full-stack, responsable de construir la interfaz de usuario front-end (UI) y la API back-end para el nuevo sistema de software de ventas del GSW General Supplies Warehouse.

El objetivo de esta sesión es traducir el diseño en una aplicación de escritorio funcional para el staff y una API sólida para gestionar los datos de productos y pedidos. Se evaluará su capacidad para crear interfaces de usuario intuitivas y visualmente atractivas, así como su destreza en el desarrollo de API seguras y eficientes.

Esta sesión está diseñada para evaluar sus habilidades en:

- **Desarrollo Front-End:** Diseño e implementación de interfaces de usuario fáciles de usar, visualmente atractivas y alineadas con la marca GSW General Supplies Warehouse.
- **Desarrollo del back-end:** Construcción de una RESTful API que se adhiera a las mejores prácticas de la industria, que maneje las interacciones de datos de forma segura y que proporcione una experiencia fluida para la aplicación front-end.
- **Integración:** Garantizar una comunicación y un intercambio de datos eficientes entre los componentes front-end y back-end.
- **Resolución de problemas:** Identificar y resolver los retos técnicos que surgen durante el desarrollo.

### Contenido

Este paquete de sesiones incluye los siguientes materiales:

1. **Instrucciones de la sesión (PDF):** Instrucciones detalladas sobre las tareas a realizar y los entregables esperados para esta sesión.
2. **Guía de Estilos:** Esta carpeta contiene recursos adicionales como el logotipo de GSW (General Supplies Warehouse) y otros activos de diseño que pueden utilizarse a través del desarrollo de la aplicación.
3. **Esquema de la base de datos (SQL):** Una secuencia de comandos SQL que contiene la estructura para las tablas que utilizará en esta sesión.

### Descripción del proyecto y las tareas

En esta sesión, creará la base de la aplicación de escritorio GSW general supplies warehouse.

Directrices:

1. **Facilidad de uso:** Haga que la interfaz sea sencilla y fácil de entender para el staff.
2. **Buen aspecto:** Utilice los elementos de la guía de estilo de GSW general supplies warehouse para el diseño.
3. **Buen funcionamiento:** Compruebe que todas las partes de la aplicación funcionen correctamente y sin errores.
4. **Segura:** Proteja los datos de los clientes y asegúrese de que la aplicación sea segura de usar.
5. **Puntualidad:** Termine todas las tareas dentro del plazo establecido.

### Consideraciones técnicas:

1. **Configuración de la base de datos:** Cree y rellene la base de datos de acuerdo con el esquema proporcionado.
2. **Desarrollo de la API:** Implemente una RESTful API que siga las mejores prácticas e incluya autenticación.
3. **Interfaz de usuario:** Desarrolle pantallas intuitivas para la gestión de productos y pedidos.
4. **Validación de datos:** Asegúrese de que la introducción de datos por parte del usuario sea correcta y esté completa en todas las operaciones.
5. **Tratamiento de errores:** Mostrar mensajes claros al usuario si hay algún problema.

### Consideraciones adicionales:

- La aplicación debe funcionar con fluidez y rapidez en el entorno de desarrollo proporcionado.
- Utilice etiquetas e instrucciones claras para todos los elementos de la UI.
- Organizar la información de forma que sea fácil de entender para el staff.
- Tenga en cuenta los casos extremos y los posibles errores en la entrada y en el tratamiento de datos por parte del usuario.
- Implemente casos pruebas para garantizar la funcionalidad de la API.



# Instrucciones para los concursantes

## Configuración de la base de datos e importación de datos

### Objetivo:

Crear y rellenar la base de datos GSW (General Supplies Warehouse) de acuerdo con el esquema proporcionado, garantizando la exactitud e integridad de los datos.

### Tareas:

#### 1. Creación de bases de datos:

- Cree una base de datos con el nombre **GSW\_GeneralSuppliesWarehouse** (o un nombre similar y pertinente).

#### 2. Ejecución de esquemas:

- Ejecute el script SQL proporcionado (Modulo3\_DatabaseSchema.sql) para crear las tablas y sus relaciones.

#### 3. Importación de Archivos:

- Importe el archivo "customers\_cleaned.csv" a la tabla correspondiente dentro de la base de datos "GSW\_GeneralSuppliesWarehouse".
- Importe el archivo "products\_cleaned.csv" a la tabla correspondiente dentro de la base de datos "GSW\_GeneralSuppliesWarehouse".
- Importe el archivo "sales\_transactions\_cleaned.csv" y distribúyalos en las tablas correspondientes de la base de datos "GSW\_GeneralSuppliesWarehouse".

### Entregables:

- **Nombre Archivo:** Modulo3\_CredencialesDB.
- **Tipo de Archivo:** Texto(.txt)
- **Contenido del Archivo:** Proporcione las credenciales necesarias para acceder a su base de datos (nombre del servidor, nombre de la base de datos, nombre de usuario, contraseña).

### Notas adicionales:

- Preste atención a los tipos de datos de los archivos de origen y las tablas de destino.
- Utilizar herramientas o técnicas de importación adecuadas para transferir eficazmente los datos a la base de datos.
- Vuelva a comprobar los datos tras la importación para garantizar su exactitud e integridad.
- La evaluación se centrará en la corrección e integridad de la configuración de la base de datos y en el proceso de importación de datos.

## Desarrollo de la API backend

### Objetivo:

Desarrollar una RESTful API segura y eficiente utilizando .NET Web API que permita a la aplicación de escritorio interactuar con la base de datos.

### Tareas:

#### 4. API Endpoints: Desarrolle las APIs que se describen a continuación:

- Productos:
  - GET /api/products: Obtener todos los productos
  - GET /api/products/{id}: Obtener un producto por ID
  - POST /api/products: Añadir nuevo producto
  - PUT /api/products/{id}: Cambiar producto por ID
  - DELETE /api/products/{id}: Eliminar producto por ID
- Clientes:
  - GET /api/customers: Obtener todos los clientes
  - GET /api/customers/{id}: Obtener un cliente por ID
  - POST /api/customers: Añadir nuevo cliente
  - PUT /api/customers/{id}: Cambiar cliente por ID
- Pedidos:
  - GET /api/orders: Obtener todos los pedidos
  - GET /api/orders/{id}: Obtener un pedido por ID
  - POST /api/orders: Añadir nuevo pedido
  - PUT /api/orders/{id}/complete: Finalizar pedido por ID
  - PUT /api/orders/{id}/cancel: Cancelar pedido por ID

#### 5. Comprobación de datos:

- Asegúrese de que la aplicación envía el tipo correcto de datos a la base de datos (números, letras, fechas, etc.).
- Limpie los datos para asegurarse de que su uso sea seguro.

#### 6. Manejo de errores:

- Envíe los mensajes correctos de vuelta a la aplicación si hay algún problema.

#### 7. Autenticación:

- Asegúrese de que sólo las personas correctas pueden utilizar la API. Utilice un sistema de contraseña (autenticación básica) en el que el nombre de usuario sea "staff" y la contraseña "GSWChile2025".

### Entregables:

- **Proyecto Web API Completo de .Net:** Toda la carpeta del Proyecto creada en Visual Studio Community.



## Desarrollo de la UI - Gestión de productos

### Objetivo:

Desarrollar una interfaz de escritorio fácil de usar para gestionar los productos en la aplicación GSW General Supplies Warehouse, utilizando los endpoints de la API creados en la tarea 3.2.

### Tareas:

1. Listado de productos:

Product management						
<input type="text" value="search on products/categories"/>				<a href="#">Add new product</a>		
Active	ProductName	Category	Price	Cost	Action	
<input type="checkbox"/>	Baguette Tradition	Bread	6.17	5.2	<a href="#">edit</a> <a href="#">delete</a>	
<input checked="" type="checkbox"/>	Croissant	Tarte	4.32	3.3	<a href="#">edit</a> <a href="#">delete</a>	
<input checked="" type="checkbox"/>	Eclair au Chocolat	Pastries	2.83	2.0	<a href="#">edit</a> <a href="#">delete</a>	
<input checked="" type="checkbox"/>	Pain au Chocolat	Pastries	5.53	4.6	<a href="#">edit</a> <a href="#">delete</a>	

- Muestre en una tabla todos los productos obtenidos desde el endpoint de la API GET `/api/products`.
- Columnas de la tabla: ProductName, Category, Price, Cost, Active (Yes/No).
- Añadir ordenación (ascendente/descendente) para cada columna.
- Añadir una barra de búsqueda para filtrar productos por nombre o categoría.

## 2. Formulario para añadir/editar productos:

Diagrama de un formulario web para añadir o editar productos. El formulario tiene un título "Add/Edit Product". Los campos incluyen:

- Category: lista desplegable
- Product name: entrada de texto
- Price: entrada numérica con flechas de incremento/decremento
- Cost: entrada numérica con flechas de incremento/decremento
- Introduced date: selector de fecha con icono de calendario
- ☐ Active
- ☐ Seasonal
- Description: área de texto grande

En la parte inferior hay dos botones: Save y Cancel.

- Cree un formulario para añadir nuevos productos o editar los existentes.
- Campos de formulario:
  - ProductName (entrada de texto, obligatorio, máx. 100 caracteres)
  - Category (lista desplegable, obligatoria, valores de products\_cleaned.csv)
  - Price (entrada numérica, obligatoria, debe ser un número positivo)
  - Cost (entrada numérica, obligatoria, debe ser un número positivo y menor que Precio)
  - Description (área de texto, opcional)
  - Seasonal (casilla de verificación)
  - Active (casilla de verificación)
- IntroducedDate (selector de fecha, obligatorio)
- El formulario debe tener los botones "Guardar" y "Cancelar".
- Validar los datos de entrada.
- Utilice la API para:
  - POST /api/products para guardar nuevos productos.
  - PUT /api/products/{id} para actualizar los productos existentes.
  - Manejar las respuestas de la API y mostrar los mensajes apropiados (éxito/error) al usuario.

## 3. Eliminar la funcionalidad del producto:

- Añadir un botón "Eliminar" junto a cada producto del listado.
- Mostrar un diálogo de confirmación antes de borrar.
- Utilice la API (DELETE /api/products/{id}) para eliminar el producto.
  - Manejar la respuesta de la API y actualizar la lista de productos en consecuencia.

## Entregables:

- Carpeta "bin", dónde se han exportado todos los archivos del proyecto de escritorio.

## Notas adicionales:

- Siga las mejores prácticas de diseño y desarrollo de UI.
- La UI debe ser responsiva y adaptarse a distintos tamaños de pantalla.
- Tenga en cuenta la gestión de errores y la respuesta del usuario en la UI.



## Desarrollo de la UI - Gestión de pedidos

**Objetivo:**

**Desarrollar una interfaz de escritorio fácil de usar para gestionar los pedidos de los clientes dentro de la aplicación GSW General Supplies Warehouse, utilizando los endpoints de la API creados en la tarea 3.2.**

Tareas:

- 1.** Lista de pedidos:

[illegible]

- Mostrar todos los pedidos de la base de datos en una tabla.
- Columnas de la tabla: ID del pedido, Nombre del cliente, Fecha, Importe total, Estado (Pendiente, Procesando, Completado, Cancelado).
- Permitir la ordenación (ascendente/descendente) por cualquier columna.
- Añadir una barra de búsqueda para filtrar los pedidos por ID, nombre de cliente o fecha.

## 2. Ver detalles del pedido:

**Order details view**  
  
**Order ID:** 802  
  
**Customer Name:** Manon Dupont  
  
**Order Date:** 4/11/2024  
  
**Total Amount:** 6.17  
  
**Order Status:** Pending  
  
**List of Items:**

Items	Quantity	Price

**Save** **Cancel**

- Crear una ventana o sección independiente para mostrar los detalles de un pedido seleccionado.
- Muestre la siguiente información:
  - ID del pedido
  - Nombre del cliente
  - Fecha y hora del pedido
  - Importe total
  - Estado del pedido
  - Lista de artículos pedidos con cantidad y precio

## 3. Actualizar el estado del pedido:

- Añadir botones o un menú desplegable para actualizar el estado del pedido.
- Estados disponibles: "Procesando", "Completado", "Cancelado".
- Utilizar la API para actualizar el estado del pedido en la base de datos.

### Entregables:

- Carpeta "bin", dónde se han exportado todos los archivos del proyecto de escritorio.

### Notas adicionales:

- Utilice el marco de interfaz de usuario .NET que prefiera.
- La UI debe ser responsiva y adaptarse a distintos tamaños de pantalla.
- La gestión de errores y la retroalimentación al usuario son esenciales (por ejemplo, mostrar mensajes cuando un pedido se actualiza correctamente o si se produce un error).



## Pruebas de aceptación y de caja negra

### Objetivo

X Desarrolle una documentación de casos de prueba para verificar que todos los métodos de las API estén funcionando de manera apropiada.

Solo Interfaces graficas

### Tareas

#### 1. Crear un documento con:

- ID del caso de prueba.
- Escenario.
- Flujo básico.
- Flujo alternativo.  
En este caso, el flujo ocurre cuando el usuario no introduce correctamente los parámetros de la búsqueda avanzada.
- Precondición de la prueba.
- Poscondición de la prueba.
- Resultado esperado.

### Entregables

- **Nombre del archivo:** Modulo3\_DocumentacionPruebas
- **Tipo de archivo:** Documento PDF