



Procedimientos Almacenados

1 Procedimiento: Registrar un Nuevo Pedido


 **Enunciado:** Crea un procedimiento almacenado llamado **RegistrarPedido** que reciba el ID del cliente y registre un nuevo pedido en la tabla **Pedidos**, estableciendo la fecha actual como fecha de pedido. Debe devolver el ID del pedido recién creado.

2 Procedimiento: Actualizar Stock de un Juego


 **Enunciado:** Crea un procedimiento almacenado llamado **ActualizarStockJuego** que reciba el ID de un juego y una cantidad a restar del stock. Si hay suficiente stock disponible, debe actualizar la cantidad. Si no hay suficiente stock, debe mostrar un mensaje de error.

Funciones

1 Función: Obtener Precio Total de un Pedido

 **Enunciado:** Crea una función llamada **CalcularTotalPedido** que reciba un ID de pedido y devuelva el precio total sumando los productos en la tabla **DetallePedidos**.


2 Función: Contar Juegos en una Categoría

 **Enunciado:** Crea una función llamada **ContarJuegosPorCategoria** que reciba el nombre de una categoría y devuelva el número total de juegos en esa categoría dentro de la tabla **Juegos**.

Ejercicios avanzados


Ejercicios con Condicionales (IF, CASE, etc.)

1 Determinar Descuento por Cliente


 **Enunciado:** Crea una función llamada `CalcularDescuentoCliente` que reciba el ID de un cliente y devuelva un descuento según su antigüedad en la tienda:

- Más de 3 años → 15%
- Entre 1 y 3 años → 10%
- Menos de 1 año → 5%

2 Validar Stock Antes de un Pedido


 **Enunciado:** Crea un procedimiento llamado `ValidarPedido` que reciba el ID de un juego y la cantidad deseada. Si hay suficiente stock, permite el pedido; si no hay suficiente, muestra un mensaje de error.

3 Clasificar Juegos por Precio


 **Enunciado:** Crea una función llamada `ClasificarJuegoPorPrecio` que reciba el precio de un juego y devuelva una categoría:

- "Económico" si es menor a 20€.
- "Intermedio" si está entre 20€ y 50€.
- "Premium" si es mayor a 50€.

4 Verificar Existencia de un Cliente

 **Enunciado:** Crea un procedimiento llamado `VerificarCliente` que reciba un ID de cliente y devuelva un mensaje indicando si el cliente existe en la tabla **Clientes** o no.


5 Asignar Estado a un Pedido

 **Enunciado:** Crea un procedimiento llamado `AsignarEstadoPedido` que reciba el ID de un pedido y su fecha. Según la diferencia con la fecha actual, asigna estos estados en la tabla **Pedidos**:


- "Pendiente" si fue creado hace menos de 3 días.
- "En camino" si tiene entre 3 y 7 días.
- "Entregado" si tiene más de 7 días.

Ejercicios con Estructuras Repetitivas (LOOP, WHILE, etc.)


6 Generar Reporte de Juegos con Poco Stock

 **Enunciado:** Crea un procedimiento llamado `ReporteStockBajo` que recorra la tabla `Juegos` y liste todos los juegos con stock menor a 5 unidades.


7 Aplicar Descuento Masivo a Juegos

 **Enunciado:** Crea un procedimiento llamado `AplicarDescuentoJuegos` que reduzca en un 10% el precio de todos los juegos de una categoría específica.


8 Actualizar Precios por Inflación

 **Enunciado:** Crea un procedimiento llamado `AjustarPreciosInflacion` que recorra todos los juegos y aumente su precio en un 5%, hasta que todos hayan sido actualizados.

9 Calcular Total de Pedidos en un Rango de Fechas

 **Enunciado:** Crea una función llamada `TotalPedidosPorRango` que reciba dos fechas y devuelva la cantidad total de pedidos registrados en ese período.

10 Simular Entregas de Pedidos

 **Enunciado:** Crea un procedimiento llamado `SimularEntregas` que recorra todos los pedidos "Pendientes" y los pase a "En camino", luego a "Entregado", uno por uno, hasta actualizar todos.