

**Desarrollo TI Castores**

# Evaluación Técnica de Programación

**PUNTUACIÓN:**

|  |  |
| --- | --- |
| **AUTOR** | **VERSIÓN** |
| Alberto Salazar Trejo | Versión 24.4 |

La organización de la evaluación y la distribución de puntos en los diferentes ejercicios propuestos se resume en la siguiente tabla:

|  |  |
| --- | --- |
| **EJERCICIOS** | **PUNTOS** |
| 1. Conocimientos SQL | 10 |
| 2. Ejercicio práctico: BD | 15 |
| 3. Ejercicio práctico: Desarrollo | 50 |
| 4. Documentación | 25 |
| **TOTAL** | **100** |

**NOTA:** A partir de la recepción de la evaluación contará con 72 horas para enviarla.

# CONOCIMIENTOS SQL

* 1. Describe el funcionamiento general de la sentencia JOIN.

Permite combinar las filas de dos o más tablas basándose en una columna (campo en común) relacionada entre ellas, regresando un set de datos de las diferentes tablas usadas.

* 1. ¿Cuáles son los tipos de JOIN y cuál es el funcionamiento de los mismos?

Inner join, left join, right join, full join, cross join.

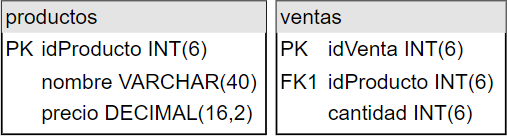
* Inner join: Retorna sola las filas que tienen coincidencias entre dos tablas.
* Left join: Retorna todas las filas de la tabla izquierda, y las filas coincidentes de la tabla derecha. Sino hay coincidencia se regresa NULL.
* Right join: Retorna todas las filas de la tabla derecha, y las filas coincidentes de la tabla izquierda. Sino hay coincidencia se regresa NULL.
* Full join: Retorna las filas cuando hay una coincidencia en una de las tablas, es decir regresa un set de filas de ambas tablas con NULL en las filas donde no hay coincidencia en la tabla opuesta.
  1. ¿Cuál el funcionamiento general de los TRIGGER y qué propósito tienen?

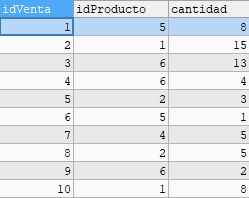
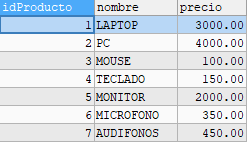
Procedimientos almacenados en una base de datos que se ejecutan automáticamente en respuesta a ciertos eventos en una tabla o vista.

Los triggers son herramientas poderosas para gestionar, asegurar la integridad y consistencias de los datos dentro de la base de datos; proporcionando automatización y control adicional.

* 1. ¿Qué es y para qué sirve un STORED PROCEDURE?

Son un conjunto de instrucciones SQL que se almacenan dentro de la base de datos y pueden ejecutar bajo demanda. Pueden compararse las funciones en los lenguajes de programación y permiten encapsular la lógica de negocios, operaciones complejas y repetitivas, mejorando la seguridad y el rendimiento en la gestión de datos.

Considerando que las siguientes tablas:

Tienen los siguientes datos insertados:

Hacer las consultas necesarias para:

* 1. Traer todos los productos que tengan una venta.

SELECT DISTINCT p.idProducto

    , p.nombre

    , p.precio

FROM productos p

INNER JOIN ventas v

    ON p.idProducto = v.idProducto

ORDER BY p.idProducto ASC;

* 1. Traer todos los productos que tengan ventas y la cantidad total de productos vendidos.

SELECT p.idProducto

    , p.nombre

    , sum(v.cantidad) AS cantidadTotalVendida

    , p.precio

FROM dbo.productos AS p

INNER JOIN dbo.ventas AS v

    ON p.idProducto = v.idProducto

GROUP BY p.idProducto

    , p.nombre

    , p.precio

ORDER BY p.idProducto ASC

* 1. Traer todos los productos (independientemente de si tienen ventas o no) y la suma total ($) vendida por producto.

SELECT p.idProducto

    , p.nombre

    , p.precio

    , COALESCE(sum(v.cantidad), 0) AS cantidad

    , COALESCE(SUM(v.cantidad \* p.precio), 0) AS sumaTotalVendida

FROM productos p

LEFT JOIN ventas v

    ON p.idProducto = v.idProducto

GROUP BY p.idProducto

    , p.nombre

    , p.precio

ORDER BY p.idProducto ASC;

# ESCENARIO: Leer el siguiente caso detenidamente para trabajar los puntos 2, 3 y 4 de tu evaluación.

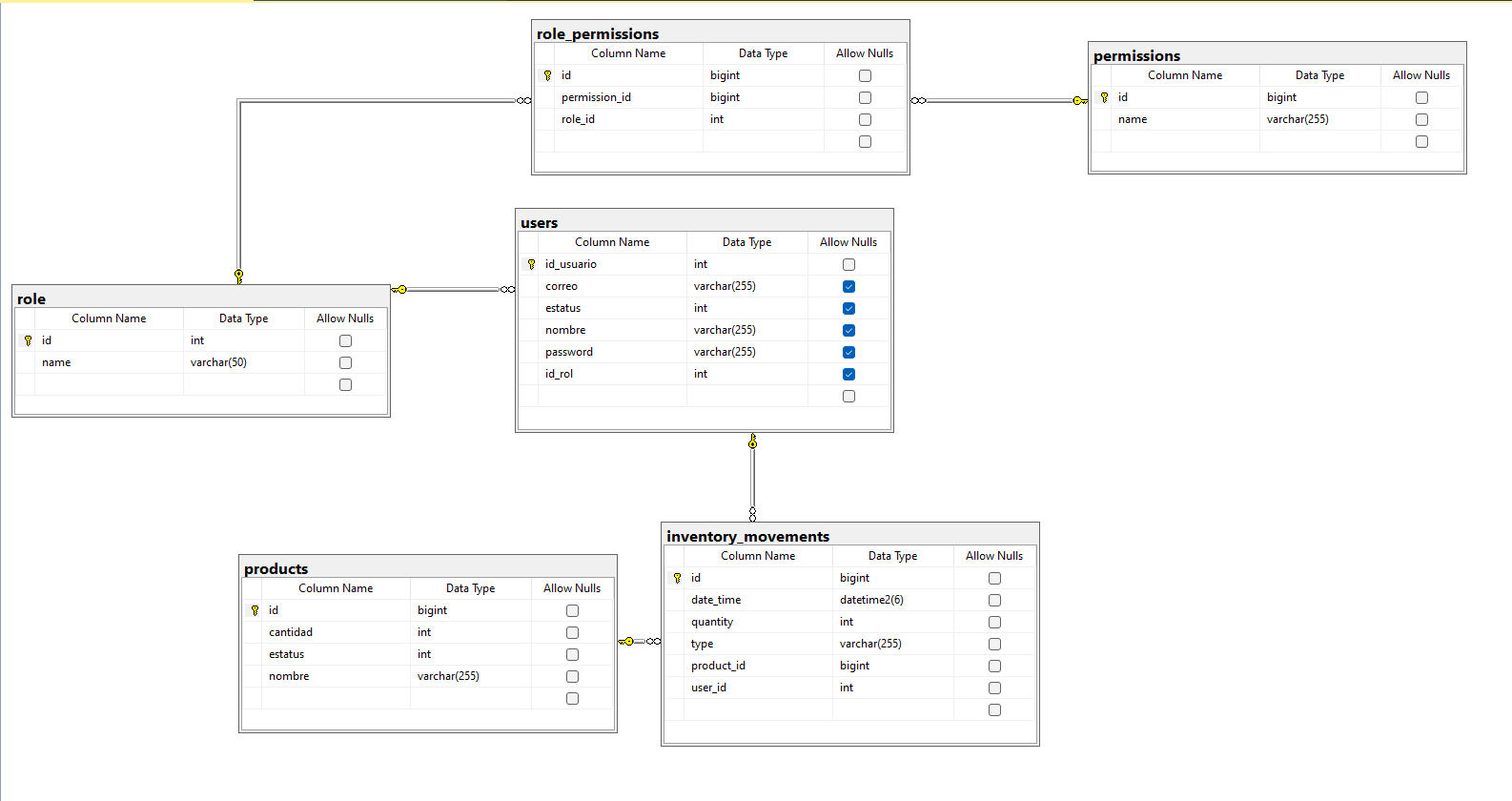
Una empresa necesita un sistema para administrar su inventario en el almacén. Este sistema debe tener las siguientes características:

* + - Debe tener un **inicio de sesión**.
    - Debe tener un **módulo** para ver el **inventario** de la empresa.
      * Se debe poder agregar nuevos productos al inventario. Al agregar un producto, la cantidad inicial será 0.
      * Se debe poder aumentar el inventario de los productos (**Entrada de productos**). Si intentas disminuir la cantidad de inventario actual, mostrará un mensaje de error.
      * Se debe poder dar de baja un producto. La baja no elimina el registro, solo actualiza el estatus. Los productos dados de baja se pueden activar nuevamente.
      * Se pueden ver los productos activos e inactivos.
    - Debe haber un **módulo** para sacar(restar) inventario del almacén (**Salida de productos**).
      * Solo se pueden ver los productos activos.
      * No se puede sacar una cantidad mayor de un producto de la que está en inventario. Si se intenta hacer, mostrará un mensaje de error.
    - Debe haber un **módulo** con el **historial** de movimientos de “Entrada” y “Salida” de productos.
      * El listado de movimientos se debe poder filtrar por tipo de movimiento (entrada o salida).
      * Cada movimiento debe registrar quién lo realizó.
      * Cada movimiento debe tener la fecha y hora en que se realizó el movimiento.
    - El sistema tendrá 2 roles y los permisos de los mismos se describen a continuación:

|  |  |  |
| --- | --- | --- |
| **Permiso** | **Administrador** | **Almacenista** |
| Ver módulo inventario | ✓ | ✓ |
| Agregar nuevos productos | ✓ | ✕ |
| Aumentar inventario | ✓ | ✕ |
| Dar de baja/reactivar un producto | ✓ | ✕ |
| Ver módulo para Salida de productos | ✕ | ✓ |
| Sacar inventario del almacén | ✕ | ✓ |
| Ver módulo del histórico | ✓ | ✕ |

# EJERCICIO PRÁCTICO: BD

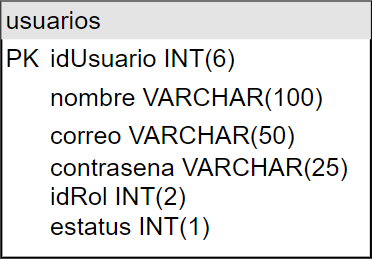
* 1. Crea un diagrama relacional de BD para el escenario descrito anteriormente.



* 1. Hacer el script para crear las tablas del punto anterior (en el punto 4.2 se especificará dónde debe anexarse dicho script)

El script se encuentra en el repositorio

**NOTA:** Actualmente solo se cuenta con la tabla de “usuarios”. Falta considerar las tablas para saber identificar el tipo de rol, los productos y para el histórico.



# EJERCICIO PRÁCTICO: DESARROLLO

* 1. Crear una aplicación **WEB** para el escenario que se planteó previamente.
     + Preferiblemente utilizar el diseño **MVC** para su desarrollo.
     + Preferiblemente utilizar **JAVA**.
     + Utilizar **MySQL** o **SQLServer**.

# DOCUMENTACIÓN:

* 1. Se necesita que el código se suba en un repositorio de **GITHUB** y esté público para que pueda ser descargado.
  2. Los scripts generados para la evaluación deben encontrarse en una carpeta llamada **SCRIPTS**, dentro del repositorio de GITHUB.
  3. Dentro del repositorio, debe haber un **README.md** donde se especifiquen los datos relevantes para el desarrollo:
     + IDE utilizado.
     + Versión del lenguaje de programación utilizado.
     + DBMS utilizado y su versión.
     + Lista de pasos para correr su aplicación.
  4. Haga un **video** donde se muestren los siguientes escenarios dentro de su aplicación:

Mostrar que los usuarios con el rol de **Administrador** pueden:

* + - Iniciar de sesión.
    - Visualizar el histórico y filtrarlo.
    - Registrar productos nuevos.
    - Dar de baja productos.
    - Reactivar productos.
    - Agregar existencias a los productos.
    - No pueden acceder al módulo para salida de material.

Mostrar que los usuarios con el rol de **Almacenista** pueden:

* + - Iniciar de sesión.
    - No pueden visualizar el histórico.
    - Solo pueden visualizar el módulo de inventario.
    - Puede acceder al módulo para salida de material y sacar material.

Mostrar los mensajes de error que se consideraron en el escenario planteado.

**NOTA**: Dicho video se puede compartir a través de Google Drive.