Plantilla de Trabajo – Evaluación N°2 TI3021

Esta plantilla debe ser completada por cada grupo y entregada junto al repositorio en GitHub. Incluye toda la documentación necesaria para la evaluación del proyecto.

# 1. Identificación del grupo

Nombre del grupo: Equipo 5

Integrantes (nombre completo y rol):

- Hector Nazareth Riffo – Modelado UML, Programacion, Conexion a BD, Tester, Documentacion/GitHub

Repositorio GitHub: <https://github.com/hectorRiffo21/Evaluacion2.git>

# 2. Caso asignado

Nombre del caso: Caso 3, Inventario De Productos de Bodega

Breve descripción del sistema a implementar:

EL Sistema tiene como Proposito gestionar el inventario de una bodega, almacen o tienda. Permite registrar los productos en distintas categorias, controlar la entrada y salida de estos, detector bajo stock de cierto producto para su reposicion, se registra el historial de movimientos indicando que persona realizo cierta operacion junto al motivo correspondiente

# Imagen de la pantalla de un celular de un mensaje en letras negras El contenido generado por IA puede ser incorrecto.3. Diagrama de Clases UML

# 4. Modelo de Base de Datos

¿Qué tablas usaron y qué datos guarda cada una?

- Tabla 1: Producto Campos: id,codigo\_marca,nombre,descripcion,categoria\_id,stock\_actual,stock\_minimo,stock\_maximo,precio,estado

- Tabla 2: Categoria Campos: id\_categoria,nombre\_categoria,Descripcion

- Tabla 3: Movimiento Campos: id\_movimiento,id\_producto,id\_empleado,tipo\_movimiento,motive,cantidad,fecha,stock\_antes,stock\_despues

- Tabla 4: Empleado Campos: id\_empleado,nombre,apellido,rut,telefono,cargo,username,clave

- Relación entre tablas:

Un producto pertenece a una categoria,

Un empleado realiza un movimiento

un producto esta asociado a un movimiento

un empleado puede registrar muchos movimientos

una categoria puede tener muchos productos

# 5. Funcionalidades implementadas

* Registro de entidades principales (CRUD básico)
* Consultas personalizadas (por filtros, búsqueda, relaciones)
* Conexión segura a base de datos (ej. SQLite, MySQL)
* Manejo de errores (try/except)
* Uso correcto de clases, herencia y encapsulamiento
* Código subido a GitHub con commits por etapa
* Presencia de diagrama UML y estructura lógica

# 6. Capturas de funcionamiento

Adjuntar capturas o enlaces con evidencia de funcionamiento (opcional):

- Captura 1: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

- Captura 2: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# 7. Reflexión final del grupo

¿Qué aprendieron desarrollando este proyecto? ¿Qué mejorarían?

(Responder en 4 a 6 líneas)