**INSTITUTO SUPERIOR DEL POLITÉCNICO CÓRDOBA**



**TECNICATURA SUPERIOR EN CIENCIAS DE DATOS E INTELIGENCIA ARTIFICIAL**

**MÓDULO INNOVACIÓN EN GESTIÓN DE DATOS**

**PRIMER AÑO**

**TRABAJO FINAL**

**PROYECTO SISTEMA DE GESTIÓN DE KIOSCO**

|  |  |  |
| --- | --- | --- |
| Estudiantes: |  | Docentes |
| * Carlos Garcia |  | * Julian Conde |
| * Félix Tapia |  | * Romina Cattaneo |
| * Raúl Moreno |  |  |

01 de noviembre de 2024

**Índice**

* **Descripción del Proyecto**

El proyecto "Sistema de Gestión del Kiosco" consiste en el desarrollo de una aplicación integral para la gestión eficiente de negocios tipo kiosco. La aplicación abarcará:

* Gestión de inventario y stock
* Control de proveedores y precios
* Sistema de facturación
* Administración de sucursales y empleados
* Generación de reportes estadísticos

Se utilizará Python como lenguaje de programación principal y MySQL para la gestión de base de datos, implementando una arquitectura orientada a objetos.

Falta etapas o actividades que compondran.

* **Justificación**

Este proyecto surge como respuesta a las siguientes problemáticas identificadas en la gestión tradicional de kioscos que permita gestionar de manera eficiente y práctica todos los aspectos de un negocio de kiosco. Dada la complejidad de manejar un amplio catálogo de productos, múltiples proveedores, y diversas relaciones internas y externas, los negocios de kiosco suelen enfrentar varios desafíos relacionados con la gestión de inventarios, proveedores y ventas. Estos problemas incluyen:

* **Desorden en la Gestión del Inventario:** La falta de un sistema organizado puede llevar a errores en el stock, como vencimientos no detectados y desactualización de precios.
* **Dificultades en el Control de Proveedores:** La gestión de múltiples proveedores y sus precios puede resultar complicada, llevando a inconsistencias en el costo y la disponibilidad de productos.
* **Problemas en la Facturación:** La falta de un registro claro de las ventas y facturas puede dificultar el seguimiento de ingresos y tipos de pagos, afectando la contabilidad y la toma de decisiones.
* **Desorganización en la Información de Sucursales y Empleados:** La falta de un sistema centralizado para gestionar la información de sucursales y empleados puede llevar a confusión y errores en la administración.
* **Objetivos**
* **Objetivo del Proyecto**

Desarrollar una aplicación que facilite la organización y gestión de un kiosco mediante la integración de diversos datos clave, tales como inventario, proveedores, facturación, sucursales y empleados.

* **Objetivo Específicos**
* Implementar un sistema de gestión de inventario que permita el control eficiente de stock y vencimientos.
* Desarrollar un módulo de administración de proveedores que facilite el seguimiento de precios y relaciones comerciales.
* Crear un sistema de facturación que permita el registro y análisis de ventas por sucursal.
* **Metodología**

El proyecto se desarrollará utilizando una metodología ágil, con las siguientes etapas y recursos:

* **Etapas de Desarrollo:**
* Análisis de requisitos y diseño de la base de datos.
* Desarrollo de la aplicación en Python, utilizando programación orientada a objetos.
* Implementación de la base de datos relacional con MySQL.
* Desarrollo de módulos específicos (inventario, proveedores, facturación, etc.).
* Integración de todos los módulos en una aplicación unificada.
* Pruebas y depuración.
* Documentación del proyecto.
* **Recursos Necesarios**
* **Equipo Humano:** 3 desarrolladores (estudiantes)
* **Equipo Técnico:**
* Computadoras con Python y MySQL instalados
* Entorno de desarrollo integrado (IDE)
* Sistema de control de versiones
* **Estructura de la Base de Datos**

Se implementará una base de datos relacional con las siguientes tablas principales:

* Productos
* Proveedores
* Productos\_Proveedor
* Ventas
* Detalle\_Venta
* Sucursales
* Empleados

Detalles de la Base de Datos

La base de datos para el Sistema de Gestión de Kiosco está diseñada para almacenar y gestionar información relacionada con productos, proveedores, ventas, sucursales y empleados. Esta estructura permite una gestión eficiente de todos los aspectos del negocio.

*Tablas*

* Productos
* ID\_Producto (PK)
* Nombre
* Descripción
* Precio\_Venta
* Cantidad\_Stock
* Fecha\_Vencimiento
* Proveedores
* ID\_Proveedor (PK)
* Nombre
* Contacto
* Teléfono
* Email
* Productos\_Proveedor
* ID\_Producto (FK)
* ID\_Proveedor (FK)
* Precio\_Compra
* Ventas
* ID\_Venta (PK)
* Fecha
* Total
* Tipo\_Pago
* ID\_Sucursal (FK)
* Detalle\_Venta
* ID\_Venta (FK)
* ID\_Producto (FK)
* Cantidad
* Precio\_Unitario
* Sucursales
* ID\_Sucursal (PK)
* Dirección
* Teléfono
* ID\_Encargado (FK)
* Empleados
* ID\_Empleado (PK)
* Nombre
* Apellido
* DNI
* Teléfono
* Fecha\_Ingreso
* Salario
* Cargo
* ID\_Sucursal (FK)

*Relaciones*

* Productos - Productos\_Proveedor: One-to-Many
* Proveedores - Productos\_Proveedor: One-to-Many
* Ventas - Detalle\_Venta: One-to-Many
* Productos - Detalle\_Venta: One-to-Many
* Sucursales - Ventas: One-to-Many
* Sucursales - Empleados: One-to-Many
* Empleados - Sucursales (Encargado): One-to-One
* **Cronograma**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Fase | Actividad | Semana 1 | Semana 2 | Semana 3 | Semana 4 |
| 1 | Análisis y Diseño | X |  |  |  |
| 2 | Desarrollo Base de Datos |  | X |  |  |
| 3 | Desarrollo Módulos |  |  | X |  |
| 4 | Integración y Pruebas |  |  |  | X |

* **Presentación del Proyecto**

6.1 Estructura del Código

El proyecto está organizado en varios módulos Python que trabajan en conjunto para proporcionar toda la funcionalidad del sistema. A continuación, se presentan los componentes principales:

6.1.1 Conexión a Base de Datos (mysql\_conexion.py)

import mysql.connector  
import config  
  
def conexion(query: str, datos: tuple = None, tipo: int = 1) -> list[dict] | int:  
 """  
 Desde esta funcion se realiza la conexion a la base de datos y se ejecuta  
 el comando sql que se recibe como parametro.  
 """  
 conexion = mysql.connector.connect(\*\*config.config)  
 cursor = conexion.cursor(dictionary=True)  
 cursor.execute(query, datos)  
  
 if tipo == 1:  
 resultado = cursor.fetchall()  
 cursor.close()  
 conexion.close()  
 return resultado  
 else:  
 conexion.commit()  
 row\_id = cursor.lastrowid  
 cursor.close()  
 conexion.close()  
 return row\_id

6.1.2 Gestión de Archivos (gestion\_de\_archivos.py)

def registrar\_log(username, password):  
 """  
 Escribe en un archivo de texto un log con el siguiente formato:  
 | Fecha actual | Nombre de usuario | Contraseña |  
 """  
 file = open('logs.txt', 'a')  
 log = f'|{datetime.datetime.now()}| usuario: {username} | clave: {password}\n'  
 file.write(log)  
 file.close()  
  
def escribir\_binario(archivo, objeto=None):  
 """  
 Escribe el archivo que se recibe por parametro con lo especificado en el parametro objeto.  
 """  
 file = open(archivo, 'wb')  
 pickle.dump(objeto, file)  
 file.close()

6.1.3 Clase Producto y Consultas (Consultas.py)

class Producto:  
 def \_\_init\_\_(self, id=None, nombre=None, tipo=None, unidades=None,   
 precio=None, vencimiento=None, id\_proveedor=None):  
 self.\_id = id  
 self.\_nombre = nombre  
 self.\_tipo = tipo  
 self.\_unidades = unidades  
 self.\_precio = precio  
 self.\_vencimiento = vencimiento  
 self.\_id\_proveedor = id\_proveedor  
  
def unidades\_vendidas():  
 """  
 Se solicita a la base de datos un listado que contenga la cantidad  
 de unidades vendidas por producto.  
 """  
 query = """select p.nombre\_producto as Producto,   
 sum(d.cantidad) as 'Unidades vendiadas'   
 from detalle\_venta d  
 inner join producto p on p.id\_producto = d.id\_producto   
 group by p.nombre\_producto  
 order by sum(d.cantidad) desc;"""  
   
 return conexion(query, datos=None, tipo=1)

6.1.4 Formateador de Datos (Formateador.py)

def formateador(resultados: list, ancho = 20) -> None:  
 """  
 Esta funcion se encarga de presentar en formato de tabla a traves   
 de la consola los datos que contiene resultado.  
 """  
 if not resultados:  
 print('\n' \* 50)  
 print('La tabla esta vacia...')  
 return  
  
 print('\n' \* 50)  
 print('=' \* len(resultados[0]) \* 31 + '=')  
 linea = '|'  
 for resultado in resultados[0].keys():  
 linea = linea + '{:^30}|'.format(str(resultado))  
 print(linea)  
 print('=' \* len(linea))

6.2 Funcionalidades Implementadas

El sistema incluye las siguientes funcionalidades principales:

* **Gestión de Productos:**
* Mostrar productos
* Insertar nuevo producto
* Actualizar producto existente
* Eliminar producto
* **Reportes de Ventas:**
* Unidades vendidas por producto
* Productos vendidos por mes
* Ventas por empleado
* Ventas por mes
* **Características Técnicas:**
* Conexión segura a base de datos MySQL
* Manejo de errores y excepciones
* Sistema de logging para seguimiento de operaciones
* Formateador de datos para presentación en consola
* Algoritmos de búsqueda y ordenamiento optimizados
* **Conclusiones**

[Las conclusiones se agregarán una vez finalizado el desarrollo del proyecto]

Durante el desarrollo de este proyecto hemos podido aplicar mucho del conocimiento estudiado durante la cursada, lo cual nos permitio afianzarlo aun mas. Ademas no solo tuvimos la oportunidad de desarrollar nuestras habilidades tecnicas sino que tambien pusimos a prueba nuestras habilidades blandas, mas especificamente, el trabajo en equipo y la comunicacion efectiva.

En lo relacionado al resultado obtenido, estamos conformes, aunque creemos que todavia existe un amplio margen de mejora.

**Fuente Bibliográfica**

* Apuntes del Módulo Innovación de Gestión de Datos (2024). Tecnicatura Superior de Ciencias de Datos e Inteligencia Artificial. Instituto Superior del Politécnico Córdoba.