

OF
T
C
E
Y
O
R
R



SISTEMA DE
GESTION

NOVIEMBRE
2024

LABORATORIO
IV 4A

PROFESOR:
ALEX ROBERT

Índice

01

introducción

02

gestión de sistemas

03

tecnología usada

04

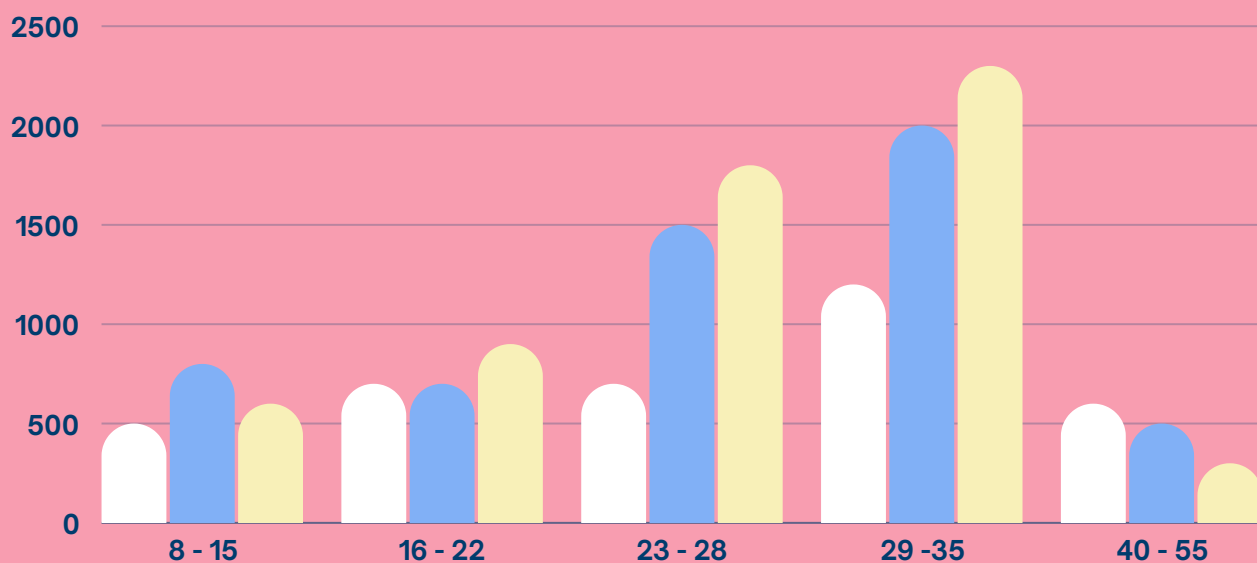
Integrantes del proyecto



introducción

01

Mediante este proyecto se pretende modernizar la ferretería, un sector históricamente rezagado en cuanto a tecnología, con el fin de optimizar sus operaciones y expandir su presencia en el mercado, haciéndola más competitiva.



 práctico

 eficiente

 rápido



02

Gestión de sistema

```
1 import sqlite3
2
3 # Función para crear la base de datos y la tabla si no existen
4 def CrearDB1():
5     try:
6         conexion = sqlite3.connect("Stock") # Conectamos o creamos la base de datos "Stock"
7         miCursor = conexion.cursor() # Creamos un cursor para ejecutar comandos SQL en la base de datos
8
9         miCursor.execute('''
10             CREATE TABLE IF NOT EXISTS STOCKDISPONIBLE (
11                 ID INTEGER PRIMARY KEY AUTOINCREMENT,
12                 NOMBRE_ARTICULO VARCHAR(50),
13                 PRECIO INTEGER,
14                 SECCION VARCHAR(50),
15                 MARCA VARCHAR(50)
16             )
17         ''')
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\doit\Desktop\back up\escritorio\proyectoFerreteria2024> & C:/Users/doit/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/doit/Desktop/back up/escritorio/proyectoFerreteria2024/corregido_con_funciones.py"
Base de datos y tabla creadas exitosamente.
Ingrese 'registrarse' o 'iniciar sesion':
```

El sistema ofrece una experiencia de usuario sencilla e intuitiva, permitiendo tanto el inicio de sesión como el registro de nuevos usuarios. Los datos de registro se almacenan para su posterior verificación y actualización

02

Gestión de sistema

```
1 import sqlite3
2
3 # Función para crear la base de datos y la tabla si no existen
4 def CrearDB1():
5     try:
6         conexion = sqlite3.connect("Stock") # Conectamos o creamos la base de datos "Stock"
7         miCursor = conexion.cursor() # Creamos un cursor para ejecutar comandos SQL en la base de datos
8
9         miCursor.execute('''
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS Python + v

Ingrese su nombre de usuario: jhonn
Ingrese su contraseña: 1234567
Bienvenido jhonn
vendedor

ID	Nombre Artículo	Precio	Sección	Marca
1	tornillo 3/4	200.00	tornillos	generico
2	arandela	50.00	arandela	generico
3	tuerca	100.00	tuercas	generica
5	destornillador	550.00	destornillador	generico
7	busca polo	2000.00	destornillador	IND ARG
8	tarugo	250.00	tarugos	fischer
10	taladro	37000.00	electricos	dewalt
11	amoladora	65000.00	electricos	bosch
12	viga/tirante	20600.00	madera	IND ARG
13	soporte/viga	3000.00	madera	IND ARG
14	caladora	76300.00	electricos	skil
15	serrucho	9300.00	madera	black jack
17	cable unipolar	31200.00	electricos	microtec
18	bastidor	2430.00	electricos	jeluz
19	cinta aisladora	954.00	cintas	tecsa
20	portalampara	4200.00	iluminacion	IND ARG

Ingrese el ID del artículo que desea agregar al carrito (0 para finalizar):

El rol de Vendedor permite gestionar ventas, incluyendo la adición de múltiples artículos al carrito y el cálculo de subtotales.

02

Gestión de sistema

```
116 def ordenarArticulosNomDesc():
117     if conexion:
118         try:
119             miCursor = conexion.cursor() # Creamos un cursor para ejecutar comandos SQL
120
121             # Consultar y ordenar por nombre de forma ascendente
122             miCursor.execute('SELECT * FROM STOCKDISPONIBLE ORDER BY NOMBRE_ARTICULO DESC')
123             articulos = miCursor.fetchall()
124
125             if not articulos: # Si no hay registros
126                 print("No hay articulos en el stock.")
127             else:
128                 print(f"{'ID':<5} {'Nombre Articulo':<20} {'Precio':<10} {'Sección':<15} {'Marca':<15}")
129                 print("-" * 65)
130                 for articulo in articulos:
131                     print(f"{articulo[0]:<5} {articulo[1]:<20} {articulo[2]:<10.2f} {articulo[3]:<15} {articulo[4]:<15}")
132
133
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python

Opción no válida. Por favor, ingrese un número del 1 al 6.

```
** MENÚ **
1. Agregar artículo
2. Mostrar artículos
3. Actualizar artículo
4. Eliminar artículo
5. Mostrar base de datos completa
6. Ordenar Stock
7. Salir
Ingrese el número de la opción que desea ejecutar: 7
¡Hasta luego!
Ingrese el área a la que desea entrar
Clientes-Proveedores(CP) / Stock(ST) / Usuario(US) / Cerrar App(FIN): 7
Ingrese el área a la que desea entrar
Clientes-Proveedores(CP) / Stock(ST) / Usuario(US) / Cerrar App(FIN):
```

Con el perfil de Administrador se pueden realizar todas las operaciones sobre el stock y los usuarios, como agregar, eliminar o modificar registros.

02

Gestión de sistema

```
167 # Función para mostrar todos los artículos en el stock
168 # Realiza la conexión, ejecuta una consulta para obtener todos los registros y los muestra en pantalla
169 def mostrarArticulos():
170     conexion = conectarDB1() # Conectamos a la base de datos
171     if conexion:
172         try:
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

3. Actualizar artículo
4. Eliminar artículo
5. Mostrar base de datos completa
6. Ordenar Stock
7. Salir
Ingrese el número de la opción que desea ejecutar: 6
¿Como desea ordenarlo?
1. Nombre Ascendente
2. Nombre Descendente
3. Precio Ascendente
4. Precio Descendente
4

ID	Nombre Artículo	Precio	Sección	Marca
14	caladora	76300.00	electricos	skil
11	amoladora	65000.00	electricos	bosch
10	taladro	37000.00	electricos	dewalt
17	cable unipolar	31200.00	electricos	microtec
12	viga/tirante	20600.00	madera	IND ARG
15	serrucho	9300.00	madera	black jack
20	portalampara	4200.00	iluminacion	IND ARG
13	soporte/viga	3000.00	madera	IND ARG
18	bastidor	2430.00	electricos	jeluz
7	busca polo	2000.00	destornillador	IND ARG
19	cinta aisladora	954.00	cintas	tecsa
5	destornillador	550.00	destornillador	generico
8	tarugo	250.00	tarugos	fischer
1	tornillo 3/4	200.00	tornillos	generico
3	tuerca	100.00	tuercas	generica

El perfil de Administrador permite gestionar el inventario de forma completa, incluyendo la organización del stock según diferentes criterios.

tecnología utilizada

03

Se decidió utilizar Python y SQLite para este proyecto. Python fue elegido por su flexibilidad y capacidad de adaptarse, mientras que SQLite se seleccionó por su desempeño en bases de datos de menor tamaño y su integración nativa con Python.

04

Los integrantes del proyecto son:

Valentin Perez

Nahuel Jimenez

Lucas Ferreira

Maximiliano Maciel

¡Gracias!



**SISTEMA DE
GESTION**