

# MANUAL TECNICO:

Este programa se desarrolló con la finalidad de poder optimizar el lugar de pedidos de un restaurante para que así se tenga un mejor control de los mismos y no poder cometer mas fallos dentro de la empresa.

## Requerimientos mínimos:

Para el desarrollo de este programa e necesito con las siguiente características:

Procesador Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz 1.99 GHz

RAM instalada 16.0 GB (15.8 GB utilizable)

Tipo de sistema Sistema operativo de 64 bits, procesador x64

Windows 11 última versión.

## DICCIONARIO DE METODOS:

- **RegresarActionPerformed:** método que maneja el evento de presionar el botón "Regresar" y muestra la ventana del menú principal de la aplicación.
- **MandarPedidoActionPerformed:** método que maneja el evento de presionar el botón "Mandar Pedido" y llama al método seleccionTabla.
- **AgregarActionPerformed:** método que maneja el evento de presionar el botón "Agregar" y llama al método agregar, refrescarTabla y serializarDatos.
- **ProductoActionPerformed:** método que maneja el evento de escribir en el campo de texto "Producto".
- **EliminarActionPerformed:** método que maneja el evento de presionar el botón "Eliminar" y elimina una fila seleccionada de la tabla ProductosMuestra, así como su correspondiente pedido y llama al método serializarDatos.
- **LargoActionPerformed:** método que maneja el evento de escribir en el campo de texto "Largo".
- **PrecioActionPerformed:** método que maneja el evento de escribir en el campo de texto "Precio".
- **VerMotosActionPerformed:** método que maneja el evento de presionar el botón "Ver Motos" y muestra la ventana de entrega de productos en motocicletas.
- **ConfirmarCompraActionPerformed:** método que maneja el evento de presionar el botón "Confirmar Compra" y actualiza la distancia que se debe recorrer en la motocicleta seleccionada.
- **refrescarTablaPedidos:** método que refresca la información de la tabla de pedidos.
- **EliminarProductoActionPerformed:** método que maneja el evento de presionar el botón "Eliminar Producto" y elimina una fila seleccionada de la tabla ProductosPedido y su correspondiente pedido.
- **refrescarTabla:** método que refresca la información de la tabla de productos disponibles.
- **seleccionTabla:** método que agrega un producto seleccionado a los pedidos.
- **cargarProductoPedido:** método que carga los productos pedidos en la tabla ProductosPedido.
- **sumaPedido:** método que calcula la suma del precio de los productos en el pedido.
- **cargarProductos:** método que carga la información de los productos disponibles en la tabla ProductosMuestra.
- **serializarDatos:** método que guarda la información de los productos en un archivo utilizando serialización.
- **cargarDatos:** método que carga la información de los productos desde un archivo utilizando deserialización.
- **deserializarDatos:** método que carga la información de los productos desde un archivo utilizando deserialización.

# DICCIONARIO DE CLASES:

- **RegresarActionPerformed**: método que maneja el evento de presionar el botón "Regresar" y muestra la ventana del menú principal de la aplicación.
- **MandarPedidoActionPerformed**: método que maneja el evento de presionar el botón "Mandar Pedido" y llama al método seleccionTabla.
- **AgregarActionPerformed**: método que maneja el evento de presionar el botón "Agregar" y llama al método agregar, refrescarTabla y serializarDatos.
- **ProductoActionPerformed**: método que maneja el evento de escribir en el campo de texto "Producto".
- **EliminarActionPerformed**: método que maneja el evento de presionar el botón "Eliminar" y elimina una fila seleccionada de la tabla ProductosMuestra, así como su correspondiente pedido y llama al método serializarDatos.
- **LargoActionPerformed**: método que maneja el evento de escribir en el campo de texto "Largo".
- **PrecioActionPerformed**: método que maneja el evento de escribir en el campo de texto "Precio".
- **VerMotosActionPerformed**: método que llama las propiedades del vehiculo.
  
- **MenuActionPerformed**: método que maneja el evento de presionar el botón "Menú" y muestra la ventana del menú principal de la aplicación.
- **EnviarTodosActionPerformed**: método que maneja el evento de presionar el botón "Enviar Todos" y crea tres objetos Recorrido para enviar las tres motocicletas a sus respectivas distancias y las inicia.
- **bmoto1ActionPerformed**: método que maneja el evento de presionar el botón "Moto1" y crea un objeto Recorrido para enviar la motocicleta 1 a la distancia especificada y lo inicia.
- **bmoto2ActionPerformed**: método que maneja el evento de presionar el botón "Moto2" y crea un objeto Recorrido para enviar la motocicleta 2 a la distancia especificada y lo inicia.
- **bmoto3ActionPerformed**: método que maneja el evento de presionar el botón "Moto3" y crea un objeto Recorrido para enviar la motocicleta 3 a la distancia especificada y lo inicia.
- **RegresarActionPerformed**: método que maneja el evento de presionar el botón "Regresar" y muestra la ventana de pedidos de la aplicación.
- **getMeta**: método que devuelve el objeto JLabel que muestra la meta de la carrera.