

NOMBRE DEL SISTEMA

Mango Bienes y Servicios

DESCRIPCIÓN

Mango Bienes y Servicios, es un prototipo de programa de intercambio de información sobre ofertas y demandas de bienes y servicios entre usuarios directamente por medio de una red p2p.

En principio esta pensado para equipos sobre mesa, pero luego en una segunda etapa se puede llegar a celulares y tablets.

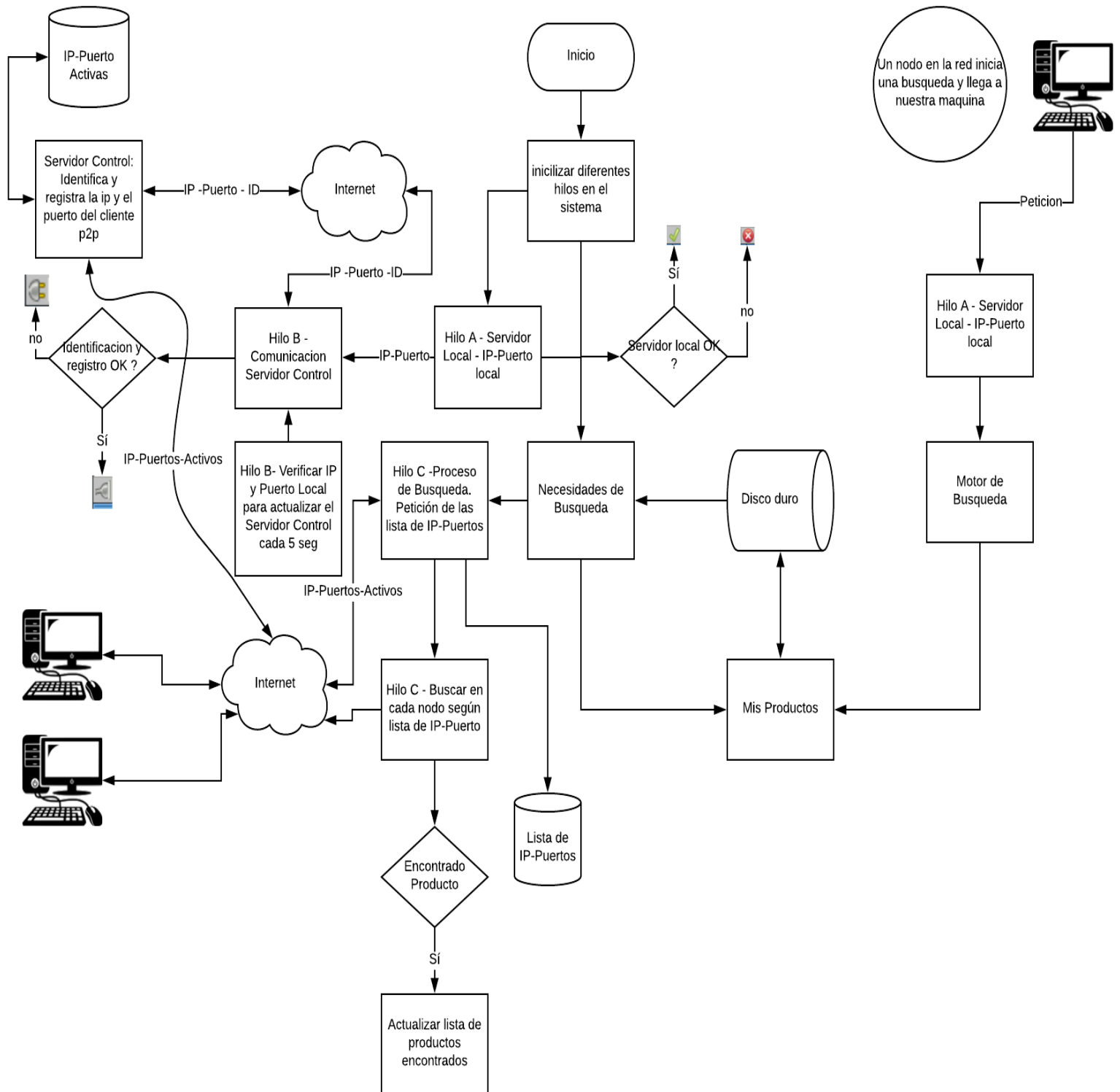
El uso del programa es para un publico muy amplio, no se limita a un sector determinado, podría ser usado por instituciones publicas y privadas, empresas del estado y privadas, organizaciones y por particulares, es crear una comunidad virtual.

Mientras mas personas puedan publicar sus productos, servicios o necesidades el sistema tendrá una amplia gama de posibilidades de encontrar lo que se necesita, o en otro sentido, de encontrar quien quiere el producto que estas ofertando.

El sistema publica por igual bienes, servicios y necesidades.

ARQUITECTURA

El sistema se basa en una red p2p híbrida ligera, sólo existirá un servidor central donde se recoge las ip y los puertos de las maquinas en linea en ese momento.



TECNOLOGÍA

El programa esta escrito en lenguaje C, Ansi-C C99, compilado por GCC. Como única librería externa usa GTK3 para el manejo de la interface gráfica.

CAPACIDAD DE AMPLIACIÓN

¿Cuenta usted con recurso humano para realizarlo? **SI**
¿Tiene usted la capacidad tecnológica para hacerlo? **SI**

ALCANCE DE LA SOLUCIÓN

En el sistema se conglomerarán y encontrarán todas los usuarios que buscan bienes y servicios, también las personas que ofertan bienes y servicios, todo bajo un estricto sistema de confianza colectiva, donde los mismos usuarios calificarán a sus integrantes, dándole mayor peso de confianza a los mejores, esto es un sistema donde se premia la excelencia y el trabajo de calidad.

Por otra parte, el sistema tendrá la posibilidad de hacer búsquedas configurables por región, sector y muchos otros criterios, no serán sólo búsquedas instantáneas al estilo Google, aparte de búsquedas instantáneas son también alertas, que cuando un usuario quizás tiempo después publica lo que tu buscas se generará una coincidencia de búsqueda encontrando también ese producto o servicio.

Busca lo que quieras... el sistema lo encontrará por ti cuando y donde lo necesites.

Objetivos

- Crear una plataforma tecnológica descentralizada de bajo costo de desarrollo y mantenimiento para la publicación de bienes, servicios y necesidades.
- Vincular por medios de algoritmos a los eferentes con los demandantes.
- Crear una plataforma tecnológica escalable, adaptable, responsiva y multiplataforma.
- Crear una comunidad virtual donde se pueda premiar al trabajo y la cálida.
- Crear canales de difusión para los nuevos emprendedores.
- Crear nuevos canales de comercialización.

ASPECTOS QUE NO INCLUYE LA SOLUCIÓN

El desarrollo de esta primera etapa de la solución no incluye versiones para android, como tampoco para tablet.

INFORMACIÓN ESPECIFICA DE LA SOLUCIÓN

A continuación se enumerarán los módulos mas generales del sistema.

ID	Servicio	Requerimiento
1	Módulo Caracterización	Registro de la información del operador y del ente responsable en la comunidad virtual por el sistema. Actualización de la información Verificación de la información
2	Módulo de Seguridad	Modulo de encriptación de la data para una transmisión segura entre pares.
3	Módulo Servidor Control	Software del lado del servidor que mantendrá la lista de los IP, y de los puertos para nodo del sistema. Verificación de seguridad para dar confirmar que el nodo es quien dice ser.
4	Módulo Motor de Búsqueda	Algoritmos especializados para aproximar las búsquedas a los resultados
5	Módulo Servidor Local	Demonio que estará a cargo de mantener un puerto en escucha para la peticiones de otros nodos.
6	Módulo de Procesamiento y Envío de Archivos	Estará a cargo de procesar en segundo plano envíos de archivos pequeños y grandes sin bloquear la aplicación y sin perder lo descargado en caso de perdidas de la conexión.
7	Módulo de Fragmentación de Archivos	Estará a cargo de controlar y administrar el envío de “partes” de los archivos grandes entre nodos
8	Módulo Necesidades	Administración de las

		<p>necesidades (entiéndase necesidades como búsquedas programadas) .</p> <p>Creación, edición, eliminación, visualización.</p>
9	Módulo Productos	<p>Administración de los bienes y servicios creados para ofertar.</p> <p>Creación, modificación, eliminación.</p>
10	Módulo Chat	<p>Abre canales de chat entre los pares de usuarios conectados.</p> <p>Se gestionara, los mensajes y duración de los mismos.</p>
11	Módulo Buscador	<p>Demonio que estará permanentemente buscando en toda la red p2p la satisfacción de las necesidades creadas.</p>
12	Módulo de Interface Gráfica	<p>Gestión de la interface gráfica en todas las partes del proyecto.</p>
13	Módulo de Actualización	<p>Gestionará las posibles actualizaciones al sistemas desde el servidor de actualizaciones.</p>
14	Módulo Servidor de Actualizaciones	<p>Será el encargado desde un punto central en la red de enviar los binarios actualizados a los clientes.</p>
15	Módulo de ayuda	<p>Gestionará la ayuda inteligente en el sistema.</p> <p>Crearé índices y preferencias de usuario para mostrar la ayuda mas precisa</p>
16	Módulo de Notificación de Ip y Puerto	<p>Demonio que mantendrá al servidor central actualizado en cuanto a la información del ip y puerto local usado</p>

DESCRIPCIÓN DE LAS FUENTES

macros.h

Archivo que contiene todas las **define** del programa de acceso global.

enums.h

Archivo que contiene todas las **enum** del programa de acceso global.

tipoDatos.h

Archivo que contiene todas las **typedef struct** del programa de acceso global.

data.h

Archivo que contiene todas las **arrays de datos** del programa de acceso global.

prototipos.h

Archivo que contiene todas las **definiciones de funciones** del programa..

variables.h

Archivo que contiene todas las **variables** del programa de acceso global.

ini.c

Archivo que contiene la inicialización de variables, carga de datos, etc.

util.c

Conjunto de funciones de utilidad disponibles en todo el programa.

mi_css.c

Definiciones del comportamiento css para la manipulación de los colores de las ventanas.

appWindow.c

Main del programa.

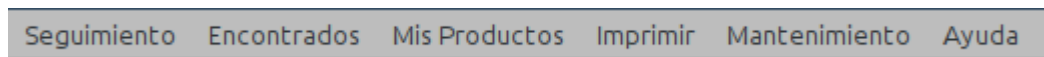
mensajesGui.c

Muestra un msgbox en pantalla.

aboutWindow.c

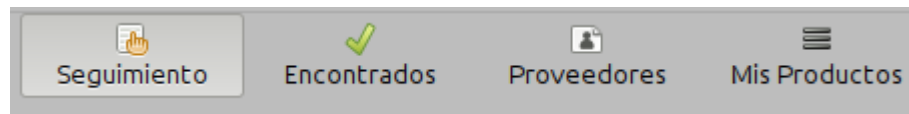
El about del programa.

menubar.c



Manejo del comportamiento del menubar.

tbutton.c



Manejo de las acciones de los toggleButton del menu principal.

.

toolbar_seguimiento.c



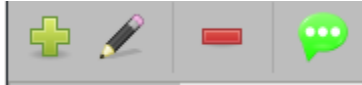
Manejo de las acciones del toolbar seguimiento, agregar, editar, buscar y borrar.

toolbar_encontrados.c



Manejo de las acciones del toolbar encontrados, visualizar informacion, grabar, borrar y abrir chat con el proveedor.

toolbar_mis_productos.c



Manejo de las acciones del toolbar productos, agregar, editar, borrar y abrir mensajes enviados por otros usuarios que han buscado el producto.

toolbar_proveedores.c



Manejo de las acciones del toolbar proveedores, abrir informacion del proveedor, buscar un proveedor dentro de la lista, calificar proveedor y eliminar de la lista un proveedor.

treeview_seguimiento.c

ID	Buscando	Nombre	Fabricante	Categoría	Unidad	Precio	Unidades Requeridas	Presentación	Estado

Manejo de la lista treeview de todos los seguimientos.

treeview_encontrados.c

Id	Nombre	Categoría	Stock	Precio	Estado	Municipio	Dirección	Teléfono	Empresa	Reputación

Manejo de la lista treeview de todos los resultados de búsquedas según el seleccionado en la lista seguimiento.

treeview_mis_productos.c

ID	Producto	Fabricante	Categoría	Stock	Precio	Unidad	Presentación	Visitas

Manejo de la lista treeview de todos los productos publicados por el usuario.

treeview_proveedores.c

ID	Producto	Reputación de la Comunidad	Reputación Local	Empresa	Categoría	Stock	Precio	Estado	Municipi

Manejo de la lista treeview de todos los proveedores grabados por el usuario.

barraNotificacion.c



Manejo de la barra encargada de mostrar el estado de la conexión online y de la disponibilidad del servidor local. El primer icono indica si el software a logrado identificarse en el sistema, el segundo muestra si programa ha logrado arrancar el servidor local para poder atender peticiones de otros clientes.

util_red.c

Conjunto de funciones de utilidad en el manejo de conexiones de red.

manejo_ip_local.c

Encargado de mantener la ip local actualizada en el servidor central de identificación.


servidorLocal.c

Tiene como responsabilidad mantener un hilo de escucha para las peticiones de otros clientes p2p. Maneja toda la lógica de la red p2p.

productoWindow.c

Producto

Información del Producto





Sector:  Sector [PM-Hogar->(Electronicos, Electricos, Mecanicos, Quimicos, Limpieza, etc.)] ✓

Nombre:

Descripción:

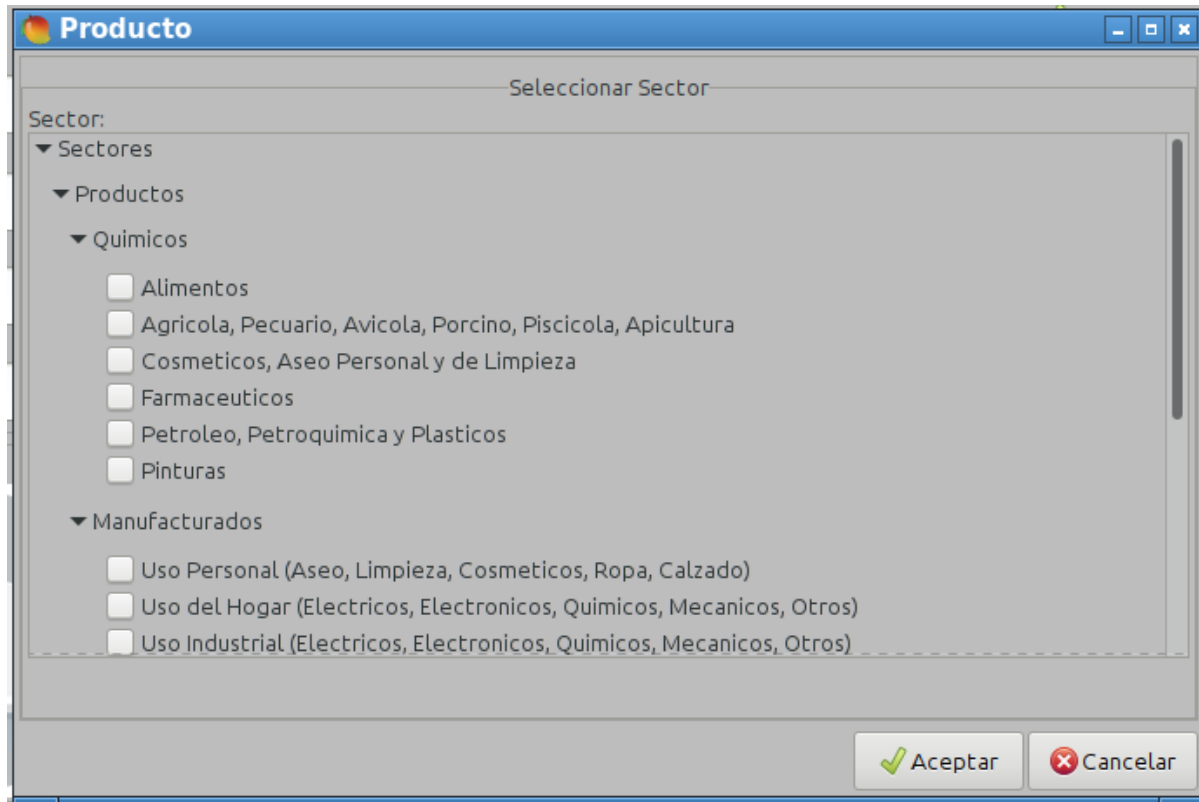
Precio de la unidad: Unidad: Stock: Presentación:

Fabricante: ☐ Fabricación ☒ Nuevo ☐ Transporte

Foto 1  Foto 2  Foto 3  Foto 4 

Maneja todo el código para la pantalla donde se carga los productos a publicar

selectWindow.c

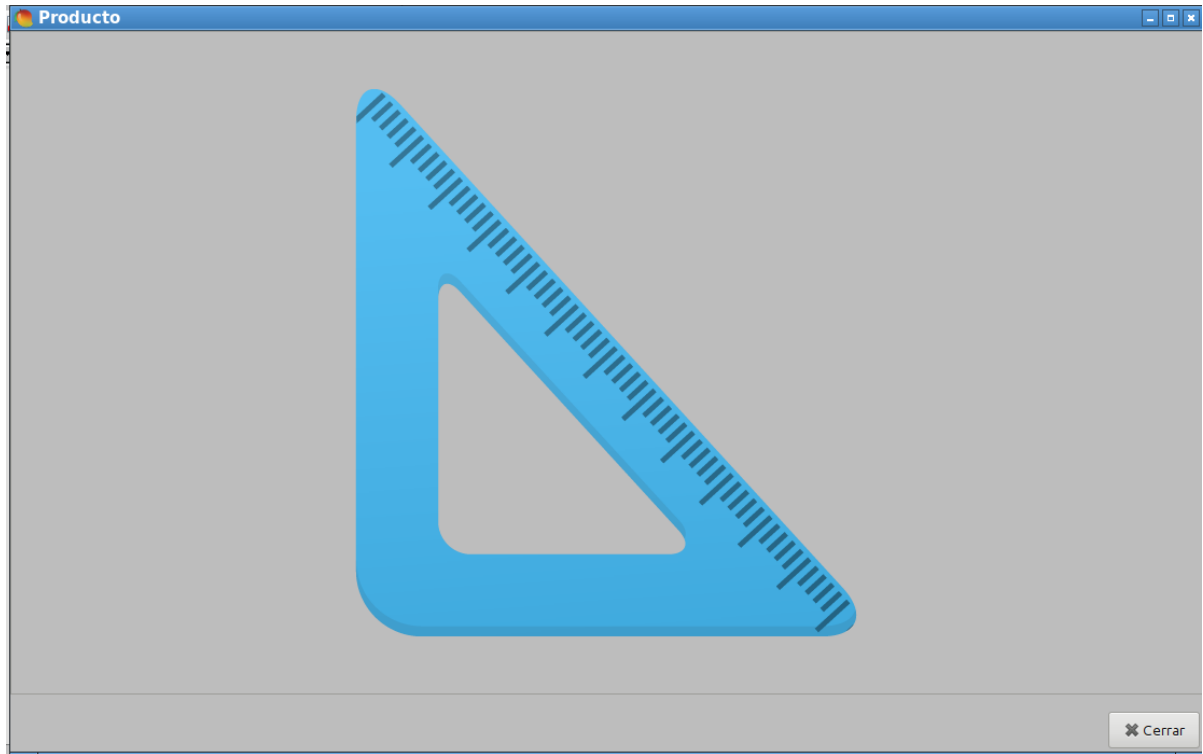


Maneja todo el código para la pantalla encargada de seleccionar el sector de un producto.

operacionesBit.c

Funciones para realizar operaciones a nivel de bit.

visorFotoWindow.c



Código que controla la ventana que muestra las imágenes en tamaño normal.

necesidadWindow.c

The screenshot shows a window titled 'Producto' with a standard Windows-style title bar. The main content area is titled 'Información sobre la Necesidad'. It contains several input fields and controls:

- Sector:** A button with a shopping cart icon and the text 'Sector'. A green checkmark is visible to its right.
- Nombre:** A text input field with a pencil icon to its right.
- Descripción:** A text input field with a pencil icon to its right.
- Precio de la unidad:** A text input field with a pencil icon to its right.
- Unidad:** A dropdown menu currently showing 'Sin Especificar'.
- Stock:** A numeric input field showing '0', with minus and plus buttons on either side.
- Presentación:** A dropdown menu currently showing 'Sin Especificar'.
- Fabricante:** A text input field with a pencil icon to its right.
- Fabricación:** A checkbox that is currently unchecked.
- Nuevo:** A checkbox that is currently checked.
- Transporte:** A checkbox that is currently unchecked.
- Estado:** A dropdown menu currently showing 'Todos'.
- Minicio:** A dropdown menu.

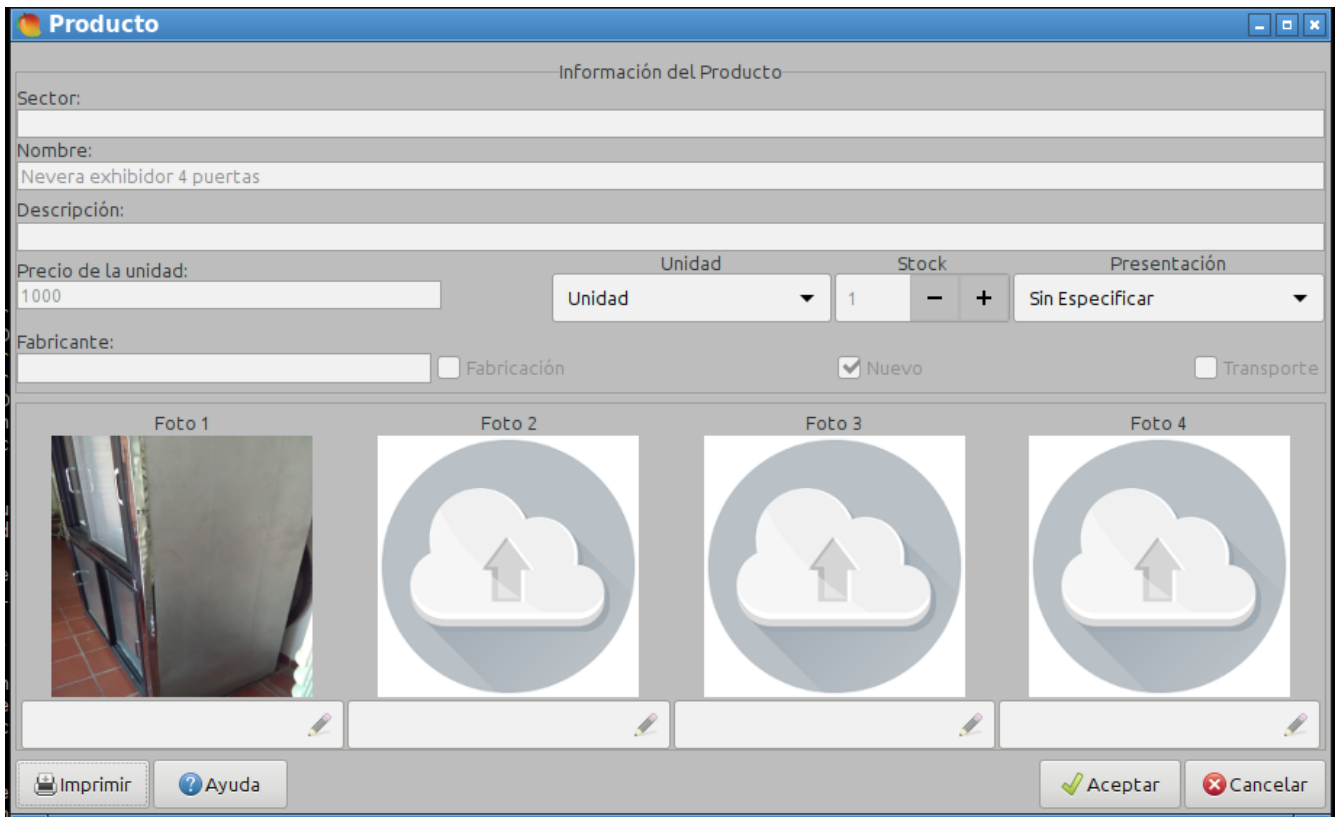
At the bottom of the window, there are four buttons: 'Imprimir' (with a printer icon), 'Ayuda' (with a question mark icon), 'Aceptar' (with a green checkmark icon), and 'Cancelar' (with a red X icon).

Código que controla la carga de una nueva necesidad.

busqueda.c

Contiene toda la lógica del motor de búsqueda de productos entre la red p2p.

encontradosWindow.c



Código que maneja toda la lógica en la ventana de encontrado.

Archivos xml con la configuración de las diferentes pantallas

AppWindowProductos.ui
EncontradosWindows.ui
NecesidadWindows.ui
ProductosWindows.ui
SectorWindows.ui
VisorFotoWindows.ui

Estructura de las carpetas

mango-productos (carpeta)

- **mango-bproductos** (archivo binario del programa)
- **bd** (carpeta)
 - **busqueda** (carpeta con los archivos de .dat de las ip de búsqueda)
 - **fotos-bajadas** (carpeta con los archivos .png de las imágenes de productos encontrados)
 - **productos** (carpeta con las imágenes de los productos cargados)
- **img** (carpeta con las imágenes del programa).

¿Cómo compilar?

Ejecutar el archivo `./linux-compile.sh` para compilar en linux.
Para compilar necesita tener las librerías GTK3 para desarrolladores.

Para efectos de probar se debe caracterizar de forma manual las fuentes antes de compilar

¿Cómo se hace eso?

Abrimos el archivo `macros.h`

En las líneas 14, 15 y 16 modificamos lo siguiente

```
#define NOMBRE_EMPRESA "Mango-Electrodomesticos"  
#define DIRECCION_EMPRESA "Parque Aragua"  
#define TELEFONO_EMPRESA "0414-365.21.98, 0426-987.25.41"
```

NOMBRE_EMPRESA → se debe colocar entre comillas el nombre de la empresa o personas
DIRECCION_EMPRESA → se debe colocar entre comillas la dirección de la empresa imaginaria
TELEFONO_EMPRESA → se debe colocar entre comillas los teléfonos de la empresa imaginaria

El largo deberá limitarse a lo que dice las macros

```
#define MAXDIRECCION 128  
#define MAXNOMBRE 64  
#define MAXFABRICANTE 40
```

```
#define MAXTELEFONO 64
```

Luego de modificado estos valores grabamos y cerramos el documento.

Ahora abrimos **ini.c**

En las lineas 33 y 34 modificamos lo siguiente

```
vg_registroUser->id = 1;  
vg_registroUser->codEstado = 4;  
vg_registroUser->codMunicipio = 1;
```

vg_registroUser->id contiene el valor id único en todo el sistema, para nuestro ejemplo es 1 (dejarlo así)

vg_registroUser->codEstado aquí se puede colocar sólo (4=Aragua), (8=Cojedes), (14=Merida) los demás no están implementados.

Acá se configura el estado donde se encuentra el registro del ofertante de productos

Luego de modificado estos valores grabamos y cerramos el documento.

Ahora abrimos **variables.h**

En la linea 38 modificamos lo siguiente

```
unsigned int vg_identificador = 1;
```

Este valor debe ser igual al id de vg_registroUser → id, para este caso es 1

Luego de modificado el valor grabamos y cerramos el documento.

Ahora abrimos **macros.h**

En la linea 61 modificamos lo siguiente

```
#define STR_IP_SERVER "127.0.0.1"
```

ESTO ES MUY IMPORTANTE (COMO TODO LO DEMÁS TAMBIÉN), LA VARIABLE **STR_IP_SERVER** TIENE LA IP DONDE ESTA UBICADO EL SERVIDOR CONTROL

Luego de todo esto se ha configurado el sistema procedemos a compilar

Ejecutar el archivo **./linux-compile.sh** para compilar en linux.

Recuerden que por cada nodo que se quiera ejecutar para probar la red p2p hay que hacer este procedimiento, es importante recordar que se debe ir aumentando el id de uno en uno, es decir primero id es 1, luego es 2, así sucesivamente.

Es importante copiar la información de quien es cada ID porque luego la vamos a necesitar cuando instalemos el servidor control.

Ejemplo hay que recordar que el ID=1 pertenece a Mango-Electrodomesticos, que es codEstado 4 y Municipio 1

y que la segunda compilación es ID=2 pertenece a Campo-Salle, que es codEstado 8 y Municipio 2

Por ejemplo:

Ahora vamos a configurar el servidor

Para compilar solamente ejecutar make

para arrancar ejecutar ./server-mp

desde otro terminal abrir un telnet en el puerto 3550

procedemos a registrar cada clientes

por ejemplo para registrar el ID=1 (los Ids son generados automaticamente cada vez que se registra un cliente)

130 crear user Mango-Electrodomesticos 4 1 15500

el ultimo parámetro se supone que es el puerto de escucha del cliente, es obvio que un sistema terminado este paso es automático entre los programas, por ahora tenemos que hacerlo nosotros.

Luego tecleamos 100 y otra vez 100 para salir

Asi lo hacemos sucesivamente por cada cliente.

Se debe tener cuidado en los pasos, porque sino esta bien configurado el sistema no funcionara.