



ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS



PROGRAMACION (TDSD214)

ASIGNATURA:

PROGRAMACION

PROFESOR:

Ing. Sergio Granizo MSc.

PERÍODO ACADÉMICO:

2024-B

PROYECTO

TÍTULO

DOCUMENTACIÓN

Integrantes:

- **BONE ZAMBRANO EMELY MAITE**
- **CHASI ROCHINA ALEXIS ALEXANDER**

OBJETIVO GENERAL

Desarrollar una aplicación utilizando el framework Qt Creator y el lenguaje de programación C++, que permita gestionar información mediante el almacenamiento de datos en archivos locales.

OBJETIVOS ESPECÍFICOS

- Adquirir conocimientos sobre el uso correcto de Qt Creator y las herramientas necesarias para el desarrollo de aplicaciones en C++ con interfaz gráfica.
- Implementar el manejo de archivos locales para el almacenamiento, lectura y actualización de la información generada por la aplicación.
- Aplicar buenas prácticas de programación para asegurar el correcto funcionamiento del sistema.

REQUERIMIENTOS

La aplicación tiene como objetivo gestionar los productos de una tienda de mascotas utilizando el lenguaje C++ y el framework Qt Creator. El sistema permite el acceso mediante autenticación de usuarios y ofrece un menú principal con las funcionalidades disponibles. La aplicación posibilita el registro, visualización, actualización, eliminación y búsqueda de productos, almacenando la información en archivos de texto locales. Los datos gestionados incluyen código, nombre, precio y stock del producto. Además, el sistema valida la información ingresada y muestra mensajes de alerta ante errores o acciones incorrectas.

MANUAL TÉCNICO PARA APLICACIÓN CRUD EN C++

1. INTRODUCCIÓN

1.1. Propósito del Documento

El documento busca detallar el proceso realizado durante la creación de la aplicación en el lenguaje C++ con Qt creator para gestionar los productos de una tienda de mascota, partiendo desde su instalación hasta la aplicación vista por el usuario.

1.2. Audiencia Destinada

Este manual está enfocado principalmente en estudiantes de desarrollo de software, con el propósito de brindar una base para la realización de sus proyectos, así como en profesionales del área, como docentes o trabajadores, que deseen analizar la lógica implementada.

1.3. Alcance del Proyecto

Aplicación desarrollada en C++ utilizando el framework Qt, orientada al registro y gestión de productos de mascotas mediante operaciones CRUD a través de interfaces gráficas, con almacenamiento persistente de la información en archivos de texto locales.

2. REQUISITOS DEL SISTEMA

2.1. Requisitos de Hardware

- **Procesador:** Intel Core i3 o superior.
- **Memoria RAM:** Mínimo 4 GB (se recomienda 8 GB para un rendimiento óptimo).
- **Almacenamiento:** Al menos 100 MB de espacio libre para el software y los archivos generados.
- **Pantalla:** Resolución mínima de 1280x720.

2.2. Requisitos de Software

- **Sistema Operativo:** Windows 10/11, Linux (Ubuntu 20.04 o superior) o macOS.
- **Entorno de Desarrollo:** Qt Creator 6.0 o superior.
- **Lenguaje de programación:** C++
- **Compilador:** MinGW 64-bit o cualquier compilador compatible con Qt.
- **Framework:** Framework de Qt5 o Qt6.

2.3. Dependencias

Archivos de datos (.txt): Almacenar los datos ingresados del usuario y los productos.

Qt Framework: Necesario para el desarrollo de la interfaz gráfica y manejo de eventos.

3. INSTALACIÓN Y CONFIGURACIÓN

Para el desarrollo de la aplicación se realizó la descarga e instalación del entorno de desarrollo Qt Creator, el cual permite la creación de aplicaciones en lenguaje C++ con interfaz gráfica.

Comenzamos accediendo al sitio web oficial de Qt, donde se descargó el instalador correspondiente al sistema operativo utilizado. Durante el proceso de instalación, se seleccionó la opción de Qt Creator junto con las herramientas necesarias para el desarrollo en C++ y el compilador MinGW.
para el Windows.

Una vez finalizada la instalación, se procedió a abrir Qt Creator y configurar el entorno de trabajo. Para ello, se verificó que el compilador estuviera correctamente reconocido por el sistema y que el kit de desarrollo de Qt se encontrara activo, lo cual permite compilar y ejecutar proyectos sin inconvenientes.

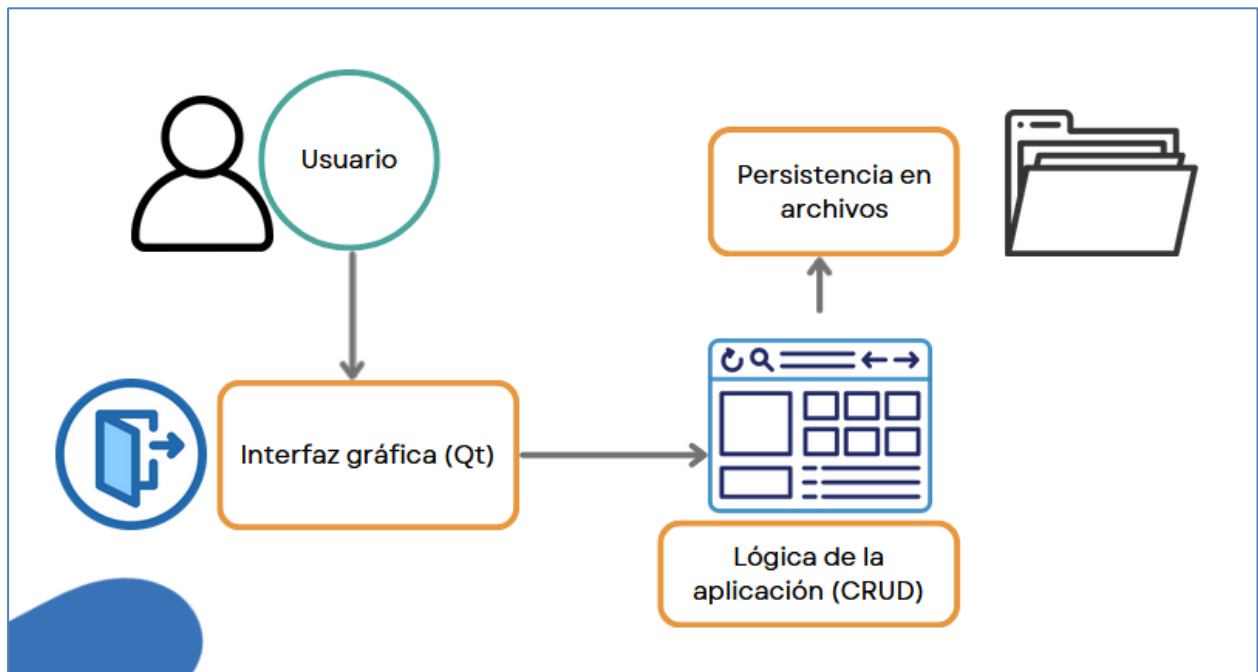
Finalmente, se creó un nuevo proyecto en Qt Creator seleccionando el tipo de aplicación correspondiente, lo que permitió comenzar con el diseño de la interfaz gráfica y la implementación de la lógica de la aplicación utilizando archivos .h, .cpp y .ui.

4. ARQUITECTURA DEL SISTEMA

4.1. Descripción General

Aplicación diseñada para el registro, lectura y edición de datos ingresados por el usuario almacenando la información en archivos.

4.2. Diagrama de Arquitectura



4.3. Diseño de Componentes

La aplicación está compuesta por varios componentes principales que permiten su correcto funcionamiento:

Autenticación: encargado de validar el acceso de los usuarios mediante un inicio de sesión.

Interfaz Gráfica: que gestiona las ventanas, formularios, botones y tablas con los que interactúa el usuario.

Gestión de la Información: se realiza mediante un componente de manejo de datos, el cual permite almacenar, consultar y modificar los registros utilizando archivos de texto.

Control de errores: Informa al usuario sobre acciones incorrectas o campos vacíos mediante ventanas emergentes.

5. IMPLEMENTACIÓN

5.1. Estructura del Código

Para la creación de la aplicación se hizo uso de archivos .h, en los cuales se declaran las clases y las funciones que posteriormente son implementadas en los archivos .cpp. Estos archivos se integran con los formularios .ui de Qt, permitiendo ejecutar correctamente cada una de las funcionalidades de la aplicación.

5.2. Descripción de Clases y Funciones

Para el desarrollo de la aplicación se utilizaron diversas clases del framework Qt que permiten la creación de interfaces gráficas y la gestión de datos.

Lista de funciones usadas del Qt:

- **QApplication:** Gestiona la ejecución principal de la aplicación y el ciclo de eventos.
- **QWidget:** Clase base de los componentes gráficos como ventanas y formularios.
- **QDialog:** Se utiliza para ventanas secundarias como el inicio de sesión.
- **QPushButton:** Permite ejecutar acciones mediante botones.
- **QLineEdit:** Campos de texto para el ingreso de datos.
- **QLabel:** Muestra textos descriptivos en la interfaz.
- **QTableWidget:** Presenta los datos en forma de tabla.
- **QMessageBox:** Muestra mensajes de alerta y confirmación.
- **QFile y QTextStream:** Permiten el manejo de archivos de texto para guardar y leer información.

Con el uso de cada una de esas funciones permitirá crear las funciones del CRUD a implementarse en la aplicación.

5.3. Manejo de Archivos

La información efectuada durante la ejecución de la aplicación se guardará en archivos.txt.

6. INTERFAZ GRÁFICA

6.1. Descripción de la Interfaz

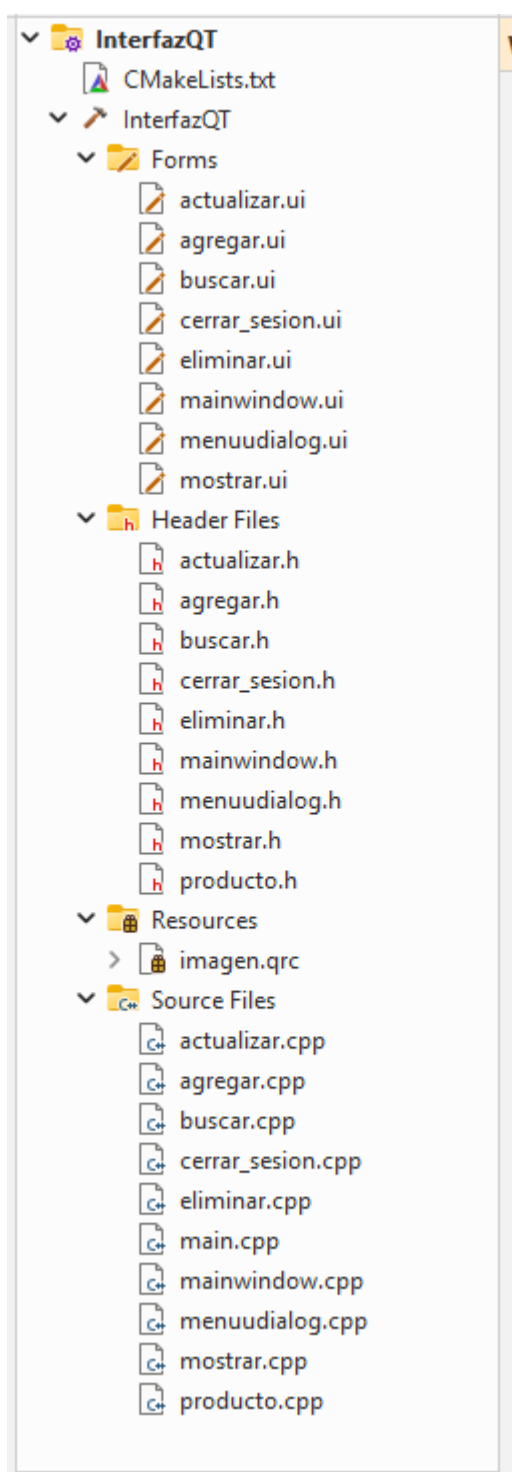
El usuario visualizará una ventana de inicio de sesión para que pueda ingresar los datos y verificar si los datos fueron correctos, si es así mostrar una ventana emergente de alerta, una vez dentro se desplegará una serie de opciones a escoger (agregar, mostrar, actualizar, buscar, eliminar y cerrar sesión) donde se desplegarán nuevas ventanas para efectuar las funciones requeridas y volver a la ventana del menú si así lo desea.

6.2. Descripción de Funcionalidades

Funcionalidad	Descripción
Agregar producto	Ingresa de datos para registrar un producto (código, producto, precio, stock) Y agregados en el archivo local generado.

Mostrar productos	Mostrar los productos registrados en el archivo.
Actualizar productos	Actualizar un producto por el código.
Eliminar Productos	Eliminar por medio del código.
Buscar Producto	Búsqueda de un producto por su nombre.
Cerrar Sesión	Finalizar la aplicación.

6.3 Estructura de la Interfaz Grafica



6.4 Diseño de la Interfaz Grafica



Dialog

Agregar Producto

Producto:

Código:

Precio:

Stock:



Guardar **Salir**

Dialog

Productos Agregados

	Código	Nombre	Precio	Stock
1	afds	Avion	9	46
2	ASL	Pan	9	8
3	csdf	camisas de	8	5
4	pan	queso	3	9



salir



A screenshot of a Windows-style dialog box titled "Dialog" with a close button (X) in the top right corner. The dialog has a light blue background and is titled "Actualizar Producto" in bold black text. It contains four input fields with blue borders, each preceded by a label: "Codigo:", "Producto:", "Precio:", and "Stock:". To the right of the "Codigo:" field is a blue paw print icon. To the left of the "Actualizar" button is another blue paw print icon. At the bottom are two buttons: "Actualizar" and "Salir".



A screenshot of a Windows-style dialog box titled "Dialog" with a close button (X) in the top right corner. The dialog has a light purple background and is titled "BUSCAR PRODUCTO" in bold black text. It contains four input fields with purple borders, each preceded by a label: "Nombre del Producto", "Codigo", "Precio", and "Stock". To the right of the input fields is a rectangular image showing a large brown dog, a parrot, and several cats. At the bottom are two buttons: "Buscar" and "Regresar".

7. USO DE LA APLICACIÓN

Al iniciar la aplicación, el usuario debe autenticarse mediante el ingreso de sus credenciales.

Una vez dentro del sistema, debes seleccionar una de las opciones disponibles en el menú principal para realizar las operaciones correspondientes.

Una vez terminada de ejecutar la aplicación retorna al menú principal hasta que el usuario decide cerrar sesión y finalizar el programa.

8. SOLUCIÓN DE PROBLEMAS

8.1. Problemas Comunes y Soluciones

Errores	Soluciones
Funciones definidas en los archivos .h pero no en los archivos .cpp	Hacer uso de la función definida o eliminarla
Problemas al abrir el proyecto por diferencias de versión de Qt por los archivos temporales del build.	Eliminar la carpeta build del proyecto y volver a compilar.

8.2. Mensajes de Error

Cada una de las funciones mostrar ventanas emergentes para cada uno de los errores que se puedan presentar (campos sin llenar, registros de datos incorrectos).

9. MANTENIMIENTO Y ACTUALIZACIÓN

9.1. Mantenimiento Regular

El mantenimiento de la aplicación primara la corrección de errores detectados durante su uso, así como la optimización del rendimiento para garantizar un funcionamiento estable y eficiente del sistema.

9.2. Actualización del Sistema

Las actualizaciones del sistema se enfocarán en la mejora del diseño de las interfaces gráficas, la incorporación de nuevas funcionalidades y la optimización del código base, con el fin de facilitar futuras modificaciones y ampliaciones de la aplicación.

10. REFERENCIAS BIBLIOGRAFICAS

Enfocadas en Qt – APA 7.ª edición)

Qt Company. (2024). *Qt documentation*. <https://doc.qt.io/>

ProgrammingKnowledge(2016, Abril 18) *Qt Tutorials For Beginners 10 - Simple Login app using QT* [Video]. YouTube.

https://www.youtube.com/watch?v=6_elY8O20I8&list=PLS1QulWo1RIZiBcTr5urECberTITj7gjA&index=10

Joseth David Guerrero. (2013, Mayo 6). *5. Pasar de ventana a ventana Qt5 C++* [Video].

YouTube. <https://www.youtube.com/watch?v=eg33SRk0WKU>

Ostyn Macias. (2014, Noviembre 12). *Como Insertar Imagen En Qt creator* [Video]. YouTube. <https://www.youtube.com/watch?v=eoIoxZf-Mks>

DuarteCorporation Tutoriales. (2014b, Agosto 22). *Tutorial Qt Creator - Archivo de recursos* [Video]. YouTube. <https://www.youtube.com/watch?v=u8xKE0zHLsE>