Documentación

Programa: [Acceso a datos - Excepciones , detección y tratamiento de errores]

Autor: [Blanca Pérez Chavarria]

Índice de contenidos

[**1. Documentación de usuario final: 2**](#_up2hbw8o1ml7)

[1.1. Guía de usuario 2](#_rws7jcualcvg)

[1.2. FAQ (Preguntas frecuentes) 2](#_9m00ugsblz7m)

[1.3. Solución de problemas comunes 2](#_qq37qk45b9ib)

[**2. Documentación técnica para desarrolladores 3**](#_j577qt8eveq)

[2.1. Introducción al software 3](#_6zdhgc3wuvm4)

[2.2. Arquitectura del sistema 3](#_xtes60ipx5sz)

[2.3. Estructura del código 3](#_52a122jpzn5d)

[2.4. Especificaciones de API 3](#_b5wg959l7e7a)

[2.5. Dependencias y configuraciones 3](#_1yz8bt1f8xhf)

[**3. Guías de instalación y despliegue 4**](#_p6n53w8oi8ll)

[3.1. Requisitos previos 4](#_u0knxtsluvjl)

[3.2. Instrucciones de instalación 4](#_31bghrjpgen1)

[3.3. Despliegue 4](#_kdki9dcsutog)

[3.4. Pruebas y verificación 4](#_8ln0ftoe7grc)

[**4. Mantenimiento y soporte 5**](#_5gc0ck381qq)

[4.1. Manejo de errores 5](#_ohoy19w00stk)

[4.2. Registro de cambios (changelog) 5](#_kbobhd6wy163)

[4.3. Soporte y contacto 5](#_ovlcepyqtf7o)

# **1. Documentación de usuario final:**

## **1.1. Guía de usuario**

*Ayuda: Incluye instrucciones sobre cómo utilizar el software. Usa capturas de pantalla, diagramas y ejemplos para hacerla más accesible.*

[Inserta aquí tu texto]

## **1.2. FAQ (Preguntas frecuentes)**

*Ayuda: Responde preguntas comunes para facilitar el soporte y resolver dudas rápidamente.*

[Inserta aquí tu texto]

## **1.3. Solución de problemas comunes**

*Ayuda: Describe errores comunes que puedan surgir y sus posibles soluciones.*

[Inserta aquí tu texto]

# **2. Documentación técnica para desarrolladores**

## **2.1. Introducción al software**

*Ayuda: Explica de qué se trata el software, su propósito, características principales y tecnologías utilizadas.*

[Inserta aquí tu texto]

## **2.2. Arquitectura del sistema**

*Ayuda: Incluye diagramas de arquitectura y una descripción de los componentes principales (como bases de datos, servidores, API, etc.) y cómo interactúan entre sí.*

[Inserta aquí tu texto]

## **2.3. Estructura del código**

*Ayuda: Explica cómo está estructurado el código. Incluye una descripción de directorios y archivos, así como los principales módulos y sus responsabilidades.*

En primer lugar hago las importaciones que utilizare mas adelante json, os , errno, tkinter.

Json para trabajar con ficheros json.

Os para trabajar con el sistema operativo.

Errno para capturar los errores.

Tkinter para crear una interfaz gráfica.

A continuación defino las clases con sus propiedades y métodos .

Creó una carpeta con el método os.makedirs dentro de un bloque try catch para gestionar los errores. Defino el método guardarCliente , que se encarga de añadir un cliente a la lista de clientes.

El método guardaDB recorre la lista de clientes , creando un fichero por cada uno , en el que el nombre de cada fichero es su id y el contenido es la información de cada cliente.

Creo unas variables para guardar la información que se escriba en los inputs de la interfaz gráfica.

Defino la estructura de la interfaz gráfica , con sus etiquetas de Entry , Label y Button .

Hay un botón que invoca al método guardaCliente y otro que invoca al método guardaDB.

Ventana.mainloop() me permite que no se cierre la interfaz gráfica.

## **2.4. Especificaciones de API**

*Ayuda: Describe las APIs o endpoints utilizados, con detalles de entrada y salida, ejemplos de solicitud y respuesta, y posibles códigos de error.*

[Inserta aquí tu texto]

## **2.5. Dependencias y configuraciones**

*Ayuda: Enumera las dependencias del software y cómo instalarlas. Proporciona detalles de configuración, como archivos de configuración, variables de entorno y bases de datos.*

Tener python instalado y instalar la libreria tk.

# **3. Guías de instalación y despliegue**

## **3.1. Requisitos previos**

*Ayuda: Lista los requisitos del sistema, como versiones específicas de software, hardware, y permisos necesarios.*

Tener instalada la versión 3 de python o superior . Con python viene el gestor de librerias pip .

Instalar la libreria tkinter.

## **3.2. Instrucciones de instalación**

*Ayuda: Proporciona una guía paso a paso para instalar el software en diferentes entornos (desarrollo, producción, etc.).*

Ir a la página oficial de python , descargar el instalador y seguir el asistente de instalación.

## **3.3. Despliegue**

*Ayuda: Explica cómo desplegar el software, ya sea en servidores locales o en la nube. Agrega procedimientos de actualización y rollback.*

El python de momento lo hemos ejecutado en local y lo ejecuto en mi ordenador. No necesita realizar ningún despliegue en este caso.

## **3.4. Pruebas y verificación**

*Ayuda: Detalla cómo ejecutar pruebas para asegurarse de que el software está funcionando correctamente después de la instalación.*

*Validar que las carpetas y datos se han creado correctamente.*

# **4. Mantenimiento y soporte**

## **4.1. Manejo de errores**

*Ayuda: Indica cómo capturar y manejar errores, incluyendo una lista de códigos de error y mensajes explicativos.*

Con el try - except , lo uso para capturar el error para verificar si la carpeta ya existe , si no tiene permisos es decir no se puede guardar o cualquier otro error.

## **4.2. Registro de cambios (changelog)**

*Ayuda: Detalla los cambios realizados en cada versión, incluyendo nuevas características, correcciones y mejoras.*

1 al 2

Creo dos clases cliente y producto

Creo un array de clientes , en el que tan solo tengo un cliente

3 al 9

Abro un archivo clientes.json en modo escritura y añado en formato json el ultimo elemento de la lista clientes.

Define un método para el cliente y devuelve texto en formato json para la serialización del objeto.

Y utiliza ese metodo al escribir el archivo.

Le pone un identificador al cliente. Y vuelco la información del ultimo que se añada.

Importa la libreria os y la crea un directorio para base de datos.

Con try except controla que la carpeta no de error al crearse , como seria que ya exista.

Importo import errno , para especializar mas los errores . Creo la variable continua para almacenar si se ha producido un error,

si se ha producido un error no ejecutara el if de vuelca el contenido en el fichero.

Recorre la lista de clientes y crea un fichero por cada uno de los clientes en la carpeta base de datos .

10 al 14

Importa la libreria tkinter para crear una interfaz gráfica y crear una ventana con sus label y sus entry y algun boton.

Crea un constructor para clientes con parametros .

Crea el metodo guardaCliente que añade un cliente al array de clientes .

Crea el metodo guardaBD la forma de guardar la información de los clientes en formato json.

Crea variables específicas para guardar la información que se introdujo en la interfaz de usuario.

## **4.3. Soporte y contacto**

*Ayuda: Proporciona información sobre cómo obtener soporte técnico, ya sea por correo, teléfono, o sistemas de tickets.*

## Para cualquier duda o consulta por favor contacte con blapecha@gmail.com.