

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

Unidad de aprendizaje: Diseño y Especificación de Software

Manual Técnico

Grupo: 008 **Hora:** N4 - N6 **Día:** Jueves **Salón:** 3206

Catedrático: M. C. Jorge Alejandro Lozano González

Equipo: 2

NOMBRE	MATRICULA	CARRERA
Omar Azael Lozano Escobar	2059459	IAS
Marisol Santiago Cruz	1978073	IAS
Karen Alicia Cuéllar Valdéz	2062860	IAS
Diego Eduardo Duran Rodriguez	2043558	IAS
Jose Angel Cardenas Contreras	1935156	IAS
Daniel Barrientos Flores	1998744	IAS
Ramiro Rodriguez Muñoz	2077583	IAS
Fabian Antonio Davalos Lozano	1926396	IAS
Sergio Roel Rodriguez Torres	1967483	IAS
César Alberto Cisneros Zúñiga	2009337	IAS
Axel Francisco Ramirez	1967851	IAS
Jesus Alejandro Martinez Garcia	1971837	IAS
Daniel Reyna Burnes	2063579	IAS
Jorge Velez Barron	1878028	IAS

Enero - Junio 2025

Febrero de 2025

CIUDAD UNIVERSITARIA, SAN NICOLÁS DE LOS GARZA, NUEVO LEÓN a Fecha
de Entrega: 16/05/2025

Contenido

INTRODUCCIÓN

En la actualidad, mantener nuestros documentos personales y oficiales de una manera en la que estén seguros y almacenados en el mismo sitio se ha convertido en una necesidad para tener acceso a ellos en cualquier momento y en cualquier lugar.

Para cubrir esta necesidad, desarrollaremos un proyecto mediante python que permitirá a los usuarios revisar, registrar y gestionar sus documentos oficiales y personales, como pasaporte, identificación oficial, tarjetas de crédito o débito y licencia de conducir.

Ademas, contará con un sistema de notificaciones el cual estará alertando a cada usuario cuando sus documentos estén a punto de expirar, esta notificación se realizará con al menos 30 días de anticipación y se les estarán proporcionando enlaces que los redirigirán a portales oficiales donde podrán realizar su renovación o ver el proceso para realizarla eliminando la necesidad de tener que buscar manualmente esta información.

Al desarrollar el sistema seguimos principios de diseño eficiente, asegurando que el uso de esta aplicación se escalable, segura y fácil de mantener. Para lograr esto implementamos estrategias de diseño de sistemas que nos van a permitir que la estructura del código sea fácil de entender, que esté de manera clara y que en el futuro permita o facilite actualizaciones o mejoras. También, se hizo el diseño de una base de datos optimizada que garantice un acceso fácil y rápido a la información, quitando redundancias y mejorando el rendimiento del sistema. La construcción del software fue elaborada con mucho cuidado y asegurando que el sistema se pudiera adaptar a diferentes necesidades y soportar un crecimiento sin afectar su desempeño.

OBJETIVO

El objetivo principal de este proyecto es proporcionar a los usuarios un acceso sencillo, rápido y seguro a sus documentos en el momento que lo requieran. Para ello, se ha desarrollado una interfaz intuitiva que permite una navegación fluida, se han optimizado los tiempos de respuesta del sistema para garantizar agilidad en el acceso, y se han implementado medidas de seguridad que protegen la información almacenada. Todo el entorno ha sido diseñado para asegurar la disponibilidad constante de los documentos.

DESCRIPCIÓN DEL SISTEMA

Nuestro sistema es una aplicación desarrollada en Python que permite a los usuarios almacenar, gestionar y consultar documentos personales y oficiales desde una única plataforma segura y accesible. Entre las funcionalidades principales se incluyen el registro de documentos como pasaporte, identificaciones oficiales, tarjetas bancarias y licencias de conducir, así como la generación de notificaciones automáticas que alertan sobre su próxima fecha de vencimiento con al menos 30 días de anticipación.

COMPONENTES PRINCIPALES

REQUERIMIENTOS DEL SISTEMA

Requerimientos mínimos del sistema

- Procesador: Core
- Memoria RAM: 1Gigabytes (GB)
- Disco Duro: 250 GB

Requerimientos mínimos del software

- Sistema Operativo: Windows Actual
- Python 3.13.3
- Visual Studio Code instalado (última versión disponible), se recomienda tener ya instalada la extensión con Python para facilitar la escritura, depuración y ejecución de Python.
- Paquetes Python requeridos, en este caso; pyinstaller, llama_cpp-python y tinyllama.
- Navegador web moderno (Google Chrome, Firefox, Edge).

HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO

Para la realización de este sistema nos basamos en las necesidades que se tienen hoy en día en el ámbito de nuestros documentos oficiales

Una de los requerimientos del sistema es que sea fácil de navegar para el usuario a través de su interfaz., es por este motivo que este sistema tiene las siguientes características:

- **Python 3.13.3**
- **Paquetes Python requeridos (pyinstaller, llama_cpp-python y tinyllama)**
- **Visual Studio Code**

Python 3.13.3



Python 3.13.3 es una versión actualizada del lenguaje de programación Python que sirve para desarrollar aplicaciones de todo tipo, como sitios web, programas de escritorio, herramientas automatizadas, inteligencia artificial, entre otros. Esta versión en particular ofrece mejoras en la velocidad de ejecución, facilita el trabajo con código en tiempo real gracias a un entorno interactivo más visual y amigable, y permite aprovechar mejor el rendimiento de las computadoras modernas que usan varios núcleos. Además, mejora la detección y visualización de errores, lo que hace que sea más fácil programar, probar y corregir aplicaciones. En resumen, Python 3.13.3 es una herramienta poderosa, moderna y eficiente para programadores que buscan construir soluciones de manera rápida y clara.

Paquetes Python requeridos (pyinstaller, llama_cpp-python y tinyllama)

Pyinstaller

Este paquete permite convertir aplicaciones desarrolladas en Python en ejecutables independientes (.exe en Windows), lo que facilita su distribución sin necesidad de que el usuario final tenga Python instalado.

llama_cpp-python

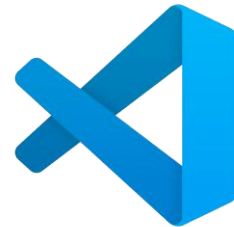
Es una interfaz en Python para ejecutar modelos de lenguaje LLaMA (como los de Meta) directamente en CPU o GPU locales. Permite realizar tareas de procesamiento de lenguaje natural (NLP) de forma eficiente, sin depender de servicios en la nube.

Tinyllama

Es un modelo de lenguaje pequeño y ligero, basado en la arquitectura LLaMA, diseñado para ser rápido y eficiente en dispositivos con recursos limitados. Se utiliza para tareas básicas de comprensión y generación de texto dentro de la aplicación.

Visual Studio Code

Visual Studio Code, al que conocemos también como VSCode, es un editor de código para programadores gratuito, de código abierto y multiplataforma. Está desarrollado por Microsoft, una compañía con una dilatada experiencia en la creación de IDEs (entornos de desarrollo integrados), que ha conseguido plasmar su larga tradición en el sector para ofrecer una herramienta ligera y práctica que la comunidad ha adoptado en masa.



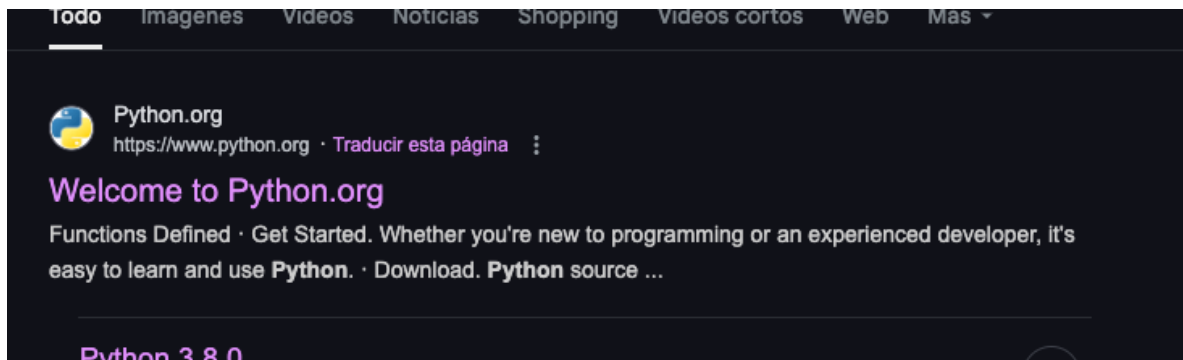
Básicamente, se trata de un editor de código. Esto quiere decir, una herramienta que nos permite editar el texto plano de los archivos de código para programación. Sin embargo, detrás de esa necesidad inicial, Visual Studio Code se puede configurar para realizar muchos tipos de tareas, incluso más allá de abrir y escribir simples archivos de texto.

Así pues, VSCode es una herramienta extremadamente versátil, capaz de facilitar muchas tareas de la programación, llegando (con las correspondientes extensiones) a funcionar casi al nivel de un IDE. Ahora resumimos algunas de sus principales funciones.

INSTALACIÓN DE REQUERIMIENTOS

INSTALACIÓN DE *Python 3.8*

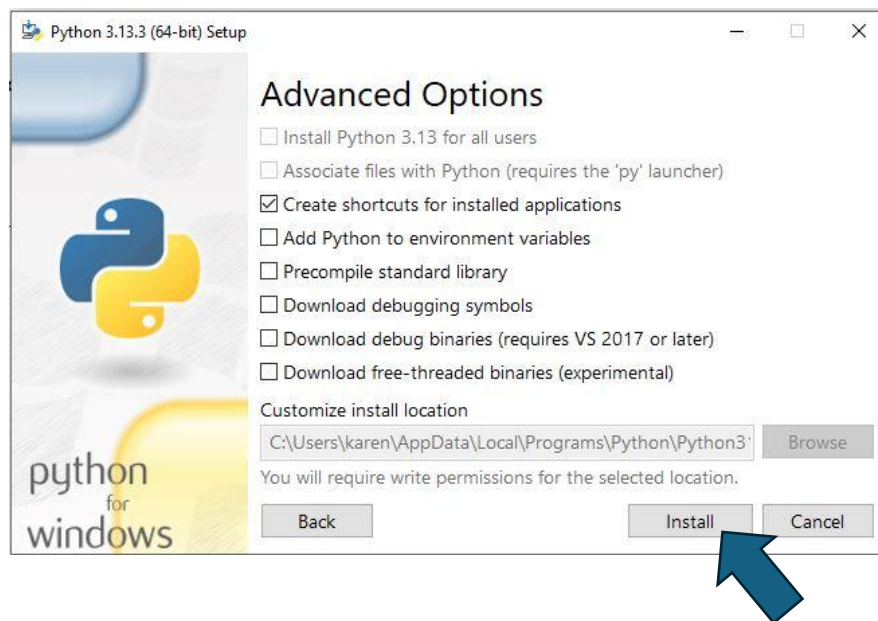
1. Abrimos el navegador Web de nuestra preferencia y buscamos sitio oficial de Python



2. Nos vamos al apartado de "Download" (Descargas) y le damos clic a Download Python 3.13.3



3. Ya que lo tengamos descargado en nuestro dispositivo, lo abrimos y le damos clic en "Install Now" y seguimos las instrucciones.



Verificamos la instalación de Python 3.13.3


1. Abrimos el símbolo del sistema (CMD)

```
cmd. Símbolo del sistema
Microsoft Windows [Versión 10.0.19045.5131]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\karen>
```

2. Ejecutamos el código python --version

Si no hay errores debera aparecernos de la siguiente manera:

 Símbolo del sistema

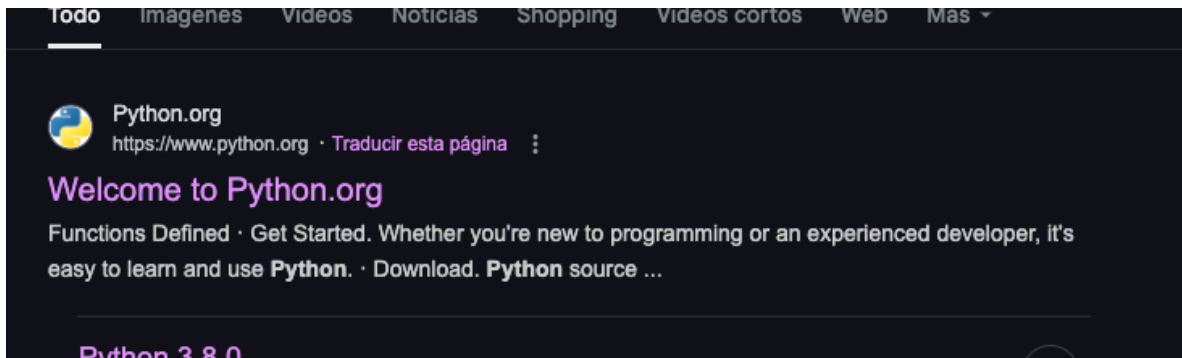
```
Microsoft Windows [Versión 10.0.19045.5131]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\karen>python --version
Python 3.13.3

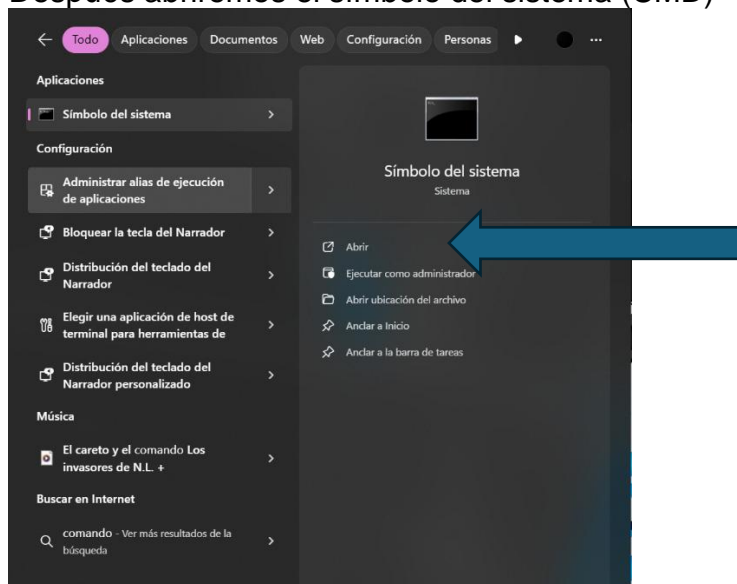
C:\Users\karen>
```

INSTALACIÓN DE PyInstaller

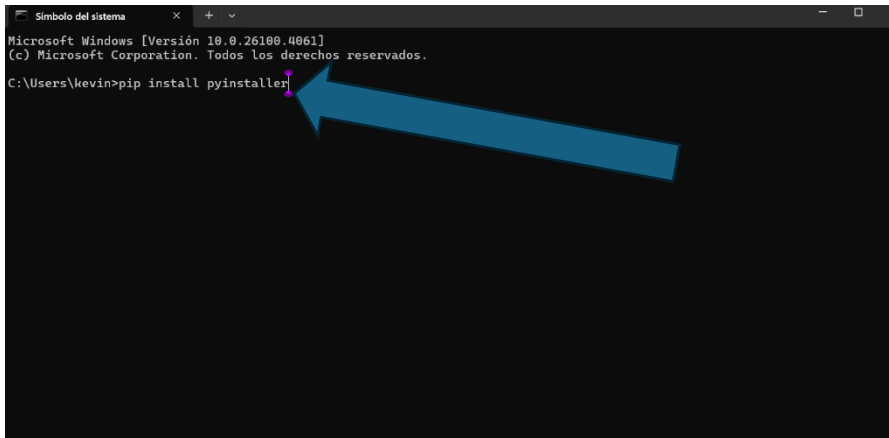
1. Para poder instalar el PyInstaller, es necesario tener instalado python



2. Después abriremos el símbolo del sistema (CMD)



3. Despues ingresamos el código “pip install pyinstaller



```
Simbolo del sistema
Microsoft Windows [Versión 10.0.26100.4061]
(c) Microsoft Corporation. Todos los derechos reservados.
C:\Users\kevin>pip install pyinstaller
```

4. Una vez que este instalado les tendrá que salir de la siguiente manera

```
C:\Users\kevin>pip install pyinstaller
Collecting pyinstaller
  Using cached pyinstaller-6.13.0-py3-none-win_amd64.whl.metadata (8.3 kB)
Collecting setuptools>=42.0.0 (from pyinstaller)
  Using cached setuptools-80.7.1-py3-none-any.whl.metadata (6.6 kB)
Collecting altgraph (from pyinstaller)
  Using cached altgraph-0.17.4-py2.py3-none-any.whl.metadata (7.3 kB)
Collecting pefile<=2024.8.26,>=2022.5.30 (from pyinstaller)
  Using cached pefile-2023.2.7-py3-none-any.whl.metadata (1.4 kB)
Collecting pywin32-ctypes>=0.2.1 (from pyinstaller)
  Using cached pywin32-ctypes-0.2.3-py3-none-any.whl.metadata (3.9 kB)
Collecting pyinstaller-hooks-contrib>=2025.2 (from pyinstaller)
  Using cached pyinstaller_hooks_contrib-2025.4-py3-none-any.whl.metadata (16 kB)
Collecting packaging>=22.0 (from pyinstaller)
  Using cached packaging-25.0-py3-none-any.whl.metadata (3.3 kB)
Using cached pyinstaller-6.13.0-py3-none-win_amd64.whl (1.4 MB)
Using cached packaging-25.0-py3-none-any.whl (66 kB)
Using cached pefile-2023.2.7-py3-none-any.whl (71 kB)
Using cached pyinstaller_hooks_contrib-2025.4-py3-none-any.whl (434 kB)
Using cached pywin32-ctypes-0.2.3-py3-none-any.whl (30 kB)
Downloading setuptools-80.7.1-py3-none-any.whl (1.2 MB)
  1.2/1.2 MB 181.3 kB/s eta 0:00:00
Downloading altgraph-0.17.4-py2.py3-none-any.whl (21 kB)
Installing collected packages: altgraph, setuptools, pywin32-ctypes, pefile, packaging, pyinstaller-hooks-contrib, pyinstaller
Successfully installed altgraph-0.17.4 packaging-25.0 pefile-2023.2.7 pyinstaller-6.13.0 pyinstaller-hooks-contrib-2025.4 pywin32-ctypes-0.2.3 setuptools-80.7.1

[notice] A new release of pip is available: 25.0.1 -> 25.1.1
[notice] To update, run: python.exe -m pip install --upgrade pip
C:\Users\kevin>
```

INSTALACIÓN DE llama_cpp-python

Requisitos Previos

Antes de comenzar con la instalación, asegúrese de tener lo siguiente instalado en su sistema:

- Python 3.8 o superior
- pip actualizado (python -m pip install --upgrade pip)
- Git
- Compilador C/C++ (ej. gcc o clang)
- CMake (recomendado: versión 3.22+)

Recomendación:

Se recomienda usar un entorno virtual para evitar conflictos de dependencias:

```
python -m venv llama_env
```

source llama_env/bin/activate # En Windows: llama_env\Scripts\activate

Instalación de llama-cpp-python

llama-cpp-python es un binding de Python para la librería llama.cpp, que permite ejecutar modelos LLM como LLaMA en CPUs o GPUs.

1. Instalación desde PyPI:

```
pip install llama-cpp-python
```

Nota: Esta opción instala una versión precompilada. Para mayor rendimiento o compatibilidad con GPU, se recomienda la compilación desde el código fuente.

2. Instalación desde código fuente (recomendado para entornos personalizados):

```
pip install cmake
```

```
git clone https://github.com/abetlen/llama-cpp-python.git
```

```
cd llama-cpp-python
```

```
pip install .
```

Opciones adicionales:

Para soporte de GPU o configuración avanzada, consulte la documentación oficial:

<https://github.com/abetlen/llama-cpp-python>

INSTALACIÓN DE tinyllama

Descarga e instalación de TinyLlama

TinyLlama es un modelo LLM de código abierto optimizado para ejecutarse eficientemente en CPUs y dispositivos con recursos limitados.

1. Descargar el modelo TinyLlama:

```
pip install huggingface_hub
```

```
from huggingface_hub import snapshot_download
```

```
snapshot_download(repo_id="TinyLlama/TinyLlama-1.1B-Chat-v1.0")
```

Esto descargará el modelo completo en el directorio predeterminado del cache de Hugging Face.

2. Convertir el modelo al formato GGUF (si es necesario):

Si el modelo no está en formato GGUF (requerido por llama.cpp), debe convertirse usando herramientas como transformers + convert.py, o buscar directamente una versión ya convertida:

Buscar modelos GGUF en Hugging Face:

<https://huggingface.co/TheBloke/TinyLlama-1.1B-Chat-v1.0-GGUF>

Descargue la versión en .gguf adecuada para su sistema (ej. q4_K_M.gguf para CPUs).

Cargar y ejecutar TinyLlama con llama-cpp-python

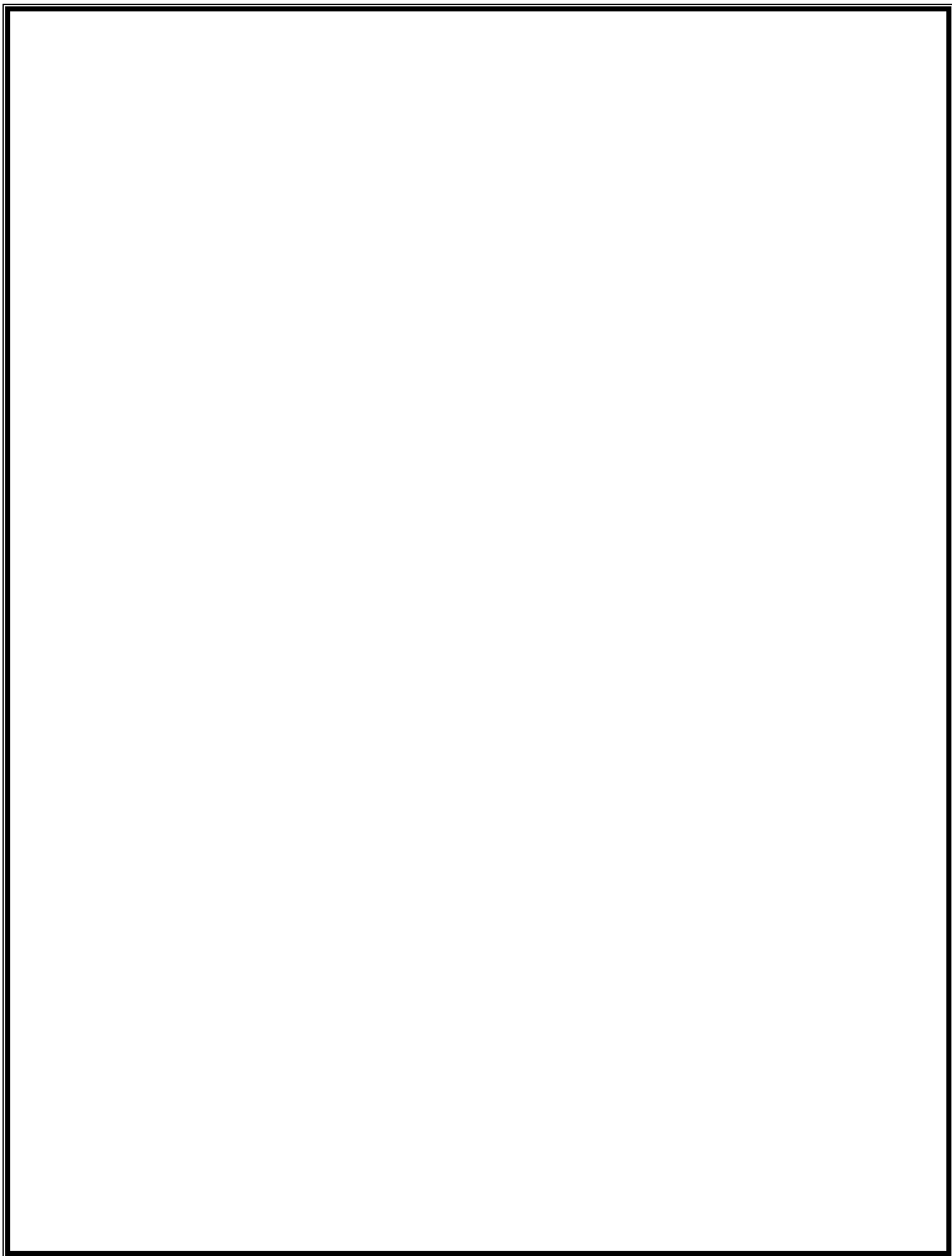
Una vez descargado el archivo .gguf del modelo:

```
from llama_cpp import Llama
```

```
llm = Llama(model_path="./TinyLlama-1.1B-Chat-v1.0.q4_K_M.gguf")
```

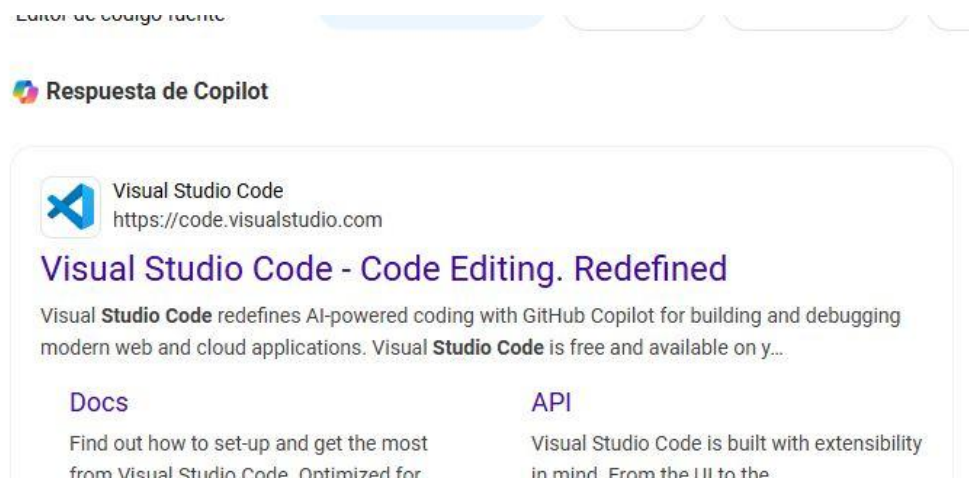
```
output = llm("¿Cuál es la capital de Francia?", max_tokens=64)
```

```
print(output)
```

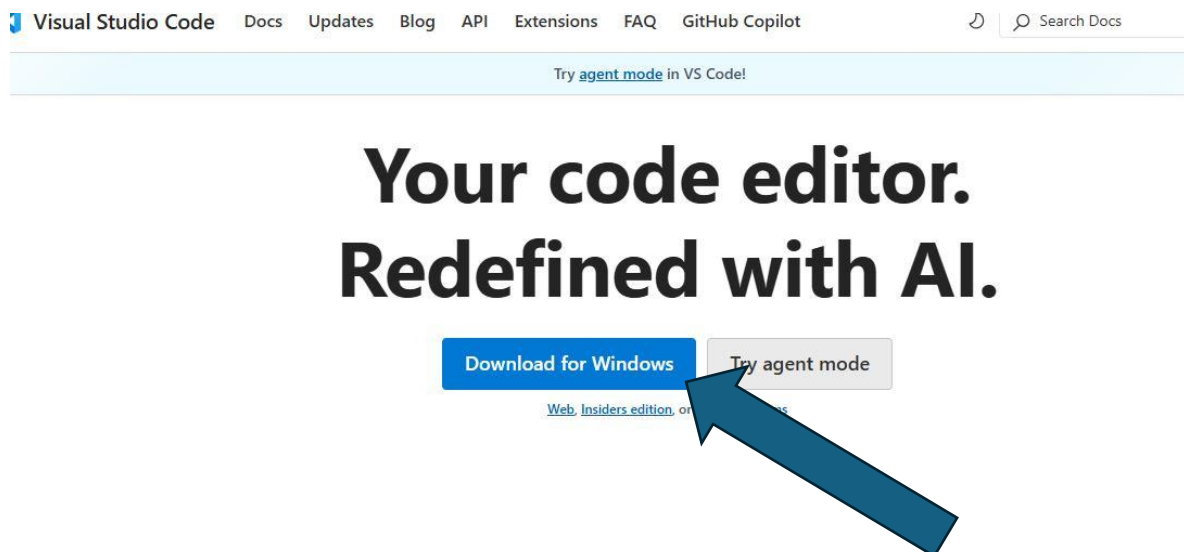


INSTALACIÓN DE VISUAL STUDIO CODE

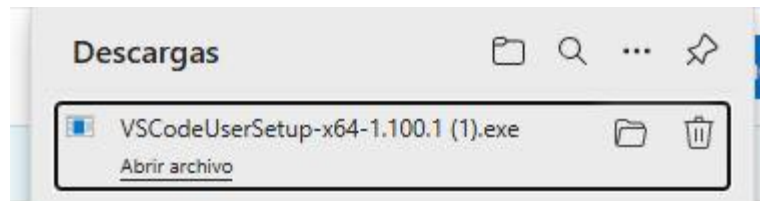
1. Ir al navegador Web de nuestra preferencia y nos dirigimos a la pagina oficial de Visual Studio Code



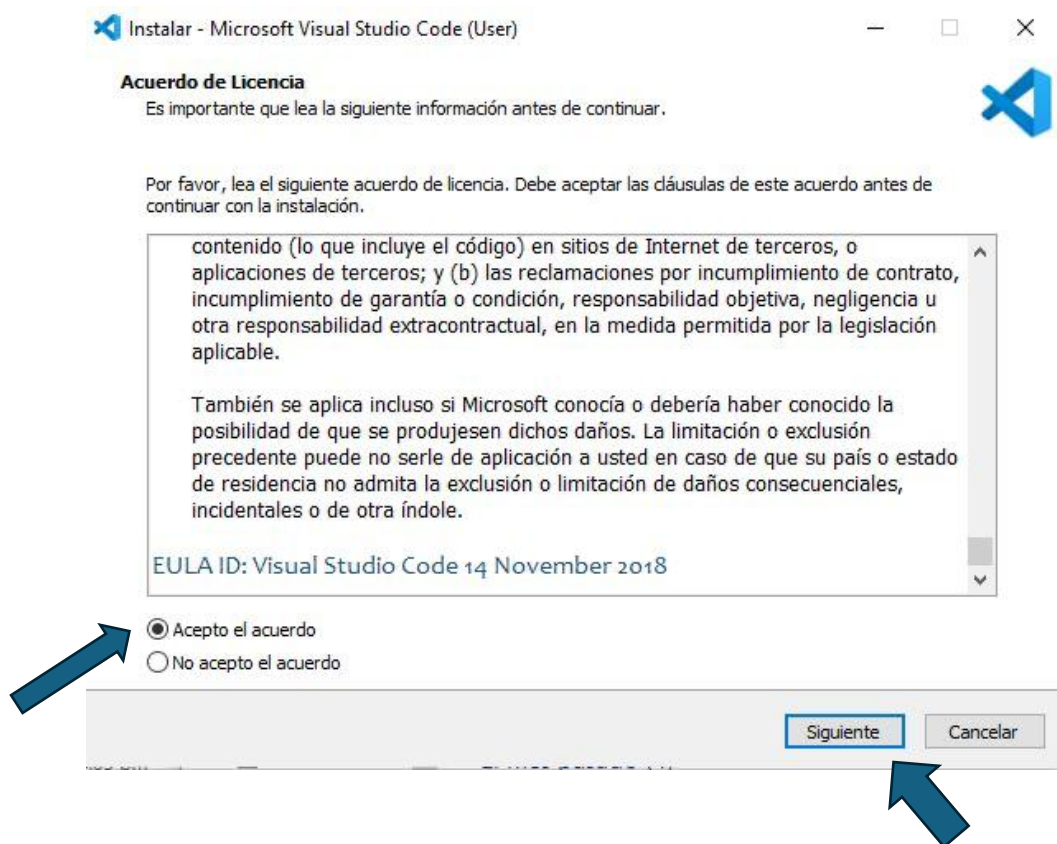
2. Le damos clic en descargar para Windows



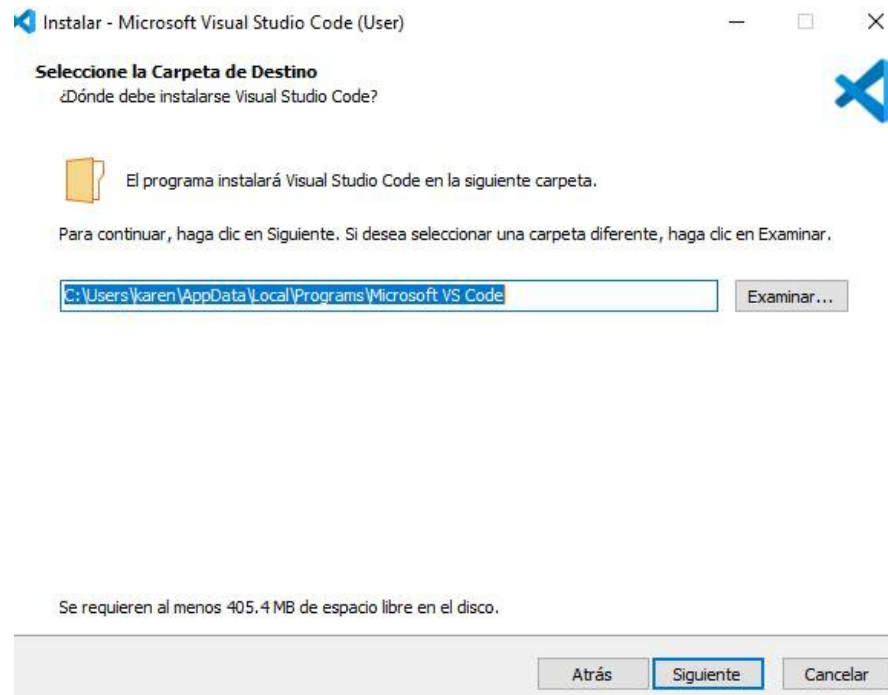
3. Ejecutamos el instalador (doble clic a la descarga)



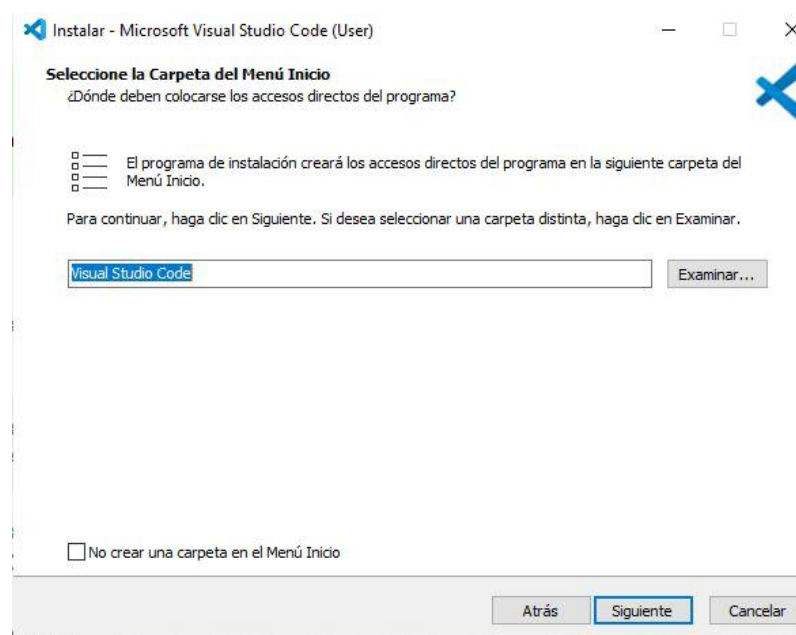
4. Aceptamos las licencias y le damos siguiente



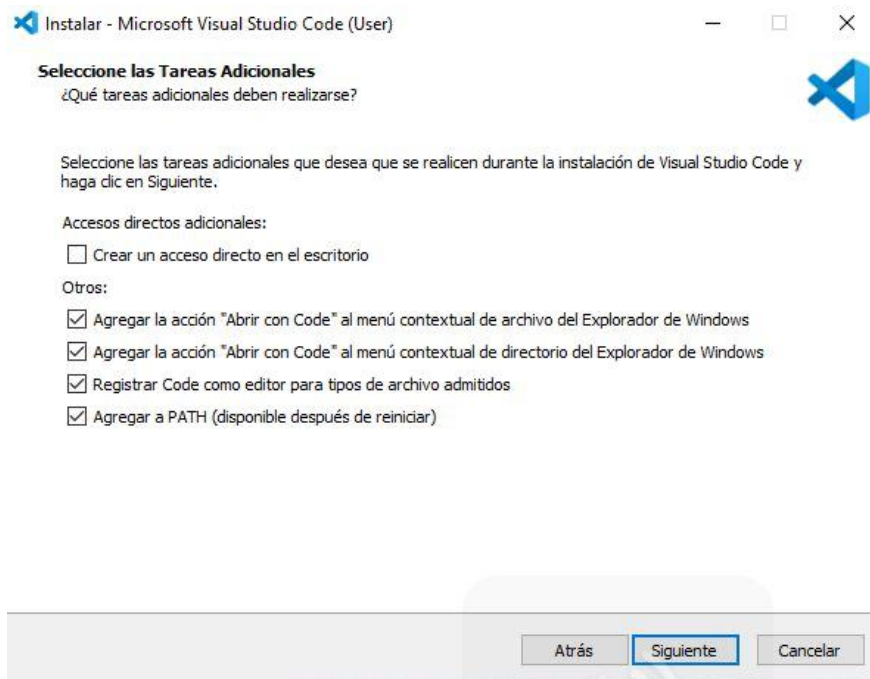
5. En la pantalla siguiente aparece la Carpeta de Destino, guardamos el programa en donde deseemos aunque nos da un destino por default, podemos examinar y navegar por las demas carpetas de nuestro dispositivo



6. A continuación nos aparecerá la Carpeta del Menú Inicio y ahí podremos guardarla con el nombre que deseemos, sin embargo es recomendable hacerlo con el nombre de Visual Studio Code.



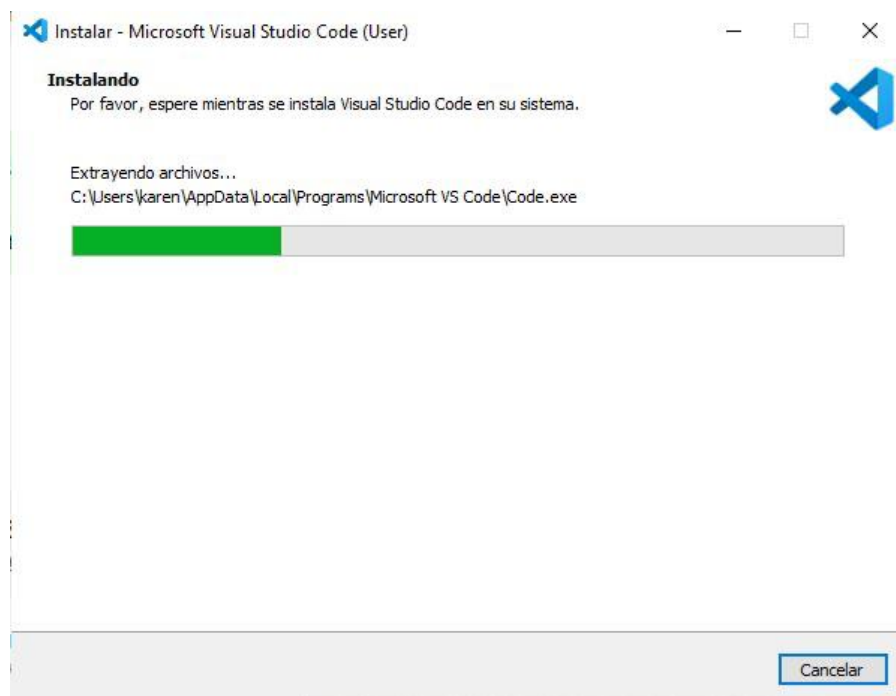
7. Por último la pantalla de Tareas Adicionales, activamos las siguientes opciones y le damos siguiente



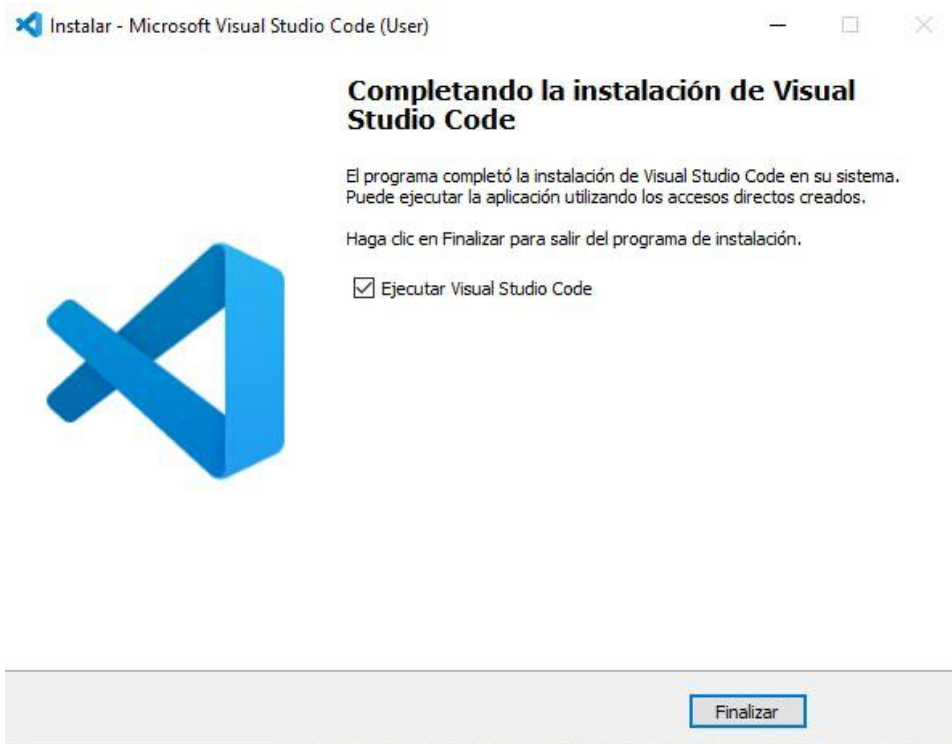
Para este punto el programa esta listo para instalar, en el siguiente apartado nos mostrara un resumen del paso 5 a 7.

- Como lo es la Carpeta de Destino donde se instalará Visual Studio Code.
- Carpeta del menú de inicio que creamos o aceptamos (como Visual Studio Code).
- Las tareas adicionales que seleccionamos (por ejemplo: añadir al PATH, opciones del menú contextual, registro como editor para tipos de archivos)
- Y algunas otras opciones de instalación

8. Ya que hayamos revisado toda la información sea correcta acorde a nuestras especificaciones, le damos en instalar



9. Visual Studio Code se instaló correctamente, le damos clic en Finish



Verificación de Visual Studio Code

Vamos a CMD y ponemos como código "code --version", si nos aparece lo siguiente está correctamente instalado

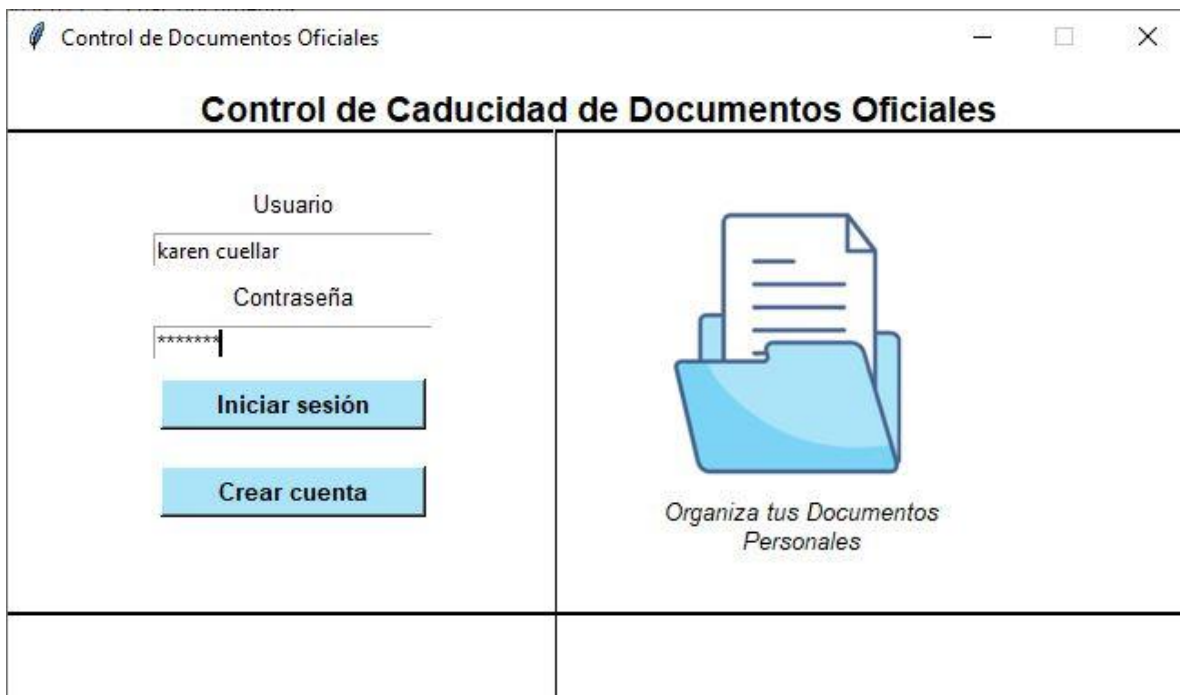
```
C:\> Símbolo del sistema
Microsoft Windows [Versión 10.0.19045.5131]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\karen>code --version
1.100.1
91fa95bccb027ece6a968589bb1d662fa9c8e170
x64

C:\Users\karen>
```

Inicio de Sesión

El inicio de sesión es la parte donde el usuario deberá ingresar los datos solicitados para poder tener acceso al sistema, es necesario contar con todos los datos solicitados o de lo contrario se puede crear una cuenta. En este caso los datos solicitados son: Usuario y Password (contraseña), se incluyen 2 botones en la parte inferior que son Iniciar Sesión y Crear Cuenta



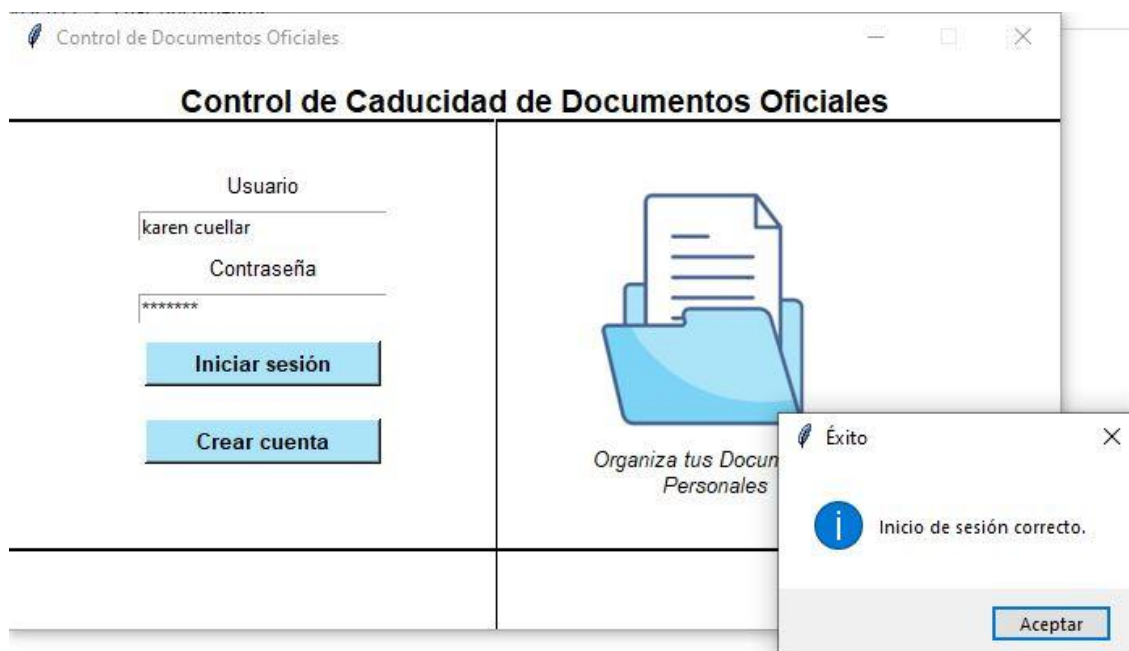
The screenshot shows a web application window titled "Control de Documentos Oficiales". The main heading is "Control de Caducidad de Documentos Oficiales". On the left side, there is a login form with two input fields: "Usuario" (containing "karen cuellar") and "Contraseña" (containing "*****"). Below these fields are two buttons: "Iniciar sesión" and "Crear cuenta". On the right side, there is a graphic of a blue folder with a document inside, and the text "Organiza tus Documentos Personales" below it.

Pasos a seguir para el inicio de sesión:

- 1.-Abrir la aplicación desde su ordenador
- 2.- Ingresar el Usuario en el campo correspondiente
- 3.- ingresar el Password (contraseña) en el campo correspondiente
- 4.- hacer click en el icono de Iniciar Sesión que está en la parte inferior para poder ingresar

Verificar que los datos ingresados sean correctos para evitar complicaciones

Posteriormente le damos clic al boton de iniciar sesion y nos aparecera un mensaje donde en estar correctos los datos nos aparecera “Inicio de Sesión correcta”



Despues entraremos al menu principal donde al costado izquierdo se nos despliegan cuatro apartados los cuales son

- Subir Documentos: donde podremos subir nuestros documentos oficiales
- Documentos cargados: donde podremos ver los documentos que hemos cargado
- Chatbot documental: es un asistente virtual por si necesitamos ayuda en el proceso
- Documentos vencidos: donde podremos ver nuestros documentos ya vencidos

Menú Principal[Subir
Documentos](#)[Documentos
Cargados](#)[Chatbot
Documental](#)[Documentos
Vencidos](#)

En el apartado de subir documentos podemos poner el nombre del documento, fecha de vencimiento donde podremos seleccionar la fecha en la cual se expira nuestro documento, URL del trámite oficial (aquí podemos poner la página oficial del gobierno), un botón de elegir archivo y se nos abrirá nuestros archivos para subir nuestro respectivo archivo, en el botón ver el documento y por último guardar el documento en el sistema

Subir Documentos

Nombre del documento:

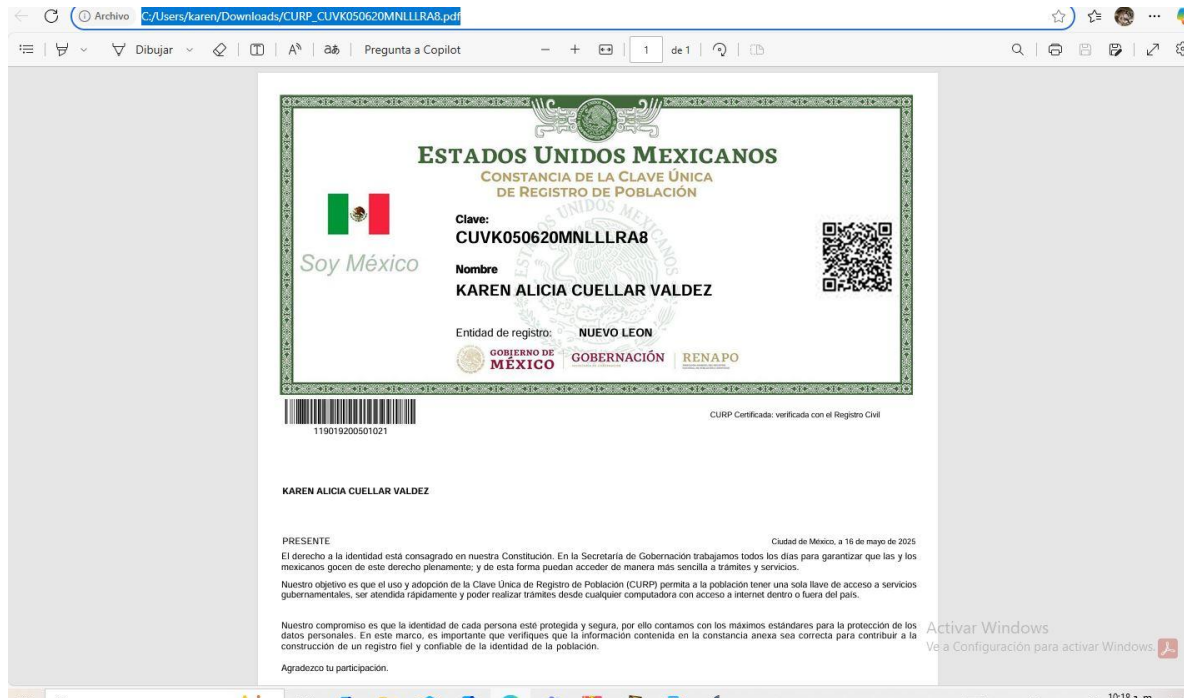
Fecha de vencimiento:

2025-05-16 ▾

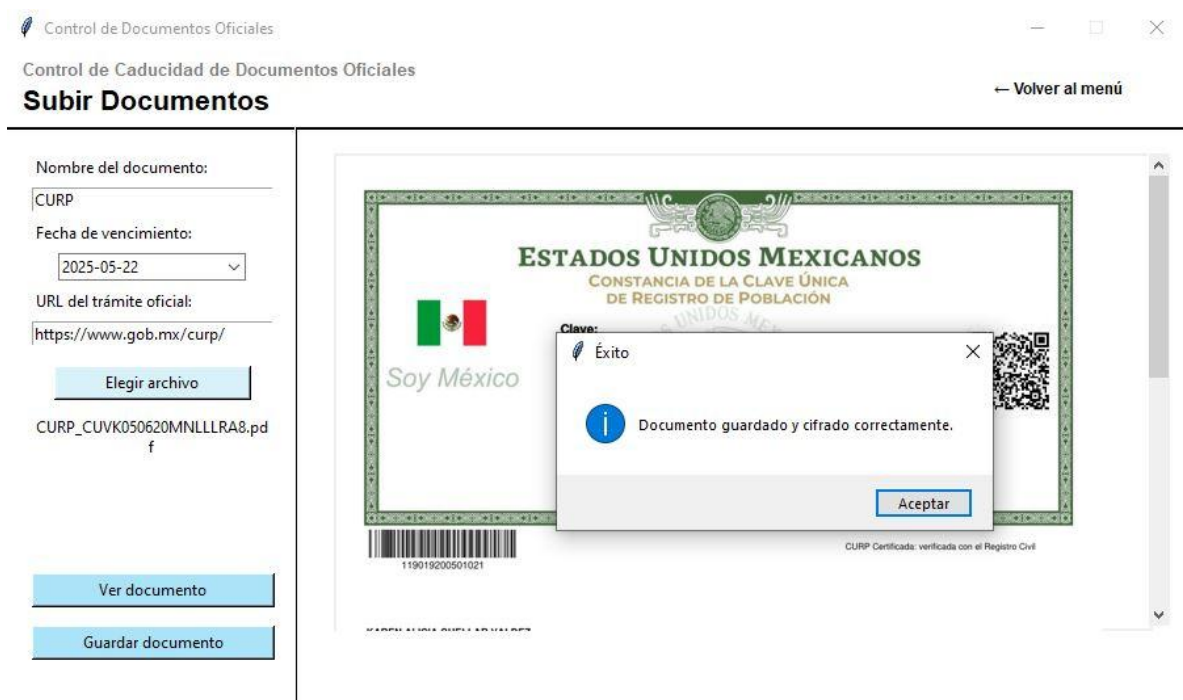
URL del trámite oficial:

[Elegir archivo](#)[Ver documento](#)[Guardar documento](#)

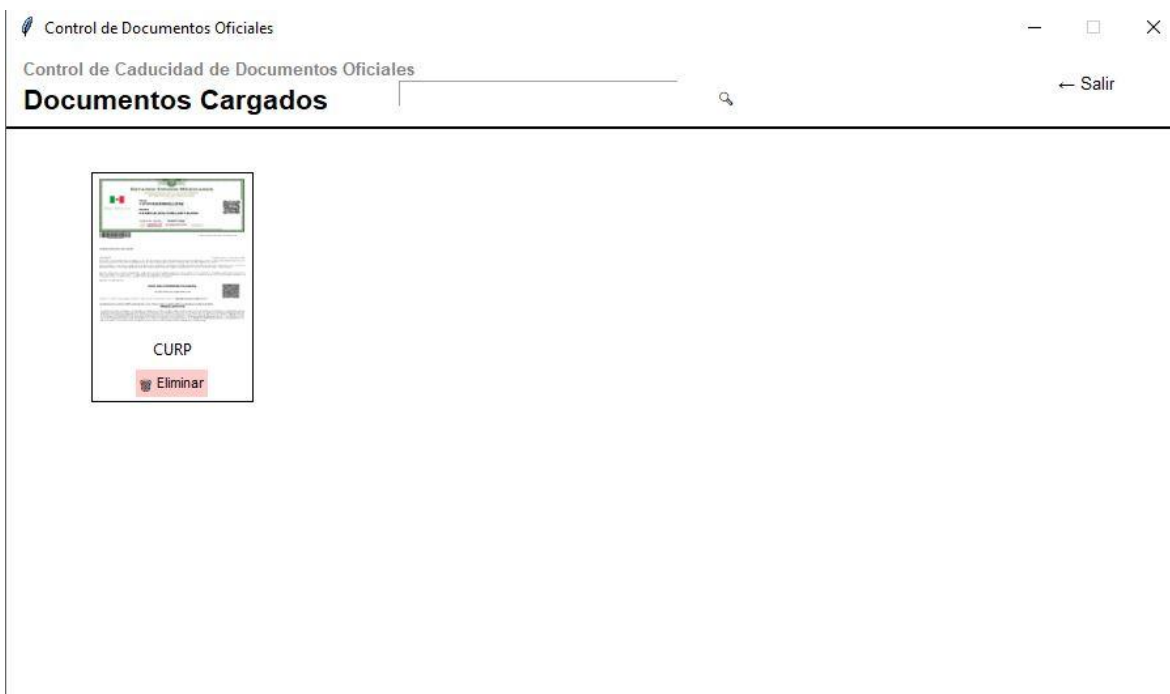
Al precionar el boton ver documento se nos abrira una ventana con nuestro documnto seleccionado



Al dar clic en guardar documento nos aparecera esta ventana y aquí sabremos que se guardo correctamente



En el apartado Documentos Cargados podremos ver los documentos que anteriormente hemos cargado



En el apartado de Asistente podremos auxiliarnos si algo sale mal con el sistema



