**Spring 2: componentes y librerías**

Inicio: 25 de febrero 2025

Fin: 28 de febrero 2025

Responsables de la actividad: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* **Infografía** 
  + Librerías y componentes
    - Clase, clase base, derivada
    - Orientado a C#
  + Debe contener:
    - Que es
    - Cuál es la diferencia entre un componente y librerías
    - Características
    - Ventajas
    - Aplicaciones de librerías y componentes
    - Fuentes bibliográficas
* 1ro: buscar información
  + Consultar o buscar información
* Estructurar información
  + Recolectar información, y estructurar de forma entendible
* Buscar una plantilla
* Diseñar la infografía
  + Utilizar canvas, o realizarla
* Validar la infografía
  + Antes de subir la infografía, validad los requerimientos de la rubrica
* Subir la infografía de manera individual
  + Documentación del proceso de la hechura del trabajo
    - Donde se sube el trabajo de cómo va iniciando la infografía, con capturas de pantalla hasta el resultado final

**Reporte de infografía**

1. **Consulta de información de diferentes fuentes**

**Componentes:**

Son unidades independientes de funcionalidad que pueden ser reutilizadas en diversas aplicaciones.

Los componentes permiten al usuario interactuar con la aplicación y proporcionar información desde el programa al usuario sobre el estado de la aplicación.

Representa todo lo que tiene una posición, un tamaño, puede ser pintado en pantalla y puede recibir eventos.

Ejemplo de componentes:

* + Botones
  + Barras de desplazamiento
  + Etiquetas
  + Listas
  + Cajas de selección

Los componentes agrupan datos y comportamientos relacionados (encapsulación)

Se pueden utilizar en diferentes contextos sin modificaciones sustanciales. (reusabilidad)

**Librerías:**

Conjunto de implementaciones de comportamiento, escritas para un lenguaje de programación, que tiene una interfaz bien definida para el comportamiento que se invoca.

Se enfoca en resolver un problema específico con funciones y objetos que los desarrolladores pueden reutilizar fácilmente.

Son herramientas para hacer el desarrollo de software más eficiente, proporcionan fragmentos de código reutilizables que los desarrolladores pueden utilizar para programar de forma rápida y sencilla.

Es la colección de código previamente que los programadores pueden utilizar para desarrollar software de maneras más ágil, estas colecciones de código reutilizable suelen resolver problemas o necesidades comunes de desarrollo.

Son colecciones de funciones y procedimientos que los desarrolladores pueden utilizar para evitar la reescritura de código común.

Proporcionan funcionalidades específicas que se pueden incluir en un programa mediante la importación.

El comportamiento que implementa una librería no espera ser utilizadas de forma autónoma, sino que su fin es ser utilizada por otros programas, independientes y de forma simultánea.

Una librería puede requerir de otras para funcionar, pues el comportamiento que definen refina, o altera, el comportamiento de la librería original, o bien la hace disponible para otra tecnología o lenguaje de programación.

¿Cómo se utilizan?

Cada librería está diseñada para proporcionar una solución a una necesidad o un problema de desarrollo específico.

* + Interfaces de usuario
  + Gestión de datos
  + Algoritmos
  + Animaciones

Las funciones en las librerías están disponibles para cualquier parte del código que las impone (accesibilidad)

Las librerías permiten a los desarrolladores elegir solo las funciones que necesitan para su aplicación.

Características

|  |  |
| --- | --- |
| Componentes | Librerías |
| Se integran en la arquitectura de una aplicación | Ofrecen funcionalidades que se invocan según sea necesario. |
| Suelen ser más pesados y complejos | Ofrecen funcionalidades que se invocan según sea necesario |
| Pueden tener dependencias entre si con formatos más complejos | Suelen ser independientes y pueden usarse por separado |
| La interacción entre componentes puede requerir APIs o eventos | Se utilizan mediante llamadas directas a funciones |

1. **Estructurar la información**

* **Conceptos**

**Componentes:** Son unidades independientes que pueden ser utilizadas en diferentes contextos sin modificaciones. Permiten al usuario interactuar con la aplicación y agrupan datos y/o comportamientos.

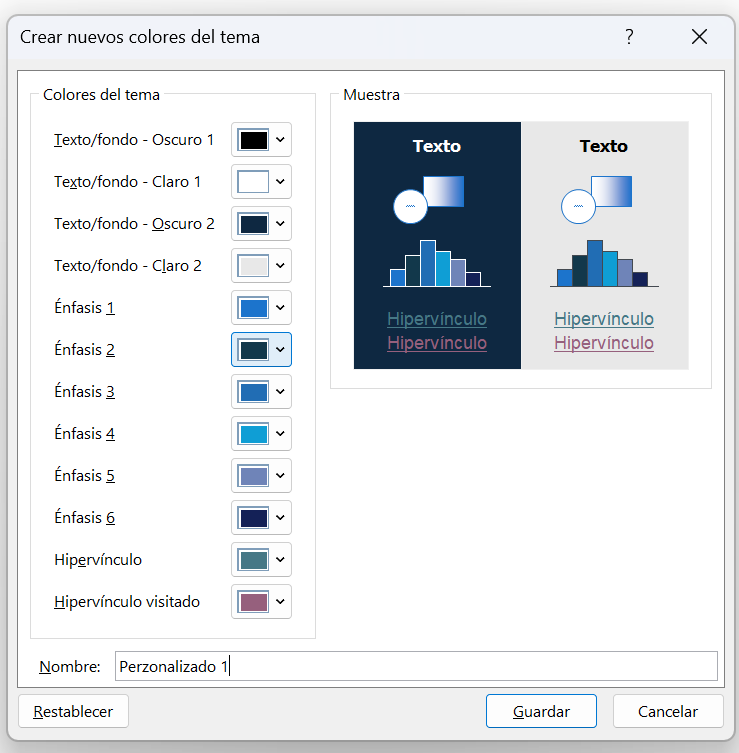
**Librerías:** Su fin es ser utilizadas por otros programas, independientes y de forma simultánea para evitar la reestructura de código común, ya que proporcionan fragmentos de código reutilizables que los desarrolladores pueden utilizar para programar de forma eficiente.

* **Características:**

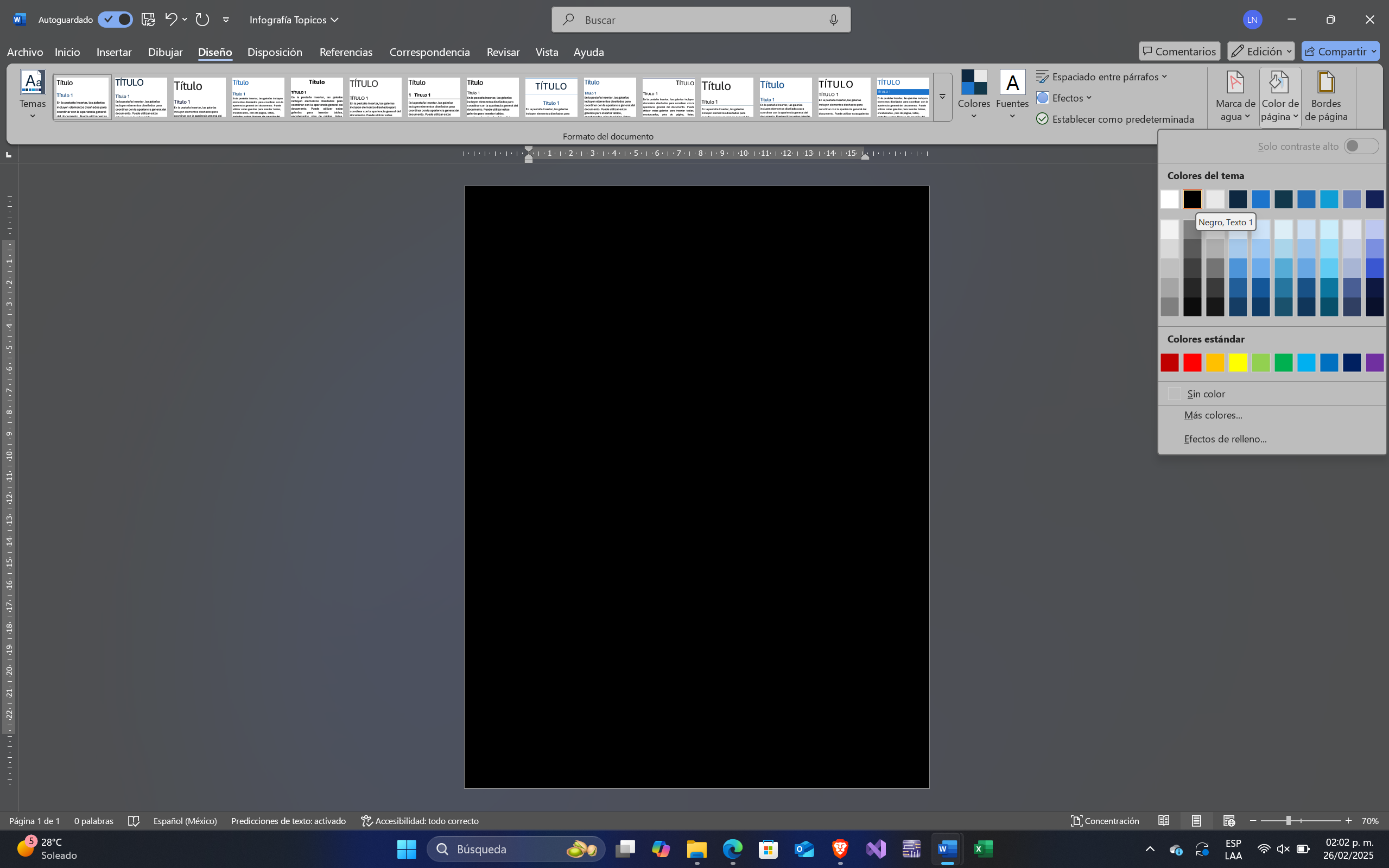
|  |  |
| --- | --- |
| Componentes | Librerías |
| Se integran en la arquitectura de una aplicación | Ofrecen funcionalidades que se invocan según sea necesario. |
| Suelen ser más pesados y complejos | Ofrecen funcionalidades que se invocan según sea necesario |
| Pueden tener dependencias entre si con formatos más complejos | Suelen ser independientes y pueden usarse por separado |
| La interacción entre componentes puede requerir APIs o eventos | Se utilizan mediante llamadas directas a funciones |
| Parte reutilizable de una aplicación con funcionalidad propia. | Conjunto de funciones o herramientas que facilitan el desarrollo. |
| Se integran dentro de una aplicación específica | Se importan para usar funcionalidades específicas. |
| Se puede incluir interfaz gráfica UI | No siempre tienen interfaz gráfica, pueden ser solo funciones |
| Dependen del framework o entorno en el que se usen. | Pueden ser usadas en múltiples proyectos y entornos. |

* **Ventajas:**
* Reducción de código
* Fácil organización de proyectos
* Mejora el rendimiento de ejecución de proyectos
* Mejor eficiencia a la hora de trabajar
* Desarrollo más rápido
* Facilita la actualización y mantenimiento de código
* Diseño de infografía en Word
* **Aplicaciones:**
* Desarrollo web y de interfaces de usuario (UI/UX)
* Desarrollo de aplicaciones móviles
* Desarrollo de videojuegos
* Desarrollo de software y automatización

1. **Diseñar la infografía**
2. Elegir una paleta de colores con la que vallamos a trabajar
3. Adaptar la paleta de colores al diseño en Word



1. Para el diseño de la infografía utilizaremos fondo negro, por lo que, en la sección de diseño y color de página, elegimos el color negro.



1. Comenzamos a diseñar la infografía insertando figuras, como primer punto insertamos un rectángulo, al cual, de fondo le agregaremos una imagen representativa del tema.



1. Con la ayuda de los cuadros de texto, diseñamos el título de nuestra infografía y modificamos el color, tamaño y tipografía para una mejor esteticidad.



1. Agregamos rectángulos para organizar la información.

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

1. **Resultado**
2. **Fuentes bibliográficas**

<https://es.scribd.com/document/503868756/COMPONENTES-Y-LIBRERIAS>

<https://topicoslesli.blogspot.com/2016/09/unidad-2-componentes-y-librerias.html>

<https://es.slideshare.net/JosAntonioSandovalAc/tpicos-avanzados-de-programacin-unidad-2-componentes-y-librerias>

<https://definicionwiki.com/definicion-de-componentes-en-programacion-ejemplos-que-es-autores/>

<https://reactiveprogramming.io/blog/es/react/que-son-los-componentes>

<https://www.unir.net/revista/ingenieria/librerias-programacion/>

**Spring 2: componentes y librerías**

Inicio: 25 de febrero 2025

Fin: 07 de marzo 2025

Trabajo en equipo en GitHub

Responsables de la actividad: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* **Estructura de librería (Reporte)**
  + Definir una librería en base a su función específica
    - Que es lo que va a hacer la librería
    - Librería para una calculadora
      * Investigar el tipo de librería a utilizar (25, 26de febrero)
  + Definir la estructura
    - Definir
      * Clases (27 febrero y 1 de marzo)
      * Métodos (1 y 2 de marzo)
      * Atributos (3 y 4 marzo)
  + Codificar los componentes (4 marzo)
    - Codificar la librería en consola
      * Solo utilizar entrada de datos por teclado
  + Probar su funcionamiento (5 marzo)
    - Hacer pruebas
  + Compilar
    - Traducir de lenguaje maquina a un lenguaje reutilizable
  + Documentar

**Spring 2: componentes y librerías**

Inicio: 25 de febrero 2025

Fin: 12 de marzo 2025

Trabajo en equipo en GitHub

Responsables de la actividad: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* **Estructurar de librería** 
  + Definir la función de la librería
  + Definir la librería
    - Clases
    - Métodos
    - Atributos
  + 14 de marzo 2025:
  + Codificar
    - UI-UX
    - Codificar en la consola
  + Probar
  + Compilar
    - Hacer pruebas
  + Pitch
    - Explicación de 2 o 3 minutos
    - Enfocado en explicar (nombre, proyecto, que implementación se utilizó)
  + Documentar

**TEMA 2: LIBRERIAS Y COMPONENTES EN VISUAL STUDIO C#**

**1: Portada**

Titulo: librerías y componentes en visual studio con c#

Subtitulo: aplicaciones y desarrollo en entornos empresariales

Imagen ilustrativa de visual studio y c#

**Definición: librerías y componentes:**

Librería conjunto estructurado de código para ser reutilizada y mejor entendible

Componente: bloques de códigos estructurados en clases que realizan una función específica

Una **librería** ofrece herramientas reutilizables para facilitar el desarrollo.

* + Estáticas (.lib)
  + Dinámicas (.dll) Librerías dinámicas (se pueden agregar más funciones)
* Características
  + Reutilización de código
  + Modularidad y mantenimiento
  + Separación de responsabilidades
* Ventajas
  + Reducción de código
  + Facilitan la organización de proyectos
  + Mejora el rendimiento de los proyectos
* Desventajas
  + Posibles problemas de compatibilidad
  + Dependencia de terceros
  + Necesidad de documentación clara para el uso
  + Mayor esfuerzo en depuración de prueba

Un **componente** es una parte estructural y funcional de una aplicación, especialmente en interfaces gráficas.

Elementos reutilizables encapsulados con funcionalidad específica

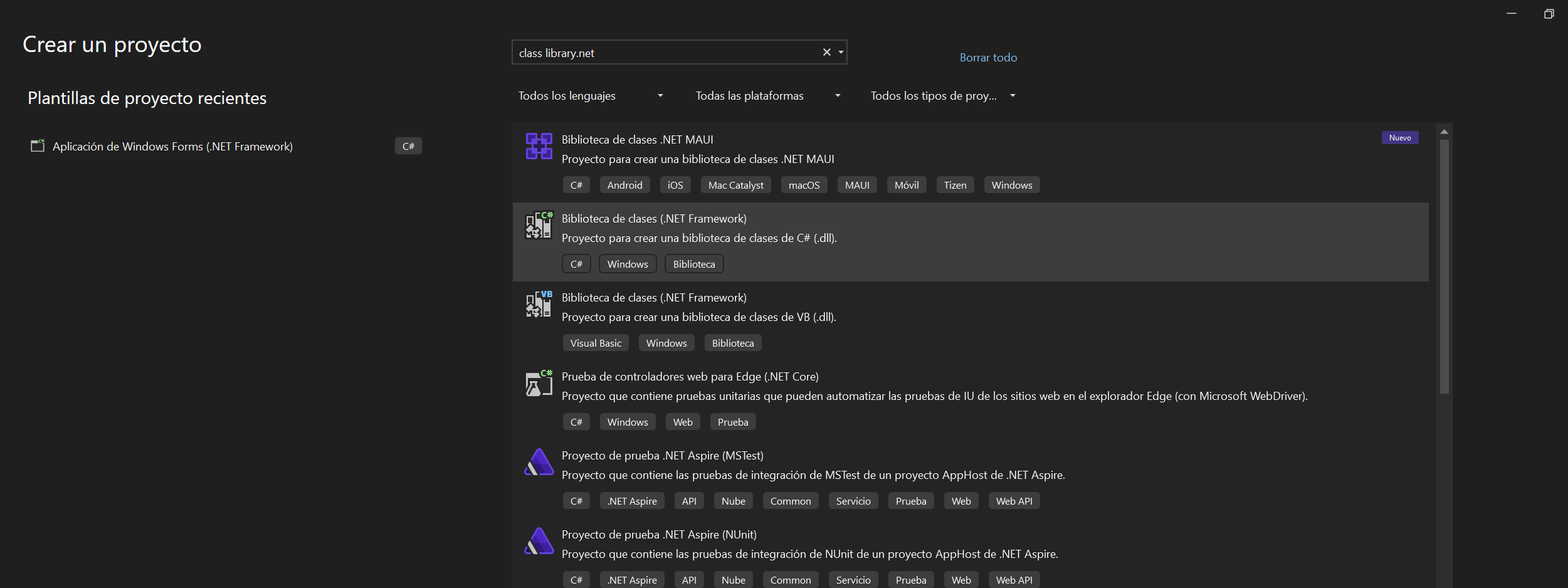
* + Gráficos UI (Windows form)
  + No gráficos (Servicios, Api)

**Diferencia** que un componente se puede instalar y desinstalar, y la librería no, puede colapsar.

Librería un bloque de código, necesita estar instalado

Componente se puede instalar y desinstalar sin afectar el programa

**Pasos para crear:**



Se creo la biblioteca

**Uso de una librería en un proyecto c#**

* + Agrega referencia en visual studio
  + Usar using para importar los espacios de nombres
  + Llamada a métodos de la librería en código

**Ejemplo de una librería:**

* + Librería para el manejo en un sistema de gestión documental
  + Clases principales (FileMAnager y Logger)
  + Implementación de métodos
* System.io() manejo de archivos
* Newtonsoft.Json (serialización JSON )
* Entity framework (ORM para bases de datos
* restSharp (consumo de APIS REST)
* dapper (alternativa)

**aplicaciones de librerías en sistemas operativos**

encriptar: ocultar los datos

* documentarlas
* probarlas
* majeo de excepciones y validaciones
* versionando y compatibilidad con otros proyectos

**04/03/2025**

¿Qué es una clase?

Conjunto de entidades que comparten características en común.

Característica de una clase

* métodos
* atributos

encapsulamiento

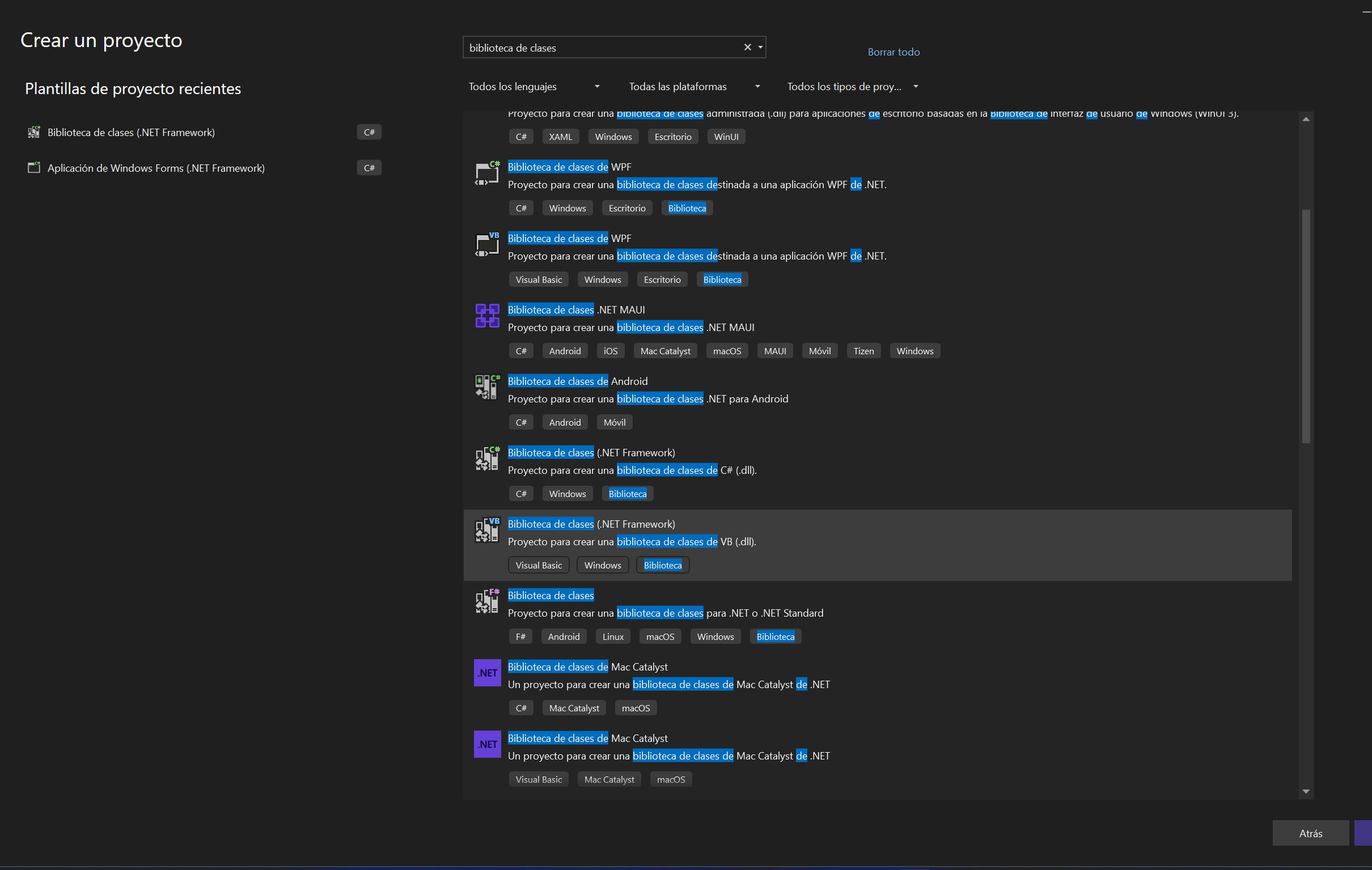
abstracción

Pase de parámetro por valor

Una copia de la variable original sin modificaciones

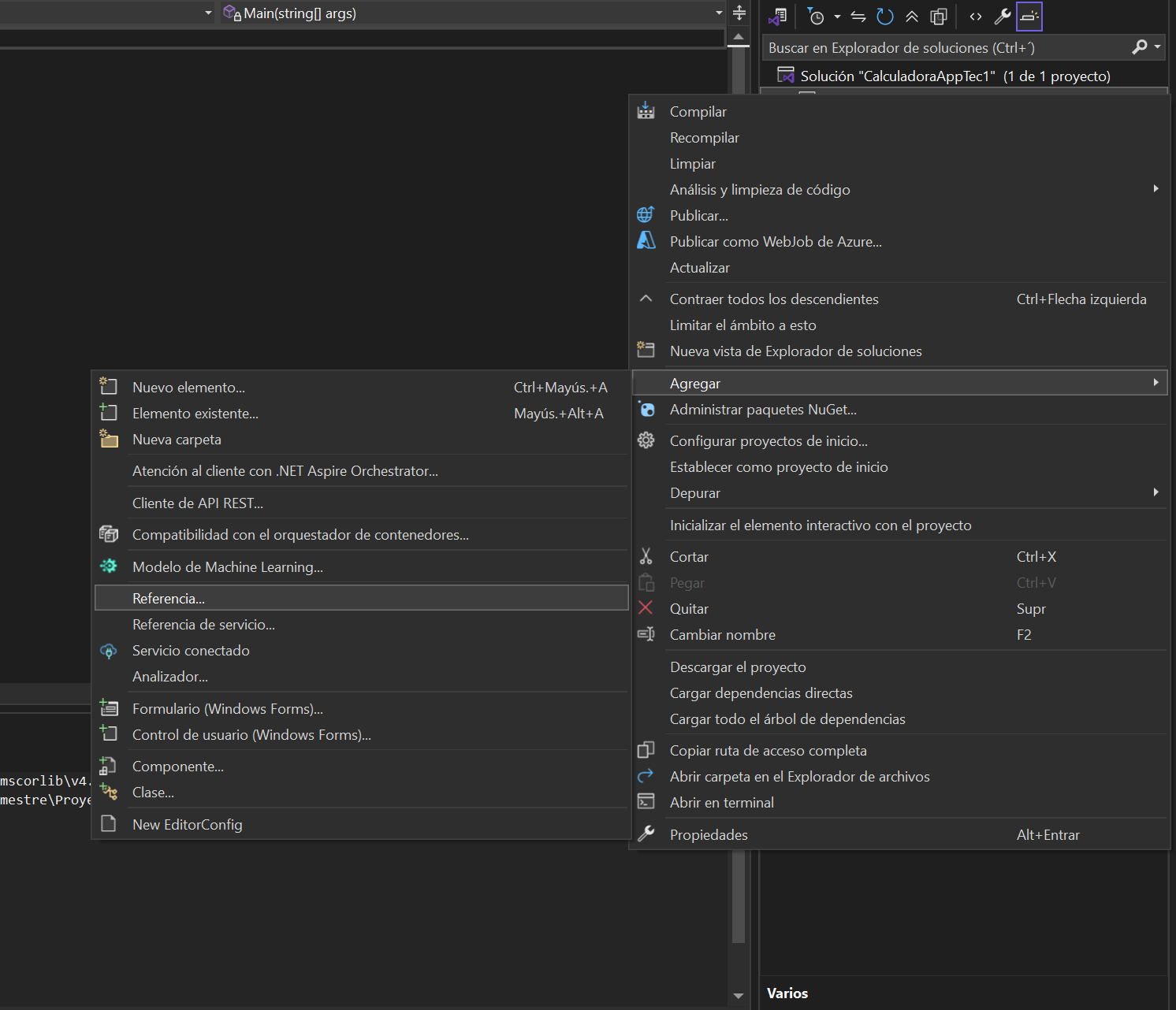
Pase de parámetro por referencia

Try catch



Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto. GenerarTexto

El contenido generado por IA puede ser incorrecto. la Biblioteca Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Git bash

Intsalat git

Comandos para

Git init

Inicializa un repositorio en blanco

Git add.

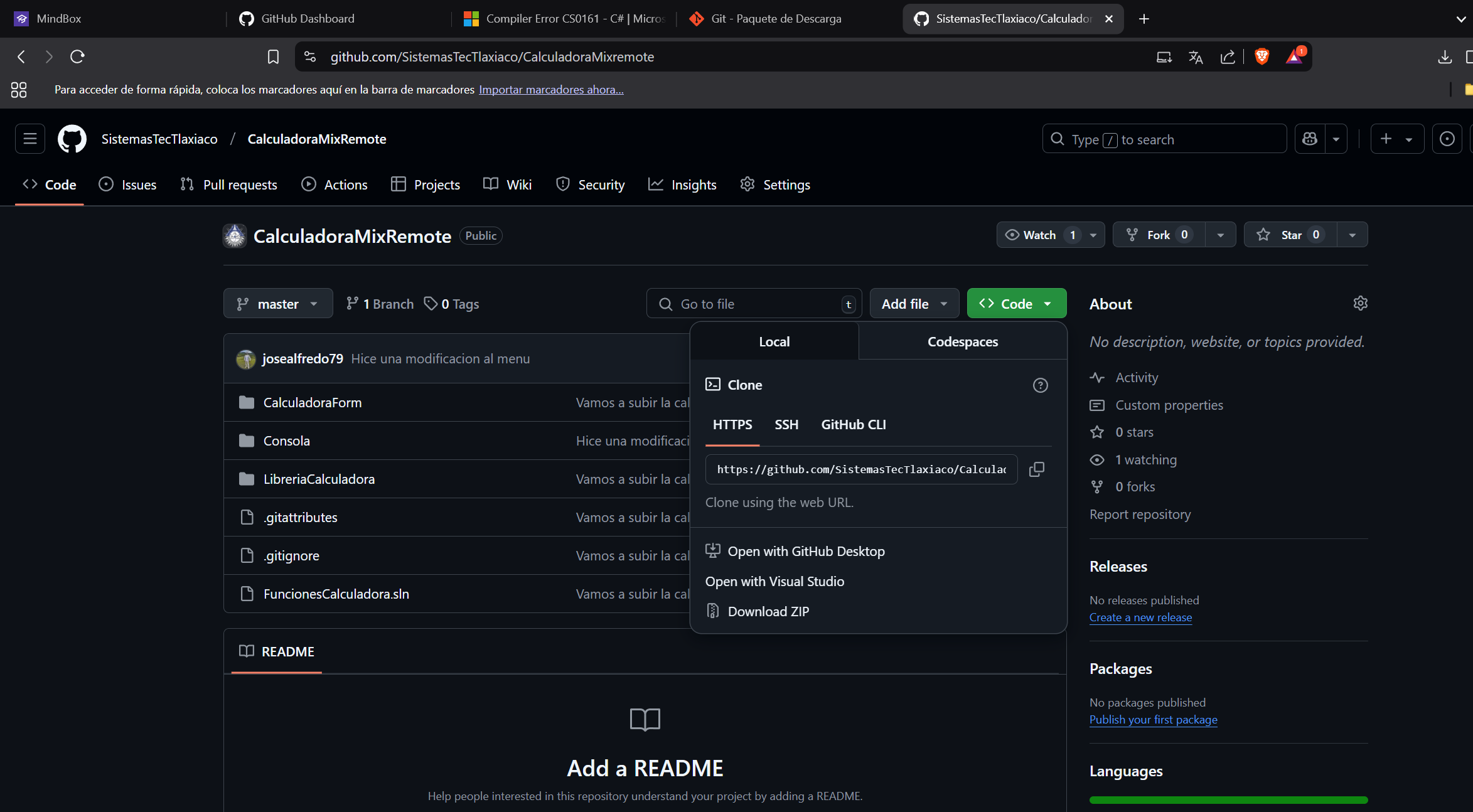
Inicializar el proyecto (guarda cambios hechos)

Git commit -m “para comentar se agregan comillas”

* investigar comandos

conformar los cambios

git push -u orign master (en ves de main debe ser master )

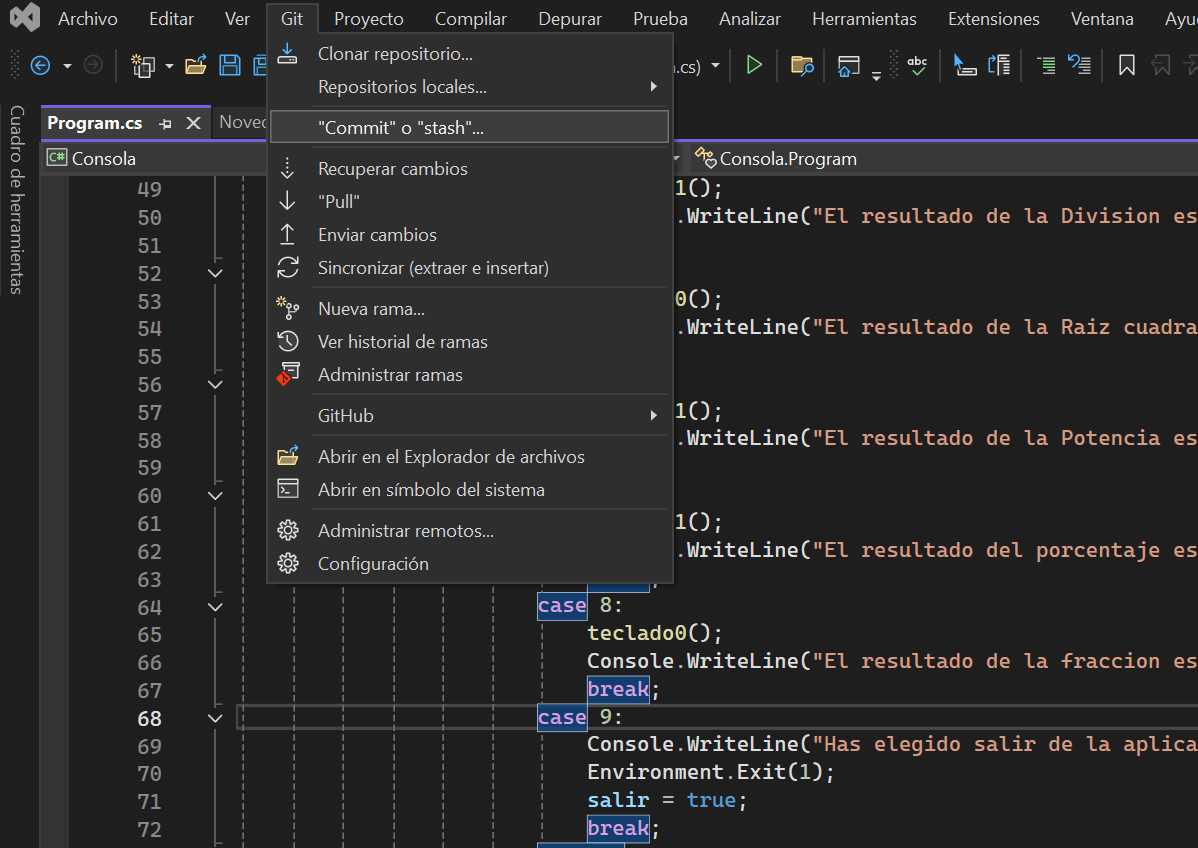


Abrir open git vbash

Copiat: <https://github.com/SistemasTecTlaxiaco/CalculadoraMixRemote.git>

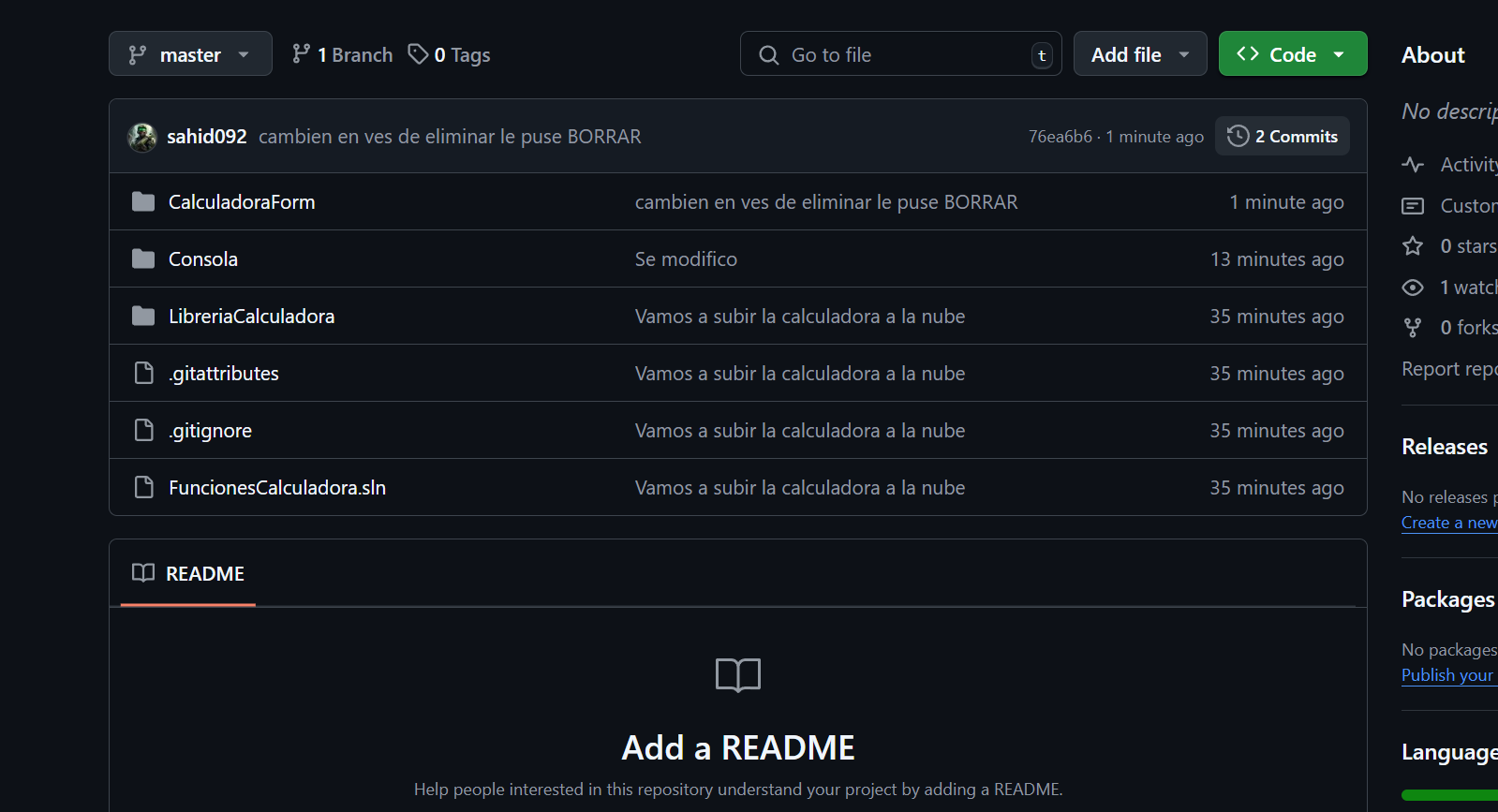
Git clone

Pegar

para guardar cambios

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.enviar cambios al equipo

commints para visualizar los cambios realizados por otrdo integrantes del equipo

Texto

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.