

UNIVERSIDAD IBEROAMERICANA

Mantenimiento de software

Actividad 4 - Utilizando sistemas de control de versiones

Laura Valentina Hidalgo Melo – Código Banner. 100139847
Valeria Stefany Rojas Sánchez – Código Banner. 100139839

26 de mayo del 2024

HISTORIAS DE USUARIO

1. Historia de usuario 1. Realizar Suma

- **Como:** Usuario de la calculadora
- **Quiero:** Poder realizar sumas de dos números
- **Para:** Obtener el resultado de la adición de esos números
- Criterios de Aceptación:
 - El usuario puede ingresar dos números.
 - El usuario puede seleccionar la operación de suma (+).
 - La calculadora muestra el resultado de la suma de los dos números.

2. Historia de usuario 2. Realizar Resta

- Como: Usuario de la calculadora
- **Quiero:** Poder realizar restas de dos números
- **Para:** Obtener el resultado de la adición de esos números
- Criterios de Aceptación:
 - El usuario puede ingresar dos números.
 - El usuario puede seleccionar la operación de resta (-).
 - La calculadora muestra el resultado de la resta de los dos números.

3. Historia de usuario 3. Realizar una multiplicación

- Como: Usuario de la calculadora
- Quiero: Poder realizar multiplicación de dos números
- Para: Obtener el resultado de la adición de esos números
- Criterios de Aceptación:
 - El usuario puede ingresar dos números.

- El usuario puede seleccionar la operación de multiplicación (*).
- La calculadora muestra el resultado de la multiplicación de los dos números.

4. Historia de usuario 4. Realizar una división

- Como: Usuario de la calculadora
- Quiero: Poder realizar división de dos números
- Para: Obtener el resultado de la división de esos números
- Criterios de Aceptación:
 - El usuario puede ingresar dos números.
 - El usuario puede seleccionar la operación de división (/).
 - La calculadora muestra el resultado de la división de los dos números.
 - Si el segundo número es cero, la calculadora muestra un mensaje de error indicando que la división por cero no es permitida.

5. Historia de usuario 5. Continuar con otra operación

- Como usuario de la calculadora,
- Quiero poder realizar múltiples operaciones consecutivamente,
- Para evitar reiniciar la calculadora después de cada operación.
- Criterios de Aceptación:
 - Después de mostrar el resultado de una operación, la calculadora pregunta al usuario si desea realizar otra operación.
 - El usuario puede elegir continuar o salir de la calculadora.

EXPLICACIÓN CÓDIGO

Bucle principal: while (true) crea un bucle infinito que permite al usuario realizar múltiples operaciones hasta que decida salir.

```
while (true)
{
```

Se limpia la consola y se muestra título

```
Console.Clear();
Console.WriteLine("Calculadora Básica en C#");
Console.WriteLine("-----");
```

Se leen las entradas del usuario

```
// Leer el primer número
Console.Write("Introduce el primer número: ");
double num1 = Convert.ToDouble(Console.ReadLine());

// Leer la operación
Console.Write("Introduce la operación (+, -, *, /): ");
char operacion = Console.ReadKey().KeyChar;
Console.WriteLine();

// Leer el segundo número
Console.Write("Introduce el segundo número: ");
double num2 = Convert.ToDouble(Console.ReadLine());
```

Se realizan cálculos y se validan operaciones, para esto se hace uso de un switch que evalúa el tipo de operación e implementa la lógica para suma, resta, multiplicación y división.

```

switch (operacion)
{
    case '+':
        resultado = num1 + num2;
        break;
    case '-':
        resultado = num1 - num2;
        break;
    case '*':
        resultado = num1 * num2;
        break;
    case '/':
        if (num2 != 0)
        {
            resultado = num1 / num2;
        }
        else
        {
            Console.WriteLine("Error: División por cero no permitida.");
            operacionValida = false;
        }
        break;
    default:
        Console.WriteLine("Operación no válida.");
        operacionValida = false;
        break;
}

```

Mostrar resultado

```

// Mostrar el resultado
if (operacionValida)
{
    Console.WriteLine($"Resultado: {num1} {operacion} {num2} = {resultado}");
}

```

Opción de continuar con otra operación o salir.

```

// Preguntar al usuario si desea continuar
Console.WriteLine("¿Deseas realizar otra operación? (s/n)");
char continuar = Console.ReadKey().KeyChar;
if (continuar != 's' && continuar != 'S')
{
    break;
}

```

COMANDOS GIT

1. Primero se debe clonar el Project haciendo uso del comando git clone

```
C:\Users\hokma\OneDrive\Documentos\Universidad\GitHub
λ git clone https://github.com/laurahidalgo78/Actividad-4---Utilizando-sistemas-de-control-de-versiones.git|
```

```
C:\Users\hokma\OneDrive\Documentos\Universidad\GitHub
λ git clone https://github.com/laurahidalgo78/Actividad-4---Utilizando-sistemas-de-control-de-versiones.git
Cloning into 'Actividad-4---Utilizando-sistemas-de-control-de-versiones'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 9 (delta 1), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (9/9), done.
Resolving deltas: 100% (1/1), done.
```

2. Para listar las ramas que están en el repositorio se realiza con el comando git branch

```
C:\Users\hokma\OneDrive\Documentos\Universidad\GitHub\Actividad-4---Utilizando-sistemas-de-control-de-versiones (main -> origin)
λ git branch
* main
```

Podemos ver que estamos en la rama master, es aquí donde se subiera el proyecto.

3. Para visualizar el estado actual del directorio de trabajo y el área de preparación y que cambios ha sido preparados se debe ver como git status

```
C:\Users\hokma\OneDrive\Documentos\Universidad\GitHub\Actividad-4---Utilizando-sistemas-de-control-de-versiones (main -> origin)
λ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    calculadora/

nothing added to commit but untracked files present (use "git add" to track)
```

4. Para agregar cambios en el directorio de trabajo al área de preparación se debe usar el comando git add *

```
C:\Users\hokma\OneDrive\Documentos\Universidad\GitHub\Actividad-4---Utilizando-sistemas-de-control-de-versiones (main -> origin)
λ git add *
```

5. Este comando debe registrar los cambios agregados en el área de preparación al repositorio local. Esto funciona como una toma instantánea de los archivos. Con un git commit de la siguiente manera.

```
C:\Users\hokma\OneDrive\Documentos\Universidad\GitHub\Actividad-4---Utilizando-sistemas-de-control-de-versiones (main -> origin)
λ git commit -m "Se agrega proyecto calculadora"
[main 3e65f9c] Se agrega proyecto calculadora
37 files changed, 433 insertions(+)
create mode 100644 Calculadora/.vs/Calculadora/DesignTimeBuild/.dtbcache.v2
create mode 100644 Calculadora/.vs/Calculadora/FileContentIndex/7ee163bc-fbe9-4d69-a001-930dc5bd1103.vsidix
create mode 100644 Calculadora/.vs/Calculadora/FileContentIndex/read.lock
create mode 100644 Calculadora/.vs/Calculadora/v17/.fudtcache.v2
create mode 100644 Calculadora/.vs/Calculadora/v17/.suo
create mode 100644 Calculadora/.vs/ProjectEvaluation/calculadora.metadatas.v5.2
create mode 100644 Calculadora/.vs/ProjectEvaluation/calculadora.projects.v5.2
create mode 100644 Calculadora/Calculadora.sln
create mode 100644 Calculadora/Calculadora/Calculadora.csproj
create mode 100644 Calculadora/Calculadora/Program.cs
create mode 100644 Calculadora/Calculadora/bin/Debug/net6.0/Calculadora.deps.json
create mode 100644 Calculadora/Calculadora/bin/Debug/net6.0/Calculadora.dll
create mode 100644 Calculadora/Calculadora/bin/Debug/net6.0/Calculadora.exe
create mode 100644 Calculadora/Calculadora/bin/Debug/net6.0/Calculadora.pdb
create mode 100644 Calculadora/Calculadora/bin/Debug/net6.0/Calculadora.runtimeconfig.json
create mode 100644 Calculadora/Calculadora/obj/Calculadora.csproj.nuget.dgspec.json
create mode 100644 Calculadora/Calculadora/obj/Calculadora.csproj.nuget.g.props
create mode 100644 Calculadora/Calculadora/obj/Calculadora.csproj.nuget.g.targets
create mode 100644 Calculadora/Calculadora/obj/Debug/net6.0/.NETCoreApp,Version=v6.0.AssemblyAttributes.cs
create mode 100644 Calculadora/Calculadora/obj/Debug/net6.0/Calculadora.AssemblyInfo.cs
create mode 100644 Calculadora/Calculadora/obj/Debug/net6.0/Calculadora.AssemblyInfoInputs.cache
create mode 100644 Calculadora/Calculadora/obj/Debug/net6.0/Calculadora.GeneratedMSBuildEditorConfig.editorconfig
create mode 100644 Calculadora/Calculadora/obj/Debug/net6.0/Calculadora.GlobalUsings.g.cs
create mode 100644 Calculadora/Calculadora/obj/Debug/net6.0/Calculadora.assets.cache
create mode 100644 Calculadora/Calculadora/obj/Debug/net6.0/Calculadora.csproj.AssemblyReference.cache
create mode 100644 Calculadora/Calculadora/obj/Debug/net6.0/Calculadora.csproj.BuildWithSkipAnalyzers.cache
create mode 100644 Calculadora/Calculadora/obj/Debug/net6.0/Calculadora.csproj.CoreCompileInputs.cache
create mode 100644 Calculadora/Calculadora/obj/Debug/net6.0/Calculadora.csproj.FileListAbsolute.txt
create mode 100644 Calculadora/Calculadora/obj/Debug/net6.0/Calculadora.dll
create mode 100644 Calculadora/Calculadora/obj/Debug/net6.0/Calculadora.genruntimeconfig.cache
create mode 100644 Calculadora/Calculadora/obj/Debug/net6.0/Calculadora.pdb
create mode 100644 Calculadora/Calculadora/obj/Debug/net6.0/_IsIncrementalBuild
create mode 100644 Calculadora/Calculadora/obj/Debug/net6.0/apphost.exe
create mode 100644 Calculadora/Calculadora/obj/Debug/net6.0/ref/Calculadora.dll
create mode 100644 Calculadora/Calculadora/obj/Debug/net6.0/refint/Calculadora.dll
create mode 100644 Calculadora/Calculadora/obj/project.assets.json
create mode 100644 Calculadora/Calculadora/obj/project.nuget.cache
```

6. Para subir los commit se realizar se usa git push

```
C:\Users\hokma\OneDrive\Documentos\Universidad\GitHub\Actividad-4---Utilizando-sistemas-de-control-de-versiones (main -> origin)
λ git push
Enumerating objects: 50, done.
Counting objects: 100% (50/50), done.
Delta compression using up to 12 threads
Compressing objects: 100% (39/39), done.
Writing objects: 100% (49/49), 157.70 KiB | 6.57 MiB/s, done.
Total 49 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), done.
To https://github.com/laurahidalgo78/Actividad-4---Utilizando-sistemas-de-control-de-versiones.git
10c9206..3e65f9c main -> main
```

7. Se puede visualizar el historial de commits haciendo uso del comando git log

```

C:\Users\hokma\OneDrive\Documentos\Universidad\GitHub\Actividad-4---Utilizando-sistemas-de-control-de-versiones (main -> origin)
λ git log
commit 3e65f9c62e47c21209fb3569b3338194b8c6e401 (HEAD -> main, origin/main, origin/HEAD)
Author: LAURA VALENTINA HIDALGO MELO <laura.hidalgo@hokma.ai>
Date: Sat May 25 16:02:49 2024 -0500

    Se agrega proyecto calculadora

commit 10c9206a658a4608b3d8cc59c1bc6ef766663d29
Author: laurahidalgo78 <69684607+lahurahidalgo78@users.noreply.github.com>
Date: Sat May 25 15:48:13 2024 -0500

    Update README.md

commit fa20da87781d8bae7e0f2ef42b125e5cb594d797
Author: laurahidalgo78 <69684607+lahurahidalgo78@users.noreply.github.com>
Date: Sat May 25 15:35:32 2024 -0500

    Update README.md

commit b38fc39eb5dde0d4e294c4af2740616cb234c9c5
Author: laurahidalgo78 <69684607+lahurahidalgo78@users.noreply.github.com>
Date: Sat May 25 15:34:44 2024 -0500

    Initial commit

```

Aportes:

Valeria Stefany Rojas Sánchez: Crea historias de usuarios

Grupo: Se crea sistema de calculadora

Laura Valentina Hidalgo Melo: Sube cambios GitHub

[LINK PARA INGRESAR A GITHUB](#)

REFERENCIAS

Atlassian. (s/f). *Comandos básicos de Git*. Atlassian. Recuperado el 25 de mayo de 2024, de <https://www.atlassian.com/es/git/glossary>