

UNIVERSIDAD JOSÉ CECILIO DEL VALLE



**UNIVERSIDAD
JOSE CECILIO DEL VALLE**

TEMA:

Investigación N° 1 acerca de GitHub

ASIGNATURA:

IIT2035 – ADM2012 Programación II

PRESENTADO POR:

Alejandro Josué Díaz Zepeda, #2021220030

CATEDRÁTICO:

Kevin Eduardo Fúnez Fúnez

FECHA:

02 de junio del 2022

TABLA DE CONTENIDO

INTRODUCCIÓN:.....	4
OBJETIVOS	5
Objetivo General:.....	5
Objetivos Específicos:	5
GITHUB	6
¿Qué es GitHub?	6
¿Cómo Empezar con GitHub?	6
Pasos para Crear una Cuenta en GitHub	6
GIT 7	
GITHUB Y VISUAL STUDIO	7
Inicio de Sesión en GitHub desde Visual Studio	7
Incorporación de una cuenta a Visual Studio	7
Autenticación de GitHub en Visual Studio.....	8
Inicio de sesión en GitHub desde Visual Studio.....	8
Creación de un Repositorio en GitHub desde Visual Studio.....	10
Publicación.....	10
Creación De Un Repositorio De Git	11
Incorporación De Los Archivos .gitignore Y .gitattributes: La Primera Confirmación	11
El Archivo .gitignore	11
El Archivo .gitattributes.....	11
Incorporación De Archivos Del Proyecto: La Segunda Confirmación	12

INVESTIGACIÓN N° 1 ACERCA DE GITHUB

Inserción en GitHub	12
Crear un Proyecto para Subirlo a GitHub	13
Inserción del proyecto en GitHub	13
Comprobación de la creación del repositorio	15
Confirmación de Cambios	18
Pasos: 18	
Comprobación de la confirmación.....	20
Envío de cambios a GitHub desde Visual Studio	20
Sincronización de Git.....	20
Envío de Cambios Forzado	21
Pasos para Envío de Cambios:.....	22
COMANDOS.....	25
Identificar quien está trabajando	25
Para Crear Repositorio con Git Local:	25
Para Crear Repositorio con Git Online:	25
Otros Comandos:	25
RAMAS	26
Manejo de las Ramas	26
Unión de Ramas.....	27
Eliminación de Ramas	27
CONCLUSIONES:.....	28
ANEXOS	29

INVESTIGACIÓN N° 1 ACERCA DE GITHUB

BIBLIOGRAFÍA	31
--------------------	----

INTRODUCCIÓN:

En el siguiente documento se estará presentando toda la información necesaria para comprender qué es Github, para qué se utiliza y como poder manejar esta herramienta que podemos enlazar con Visual Studio.

Además, aclarar que para términos explicativos, usaremos como base Visual Studio Community 2019.

También se hablará sobre las ramas y cómo hacer uso de estas dentro de Visual Studio con el lenguaje de programación de C#.

OBJETIVOS

Objetivo General:

Comprender qué es la herramienta Github para poder hacer uso de ella mediante una explicación detallada de cómo usarla en Visual Studio C#, determinando sus comandos y conceptos relacionados a esta aplicación de modo que se facilite su uso.

Objetivos Específicos:

- Definir qué es Github.
- Crear una cuenta en Github.
- Enumerar los comandos utilizados para el subir un repositorio y clonar un repositorio de Github.

GITHUB

¿Qué es GitHub?

GitHub es una herramienta esencial para los ingenieros de software, y su popularidad es inigualable. Actualmente cuenta con más de 25 millones de usuarios. Se trata de un número considerable de profesionales que recurren a GitHub para mejorar el flujo de trabajo y la colaboración.

En resumen, GitHub es un servicio basado en la nube que aloja un sistema de control de versiones (VCS) llamado Git. Éste permite a los desarrolladores colaborar y realizar cambios en proyectos compartidos, a la vez que mantienen un seguimiento detallado de su progreso.

¿Cómo Empezar con GitHub?

Puedes probar GitHub con tu equipo de forma gratuita. Hay un plan básico que incluye repositorios y colaboradores ilimitados, pero sólo 500 MB de espacio de almacenamiento.

Si quieres disfrutar de todas las funciones de GitHub, puedes elegir uno de sus planes de pago. (IMG 1.1)

Tras elegir un plan, introducir los datos requeridos y completar el proceso de registro, podrás empezar a explorar todo lo que GitHub tiene para ofrecer. GitHub no requiere ningún tipo de codificación o uso de la línea de comandos.

Pasos para Crear una Cuenta en GitHub

1. Visitar el sitio oficial de GitHub: <https://github.com/>
2. Presionar el botón de Inscribirse.
3. Le solicitará ingresar un correo electrónico.
4. Luego de haber ingresado su correo y una contraseña, presionar continuar.
5. Ahora deberá ingresar un nombre de usuario sin espacios.
6. A continuación, deberá escribir “y” si desea recibir anuncios en su correo.
7. Lo siguiente será realizar la verificación que se le solicite.
8. Por último, presione crear cuenta.

INVESTIGACIÓN N° 1 ACERCA DE GITHUB

GIT

Git es un sistema de control de versiones moderno más utilizado del mundo. Git es un proyecto de código abierto maduro y con un mantenimiento activo que desarrolló originalmente Linus Torvalds, el famoso creador del kernel del sistema operativo Linux, en 2005. Un asombroso número de proyectos de software dependen de Git para el control de versiones, incluidos proyectos comerciales y de código abierto. Los desarrolladores que han trabajado con Git cuentan con una buena representación en la base de talentos disponibles para el desarrollo de software, y este sistema funciona a la perfección en una amplia variedad de sistemas operativos e IDE (entornos de desarrollo integrados).

Para poder instalar git nos dirigiremos a <http://git-scm.com/downloads> dónde deberemos seleccionar nuestro sistema operativo.

GITHUB Y VISUAL STUDIO

Inicio de Sesión en GitHub desde Visual Studio

Hay diferentes maneras de usar GitHub con un proyecto. A veces, se clona un proyecto para obtener una copia en la que poder trabajar posteriormente. Otras veces es al contrario: se tiene un proyecto local que se quiere colocar en GitHub.

Antes de hacer nada con GitHub, hay que asegurarse de estar autenticado. GitHub debe saber quién es para poder proporcionarle acceso a los repositorios correctos.

Normalmente, la autenticación puede ser un proceso tedioso de varios pasos que implica la generación y el uso de claves SSH. Una de las ventajas de usar Visual Studio para trabajar con GitHub es que el IDE puede controlar automáticamente toda la configuración de autenticación. (IMG 2.1)

Incorporación de una cuenta a Visual Studio

En Visual Studio, normalmente se tiene una única cuenta de personalización que se usa para iniciar sesión en el IDE. Pero puede agregar varias cuentas a Visual Studio para poder acceder a los recursos desde cualquier cuenta sin tener que iniciar sesión por separado. Este conjunto de cuentas se denomina cadena de claves de Visual Studio. La cadena de claves le acompaña si inicia sesión en Visual Studio en otra máquina.

INVESTIGACIÓN N° 1 ACERCA DE GITHUB

Solo puede agregar una cuenta de GitHub a Visual Studio. Si tiene que acceder a recursos desde otra cuenta de GitHub, primero debe quitar cualquier cuenta de GitHub existente de Visual Studio y cerrar la sesión en la web (GitHub.com). Después, puede reiniciar el proceso para agregar la otra cuenta de GitHub a Visual Studio.

Autenticación de GitHub en Visual Studio

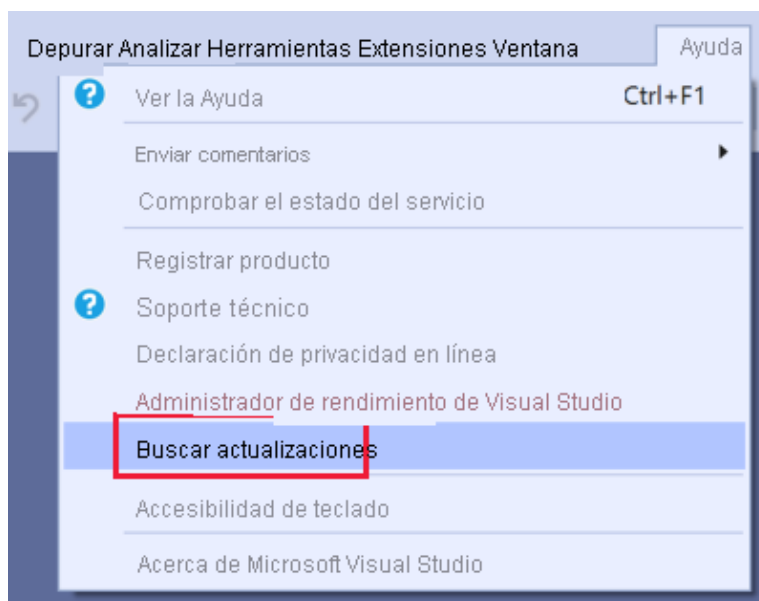
Al iniciar sesión con la cuenta de GitHub, Visual Studio almacena esas credenciales de forma segura y puede ayudar a acceder a las características de GitHub. Visual Studio permite examinar los repositorios privados y públicos de GitHub para clonar copias locales o crear nuevos repositorios en GitHub. Puede usar la funcionalidad de Git integrada para extraer de los repositorios remotos de GitHub e insertar en ellos. (IMG 2.2)

Inicio de sesión en GitHub desde Visual Studio

El inicio de sesión en una cuenta de GitHub por medio de Visual Studio le permitirá acceder a los recursos de GitHub desde Visual Studio. Todavía puede usar las herramientas básicas de Git en Visual Studio sin iniciar sesión en GitHub. Pero para crear repositorios y enviarlos a GitHub, tendrá que iniciar sesión.

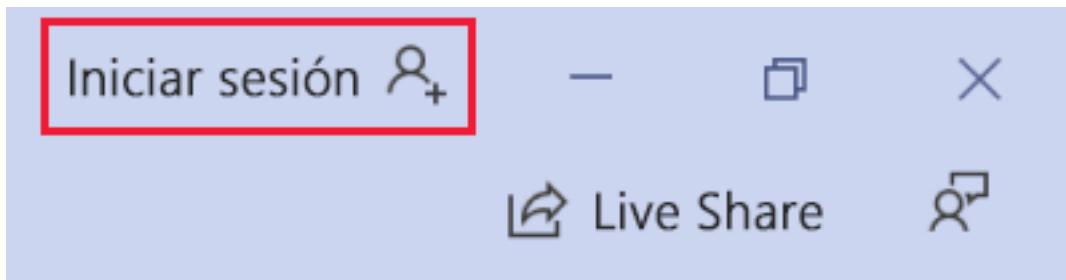
Aquí se agregará una cuenta de GitHub a la cadena de claves de Visual Studio y se autenticará para usar recursos de GitHub.

1. Asegúrese de que tiene instalada la versión más reciente de Visual Studio. Abra Visual Studio, vaya al menú **Ayuda** y seleccione **Buscar actualizaciones**.

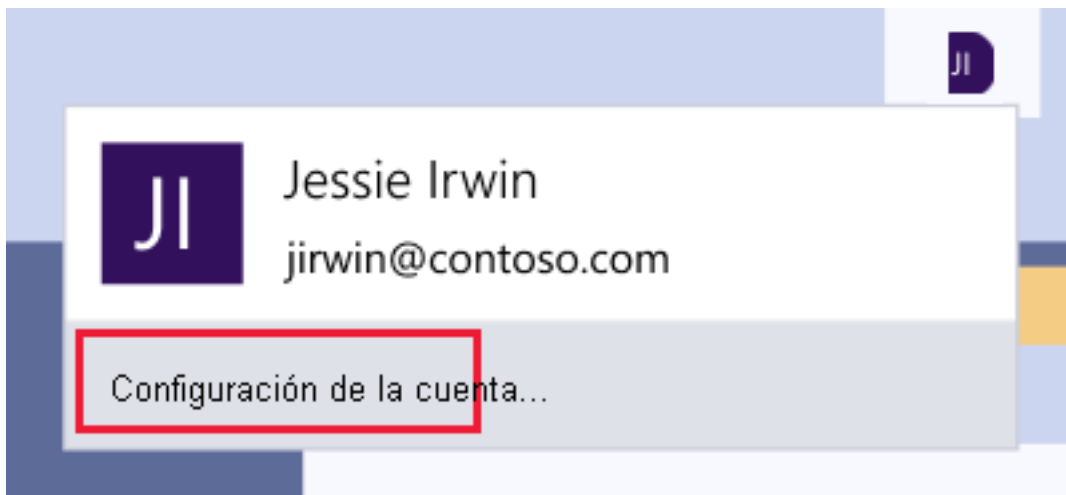


INVESTIGACIÓN N° 1 ACERCA DE GITHUB

2. Si todavía no ha iniciado sesión en Visual Studio, seleccione el botón **Iniciar sesión** de la esquina superior derecha de la ventana de Visual Studio. Continúe para iniciar sesión con la cuenta profesional o personal.



3. Una vez que se ha iniciado sesión en Visual Studio, se ve un botón de avatar en la esquina superior derecha. Selecciónelo y luego seleccione **Configuración de la cuenta** en la lista desplegable.



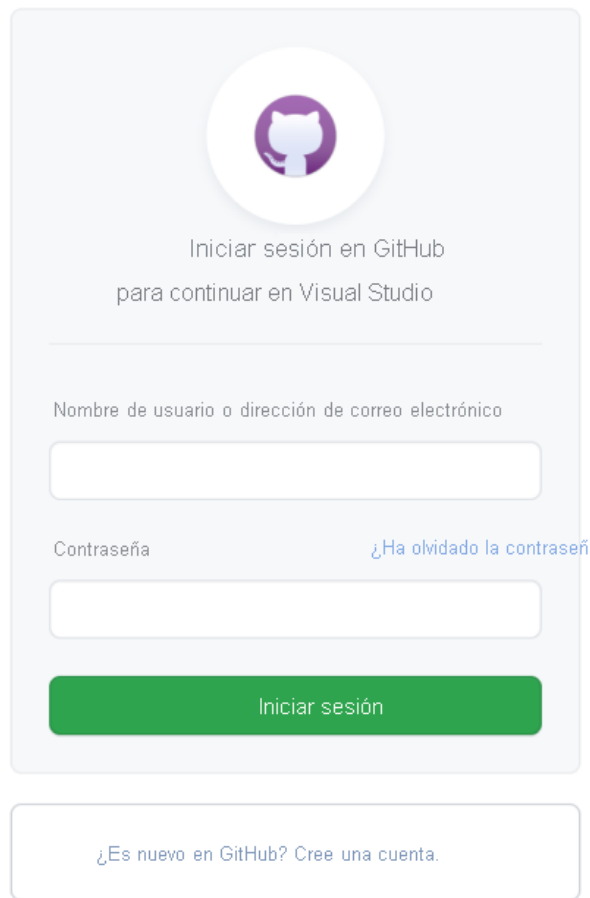
4. En la ventana **Configuración de la cuenta**, seleccione el botón **Agregar** situado junto a **Todas las cuentas**. Luego seleccione **GitHub**.



5. Se le redirigirá a la página de inicio de sesión de GitHub en el explorador predeterminado. Escriba las credenciales de GitHub y seleccione **Iniciar sesión**. Es posible que tenga que

INVESTIGACIÓN N° 1 ACERCA DE GITHUB

especificar un código de autenticación para la autenticación en dos fases.

The image shows a GitHub login interface. At the top, there is a GitHub logo (Octocat) inside a white circle. Below the logo, the text reads "Iniciar sesión en GitHub para continuar en Visual Studio". There is a horizontal line separating the header from the input fields. Below the line, there are two input fields: the first is labeled "Nombre de usuario o dirección de correo electrónico" and the second is labeled "Contraseña". To the right of the password field, there is a link that says "¿Ha olvidado la contraseña?". Below the input fields is a green button labeled "Iniciar sesión". At the bottom of the form, there is a link that says "¿Es nuevo en GitHub? Cree una cuenta."

Creación de un Repositorio en GitHub desde Visual Studio

Una vez autenticado en GitHub, Visual Studio puede mejorar el flujo de trabajo de GitHub. Una de estas mejoras consiste en la posibilidad de enviar (también denominado publicar) un proyecto local directamente a GitHub con un solo clic.

Publicación

Si va a comenzar con un proyecto local en Visual Studio, el proceso para colocarlo en GitHub tiene muchos pasos:

1. Crear un repositorio en GitHub.com.
2. Copiar el punto de conexión remoto.
3. Inicializar el repositorio localmente.
4. Agregar el punto de conexión remoto.
5. Agregar un archivo .gitignore.
6. Agregar un archivo .gitattributes.
7. Agregar los archivos del proyecto al "stage".

INVESTIGACIÓN N° 1 ACERCA DE GITHUB

8. Confirmar los archivos del proyecto.
9. Insertar en el repositorio remoto.
10. Comprobar que la operación ha sido correcta.

Visual Studio tiene una funcionalidad integrada que automatiza este proceso por completo.

Creación De Un Repositorio De Git

Cuando Visual Studio inicializa el repositorio, primero ejecuta un comando `git init`. La operación crea un subdirectorio `.git` en la raíz del proyecto, que contiene todos los metadatos de Git necesarios para el nuevo repositorio. Aparte de la carpeta `.git`, Visual Studio no modifica nada en el proyecto al inicializar.

Incorporación De Los Archivos .gitignore Y .gitattributes: La Primera Confirmación

Una vez que Visual Studio inicializa el repositorio, crea la primera confirmación para agregar los archivos `.gitignore` y `.gitattributes` al repositorio.

El Archivo .gitignore

No todos los archivos que se crean o actualizan en el código se deben confirmar en Git. Los archivos temporales del entorno de desarrollo, las salidas de prueba y los registros son ejemplos de archivos que se crean pero que no forman parte del código base.

Visual Studio crea automáticamente un archivo `.gitignore` básico con contenido personalizado adaptado al entorno de desarrollo de Visual Studio. Puede ver el archivo `.gitignore` de Visual Studio en la [colección de plantillas de archivo .gitignore de GitHub](#). Puede usar diferentes tipos de plantillas de archivo `.gitignore`, en función del tipo de proyecto que vaya a crear.

El Archivo .gitattributes

El archivo `.gitattributes` ayuda a Git a entender los tipos de archivo del repositorio. Es un archivo opcional, aunque importante, que Visual Studio crea de forma automática con su propia configuración personalizada.

La línea más importante del archivo `.gitattributes` es `* text=auto`, que establece el comportamiento predeterminado para normalizar automáticamente los finales de línea. Con esta línea, el

INVESTIGACIÓN N° 1 ACERCA DE GITHUB

repositorio es explícito con respecto al control de texto, para que todos los usuarios que utilizan el repositorio sigan el mismo conjunto de reglas. El archivo `.gitattributes` también establece formatos de imagen comunes como controladores de combinación comunes y binarios para las extensiones de Visual Studio.

Incorporación De Archivos Del Proyecto: La Segunda Confirmación

Con la segunda confirmación, Visual Studio agrega todos los archivos del proyecto al repositorio. La confirmación incluye todos los archivos no ocultos en la raíz del directorio del proyecto y sus subdirectorios.

Inserción en GitHub

Por medio del cuadro de diálogo **Crear un repositorio** de Visual Studio, puede crear un repositorio e insertarlo en GitHub al mismo tiempo. En la pestaña **GitHub** predeterminada, puede especificar a qué cuenta y propietario se atribuye el nuevo repositorio de GitHub. Como ya ha iniciado sesión en GitHub, estos campos se deberían haber rellenado de manera predeterminada.

El repositorio en GitHub tiene dos campos de metadatos principales: el nombre y la descripción. El nombre es un campo obligatorio y es el nombre del proyecto local de manera predeterminada. Puede cambiar el nombre por cualquier palabra alfanumérica. Si agrega espacios al nombre, Visual Studio los reemplaza por un guion (-). La descripción del repositorio es opcional, aunque resulta útil para explicar a otros usuarios cuál es el propósito del proyecto.

Todos los repositorios de GitHub creados desde Visual Studio se establecen como privados de forma predeterminada. Esto significa que solo puede ver el repositorio y confirmarlo. Para cambiar esta configuración, agregue colaboradores a los repositorios privados y administre el acceso por medio de la página **Configuración** del repositorio en GitHub.com. Al desactivar la casilla, el repositorio se cambia a público, lo que permite a cualquier usuario de Internet ver el repositorio y su contenido en GitHub.com. Pero todavía puede elegir quién puede contribuir directamente.

Además de insertar en GitHub, puede hacerlo en cualquier punto de conexión remoto (como Azure DevOps) mediante la pestaña **Repositorio remoto existente**. También puede usar la pestaña **Solo local** para inicializar el repositorio de Git en local. En el último caso, Visual Studio simplemente agrega la carpeta `.git` y no realiza ninguna confirmación.

Crear un Proyecto para Subirlo a GitHub

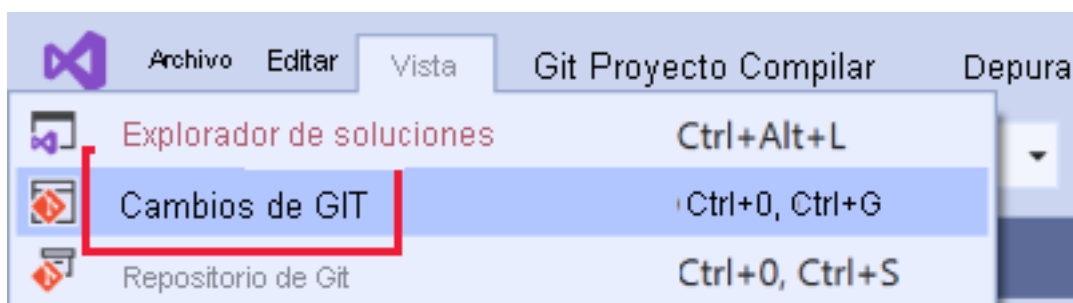
La forma más sencilla de empezar a programar en Visual Studio es con una plantilla de proyecto. Las plantillas de proyecto de Visual Studio proporcionan la técnica de scaffolding de código en soluciones de Visual Studio para facilitar la compilación y ejecución del proyecto. Aquí se va a crear un proyecto de aplicación de consola de .NET en C#.

1. En Visual Studio, vaya a **Archivo>Nuevo>Proyecto**.
2. En el cuadro de diálogo **Crear un nuevo proyecto**, escriba **aplicación de consola** en el cuadro de búsqueda y seleccione **Aplicación de consola de C#** en los resultados de la búsqueda. Luego, seleccione **Siguiente**.
3. En la página **Configurar el nuevo proyecto**, conserve los valores predeterminados y seleccione **Siguiente**.
4. En la página **Información adicional**, conserve los valores predeterminados y seleccione **Crear**.
5. Se crea el proyecto y se puede ver el contenido en el Explorador de soluciones.

Inserción del proyecto en GitHub

Ahora va a inicializar un repositorio de Git en el directorio raíz del proyecto, a crear un repositorio en GitHub y a insertar el contenido del repositorio en GitHub. Visual Studio tiene un cuadro de diálogo **Crear un repositorio Git** para hacer todo eso. Puede acceder a él por medio del menú **Git**, la ventana **Cambios de Git** o el control de barra de estado. Aquí se va a usar la ventana **Cambios de Git**.

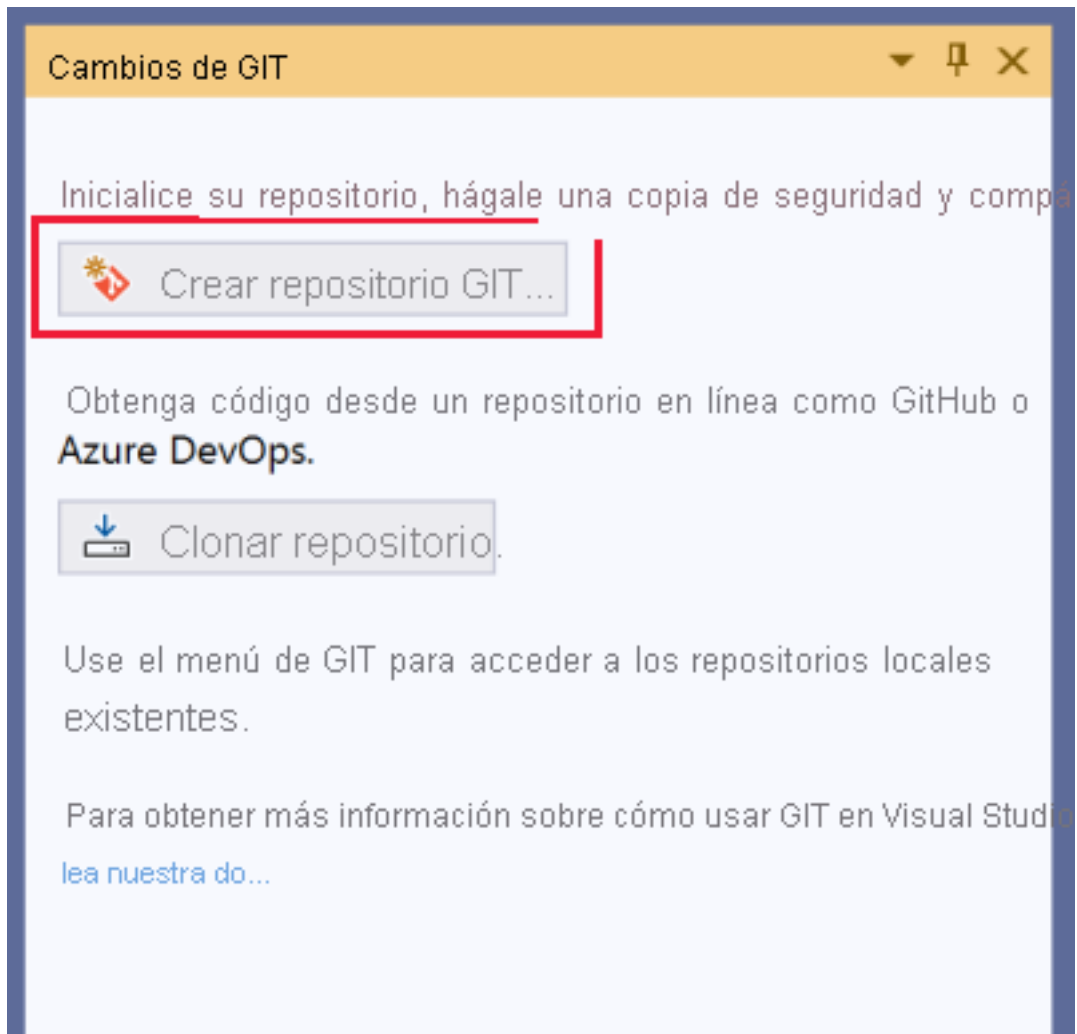
1. En Visual Studio, vaya a **Ver>Cambios de Git**.



Se abre la ventana **Cambios de Git**. En su estado vacío, la ventana proporciona puntos de entrada donde se pueden crear, clonar o abrir repositorios. Si la ventana está rellena, muestra los cambios en el repositorio abierto actualmente y permite confirmar.

INVESTIGACIÓN N° 1 ACERCA DE GITHUB

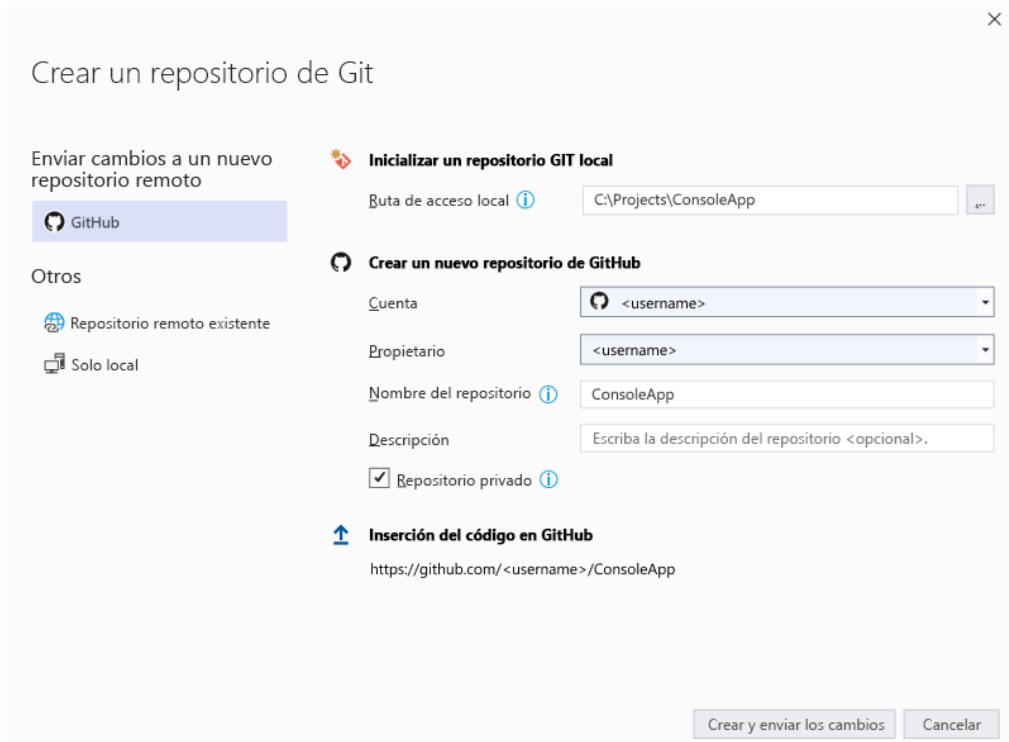
2. Seleccione el botón **Crear repositorio de Git**.



3. En el cuadro de diálogo **Crear un repositorio Git**, compruebe la ruta de acceso local del proyecto.

Visual Studio rellena automáticamente los cuadros **Cuenta** y **Propietario** porque ha iniciado sesión en GitHub. Compruebe el valor de **Nombre del repositorio** y asegúrese de que la casilla **Repositorio privado** esté activada.

INVESTIGACIÓN Nº 1 ACERCA DE GITHUB



4. Seleccione **Crear e insertar**.

El cuadro de diálogo se cerrará. Puede ver el progreso de la creación del repositorio en las ventanas **Cambios de Git** y **Salida**.

Comprobación de la creación del repositorio

Se puede validar la creación correcta del repositorio de forma local y remota.

1. Vaya a **Ver>Repositorio de Git**.



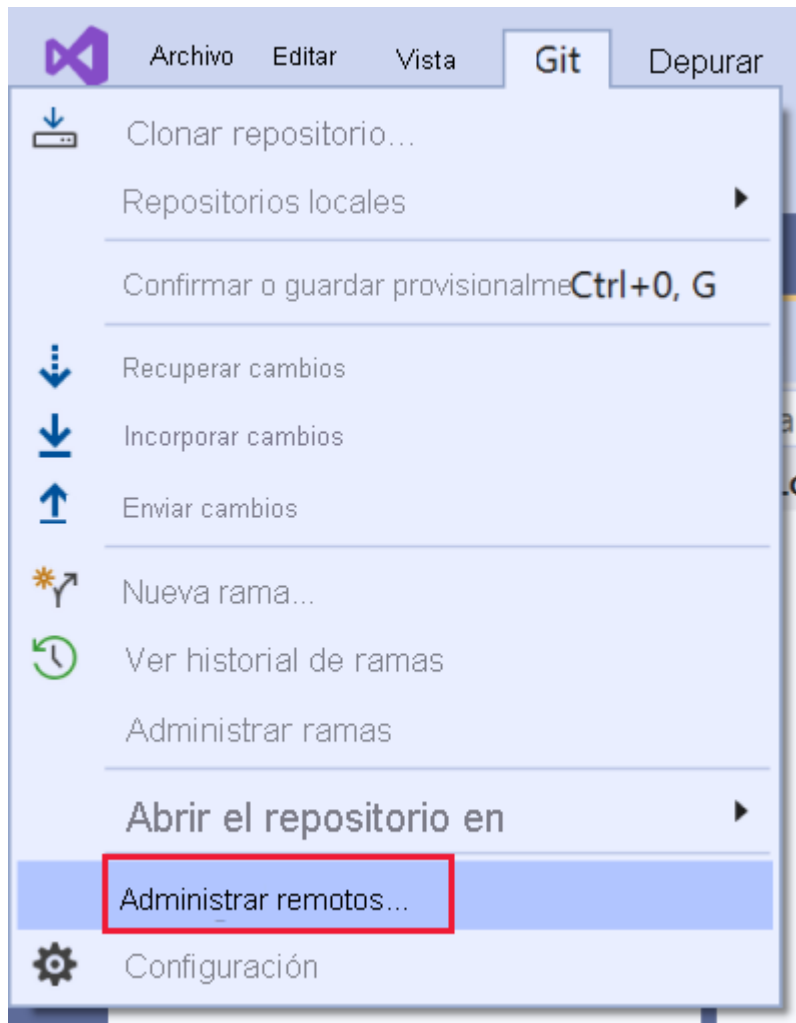
Se abre la ventana **Repositorio de Git**. Aquí puede ver todas las ramas locales y remotas del repositorio, junto con el historial de confirmaciones de cada rama.

2. Seleccione la rama predeterminada para actualizar su gráfico de historial. Ahora puede ver las dos confirmaciones iniciales que ha realizado Visual Studio para crear el repositorio.

INVESTIGACIÓN N° 1 ACERCA DE GITHUB

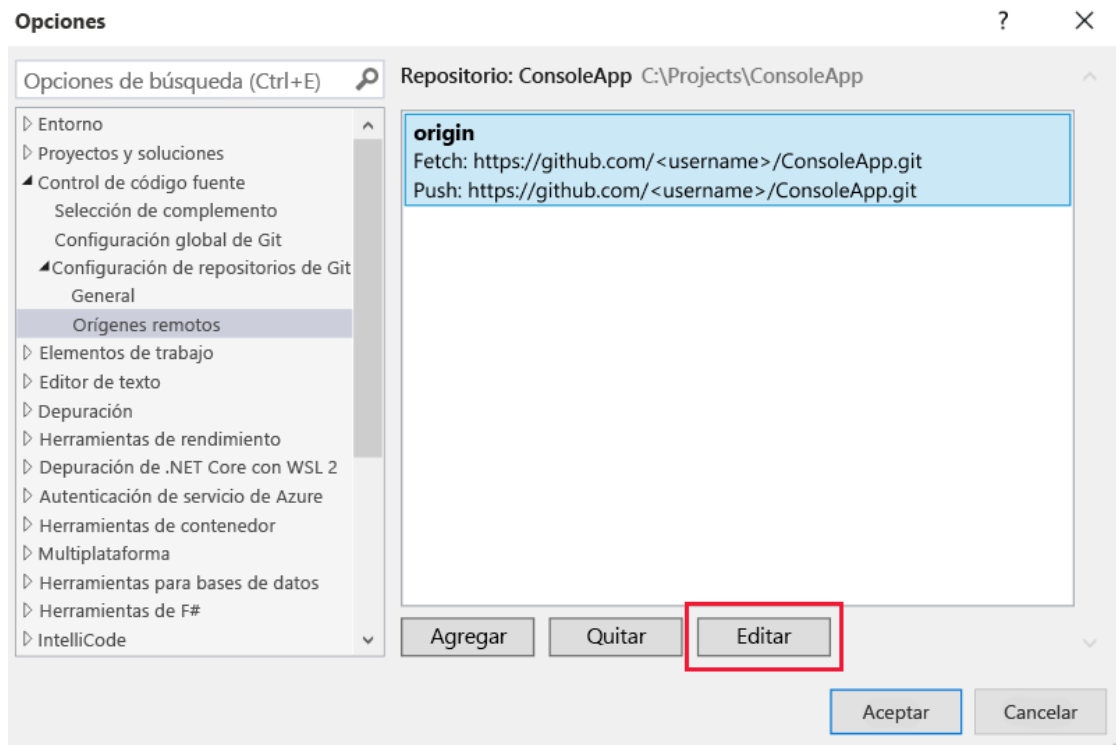


3. Vaya a **Git>Administrar orígenes remotos**.

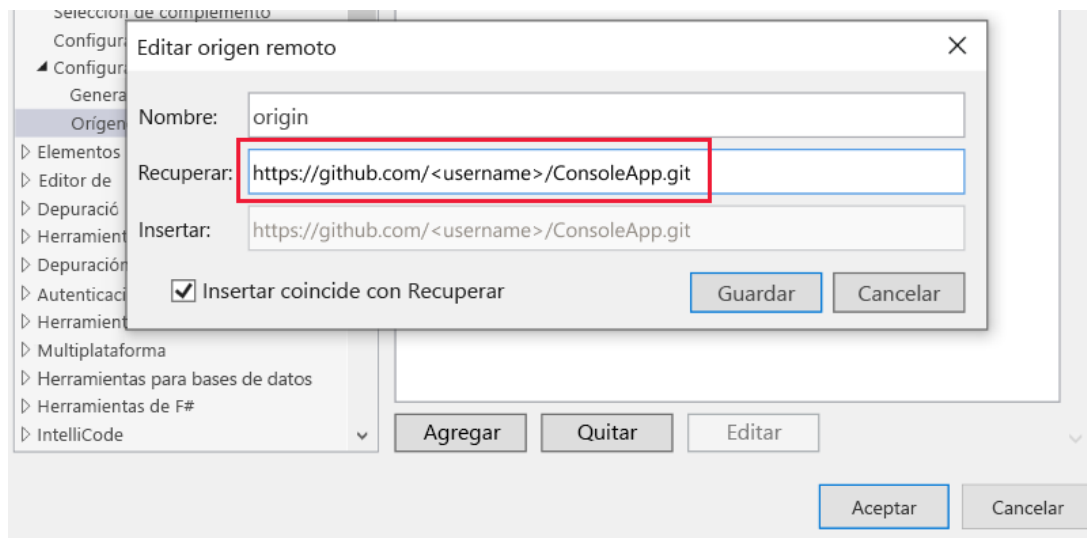


4. El cuadro de diálogo **Opciones** muestra que Visual Studio ha agregado un origen remoto al repositorio bajo el nombre **origin**. El origen remoto **origin** es el predeterminado que se va a usar para recuperar, extraer e insertar confirmaciones. Aquí también puede agregar varios orígenes remotos si usa un repositorio que se haya bifurcado de un origen remoto ascendente. Seleccione **Editar**.

INVESTIGACIÓN N° 1 ACERCA DE GITHUB

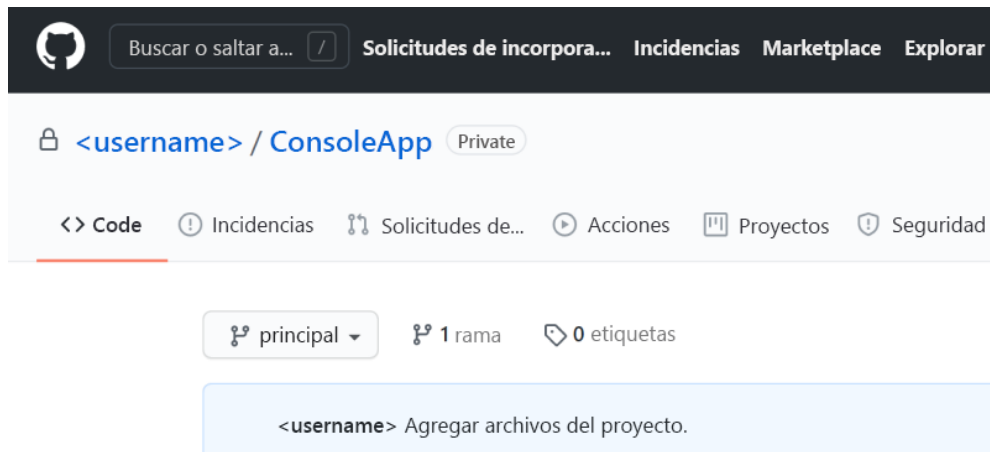


5. Copie la dirección URL del cuadro **Recuperar**.



6. Vaya al explorador web predeterminado y pegue el vínculo. La página web del repositorio se abre en GitHub.com.

INVESTIGACIÓN N° 1 ACERCA DE GITHUB

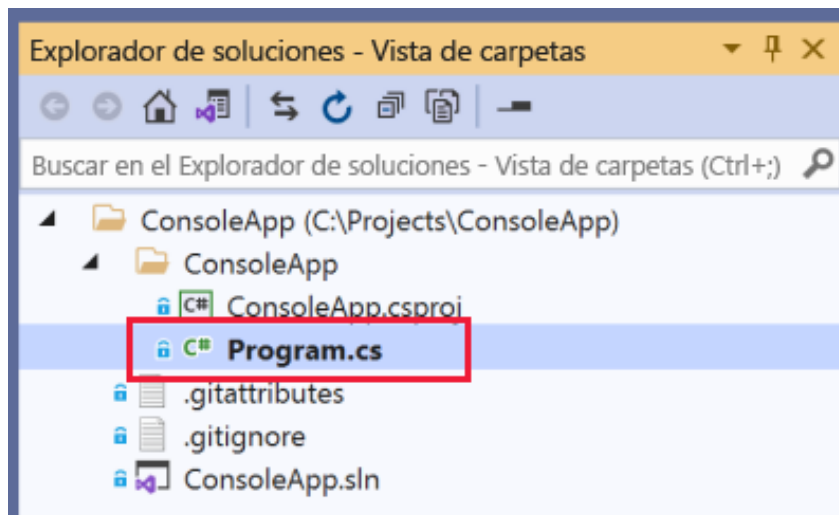


El repositorio de Git se ha creado y almacenado tanto en local como en remoto.

Confirmación de Cambios

Pasos:

1. Para abrir **Program.cs**, seleccione el archivo en el Explorador de soluciones.

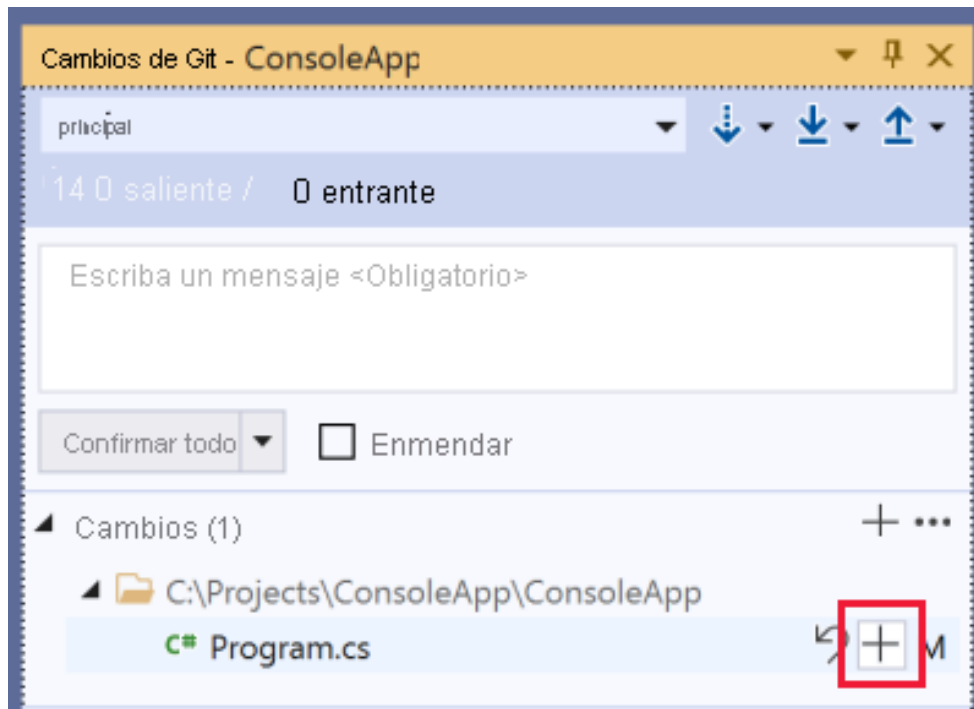


2. Cambie el mensaje de salida por "Hola a todos". Guarde el archivo.

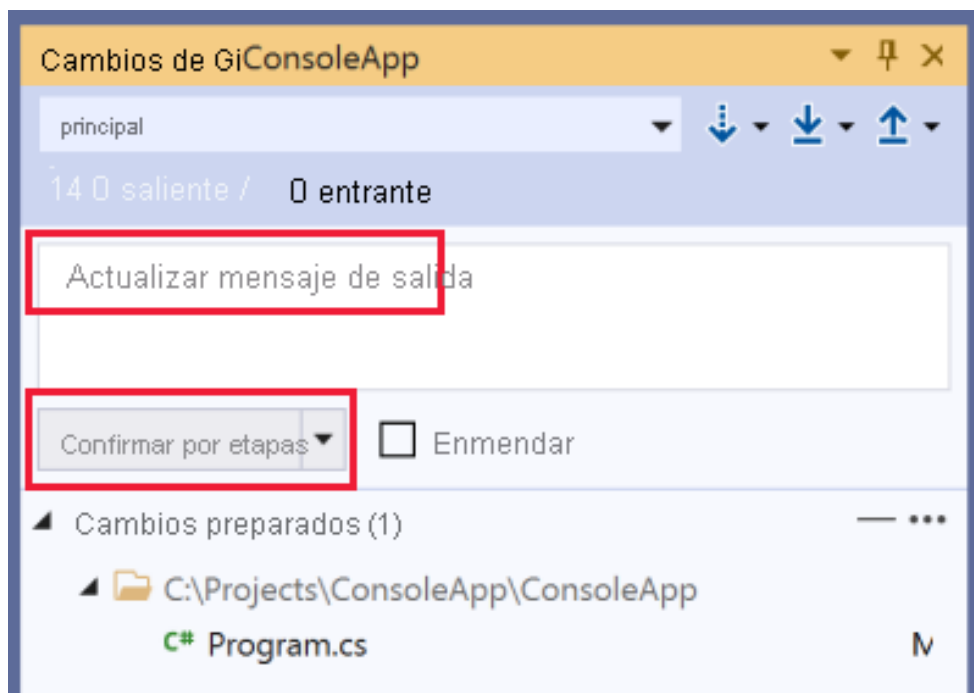


INVESTIGACIÓN N° 1 ACERCA DE GITHUB

- Abra la ventana **Cambios de Git** para ver los cambios pendientes. Seleccione el botón de signo más situado junto a **Program.cs** para almacenar provisionalmente el archivo.



- Ahora que los cambios se han almacenado provisionalmente, escriba un mensaje de confirmación en el cuadro de texto que en este momento indica **Update output message** (Actualizar mensaje de salida) y seleccione **Confirmar almacenados provisionalmente**.



Comprobación de la confirmación

Aparece una barra de información dorada en la ventana **Cambios de Git**, que indica que la confirmación se ha creado localmente. Seleccione el id. de confirmación para abrir los detalles de esta. También puede seleccionar el vínculo **1 salientes / 0 entrantes** situado encima del cuadro de texto para abrir la ventana **Repositorio de Git**.



Vea la confirmación de salida que ha creado en el grafo de confirmación de la rama actual para comprobar su creación.

Envío de cambios a GitHub desde Visual Studio

Después de confirmar en el repositorio, ha guardado de forma inherente instantáneas locales del código. El paso siguiente es insertar las confirmaciones en GitHub para almacenarlas como copias de seguridad o compartir el código con otros usuarios. Visual Studio usa el comando **git push** para cargar el contenido del repositorio local en el remoto.

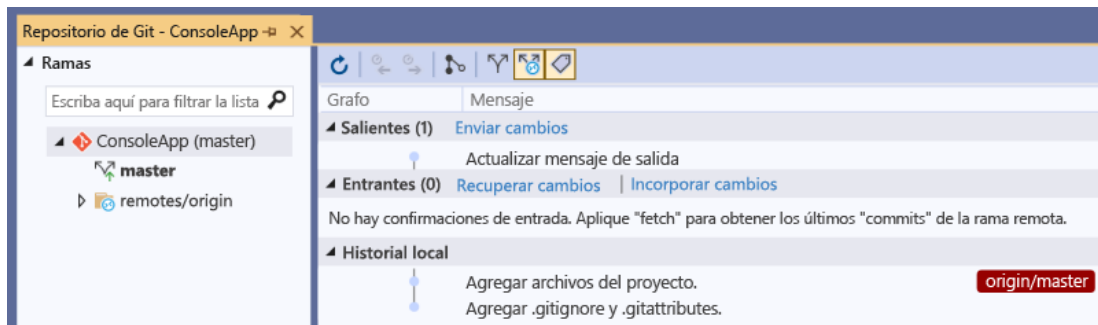
Sincronización de Git

Visual Studio le ayuda a mantener la rama local sincronizada con la rama remota por medio de las operaciones de descarga (recuperación y extracción) y de carga (inserción). Estas operaciones son visibles en la ventana **Repositorio de Git**.

En el panel principal de la ventana **Repositorio de Git** se muestra el **Historial local** de la rama seleccionada. Encima de **Historial local** hay dos secciones opcionales:

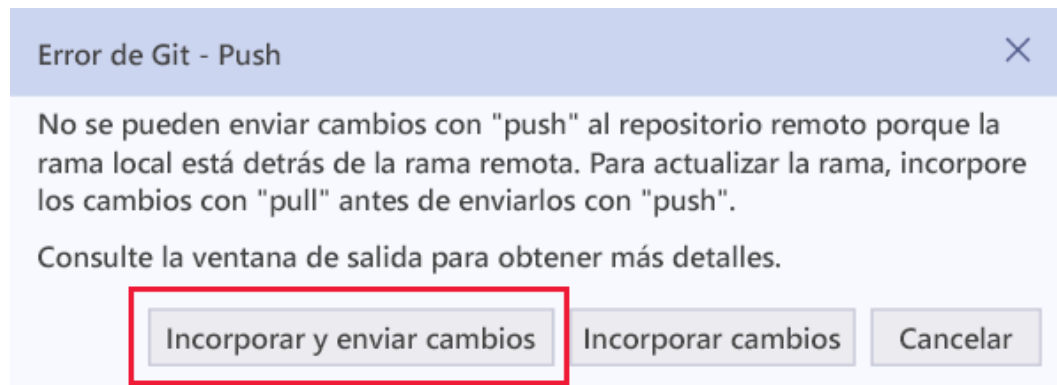
- En la sección **Salientes** se muestran las confirmaciones que todavía no se han insertado en el repositorio remoto.
- En la sección **Entrantes** se muestran las confirmaciones que se han recuperado pero que todavía no se han extraído del origen remoto.

INVESTIGACIÓN N° 1 ACERCA DE GITHUB



Antes de insertar es importante recuperar y extraer. La recuperación comprueba si hay confirmaciones remotas que deban incorporarse a los cambios locales. Si ve alguna, realice primero la extracción para evitar conflictos de combinación ascendentes.

Visual Studio no permite insertar confirmaciones si la rama local está detrás de la rama remota. Si intenta insertar, un cuadro de diálogo le pide que extraiga antes de hacerlo. El cuadro de diálogo incluye un botón **Extraer e insertar**.

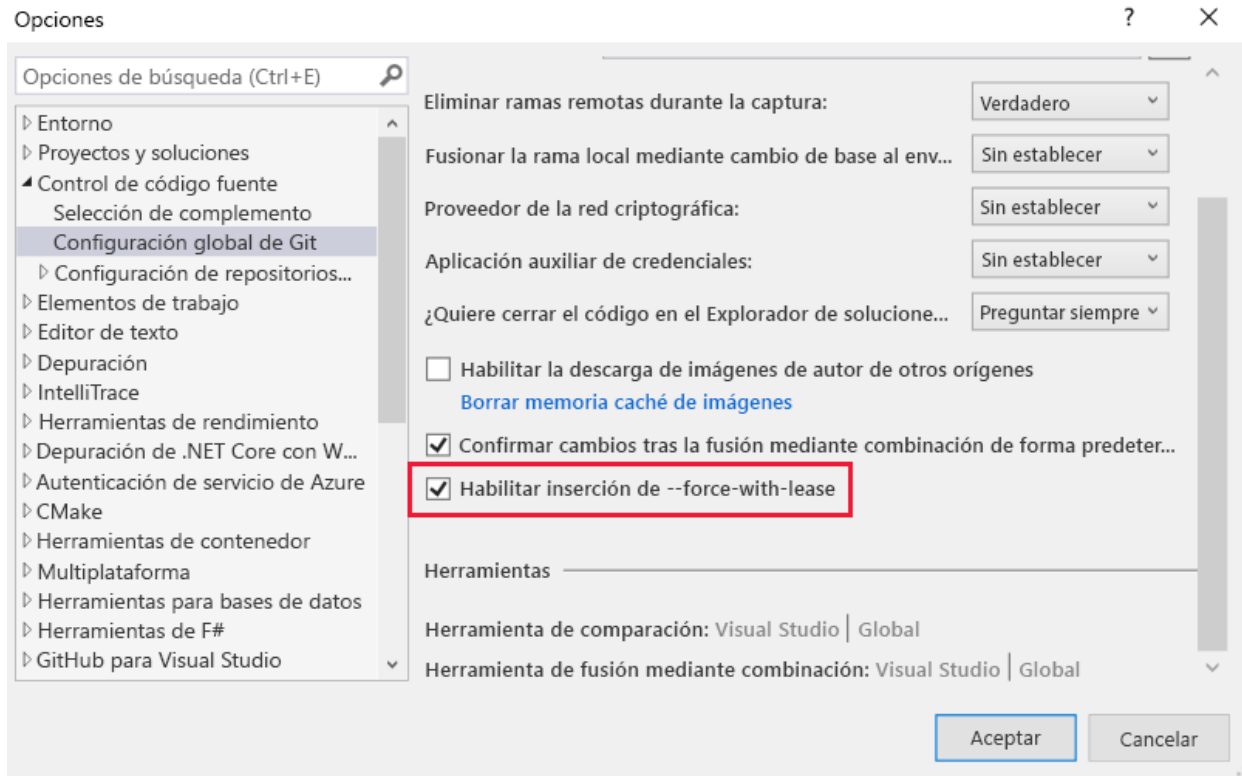


Envío de Cambios Forzado

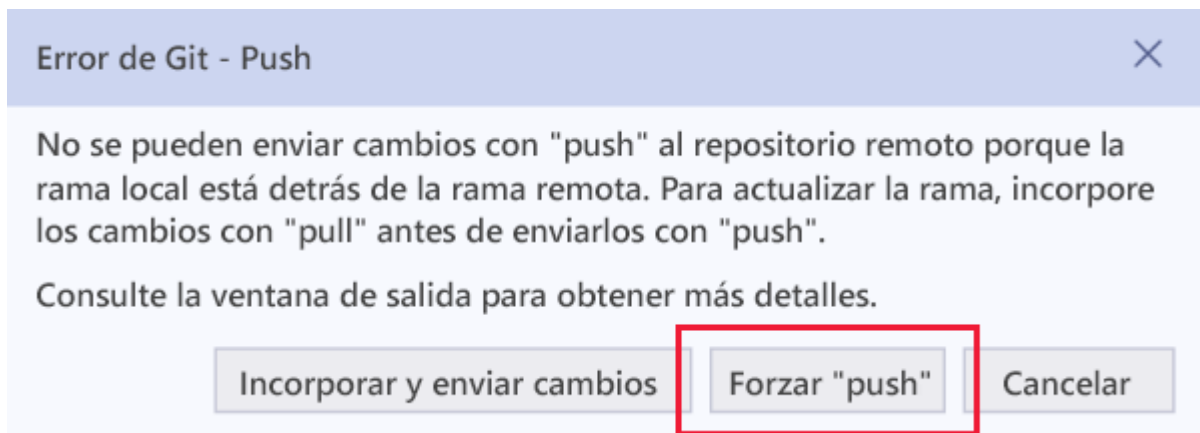
En ocasiones, es posible que quiera sobrescribir las confirmaciones en la rama remota con confirmaciones locales propias. Es posible que sea necesario cuando las confirmaciones que cargó anteriormente no sean correctas y quiera modificarlas ahora.

Puede usar la marca **--force** de Git, pero es un comando peligroso porque elimina las confirmaciones remotas que puedan haberse agregado desde la última extracción. En su lugar, Visual Studio proporciona la marca modificada **--force-with-lease**. Es una opción de marca más segura, ya que garantiza que no se sobrescriba el trabajo de otra persona al forzar el envío de cambios. Para habilitar el comando, vaya a **Git>Settings** (Configuración) y active la casilla **Enable push--force-with-lease** (Habilitar envío de cambios: forzar con concesión).

INVESTIGACIÓN Nº 1 ACERCA DE GITHUB



La próxima vez que intente insertar las confirmaciones que no se extrajeron anteriormente, el cuadro de diálogo que le pide que extraiga antes de insertar incluye un botón **Force Push** (Forzar envío de cambios).

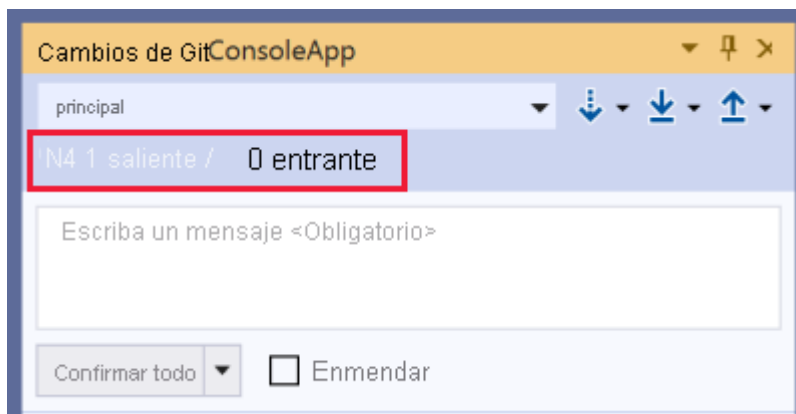


Pasos para Envío de Cambios:

La fase final de un flujo de trabajo de Git simple consiste en enviar los cambios al repositorio remoto. En este ejercicio, obtendrá información sobre cómo Visual Studio le ayuda a visualizar los cambios locales y a enviar las confirmaciones a la rama remota en GitHub.

INVESTIGACIÓN N° 1 ACERCA DE GITHUB

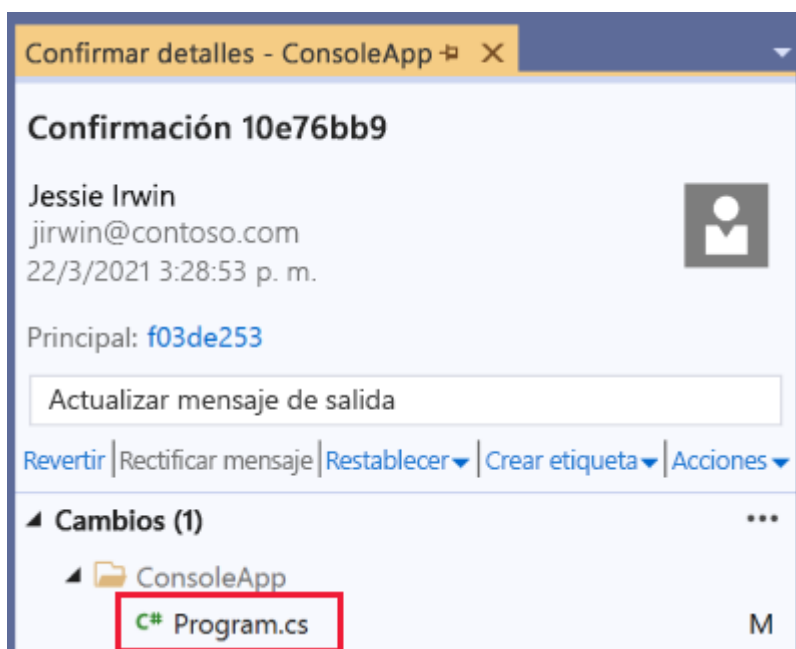
1. En la ventana **Cambios de Git**, seleccione el vínculo **1 salientes / 0 entrantes**.



2. En la ventana **Repositorio de Git**, en la sección **Salientes**, haga doble clic en la confirmación.

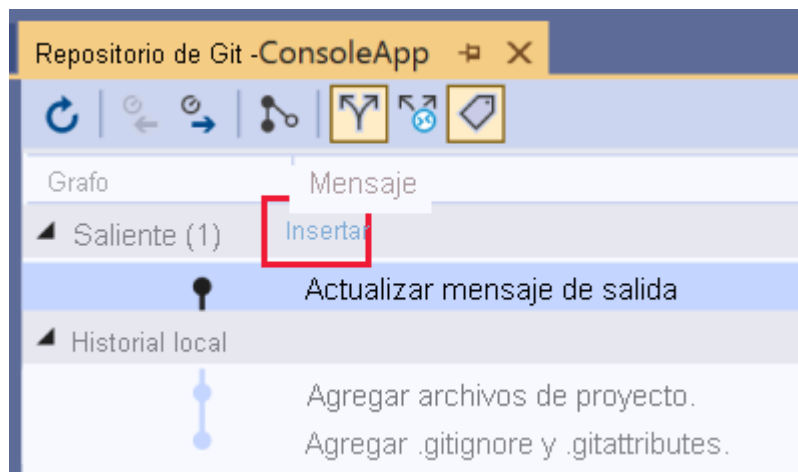


3. En la ventana **Detalles de la confirmación**, seleccione el archivo **Program.cs** para ver las diferencias entre su versión base y la confirmación.

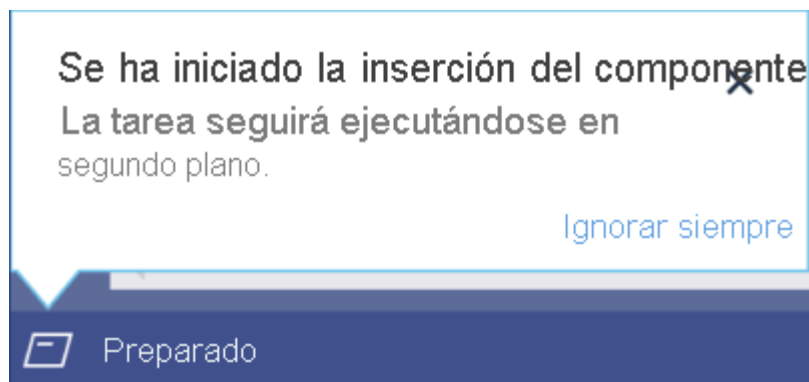


INVESTIGACIÓN N° 1 ACERCA DE GITHUB

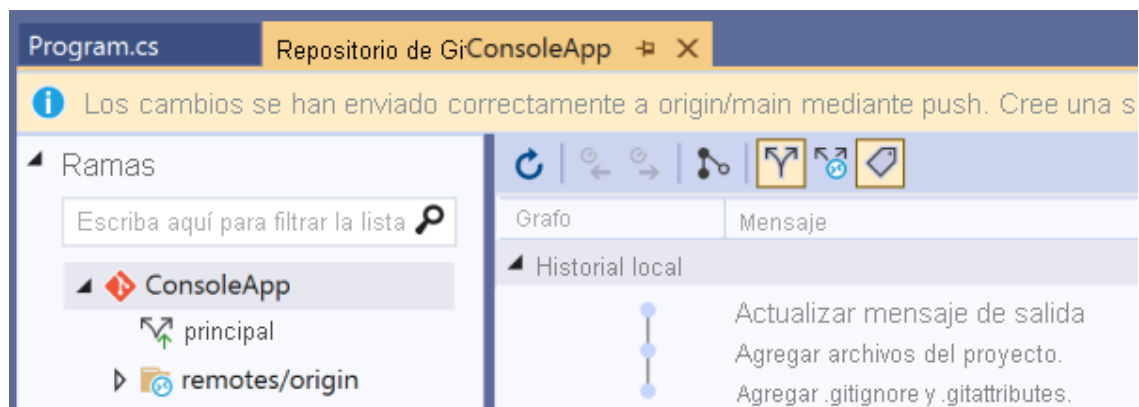
4. En la ventana **Repositorio de Git**, en la sección **Salientes**, seleccione **Insertar**.



5. Puede ver que la operación que se realiza en segundo plano por medio de la notificación del Centro de estado de tareas.



6. Para confirmar la finalización correcta de la operación, vea la barra de información dorada de la ventana **Repositorio de Git**.



COMANDOS

Identificar quien está trabajando

1. `git config --global user.name "NombreUsuario"`: Especifica el usuario que podrá realizar cambios (el comando se hace una única vez).
2. `git config --global user.email "EmailUsuario"`: Especifica el correo del usuario (el comando se hace una única vez).

Para Crear Repositorio con Git Local:

Creamos un repositorio en GitHub.

1. `git init`: Inicia el proceso de subida de repositorio.
2. `git status`: Conocer los archivos modificados.
3. `git add .`: Agregar los archivos al repositorio.
4. `git commit -m "Mensaje"`: Confirma los cambios realizados luego de subir los archivos al repositorio.

Para Crear Repositorio con Git Online:

Una vez creado el repositorio con Git Local:

1. `git remote add origin https://github.com/miUsuario/nombreRepositorio.git`: Crea el git online.
2. `git branch -M main`: Crear la rama main
3. `git push -u origin master`: Creamos la rama base y subimos los archivos.

Otros Comandos:

1. `git diff`: Muestra una comparación de dónde se ha realizado cambios.
2. `git reset`: Elimina la acción realizada por el comando add.
3. `git checkout --`: Devuelve todos los cambios almacenados desde el último commit.
4. `git log`: Muestra el listado de todos los commit. (Para salir, presionar "q").
5. `git commit --amend -m "nuevo mensaje"`: Permite modificar el último commit. (Luego deberá presionar "i" editar). Para salir, presionar tecla **ESC** + **:wq!**.
6. `git checkout -b nombreRama`: Crea una rama.
7. `git checkout master`: Mostrará solamente los archivos de la rama principal (Master).
8. `Git merge nombreRama`: Agrega la rama seleccionada a master. (Para salir presionar tecla **ESC** + **:wq!**)

INVESTIGACIÓN N° 1 ACERCA DE GITHUB

9. `git branch -d nombreRama`: Elimina la acción realizada por el comando add.
10. `git push`: Carga los archivos.
11. `git commit -am "Mensaje"`: Es la combinación de git add . y git commit -m "Mensaje"
12. `git clone url`: Clona repositorio de github (Comprimir y descargar)
13. `git pull origin nombreRama`: Comprueba si hay cambios en la rama nombreRama para luego descargarlos desde un remoto (lugar donde se encuentra esa rama)

RAMAS

Las ramas son simplemente una división de nuestro trabajo actual, con el objetivo de poder realizar en esa nueva división algunos cambios que no se están seguros que formarán parte de la rama principal.

Manejo de las Ramas

- Para poder hacer uso de una rama, lo primero que hay que hacer es crearla.
Comando a usar: `git branch nombreRama`
- Ahora debemos cambiarnos a la rama creada, pero antes debemos ver dónde nos encontramos.
Comando a usar: `git log -oneline --decorate --all --graph`
Aparecerá de una forma más visual todo el Proyecto y por encima de todo un "HEAD" que indica la rama en la que nos encontramos.
- Ahora sí nos movemos a la nueva rama creada.
Comando a usar: `git checkout nombreRama`
Para verificar el cambio de rama, solamente se escribe el comando antes usado. Si agregamos carpetas y/o archivos a la rama "nombreRama" y ahora nos regresamos a la rama principal usando checkout master, podremos notar que todo aquello que agregamos a la rama "nombreRama" desaparece, porque no pertenece al "master". Para que se muestren de nuevo todos los archivos, simplemente usamos checkout nombreRama.

Para poder simplificar comandos, o sea, crear la nueva rama y seleccionarla al mismo tiempo, podemos usar el siguiente comando: `git branch -d nombreRama`.

INVESTIGACIÓN Nº 1 ACERCA DE GITHUB

Unión de Ramas

Luego de crear una nueva rama y añadirle nuevos archivos, debemos agregarlos a nuestro proyecto.

- Primero nos movemos a la rama principal.
Comando a usar: `git checkout main`
- Luego, para unir la nueva rama usamos lo siguiente:
Comando a usar: `git merge nombreRama`

Eliminación de Ramas

Git nos permite eliminar una rama que no necesitemos, ya sea porque se agregó una rama que no era la que se debía subir o porque al implementarla no funcionó como se esperaba. Para hacerlo debemos hacer lo siguiente:

- Primero nos movemos a la rama principal.
Comando a usar: `git checkout master`
- Luego, para eliminar una rama usamos lo siguiente:
Comando a usar: `git branch -d nombreRama`

CONCLUSIONES:

- Se ha logrado dar las especificaciones necesarias para poder conocer qué es Github, cómo se usa, para qué se usa y cómo podemos usarla mediante Visual Studio.
- Se dieron a conocer los comandos básicos más usados a nivel de consola.
- La cuenta de Github ha sido creada de manera satisfactoria y está lista para su uso.

INVESTIGACIÓN Nº 1 ACERCA DE GITHUB

ANEXOS

Choose the plan that's right for you.

The screenshot displays three pricing plans for GitHub. The 'Free' plan is for individuals and organizations, offering unlimited public/private repositories, 2,000 Actions minutes/month (free for public repositories), 500MB of Packages storage (free for public repositories), and community support, at \$0 per month. The 'Team' plan, marked as 'MOST POPULAR', offers advanced collaboration for individuals and organizations, including everything in the Free plan plus protected branches, multiple reviewers in pull requests, draft pull requests, code owners, required reviewers, pages and wikis, 3,000 Actions minutes/month (free for public repositories), 2GB of Packages storage (free for public repositories), and web-based support, at \$4 per user/month. The 'Enterprise' plan is for security, compliance, and flexible deployment, offering everything in the Team plan plus automatic security and version updates, SAML single sign-on, advanced auditing, GitHub Connect, 50,000 Actions minutes/month (free for public repositories), 50GB of Packages storage (free for public repositories), and exclusive options like token, secret, and code scanning, and premium support, at \$21 per user/month. Each plan has a corresponding button: 'Create a free organization' for Free, 'Continue with Team' for Team, and 'Contact Sales' or 'Start a free trial' for Enterprise.

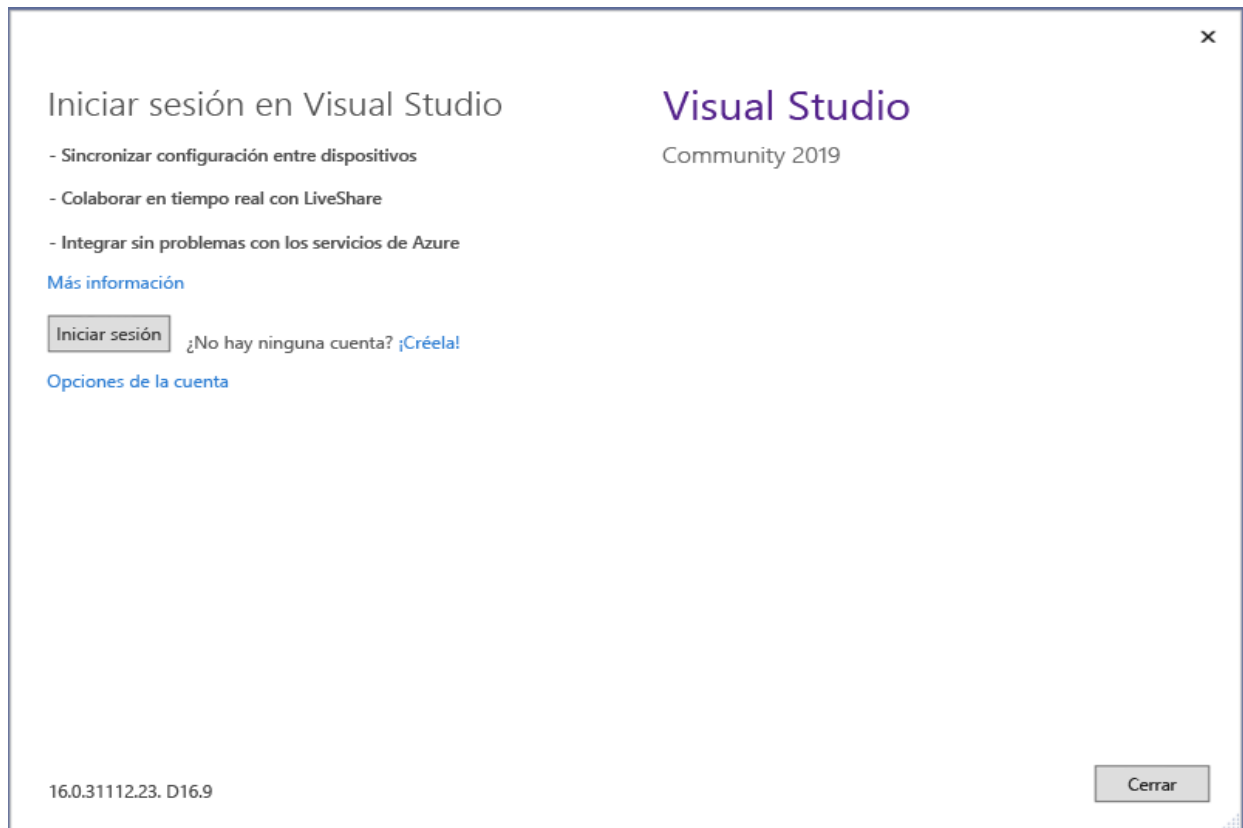
Plan	Target Audience	Key Features	Price	Action
Free	Individuals and organizations	Unlimited public/private repositories 2,000 Actions minutes/month (Free for public repositories) 500MB of Packages storage (Free for public repositories) Community support	\$0 per month	Create a free organization
Team	Advanced collaboration for individuals and organizations	Everything included in Free, plus... Protected branches Multiple reviewers in pull requests Draft pull requests Code owners Required reviewers Pages and Wikis 3,000 Actions minutes/month (Free for public repositories) 2GB of Packages storage (Free for public repositories) Web-based support	\$4 per user/month	Continue with Team
Enterprise	Security, compliance, and flexible deployment	Everything included in Team, plus... Automatic security and version updates SAML single sign-on Advanced auditing GitHub Connect 50,000 Actions minutes/month (Free for public repositories) 50GB of Packages storage (Free for public repositories) EXCLUSIVE OPTIONS Token, secret, and code scanning Premium support	\$21 per user/month	Contact Sales / Start a free trial

IMG 1.1

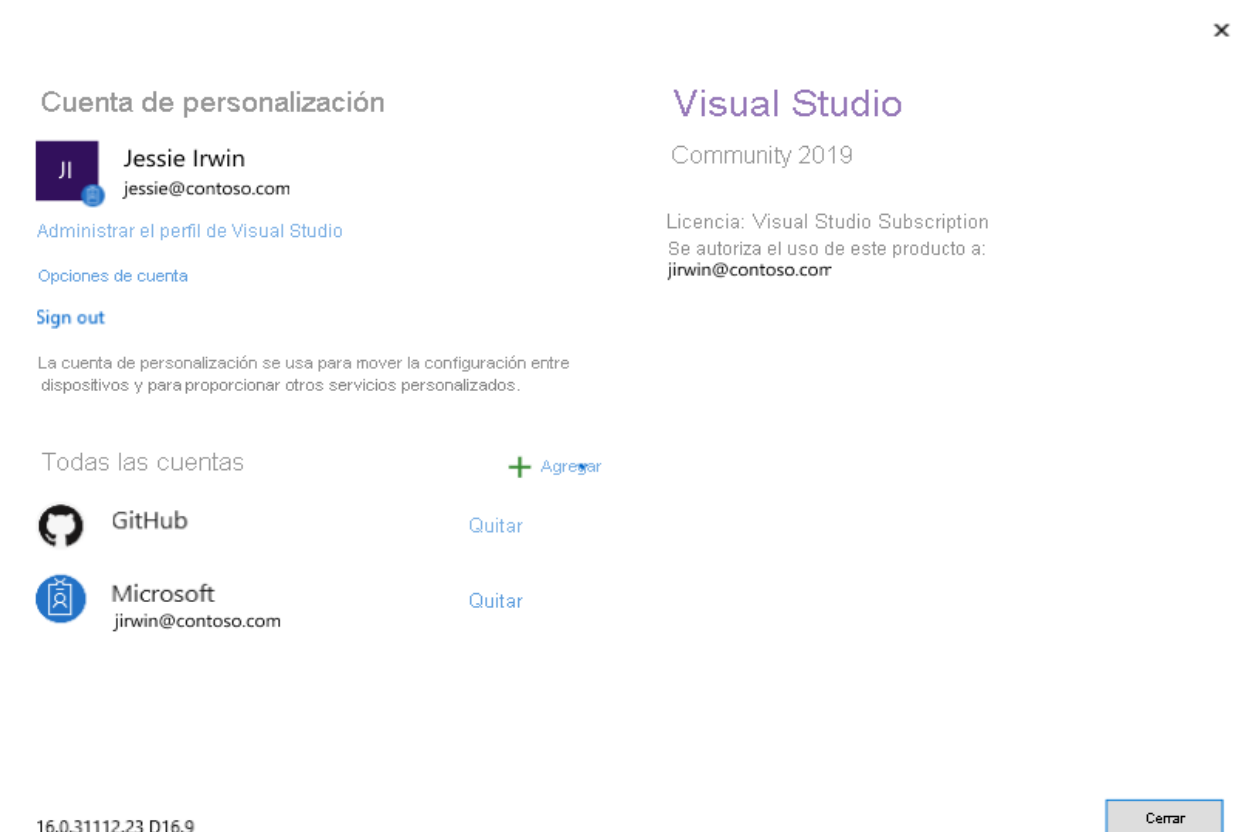
The screenshot shows the GitHub homepage with a dark blue background and a glowing globe with red and blue lines representing code connections. The main headline reads 'Where the world builds software'. Below it, a sub-headline states 'Millions of developers and companies build, ship, and maintain their software on GitHub—the largest and most advanced development platform in the world.' At the top, there is a navigation bar with links to Product, Team, Enterprise, Explore, Marketplace, and Pricing. A search bar and 'Sign in'/'Sign up' buttons are also present. At the bottom, there is a sign-up form with an 'Email address' input field and a 'Sign up for GitHub' button.

IMG 1.2

INVESTIGACIÓN N° 1 ACERCA DE GITHUB



IMG 2.1



IMG 2.2

BIBLIOGRAFÍA

Hostinger. (2021, 8 mayo). *¿Qué es GitHub y Cómo Usarlo?* Recuperado 29 de mayo de 2022, de <https://www.hostinger.es/tutoriales/que-es-github>

Microsoft. (2022, 16 marzo). *GitHub y Visual Studio*. Recuperado 29 de mayo de 2022, de https://docs.microsoft.com/es-es/visualstudio/version-control/?WT.mc_id=visualstudio_git_aka_vsgitdocs&view=vs-2022

Atlassian. (2020). *Qué es Git: conviértete en todo un experto en Git con esta guía*. Recuperado 29 de mayo de 2022, de <https://www.atlassian.com/es/git/tutorials/what-is-git>

Herrera, F. (2021). *Node: De cero a experto*. Udemy. Recuperado 2 de junio de 2022, de <https://www.udemy.com/course/node-de-cero-a-experto/>

Curso de Git y Github - 8 Ramas y Uniones (Merge). (2018, 4 marzo). YouTube. Recuperado 2 de junio de 2022, de <https://www.youtube.com/watch?v=BaY7FvCIH18&list=PLvimn1Ins-43-1sXQmGZPWLjNjPyGNi0R&index=8>