



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Ing. Julio de León Razo

Profesor:

Fundamentos de programación.

Asignatura:

13

Grupo:

1

No. de práctica(s):

Integrante(s):

Barradas Guemes Jose Eduardo
Nova Solís Jorge Iker
Ramos Carlos

4

No. de lista o brigada:

1

Semestre:

Miércoles 7 de septiembre de 2022.

Fecha de entrega:

Observaciones:

CALIFICACIÓN: _____

Práctica de Laboratorio Fundamentos de Programación

1. La computación como herramienta de trabajo del profesional de ingeniería

Objetivo:

El alumno conocerá y utilizará herramientas de software que ofrecen las Tecnologías de la Información y Comunicación que le permitan realizar actividades y trabajos académicos de forma organizada y profesional a lo largo de la vida escolar, tales como manejo de repositorios de almacenamiento y buscadores con funciones avanzadas.

Actividades:

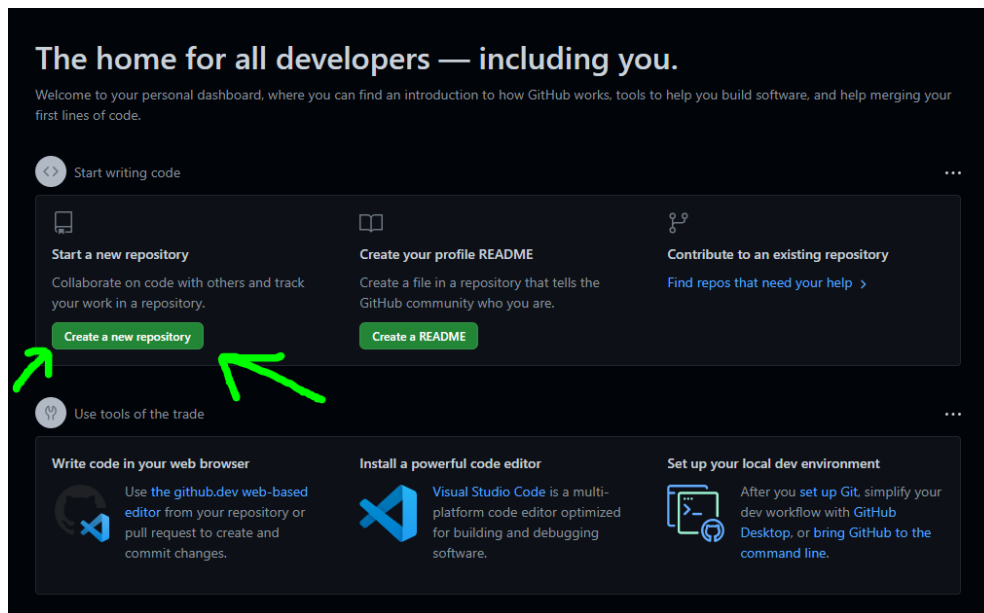
- Crear un repositorio de almacenamiento en línea.
- Realizar búsquedas avanzadas de información especializada.

Desarrollo:

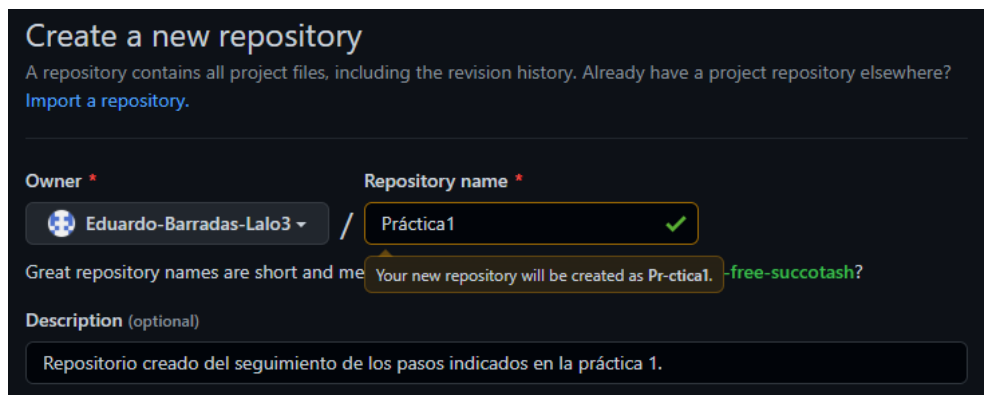
Actividad: Crear un repositorio de almacenamiento en línea.

1. Entrar al sitio web GitHub (<https://github.com>) y crear una cuenta.

2. Crear un repositorio.



3. Darle nombre al repositorio y agregar una descripción opcional.



4. Configurar el tipo de acceso del repositorio. Si solo será de uso privado elegir la opción “Privado” (Recomendable), si será para uso público, permitiendo a cualquier persona el ingreso, elegir la opción “público” (cuidado).

- ☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.
- ☐ **Private**
You choose who can see and commit to this repository.

5. Presionar en crear repositorio.

Create repository

6. Crear un nuevo archivo.

The screenshot shows the GitHub repository page for 'carlossramoss / Pratica-1'. The 'Quick setup' section is highlighted, providing instructions for creating a new repository. It includes options for 'Set up in Desktop', 'HTTPS', and 'SSH'. The 'creating a new file' link is highlighted with a green box and an arrow. Below this, there are three sections: '...or create a new repository on the command line', '...or push an existing repository from the command line', and '...or import code from another repository'. The 'Import code' button is also visible.

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH <https://github.com/carlossramoss/Pratica-1.git>

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# Pratica-1" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/carlossramoss/Pratica-1.git
git push -u origin main
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/carlossramoss/Pratica-1.git
git branch -M main
git push -u origin main
```

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

[Import code](#)

ProTip! Use the URL for this page when adding GitHub as a remote.

7. Nombrar el nuevo archivo como: "demo.c".

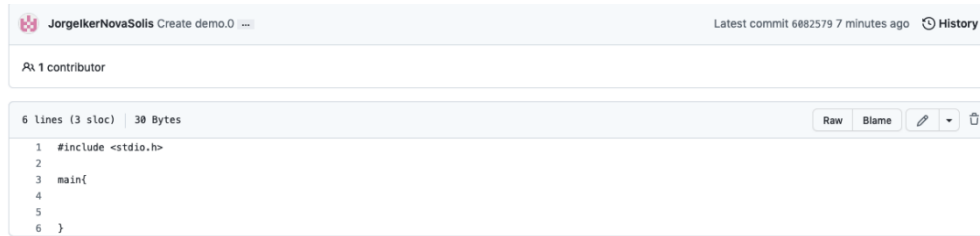
The screenshot shows the GitHub repository page for 'carlossramoss / Pratica-1'. The 'Code' tab is selected, and the 'demo.c' file is being created. The file name 'demo.c' is highlighted with a green box and an arrow. The 'main' branch is also highlighted. The 'Edit new file' button is visible at the bottom.

Pratica-1 / demo.c in **main**

[Cancel changes](#)

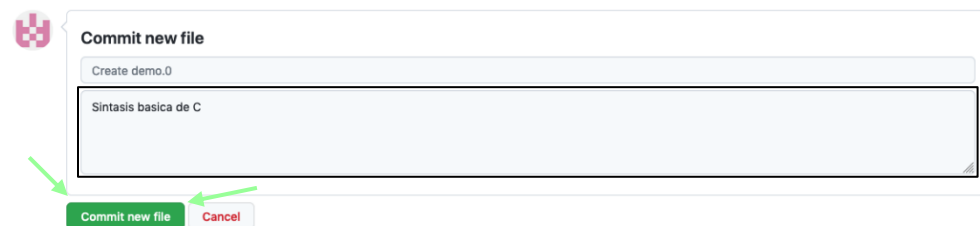
[Edit new file](#) [Preview](#) [Spaces](#) [2](#) [No wrap](#)

8. Agregar el código mostrado en el desarrollo de la práctica.



```
1 #include <stdio.h>
2
3 main{
4
5
6 }
```

9. Agregar un comentario y darle clic al botón de “commit new file”



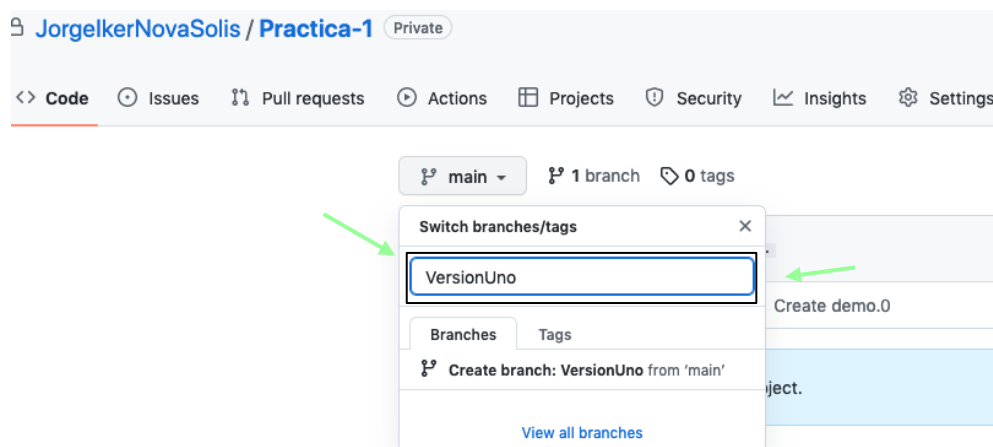
Commit new file

Create demo.0

Sintaxis basica de C

Commit new file Cancel

10. Crear una nueva rama (Branch). Darle el nombre de “VersionUno”.



JorgelkerNovaSolis / Practica-1 Private

Code Issues Pull requests Actions Projects Security Insights Settings

main 1 branch 0 tags

Switch branches/tags

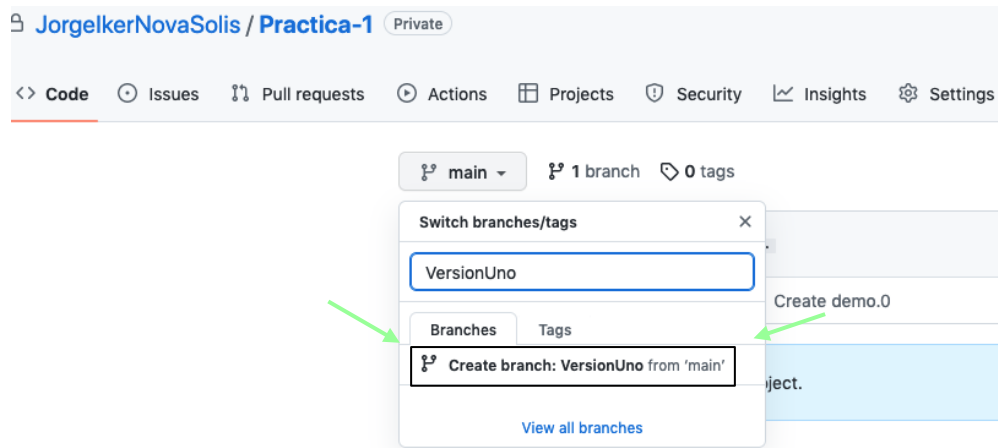
VersionUno

Branches Tags

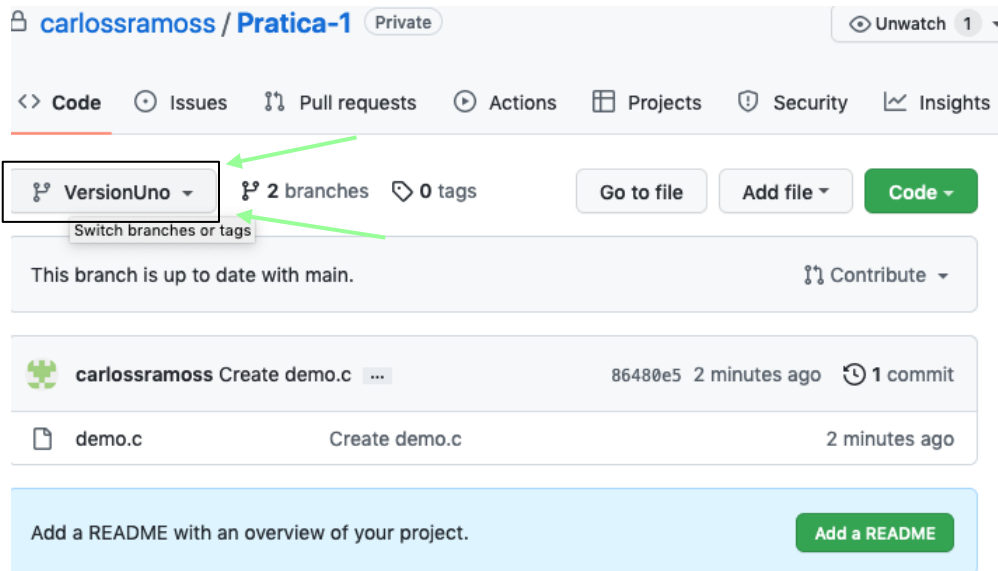
Create branch: VersionUno from 'main'

View all branches

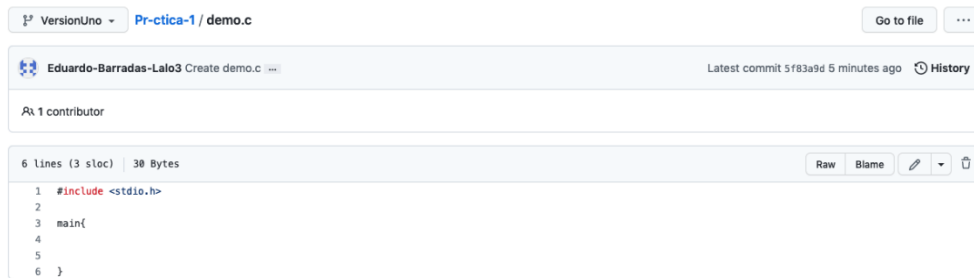
11. Dar clic en “Create Branch: “” ‘from main””



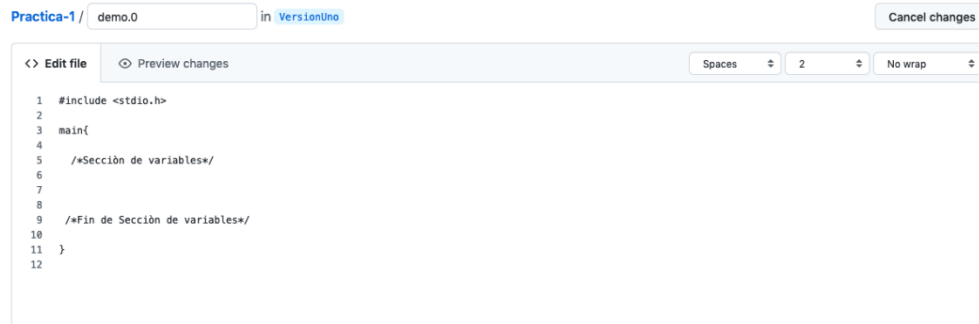
12. Verificar que estemos en la rama creada.



13. Dar clic sobre el archivo “demo.c”. En donde aparece lo subido en el archivo con el mismo nombre que se encuentra en la rama principal.



14. Dar clic en el símbolo de editar (Lápiz) y agregar las líneas de código indicadas en el desarrollo de la práctica.

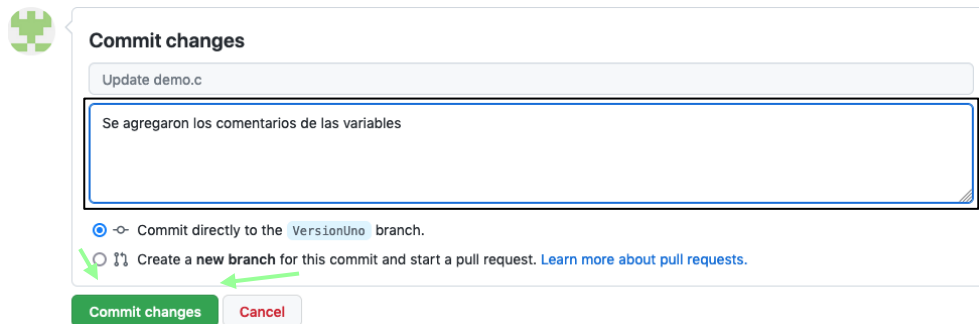


The screenshot shows a code editor interface. At the top, there's a tab labeled 'Practica-1 / demo.0' and a 'Cancel changes' button. Below the tab bar, there are two tabs: 'Edit file' (active) and 'Preview changes'. The code is as follows:

```
1 #include <stdio.h>
2
3 main{
4
5     /*Sección de variables*/
6
7
8
9     /*Fin de Sección de variables*/
10
11 }
12
```

On the right side of the editor, there are settings for 'Spaces' (set to 2) and 'No wrap'.

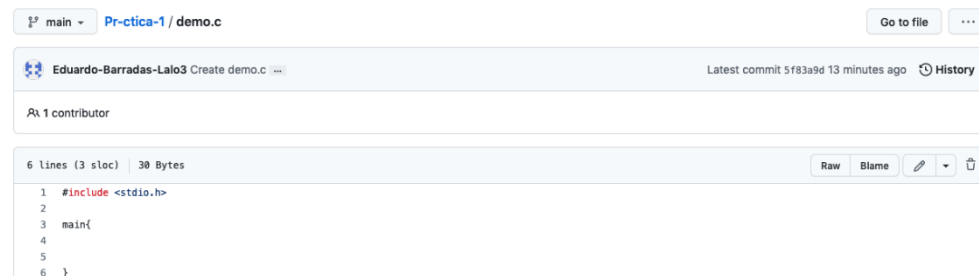
15. Agregar un comentario y dar clic en “Commit changes”.



The screenshot shows a 'Commit changes' dialog box. It has a title bar with a green icon and the text 'Commit changes'. Below the title bar, there's a text input field containing 'Update demo.c'. Below that, there's a larger text area containing the comment 'Se agregaron los comentarios de las variables'. At the bottom, there are two radio buttons: the first is selected and labeled 'Commit directly to the VersionUno branch.', and the second is labeled 'Create a new branch for this commit and start a pull request. Learn more about pull requests.' Below the radio buttons, there are two buttons: 'Commit changes' (green) and 'Cancel' (red). A green arrow points to the 'Commit changes' button.

16. Verificar en cada rama “main” y “VersionUno” que se encuentran dos archivos con el mismo nombre, pero cada uno con contenido diferente.

Rama main:





The screenshot shows the GitHub file view for 'demo.c' on the 'main' branch. At the top, there's a breadcrumb 'main / Practica-1 / demo.c' and a 'Go to file' button. Below that, there's a commit summary by 'Eduardo-Barradas-Lalo3' with the message 'Create demo.c ...'. The latest commit hash is '5f83a9d' and it was committed '13 minutes ago'. Below the commit summary, it says '1 contributor'. The file content is shown in a code block:

```
1 #include <stdio.h>
2
3 main{
4
5
6 }
```

At the bottom right, there are buttons for 'Raw', 'Blame', and a dropdown menu.

Rama VersionUno:

 VersionUno ▾ **Pratica-1 / demo.c**

 **carlossramoss** Update demo.c ...



1 contributor

11 lines (5 sloc) | 100 Bytes

```
1  #include <stdio.h>
2
3  main{
4
5      /*Seccion de variables*/
6
7
8
9      /*Fin seccion de variables*/
10
11 }
```

17. Investiga que es y para qué sirve un “Pull Request”. Muestra la evidencia de que lo hiciste.

Los pull requests son una funcionalidad que facilita la colaboración entre desarrolladores que usan Bitbucket. Ofrecen una interfaz de web intuitiva para debatir los cambios propuestos antes de integrarlos en el proyecto oficial.

[Todos](#) [Imágenes](#) [Videos](#) [Noticias](#) [Libros](#) [Más](#) [Herramientas](#)

Cerca de 452,000,000 resultados (0.34 segundos)

<https://aprendegit.com/que-es-un-pull-request>
¿Qué es un pull request? - Aprende GIT
14 feb 2013 — Un pull request es una petición que el propietario de un fork de un repositorio hace al propietario del repositorio original para que este ...

Preguntas relacionadas

[¿Qué es un Pull Request?](#) ▾

[¿Qué es el Pull Request en git?](#) ▾

[¿Cómo hacer un Pull Request?](#) ▾

[¿Qué es un push request?](#) ▾

[Comentarios](#)

<https://www.atlassian.com/git/making-a-pull-request>
Pull-requests | Atlassian Git Tutorial
Las pull requests son una funcionalidad que facilita la colaboración entre desarrolladores que usan Bitbucket. Ofrecen una interfaz web intuitiva para ...

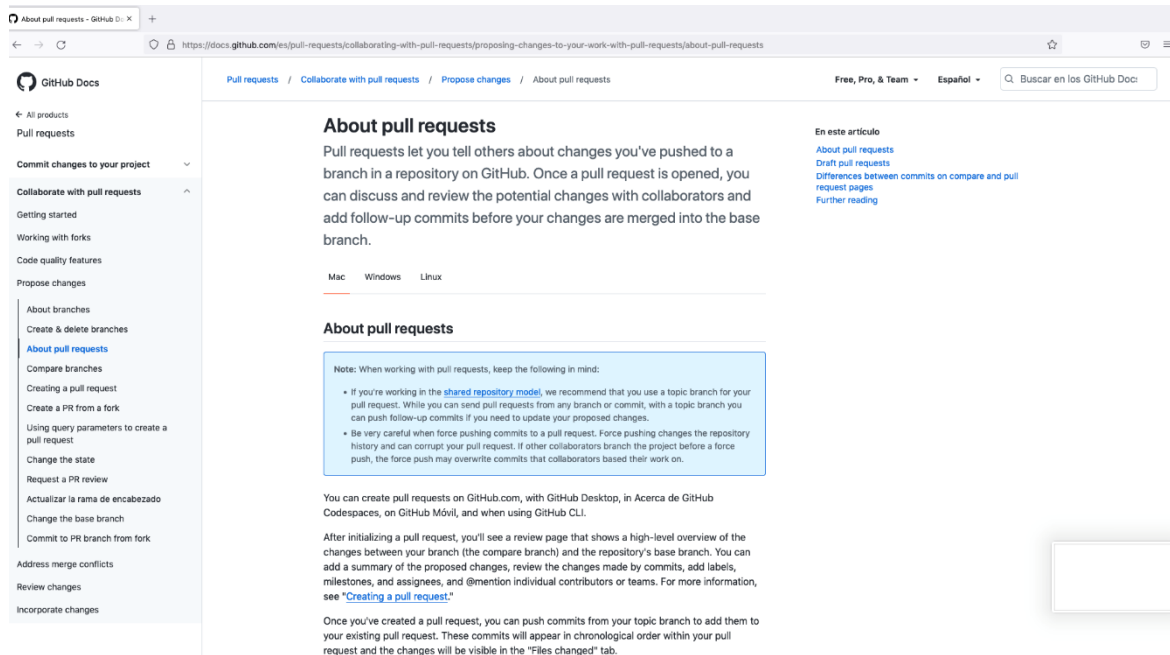
para que sirve pull request en github

[Todos](#) [Videos](#) [Imágenes](#) [Noticias](#) [Shopping](#) [Más](#) [He](#)

Cerca de 46,000,000 resultados (0.45 segundos)

Los pull request permiten no solo llevar de forma más ordenada las tareas en la etapa del desarrollo, sino también crear propuestas o cambios que puedan ser integrados posteriormente a dicho proyecto. Básicamente un pull request es una petición para integrar nuestras propuestas o cambios de código a un proyecto. 28 jul 2015

<https://styde.net/pull-request-en-github>
Pull request en GitHub - Styde.net



18. Realiza un “Merge pull request” muestra la evidencia e indica si existe algún riesgo.

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).

base: main

compare: VersionUno

✓ Able to merge. These branches can be automatically merged.

Discuss and review the changes in this comparison with others. [Learn about pull requests](#) Create pull request

1 commit

1 file changed

1 contributor

Commits on Aug 24, 2022

Update demo.0

JorgeIkerNovaSolis committed 8 minutes ago

Verified

1a5d2da

<>

Showing 1 changed file with 6 additions and 1 deletion.

Split

Unified

deno.0

@@ -2,5 +2,10 @@

2 2

3 3 main{

4 4

5 -

5 + /*Sección de variables*/

6 +

7 +

8 +

9 + /*Fin de Sección de variables*/

10 +

6 11 }

Un riesgo posible es que al realizar este proceso ocurriría una superposición entre los repositorios creados, cualquier persona con acceso podría crearlo llevando a un adelanto en el proceso final, aunque el código de alguno no se encuentre terminado.

Actividad: Realizar búsquedas avanzadas de información especializada.

1. Crea una carpeta en GoogleDrive y compártela con tu brigada.

Nombre ↓



Prácticas Fundamentos de programación brigada 4

2. Crea un archivo en documentos (tipo word) y cada integrante deberá agregar una búsqueda especializada del buscador de Google indicando descripción y captura de pantalla. Deberán ser diferentes a las comentadas en la guía práctica.

Mi unidad > Prácticas Fundamentos de programación brigada 4 > Práctica 1 ▾ 👤

Nombre ↓



Lalo.doc 👤



Iker.doc 👤



Carlos.docx 👤

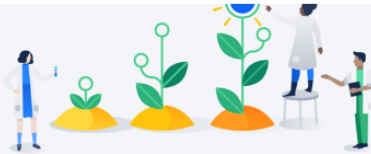


Búsquedas avanzadas de información especializada 👤

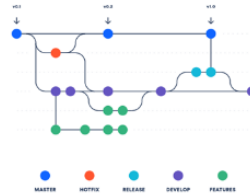
Software de control de versiones para equipos profesionales

Bitbucket admite Sistemas de control de versiones para Git.

[Más información →](#)



Un sistema de control de versiones lo VCS, por sus siglas en inglés, también conocido como sistema de control de revisiones o de fuentes, es una herramienta de software que monitoriza y gestiona cambios en un sistema de archivos. Asimismo, un VCS ofrece herramientas de colaboración para compartir e integrar dichos cambios en otros usuarios del VCS. Al operar al nivel del sistema de archivos, un VCS monitorizará las acciones de adición, eliminación y modificación aplicadas a archivos y directorios. Un repositorio es un término del VCS que describe cuando un VCS está monitorizando un sistema de archivos. En el alcance los archivos individuales de códigos fuente, un VCS monitorizará las adiciones, eliminaciones y modificaciones de las líneas de texto que contiene ese archivo. Entre las opciones populares de VCS del sector de software, se incluyen Git, Mercurial, SVN y preface.



Definición: ¿qué significa Application Client?

Un cliente de aplicación es una aplicación independiente que se ejecuta en la máquina del cliente y está configurada para funcionar como un componente J2EE.

El cliente de la aplicación se utiliza para realizar tareas, como la administración del sistema o la aplicación. Típicamente se incluye con componentes empresariales sólidos, como servicios web y EJB, que se implementan en un servidor de aplicaciones remoto.

Inicio

Repositorios

Comunidad

Recursos

Empresas

Seguridad

Blog

GitHub

InicioRepositoriosComunidadRecursosEmpresasSeguridadBlogGitHub

https://github.com/usuario/repositorio/commit/commit-hash

Empresaz / Comenzar con Git / Administrar repositorios remotos

Free, Pro, & TeamEspañol

Buscar en los GitHub Docs

Comenzar con Git

Configura tu nombre de usuario

Guardar credenciales en caché

Contraseña de Git

Credenciales de Keychain de macOS

Flujos de trabajo de Git

Acerca de los repositorios remotos

Administrar repositorios remotos

Editores de texto asociados

Manejar los extremos de línea

Ignorar archivos

Utilizar GitHub

Personaliza tu flujo de trabajo

Privacidad en GitHub

Administrar repositorios remotos

Aprende a trabajar con tus repositorios locales en tu computadora y repositorios remotos alojados en GitHub.

MacWindowsLinux

Agregar un repositorio remoto

Para agregar un remoto nuevo, utiliza el comando `git remote add` en la terminal, en el directorio en el cual está almacenado tu repositorio.

El comando `git remote add` toma dos argumentos:

- Un nombre remoto, por ejemplo, `origin`
- Una URL remota, por ejemplo, `https://github.com/usuario/repo.git`

Por ejemplo:

En este artículo

Agregar un repositorio remoto

Cambiar la URL del repositorio remoto

Renombrar un repositorio remoto

Ejemplo

Eliminar un repositorio remoto

Ejemplo

Leer más