

CI-2125

Introducción
al
Laboratorio



Requisitos

Tener computadora

Haber llenado la encuesta, enviándola de vuelta

Haber leído la presentación: Herramientas de Programación.pdf

Nota:

Por favor, **reporten cualquier error** que puedan conseguir aquí
incluso si no están seguros

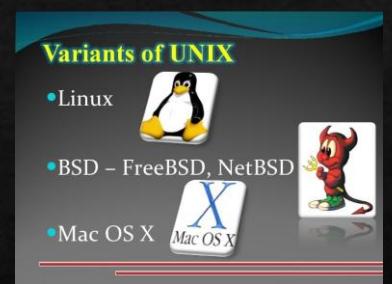
Me tardé mucho tiempo en revisar y hacer cambios debidos a que
estamos haciendo algunas cosas diferente del trimestre pasado.

Y los cambios han sido ¡Para simplificarles el proceso!

Puede que al ver este material se sorprendan de que era aun más
complicado el trimestre pasado, pero es la verdad.

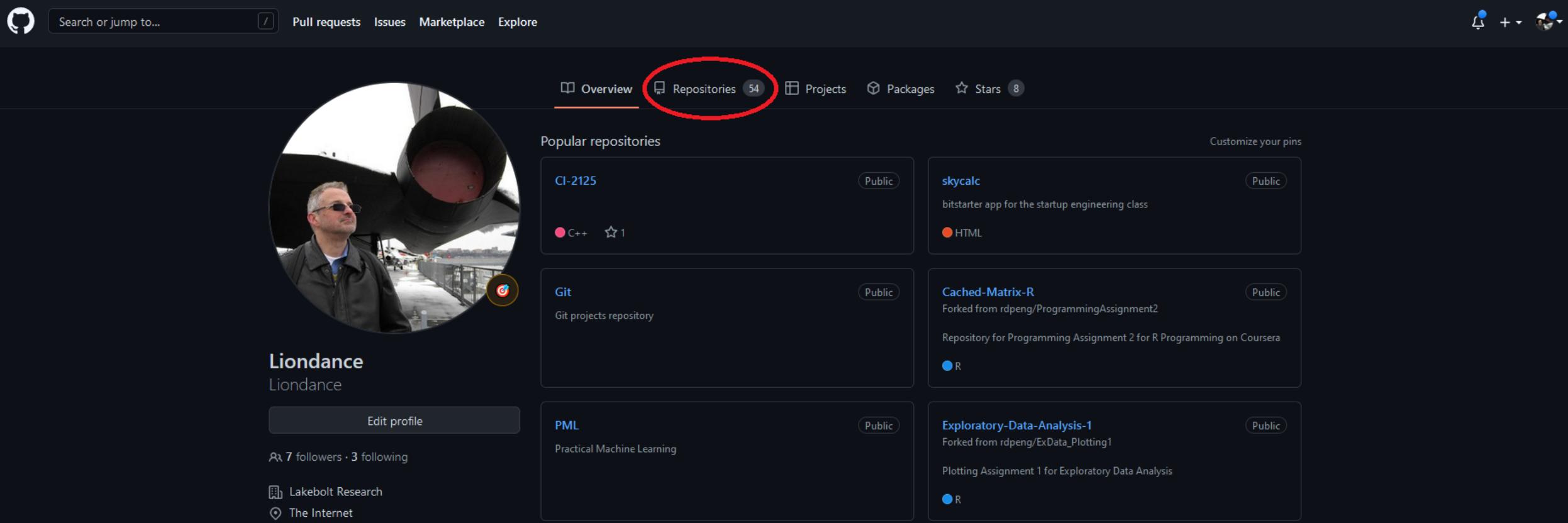
Primeros pasos: (a) instalar a Git

- ❖ Instalar Git
 - ❖ <https://git-scm.com/downloads>
- ❖ En Windows, lo anterior también debería instalar Git Bash automáticamente
 - ❖ Git Bash es un “shell” al estilo Unix, es decir una interfaz para líneas de comandos (CLI)
 - ❖ De hecho, es un “porte aproximado” de Bash: [https://en.wikipedia.org/wiki/Bash_\(Unix_shell\)](https://en.wikipedia.org/wiki/Bash_(Unix_shell))
- ❖ En plataformas *x (Unix, Linux, etc.) no necesitan Git Bash: ¡ya tienen el Bash Shell!
 - ❖ Usuarios de Apple, el Mac OS X pertenece a la familia Unix: ya deben tener un shell allí
- ❖ A lo largo de todo el laboratorio, vamos a usar a Git Bash para ejecutar comandos
 - ❖ Excepto, claro esta, cuando queremos ilustrar ambientes integrados como MS Visual Studio
 - ❖ Los “shell” son lo más cercano a un mínimo común denominador para operar en varias plataformas
- ❖ Busquen en la Web si necesitan ayuda adicional para instalar Git y Git Bash
 - ❖ Aquí esta un enlace prometedor: <https://www.gitkraken.com/blog/what-is-git-bash>



Primeros pasos: (b) crear una cuenta en GitHub

- ❖ La gran mayoría de ustedes ya han creado su cuenta
- ❖ Si no, visiten <https://github.com/> y sigan los pasos para hacer “Sign up”
- ❖ En la página principal de su cuenta consigan el enlace para ver los repositorios (oval rojo)



Primeros pasos: (c) crear un nuevo repositorio en GitHub

- ❖ Vayan a la pagina (su pagina, no la mía, obviamente) de sus **repositorios** en GitHub
- ❖ Noten el bello botón **verde** que dice “New” ... bueno, por lo menos es verde y dice “New” en *mi* pagina con *mi* tema de colores.
- ❖ Correcto, como ya se han imaginado, allí es donde deben hacer “click” ... ¡denle ya! (peeled potato will continue in the next slide)

The screenshot shows a GitHub user profile with the following details:

- User Profile:** A circular profile picture of a man wearing sunglasses and a dark jacket, with the name "Liondance" and the handle "Liondance" below it.
- Repositories:** The "Repositories" tab is selected, showing 54 repositories. The repositories listed are:
 - CI-2125** (Public): C++ repository, 1 star, updated 21 hours ago.
 - vidly-client** (Public): JavaScript repository, updated 25 days ago.
 - Lakebolt-www** (Private): CSS repository, updated 25 days ago.
 - ZCC** (Private): OCaml repository, 1 issue needs help, updated on Mar 29.
- Profile Summary:** Shows 7 followers and 3 following.
- Recent Activity:** Shows "Lakebolt Research" and "The Internet".
- Achievements:** Shows various achievement icons.

A large orange arrow points to the green "New" button located at the top right of the repository list area.

Primeros pasos: (c) crear un nuevo repositorio en GitHub

- ❖ Deben llenar la forma como se indica aquí
- ❖ De otra forma pueden perder puntos
- ❖ Nombren al repositorio **CI-2125-0**
- ❖ Pongan una descripción con sentido
- ❖ Muy importante: el repositorio debe ser **privado**
- ❖ Pidan añadir un README file
 - ❖ Sin embargo, no es necesario editarlo
- ❖ Selecciónen el molde (template) para el **.gitignore**
 - ❖ escojan C++ o C
- ❖ No necesitamos licencia: escojan “None”
- ❖ Denle al botón verde para **crear el repositorio**

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * Repository name *

Liondancे / CI-2125-0 ✓

Great repository names are short and memorable. Need inspiration? How about [fuzzy-goggles](#)?

Description (optional)

CI-2125 repositorio para la primera tarea

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.

Initialize this repository with:

Skip this step if you're importing an existing repository.

Add a README file
This is where you can write a long description for your project. [Learn more](#).

Add .gitignore

Choose which files not to track from a list of templates. [Learn more](#).

.gitignore template: C++ ▾

Choose a license

A license tells others what they can and can't do with your code. [Learn more](#).

License: None ▾

This will set `main` as the default branch. Change the default name in your [settings](#).

ⓘ You are creating a private repository in your personal account.

Create repository

Primeros pasos: (d) crear el directorio local para los proyectos

❖ Mejores prácticas (consejos)

- ❖ En su computadora es buena idea tener un directorio para proyectos: el mío se llama **Prj**
- ❖ Dentro de ese directorio sería bueno tener un directorio para los proyectos de la **USB**
- ❖ Y crear un directorio allí para cada curso de la **USB** que requiere programación
- ❖ Así que yo tengo un directorio con camino de esta forma: .../**Prj**/**USB**/**CI-2125**
 - ❖ La ubicación del directorio **Prj** en sí no es muy importante: pónganlo donde tenga sentido para Uds.
 - ❖ De hecho, en mi caso, la ubicación depende del ambiente (Linux o Windows) aunque *es el mismo directorio*
 - ❖ **Prj** es compartido por los dos sistemas de operación: el directorio de Windows está montado (*mounted*) en Linux
 - ❖ Esto es posible porque he habilitado el WSL (Windows Subsystem for Linux) en Windows 10
- ❖ La organización anterior se basa en mi experiencia. No tienen porque hacerla igual, pero ...
- ❖ ... yo me voy a referir al directorio **CI-2125** como el directorio del curso o directorio **CI-2125** ...
- ❖ ... independientemente de la ubicación y el nombre que le pongan. Así que no se confundan.

Primeros pasos: (e) clonar repositorios remotos

- ❖ Antes que nada, si ya clonaron mi repositorio publico CI-2125 **sugiero que lo borren**
 - ❖ Como verán en un momento no tiene nada que ver con **su directorio del curso**
- ❖ Usando GitHub (en la Web) localicen el siguiente repositorio (materiales del curso) publico:
 - ❖ <https://github.com/Liondance/CI-2125-M>
- ❖ Favor, referirse al diagrama en la lamine siguiente para entender las instrucciones
- ❖ Hagan click en el botón **verde** que dice “Code” con un triangulito apuntando hacia abajo
- ❖ Podrán apreciar (ver imagen en la lamine siguiente) tres etiquetas con los protocolos soportados
 - ❖ HTTPS
 - ❖ SSH
 - ❖ GitHub CLI
- ❖ Son los tres protocolos que permiten comunicar su computadora con GitHub

Primeros pasos: (e) clonar repositorios remotos

The screenshot shows a GitHub repository page for a user named 'Liondance'. The repository name is 'new period reset'. The page displays several files: '0.Basics' (new period reset), '.gitignore' (new period reset), and 'README.md' (oops: moved README to proper lo). Below the files, there is a large heading 'CI-2125' and a sub-section 'Carpetas con materiales para CI-2125'. At the top of the page, there are navigation links: Actions, Projects, Wiki, Security, Insights, Settings, main branch, 1 branch, 0 tags, Go to file, Add file, and a green 'Code' button. A context menu is open on the right side, triggered by the 'Code' button. The menu is titled 'Clone' and includes three options: HTTPS (selected, highlighted with a yellow arrow), SSH, and GitHub CLI. The HTTPS URL is <https://github.com/Liondance/CI-2125-M>. Below the URL, there is a note: 'Use Git or checkout with SVN using the web URL.' To the right of the URL is a copy icon. A purple arrow points from the left towards the copy icon. The menu also contains 'Open with GitHub Desktop', 'Open with Visual Studio', and 'Download ZIP'.

Primeros pasos: (e) clonar repositorios remotos

- ❖ Estas instrucciones se refieren al protocolo HTTPS, pero ... hay ventajas en usar el protocolo **SSH** aunque requiere más trabajo de configuración
 - ❖ Para configurar **SSH**, primero deben crear su **key-pair** (clave pública y privada) y seguir las instrucciones en GitHub
 - ❖ Aquí dice como hacerlo: <https://docs.github.com/en/authentication/connecting-to-github-with-ssh>
 - ❖ El resto es completamente análogo a usar **HTTPS**, como verán en lo que sigue ...
- ❖ En el pop-up, seleccionen el protocolo **HTTPS** (ver **flecha amarilla** en la imagen anterior) o **SSH** si lograron configurarlo
- ❖ Hagan click en el botón para copiar el texto (ver la **flecha violeta**) llamado “connection string”
- ❖ En su desktop, usando Git Bash dentro del directorio del curso, escriban “git clone ” sin darle a Enter
- ❖ Denle al botón derecho del mouse (right click) en la ventana del shell (como ya saben, el shell es Git Bash)
- ❖ Deberían ver (HTTPS) esto: git clone <https://github.com/Liondance/CI-2125-M.git>
- ❖ Alternativamente (**SSH**) esto: git clone <git@github.com:Liondance/CI-2125-M.git>
- ❖ Ahora si: Enter! Si todo va bien, el repositorio remoto (en GitHub) va a ser copiado dentro de su CI-2125
- ❖ Van a tener un repositorio CI-2125-M (materiales generales para el curso) dentro de CI-2125, su directorio del curso
- ❖ Su directorio del curso no es un repositorio de Git: contiene repositorios de Git, ligados a repositorios remotos en GitHub

Primeros pasos: (e) clonar repositorios remotos

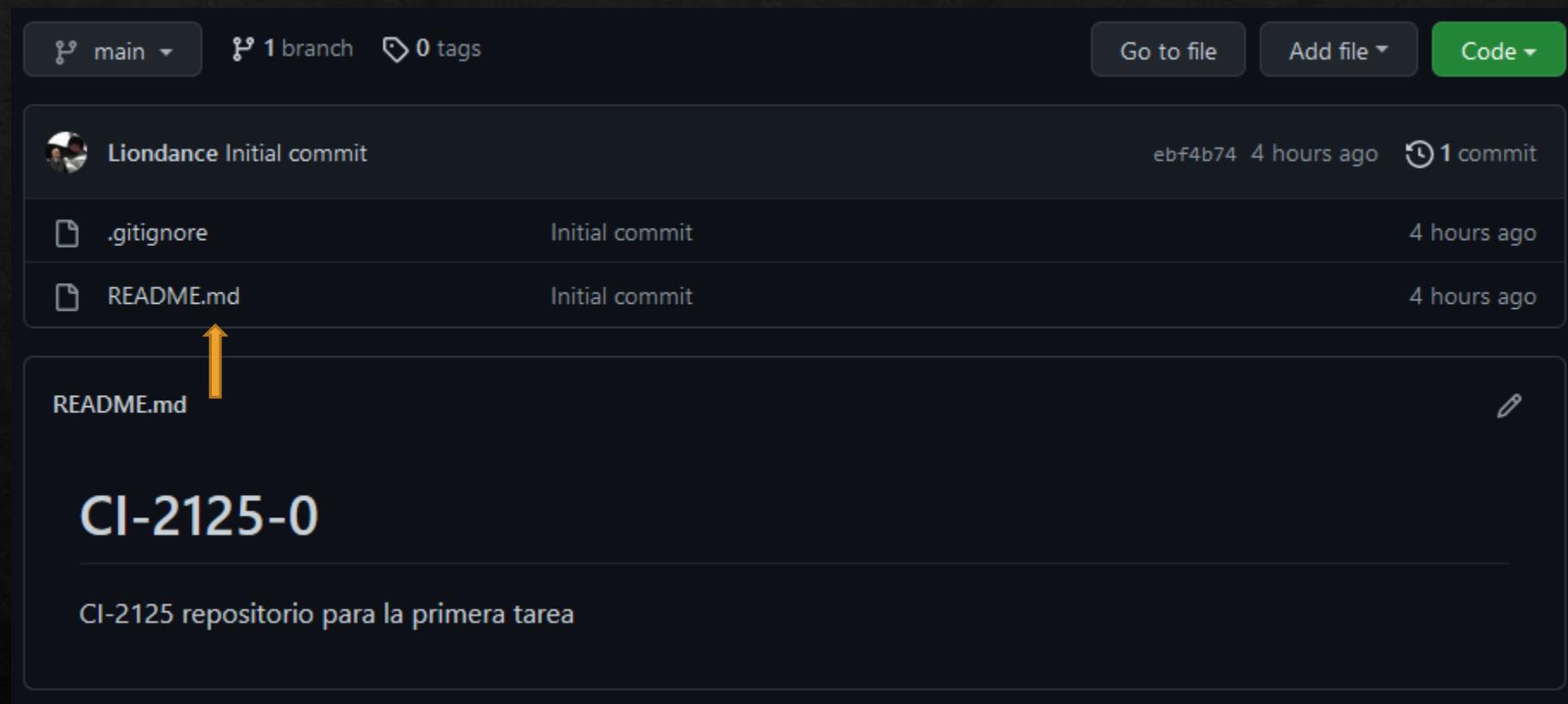
- ❖ Ahora van a clonar su repositorio (privado) remoto CI-2125-0 dentro de su directorio del curso
- ❖ Simplemente repitan los pasos anteriores, pero ... ¿como pueden ver un repositorio privado?
- ❖ ¡Duh: Lo pueden ver porque Uds. son los dueños! Claro está que deben dar credenciales.

- ❖ Si hicieron todo bien, deben tener dos directorios dentro de CI-2125 (su directorio del curso)
 - ❖ CI-2125-M: el repositorio con materiales del curso, controlado por el “Lab Chief”
 - ❖ CI-2125-0: el repositorio con su primera tarea

- ❖ **¡Felicitaciones!** Casi listos, pero todavía no hemos terminado ...
- ❖ **Ahora vamos a probar la instalación y aprender comandos esenciales de GitHub**
- ❖ **Por favor, vayan a la pagina siguiente ...**

Primeros pasos: (f) sincronizando repositorios

- ❖ Usando GitHub, vayan a la pagina de su repositorio CI-2125-0 y hagan click sobre README.md (flecha amarilla)



Primeros pasos: (f) sincronizando repositorios

- ❖ Ahora tienen que darle al lápiz (flecha amarilla) ...

main ▾ CI-2125-0 / README.md Go to file ...

Liondance Initial commit Latest commit ebf4b74 4 hours ago History

1 contributor

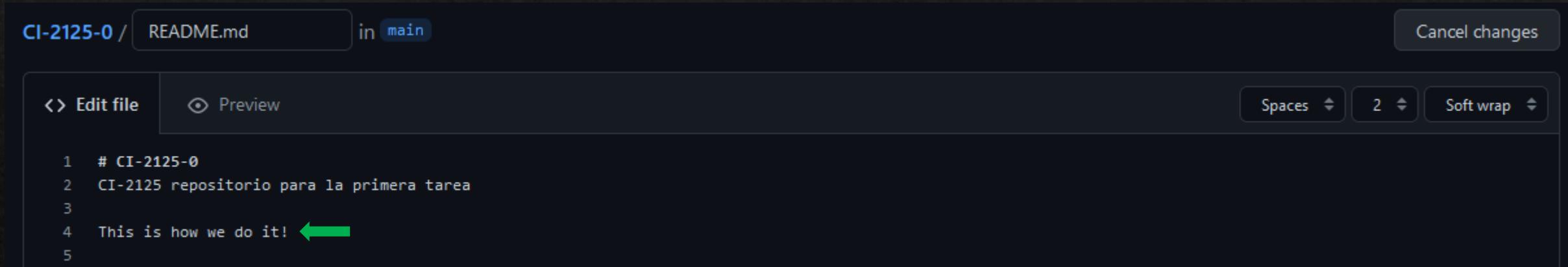
2 lines (2 sloc) | 54 Bytes <> Raw Blame 🖥 🖊 🗑

CI-2125-0

CI-2125 repositorio para la primera tarea

Primeros pasos: (f) sincronizando repositorios

- ❖ Por fin podemos editar. Agreguen una línea a la documentación, como yo lo hice aquí (flecha verde)

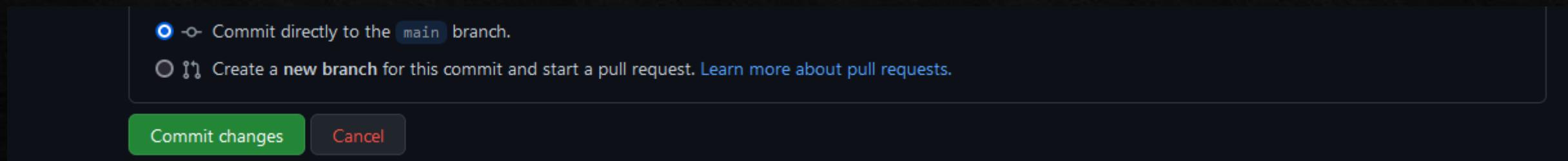


A screenshot of a GitHub code editor interface. At the top, it shows the repository path "CI-2125-0 / README.md" and the branch "in main". On the right, there are buttons for "Cancel changes", "Edit file", "Preview", "Spaces", "2", and "Soft wrap". The code editor area contains the following text:

```
1 # CI-2125-0
2 CI-2125 repositorio para la primera tarea
3
4 This is how we do it! ←
```

The line "This is how we do it!" is highlighted with a green arrow pointing to its end.

- ❖ Miren para abajo y hagan click en el botón verde (Commit changes) para hacer el cambio permanente.



Continua ...

Primeros pasos: (f) sincronizando repositorios

- ❖ Entren al directorio CI-2125-0 en su computadora y denle un vistazo a README.md
 - ❖ En el shell pueden usar este comando: `cat README.md`
 - ❖ ... o pueden usar su editor preferido
- ❖ Lógicamente, no deben esperar ver el cambio que hicieron en GitHub
 - ❖ El archivo en el repositorio remoto está en algún sitio en la nube, desconocido para nosotros
- ❖ Para traerse los cambios, deben ejecutar este comando de Git:
`git pull`
- ❖ Ahora pueden apreciar los cambios en README.md
- ❖ Pero, ¿Cómo podemos sincronizar el repositorio remoto cuando cambiamos el local? ...

Continua ...

Primeros pasos: (f) sincronizando repositorios

- ❖ Usando su editor, cambie el archivo como se muestra aquí (si quieren ver algo divertido) ... o como quieran

The screenshot shows a code editor window with a dark theme. The title bar says "i README.md 2, M X". The content of the file is as follows:

```
1 # CI-2125-0
2 CI-2125 repositorio para la primera tarea
3
4 This is how we do it: https://www.youtube.com/watch?v=0hiUuL5uTKc
5
```

- ❖ En el shell, ejecute estos comandos de Git:

```
git add .
```

```
git commit -m "probando ..."
```

```
git push origin main
```

- ❖ Significado:

Agregar archivos modificado en el directorio actual al conjunto de “commit”

Registrar los cambios (hacerlos permanentes) en el repositorio local de Git

Publicar los cambios locales al repositorio remoto

- ❖ Refresquen la página de GitHub: ahora deberían poder ver el video haciendo click en su enlace, en la página de GitHub

Aprendiendo a Usar Git en la Marcha

- ❖ Solo hemos visto 5 comandos de git; lo mínimo para poder algo útil:
 - ❖ `git clone <conexión>`
 - ❖ `git pull`
 - ❖ `git add .`
 - ❖ `git commit -m "<mensaje descriptivo>"`
 - ❖ `git push origin <rama correspondiente del repositorio remoto>`
- ❖ Referencia de Git – muy útil a medida que lo necesiten: <https://git-scm.com/docs/>
- ❖ Tutoriales de Git en la Web:
 - ❖ <https://www.educba.com/introduction-to-git/>
 - ❖ <https://www.hostinger.com/tutorials/basic-git-commands>
 - ❖ <https://www.geeksforgeeks.org/basic-git-commands-with-examples/>

¿Que mas necesitamos?

- ❖ Obviamente, para programar en C y C++ necesitamos
 - ❖ por lo menos un editor (por ejemplo: Visual Studio Code, Sublime, Atom)
 - ❖ por lo menos un compilador / enlazador de C (por ejemplo: gcc, g++, clang)
 - ❖ los IDE (Visual Studio 2017 y 2022) combinan lo anterior y más
 - ❖ pero también requieren más memoria y tienen su curva de aprendizaje
- ❖ En la presentación de las herramientas vieron cuales son bendecidas por el laboratorio y los enlaces para instalarlas
- ❖ En calidad de ingenieros y usuarios inteligentes, deberían ser capaces de completar los requisitos
 - ❖ Como mínimo, hagan **su mejor esfuerzo** instalando uno de los siguientes
 - ❖ <https://www.msys2.org/>
 - ❖ MinGW: <https://sourceforge.net/projects/mingw/>
 - ❖ Yo hice click en el botón **verde** (Download) y logre completar la instalación sin problemas
- ❖ En las siguientes láminas vamos a explicar como terminar de probar su instalación

Copiar los Materiales a su Carpeta de Trabajo y Entrega

- ❖ Primero, copiar los materiales en la carpeta **CI-2125/CI-2125-M/** a **CI-2125/CI-2125-0/**
- ❖ Como pueden ver, la carpeta **0.Basics** en los materiales contiene varias carpetas y un file **.gitignore**
 - ❖ La copia va a sobre-escribir el archivo **.gitignore** en **CI-2125-0**: está bien, es intencional
- ❖ Tenemos dos formas de copiarlos: Hardcore Unix Style y Windows Style
- ❖ 1. Hardcore Unix Style: ubicados en la carpeta **CI-2125**, ejecutar los comandos
cp -r CI-2125-M/* CI-2125-0/
- ❖ 2. Windows Style: usando el File Manager (ilustraciones a continuación)
 - ❖ Selectionen las 9 carpetas y el **.gitignore** ubicados en **CI-2125-M**
 - ❖ Hagan right-click y selectionen Copy (copiar) en el drop-down menu
 - ❖ Abran la carpeta **CI-2125-0**, hagan right-click dentro de ella y selectionen Paste

Compilando y corriendo archivos

- ❖ Estamos listos para probar la instalación. En la carpeta **CI-2125-0\3.args**, ejecuten los siguientes comandos

```
gcc args.cxx -o args  
.args primero segundo funciona!
```

- ❖ Deberían ver algo así usando Windows: en Linux tendrían / en vez de \ :

```
argv[0]: ...\\Prj\\USB\\CI-2125\\CI-2125-0\\3.args\\args.exe  
argv[1]: primero  
argv[2]: segundo  
argv[3]: funciona!
```

- ❖ Compilen nuevamente usando g++ y vuelvan a correr el programa

```
g++ args.cxx -o args  
.args primero segundo funciona!
```

- ❖ Denle un vistazo a todas las carpetas, en orden: les va a ayudar a aprender
- ❖ Además, **allí esta la tarea**: la vamos a explicar un poco el miércoles pero a buen entendedor ...
- ❖ Eso es todo, si han logrado compilar y correr el ejemplo en 3.args están listos para hacer la tarea
- ❖ Nos vemos el próximo viernes o antes, dependiendo de como están

Hasta la Próxima

<https://www.youtube.com/watch?v=0hiUuL5uTKc>