

Undefined Academy

Bootcamp de JavaScript Full-stack

Semana 1, Clase 2:

Git: Control de Versiones



Metodología de trabajo

Formato de las sesiones



(1h 30m)

- 10 min Revisión de ejercicio anterior
- 30 min Presentación del tema
- 40 min Ejercicios prácticos
- 10 min Preguntas y discusión

Herramientas del Bootcamp

- **Twitch:** <https://undf.sh/en-vivo>
- **Discord:** <https://undf.sh/discord>
- **Notion:** <https://undf.sh/base>

Anuncios Parroquiales

Anuncios Parroquiales

- Estamos trabajando en un sistema para no dejar a nadie sin respuesta en Discord.
- Tenemos nuevo canal de `#feedback` dónde hay instrucciones de cómo darnos feedback.
- Las preguntas del bootcamp en `#preguntas-bootcamp`
- No hay un hard-deadline para entregar los ejercicios.

[**https://undf.sh/empezar**](https://undf.sh/empezar)

Agenda

- ¿Qué es Git?
- Cómo crear un repositorio local y hacer tu primer commit
- Las ramas y el uso de merge
- Entendiendo cómo funciona git rebase
- Git y los repositorios remotos

**Revisión de
ejercicio anterior**

Ejercicio

Crear un Github Profile README¹

1. Este debe tener una pequeña descripción tuya tipo CV.
2. Añade una tabla en Markdown (Obviamente) con al menos 5 comandos de la Terminal que hayas aprendido, de forma que tenga nombre del comando y descripción en columnas separadas.
3. Añade una sección con al menos 2 comandos alias que te parezcan útiles incluyendo columna de descripción.
4. (Opcional) Clona tu repositorio y ejecuta `l1t_y` como en la práctica.

1. <https://docs.github.com/en/account-and-profile/setting-up-and-managing-your-github-profile/customizing-your-profile/managing-your-profile-readme>

Alias para listar el top 10 comandos más usados

```
alias mostused='history | awk '\''{print $2}'\''  
| sort | uniq -c | sort -nr | head -n 10'
```


- ¿Qué es Git?
- Cómo crear un repositorio local y hacer tu primer commit
- Las ramas y el uso de merge en git
- Entendiendo cómo funciona git rebase
- Git y los repositorios remotos

Git

es un sistema de control de versiones distribuido, gratuito y Open Source¹.

1. Código abierto

!! *Qué sea **Open Source** no significa que sea gratis.*

 *¿Cuál es la diferencia entre un sistema de control de versiones **distribuido** y uno **centralizado**?*

Control de versiones

**Imagina que es 1998 y tú y un
amigo estuvieran trabajando
en un libro mediante
Microsoft Word.**

**¿Cómo podrían colaborar los
dos de manera
independiente?**

**¿Sí hacemos correcciones,
cómo sabemos cuál es la
versión corregida final?**

tesis_final_version_2_definitiva_marzo(1).docx

Los sistemas de control de versiones, solucionan este tipo de problema.

Storytelling videos with Emma

docs.google.com/document/d/1HK0ym3edH0A-hxZnJeRhUCFatrku3MFO_2sy8YO-LE/edit#heading=h.5utn733jfpvr

Version history

All versions

NOVEMBER 2022

November 4, 2022, 11:54 AM

Current version

Guillermo Rodas

JUNE 2021

▶ June 10, 2021, 9:03 PM

Emma Carvajal

Guillermo Rodas

June 4, 2021, 3:09 AM

Guillermo Rodas

▶ June 3, 2021, 9:40 PM

Emma Carvajal

Guillermo Rodas

▶ June 1, 2021, 10:09 PM

Guillermo Rodas

Emma Carvajal

▶ June 1, 2021, 9:19 PM

Emma Carvajal

Guillermo Rodas

MAY 2021

← November 4, 2022, 11:54 AM

100%

Total: 3 edits

General script

- Objective: **Inspire others with my story and first hand knowledge**
 - Transmit emotions: inspire, transparency,
 - Impress visually
 - Easy to digest
 - Relatable
- Message: **If you want to accomplish your professional goals, do things right. No career has short cuts.**
 - No shortcuts (do things right). Transparency.
 - Who I am, what I've done, Why am I addressing you?
 - After basic needs are met, It's not what you have, it's what you do with it. -
- Media:
 - Youtube
 - Twitch
- Target audience:
 - Expand current audience.
- Script:
 - 1. Where I came from

Pre-work

notion.so/undefinedshell/Pre-work-c5030a50ce2e46959a989d995d50de96

Share

Updates

Analytics

All comments

Guillermo Rodas edited

Pre-work

Feb 26 · Undefined Academy / Knowledgebase

Aa Title

Guía de Pre-work

¿Necesitas una guía para ayuda para entender esta guía?

Contenidos

Guillermo Rodas edited

Pre-work

Feb 25 · Undefined Academy / Knowledgebase

Video

https://youtu.be/mfmRR1NRkd4

Clic en la flecha — Puedes ver un video explicativo si quieres.

¿Necesitas una guía para la guía?

Guillermo Rodas edited

Pre-work

Feb 25 · Undefined Academy / Knowledgebase


Callout

Para el Bootcamp puedes usar cualquier

Pre-work

Knowledgebase

Pre-work



Pre-work

1 backlink

El Pre-work es el trabajo inicial de configuración de herramientas que necesitas para poder empezar a programar y tomar el Bootcamp.

Contenidos

¿Necesitas ayuda para entender esta guía?

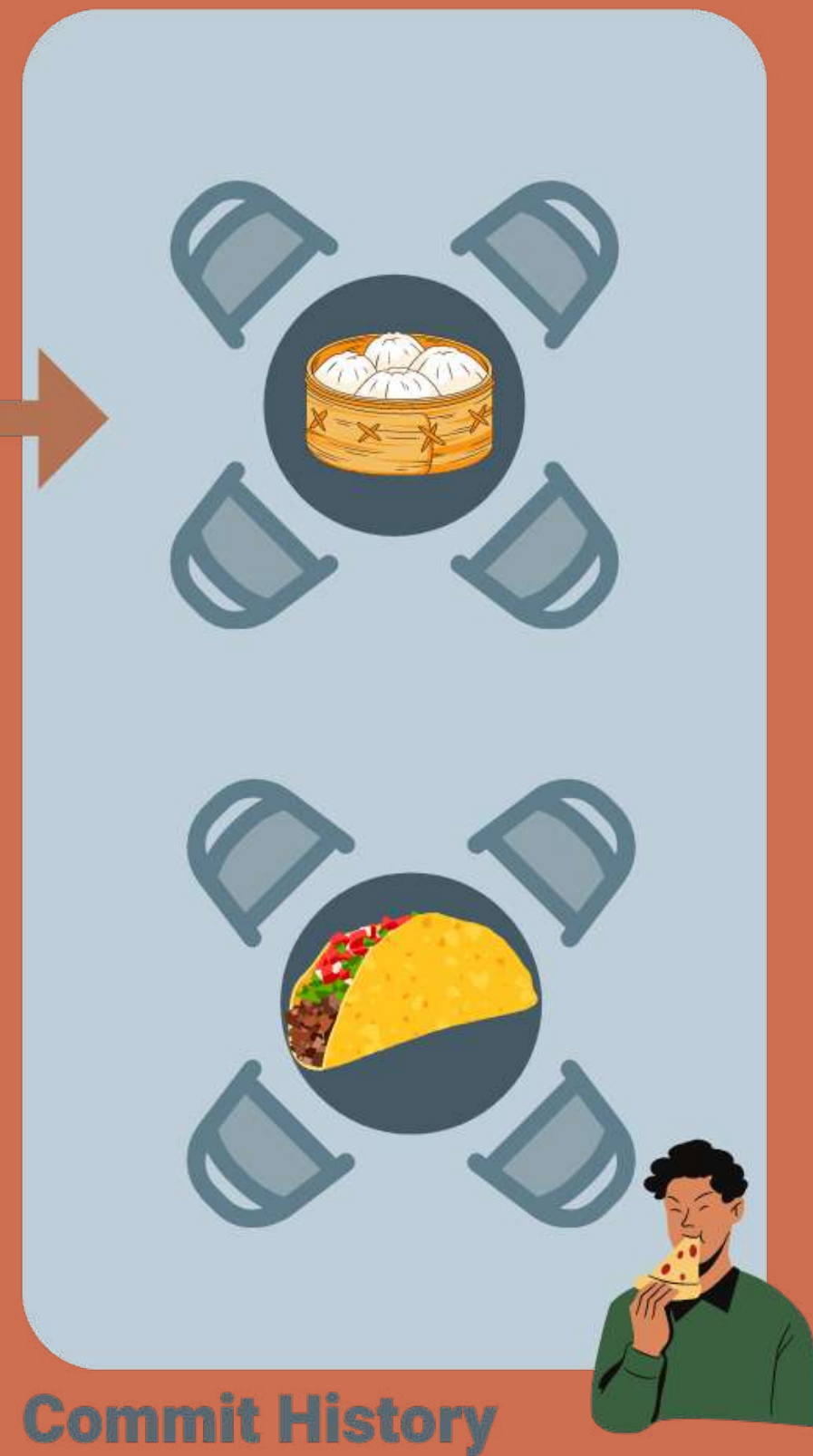
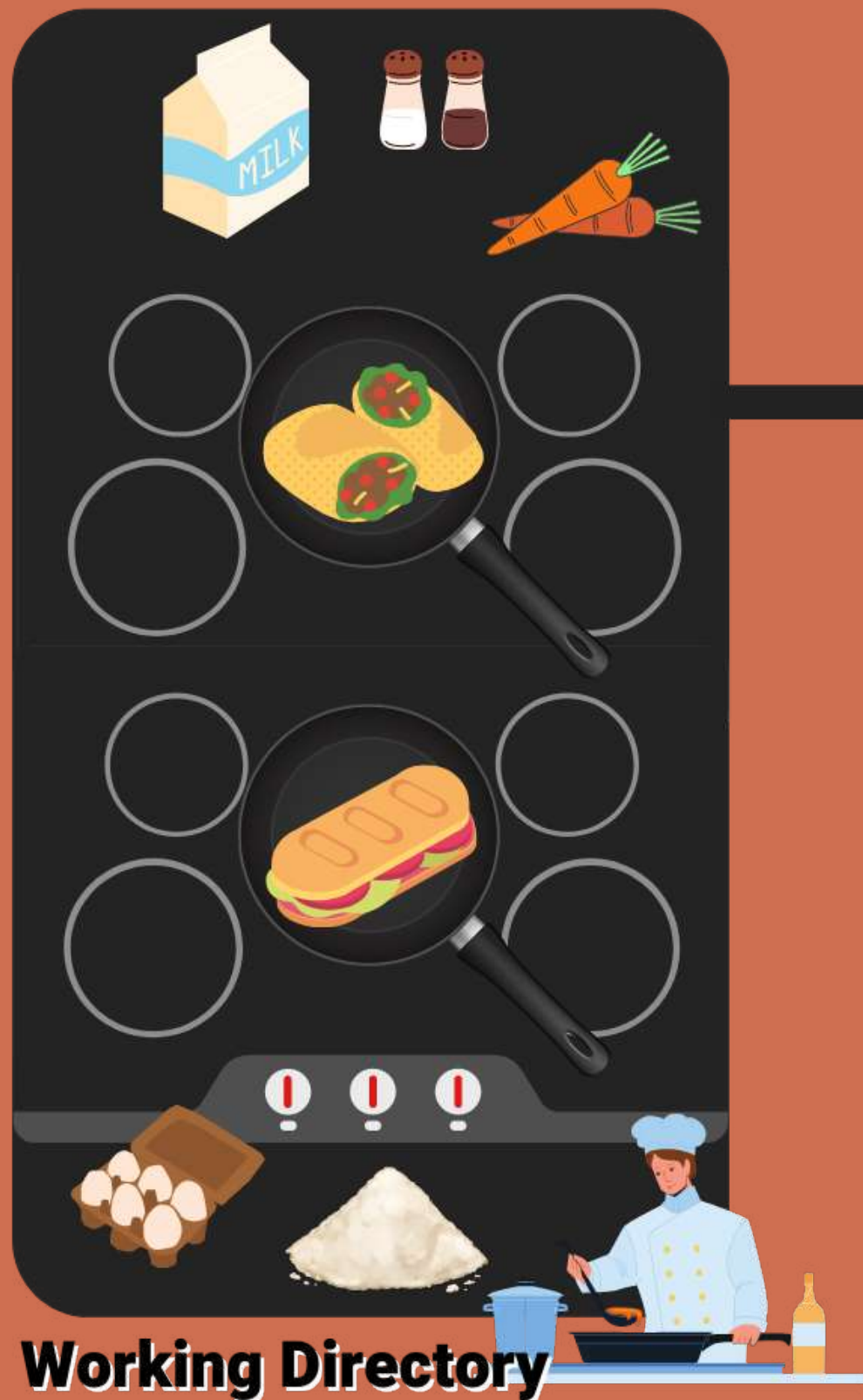
Pasos a seguir

Guía de emojis

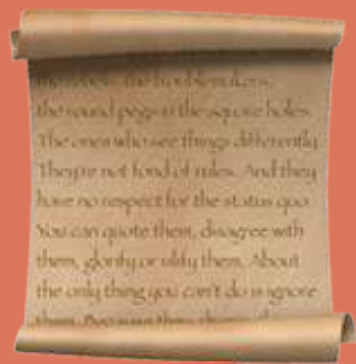
**Aunque Git no es
exactamente así.**

Git Staging Area

La historia que no te han contado.



Teórico-practico



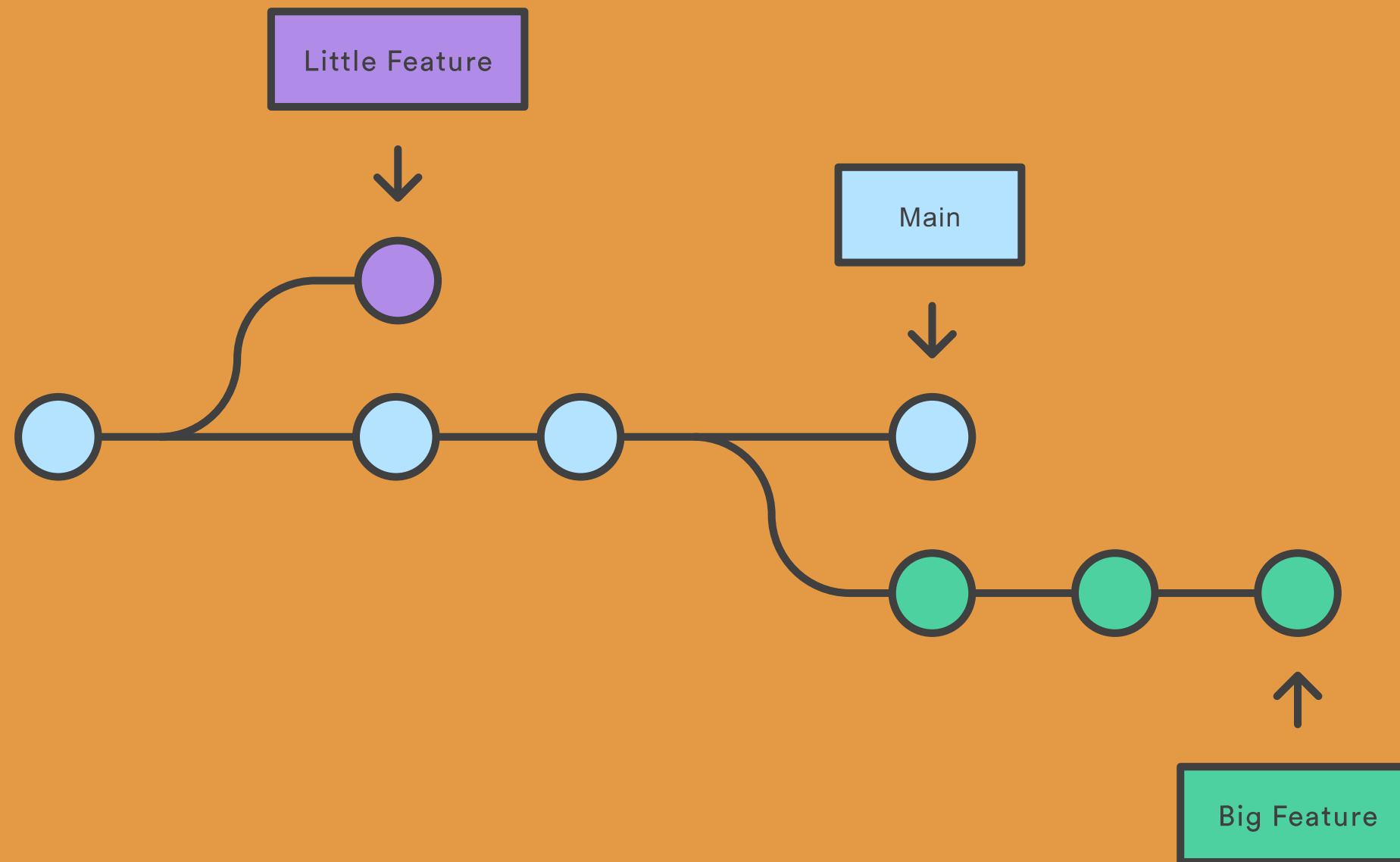
**Cómo crear un
repositorio local y
hacer tu primer
commit**

Cómo crear un repositorio local y hacer tu primer commit

1. Inicializamos el control de versiones con `git init`
2. Chequeamos el estado con `git status`
3. Agregamos al Staging area con `git add <file>`
4. Hacemos commit con `git commit -m "<mensaje>"`
5. Revisamos el log de commits con `git log`

Las ramas y el uso de merge

Branches: ¿Cómo funciona?¹



1. <https://www.atlassian.com/git/tutorials/using-branches>

Las ramas y el uso de merge

1. `git branch <nombre-de-la-rama>` crea una nueva rama.
2. `git branch` lista las ramas.
3. `git checkout <nombre-de-la-rama>` cambiar de rama.
4. `git checkout -b <nombre-de-la-nueva-rama>` cambiar y crear una nueva rama.

Sintaxis moderna

1. `git switch <nombre-de-la-rama>` cambiar de rama
2. `git switch -c <nombre-de-la-nueva-rama>` cambiar y crear una nueva rama.

Merge: ¿Cómo funciona?¹



1. <https://www.atlassian.com/git/tutorials/using-branches/git-merge>

Las ramas y el uso de merge

1. `git switch main`
2. `git merge <nombre-de-la-rama>`

Entendiendo cómo funciona git rebase

Rebase: ¿Cómo funciona?¹



1. <https://www.atlassian.com/git/tutorials/merging-vs-rebasing>

Rebase: ¿Cómo funciona?¹



1. <https://www.atlassian.com/git/tutorials/merging-vs-rebasing>

Rebase: ¿Cómo funciona?¹



1. <https://www.atlassian.com/git/tutorials/merging-vs-rebasing>

Regla de oro

Nunca usar rebase en ramas publicas.



Git y los repositorios remotos

Git y los repositorios remotos

1. `git remote add origin <url>`

agregamos un nuevo repositorio remoto.

2. `git push -u origin main`

3. `git pull origin main`

Recursos

- <https://www.atlassian.com/git/tutorials/learn-git-with-bitbucket-cloud>
- <https://www.atlassian.com/git/tutorials/merging-vs-rebasing>
- <https://www.atlassian.com/git/tutorials/advanced-overview>
- <https://www.git-tower.com/learn/git/ebook>
- <https://git-scm.com/docs>
- <https://learngitbranching.js.org/>
- <https://leanpub.com/aprendiendo-git>

Ejercicio

1. Hacer un fork del repositorio `https://github.com/undefined-academy/semana-1`
2. Clonar tu repositorio "forked" en tu carpeta de **HOME**.
3. Agregar allí un archivo markdown de tu perfil de GitHub, dentro de la carpeta `github-profiles` y el nombre de archivo debe ser tu usuario de Discord incluyendo el tag numérico¹.
4. Incluye dentro del archivo un link que nos lleve directamente a tu perfil de GitHub.
5. Haz commit de dicho archivo.
6. Ve al repositorio del paso 1 y crea tu primer Pull Request.
7. No dejes la descripción del Pull Request vacía, escribe cómo fue el proceso de resolver este ejercicio, usar Markdown.

1. No puedes poner un # como nombre de archivo, cámbialo por un guion (-).

¿Preguntas?