



Git, GitHub y GitHub pages





**Activen las cámaras los que puedan y
pasemos asistencia**

***Desarrollar
colaborativamente un
proyecto con GIT, aplicando
conceptos abordados a lo
largo del módulo.***

- Unidad 1: Introducción a HTML.
- Unidad 2: Introducción a CSS
- Unidad 3: Alineando contenido con flex
- Unidad 4: Bootstrap
- Unidad 5: Terminal, Git y Github
- Unidad 6: Trabajo colaborativo y Github Pages



Te encuentras aquí





Inicio

{desafío}
latam_



/ Subir una página web a Github Pages. */*

/ Resolver conflictos en GIT. */*

A vertical line separates the white left side from the orange right side. It features a series of white symbols: a closing curly brace '}', an '@' symbol, an opening curly brace '{', and a stylized '@' symbol, all arranged vertically.

Objetivo



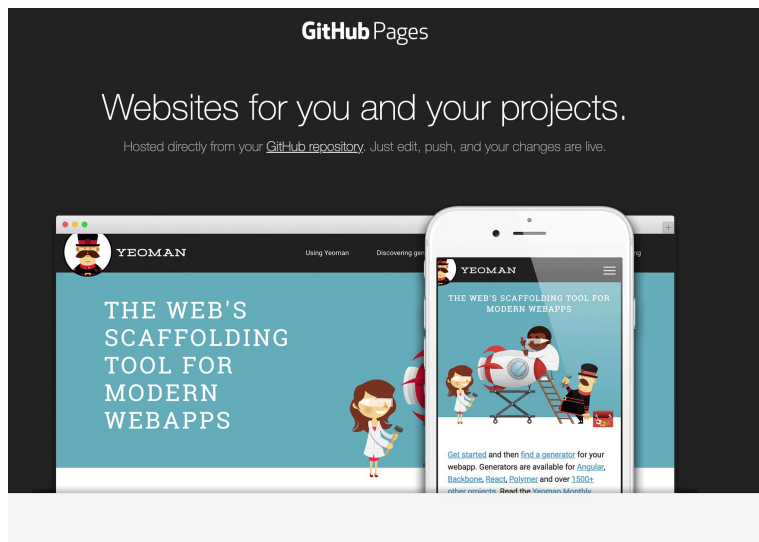
Desarrollo

{desafío}
latam_



GitHub Pages

¿Qué es GitHub pages?



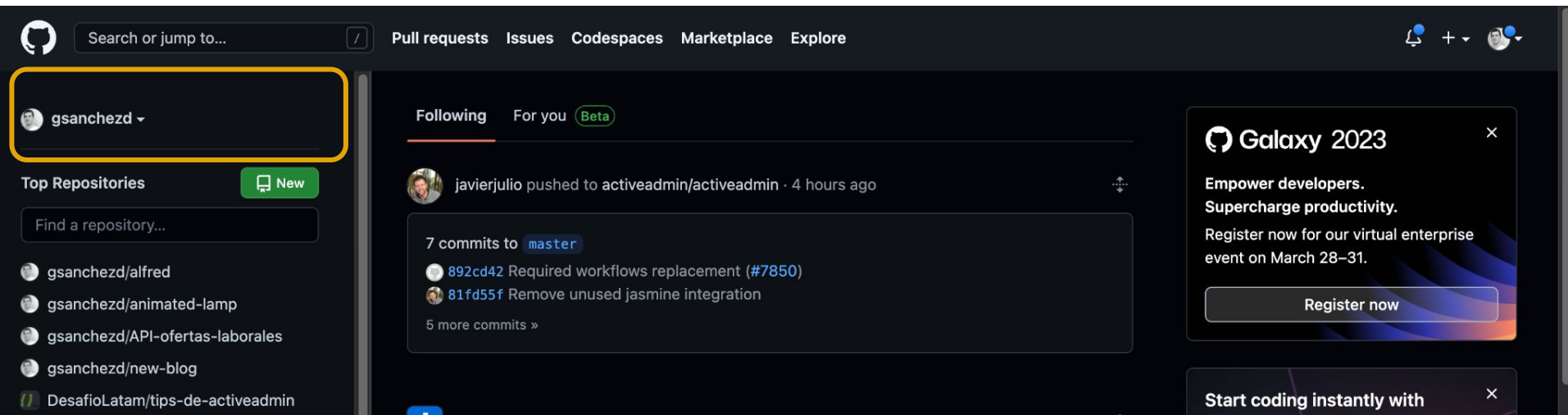
GitHub Pages nos permite levantar una página web gratuita a partir de un repositorio en Github.

En Github pages podemos tener:

- Un sitio web para nuestro usuario en Github, aquí podemos dejar nuestro CV y portafolio.
- Un sitio web para cada uno de nuestros proyectos, de esta forma podemos crear páginas web para presentarlos.

Github Pages

¿Recuerdas tu nombre de usuario?



Aparece en la página inicial al
entrar con tu cuenta.

Github Pages

Creando un repositorio para nuestro usuario

Para crear una cuenta asociada a tu usuario, el repositorio creado tiene que tener una URL con la siguiente estructura **username.github.io**, en caso contrario no va a funcionar.

- Nuestro primer paso será en Github crear nuestro repositorio.
- Si tu usuario es camila, la url debe ser camila.github.io
- Si ya tienes uno generado, no puedes crear otro con el mismo nombre, pero es posible hacerlo si borraras un repositorio y creas uno nuevo.

Github Pages

Recordando el concepto de remote

Un proyecto git en tu computador puede tener una o más referencias a un repositorio remoto, como en el caso de github, esto es el **remote**.

Cuando uno agrega un **remote** a un proyecto debe ponerle un nombre, usualmente, por convención, se le pone el nombre **origin**, pero podrías poner un nombre distinto o incluso configurar más de un **remote**.

Github Pages

Creando un repositorio para nuestro usuario

Los pasos a seguir dependen de cuál sea nuestra situación.

- ¿Es un repositorio nuevo? (no hemos hecho git init en nuestro proyecto).
 - Creamos el repositorio en github y seguimos los pasos de instalación.
- ¿Es un repositorio antiguo, pero no tiene un **remote** configurado? (el remote es la referencia a un repositorio remoto)
 - Agregamos el remoto.
- ¿Es un repositorio antiguo y tiene un **remote** configurado?
 - Opción 1: Agregar un nuevo remote con un nombre distinto.
 - Opción 2: Eliminar el remote anterior y crear uno nuevo referenciando al nuevo proyecto.

Github Pages

Pasos con un repositorio antiguo y remote configurado

Para esta dinámica subiremos el CV que hicimos en la unidad anterior. Como ese proyecto ya lo hemos subido a un repositorio de github tendremos que configurar el remote.



Git remote

Revisando los remotes

- El remote de git nos indica a qué repositorio apunta.
- Un proyecto incluso podría apuntar a múltiples repositorios distintos.
- Podemos conocer los puntos remotos de un proyecto en git con `git remote -v`

```
origin  git@github.com:gsanchezd/fdsw-github.git (fetch)  
origin  git@github.com:gsanchezd/fdsw-github.git (push)
```

Git remote

Agregando un remote

Podemos agregar un nuevo remote con:

- `git remote add alias direccion_al_remoto`

Por ejemplo:

- `git remote add deploy git@github.com:usuario/usuario.github.io.git`

Luego, podemos enviar los cambios con:

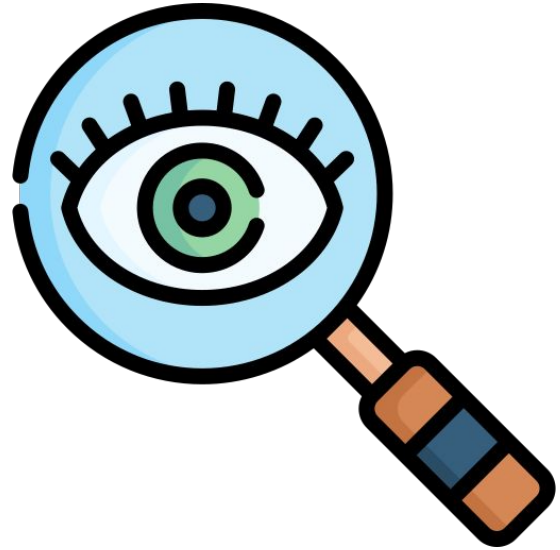
- `git push deploy main`




Revisando nuestra página

Entrar a tuusuario.github.io

Si hicimos todos los pasos anteriores correctamente deberíamos poder entrar a tuusuario.github.io y ver nuestra página web.



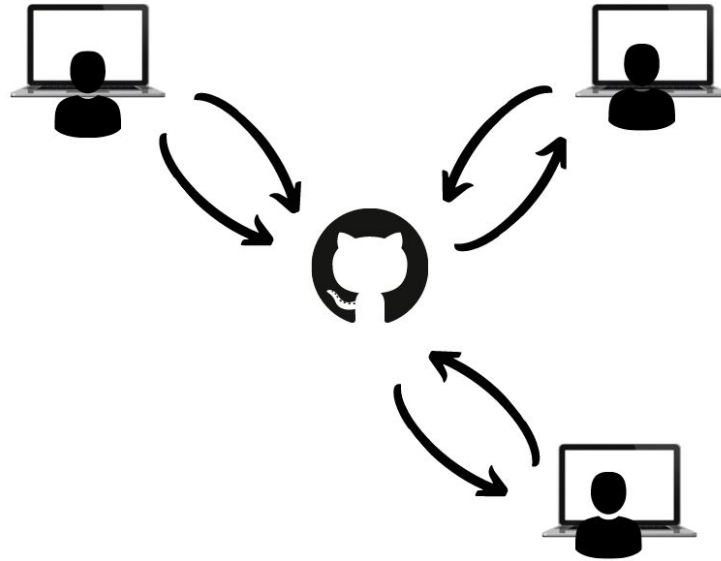
`/* Subir una página web a Github Pages. */` 

`/* Resolver conflictos en GIT. */`

Objetivo

Trabajo colaborativo con GIT

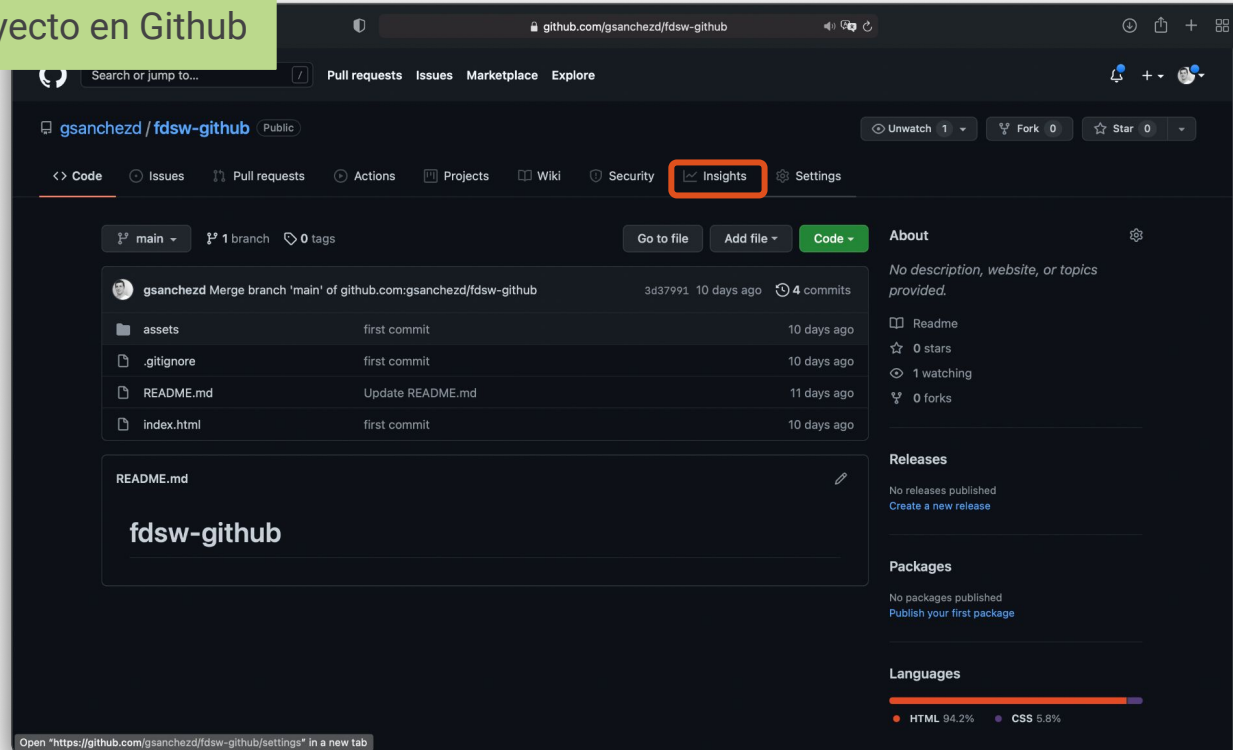
Múltiples personas en un repositorio



Trabajo colaborativo con GIT

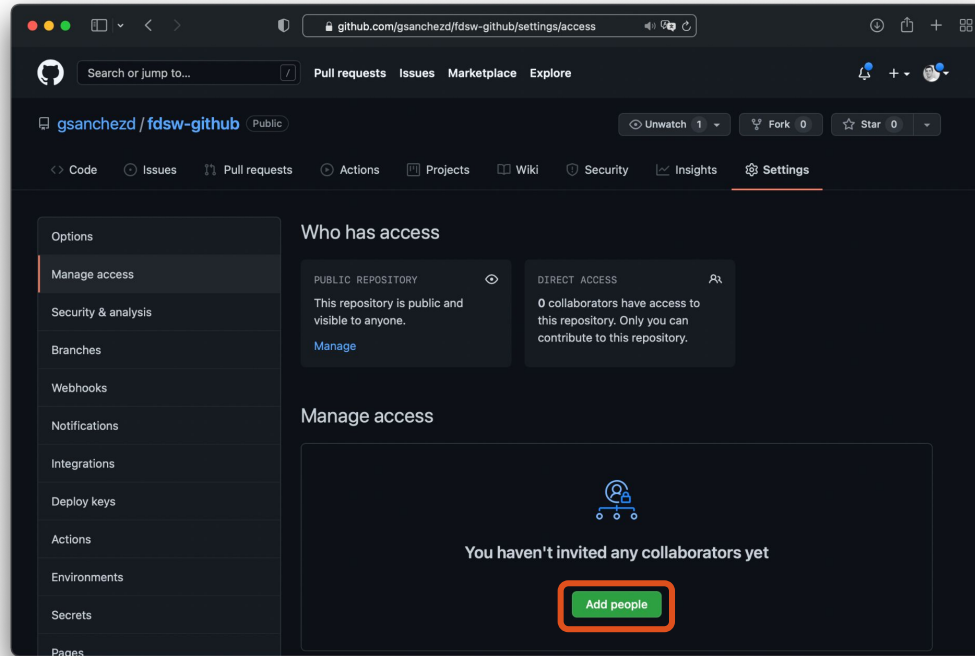
Tenemos que agregar a la persona a github

Dentro del proyecto en Github



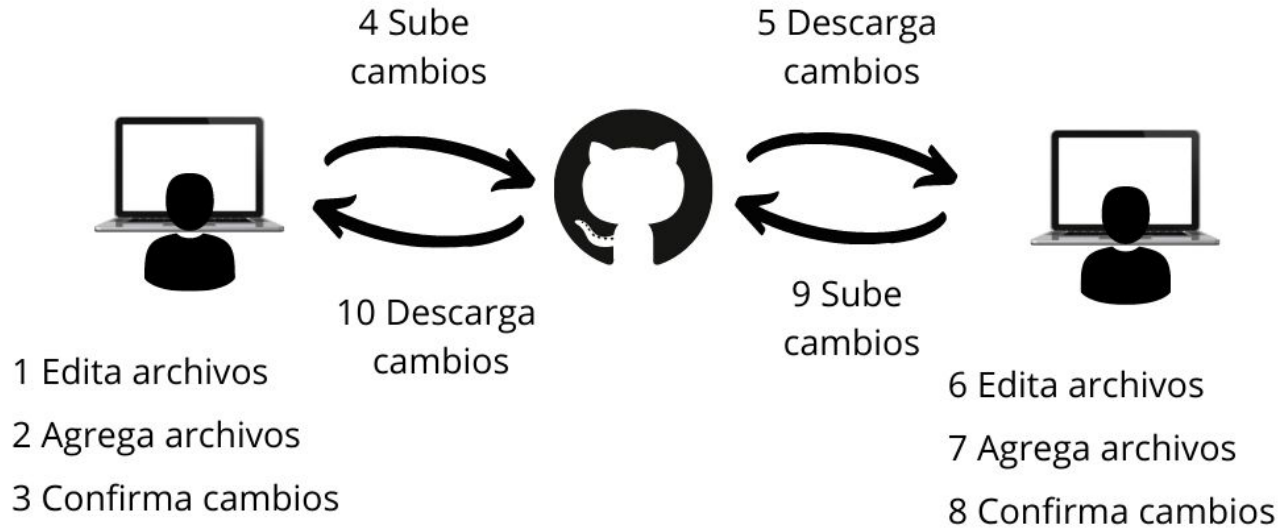
Trabajo colaborativo con GIT

Vamos a manejo de accesos > Agregar persona



Trabajo colaborativo con GIT

Flujo de trabajo



Flujo de trabajo

Subiendo cambios

Para subir los cambios al repositorio remoto, debemos escribir:

```
git push origin main
```

En caso de que hayas creado una rama paralela a la generada por defecto por git, utiliza el comando:

git push origin nombre_rama.

Flujo de trabajo

Bajando cambios

Para bajar los cambios al repositorio remoto, debemos escribir:

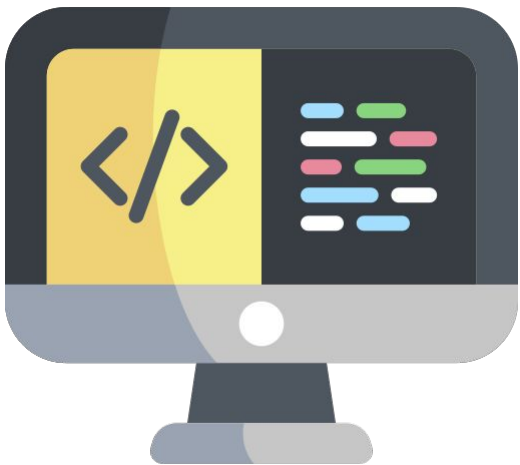
```
git pull origin main
```

En caso de que hayas creado una rama paralela a la generada por defecto por git, utiliza el comando:

git pull origin nombre_rama.

Trabajo colaborativo con GIT

¿Qué es un conflicto?



Un conflicto se produce cuando se está trabajando colaborativamente en GIT y dos personas modifican las mismas líneas.

Trabajo colaborativo con GIT

Manejo de conflictos

El siguiente escenario puede llevar a tener que resolver un conflicto:

1. Bob edita la línea 3 del archivo index.html
2. Bob confirma el cambio, pero no hace push
3. Alice edita la línea 3 del archivo index.html
4. Alice confirma el cambio y hace pull
5. Bob hace pull para descargar los cambios de Alice y ... Conflicto!! Puesto que ambos modificaron el mismo archivo y ahora, Bob debe determinar cuáles cambios serán los finales, los suyos o los de Alice.

Trabajo colaborativo con GIT

Los conflictos son buenos, simplemente hay que determinar cuál manda de dos cambios

Resolver el conflicto requiere de tomar una de las siguientes decisiones:

1. Bob deja su cambio deshaciendo el de Alice.
2. Bob deja los cambios de Alice deshaciendo sus cambios.
3. Bob modifica el archivo manualmente intentando dejar ambos cambios o una mezcla de ellos.

Una vez elegido cuál será el cambio resultante, escoge la opción, luego deberás añadir los archivos utilizando git add, para realizar un commit indicando los cambios y finalmente, hacer un push.

Trabajo colaborativo con GIT

Resumen de comandos más utilizados

Comandos básicos de GIT

git init: Crea un nuevo repositorio de Git en un directorio local.

<Se ejecuta solo una vez en un proyecto nuevo>

Comandos de inicio

git log: Muestra un registro de commits en orden cronológico inverso.

Comandos de registro

Comandos de configuración

git config: Configura opciones de git como el nombre del usuario y el correo electrónico.

git add: Agrega cambios al índice para ser incluidos en el próximo commit.

git commit: Crea un nuevo commit con los cambios en el índice.

git status: Muestra el estado actual del repositorio, incluyendo archivos modificados y sin seguimiento.

Comandos de trabajo con archivos

git push: Envía los cambios locales al repositorio remoto.

git pull: Descarga los cambios del repositorio remoto y los fusiona con los cambios locales.

Comandos de colaboración remota

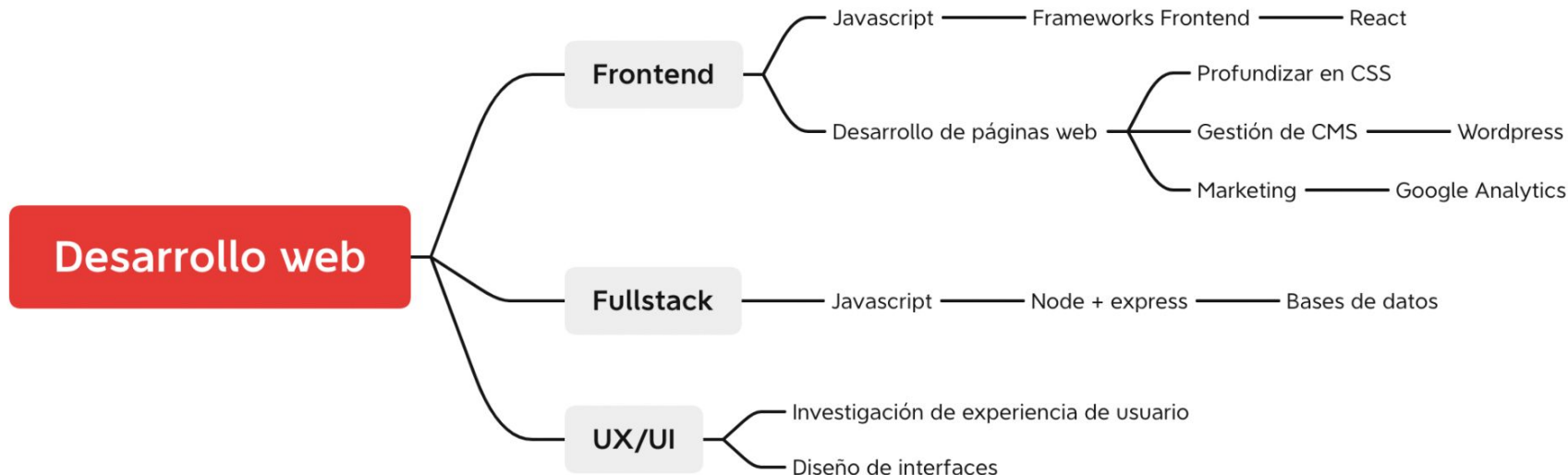
*/** Subir una página web a Github Pages. **/* ✓

*/** Resolver conflictos en GIT. **/* ✓

Objetivo

Más allá del desarrollo web

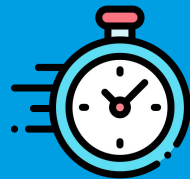
¿Qué debo seguir aprendiendo?





Cierre

{desafío}
latam_



¿Existe algún concepto que no
hayas comprendido?

Reflexionemos

- Revisar en conjunto el desafío.

¿Qué sigue?



*Academia de
talentos digitales*

www.desafiolatam.com



/DesafioLatam



/DesafioLatam



/DesafioLatam



/DesafioLatam