



**CICLO: DAW**  
**MÓDULO DE DESARROLLO WEB**  
**ENTORNO SERVIDOR**

# **GITHUB**

**Alumno:**  
**Edward-Ionut, Bunoaica**  
**Y1963355C**

*Los documentos, elementos gráficos, vídeos, transparencias y otros recursos didácticos incluidos en este contenido pueden contener imprecisiones técnicas o errores tipográficos. Periódicamente se realizan cambios en el contenido. Fomento Ocupacional FOC SL puede realizar en cualquier momento, sin previo aviso, mejoras y/o cambios en el contenido.*

*Es responsabilidad del usuario el cumplimiento de todas las leyes de derechos de autor aplicables. Ningún elemento de este contenido (documentos, elementos gráficos, vídeos, transparencias y otros recursos didácticos asociados), ni parte de este contenido puede ser reproducida, almacenada o introducida en un sistema de recuperación, ni transmitida de ninguna forma ni por ningún medio (ya sea electrónico, mecánico, por fotocopia, grabación o de otra manera), ni con ningún propósito, sin la previa autorización por escrito de Fomento Ocupacional FOC SL.*

*Este contenido está protegido por la ley de propiedad intelectual e industrial. Pertenecen a Fomento Ocupacional FOC SL los derechos de autor y los demás derechos de propiedad intelectual e industrial sobre este contenido.*

*Sin perjuicio de los casos en que la ley aplicable prohíbe la exclusión de la responsabilidad por daños, Fomento Ocupacional FOC SL no se responsabiliza en ningún caso de daños indirectos, sean cuales fueren su naturaleza u origen, que se deriven o de otro modo estén relacionados con el uso de este contenido.*

*© 2022 Fomento Ocupacional FOC SL todos los derechos reservados.*

## Contenido

1. Documentos que se adjuntan a este informe. .... ¡Error! Marcador no definido.
2. Resto de epígrafes que componen el desarrollo de este informe. ....2

(Una vez realizado el informe, no olvidar actualizar esta tabla del índice **(F9 + Actualizar toda la tabla)**, con el fin de que se actualicen todos los epígrafes y números de página)

## 1. ¿Qué es Git?

- Explica qué es Git y cuál es su propósito
  - Git es un sistema de control de versiones, esto se refiere a que es un sistema en el cual se pueden subir las diferentes versiones de un programa que cuenta con muchos archivos de código fuente.
- ¿Por qué es importante en el desarrollo de software?
  - Git es súper importante debido a que nos protege nuestro código fuente que trabajamos o gestionamos, nos almacena las diferentes versiones de nuestros programas y cambios realizados.
- Diferencia entre Git y otros sistemas de control de versiones
  - La principal diferencia entre Git y otros sistemas es que es muy eficaz, sirve como multiplicador de todas nuestras actividades y garantiza que todas las partes hagan su trabajo ordenadamente, a parte a los programadores juniors les brinda un entorno seguro en donde programar y subir sus proyectos.

## 2. ¿Qué es GitHub?

- Describe qué es GitHub y para qué se usa
  - GitHub es un servicio basado en la nube de control de versiones, es como Git pero pudiendo subir los programas y proyectos a la nube para compartirlos con la comunidad y seguir haciendo cambios mejorando los mismos y haciendo un seguimiento de los proyectos.
- ¿En qué se diferencia de Git?
  - La principal y mayor diferencia entre Git y GitHub es que uno es un programa de control de versiones ejecutado en local, mientras que GitHub es un programa de gestión de versiones alojado en la nube.
- Menciona al menos dos alternativas a GitHub
  - Dos alternativas muy buenas a GitHub son Amazon S3, creado por AWS y Google Docs, creado por Google.

## 3. Creación de una cuenta in GitHub

- Explica los pasos para registrarse en GitHub
  - Entrar a la página web de GitHub
  - Entrar en el apartado de "Registrarse"
  - Seguir los pasos indicados en la pagina(email, nombre completo, edad, uso de GitHub, etc).
- ¿Qué configuraciones básicas se pueden realizar en el perfil?
  - Principalmente podemos hacer 3 cosas básicas en nuestro perfil, Editar nuestra foto, nuestra biografía y crear un archivo readme.md que contiene información sobre nosotros.

## 4. Creación de un repositorio en GitHub

- ¿Qué es un repositorio y para qué sirve?
  - Un repositorio es un tipo de almacenamiento centralizado que usan los desarrolladores para realizar y administrar cambios en el código fuente de una aplicación.
- Explica cómo se crea un repositorio nuevo en GitHub
  - En la esquina superior derecha de cualquier página de GitHub tendremos un apartado de “Crear Repositorio”. Escribimos un nombre, opcionalmente, una descripción y la visibilidad del mismo. Con eso ya estaría creado.
- Diferencia entre repositorios públicos y privados
  - Los repositorios públicos pueden ser vistos y descargados por cualquier persona, mientras que los privados solo tenemos acceso nosotros.
- ¿Cuáles son los archivos esenciales en un repositorio y cuál es su función?
  - El archivo esencial que debe tener un repositorio es tener un archivo readme.md. La función de este archivo es la de mantener todos los cambios y la información del repositorio.

## 5. Instalación y configuración de Git en el equipo

- ¿Cómo se instala Git en los diferentes sistemas operativos?
  - En el caso de Windows tenemos un archivo ejecutable que nos lleva la descarga paso a paso, luego en MacOS y Linux tenemos que hacerlo mediante la consola de comandos, pero normalmente en MacOS suele venir instalado ya.
- ¿Qué comandos se utilizan para configurar Git con el nombre y el correo del usuario?
  - Se utiliza el comando `git config --global user y nuestro email`.
  - Para el usuario es `git config user.name` y el nombre
- ¿Cómo se verifica que Git está correctamente instalado?
  - Poniendo en nuestra consola de comandos `git --version` y si git nos devuelve su versión significa que está bien instalado.

## 6. Vinculación de Git con el equipo

- ¿Qué métodos existen para autenticar Git con GitHub?
  - Tenemos dos maneras, mediante la página de GitHub o mediante la consola de Git con la contraseña cuando nos la pida
- Explica cómo clonar un repositorio desde GitHub a tu equipo.
  - Utilizando la terminal, copiamos la url del repositorio a clonar y lo clonamos con el comando `git clone` donde queramos
- ¿Cómo se enlaza un repositorio local a GitHub?
  - Se utiliza el comando `git remote` y la url del repositorio que queramos clonar.

## 7. Comandos básicos de Git

- ¿Cómo se agregan archivos al área de preparación de Git?
  - Se agregan con el comando git add y el nombre del archivo o si no colocamos un "." para meter todos los ficheros nuevos.
- ¿Cómo se realiza un commit y qué significa el proceso?
  - Realizar un commit significa que los cambios o ficheros en el área de preparación son los que se van a confirmar para después ser subidos a la nube o ser guardados hasta nuevo cambio.
- ¿Cómo se envían los cambios de un repositorio local a GitHub?
  - Se envían los cambios con git push, esto envía todos los cambios.
- ¿Cómo se descargan actualizaciones del repositorio remoto?
  - Para descargar las actualizaciones se utiliza el comando git pull
- ¿Cómo se puede ver el historial de commits?
  - El historial de commits se ve con el comando git log

## 8. Trabajo en equipo en GitHub

- ¿Qué es un fork y para qué se usa?
  - Un fork es una copia de un repositorio donde se puede escribir y hacer públicos tus propios cambios o trabajar sobre ese repositorio como base.
- Explica qué es un pull request y cómo se hace
  - Hacer un pull request es pedir a otro desarrollador que trabaja en el mismo repositorio que haga un incorpore su rama de proyecto a la tuya.
- ¿Qué son las ramas en Git y cómo se crean?
  - Las ramas en git son una serie de líneas que pueden seguir nuestros proyectos con diferentes versiones o commits realizados.
  - Las ramas se crean con git branch y el nombre de la rama
- ¿Cómo se pueden fusionar ramas en un repositorio?
  - Las ramas se pueden fusionar con git merge, este comando fusiona dos ramas.

## 9. Consejos y buenas prácticas

- ¿Qué características debe tener un buen mensaje en commit?
  - Ser corto breve y conciso con la información sobre la que trata el commit
- ¿Qué archivos no deberían subirse a un repositorio y cómo se pueden excluir?
  - Principalmente contraseñas, datos personales, etc. Estas cosas se pueden excluir usando un archivo ". gitignore".
- ¿Por qué es importante mantener actualizado el repositorio y como se logra?
  - Debido a que así podemos conservar nuestros proyectos en un lugar seguro en caso de errores o perdidas y otro tipo de problemas más, además los códigos de un repositorio no se pueden modificar dentro del mismo a no ser que te los descargues o seas el propietario.