**CLASE 2 DEL CURSO DE PYTHON**

**AULA MATRIZ**

* Interfaces de usuario:

Manera por la cual una persona puede establecer comunicación con una aplicación.Ha ido evolucionado a medida que ha pasado el tiempo desde:

CLI: Consola de Comandos

GUI: Interfaz Gráfica de Usuario

NUI: Interfaz de Usuario Natural

* Principios:

Estructura Simple: Organización significativa

Simplicidad: Comunicar fácil y limpio en el mismo lenguaje del usuario.

Visibilidad: Mostrar sin distraer con información redundante.

Realimentación: Informar sobre cambios en la aplicación de manera efectiva.

Tolerancia: Reducir costos de errores de uso.

Reusabilidad: Reducir la necesidad de repensar o recordar.

* Experiencia de usuario:

Usability – Design – Marketing – Human Factor – Accessibility – Utility – HCI – Ergonomics – System Perfomance.

* Prototipado:

Complementa otras metodologías de desarrollo.

Pretende reducir riesgos en un proyecto.

El cliente esta implicado en el proceso de desarrollo.

Algunos prototipos.

* Desarrollo Ágil

Modelo Corporativo.

Equipos cross-functional autoorganizados.

Actividades iterativas de desarrollo.

Aproximaciones centradas en el punto de vista de las personas.

Ej. Scrum, Kanban.

**GITHUB Y GIT**

Repositorio: Lugar centralizado donde está el código base y los demás pueden acceder a él.

Se implementaron los siguientes comandos para conocer el proceso de subir commits a un repositorio en GitHub.

Para descargar el repositorio a un nuestro computador:

1. Creamos una carpeta donde se clonarán los archivos.
2. Abrimos el Bash en ese directorio.
3. Pasamos el comando git clone URL\_del\_repositorio.

Para subir al repositorio los cambios realizados:

1. Para ver el estado de nustros archivos git status.
2. Para ver los cambios realizados en un archivo git diff Nombre del archivo
3. Para pasar los archivos del Working Directory al Staging Area git add Nombre del archivo.
4. Debo configurar mi nombre y correo para que las otras personas vean quien realizo el cambio. git config - - global user.name “…”
5. git config - - global user.email “…”
6. Se crea el commit con git commit -m “Alguna característica que se quiera dar del archivo”
7. Revisamos que no se hayan realizado ningún cambio durante el tiempo que duramos en descargar el repositorio y subir los cambios git pull.

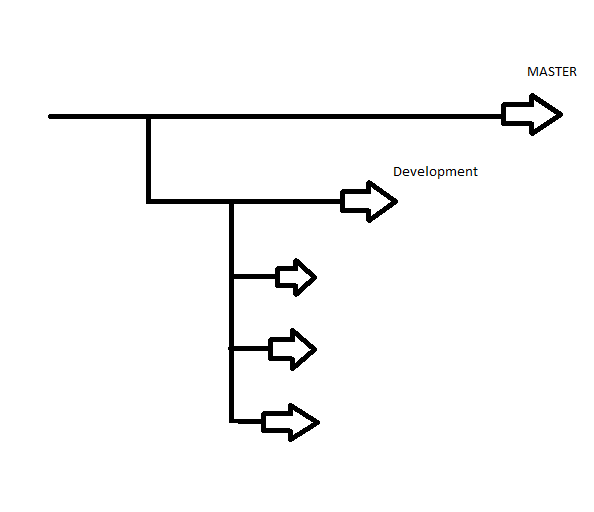
(Siempre hacer git pull antes del git push)

1. Finalmente lo enviamos a la nube con git push.

Para las versiones alternativas o Branch:

1. Este comando nos permite ver las versiones disponibles git branch
2. git branch nombre de la versión
3. Para moverme entre versiones git checkout nombre de la version

Los Branch o ramas se explican en el siguiente gráfico 1:



*Grafica 1/Explicación de los Branch o ramas en los repositorios de GitHub*