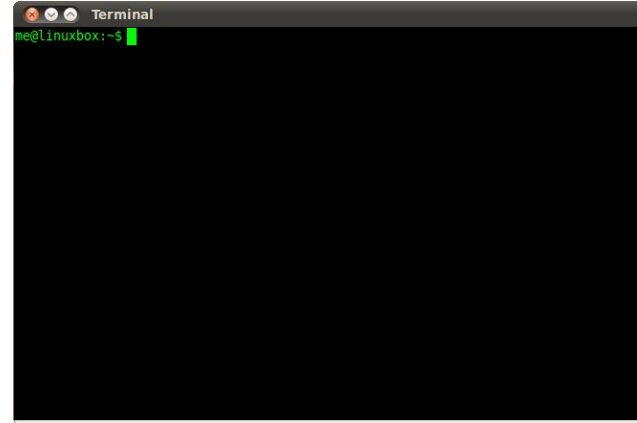


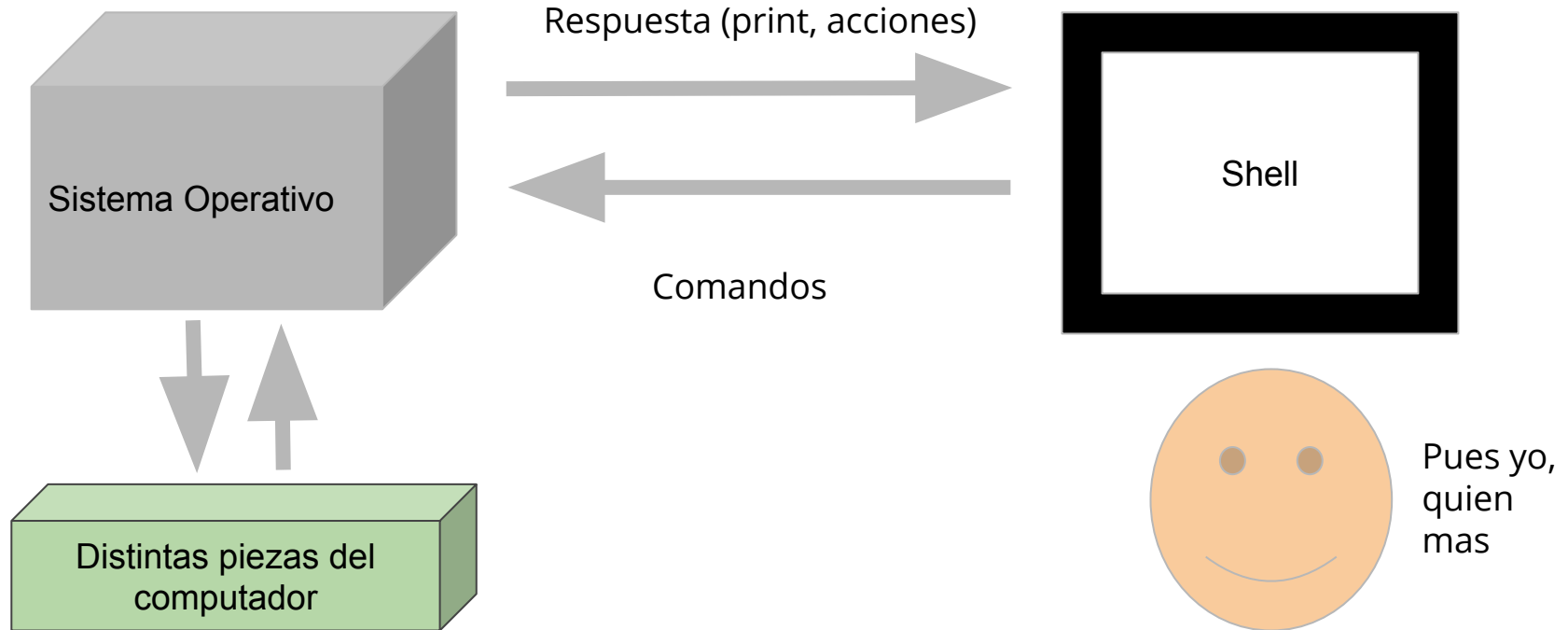
Ayudantia 01 GIT/Shell

Juan Pablo Olivares / Francisca Cattán



- *Tiempo final de la ayudantía lo dejaremos para dudas de la TC1
- *Si, se va a grabar
- *dudas ojalá por voz y con camara encendida (avisen si voy rapido)

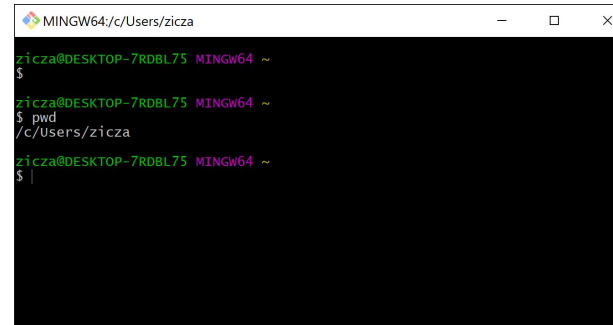
Para qué sirve el “Shell”



¿Por qué queremos ocupar el shell?

- Más rápido (eficiencia).
- Más funciones.
- Deja un log de lo que hemos realizado.

El poder ocupar fácilmente el shell nos permite tener “todo a la mano”, podremos compilar, correr, mover y guardar programas sin tener que abrir y cerrar pantallas constantemente.

A screenshot of a MINGW64 terminal window. The title bar shows the path 'MINGW64/c/Users/zicza'. The terminal has a black background with green text. It shows a user 'zicza' at a prompt '\$' in a directory '~'. The user enters 'pwd' and the terminal outputs '/c/Users/zicza'.

```
MINGW64/c/Users/zicza
zicza@DESKTOP-7RDBL75 MINGW64 ~
$
zicza@DESKTOP-7RDBL75 MINGW64 ~
$ pwd
/c/Users/zicza
zicza@DESKTOP-7RDBL75 MINGW64 ~
$ |
```

Ojo: No todos los terminales son iguales

En mi experiencia recomiendo utilizar terminales que soportan comandos tipo **UNIX**

El “*command prompt*” (**cmd**) de windows no tiene esto.

Algunos comandos que aparecen acá no van a funcionar en el **cmd** (generalmente existe pero con otro nombre)

***git bash si soporta UNIX**



Logo de git bash para windows

Comandos útiles

Para navegar:

- **cd** "*ruta_carpeta*": para moverse entre carpetas (cd .. para volver atrás)
- **ls**: muestra todo lo contenido dentro de la carpeta

Otros:

- **mkdir** "*nombre*" : crear carpeta
- **echo** "*texto*" : imprime texto en la consola
- **touch** "*archivo*" : crea un archivo (por default vacío)
- **cat** "*archivo*": lee archivo/ejecuta

Estos pueden ser concatenados a través de conectores (|, &, >), **útiles para la tarea**

ej: echo hola > bienvenida.txt

(crea un archivo bienvenida.txt con el texto hola)

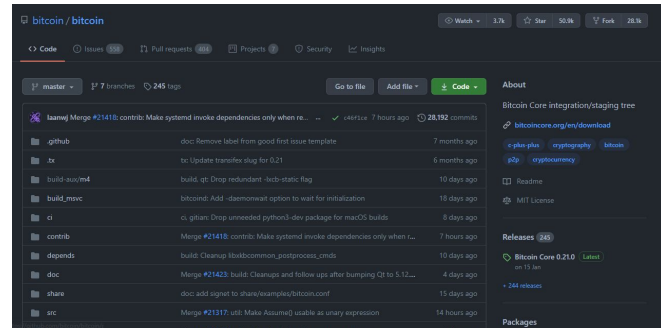
Git (no confundir con Github)

Git es un **software** utilizado en **proyectos**. Nos permite manejar versiones y avances fácilmente. Con git **varias personas** pueden trabajar tranquilamente en un repositorio sin preocuparse de perder el progreso, “pisarse las colas” o desordenarse.

GitHub es un **sitio Web** que, a traves de git, **almacena** repositorios, generando un ambiente útil para desarrolladores y para compartir código

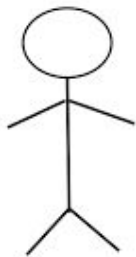


fuelle: git

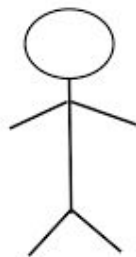


repositorio open source de bitcoin

Hola creemos
una página de
ventas?



Ok, creemos un
repositorio en
github para la
pagina



Quiero probar
una nueva
característica del
código



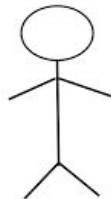
Crea otra rama y
trabaja sobre ella



Metí la pata con la
aplicación web



No te
preocupes
retrocedamos
unos commits



Quiero trabajar
con tal
librería/aplicación



Clona su
repositorio, es
open source!



Cómo funciona Git

Vamos a realizar un ejemplo práctico de clonar la tarea. Tratemos de enfocarnos en los **pasos y los comandos**.

