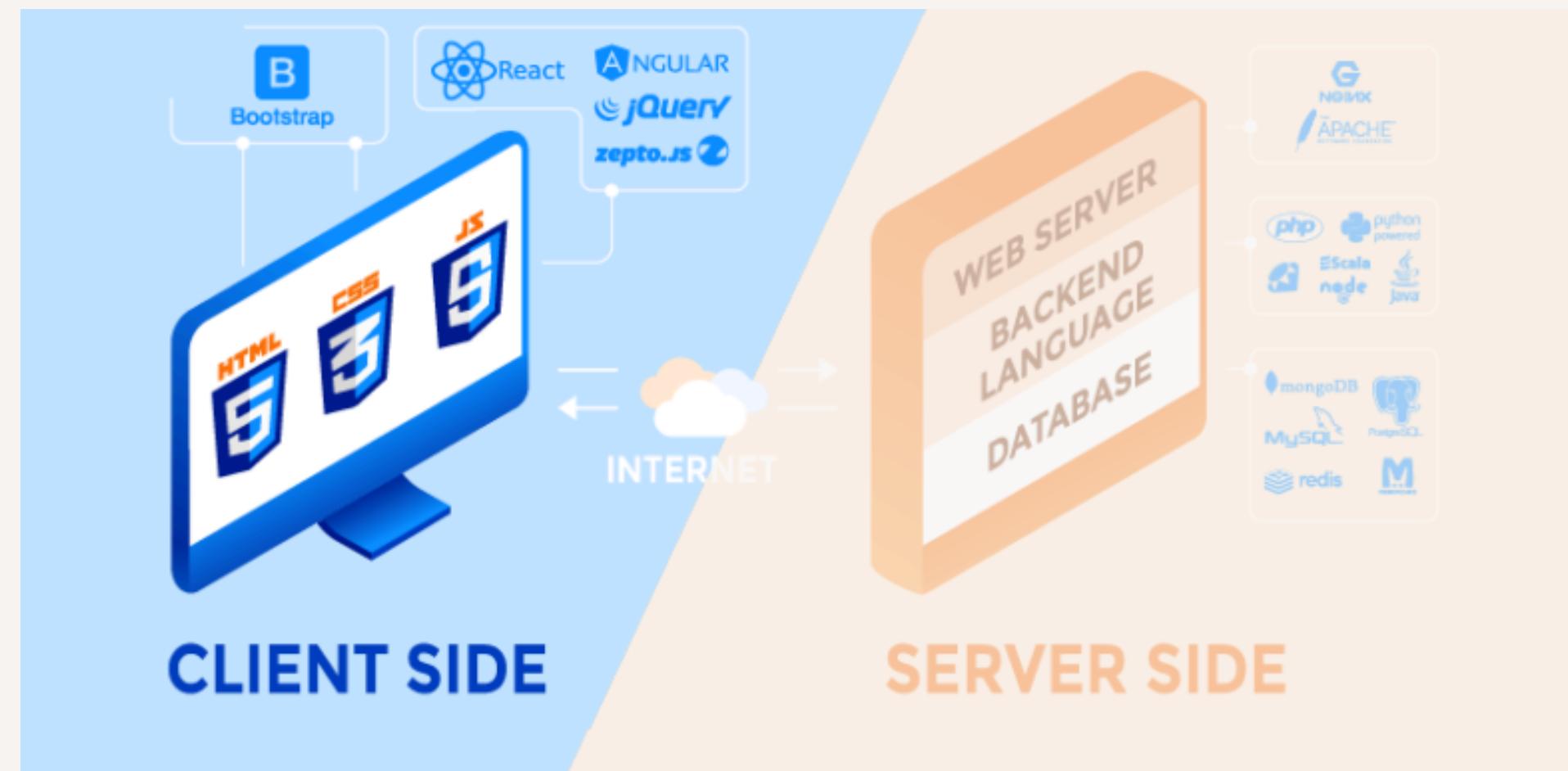


Unit 1

INTRODUCTION TO CLIENT-SIDE PROGRAMMING

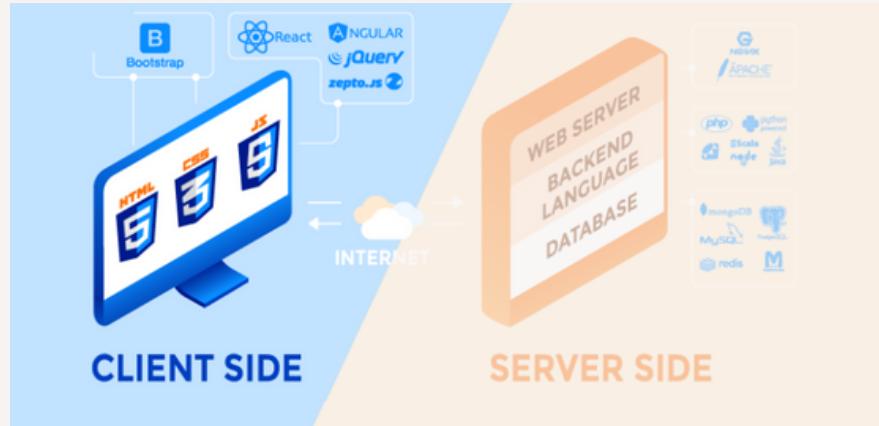
DESARROLLO WEB EN ENTORNO CLIENTE 24/25

ALEJANDRO VIANA RÍOS



Unit 1

Introduction to client-side programming



Index

1. Basic concepts

- Static vs dynamic websites
- Frontend vs backend
- Frameworks

2. Introduction to client-side programming

- Scripting vs programming languages
- Low-level, mid-level and high-level language/script
- Programming tools

3. Introduction to JavaScript

- What is JavaScript?
- What in-browser JavaScript can and can't do?
- JavaScript versions
- JavaScript engines
- JavaScript-based languages

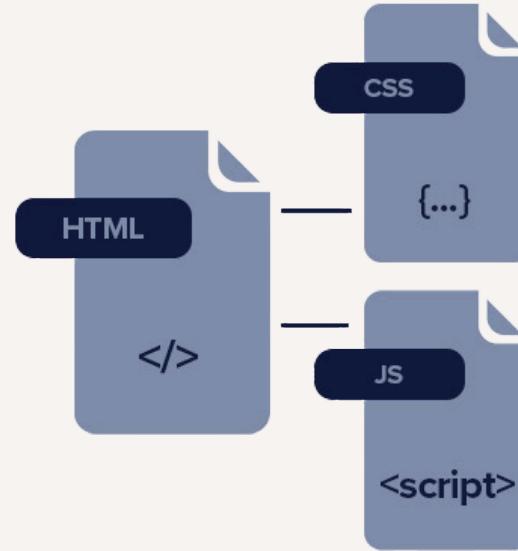
1.- Basic concepts

1. Static vs dynamic websites
2. Frontend vs backend
3. Frameworks

Static vs dynamic websites

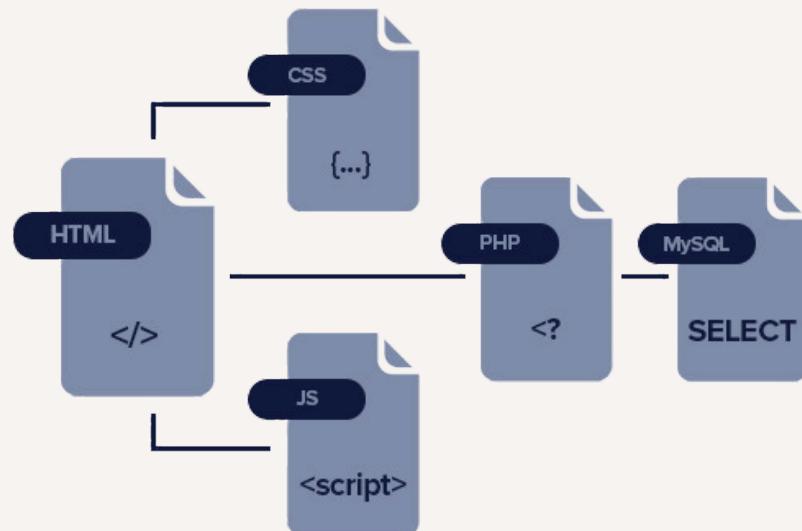
3.- Introduction to JavaScript > Static vs dynamic websites

Static Website



Static websites

- Web page is already present in the server with the information fullfilled
- Client gets web page from server
- Renders it and shows to the user
- It may have a small interaction, but information don't change



Dynamic websites

- Web page is not present in the server when the client ask for it
- When client requires a web page, the server creates it by connecting to a database
- Every time a web page is required, it could be different depending on user, time, settings...

Dynamic Website

Front-end

1.- Basic concepts > Front-end vs Back-end

What is?

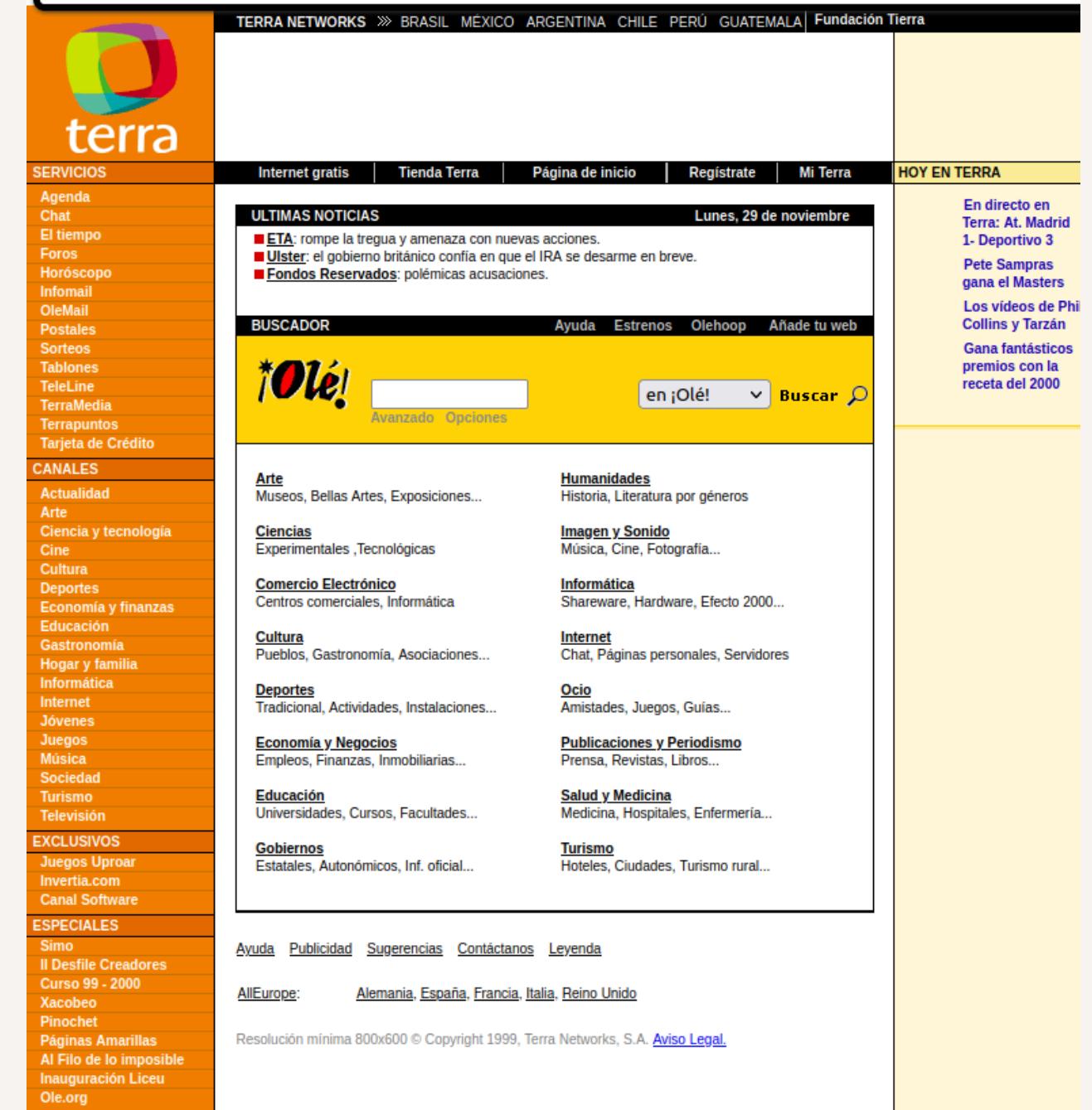
- Client-side part of a web application
- Everything a user sees and interacts with
- How fast the website loads
- How easy and comfortable it is to navigate through it
- How accessible it is to people with disabilities.

Why is so important?

Clients may run away, never to be seen, if a bad user experience is provided

Where could I get more information?

- Diseño web en entorno cliente (DWEC, 2º DAW)
- Diseño de interfaces web (DIW, 2º DAW)



Back on the 1990's, all sites that wanted to be on the top offered a lot of information. It was not easy to find the information you were looking for nor navigate through them

Front-end

1.- Basic concepts > Front-end vs Back-end

Developer tasks

- Create a style guide (typography, images, colors, etc.)
- Create a user interface that matches brand's identity
- Create a website usable and accessible on a wide variety of devices, browsers, connections and screen widths.
 - Usable: optimized load times, easy interaction...
 - Accessible: people with disabilities

Developer skills

- Knowledge of your target audience and demographic.
- Visual creativity
- Prefer artistic side of the job
- Careness about how users feel



Front-end technologies for static websites

1.- Basic concepts > Front-end vs Back-end

- HTML 5. Markup language used for structuring and presenting content on the World Wide Web
- CSS 3. Language used for describing the presentation of an HTML document.
- JavaScript ES6 or above (up to ES13). Language used to provide interactivity to a web site



Front-end technologies for dynamic websites

1.- Basic concepts > Front-end vs Back-end

- JavaScript ES6 or above (up to ES13)
- Frameworks (Vue, Redux, Angular, etc) (DWEC). Set of tools, focused on a programming language, to support and aid on building software
- Libraries (Node.js, JQuery, etc.). Pieces of functional code, already proven, free and ready to be used on a project.
- Prototyping (Figma, Sketch, etc.) (DIW)



Back-end

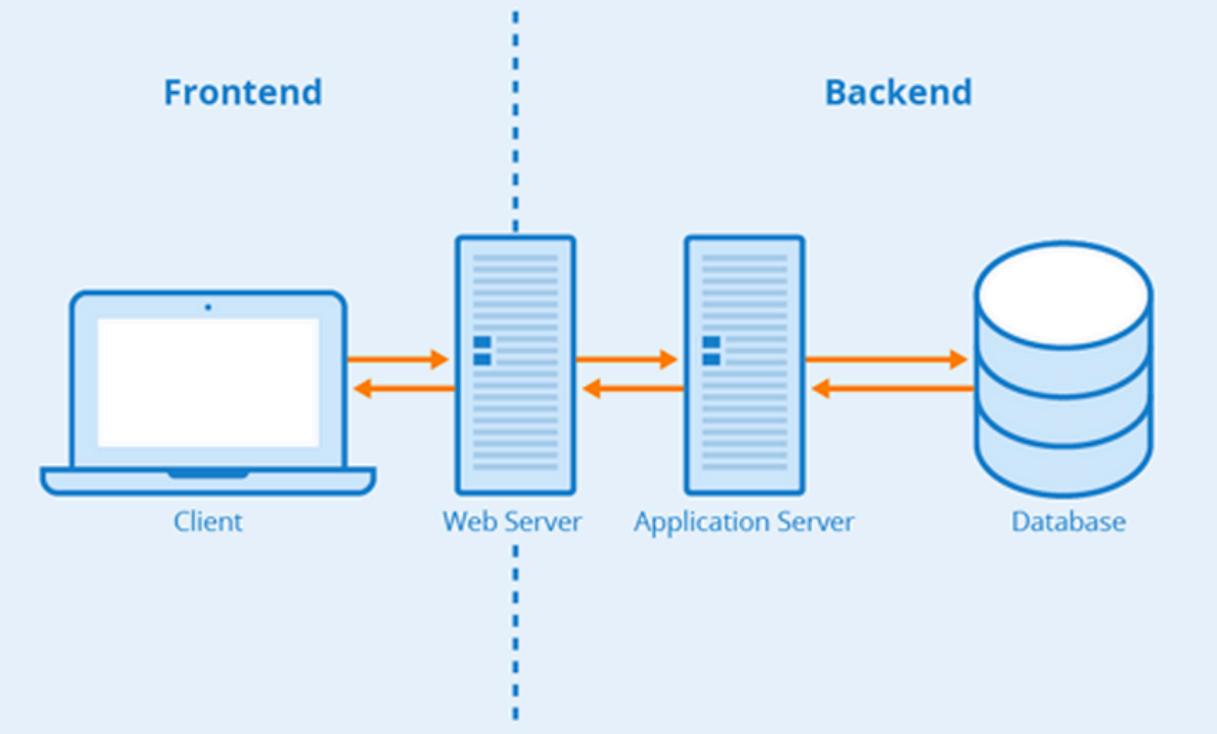
1.- Basic concepts > Front-end vs Back-end

What is?

- Server-side part of a web application
- Related with how data is stored and manipulated
- Algorithm that determines what application does

Why is so important?

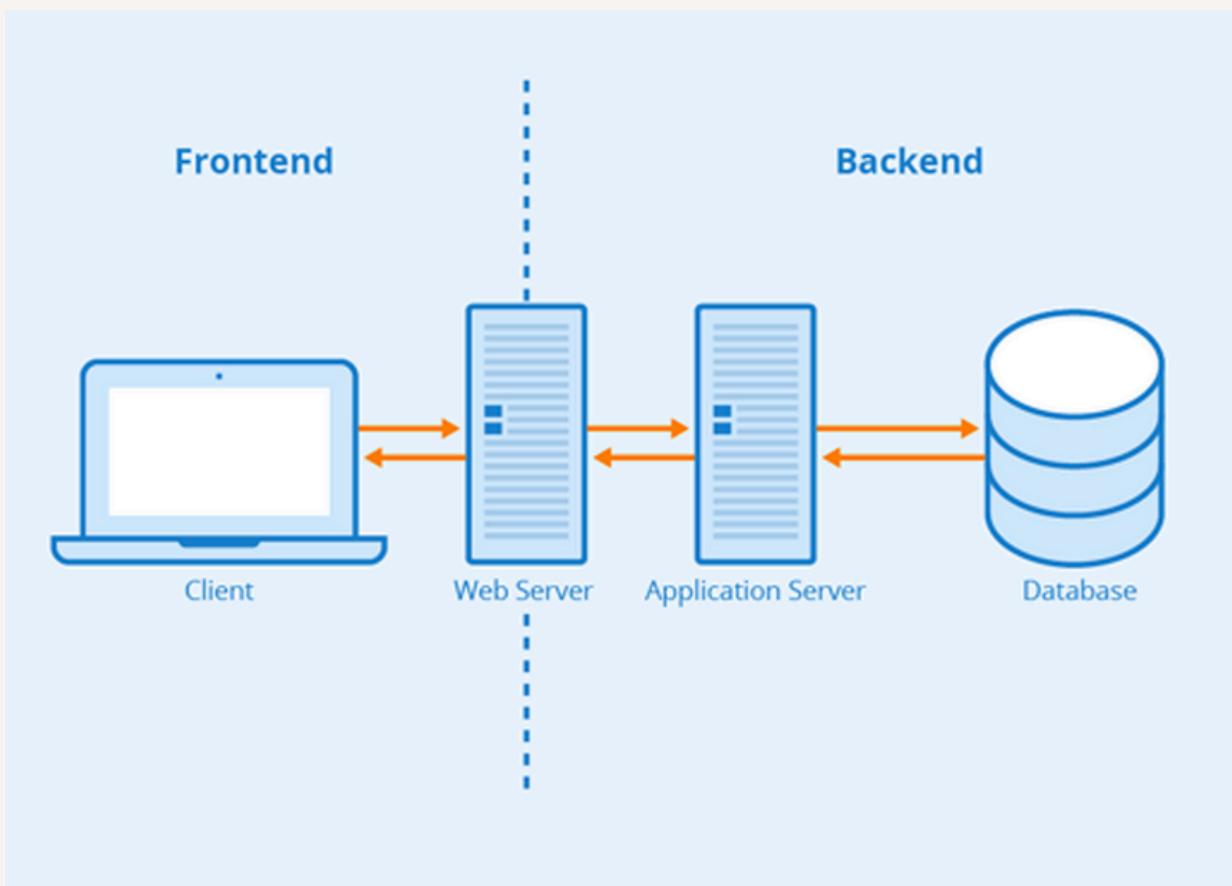
- Responsible of presenting data to client
- Responsible of how the website behaves and what it allows to users
- Bad design could leave website unusable or slow



Frontend and backend. It is common that web server, application server and database server are hosted on the same server

Back-end

1.- Basic concepts > Front-end vs Back-end



Frontend and backend. It is common that web server, application server and database server are hosted on the same server

How does it work?

- A server (web server) is listening for incoming requests from the client (web browser)
- Server-side scripts processes the request
- They collects data from database, process it and send it to the client

Where could I get more information?

Diseño web en entorno servidor (2º DAW)

Back-end

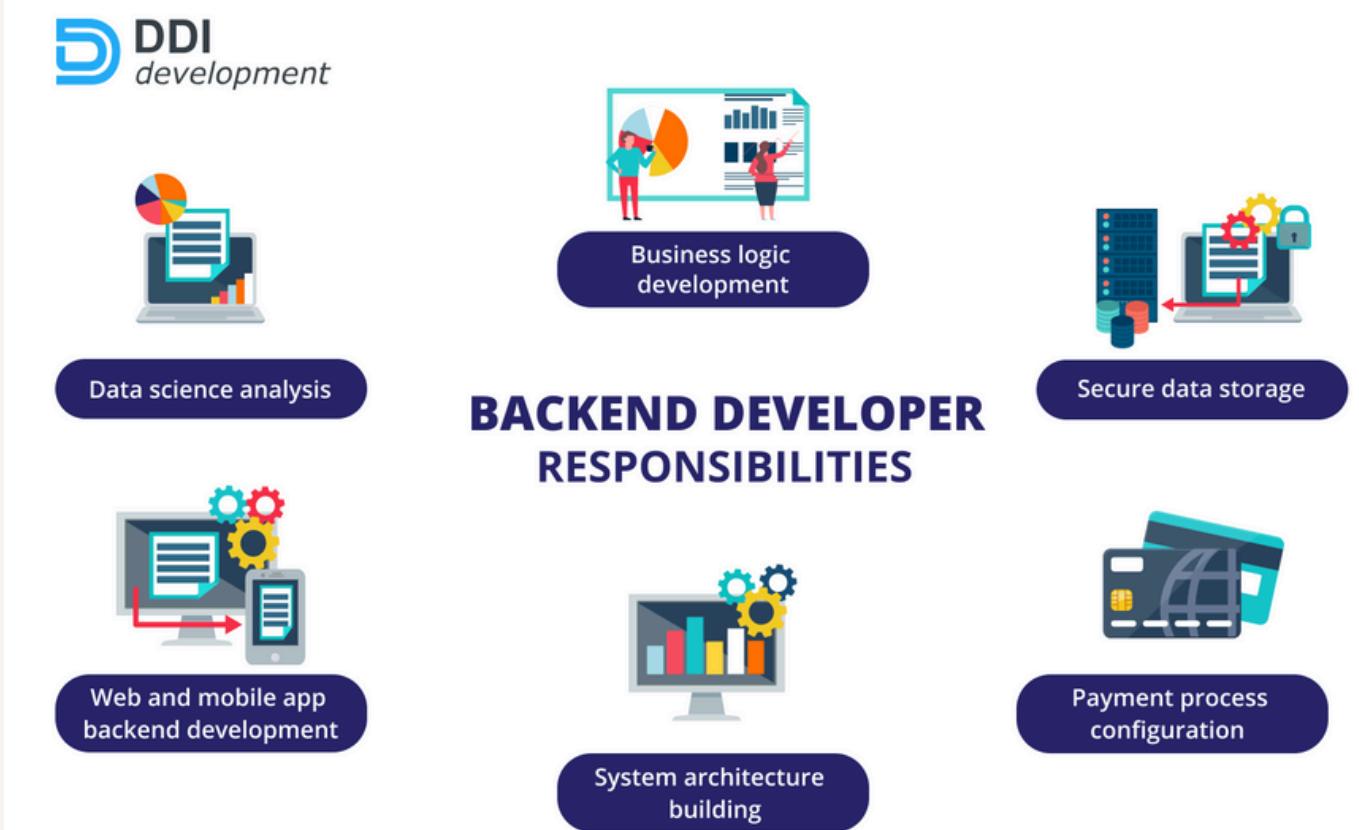
1.- Basic concepts > Front-end vs Back-end

Developer tasks

- Creating, managing, and maintaining the database
- Installing and maintaining servers
- Creating scripts to develop website functionality
- Working with APIs that, usually, support CRUD operations
- Interacting with third-parties services

Developer skills

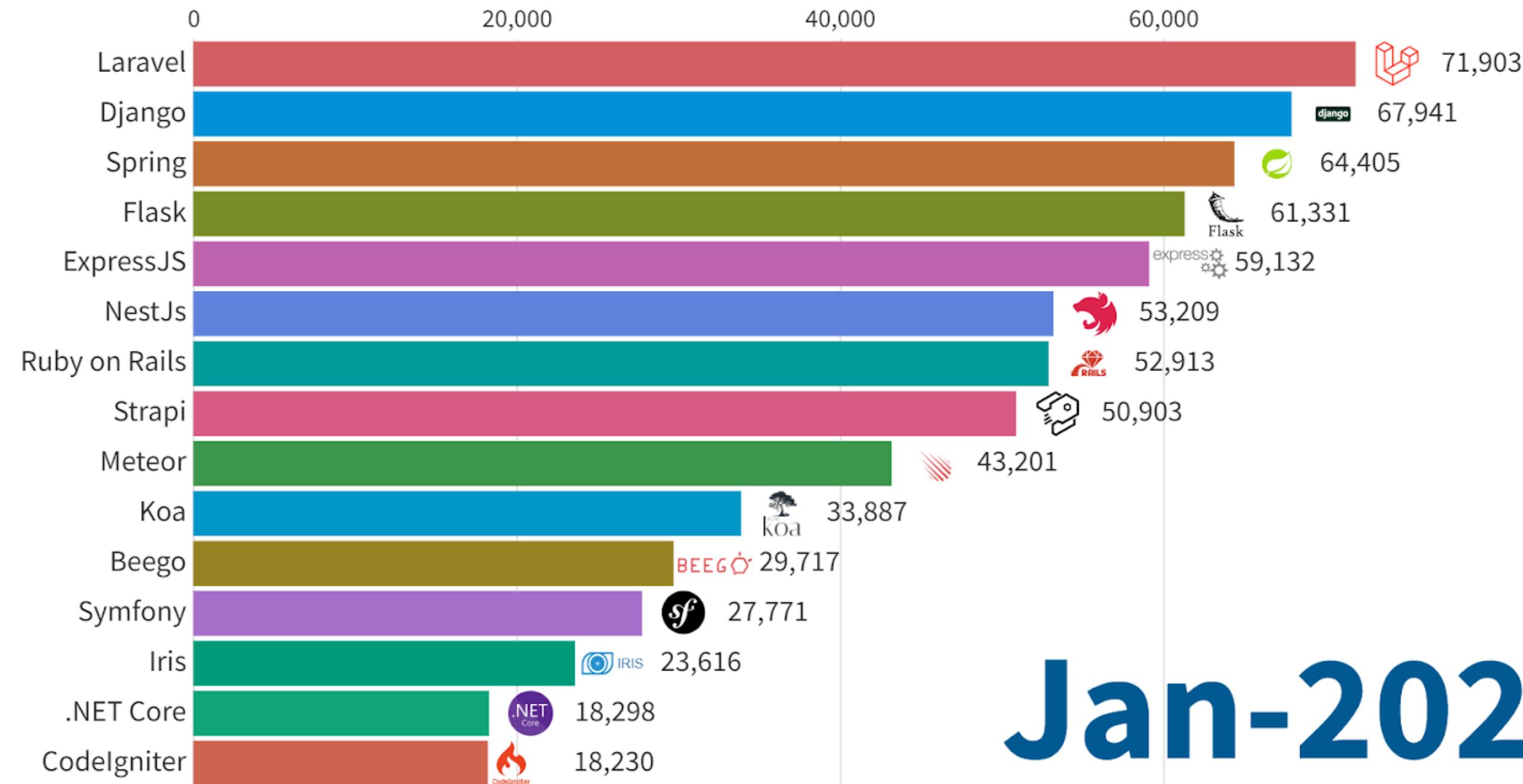
- Problem-oriented profile
- Structured, analytical and methodical person
- Proficiency analyzing and organizing large amounts of data



Back-end technologies

1.- Basic concepts > Front-end vs Back-end

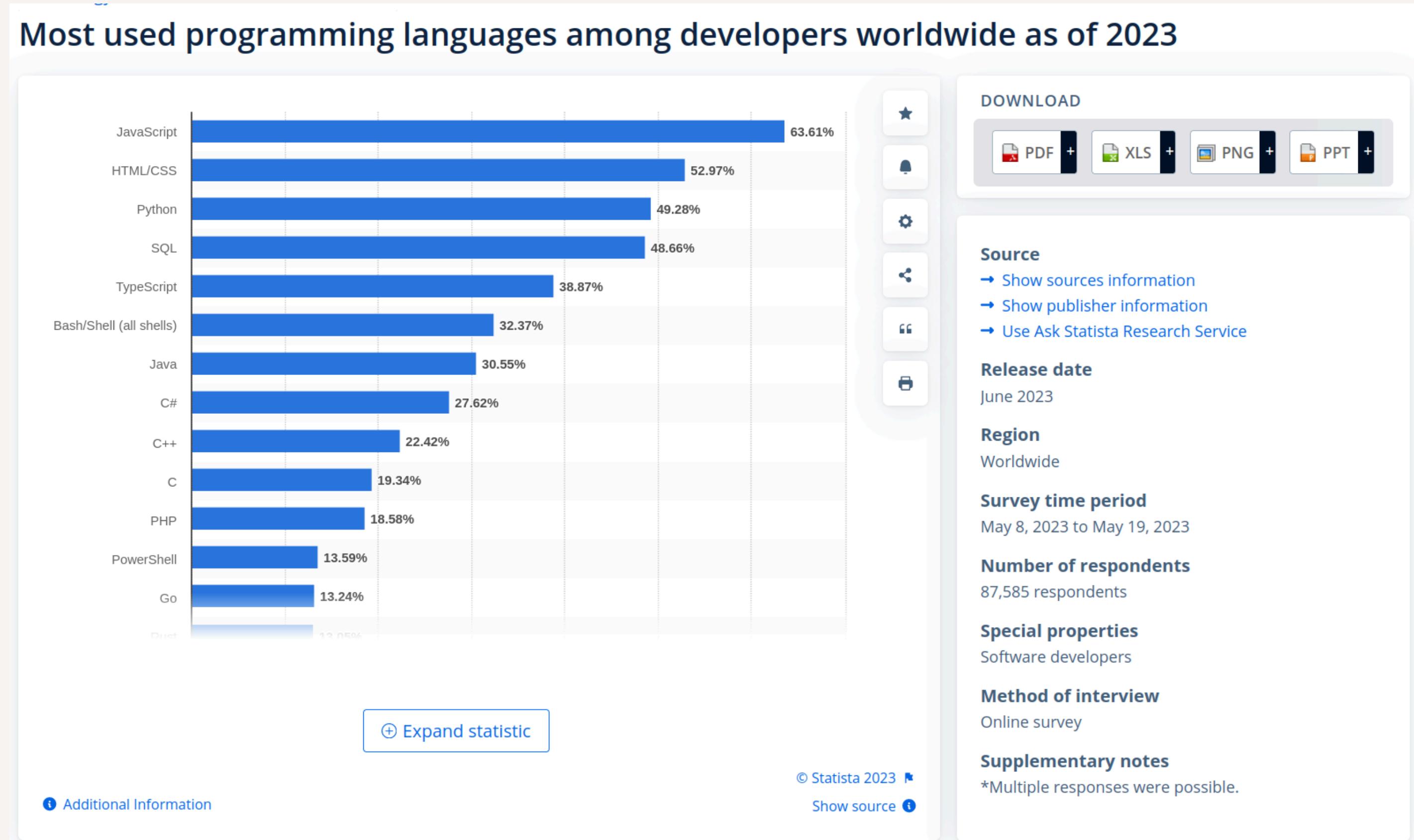
Most Popular Backend Frameworks



Backend frameworks. A framework is a Set of tools, focused on a programming language, to support and aid on building software

Back-end technologies

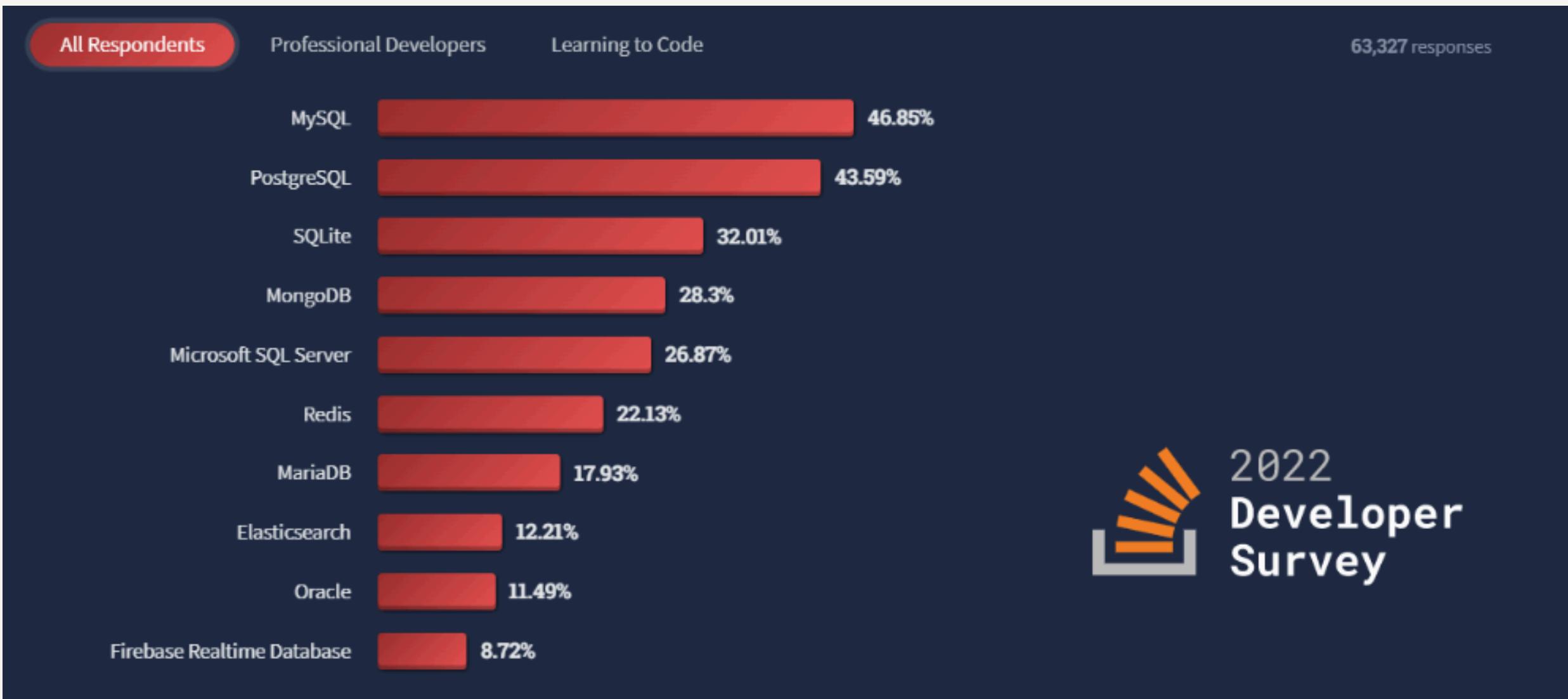
1.- Basic concepts > Front-end vs Back-end



Backend most used programming languages. Yes, Javascript. Thanks to Node.js

Back-end technologies

1.- Basic concepts > Front-end vs Back-end



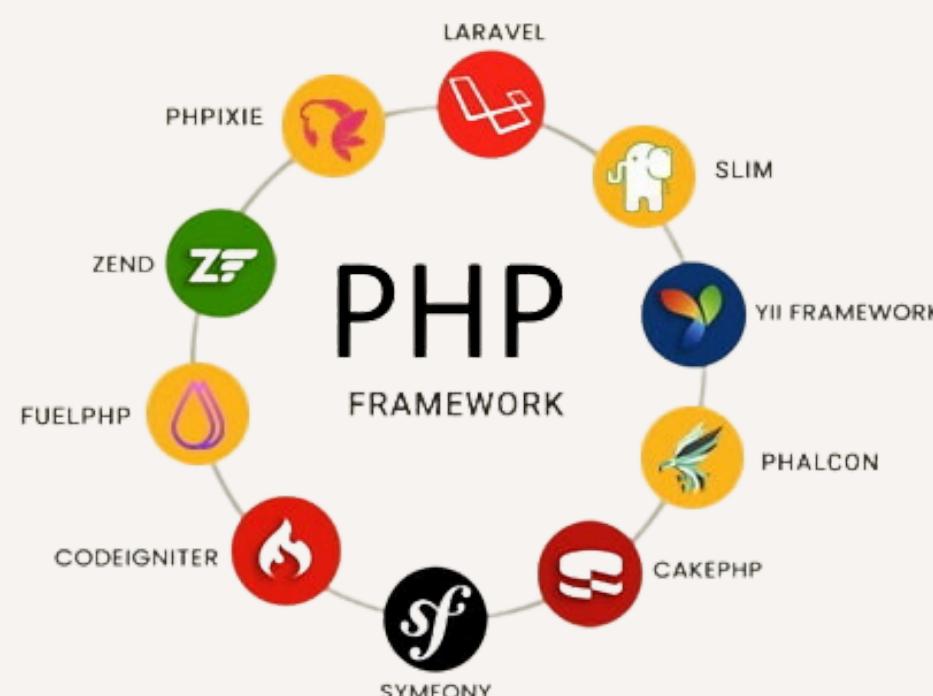
Backend most used database servers

Frameworks

1.- Basic concepts > Frameworks



Javascript Frameworks



PHP Frameworks

What is a Framework?

- Software that helps to build software
- Provides a scaffolder to work with
- Provides aids, tasks and programs that speeds up software building
- Each one is usually focused on one programming language

Frameworks

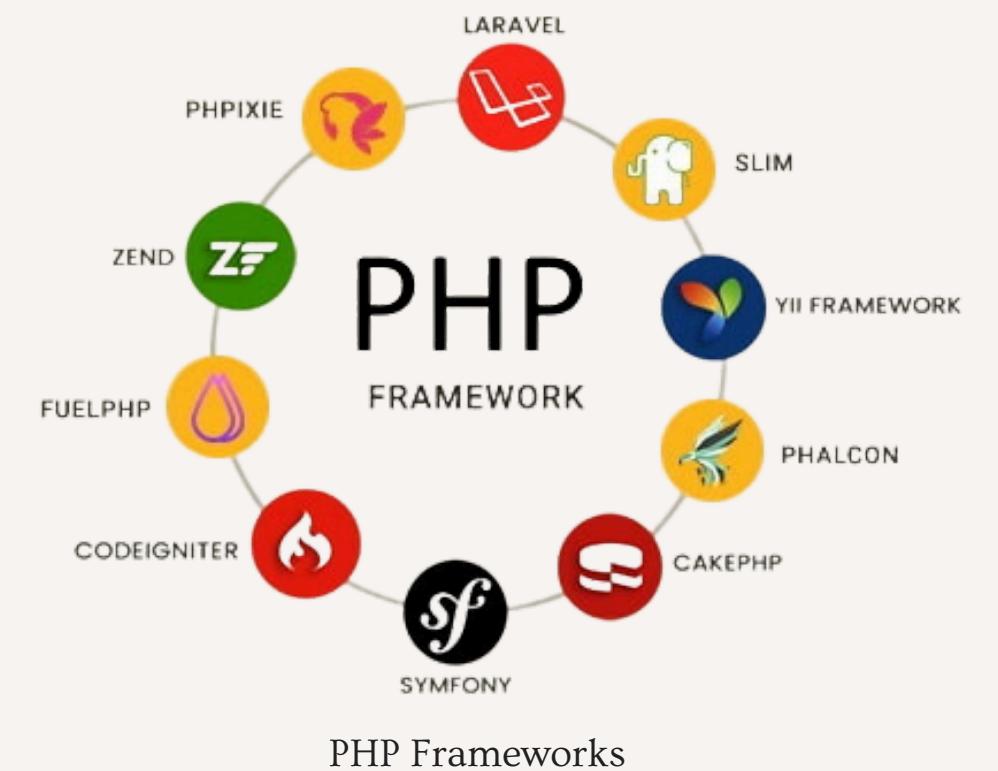
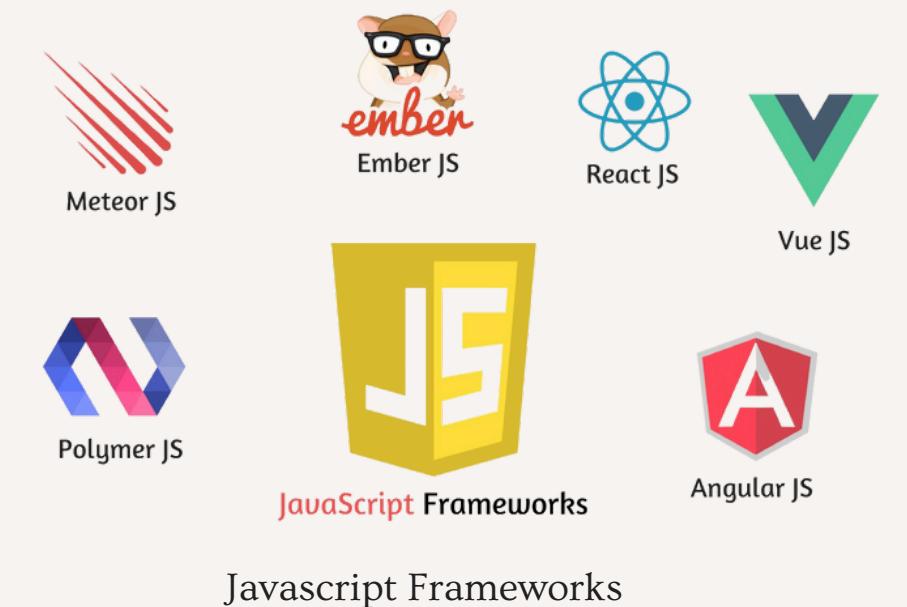
1.- Basic concepts > Frameworks

Advantages of using a Framework

- you can focus on the core software development rather than the details
- It saves time
- Reduces the number of errors
- It's easier to collaborate with colleagues
- Avoids redundant software
- Helps to write clean and secure code

Disadvantages of using a Framework

- the code is not optimal
- you lose control over your code



2.- Introduction to client-side programming

1. Scripting vs programming languages
2. Programming language levels
3. Translating languages
4. Programming tools

Scripting vs programming code

2.- Introduction to client-side programming > Scripting vs programming code

Scripting language

- Used to automate tasks (maybe interacting with applications)
- They are interpreted
- Less code-intensive
- Every scripting language is a programming language (not the other way round)
- Have limited access to computer resources

Types

- Server-side: PHP, Ruby, Phyton, (JavaScript)
- Client-side: JavaScript, CSS

Programming language

- Used to create applications
- They are compiled
- More code-intensive
- Have unlimited access to computer resources

Examples

- C, C++, C#, Java...

Low-level programming language

2.- Introduction to client-side programming > Programming language levels

Code is expressed in binary and uses CPU instructions set

Advantages

- Directly executable by CPU. No translation needed
- Execution is the fastest possible
- Code is optimized for this architecture (a group of CPUs)
- Lightweight code

Disadvantages

- Difficult to understand by people
- Difficult to express a real world problem in terms of a low-level code
- Prone to make mistakes
- Code is CPU dependant. New CPU requires new code

01101100
01101111
01110110
01100101

Mid-level programming language (assembly)

2.- Introduction to client-side programming > Programming language levels

Nemotecnics and variable names are utilized to make programming easier

Advantages

- Easier to code with than binary

Disadvantages

- Code utilizes nemotecnics, so it needs translation. CPU does not understand it
- Not the best performance, neither the worst
- Easier for programmers to understand and to solve real life problems than low-level but still difficult
- Code can be executed only on the same architecture (a group of CPUs)
- Code weights a little bit more

```
Load r1, X
Load r2, Y
Mult r3, r2, r1
Load r4, A
Mult r2, r4, r1
Add r5, r2, r4
Mult r1, r2, r5
Load r3, B
Add r7, r1, r3
```

High-level programming language

2.- Introduction to client-side programming > Programming language levels

Advantages

- The easiest to code with
- Programmers can easily express real life problems and understand code
- Code can be executed on several CPUs, even from different architectures

Disadvantages

- Code needs translation (compilation or interpretation). CPU does not understand it
- Code weights a lot more
- The worst performance. Code is not optimized at all

Examples

- PHP, C, C++, C#, Java, JavaScript, ...

```
const para = document.querySelector("p");
para.addEventListener("click", updateName);

function updateName() {
    const name = prompt("Enter a new name");
    para.textContent = `Player 1: ${name}`;
}
```

a Javascript code

Compilated languages

2.- Introduction to client-side programming > Translating languages

Compilation

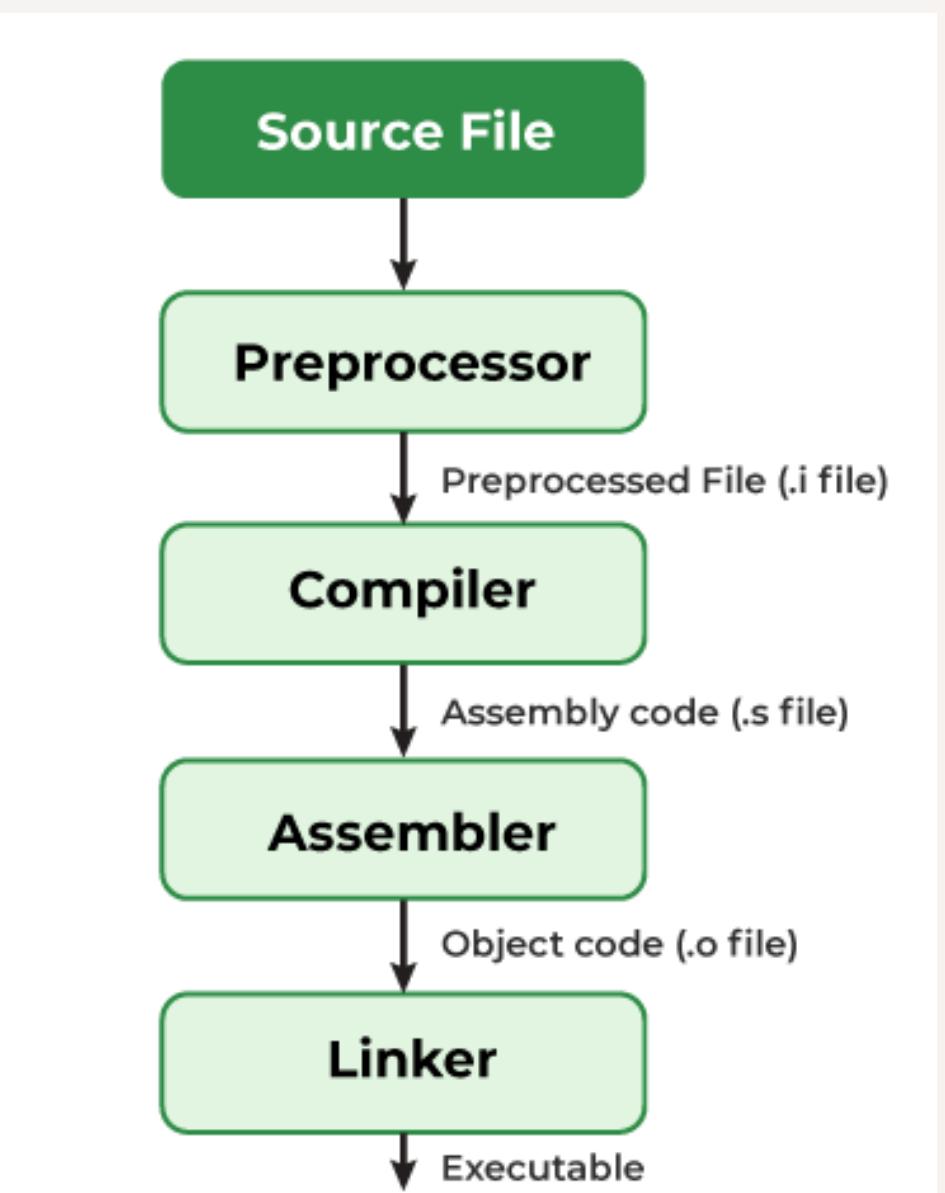
- All code is translated (compiled) using several stages, and stored in one executable architecture-dependant file.
- If executing in another architecture is needed, code must be recompiled (with a compiler for this architecture)
- Once executable file is obtained, it is fast to execute the program

Advantages

- Once obtained executable, it is fast to execute the program
- All errors are detected before execution

Disadvantages

- It can be slow to get the executable file



Interpreted languages

2.- Introduction to client-side programming > Translating languages

Interpretation

- Interpreter takes a line, search for errors, optimize it, translate it into machine code and execute. No file is created nor stored
- If executing in another architecture is needed, an interpreter for such architecture executes source code must

Advantages

- Memory efficient
- Quick to start executing

Disadvantages

- At the end executing is slower

Interpreted vs compiled languages

2.- Introduction to client-side programming > Translating languages

Interpreter Vs Compiler

Interpreter	Compiler
Translates program one statement at a time.	Scans the entire program and translates it as a whole into machine code.
Interpreters usually take less amount of time to analyze the source code. However, the overall execution time is comparatively slower than compilers.	Compilers usually take a large amount of time to analyze the source code. However, the overall execution time is comparatively faster than interpreters.
No Object Code is generated, hence are memory efficient.	Generates Object Code which further requires linking, hence requires more memory.
Programming languages like JavaScript, Python, Ruby use interpreters.	Programming languages like C, C++, Java use compilers.

Writing code

2.- Introduction to client-side programming > Programming tools

variables.js - Sin título (área de trabajo) - Visual Studio Code

Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda

EXPLORADOR

SIN TÍTULO (ÁREA DE TRABAJO)

- > Diseño
- < Programación
 - < estilos
 - main.scss
 - < js
 - duerme.js
 - hola.js
 - variables.js**
 - index.html
 - index2.html

index.html js variables.js

Programación > js > JS variables.js > ...

```
1 //puesto que vamos a usar código moderno, hay que especificar use strict
2 "use strict";
3
4 //ejemplo del uso de variables, constantes y tipos
5 //las variables se deben definir con let. var es anticuado
6 let numero=5;
7 let cadena;
8 cadena="5";
9 const NUMERO=7, CADENA="6";
10
11 const NOMBRE = prompt("Digame su nombre");
12 console.log (`Encantado de saludarle, ${NOMBRE}, tengo entendido que tiene ${prompt}`);
13 alert (`Buenos días ${NOMBRE}`);
14
15 console.log ("El resultado de hacer la operación numero+cadena es", numero+cadena);
16 console.log ("El resultado de hacer la operación numero+NUMERO es ${numero+NUMERO}");
17
18 /*Las variables en javascript son débilmente tipado, es decir, no declaro su tipo
19 Es recomendable que los nombres de las variables sean lo más descriptivos posible
20
21 cadena=5 //la variable llamada "cadena" es de tipo entero! ¡¡MAL!!
22 console.log ("El resultado de hacer la operación numero+cadena es", numero+cadena);
```

EXTENSIONES

Buscar extensiones en Marketplace

INSTALADO

- Git Graph 80ms mhutchie
- HTML Boilerplate sidthesloth
- Live Server 45ms Ritwick Dey
- Spanish Language Pack for Visual Studio Code Microsoft

Lín. 24, col. 1 Espacios: 4 UTF-8 LF {} JavaScript Go Live

Visual Studio code is a powerful code editor with lots of extensions and aids

Testing code

2.- Introduction to client-side programming > Programming tools

The screenshot shows the Playcode interface. At the top, there's a navigation bar with icons for file operations, a title "JavaScript Playgro...", a "Share" dropdown, and user links for "Learn", "Sign In", and "Sign Up". Below the navigation is a dark-themed sidebar with "FILES +" and "PACKAGES +". Under "FILES +", there are files: "index.html" (closed), "script.js" (open), and "style.css". Under "PACKAGES +", there is "index.html". The main workspace contains two tabs: "Console" and "Web View". The "Console" tab shows the output "Hello world". The "Web View" tab displays the text "Hello world" in large yellow font. A status bar at the bottom right includes a question mark icon.

```
const message = 'Hello world' // Try edit me
// Update header text
document.querySelector('#header').innerHTML = message
// Log to console
console.log(message)
```

Playcode is a powerful online JavaScript (among others) execution environment

Looking for help

2.- Introduction to client-side programming > Programming tools

The screenshot shows the main homepage of Stack Overflow. A large central banner features the text "Every developer has a tab open to Stack Overflow". Below the banner, several key statistics are displayed: "100+ million monthly visitors to Stack Overflow & Stack Exchange", "45.1+ billion times a developer got help since 2008", "191% ROI from companies using Stack Overflow for Teams", and "5,000+ Stack Overflow for Teams instances active every day". To the left and right of the banner are two callout boxes: one encouraging users to "Join the community" and another promoting "Discover Teams". The background of the page is dark with colorful geometric shapes.

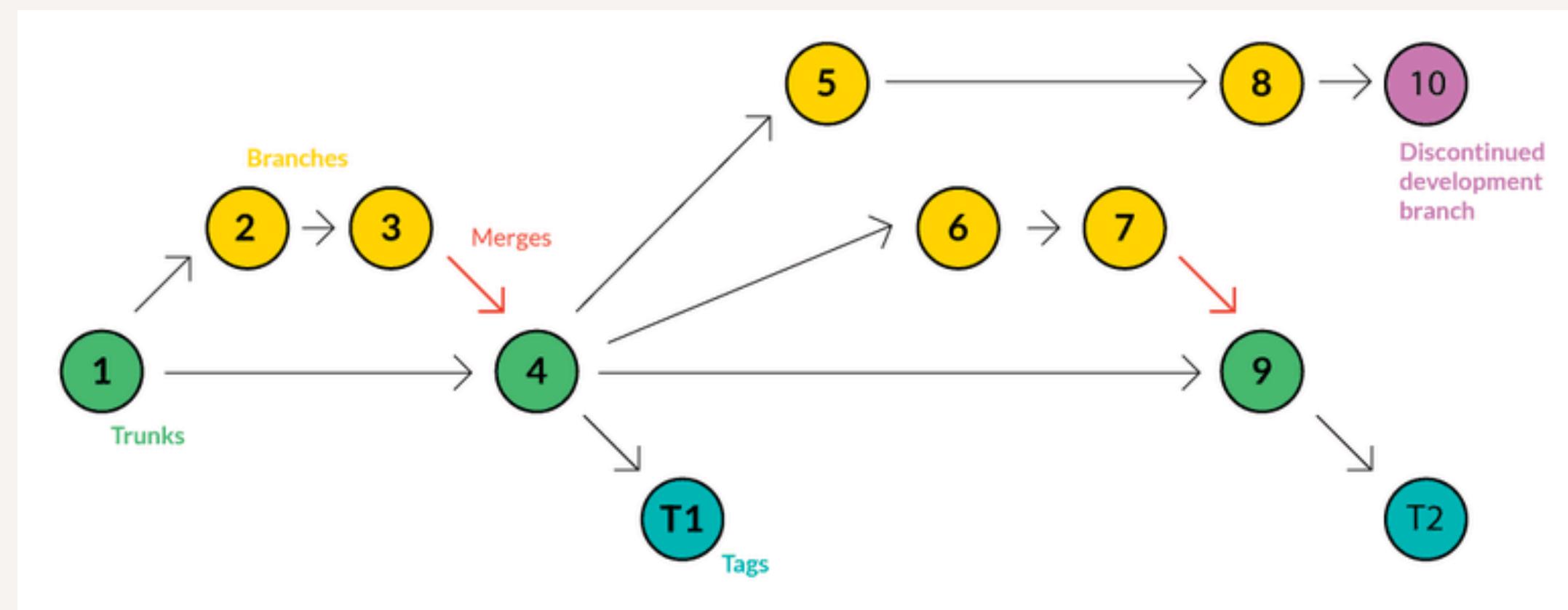
The top part of the screenshot shows a question page for "What does 'use strict' do in JavaScript, and what is the reasoning behind it?". The question was asked 14 years ago and has 8430 answers. The bottom part shows a search results page for the tag "javascript", with 34,238 questions listed. Both pages include navigation menus, user profiles, and sidebar information.

StackOverflow is the place to go when you have any question. [English](#) and [Spanish](#)

Benefits of using a version control system

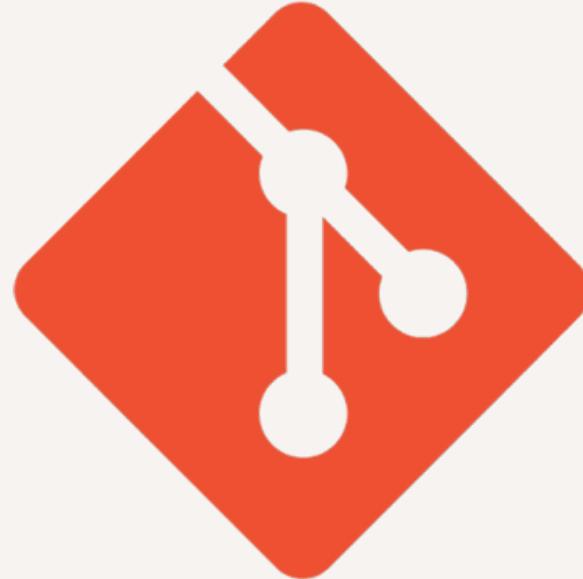
2.- Introduction to client-side programming > Programming tools > version control system

- Tracks changes on source code over time allowing to revert them
- Prevents source code from accidental changes
- Allows users to work at the same time, on different parts of the same code and eventually merge them
- Allows moving along versions effortlessly
- Avoids having to use messy file names like “definitivo”, “último”, “final”, “final v1”,...



Git

2.- Introduction to client-side programming > Programming tools > Version control system

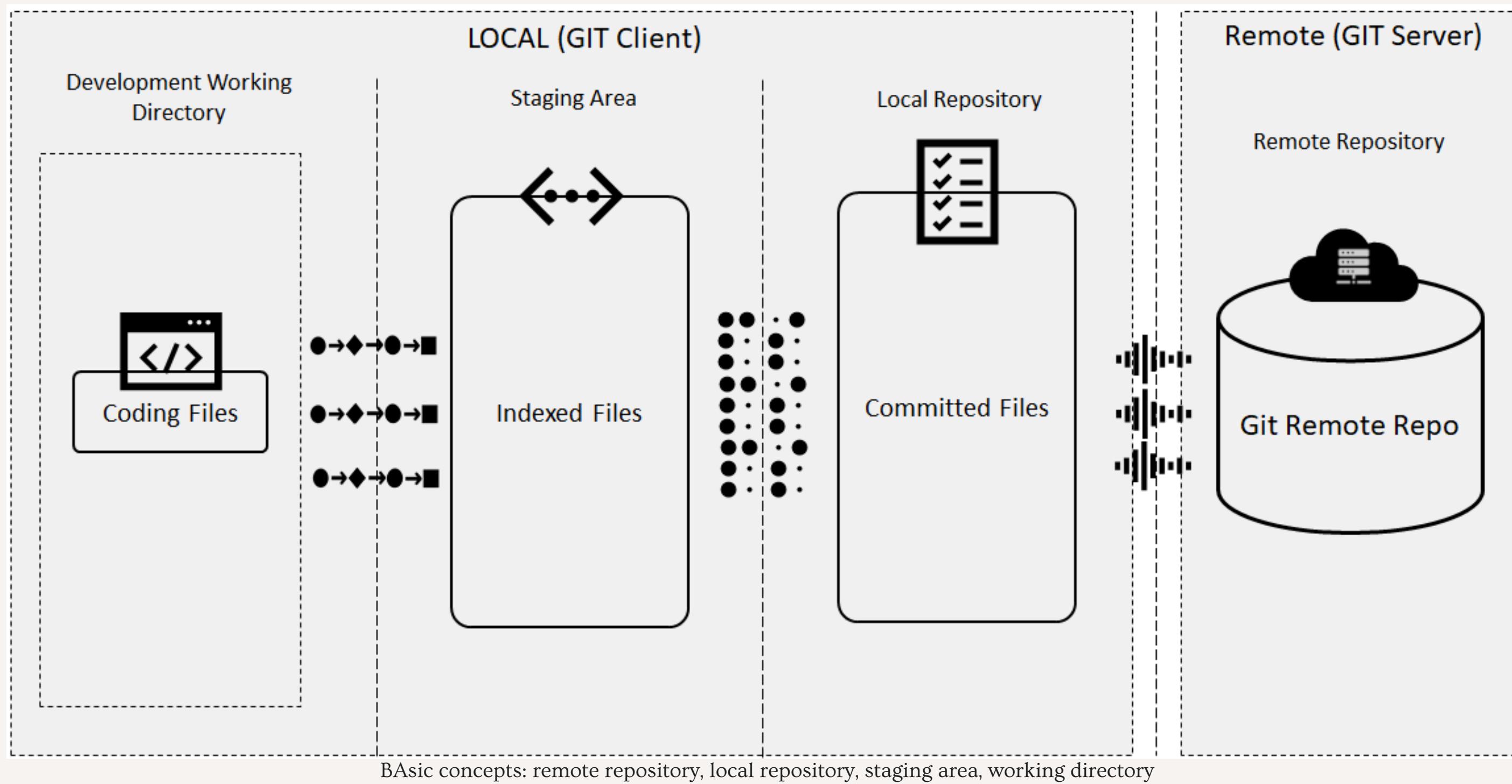


git

- It is a distributed control version system
- It is not the only one, but it is the de facto standard
- It is free and open source
- There are several websites working with Git system
(github, gitlab, bitbucket)...

Basic concepts

2.- Introduction to client-side programming > Programming tools > Version control system



Basic concepts

2.- Introduction to client-side programming > Programming tools > version control system

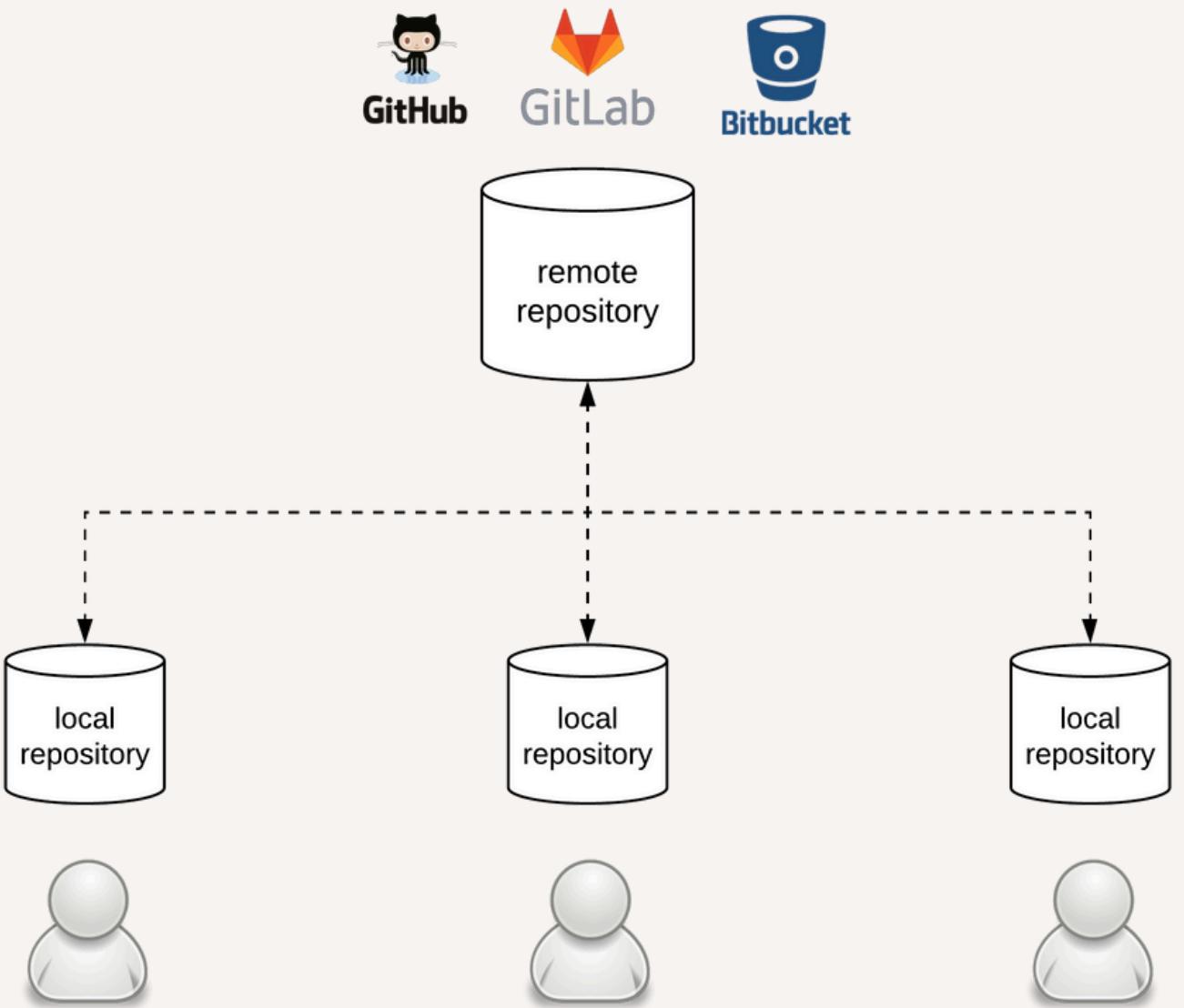
Remote repository

- Contains a master copy of the whole project
- Each developer copies (clone) remote repository into his local repository



Local repository

- Contains the original project modified by the developer
- When a milestone is reached, code is uploaded to remote repository
- If something breaks at local, a copy of remote repository can be cloned again



Basic concepts

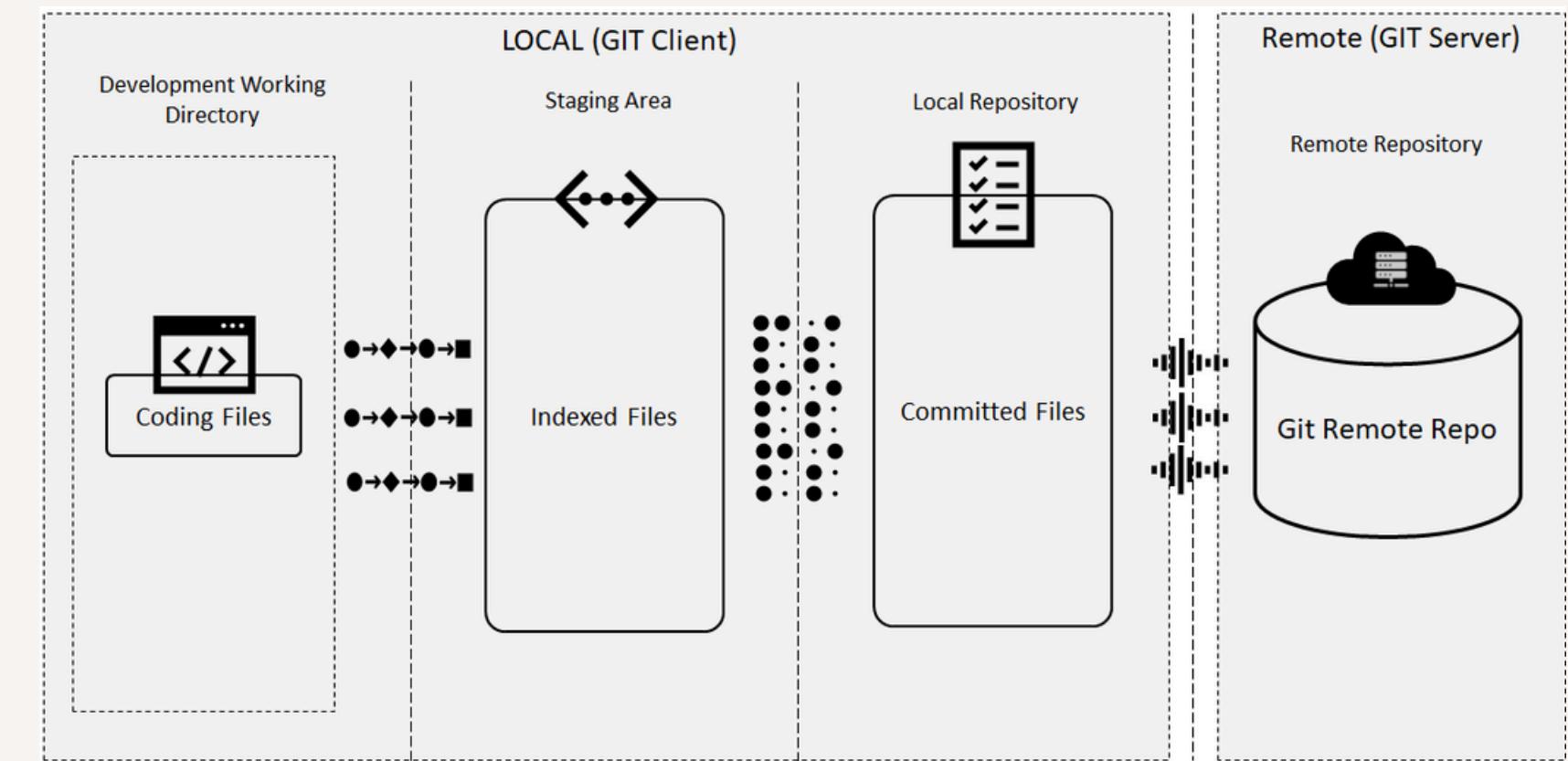
2.- Introduction to client-side programming > Programming tools > version control system

Staging area

- Intermediate storage area prior to update local repository
- It implies information about local changes being stored in `./git/index` file

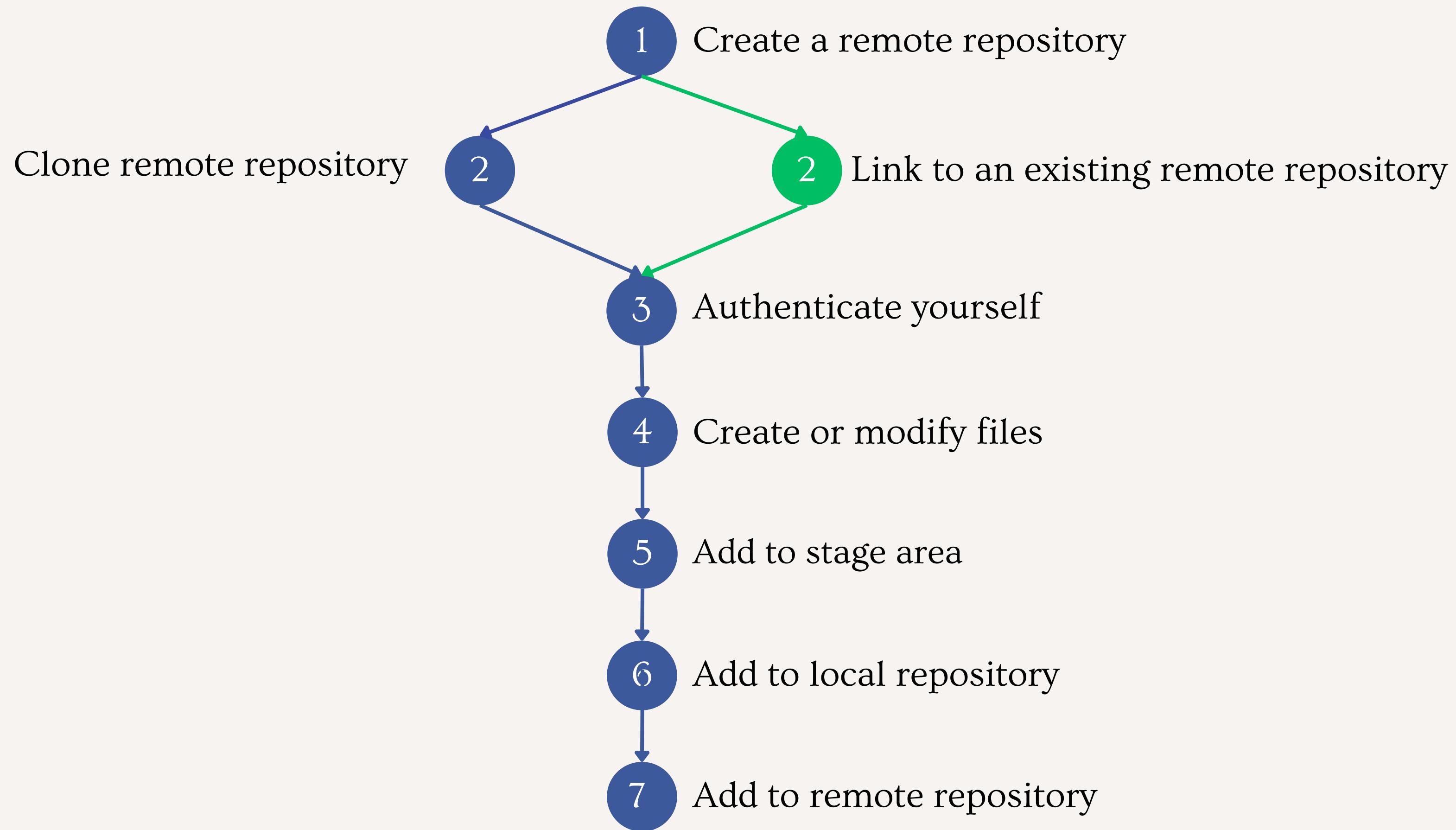
Working directory

- Directory where remote repository is downloaded (cloned)
- Is where developer will work at



Basic workflow

2.- Introduction to client-side programming > Programming tools > Version control system



Cloning an already existent github project

2.- Introduction to client-side programming > Programming tools > Version control system

The screenshot shows a GitHub repository page for 'pruebaDWEC'. The 'Code' dropdown menu is open, highlighting the 'Clone' option under the 'Local' tab. The URL 'https://github.com/avianarios/pruebaDWEC.' is displayed, along with options for 'HTTPS', 'SSH', and 'GitHub CLI'. The repository has 2 branches and 0 tags. The main branch is 'main'. The repository description is 'Pruebas para manejar git en DWEC'. The 'About' section shows 0 stars, 1 watching, and 0 forks. The 'Releases' section indicates no releases published. The 'Languages' section shows HTML at 97.2% and JavaScript at 2.8%. A note at the bottom says 'How to get HTTPS URL, needed to clone a remote repository'.

avianarios / pruebaDWEC

Type ⌘ to search

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

pruebaDWEC Public

Pin Unwatch 1 Fork 0 Star 0

main 2 branches 0 tags

Your main branch isn't protected

Protect this branch from force pushing or deletion, or require status checks before

alejandro añadiendo

js primer commit

README.md primer commit

a añadiendo

b añadiendo

c añadiendo

index.html añadido un párrafo

25 minutes ago

25 minutes ago

2 days ago

README.md

repository de pruebas

Go to file Add file ▾ Code

Local Codespaces (New)

Clone

HTTPS SSH GitHub CLI

https://github.com/avianarios/pruebaDWEC.

Use Git or checkout with SVN using the web URL.

Download ZIP

About

Pruebas para manejar git en DWEC

Readme

Activity

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Languages

HTML 97.2% JavaScript 2.8%

How to get HTTPS URL, needed to clone a remote repository

Identifying

2.- Introduction to client-side programming > Programming tools > Version control system

The screenshot shows the GitHub Developer Settings interface for managing personal access tokens.

Top Left: Personal access tokens (classic) screen. A note says "Some of the scopes you've selected are included in other scopes. Only the minimum set of necessary scopes has been saved." A red error message box says "Note can't be blank".

Bottom Left: New personal access token (classic) creation form. It includes a "Note" field (set to "terminal"), a "Note can't be blank" error message, and a "30 days" expiration dropdown (set to Fri, Oct 18 2024).

Bottom Center: Select scopes section. It lists scopes under three categories: repo, workflow, and write:packages. Under repo, checked scopes include repo, repo:status, repo_deployment, public_repo, repo:invite, and security_events. Under workflow, unchecked scope is workflow. Under write:packages, unchecked scope is write:packages.

Bottom Right: Personal access tokens (classic) list. It shows a single token named "visual studio" with scopes delete:packages, repo, workflow, write:packages. It was last used within the last 10 months and expires on Mon, Oct 7 2024. A "Delete" button is visible next to the token entry.

Bottom Footer: How to create a token, needed to login by using HTTPS

Basic workflow

2.- Introduction to client-side programming > Programming tools > Version control system

```
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DWEC/proyecto$ git init
ayuda: Usando 'master' como el nombre de la rama inicial. Este nombre de rama predeterminado
ayuda: está sujeto a cambios. Para configurar el nombre de la rama inicial para usar en todos
ayuda: de sus nuevos repositorios, reprimiendo esta advertencia, llama a:
ayuda:
ayuda: git config --global init.defaultBranch <nombre>
ayuda:
ayuda: Los nombres comúnmente elegidos en lugar de 'master' son 'main', 'trunk' y
ayuda: 'development'. Se puede cambiar el nombre de la rama recién creada mediante este comando:
ayuda:
ayuda: git branch -m <nombre>
Inicializado repositorio Git vacío en /home/alejandro/Documentos/Trabajo/Instituto/DAW/DWEC/proyecto/.git/
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DWEC/proyecto$ echo "#primer fichero" >> README.md
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DWEC/proyecto$ git add .
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DWEC/proyecto$ git commit -m "primer commit"
[master (commit-raíz) 0cdafb9] primer commit
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DWEC/proyecto$ git branch -M principal
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DWEC/proyecto$ git status
En la rama principal
nada para hacer commit, el árbol de trabajo está limpio
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DWEC/proyecto$ git remote add origen https://github.com/avianarios/proyecto_prueba.git
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DWEC/proyecto$ git push -u origen principal
Username for 'https://github.com': avianarios
Password for 'https://avianarios@github.com':
Enumerando objetos: 3, listo.
Contando objetos: 100% (3/3), listo.
Escribiendo objetos: 100% (3/3), 231 bytes | 231.00 KiB/s, listo.
Total 3 (delta 0), reusados 0 (delta 0), pack-reusados 0
To https://github.com/avianarios/proyecto_prueba.git
 * [new branch]      principal -> principal
Rama 'principal' configurada para hacer seguimiento a la rama remota 'principal' de 'origen'.
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DWEC/proyecto$
```

1. Initialize local repository
2. Create a local file
3. Add changes to stage
4. Add changes to local repository
5. Rename “master” branch to “principal” (optional)
6. Add a remote origin as “origen”
7. Add changes to remote repository
(use username and token)

Option 1: Create a remote repository and link it to an existing local directory

Storing credentials

2.- Introduction to client-side programming > Programming tools > Version control system

```
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DWEC/proyecto$ git config --global credential.helper store
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DWEC/proyecto$ touch fich7
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DWEC/proyecto$ git add .
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DWEC/proyecto$ git commit -m "7"
[principal 6f2d0b0] 7
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 fich7
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DWEC/proyecto$ git push
Username for 'https://github.com': avianarios
Password for 'https://avianarios@github.com':
Enumerando objetos: 3, listo.
Contando objetos: 100% (3/3), listo.
Compresión delta usando hasta 8 hilos
Comprimiendo objetos: 100% (2/2), listo.
Escribiendo objetos: 100% (2/2), 222 bytes | 222.00 KiB/s, listo.
Total 2 (delta 1), reusados 0 (delta 0), pack-reusados 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/avianarios/proyecto_prueba.git
  caca82d..6f2d0b0 principal -> principal
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DWEC/proyecto$ touch fich8
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DWEC/proyecto$ git add .
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DWEC/proyecto$ git commit -m "8"
[principal 1f64729] 8
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 fich8
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DWEC/proyecto$ git push
Enumerando objetos: 3, listo.
Contando objetos: 100% (3/3), listo.
Compresión delta usando hasta 8 hilos
Comprimiendo objetos: 100% (2/2), listo.
Escribiendo objetos: 100% (2/2), 222 bytes | 222.00 KiB/s, listo.
Total 2 (delta 1), reusados 0 (delta 0), pack-reusados 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/avianarios/proyecto_prueba.git
  6f2d0b0..1f64729 principal -> principal
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DWEC/proyecto$
```

HTTPS is the basic way to authenticate. In order to avoid git asking credentials every time, they can be saved in a plain text (not very safe!) by

using git config --global credential.helper store

Cloning an already existent github project

2.- Introduction to client-side programming > Programming tools > Version control system

```
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DWEC$ git clone https://github.com/avianarios/proyecto_prueba.git
Clonando en 'proyecto_prueba'...
remote: Enumerating objects: 32, done.
remote: Counting objects: 100% (32/32), done.
remote: Compressing objects: 100% (22/22), done.
remote: Total 32 (delta 11), reused 15 (delta 3), pack-reused 0 (from 0)
Recibiendo objetos: 100% (32/32), 5.08 KiB | 5.08 MiB/s, listo.
Resolviendo deltas: 100% (11/11), listo.
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DWEC$ cd proyecto_prueba/
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DWEC/proyecto_prueba$ ls
a b c d fich1 fich2 fich3 fich4 fich5 fich6 fich7 fich8 README.md
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DWEC/proyecto_prueba$ touch fich9
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DWEC/proyecto_prueba$ git status
En la rama principal
Tu rama está actualizada con 'origin/principal'.

Archivos sin seguimiento:
  (usa "git add <archivo>..." para incluirlo a lo que será confirmado)
    fich9

no hay nada agregado al commit pero hay archivos sin seguimiento presentes (usa "git add" para hacerles seguimiento)
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DWEC/proyecto_prueba$ git add fich9
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DWEC/proyecto_prueba$ git commit -m "otro fichero más"
[principal 6d24b14] otro fichero más
 1 file changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 fich9
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DWEC/proyecto_prueba$ git push
Enumerando objetos: 3, listo.
Contando objetos: 100% (3/3), listo.
Comprimiendo objetos: 100% (2/2), listo.
Escribiendo objetos: 100% (2/2), 260 bytes | 260.00 KiB/s, listo.
Total 2 (delta 1), reusados 0 (delta 0), pack-reusados 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/avianarios/proyecto_prueba.git
  515b54f..6d24b14  principal -> principal
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DWEC/proyecto_prueba$
```

Option 2: clone a remote repository into an existing local directory

Basic workflow

2.- Introduction to client-side programming > Programming tools > Version control system

```
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DWEC/proyecto$ touch fich2
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DWEC/proyecto$ git add fich2
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DWEC/proyecto$ git commit -m "segundo fichero"
[principal 1202057] segundo fichero
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 fich2
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DWEC/proyecto$ git push
Username for 'https://github.com': avianarios
Password for 'https://avianarios@github.com':
Enumerando objetos: 3, listo.
Contando objetos: 100% (3/3), listo.
Compresión delta usando hasta 8 hilos
Comprimiendo objetos: 100% (2/2), listo.
Escribiendo objetos: 100% (2/2), 272 bytes | 272.00 KiB/s, listo.
Total 2 (delta 0), reusados 0 (delta 0), pack-reusados 0
To https://github.com/avianarios/proyecto_prueba.git
 811a552..1202057  principal -> principal
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DWEC/proyecto$ █
```

First time push is done a relationship between local branch “principal” and remote branch has to be established. That’s why git push -u “remote_server” “local_branch” was used.

Once established, you can just use git push

Getting remote changes

2.- Introduction to client-side programming > Programming tools > Version control system

```
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DWEC/proyecto$ git fetch
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
Desempaquetando objetos: 100% (3/3), 907 bytes | 907.00 KiB/s, listo.
Desde https://github.com/avianarios/proyecto_prueba
  6382930..515b54f principal -> origen/principal
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DWEC/proyecto$ git merge
Actualizando 6382930..515b54f
Fast-forward
 d | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 d
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DWEC/proyecto$
```

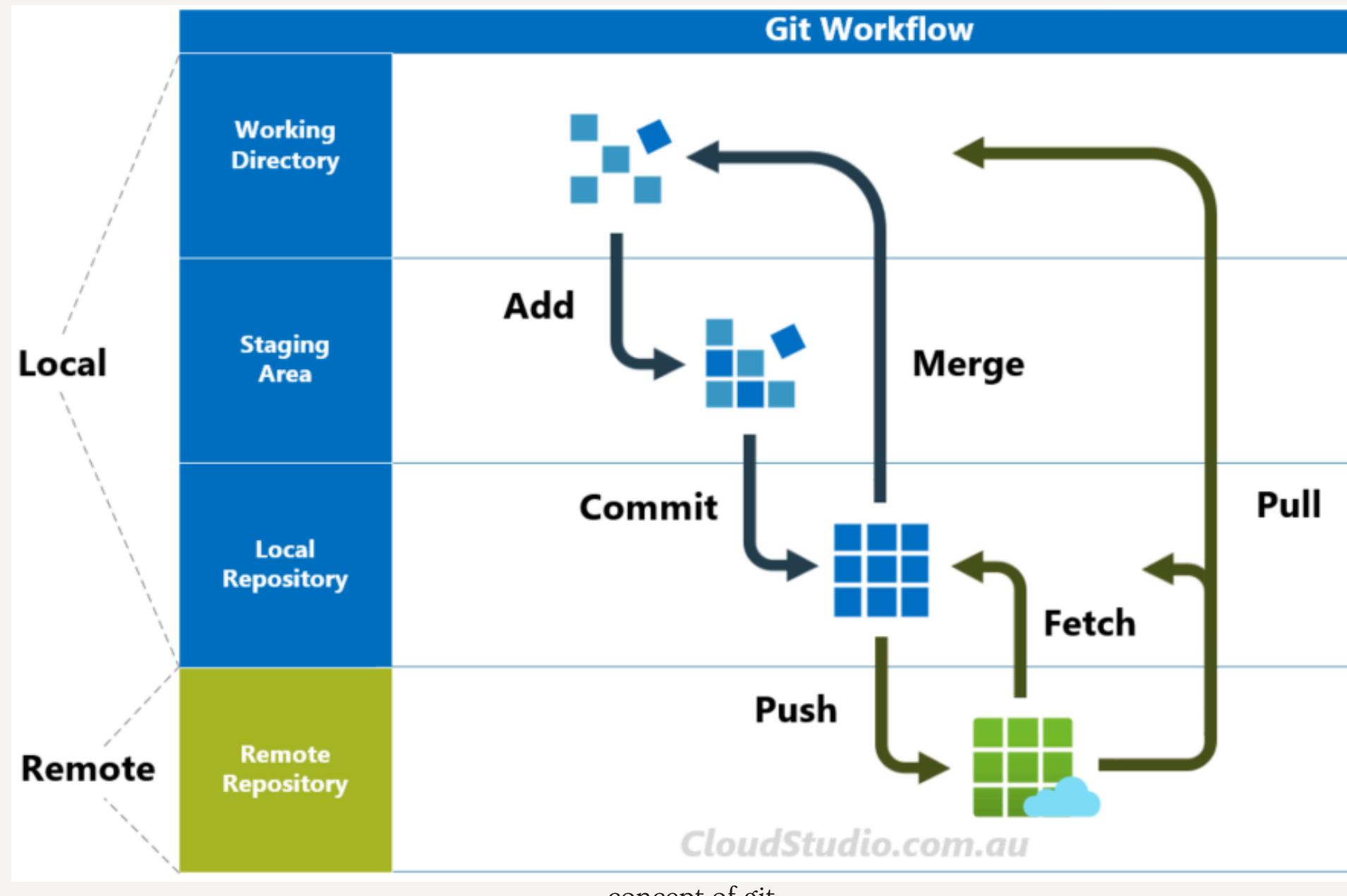
git fetch bring information about what has changed in remote server but without changing your local repository

git merge bring the remote changes and stores in your local repository

git pull is equivalent to git fetch+git merge

Version control system

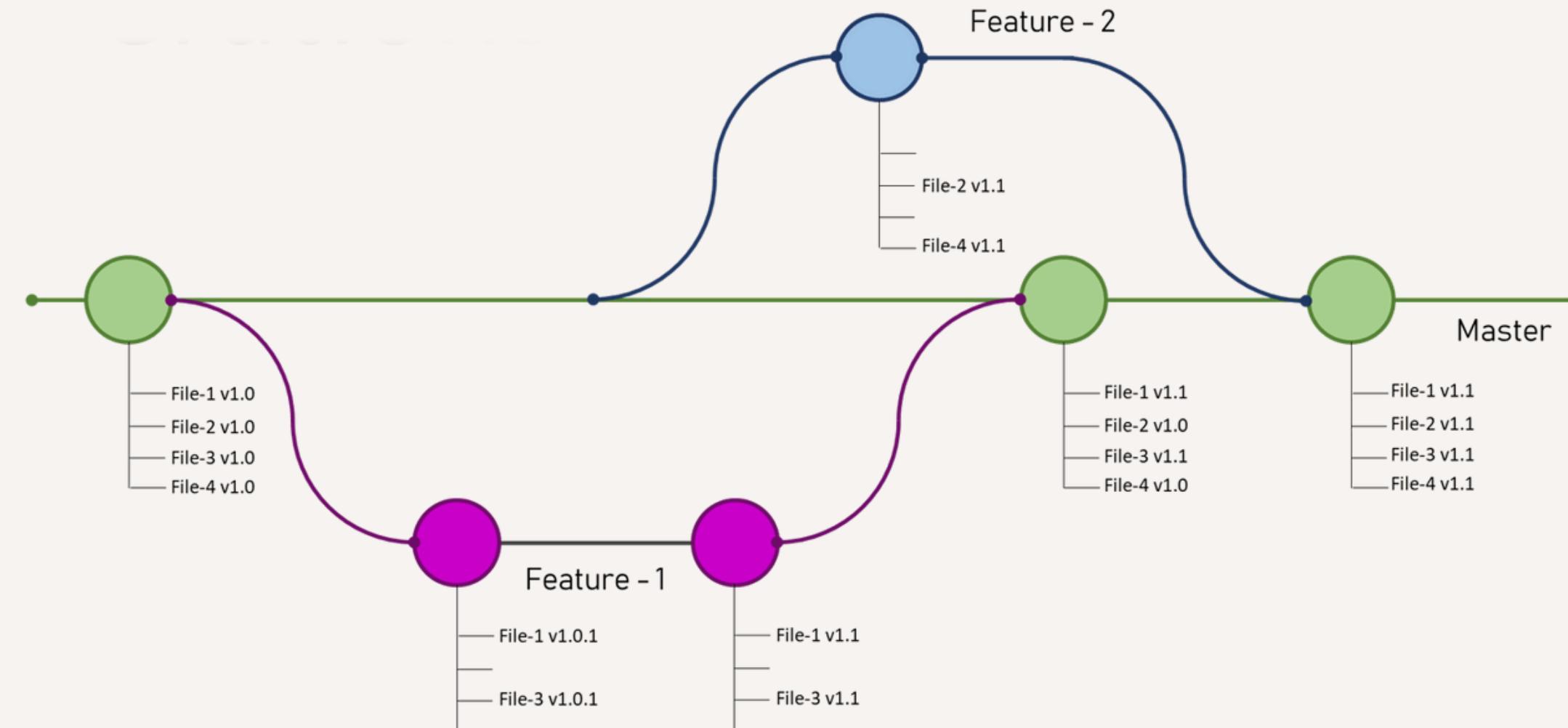
2.- Introduction to client-side programming > Programming tools



Branches

2.- Introduction to client-side programming > Programming tools > Version control system

- Master branch should contain only well-tested, working code.
- A replica of master must be taken when programmer wants to make changes and improvements.
- Once finished and tested, branch is merged with master.



Version control system

2.- Introduction to client-side programming > Programming tools

```
alejandro@alejandro-DMAF5:~/Documentos/Trabajo/Instituto/DWEC/pruebaDWEC$ git branch -vv
* main 602f1c3 [origin/main] crear ficheros
alejandro@alejandro-DMAF5:~/Documentos/Trabajo/Instituto/DWEC/pruebaDWEC$ git branch ramal
alejandro@alejandro-DMAF5:~/Documentos/Trabajo/Instituto/DWEC/pruebaDWEC$ git checkout ramal
Cambiado a rama 'ramal'
alejandro@alejandro-DMAF5:~/Documentos/Trabajo/Instituto/DWEC/pruebaDWEC$ touch a b c
alejandro@alejandro-DMAF5:~/Documentos/Trabajo/Instituto/DWEC/pruebaDWEC$ git add .
alejandro@alejandro-DMAF5:~/Documentos/Trabajo/Instituto/DWEC/pruebaDWEC$ git commit -m "nuevos ficheros"
En la rama ramal
nada para hacer commit, el árbol de trabajo está limpio
alejandro@alejandro-DMAF5:~/Documentos/Trabajo/Instituto/DWEC/pruebaDWEC$ git push
fatal: La rama actual ramal no tiene una rama upstream.
Para realizar un push de la rama actual y configurar el remoto como upstream, usa
    git push --set-upstream origin ramal

alejandro@alejandro-DMAF5:~/Documentos/Trabajo/Instituto/DWEC/pruebaDWEC$ git branch -vv
  main 602f1c3 [origin/main] crear ficheros
* ramal 602f1c3 crear ficheros
alejandro@alejandro-DMAF5:~/Documentos/Trabajo/Instituto/DWEC/pruebaDWEC$ git push -u origin ramal
Username for 'https://github.com': avianarios
Password for 'https://avianarios@github.com':
Rama 'ramal' configurada para hacer seguimiento a la rama remota 'ramal' de 'origin'.
Everything up-to-date
alejandro@alejandro-DMAF5:~/Documentos/Trabajo/Instituto/DWEC/pruebaDWEC$ git branch -vv
  main 602f1c3 [origin/main] crear ficheros
* ramal 813f605 [origin/ramal] primer fich
alejandro@alejandro-DMAF5:~/Documentos/Trabajo/Instituto/DWEC/pruebaDWEC$
```

Creating first branches

- There are two concepts related with remote repository
 - remote server pull is going to take files from (origin)
 - remote repository push is going to upload to (upstream)
- When creating local branches, there is no origin nor upstream so they must be set at first push (git push --set-upstream origin ramal or git push -u origin ramal)

Version control system

2.- Introduction to client-side programming > Programming tools

The screenshot shows a terminal window with a graphical interface above it. The graphical interface displays a commit history with three branches: rama2, rama1, and master. The master branch has a commit for 'tercer fichero' and another for 'primera subida de ficheros'. The rama1 branch has a commit for 'f1'. The rama2 branch has a commit for 'f2'. Below the interface is a terminal window with the following command history:

```
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
● alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DIW/codigo_DIW/codigo-prueba$ git branch rama2
● alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DIW/codigo_DIW/codigo-prueba$ git checkout rama2
Cambiado a rama 'rama2'.
● alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DIW/codigo_DIW/codigo-prueba$ echo "f2" > f2
● alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DIW/codigo_DIW/codigo-prueba$ git add .
● alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DIW/codigo_DIW/codigo-prueba$ git commit -m "f2"
[rama2 19fe640] f2
 1 file changed, 1 insertion(+)
 create mode 100644 f2
● alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DIW/codigo_DIW/codigo-prueba$ git push -u origin rama2
Enumerando objetos: 4, listo.
Contando objetos: 100% (4/4), listo.
Compresión delta usando hasta 8 hilos
Comprimiendo objetos: 100% (2/2), listo.
Escribiendo objetos: 100% (3/3), 256 bytes | 256.00 KiB/s, listo.
Total 3 (delta 1), reusados 0 (delta 0), pack-reusados 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'rama2' on GitHub by visiting:
remote:     https://github.com/avianarios/codigo-prueba/pull/new/rama2
remote:
To https://github.com/avianarios/codigo-prueba.git
 * [new branch]      rama2 -> rama2
Rama 'rama2' configurada para hacer seguimiento a la rama remota 'rama2' de 'origin'.
● alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DIW/codigo_DIW/codigo-prueba$ git branch -a
  master
  rama1
* rama2
  remotes/origin/HEAD -> origin/master
  remotes/origin/master
  remotes/origin/rama1
  remotes/origin/rama2
○ alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DIW/codigo_DIW/codigo-prueba$
```

Branch creation

Reverting changes

2.- Introduction to client-side programming > Programming tools

```
alejandro@alejandro-DMAF5:~/Documentos/Trabajo/Instituto/DWEC/pruebaDWEC$ git checkout ramal
Cambiado a rama 'ramal'
Tu rama está actualizada con 'origin/ramal'.
alejandro@alejandro-DMAF5:~/Documentos/Trabajo/Instituto/DWEC/pruebaDWEC$ git log
commit 813f605558624398e355531fc1f53ab2cf5b8aaa (HEAD -> ramal, origin/ramal)
Author: avianarios <avri.web@protonmail.com>
Date:   Sat Sep 23 17:47:53 2023 +0200

    primer fich

commit 602f1c3e1769bc61ef6388d59d71f81b589b71be (origin/main, origin/HEAD, main)
Author: avianarios <avri.web@protonmail.com>
Date:   Sat Sep 23 17:42:07 2023 +0200

    crear ficheros

commit 9082595c36callc8cdcb76980a9b385c11ff441b
Author: alejandro <alejandro@micorreo.com>
Date:   Sat Sep 23 17:07:08 2023 +0200

    añadiendo

alejandro@alejandro-DMAF5:~/Documentos/Trabajo/Instituto/DWEC/pruebaDWEC$ git show 813f60
commit 813f605558624398e355531fc1f53ab2cf5b8aaa (HEAD -> ramal, upstream/ramal, origin/ramal)
Author: avianarios <avr.web@protonmail.com>
Date:   Sat Sep 23 17:47:53 2023 +0200

    primer fich

diff --git a/a b/a
index e69de29..5c1b149 100644
--- a/a
+++ b/a
@@ -0,0 +1 @@
+hola
alejandro@alejandro-DMAF5:~/Documentos/Trabajo/Instituto/DWEC/pruebaDWEC$
```

Reverting changes

- There is log storing all commits
 - git log -> shows actual branch info
 - git log --all -> shows all branches info
- each commit has an ID

Reverting changes

2.- Introduction to client-side programming > Programming tools > Version control system

```
alejandro@alejandro-DMAF5:~/Documentos/Trabajo/Instituto/DWEC/pruebaDWEC$ git revert 602f1
[ ramal 8989249] Revert "crear ficheros"
 3 files changed, 0 insertions(+), 0 deletions(-)
 delete mode 100644 f1
 delete mode 100644 f2
 delete mode 100644 f3
alejandro@alejandro-DMAF5:~/Documentos/Trabajo/Instituto/DWEC/pruebaDWEC$ git push
Username for 'https://github.com': avianarios
Password for 'https://avianarios@github.com':
Enumerando objetos: 3, listo.
Contando objetos: 100% (3/3), listo.
Compresión delta usando hasta 8 hilos
Comprimiendo objetos: 100% (1/1), listo.
Escribiendo objetos: 100% (2/2), 273 bytes | 273.00 KiB/s, listo.
Total 2 (delta 1), reusados 1 (delta 1), pack-reusados 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/avianarios/pruebaDWEC.git
  bb1338f..8989249  ramal -> ramal
alejandro@alejandro-DMAF5:~/Documentos/Trabajo/Instituto/DWEC/pruebaDWEC$
```

- git revert only needs the firsts ID characters
- git push is needed as changes are performed only in local repository

Creating and removing branches

2.- Introduction to client-side programming > Programming tools > Version control system

```
alejandro@alejandro-DMAF5:~/Documentos/Trabajo/Instituto/DWEC/pruebaDWEC$ git branch -a
  main
* ramal
  rev1
  remotes/origin/HEAD -> origin/main
  remotes/origin/main
  remotes/origin/ramal
  remotes/origin/rev1
alejandro@alejandro-DMAF5:~/Documentos/Trabajo/Instituto/DWEC/pruebaDWEC$ git branch -d rev1
Eliminada la rama rev1 (era 7dc5e3d).
alejandro@alejandro-DMAF5:~/Documentos/Trabajo/Instituto/DWEC/pruebaDWEC$ git branch -a
  main
* ramal
  remotes/origin/HEAD -> origin/main
  remotes/origin/main
  remotes/origin/ramal
  remotes/origin/rev1
alejandro@alejandro-DMAF5:~/Documentos/Trabajo/Instituto/DWEC/pruebaDWEC$ git push origin --delete rev1
To https://github.com/avianarios/pruebaDWEC.git
 - [deleted]          rev1
alejandro@alejandro-DMAF5:~/Documentos/Trabajo/Instituto/DWEC/pruebaDWEC$ git branch -a
  main
* ramal
  remotes/origin/HEAD -> origin/main
  remotes/origin/main
  remotes/origin/ramal
alejandro@alejandro-DMAF5:~/Documentos/Trabajo/Instituto/DWEC/pruebaDWEC$
```

How to remove branches, locally and remotely

-a lists remote and local branches

-d removes locally. Push must be done in order to remove remotely

Merging branches

2.- Introduction to client-side programming > Programming tools > Version control system

The screenshot shows a Git Graph interface with a timeline of commits. The main pane displays a commit from 'main' branch titled 'Merge branch 'voy a mezclar la rama 1''. This commit includes three merge commits from 'rama1' branch: 'primeros fich', 'crear ficheros', and 'Displaying all changes from 602f1c3e1769bc61ef6388d59d71f81b589b71be to 3379f128cf1fecd905710a003d0e12136f947597.'. Below this, there are three additional commits: 'añadiendo', 'añadido el fichero a', and 'añadido un párrafo'. The bottom part of the interface shows a terminal window with the command 'git status' and the output:

```
alejandro@alejandro-DMAF5:~/Documentos/Trabajo/Instituto/DWEC/pruebaDWEC$ git status
En la rama main
Tu rama está actualizada con 'origin/main'.

nada para hacer commit, el árbol de trabajo está limpio
● alejandro@alejandro-DMAF5:~/Documentos/Trabajo/Instituto/DWEC/pruebaDWEC$ git merge ramal
Auto-fusionando a
Merge made by the 'ort' strategy.
 a | 1 +
 1 file changed, 1 insertion(+)
○ alejandro@alejandro-DMAF5:~/Documentos/Trabajo/Instituto/DWEC/pruebaDWEC$
```

git merge ramal incorporates changes from the branch "rama1" into the current one

Conflict resolution

2.- Introduction to client-side programming > Programming tools > Version control system

```
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DIW/codigo_DIW/codigo-prueba$ echo "f2 ramal" > f2
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DIW/codigo_DIW/codigo-prueba$ git add .
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DIW/codigo_DIW/codigo-prueba$ git commit -m "f2 ramal"
[ramal a80f00e] f2 ramal
 1 file changed, 1 insertion(+)
 create mode 100644 f2
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DIW/codigo_DIW/codigo-prueba$ git push
Enumerando objetos: 4, listo.
Contando objetos: 100% (4/4), listo.
Compresión delta usando hasta 8 hilos
Comprimiendo objetos: 100% (2/2), listo.
Escribiendo objetos: 100% (3/3), 266 bytes | 266.00 KiB/s, listo.
Total 3 (delta 0), reusados 0 (delta 0), pack-reusados 0
To https://github.com/avianarios/codigo-prueba.git
 65f39cf..a80f00e ramal -> ramal
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DIW/codigo_DIW/codigo-prueba$ git checkout main
Cambiado a rama 'main'
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DIW/codigo_DIW/codigo-prueba$ echo "f2 main" > f2
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DIW/codigo_DIW/codigo-prueba$ git add .
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DIW/codigo_DIW/codigo-prueba$ git commit -m "f2 main"
[main f7bd73c] f2 main
 1 file changed, 1 insertion(+)
 create mode 100644 f2
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DIW/codigo_DIW/codigo-prueba$ git push
Enumerando objetos: 4, listo.
Contando objetos: 100% (4/4), listo.
Compresión delta usando hasta 8 hilos
Comprimiendo objetos: 100% (2/2), listo.
Escribiendo objetos: 100% (3/3), 267 bytes | 267.00 KiB/s, listo.
Total 3 (delta 0), reusados 0 (delta 0), pack-reusados 0
To https://github.com/avianarios/codigo-prueba.git
 870bd62..f7bd73c main -> main
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DIW/codigo_DIW/codigo-prueba$ git merge ramal
Auto-fusionando f2
CONFLICTO (agregar/agregar): Conflicto de fusión en f2
Fusión automática falló; arregle los conflictos y luego realice un commit con el resultado.
alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DIW/codigo_DIW/codigo-prueba$
```

merging can result in conflicts when the same file has different content at different branches

Conflict resolution

2.- Introduction to client-side programming > Programming tools > Version control system

The screenshot shows a Git merge interface with three main panes:

- Top Left Pane:** Displays a diff view between the current branch (HEAD) and a pull request branch (rama1). It highlights changes in the file f2. A context menu at the top offers options: 'Aceptar cambio actual', 'Aceptar cambio entrante', 'Aceptar ambos cambios', and 'Comparar cambios'. The commit history shows 'f2 main' and 'f2 rama1'.
- Top Right Pane:** Shows the current state of the repository. It lists 'Entrante' (remote origin) and 'Actual' (local main branch). Both branches have their latest commits ('f2 rama1' and 'f2 main') highlighted.
- Bottom Center Pane:** Shows the result of the merge. It displays a message: 'No se acepta ningún cambio' (No change is accepted). A button labeled 'Resolver en el Editor de combina' (Resolve in the Merge Editor) is visible. Below this, a 'Completar la fusión mediante combinación' (Complete the merge via combination) button is shown.
- Bottom Right Terminal:** A terminal window titled 'nano - codigo-prueba' showing the same diff output as the top-left pane. The terminal title bar also shows 'f2'.

conflicts can be solved by opening the file with nano or VS code and selecting information to be kept

Checking differences between branches

2.- Introduction to client-side programming > Programming tools > Version control system

Graph	Description	Date	Author	Commit
	o main Merge branch 'rama1' o origin/main f2 main o rama1 origin f2 rama1 f1 rama1 f1 master	22 Sep 2024 20:35 22 Sep 2024 20:28 22 Sep 2024 20:27 22 Sep 2024 20:27 22 Sep 2024 20:26	avianarios avianarios avianarios avianarios avianarios	0929c988 f7bd73c6 a80f00ea 65f39cf4 870bd620

PROBLEMASSALIDACONSOLA DE DEPURACIÓNTERMINALPUERTOS

bash - codigo-prueba

```
○ alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DIW/codigo_DIW/codigo-prueba$  
● alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DIW/codigo_DIW/codigo-prueba$ git diff ramal main  
diff --git a/f2 b/f2  
index 89e0ca0..349324d 100644  
--- a/f2  
+++ b/f2  
@@ -1 +1,2 @@  
+f2 main  
 f2 ramal  
○ alejandro@minisforum-DMAF5:~/Documentos/Trabajo/Instituto/DAW/DIW/codigo_DIW/codigo-prueba$
```

More actions

2.- Introduction to client-side programming > Programming tools > Version control system

It can do a lot more...

- Rebase (similar to merge but makes the commit tree linear) -> git rebase rama_a_rebasar
- Detach HEAD -> git checkout commit_id
- Move along commit tree -> git checkout HEAD^ or git checkout HEAD~3
- Reassign a branch to a commit -> git branch -f main HEAD~3 (with a detached HEAD)
- Incorporate some commits to current branch -> git cherry-pick commit_id1 commit_idn
- Interactive rebase (-i option. Avoids having to know commits id) -> git rebase -i HEAD~4

3.- Introduction to JavaScript

1. What is JavaScript?
2. What in-browser JavaScript can and can't do?
3. JavaScript versions
4. JavaScript engines
5. JavaScript-based languages

What is Javascript?

3.- Introduction to JavaScript

- Interpreted high-level Programming language used to provide interaction to websites
- Alongside HTML and CSS, it's the third technology absolutely essential in website creation
- Traditionally client-side but, thanks to Node, also server-side

```
const para = document.querySelector("p");

para.addEventListener("click", updateName);

function updateName() {
  const name = prompt("Enter a new name");
  para.textContent = `Player 1: ${name}`;
}
```

a Javascript code

```
1 const colors = ["green", "red", "rgba(133,122,200)", "#f15025"];
2 const btn = document.getElementById("btn");
3 const color = document.querySelector(".color");
4
5 btn.addEventListener("click", function () {
6   const randomNumber = getRandomNumber();
7   // console.log(randomNumber);
8
9   document.body.style.backgroundColor = colors[randomNumber];
10  color.textContent = colors[randomNumber];
11});
12
13 function getRandomNumber() {
14   return Math.floor(Math.random() * colors.length);
15 }
```

a Javascript code

What in-browser Javascript can and can't do?

3.- Introduction to JavaScript

What can be done?

- Modify an HTML document. Anything
- React to user actions
- Interact with remote servers
- Running code in response to events
- Work with cookies
- Work with browsers' APIs: DOM, Geolocation, Audio, Video, Canvas (2D graphics) and WebGL (render 2D and 3D graphics).
- Work with third parties' APIs

What can't be done?

- Interact with other tabs or windows
- Communicate with other servers different where the page came from without permission
- Read or write files on hard disk without permission
- Execute programs
- Access to camera or microphone without permission

JavaScript versions

3.- Introduction to JavaScript

- Netscape, creator of JavaScript, decides to standardise it by sending it to ECMA (European Computer Manufacturers Association)
- There has been lots of versions:
 - ECMAScript 2015 or ES6
 - ECMAScript 2017 or ES7
 - ...
 - Last one is ECMAScript 2023 or ES14
- Some of them only bring minor improvements so, for us, it is the same working on ES6 or above

JavaScript engine

3.- Introduction to JavaScript

- Is a program that executes JavaScript code
- First ones were interpreters. Nowadays they use just-in-time compilation for improved efficiency
- Typically developed for web browsers (but not always)
- Most popular:
 - V8 (Google). Used in Google Chrome, Chromium-based and Edge browsers, as well as Node.js and Deno
 - SpiderMonkey (Mozilla), used in Firefox and its forks
 - JavaScriptCore (Apple), used in its Safari browser
 - Chakra (Microsoft), used in Internet Explorer browser

JavaScript-based languages

3.- Introduction to JavaScript

Other languages

- JavaScript syntax does not meets everybody's needs
- Some languages, similar to JavaScript, have appeared : CoffeScript, TypeScript, etc.
- They have to be translated (transpiled) into JavaScript for the browser to execute it.

TypeScript	JavaScript
<pre>class Greeter { greeting: string; constructor (message: string) { this.greeting = message; } greet() { return "Hello, " + this.greeting; } }</pre>	<pre>var Greeter = (function () { function Greeter(message) { this.greeting = message; } Greeter.prototype.greet = function () { return "Hello, " + this.greeting; }; return Greeter; })();</pre>