Intro

2 paradigmas de la programación:

Imperativo o por procedimientos. Es el mas viejo. Se escribe paso a paso.

Declarativo. Se centra en el qué, en lugar del cómo.

Funcional (arma bloques de código o funciones. Permite la reutilización del código) Vs Procedimentales (Según el muchacho no sería los mismo que imperativo, aunque lo asocia).

Orientado a objetos.

División por niveles

Cuanto mas bajo es el nivel, mas podría tocarse el hardware.

El lenguaje máquina es el mas bajo. Los microchips tienen transistores que tienen 2 estados: apagado o encendido, 0 y 1.

Lenguaje ensamblador. Difícil de entender.

A estos 2 se los conoce como los de bajo nivel.

C y C++ son de medio-bajo nivel.

De medio-alto: java, php, javascript, C#, Python.

Alto nivel: Frameworks. Se basan en los de medio o alto nivel. Son funcionalidades extras de los lenguajes ya conocidos, para que programar con esos lenguajes sea aún mas fácil y tengan mas características.

Proceso de conversión

De nuestro lenguaje hasta los 0 y 1 del lenguaje máquina.

Lenguajes de programación compilados e interpretados.

Compilar, transforma nuestro código y construye un archivo lleno de 0s y 1s. Interprete usan JavaScript y Python.

Que sucede si al compilar hay algún error, no se ejecuta. Luego, los archivos compilados se interpretan.

Los lenguajes interpretados, no necesitan de la compilación. En la salida aparecen los mensajes de error. Es decir, se ejecuta mas allá de que haya errores.

IDE

Entorno de desarrollo integrado.

Podría escribirse código en un bloc de notas.

Los IDE facilitan la escritura y lectura del código. Nombra al visual studio code como el mas utilizado.

Control de versiones

Creas carpetitas y copias ahí los archivos es la de k-pos.

Pero, si tenés que trabajar con otros se complica.

Podría ser con Google drive o Dropbox. Tampoco es tan efectiva.

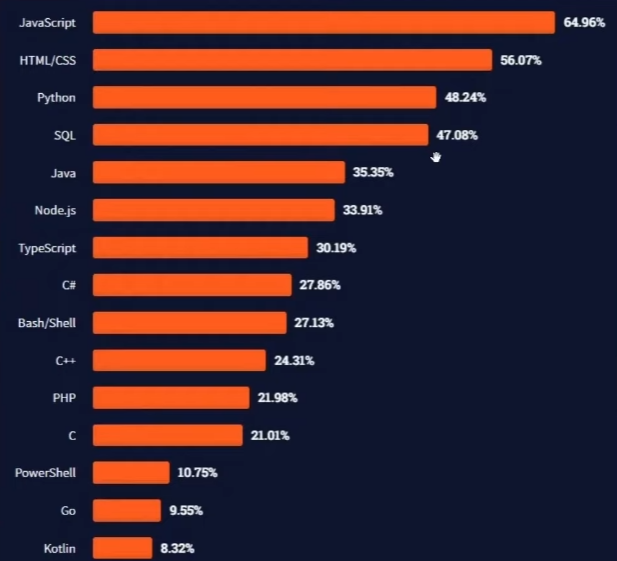
A ppios de los años 2000 nace GIT. Antes estaban CVS, SVN.

GitHub y GitLab son 2 plataformas que lo utilizan. Se ve el día en que se hacen los cambios. GitLab se centra mas en repositorios privados.

**Lenguajes de programación**

* Python es el mas utilizado actualmente. Es fácil de utilizar y aprender.
* Django y Flask son frameworks para el desarrollo web. El primero es mas utilizado.
* Java. Orientado a objetos. Lo usan mucho las grandes corporaciones. Se usó y aún se usa mucho en Android. Es relativamente sencillo.
* JavaScript. Es el mas usado para aplicaciones y páginas web. Es mas fácil que java. Framework NodeJs. Es el lenguaje mas adecuado para startups modernas, por su versatilidad.
* TypeScript. Nace basado en JavaScript, es de Microsoft. Es un lenguaje tipado. Es por esto, mas complejo que JavaScript.
* C#. Es muy versátil. Se usa en back end, así como en videojuegos y apps de microsoft. Es derivado de C y C++. Dificultad moderada, algo mas que Java.
* ASP. Net. Es un framework de Microsoft, para hacer apps webs, apis, servicios. Dificultad moderada, como HTML.
* PHP. Muy usado para apps web. Compite con Javascript y Python. Fácil de aprender. Las webs suelen ser un poco mas lentas.
* HTML. Es un lenguaje de etiquetado. Sitios o apps webs estáticas. Hace al esqueleto
* CSS. Lenguaje de diseño gráfico. Le da estilo al esqueleto.
* Ajax. Complementa Javascript, es una extensión. Para crear apps web asíncronas.
* Ruby. Se utiliza para servidores. Ruby on rails es otra cosa, es un framework (twitch, twitter lo usa).
* Perl. Se usaba para procesadores de texto. Hoy para administrar sistemas, desarrollo web y de GUI.
* Dart. Similar a C. Relativamente complejo.
* Kotlin. Para hacerle competencia a Java. Es ruso y mas moderno. Puede interactuar con Java. Se usa para el desarrollo de aps móviles.
* Swift. Es el lenguaje exclusivo para IOS, Apple.

Lenguajes mas utilizados en 2022:



Programación estructurada

Se escribe paso por paso. Es una secuencia. Es lineal. Por mas que puedan introducirse ifs y arborizarse. Afirma que no funciona este paradigma, salvo para cosas muy sencillas.

Programación orientada a objetos

Aquí hay objetos, y se interrelacionan entre sí. Deja de ser algo lineal.

Tienen 2 tipos de “propiedades”: atributos y funciones, a través de métodos.

4 conceptos clave de la programación orientada a objetos:

Abstracción.

Encapsulamiento.

Herencia.

Polimorfismo.

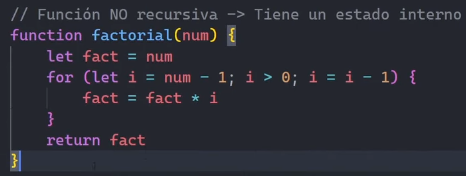
Programación funcional

Se va encapsulando código, se encapsulan funciones para poder utilizarlas luego.

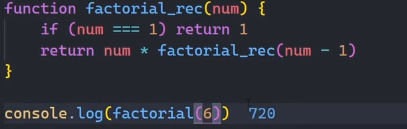
Utiliza funciones puras. Siempre funcionan igual.

Habla de funciones sin estado.

Recursividad. Cuando una función se llama a si misma.



Esto si es recursivo:



Desarrollo web: Front-End y Back-End

Front End va a lo visual. Back a todos los procesos que van detrás.

Lenguajes de servidor:



Agrega NodeJs. Framework de Javascript.

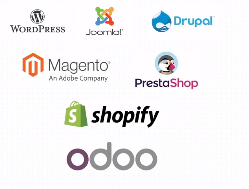
El servidor es quien se encarga de realizar la mayor parte de los cómputos.

Lenguajes desde el punto de vista del cliente.



CMS

Content management system.



Son softwares que ya lo tienen todo. Se hace todo (back y full) sin programar. Se hace todo bastante estandarizado.

Los 5 de arriba están programados en PHP y Javascript.

Shopify con HTML y Javascript. Se usa mucho para tiendas online.

Odoo con Python y Javascript. Está enfocado a CRM client relationship manager. Softwares de gestion de clientes.

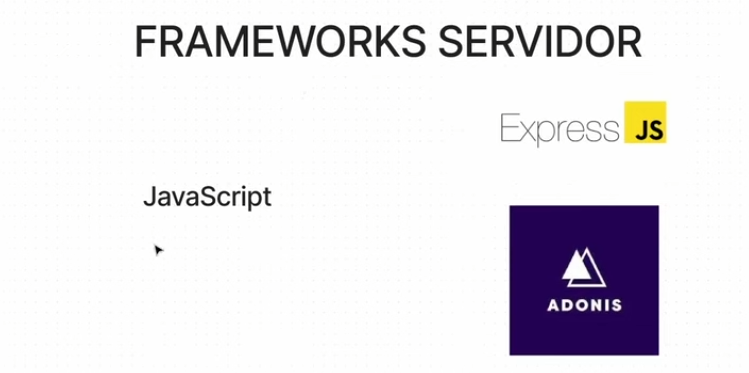
Wordpress es el mas utilizado.

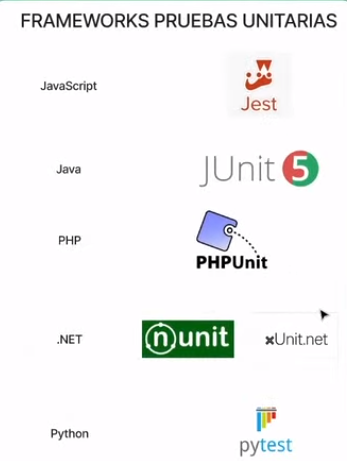
Frameworks

Librerías que facilitan creación de distintas cosas.

Angular, View y React. Son la Piedra angular de los que usan javascript por debajo.







Librerías (Punto de vista del cliente)

React y Jquery. Trabajan con el DOM. Document Object Model. Ayudan a manipular la parte visible de HTML.

Bootstrap trabaja con Css. Recomendadísimo.

Servidores

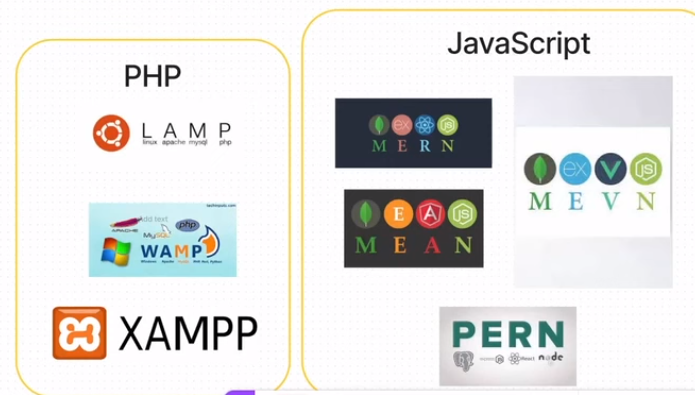
Apache es el mas utilizado. Secunda Microsoft IIS. El primero sería mas baratongo.

Sirven páginas web, FTP, correo electrónico y BBDD.



Stacks

Conjuntos de tecnologías que permiten crearlo todo.



No lo plantea como algo super estricto. Es decir, podría usarse otra BBDD que la indicada, y quizás otro lenguaje de programación.

Desarrollo móvil

Plataformas: Android e IOS.

Entornos de desarrollo. XCode para IOS; Android Studio para Android. Cada uno tiene su propio emulador de un dispositivo celular. Por eso los prefiere antes que visual studio code.

Con Kotlin y Java se codea para Android. Swift para IOS.

Frameworks y librerías:

Reitera que es mas genérico, pero mas fácil también.

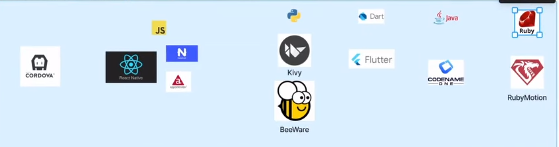
Ionic se basa en angular, aunque también usa react y view actualmente, para el desarrollo de apps webs.

Xamarin (C# con .net), Flutter (Dart) y React (Javascript con React). También sirven para IOS. Crean Código native. Ionic no, usa un emulador web.

Desarrollo multiplataforma



No se usan en si esos lenguajes, sino diferentes frameworks.



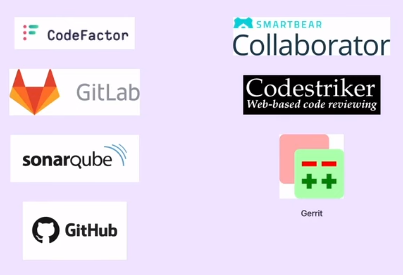
Cordova trabaja con muchos lenguajes. No es un framework en si, funciona como puente, dice. Trabaja con C#.

Code Review  
Revisión del código

Se crea un Branch, luego un Pull Request y alguien lo revisa antes de hacerse el Merge Branch. Podría aceptar o rechazar la nueva versión del código.

Se compara la nueva versión con la anterior y ver si se genera algún problema.

Hay herramientas que facilitan la revisión del código:



Muestran lo modificado y lo que no se alteró:



Lo verde está añadido, lo rojo ha sido eliminado.

Si son pequeños cambios se puede revisar solo leyendo el código. Si son cambios mas grandes se debería descargar y ejecutar.

BBDD SQL y NOSQL

SQL son relacionales. Se crean relaciones en base a claves.

No da a SQL Server como una de las mas utilizadas.

NoSQL. Bases de datos tipo Key-Value. Son muy rápidas.

También pueden almacenarse objetos o documentos.

También están los grafos. Lleno de nodos.



DevOPS y CI/CD

Viene de Development y IT Operations.

Son 2 departamentos que deben trabajar codo a codo, pero solía haber mucho conflicto.

Es una metodología de desarrollo de software, para que ambas partes puedan trabajar conjuntamente.

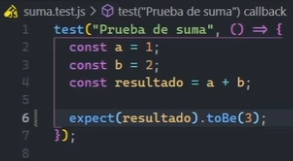
CI Continous Integration; CD Continous Deployment.

Uno hace una nueva versión de algo y antes de que la oficialice, se realiza una serie de tests. Si supera las pruebas se realiza el push (subida de código). Algo similar sucede cuando se intenta un deploy.

CI Publica el Código (Merge, Pull, Push).

CD Despliega la App para que se pueda utilizar.





Eso es un test, que también se puede automatizar. Parece bastante quilombo.

.

Agregado de la web:

