




# Programación Web

2022

## JavaScript ✨

- JavaScript es un lenguaje de programación
- ◆ Se ajusta a la especificación llamada ECMAScript
- ◆ European Computer Manufacturers Association
- ◆ ECMAScript es un plano (blueprint) para crear lenguajes de scripts
- ◆ JavaScript es una implementación de ese plano descrito en el estándar ECMA-262 

Teniendo en cuenta el concepto de lenguaje de programación como lo es JavaScript u otros.

¿Consideran que HTML es un lenguaje de programación?

# JavaScript

- Lenguaje con peculiaridades (¡como todos!)
- ◆ El scope (o contexto) define la vida útil y visibilidad de una variable
- ◆ Las variables no son visibles fuera del contexto en donde fueron declaradas
- ◆ var vs let vs const

# var vs let vs const ?

→

**var**

- ◆ Scope local para funciones
- ◆ Permite redeclaraciones
- ◆ Aplica hoisting
- ◆ Casos confusos que piden arreglo con let/const

- restrictivo

→

**let**

- ◆ Scope de bloque
- ◆ Se puede actualizar pero no redeclarar

→

**const**

- ◆ Como **let** pero no se puede actualizar ni redeclarar

+ restrictivo

Redeclaración:

```
var saludo = "buenas como va?";
var saludo = "hola a secas";
```

Hoisting:

```
console.log(saludo); //undefined
var saludo = "buenas como va?"
var saludo;
console.log(saludo); // greeter is undefined
saludo = "buenas como va?"
```

Caso confuso var:

```
var saludo = "buenas";
var veces = 4;
if (veces > 3) {
  var saludo = "hola";
}
console.log(saludo) // "hola"
```

let:

```
let saludo = "buenas";
let veces = 4;
if (veces > 3) {
  let hola = "holaa";
```

```
    console.log(hola); // "holaa"
  }
  console.log(hola) // hola is not defined
```

Actualizar no redeclarar:

```
let saludo = "buenas";
saludo = "hola";

let saludo = "buenas";
let saludo = "hola"; // error: Identifier 'saludo' has
already been declared
```

Redeclarar en diferentes contextos:

```
let saludo = "hola";
if (true) {
    let saludo = "buenas";
    console.log(saludo); // "buenas"
}
console.log(saludo); // "hola"
```

Just like `var`, `let` declarations are hoisted to the top.

Unlike `var` which is initialized as `undefined`, the `let` keyword is not initialized. So if you try to use a `let` variable before declaration, you'll get a `Reference Error`.

## Software y frameworks



Clasificación de galaxias



Más sobre los planetas FE



¿Cómo elegimos?



Exploración



Primero tenemos las galaxias en donde identificamos las siguientes:

- Escritorio: Software o aplicaciones o apps para los sistemas operativos que utilizan nuestras computadoras como Windows de Microsoft, MacOS de Apple y Linux **open source**. ¿Hoy cual están usando en sus computadoras? Estas aplicaciones pueden tener acceso a los periféricos para poder usarlos, como por ejemplo, la impresora, la webcam, etc.





- Móvil: Software para los sistemas operativos que utilizan nuestros teléfonos móviles, los celulares, como el sistema operativo iOS de Apple en celulares iPhone o tablets iPad, o el sistema operativo Android en celulares Galaxy de Samsung, Huawei de Huawei, Xiaomi de Xiaomi. Estas aplicaciones también pueden tener acceso a lo que ofrece un teléfono celular, como la cámara, el acelerómetro, el GPS, etc.

## Clasificación de galaxias

PW2022



Escritorio



Web



Móvil

- Web: Software para los navegadores que funcionan en todos los sistemas operativos mencionados, tanto en escritorio como de móvil.



Este es el universo del software que vamos a hablar hoy.

Hay más galaxias y lugares nuevos todavía por descubrir, y eso es algo que nunca va a cambiar en esta industria, y es lo bueno y lo malo al mismo tiempo.

Lo malo es que si uno no supera la idea de que uno solo sabe una fracción del universo, ni tiene mucha idea de cómo y qué significa moverse entre las galaxias, puede ser muy abrumador. Incluso si uno elige una galaxia y se siente cómodo en cierto momento trabajando ahí, como las tecnologías y las necesidades de los clientes evolucionan, las galaxias también; van a haber planetas que dejan de existir, nuevos planetas, etc. y si uno se queda muy quieto, se puede quedar en el pasado.

Lo bueno es siempre van a haber lugares que explorar, y eso si bien puede ser abrumador como decía antes, como ya alguno de ustedes nos decía la primera vez que hablamos, una vez que uno tiene las básicas, es bastante accesible moverse entre galaxias y los planetas que contienen de las que vamos a hablar ahora. Ojalá que esta charla les sirva para ver lo bueno más que nada; una especie de mapa espacial para que se sientan más cómodos que ayer para navegar en este universo.

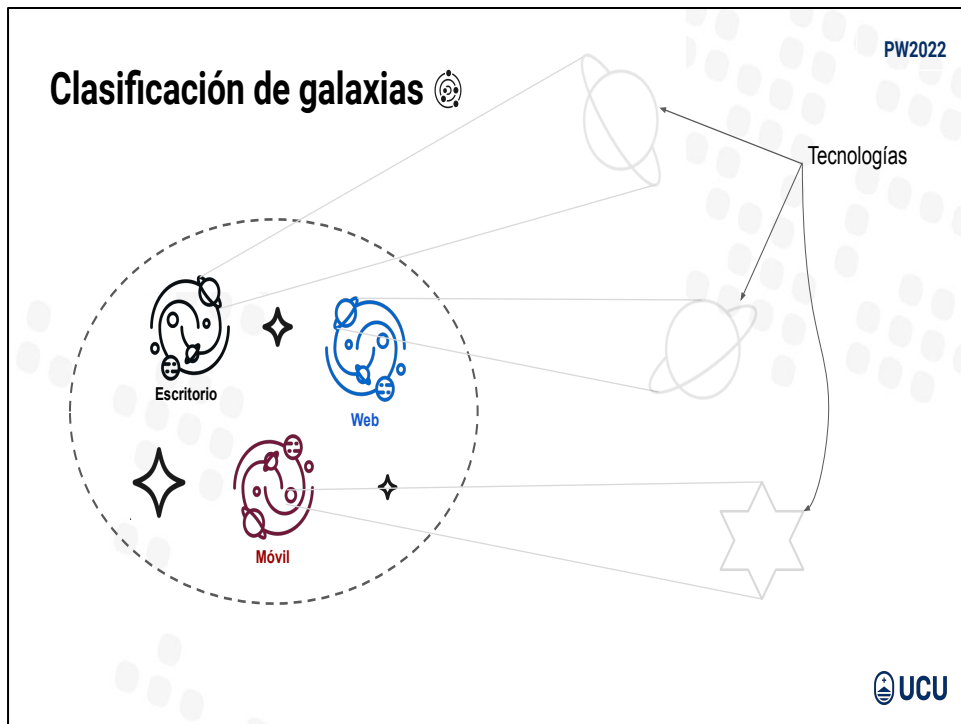


Entonces, estas galaxias son lo que llamamos plataformas.

Uno desarrolla aplicaciones para que funcionen sobre estas plataformas.

Así que si la plataforma es la galaxia, lo que componen las galaxias son lo que nos permiten desarrollar esas aplicaciones.

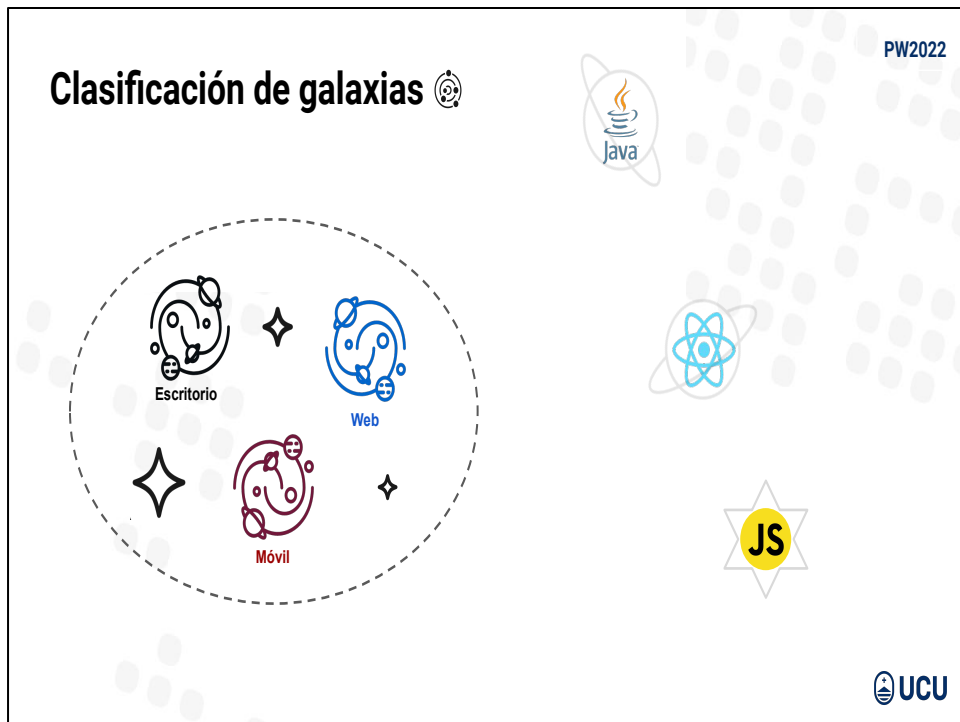
Hablamos de...



Planetas y estrellas... que en la analogía de universo en el desarrollo de software vienen a ser... ¡tecnologías!

Lo que ustedes ya nos compartieron que usan y han probado y están experimentando se les llama tecnologías indistintamente si se utilizan diferentes lenguajes de programación. Cada tecnología es un planeta o una estrella en una galaxia.

Ahora vamos a ver ejemplos.



Antes de seguir, pequeña nota al paso, recomendación de mi parte. Cuando les pregunten con qué tecnologías trabajan eviten responder lenguajes de programación, por una simple razón que les va a ayudar a encontrar su lugar en este universo tan grande.

Como ya vamos a ver, un lenguaje de programación se utiliza en varias plataformas, para hacer cosas diferentes, y si bien tienen la misma base, varía mucho su utilización de plataforma en plataforma.

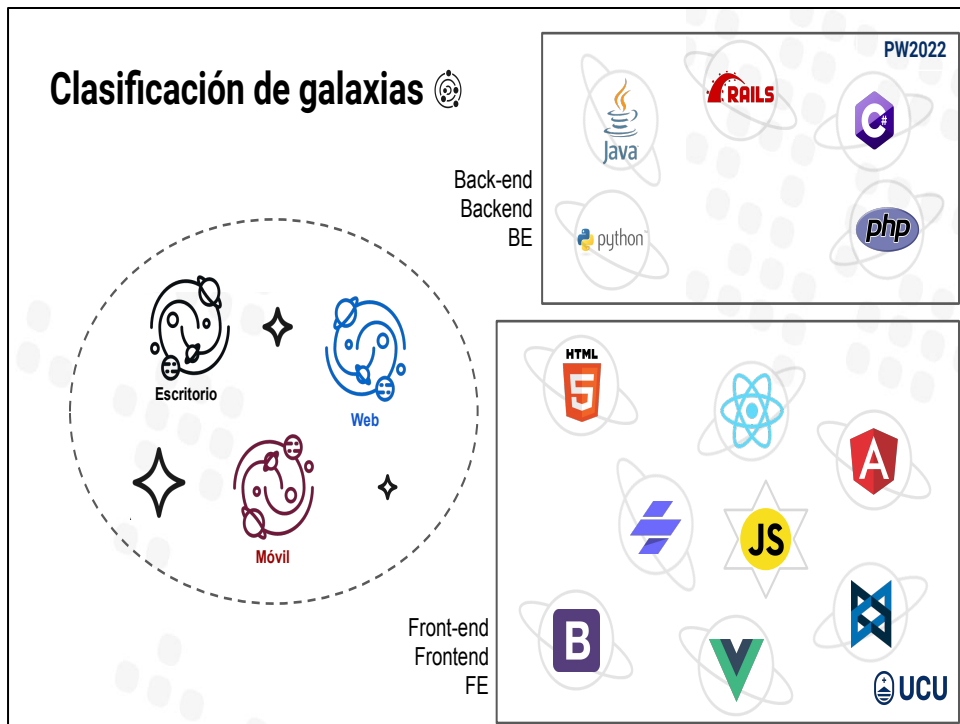
Como bien hicieron algunos al responder la ronda de “edad, con que se sienten cómodos y qué helado les gusta más”, cuando dice React, se asume que trabaja con JavaScript o alguno de sus “sabores” que ya mencionaremos, peeeero, si uno responde JavaScript, puede dejar entrever que uno tiene conocimiento suficiente como para abarcar más que React. De hecho hay dos caminos que se suelen recomendar al meterse a explorar la estrella JS, 1. aprender muy bien JS y toda su flexibilidad para poder acceder más fácilmente a todos los planetas que circundan JS como es React (...), o aprender directamente los planetas directo, lo que hace que te especialices, y al cabo de un tiempo de todas formas vas aprendiendo JS.



Ahora que ya se van haciendo una idea, expandamos un poco más el universo con más planetas, ya que hay más tecnologías... ¿conocen alguno más de estos?

Haciendo la correspondencia, tenemos que... Java sirve para los 3, Rails esencialmente para Web pero se puede también Móvil, C# para los 3, PHP para web

Como les decía, la estrella JS tiene más planetas que la orbitan... esos planetas existen por la capacidad y flexibilidad de JS



Existe una clasificación más entre los planetas y es a qué lado pertenecen. El cuadrante al que pertenecen.

Por lado me refiero a que los de arriba son tecnologías que corren en un servidor, en la famosa nube, que de una manera u otra ocultan su presencia del usuario final.

Con el BE hay que saber muy bien cómo comunicarse con él para poder consumir sus servicios y lograr los objetivos propuestos en el software que se quiere implementar.

Por otro lado tenemos las tecnologías de FE, que manipulando determinadas propiedades de elementos y agregándoles comportamiento conseguimos construir interfaces de usuario, aquello que el usuario final ve y con lo que interactúa para usar la aplicación.

También cabe aclarar que con tecnologías de BE se pueden hacer interfaces para aplicaciones de escritorio y móvil, no solo con las tecnologías de FE se puede producir interfaces de usuario.

Pregunta: aquellos que comentaban que les gusta el diseño, ¿identifican que lo que decían realmente es que les gusta las tecnologías de FE? Porque el diseño de por sí de las interfaces no las hacen los desarrolladores. Ya veremos eso en la sección de ¿Cómo elegimos?



Ahora para mostrarles últimos eventos “cósmicos” importantes...





La estrella JS que está del lado del FE generó un planeta nuevo del lado del BE, lo que hizo que la manera de pensar el BE cambie en cierta manera para los desarrolladores web. De pronto la tecnología que usan de fondo como es JS se podía usar no solo en FE, sino en BE...

Y luego algunos que trabajaron haciendo Node.js tomaron la lista de cosas que no le gustaron de Node.js e hicieron otra implementación...

Más sobre los planetas FE 🌐

PW2022

 vs. 

Librería Framework


UCU

Librería vs framework

¿Conocen esta distinción? ¿Quien se anima a aventurar una explicación?


## Más sobre los planetas FE

PW2022




Librería

vs.



Framework



Varios problemas resueltos



Hay un debate muy grande sobre esta distinción, pero lo que dice el sentido común es que cuando uno usa un framework uno siente que varios problemas de “los de siempre” ya fueron resueltos y uno consume una función solamente pasándole parámetros para conseguir lo que uno busca. Por ejemplo, el ruteo. Por ejemplo Angular, que es un framework que provee un marco de trabajo para implementar ruteo entre páginas, y consumir servicios para traer información de una base de datos por ejemplo, etc.

Más sobre los planetas FE 🌐

PW2022

 vs. 

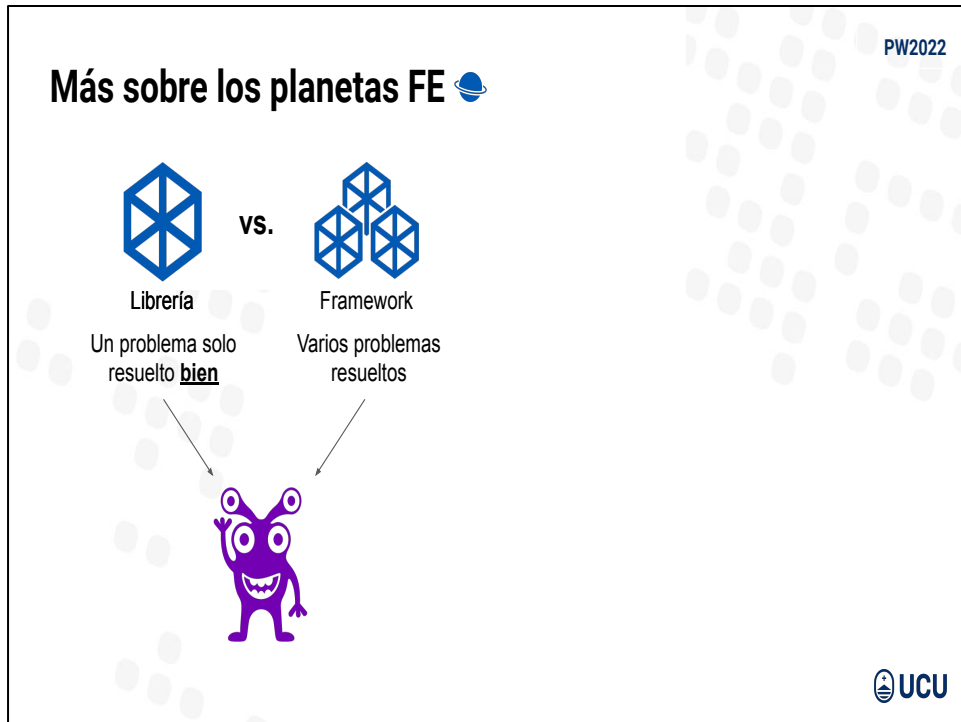
Librería      Framework

Un problema solo resuelto bien      Varios problemas resueltos

UCU

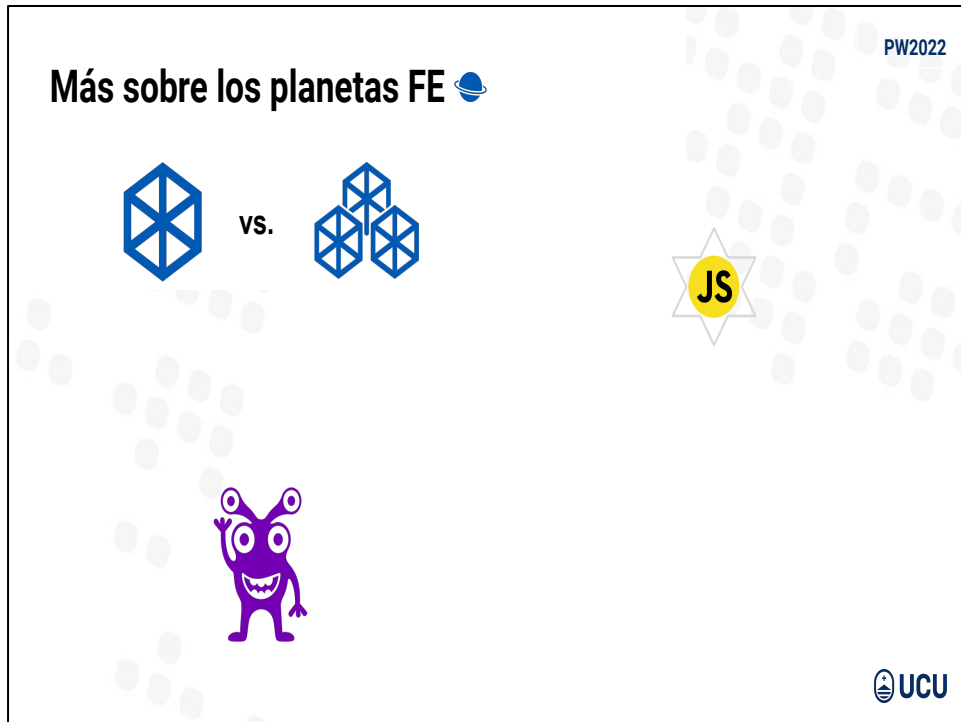
Por otro lado están las librerías, que por lo general solucionan un problema solo pero muy muy bien. Por ejemplo, hay una librería que quizás escucharon hablar, se llama MomentJS, y su funcionalidad es trabajar con fechas.

Pero como todo, también hay cosas raras que intentan hacer las dos cosas, solucionando temas muy particulares pero de manera que circunde un marco de trabajo....

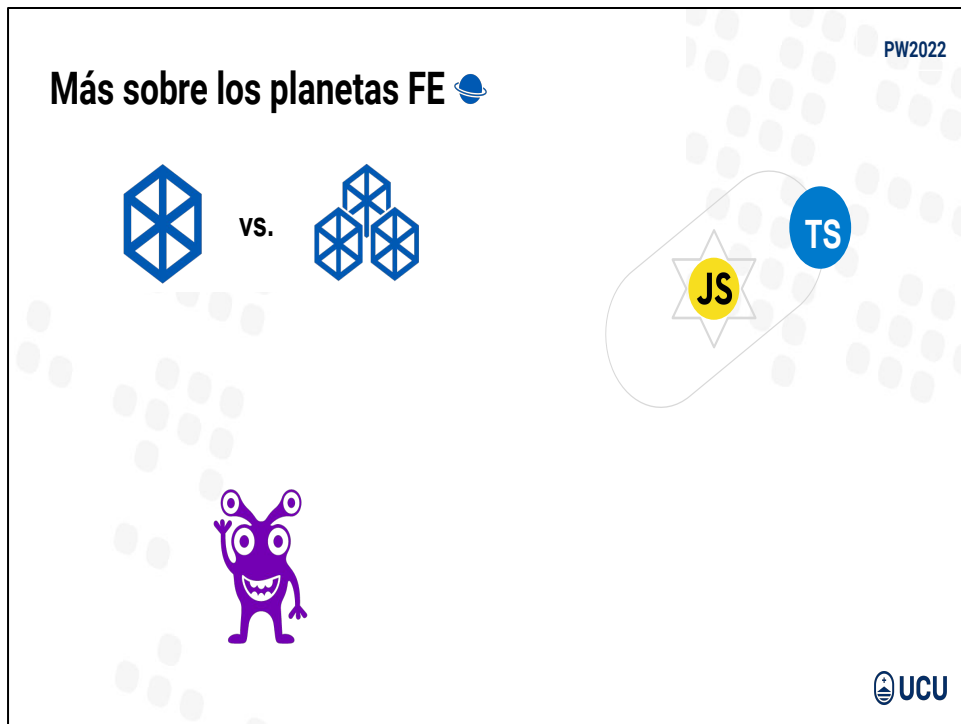


Existen híbridos, y es ahí donde la línea que los separa desaparece... entonces no termina aplicando la separación...

Solamente tengan en cuenta que existe esta distinción porque es la jerga común en el desarrollo de FE.



Luego con respecto a la estrella JS, tenemos también que hablar de sus características... por ejemplo, hay otra estrella que lo orbita, y se llama...



Typescript... ¿oyeron hablar de eso?

En pocas palabras, como la estrella JS es tan tan flexible como para permitir tener planetas que lo rodean como Angular, React, etc, existe la posibilidad de cometer errores de implementación que refieren a errarle de tipo de datos al trabajar con variables y funciones. Si yo quiero con JS puedo asignar un número a una variable y luego asignarle un texto, y JS no se quejaría, es parte de cómo funciona.

Typescript le pone una especie de parche a JS: le agrega tipos de variable explícitos a JS para que se entienda mejor qué tipo de dato se maneja en la aplicación y así evitar errores de ese estilo. Y eso es solo una de las características de Typescript, les recomiendo que lo vean más a fondo, más sobre cómo pueden experimentar con eso en la sección de Exploración más adelante.



## ¿Cómo elegimos? 🚀

- Primero que nada elijan a conciencia en lo posible, no por descarte

PW2022

UCU

Es importante que con este vistazo rápido del universo, puedan elegir algo a conciencia. Así se hace más llevadero el pasar del trabajo como desarrollador.

Estamos hablando que su elección va a determinar de alguna manera como van a pasar su tiempo al menos el que le van a dedicar a esta práctica formativa.

Así que preguntense... ¿en qué galaxia (o plataforma) quieren programar? ¿Web? ¿Móvil? ¿Qué capacidades de la galaxia quisieran usar? ¿Acelerómetro? ¿GPS? y con eso definido mirando los planetas (o tecnologías) y lo que cada una habilita hacer, ¿en qué cuadrante quieren enfocarse? ¿FE? ¿BE? Quizás ambos, que también es posible... ya lo voy a mencionar.


También pueden seguir la corriente e ir teniendo que aprender lo que se les propone en el momento, pero háganlo sabiendo que es ese el camino que quieren recorrer frente a este universo tan grande.

Siguiendo con la lista de recomendaciones a la hora de elegir su camino, y como para sacar de lado mitos y creencias...

## ¿Cómo elegimos? 🚀

- Primero que nada elijan a conciencia en lo posible, no por descarte
- Luego si quieren cambiarse... ¡se puede!

PW2022



En un universo tan grande con tantas galaxias, diferentes cuadrantes y planetas y estrellas, es muy difícil que uno solo se quede con la elección que hizo en su momento.

Sin ir muy lejos en mi caso yo empecé con en la plataforma Web, usando PHP en el cuadrante de FE y hace ya como 9 años que me pasé al cuadrante de FE y me sigo moviendo ahí. Y en este momento estoy trabajando también con Node.js en el cuadrante de BE al mismo tiempo.

Sientanse libres de moverse entre cuadrantes, planetas, galaxias... es una de las cosas buenas tener mucho para elegir, nunca te vas a aburrir con tanto para elegir.

Acá en Globant es bastante más común de lo que se piensa cambiarse de tecnología. Y eso es algo bueno que tiene una empresa como Globant, donde el universo de clientes con necesidades muy diferentes es casi tan grande como el de plataformas y tecnologías. Otras empresas solo trabajan con una sola plataforma en su propio software y a menos que cambie o evolucione, tendrán más chances de quedarse estancados en un planeta.

## ¿Cómo elegimos? 🚀

- Primero que nada elijan a conciencia en lo posible, no por descarte
- Luego si quieren cambiarse... ¡se puede!
- Variedad de perfiles, por ejemplo en FE:

Ahora siguiendo con la lista... ¿sabían que existen muchas maneras de colaborar con este universo, no solo como desarrollador?

## ¿Cómo elegimos? 🚀

- Primero que nada elijan a conciencia en lo posible, no por descarte
- Luego si quieren cambiarse... ¡se puede!
- Variedad de perfiles, por ejemplo en FE:
  - ◆ User interface (UI) designer
  - ◆ User experience (UX) designer
  - ◆ Front-end developer
  - ◆ Full-stack developer

Tenemos los diseñadores de interfaces de usuario, que con conocimientos de diseño de interfaces generan insumos para que los desarrolladores lo implementen y se vea como fue descripto.

Diseñadores de experiencia de usuarios, que con conocimiento de comportamiento de usuarios generan insumos para que los diseñadores de interfaces puedan trasladarlo a su trabajo. Por lo general los diseñadores suelen hacer ambas cosas al mismo tiempo y se les llama diseñadores de interfaces y experiencias de usuario, muy original el nombre.

Después vienen los desarrolladores de FE, que con los insumos de los perfiles anteriores desarrollan con una tecnología para una plataforma lo que se espera para el FE.

Y lo que les decía de que pueden abarcar tanto FE como BE, se les llama full-stack.

## ¿Cómo elegimos? 🚀

- Primero que nada elijan a conciencia en lo posible, no por descarte
- Luego si quieren cambiarse... ¡se puede!
- Variedad de perfiles, por ejemplo en FE:
  - ◆ User interface (UI) designer
  - ◆ User experience (UX) designer
  - ◆ Front-end developer
  - ◆ Full-stack developer
- ¡Hay más perfiles!
  - ◆ Testing
  - ◆ Analista de negocio

Y tenemos aún más perfiles dentro del universo, por ejemplo, personas encargadas del testing. Las cuales tomando como insumo el trabajo del desarrollador, se aseguran que se haya implementado lo que los diseñadores dijeron.

Y los diseñadores utilizan el trabajo generado por los analista de negocios que generan documentación de lo que el cliente quiere hacer en ese software a desarrollar. Los analistas de negocio tienen el trabajo de estar en contacto cercano con el cliente y entender la problemática que el software tiene que resolver para traducirlo en documentación para los diseñadores y para los desarrolladores.



Como para darles herramientas para que puedan explorar en este universo tan grande, así los ayuda a elegir mejor en caso que quisieran y puedan elegir...

¿Conocen qué son los meetups? Existen de varios tipos, por ejemplo los orientados a tecnología.


Son espacios donde se hablan de plataformas y tecnologías en particular, donde hay alguien contando su experiencia, y suelen estar abierto a todo público, o sea que no requiere conocimiento previo. Se pueden fijar en la descripción de los eventos a ver para qué tipo de perfil es orientada la charla, principiante, intermedio etc. Antes de la pandemia se hacían presenciales, ahora son todas virtuales prácticamente.

Otra manera de experimentar y aprender es...

# Exploración

- Meetups ([meetup.com](https://www.meetup.com))
- Playgrounds

PW2022



Los playgrounds.

Quizás ya lo conozcan, pero el que les comentaba al hablar de Typescript...

# Exploración

- Meetups ([meetup.com](https://www.meetup.com))
- Playgrounds
  - ◆ Typescript ([typescriptlang.org/play](https://typescriptlang.org/play))

PW2022

 UCU

Está en la misma página oficial de Typescript

O tenemos otros...



# Exploración

- Meetups ([meetup.com](https://www.meetup.com))
- Playgrounds
  - ◆ Typescript ([typescriptlang.org/play](https://typescriptlang.org/play))
  - ◆ JS Fiddle ([jsfiddle.net](https://jsfiddle.net))

PW2022



JSFiddle es uno de los más conocidos donde puedes ver el resultado de escribir HTML, CSS y JS, incluso se puede configurar para que use las tecnologías que quieras, para usar React, Angular, etc

# Exploración

- Meetups ([meetup.com](https://www.meetup.com))
- Playgrounds
  - ◆ Typescript ([typescriptlang.org/play](https://www.typescriptlang.org/play))
  - ◆ JS Fiddle ([jsfiddle.net](https://jsfiddle.net))
  - ◆ Repl.it ([repl.it/languages](https://repl.it/languages))
    - Python, node, Java, Ruby, HTML/CSS/JS, Deno, etc