Lenguajes de Programación

**TypeScript**

1. Tipo de Ejecución

TypeScript extiende la sintaxis de JavaScript, por tanto cualquier código JavaScript existente debería funcionar sin problemas. Está pensado para grandes proyectos, los cuales a través de un compilador de TypeScript se traducen a código JavaScript original. El compilador de TypeScript está escrito asimismo en TypeScript, compilado a JavaScript y con [Licencia Apache 2](https://es.wikipedia.org/wiki/Apache_License).

1. Tipos de Desarrollo

2012 fue el año en que Typescript apareció (luego de 2 años de desarrollo), una solución de Microsoft para el desarrollo de aplicaciones con Javascript a gran escala, para ellos y para sus clientes. Se creó para solución de muchos problemas para bases de código grandes, aplicaciones de gran escala y proyectos con muchos años de desarrollo.

Utilizar TypeScript en cualquier sitio, en el frontend o en el backed, ya que TypeScript es compilado en JavaScript. La ventaja de TypeScript estará en proyectos empresariales complejos en el lado del servidor.

1. Herramientas de TypeScript

Si trabajas con **frameworks**, usar TypeScript es casi una obligación. **Angular**, como hemos dicho antes, está escrito en este lenguaje y, por tanto, su documentación. Si a esto le sumamos que en todas las empresas de **desarrollo software** los principales proyectos utilizan alguno de estos frameworks o librerías, blanco y en botella. Hay a disposición el enorme ecosistema de librerías y frameworks que existen para JavaScript. Con Typescript podemos desarrollas aplicaciones con React, Vue, Angular, etc.

1. Sintaxis

let x : number;

x = 20;

let y : number;

y = 10;

let z: number;

z= x+y;

console.log(z);

**Erlang**

1. Tipo de Ejecución

Es un [lenguaje](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n) de [programación concurrente](https://es.wikipedia.org/wiki/Programaci%C3%B3n_concurrente) (u [orientado a la concurrencia](https://es.wikipedia.org/w/index.php?title=Orientado_a_la_concurrencia&action=edit&redlink=1)) y un sistema de ejecución que incluye una máquina virtual (BEAM) y bibliotecas (OTP). La implementación de Erlang es principalmente interpretada, pero también incluye un compilador [HiPE](https://es.wikipedia.org/wiki/HiPE" \o "HiPE) (sólo soportado en algunas plataformas).

1. Tipos de Desarrollo

Fue diseñado en la compañía [Ericsson](https://es.wikipedia.org/wiki/Ericsson) para realizar aplicaciones distribuidas, tolerantes de fallos, [soft-real-time](https://es.wikipedia.org/wiki/Sistema_de_tiempo_real" \o "Sistema de tiempo real) y de funcionamiento ininterrumpido. Proporciona el [cambio en caliente](https://es.wikipedia.org/wiki/Hot_Swap) de código de forma que este se puede cambiar sin parar el sistema.

 Erlang está diseñado para correr en ambientes distribuidos. Una máquina virtual de Erlang es un nodo de Erlang. Un sistema distribuido de Erlang es una red de nodos (generalmente uno por procesador). Un nodo Erlang puede crear procesos paralelos ejecutando en otros nodos, lo cuales podrían utilizar otros sistemas operativos. Los procesos que residen en diferentes nodos se comunican exactamente de la misma forma que si estuviesen en un nodo local.

1. Herramientas de Erlang

El paquete de software [LYME](https://es.wikipedia.org/w/index.php?title=LYME&action=edit&redlink=1) se basa en Erlang y ofrece una alternativa a [LAMP](https://es.wikipedia.org/wiki/LAMP) la plataforma OTP ([*Open Telecom Plataform*](https://www.ecured.cu/index.php?title=Open_Telecom_Plataform&action=edit&redlink=1)) que es el núcleo del lenguaje Erlang; su máquina virtual está basada en esa arquitectura, donde los procesos sólo pueden comunicarse mediante el paso de mensajes. Debido a que es orientado a [concurrencia](https://www.ecured.cu/Concurrencia) y a procesos todo lo que se programa son procesos concurrentes y totalmente independientes, siendo así que si falla alguno no afecta el funcionamiento de los demás y también puede ser reiniciado instantáneamente por su proceso supervisor (comportamiento OTP árbol de supervisión de procesos).

1. Sintaxis

X=2.

Y=3.

add(X,Y)

main(\_) -> io:format(X).