Mejorando la calidad de código con ECMA 6 y TypeScript

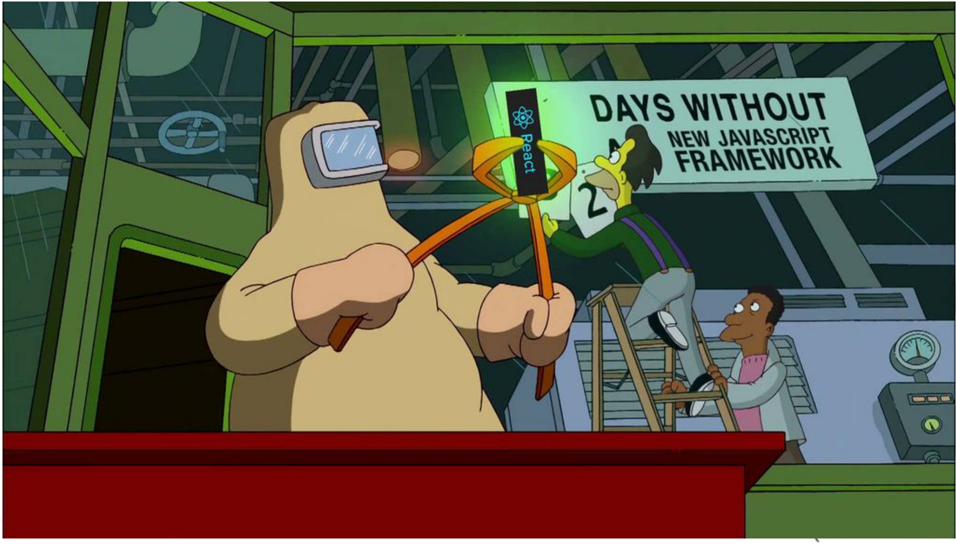
Víctor Orozco

30 de mayo de 2020

@tuxtor

ACADEMIK

JavaScript



Clientes JavaScript

1995-2012: JavaScript es malo! - Developer X con conocimientos de otro lenguaje que no sea JS.

- Orientado a hacks
- Imperativo (manipulación DOM)

- 2009: Node.js
- 2009: CoffeeScript
- 2011: Dart
- **2012: TypeScript**
- **2015: ECMA 6**

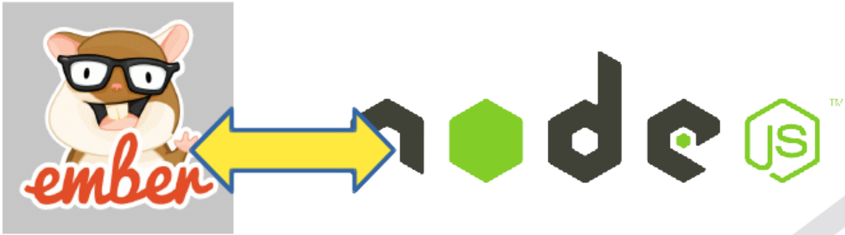
ACADEMIK

Clientes JavaScript/HTML5

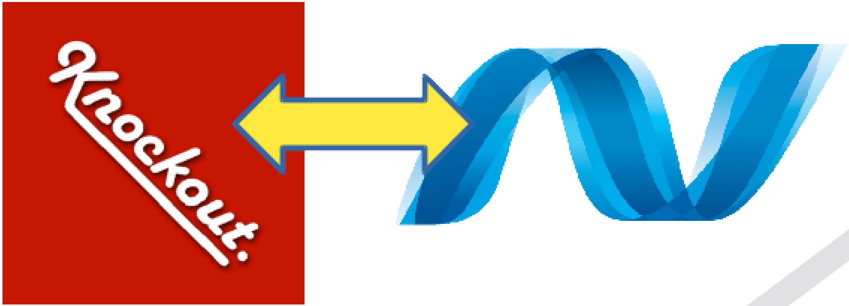
- Rich clients = HTML+JS+CSS3
- MVVM +- MVC del lado del cliente
- JSON/XML
- Rest - Request-response
- Websockets - Full duplex

ACADEMIK

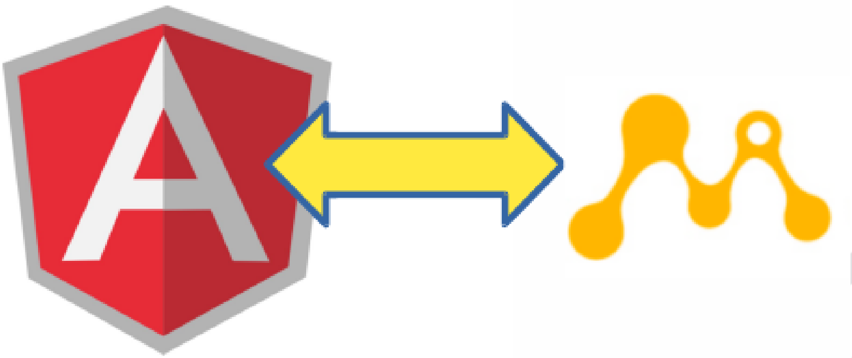
Arquitectura 2015



Arquitectura 2017



Arquitectura 2019



Clientes JavaScript

- AJAX (Microsoft)
- jQuery, YUI, Dojo

ACADEMIK

Clientes JavaScript

- AJAX (Microsoft)
- jQuery, YUI, Dojo
- GWT, Icefaces/Primefaces, Vaadin

Clientes JavaScript

- AJAX (Microsoft)
- jQuery, YUI, Dojo
- GWT, Icefaces/Primefaces, Vaadin
- HTML5, CSS3, WebSockets, WebRTC, HTML Components

ACADEMIK

Clientes JavaScript

- AJAX (Microsoft)
- jQuery, YUI, Dojo
- GWT, Icefaces/Primefaces, Vaadin
- HTML5, CSS3, WebSockets, WebRTC, HTML Components
- AngularJS, Knockout.js, Ember.js

Clientes JavaScript

- AJAX (Microsoft)
- jQuery, YUI, Dojo
- GWT, Icefaces/Primefaces, Vaadin
- HTML5, CSS3, WebSockets, WebRTC, HTML Components
- AngularJS, Knockout.js, Ember.js
- 2013: React, 2014: Vue (DIY/Biblioteca)

Clientes JavaScript

- AJAX (Microsoft)
- jQuery, YUI, Dojo
- GWT, Icefaces/Primefaces, Vaadin
- HTML5, CSS3, WebSockets, WebRTC, HTML Components
- AngularJS, Knockout.js, Ember.js
- 2013: React, 2014: Vue (DIY/Biblioteca)
- 2015: Oracle JET, 2016: Angular, 2019: Nuxt.js (Framework)

Clave MVVM

Clientes JavaScript

- AJAX (Microsoft)
- jQuery, YUI, Dojo

Clientes JavaScript

- AJAX (Microsoft)
- jQuery, YUI, Dojo
- GWT, Icefaces/Primefaces, Vaadin

Clientes JavaScript

- AJAX (Microsoft)
- jQuery, YUI, Dojo
- GWT, Icefaces/Primefaces, Vaadin
- HTML5, CSS3, WebSockets, WebRTC, HTML Components

Clientes JavaScript

- AJAX (Microsoft)
- jQuery, YUI, Dojo
- GWT, Icefaces/Primefaces, Vaadin
- HTML5, CSS3, WebSockets, WebRTC, HTML Components
- AngularJS, Knockout.js, Ember.js

Clientes JavaScript

- AJAX (Microsoft)
- jQuery, YUI, Dojo
- GWT, Icefaces/Primefaces, Vaadin
- HTML5, CSS3, WebSockets, WebRTC, HTML Components
- AngularJS, Knockout.js, Ember.js
- 2013: React, 2014: Vue (DIY/Biblioteca)

Clientes JavaScript

- AJAX (Microsoft)
- jQuery, YUI, Dojo
- GWT, Icefaces/Primefaces, Vaadin
- HTML5, CSS3, WebSockets, WebRTC, HTML Components
- AngularJS, Knockout.js, Ember.js
- 2013: React, 2014: Vue (DIY/Biblioteca)
- 2015: Oracle JET, 2016: Angular, 2019: Nuxt.js (Framework)

Clave MVVM

Clientes JavaScript - Resumen

- **JS, TS**, Dart, CoffeeScript (lenguajes)
- **Angular**, Nuxt.js (frameworks)
- **Webpack**, Parcel, Brocoli (SCM)
- **npm**, yarn (Dependencias)

ACADEMIK

TS

TypeScript

- Microsoft
- Transpila TS -> JS
- Idea general: JS + Tipos + ES.Next
- Todo código JS es código TS valido
- El código JS es difícil de escalar a menos que sean genios

ACADEMIK

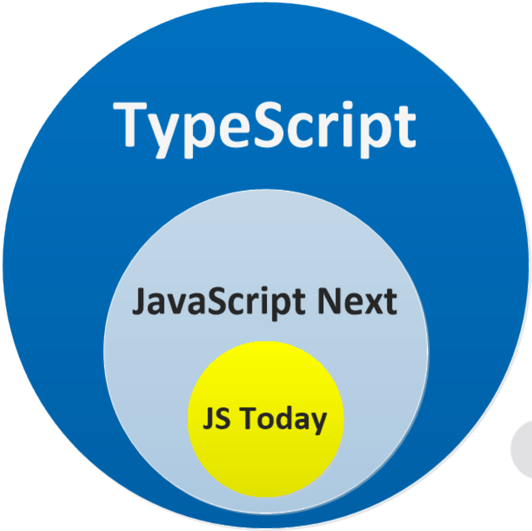
TypeScript vs JavaScript

- debug
- Autocompetion
- Refactoring
- Navegación en los IDEs
- Linting

TypeScript transpiler

- Resultado final es JS idiomatico
- Por defecto siempre produce salida
- Superset
- Tipado estático

TS vs JS

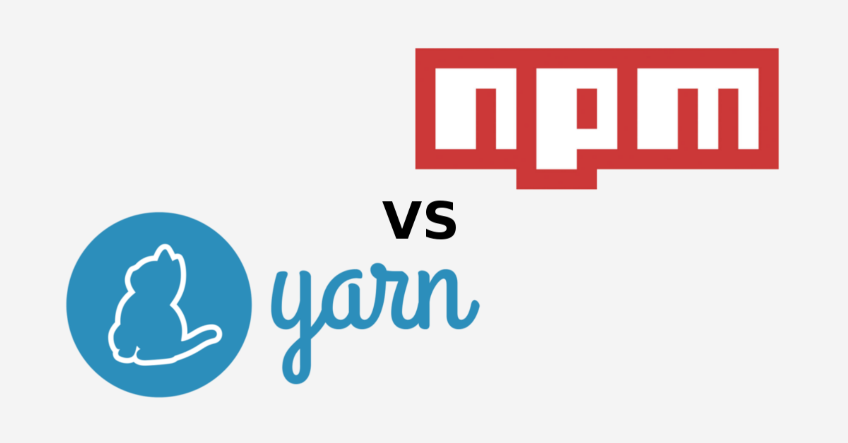


TypeScript

JavaScript Next

JS Today

JS en CLI



TypeScript instalación

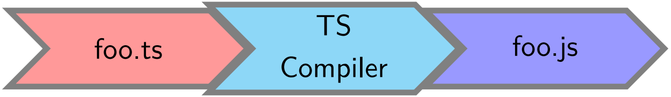
- Node.js
- Gestor de paquetes
- Editor o IDE (VSCode/IntelliJ IDEA)

```
npm install -g typescript
npm view typescript version
```

Luego

```
tsc -v
```

TypeScript



1. tsc
2. Webpack
3. ng-cli

ACADEMIK

TypeScript - Ejercicio 1

```
tsc ex1.ts
```

```
var a = 123
a.trim()
```

ACADEMIK

TypeScript - Ejercicio 1

```
tsc ex1.ts --watch
```

```
var a : string = 123
a.trim()
```

TypeScript

- Type annotations
- Type inference
- Compile time type checking
- Optional and default params
- Classes
- Interfaces
- Structural typing
- Arrow function expression

- Enums
- Generics
- Modules
- Tuple types
- Union types and type guards

TypeScript - Típos

- Object - any
- void - void
- boolean - boolean
- integer, double . . . - number
- String, char - string
- Type[] - type[]

ACADEMIK

NPM

NPM

- Package manager
- Task runner (invoker)
- Fuerte por la comunidad Node.js
- package.json

ACADEMIK

NPM

```
mkdir ex2
cd ex2
npm init
```

Package.json

NPM - Package.json

```json
{
    "name": "ex2",
    "version": "1.0.0",
    "description": "Foo package",
    "main": "index.js",
    "scripts": {
        "test": "echo \"Error: no test specified\" && exit 1"
    },
    "author": "",
    "license": "ISC"
}
```

NPM - Paquetes

https://npmjs.org

```
npm install cowsay
ls -l node_modules
```

ACADEMIK

TypeScript - Ejercicio 3

cowsay.ts

```
var cowsay = require("cowsay");

console.log(cowsay.say({
    text : "I'm a moooodule",
    e : "oO",
    T : "U "
}));
```

```
node cowsay.js
```

NPM - Resumen

- npm install
- npm install -g
- npm install –save
- require();
- npm uninstall

ACADEMIK

ECMAScript 6 + TS

ES6

http://es6-features.org/

Classes

```
class Car {
    constructor(color, doorQty) {
        this.color = color;
        this.doorQty = doorQty;
    }

    run(){
        console.log('I am the '+ this.color + ' car and I have ' + this.
            doorQty + ' doors')
    }
}

var c1 = new Car(”Red”, 4);
c1.run();
```

Classes TS

```
class Car {
    color: string;
    doorQty: number;
    constructor(color:string, doorQty: number) {
        this.color = color;
        this.doorQty = doorQty;
    }
    run(){
        console.log(`I am the ${this.color} car and I have ${this.doorQty} doors`)
    }
}

var c1 = new Car(”Red”, 4);
c1.run();
```

ACADEMIK

Arrow function

```
function ScopeTest(tuxAge){
    this.tuxAge = tuxAge;
    this.increaseAge = function(){
        console.log(”incrementando edad”);
        this.tuxAge++;
    }
}

var sTest = new ScopeTest(10);
setTimeout(sTest.increaseAge,1000);
setTimeout(function(){console.log(sTest.tuxAge)},2000);
```

ACADEMIK

Arrow function

```javascript
function ScopeTest(tuxAge){
    var self = this;
    this.tuxAge = tuxAge;
    this.increaseAge = function(){
        console.log(”incrementando edad”);
        self.tuxAge++;
    }
}

var sTest = new ScopeTest(10);
setTimeout(sTest.increaseAge,1000);
setTimeout(function(){console.log(sTest.tuxAge)},2000);
```

Arrow function

```
function ScopeTest(tuxAge){
    this.tuxAge = tuxAge;
    this.increaseAge = () => { //Arrow function
        console.log("incrementando edad");
        this.tuxAge++;
    }
}

var sTest = new ScopeTest(10);
setTimeout(sTest.increaseAge,1000);
setTimeout(function(){console.log(sTest.tuxAge)},2000);
```

Rest parameters

```
function iTakeItAll(first, second, ...allOthers) {
    console.log(allOthers);
}
iTakeItAll('foo', 'bar');
iTakeItAll('foo', 'bar', 'bas', 'qux');
```

let

```
function scopeTest(){
    var info = 123;
    if(true){
        var info = 456;
    }
    console.log(info);
}
scopeTest();
```

const

```
const constante = 12345

const foo = { bar: 123 };
foo.bar = 456;
```

Block scoped

Destructuring

```
var rect = { x: 0, y: 10, width: 15, height: 20 };

var {x, y, width, height} = rect;
console.log(x, y, width, height);

var {w, x, ...remaining} = {w: 1, x: 2, y: 3, z: 4};
console.log(w, x, remaining);
```

Spread operator

Como argumentos

```
function foo(x, y, z) { }
var args = [0, 1, 2];
foo(...args);
```

Como arreglo (destructuring)

```
var list = [1, 2];
list = [...list, 3, 4];
console.log(list);
```

ACADEMIK

TS

Compilation context

- Colección de archivos a ser analizados por tsc
- tsc –init
- tsconfig.json

Compilation context

```
{
    "include":[
        "./folder"
    ],
    "exclude":[
        "./folder/**/*.spec.ts",
        "./folder/someSubFolder"
    ]
}
```

```
{
    "files":[
        "./some/file.ts"
    ]
}
```

Declaration spaces

Type declaration space

```
class Tuz {};
interface Tux {};
type Penguin = {};
```

Pueden ser usados como tipos

```
var foo: Tuz;
var bar: Tux;
var bas: Penguin;
```

Declaration spaces

Variable declaration space

```
class Tuz {};
var someTuz =Tuz;
var someOtherVar = 123;
```

Pueden ser usados como tipos

```
var foo = 123;
var bar: foo; // ERROR
```

Modulos

- AMD - RequireJS, solamente para browser
- CommonJS - Node.js
- SystemJS - superado por ES6 modules
- ES6 modules

Al utilizar módulos, se hace obligatorio el uso de un bundler para paginas web

Modulos TS

- Basado en ES6 modulos
- Global module
- File module
- globals.d.ts

File module

uno.ts

```
export var penguin = "Tucs";
```

dos.ts

```
import { penguin } from "./penguin";
var bar = penguin;
console.log(bar)
```

Modules - Alias

Export

```
let someVar = 123;
export { someVar as aDifferentName };
```

Import

```
import { someVar as aDifferentName } from './foo';
```

ACADEMIK

Modules - Default

Export

```
export default someVar = 123;
export default function someFunction() { }
export default class SomeClass { }
```

Import

```
import someLocalNameForThisFile from ”../foo”;
```

Modules - Resolución

- Por path
- Node
  - ./node_modules/something/foo
  - ../node_modules/something/foo
  - Hasta llegar a /

ACADEMIK

ACADEMIK

TS - Más tipos

Tipos - inline

```
var person: {
    firstName: string;
    secondName: string;
};
person = {
    firstName: 'John',
};
```

ACADEMIK

Tipos - any

```
var penguin: any;

// Takes any and all types
penguin = 'Tuz';
penguin = 123;
```

ACADEMIK

Tipos - Generics

```
function reverse<T>(items: T[]): T[] {
    var toreturn = [];
    for (let i = items.length - 1; i >= 0; i--) {
        toreturn.push(items[i]);
    }
    return toreturn;
}
```

Tipos - Union Type

```
function printText(text: string[]|string) {
    var line = '';
    if (typeof command === 'string') {
        console.log(text.trim());
    } else {
        console.log(text.join(' '));
    }
}
```

Tipos - Intersection Type

```
function extend<T, U>(first: T, second: U): T & U {
    return { ...first, ...second };
}

const x = extend({ a: "hello" }, { b: 42 });
```

ACADEMIK

Tipos - Tuple Type

```
var nameNumber: [string, number];

nameNumber = ['Jenny', 8675309];

nameNumber = ['Jenny', '867-5309'];
```

ACADEMIK

ACADEMIK

WebApps

Frameworks

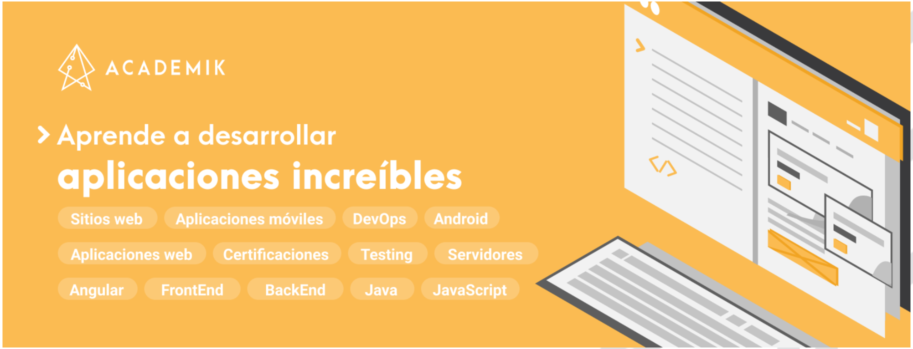
- Angular
- Vue
- React
- Nest

Si es JS, es TS

MicroEjemplo

```
ng new ex6
cd new ex6
ng serve ——open
```

ACADEMIK


ACADEMIK
Aprende a desarrollar
aplicaciones increíbles
Sitios web
Aplicaciones móviles
DevOps
Android
Aplicaciones web
Certificaciones
Testing
Servidores
Angular
FrontEnd
BackEnd
Java
JavaScript

ACADEMIK

Víctor Orozco

- vorozco@nabenik.com
- @tuxtor
- http://vorozco.com
- http://tuxtor.shekalug.org

ACADEMIK

Escríbenos a cursos@academik.io

www.academik.io