

# Archivos y conceptos necesarios en toda aplicación

José María de la Torre

Javier Díaz

Felipe Riquelme

# Agenda

- Changelog
- .gitignore
- Contributing
- License
- Readme
- Badges
- Releases
- Patents
- Dependencias
- Configuraciones

# Changelog

Un Changelog es un archivo donde se puede encontrar una lista cronológica de los cambios notorios en cada versión de un proyecto

## 15.3.2 (September 19, 2016)

---

### React

- Remove plain object warning from `React.createElement` & `React.cloneElement`. (@spudly in #7724)

### React DOM

- Add `playsInline` to supported HTML attributes. (@reaperhulk in #7519)
- Add `as` to supported HTML attributes. (@kevinslin in #7582)
- Improve DOM nesting validation warning about whitespace. (@spicyj in #7515)
- Avoid "Member not found" exception in IE10 when calling `preventDefault()` in Synthetic Events. (@g-palmer in #7411)
- Fix memory leak in `onSelect` implementation. (@AgtLucas in #7533)
- Improve robustness of `document.documentMode` checks to handle Google Tag Manager. (@SchleyB in #7594)
- Add more cases to controlled inputs warning. (@marcin-mazurek in #7544)
- Handle case of popup blockers overriding `document.createEvent`. (@Andarist in #7621)
- Fix issue with `dangerouslySetInnerHTML` and SVG in Internet Explorer. (@zpao in #7618)
- Improve handling of Japanese IME on Internet Explorer. (@msmania in #7107)

### React Test Renderer

- Support error boundaries. (@millermedeiros in #7558, #7569, #7619)
- Skip null ref warning. (@Aweary in #7658)

### React Perf Add-on

- Ensure lifecycle timers are stopped on errors. (@gaelaron in #7548)

## 15.3.1 (August 19, 2016)

---

### React

- Improve performance of development builds in various ways. (@gaelaron in #7461, #7463, #7483, #7488, #7491, #7510)
- Cleanup internal hooks to improve performance of development builds. (@gaelaron in #7464, #7472, #7481, #7496)
- Upgrade fbjs to pick up another performance improvement from @gaelaron for development builds. (@zpao in #7532)
- Improve startup time of React in Node. (@zertosh in #7493)

# .gitignore

Archivo que indica a git que archivos debe ignorar. Si el archivo ya es está en el repositorio, este no se borra.

```
1 .DS_STORE
2 node_modules
3 *~
4 *.pyc
5 static
6 .grunt
7 _SpecRunner.html
8 __benchmarks__
9 build/
10 coverage/
11 .module-cache
12 *.gem
13 docs/.bundle
14 docs/code
15 docs/_site
16 docs/.sass-cache
17 docs/js/*
18 docs/downloads/*.zip
19 docs/vendor/bundle
20 examples/shared/*.js
21 examples/**/bundle.js
22 test/the-files-to-test.generated.js
23 *.log*
24 chrome-user-data
25 *.sublime-project
26 *.sublime-workspace
27 .idea
28 *.iml
29 .vscode
```

# Contributing

Archivo Markdown (generalmente) que indica las instrucciones para contribuir al proyecto

---

## How to contribute to Travis CI

Found a bug in our code, or have an idea for a feature? Here are some notes on how to best report them so we can get it fixed as quickly as possible.

### Security Issues

*Any security issues should be submitted directly to [security@travis-ci.org](mailto:security@travis-ci.org)*

In order to determine whether you are dealing with a security issue, ask yourself these two questions:

1. Can I access something that's not mine, or something I shouldn't have access to?
2. Can I disable something for other people?

If the answer to either of those two questions are "yes", then you're probably dealing with a security issue. Note that even if you answer "no" to both questions, you may still be dealing with a security issue, so if you're unsure, just email us at [security@travis-ci.org](mailto:security@travis-ci.org).

## Pull Requests

Know how to fix something? We love pull requests! Here's a quick guide:

1. [Check for open issues](#), or open a fresh issue to start a discussion around a feature idea or a bug. Opening a separate issue to discuss the change is less important for smaller changes, as the discussion can be done in the pull request.
2. Fork the relevant repository on GitHub, and start making your changes.
3. Check out the README for the project for information specific to that repository.
4. Push the change (we recommend using a separate branch for your feature).
5. Open a pull request.
6. We try to merge and deploy changes as soon as possible, or at least leave some feedback, but if you haven't heard back from us after a couple of days, feel free to leave a comment on the pull request.

# License

- Instrumento legal explicando el uso y/o distribución de un software
- Existen varios como por ejemplo
  - **GNU General Public License:** garantiza a los usuarios finales la libertad de usar, estudiar, compartir (copiar) y modificar el software. Al distribuir el código siempre debe ser código abierto.
  - **MIT License:** Permite hacer todo lo que uno quiera con el código mientras se incluya el copyright original y una copia de la licencia original. Esto se debe incluir en todas las copias del software. Pero no necesariamente tiene que ser open source.
  - **Apache License:** Similar a MIT solo que además provee el uso de patentes de los contribuidores a los usuarios.

# ReadMe

- Documento breve que forma parte de la documentación del software
- Entrega información básica y crucial del software:

Instrucciones generales      Información de contacto

Instrucciones de Configuración      Instrucciones de instalación


Troubleshoot      Como operar el software      Bugs

Información sobre archivos del software ( Manifest )

Copyright & licensing      Creditos

Etc..

# Badges

- Entregan Metadatos del proyecto de forma rápida, fácil y resumida.
- La información entregada debería ser autoexplicativa → 
- Descripción a la izquierda, Información a la derecha.
- Muchas veces se encuentran en los ReadMe
- Existen varios tipos:
  - Status
  - Build
  - Dependencias
  - Versión
  - Etc...



# Releases

- Es una distribución de una versión estable y final del software
- Versión pensada para el uso fuera del equipo de desarrollo
- Puede Ser Pública o Privada
- Se realiza después del periodo de testing del software ( Alpha, Beta )
- Es un “check-point” en sistemas de control de versiones.
- No es un Deploy !

# Patentes

Especificaciones sobre patentes y permisos de uso del software.

## Ejemplo React

Facebook, Inc. ("Facebook") hereby grants to each recipient of the Software ("you") a perpetual, worldwide, royalty-free, non-exclusive, irrevocable (subject to the termination provision below) license under any Necessary Claims, to make, have made, use, sell, offer to sell, import, and otherwise transfer the Software. For avoidance of doubt, no license is granted under Facebook's rights in any patent claims that are infringed by (i) modifications to the Software made by you or any third party or (ii) the Software in combination with any software or other technology.

# Dependencias

- Archivos con declaraciones de las dependencias del proyecto. Pueden especificarse versiones específicas o mínimas de librerías por lo general
- Ejemplo Ruby
  - Gemfile
- Ejemplo Node.js
  - package.json



```
source 'https://rubygems.org'
# Bundle edge Rails instead: gem 'rails',
github: 'rails/rails'
gem 'rails', '~> 5.0.0', '>= 5.0.0.1'
# Use sqlite3 as the database for Active
Record
gem 'sqlite3'
# Use Puma as the app server
gem 'puma', '~> 3.0'
https://github.com/rails/jbuilder
gem 'jbuilder', '~> 2.5'
```

# Configuraciones

- Archivos de configuración de editores, linters, debuggers etc.
- Ejemplo Ruby
  - `.rubocop.yml`

```
AllCops:  
  Exclude:  
    - 'samples/**/*'  
    - 'tmp/**/*'  
    - 'vendor/**/*'  
  TargetRubyVersion: 2.1
```

```
# FIXME: Make the class  
shorter  
Metrics/ClassLength:  
  Exclude:  
    -  
lib/reek/context_builder.rb  
  - lib/reek/cli/options.rb
```



Gracias