

# ¿Qué es un componente en REACT?

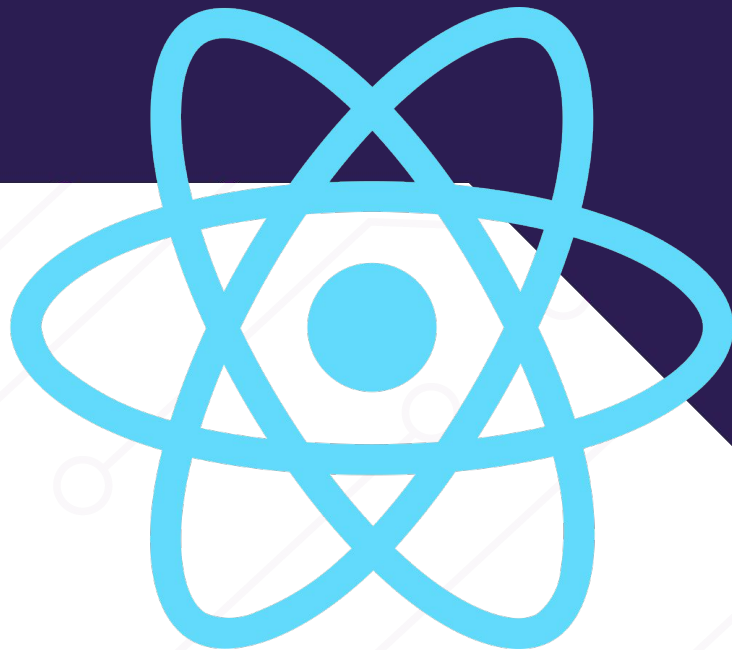
**DEV.F**  
DESARROLLAMOS(PERSONAS);

dev

# Componentes

Es un fragmento de código encapsulado con la capacidad de ser reutilizable que puede tener un estado o no

Mientras que un estado es como se encuentra el contenido del componente a través de su uso





Buscar



Inicio



Mi red



Empleos



Mensajes



Notificaciones

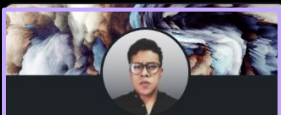


Yo



Productos

[Prueba Premium gratis](#)



## Daniel Gloria Florencio

CEO at Taquito Software S.A.S. de C.V. | Senior Software Developer at Softtek | Web Developer Sensei at Dev.f

Quién ha visto tu perfil 73

Impresiones de tu publicación 12

Accede a información y herramientas exclusivas  
[Prueba Premium gratis](#)

Marcapáginas

## Mis páginas (2)



IEEE UAMI

Notificaciones de la pá... 2



Taquito Software S.A....

Notificaciones de la pá... 3

Recientes



Crear publicación



Foto



Vídeo



Evento



Escribir artículo

Ordenar por: Principal



Norberto Rojas ha comentado esto



Jaikel Chacón · 2º

Supervisor de Recursos Humanos

6 horas ·

+ Seguir

# DETRÁS DE CADA CV:

Hay una persona con **sueños**,  
esperanzas,  
**una familia.**

## LinkedIn Noticias

- **Endeudados y sin saber ahorrar, así vi...**  
hace 1 día
- **El FC Barcelona femenino firma acuer...**  
hace 1 día
- **Peligro: mujeres agotadas**  
hace 4 días · 1.506 lectores
- **Cuentas abandonadas: Esto es lo que...**  
hace 2 días · 304 lectores
- **Pinta mal el panorama económico de ...**  
hace 3 días · 1.064 lectores

Mostrar más

Anuncio

Obtén notificaciones de anuncios de empleo y noticias del sector



Daniel, explora oportunidades relevantes con XPO Logistics, Inc.

Seguir

Acerca de

Centro de ayuda

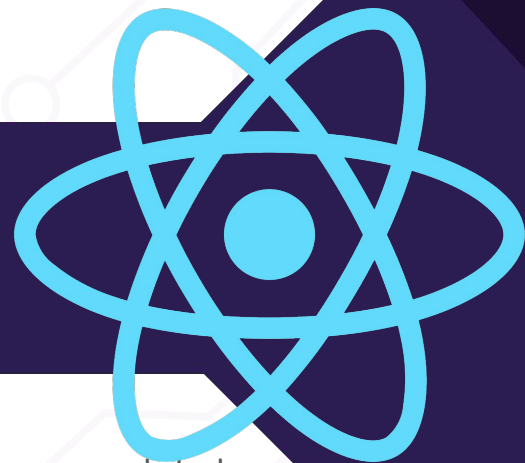


Mensajes

...

DEV.F

# Create-react-app



Create-react-app en React es un paquete en el que podemos instalar de manera global para crear nuestras aplicaciones directamente dentro la librería de React.

Al crear una aplicación React con este paquete, obtendremos un entorno de desarrollo ya configurado.

Create-react-app está listo para que podamos ejecutar todo tipo de acciones e insertar todo tipo de elementos y componentes. Además, esta herramienta está preparada para que nuestro JSX se entienda con Babel y soporta las últimas novedades de JavaScript.

# ¿Qué es JSX?



```
1  const element = <h1>Qué onda pipol!</h1>;
```

Esta curiosa sintaxis de etiquetas no es ni un string ni HTML.

Se llama JSX, y es una extensión de la sintaxis de JavaScript. Recomendamos usarlo con React para describir cómo debería ser la interfaz de usuario. JSX puede recordarte a un lenguaje de plantillas, pero viene con todo el poder de JavaScript y JSX produce “elementos” de React.

The logo consists of the text 'DEV.F.' in a bold, white, sans-serif font. The 'F' is stylized with three small squares at its top right corner. The logo is centered within a dark blue diamond shape.

**DEV.F.**

**Aprendamos la  
estructura de una  
app Ract**

The logo consists of the text 'DEV.F' in a bold, white, sans-serif font. The 'F' is stylized with three small squares at its base. The logo is centered within a dark blue diamond shape.

**DEV.F**

**No me gusta ese  
nombre, ¿Se  
escucha muy poco  
profesional no?**

**DEV.F.**

## Clean Architecture

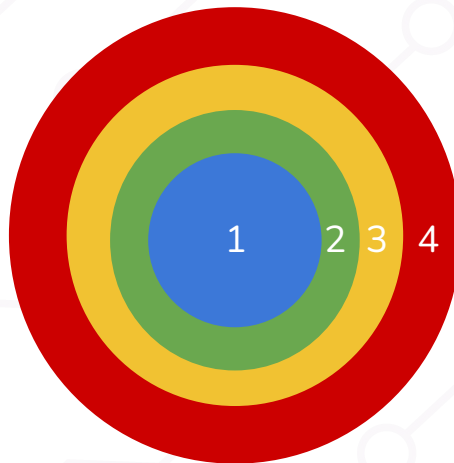




# ¿Qué es Clean architecture?

Como su nombre lo indica es una arquitectura que nos determina un patrón a seguir de cómo organizar nuestros proyectos y se trabaja de la siguiente manera.

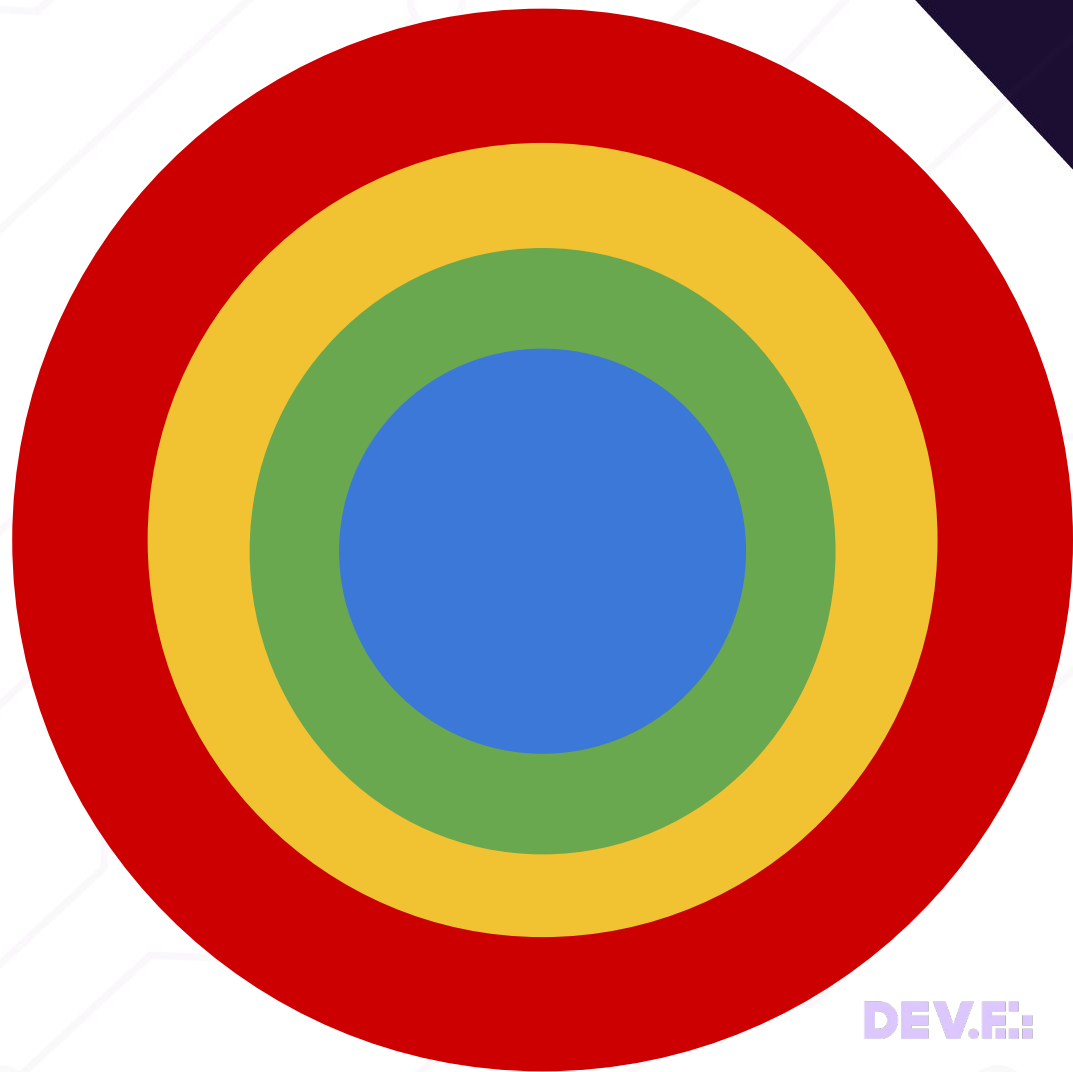
1. **Modelo:** todas las interfaces, clases y entidades así como sus estados.
2. **Casos de uso/ Componentes:** Aquí vive la lógica del negocio, accionan lo que está en el modelo.
3. **Adaptadores:** Sirven como interacción entre servicios externos y sus casos de uso.
4. **Servicios:** Viven todas las llamadas a lógica externa fuera del front.



# Estructura de carpetas

- adapters
- assets
- components
- contexts
- hooks
- interceptors
- models
- pages/views
- redux
- services
- styled-components (componentes de estilo que no tienen lógica)
- utilities

- adapters
- assets
- components
- contexts
- hooks
- interceptors
- models
- pages/views
- redux
- services
- styled-components  
(componentes de estilo que no tienen lógica)
- utilities



# Carpetas reutilizables

Es decir, estas carpetas pueden repetirse dentro de una page/view lo cual es recomendable es por qué los adaptadores solo tendrán un ciclo de vida en su page/view

- **adapters**
- **components**
- **hooks**
- **interceptors**
- **models**
- **services**
- **styled-components** (componentes de estilo que no tienen lógica)



**“Un programador que  
escriba un código limpio,  
entiende perfectamente el  
problema antes de escribir  
el código”**

**Chayanne**

**DEV.F.**  
DESARROLLAMOS(PERSONAS);

dev