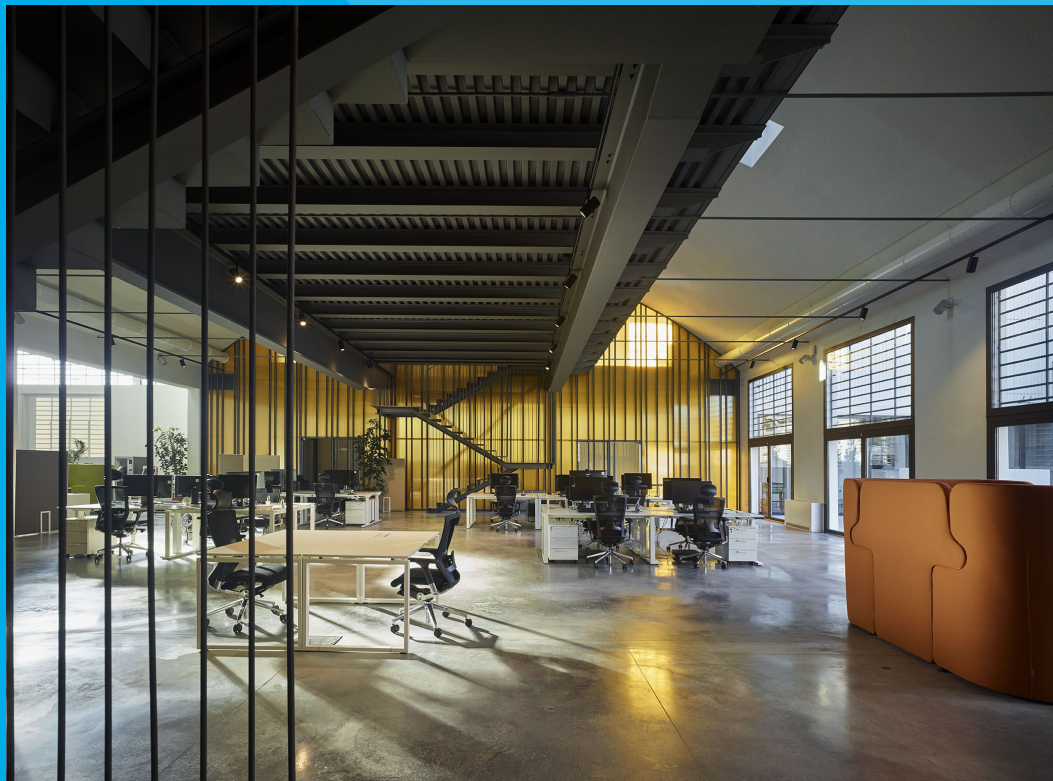


# I CHI SIAMO

Develer un'Azienda che  
che progetta e realizza  
soluzioni hardware e  
software in ambiti  
industriali innovativi.



**L'evento inizierà  
alle 18:35**

## Vita in Develer



**L'evento inizierà  
alle 18:35**

## Vita da develeriani



### | Equilibrio tra vita lavorativa e privata

- Sede aperta 24h su 24
- Flessibilità oraria
- Banca ore





**L'evento inizierà  
alle 18:35**

## Vita da develeriani



| Al passo con i tempi

*Incoraggiamento alla formazione annuale:*

- Budget 800€
- Budget 80 ore



**L'evento inizierà  
alle 18:35**

## Lavora in Develer

- | Web Full Stack Developer
- | C++ Developer
- | Python Developer



**Claudia Cimino**  
*Responsabile risorse umane*

✉ [jobs@develer.com](mailto:jobs@develer.com)



Salvatore Ventrone (aka Ventrosky)

Sviluppatore Web Full Stack

# Guida pratica agli Hooks di React

Develer Workshop

26/10/2022

develer

# I COSA REALIZZEREMO

- SPA gestione libreria
- Utilizzando i seguenti hooks:
  - useState
  - useEffect
  - useContext
  - useRef

# Cos'è React?

- Una libreria JavaScript lato client
- Dichiarativa
- Basata su Componenti
- Tipicamente utilizzata per sviluppare interfacce reattive per il web



## Perchè gli hooks - Ciclo di vita dei componenti

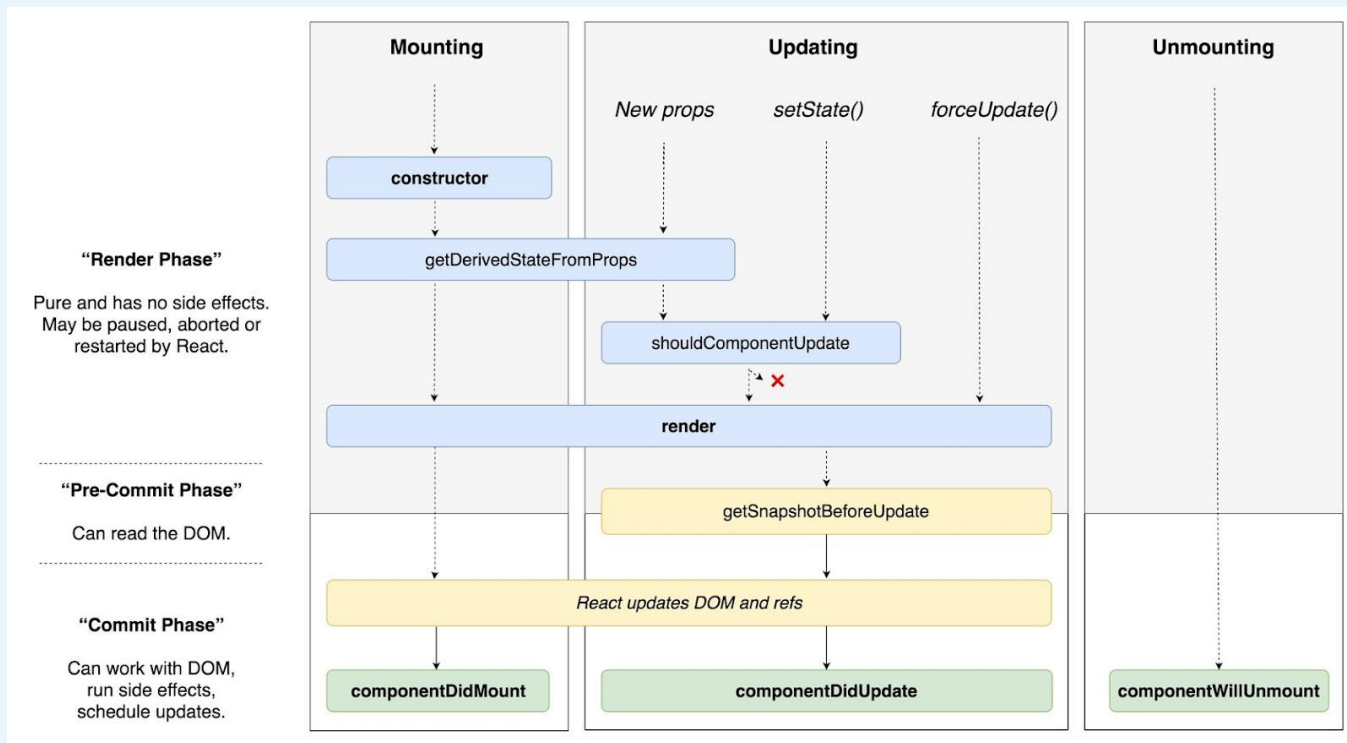


Diagramma da [Dan Abramov](#)

[Introduzione agli Hooks di React - Salvatore Ventrone](#)

# Le regole degli hooks

- Utilizza gli Hooks solo al Top Level
- Utilizza gli Hooks da funzioni React

## I Regola 1 - solo al Top Level

```
// ❌ Hook in una condizione
if (nome !== "") {
  useEffect(() => {
    localStorage.setItem("data", nome);
  });
}
```

```
useEffect(() => {
  // ✅ condizione nel hook
  if (nome !== "") {
    localStorage.setItem("data", nome);
  }
});
```

## Regola 2 - solo da functional components

```
// ❌ Hook in un class component
```

```
render () {  
  useEffect(() => {});  
  return <span></span>;  
}
```

```
// ❌ Hook in una funzione pura JS
```

```
const someUtilFunc = () => {  
  useEffect(() => {});  
};
```

```
// ✅ Hook in un functional component
```

```
const MyComponent = () => {  
  useEffect(() => {});  
  return <span></span>;  
}
```

# Setup

<https://github.com/develer-staff/react-hooks-workshop>



Hooks Base

# useState

<https://reactjs.org/docs/hooks-reference.html#usestate>

Hooks Specifico

## useRef

<https://reactjs.org/docs/hooks-reference.html#useRef>

## Esercizio 1 – useState

git checkout part-1-usestate

Modifichiamo:

- client/src/components/Login.js

Hooks Base

# useEffect

<https://reactjs.org/docs/hooks-reference.html#useeffect>

## Esercizio 2 – useEffect

git checkout part-2-useeffect

Modifichiamo:

- client/src/components/Bookshelf.js



## **Esercizio 3** – useEffect cleanup

git checkout part-3-useeffect-cleanup

Modifichiamo:

- client/src/components/Bookshelf.js

Hooks Base

# useContext

<https://reactjs.org/docs/hooks-reference.html#usecontext>

## Esercizio 4 – useContext

git checkout part-4-usecontext

Modifichiamo:

- client/src/App.js
- client/src/Router.js
- client/src/components/Bookshelf.js
- client/src/components/NavBar.js

## **Extra** – custom hook

git checkout part-5-final-extra

# Ricapitoliamo

- Regole degli Hooks
- useState
- useEffect
- useContext
- useRef



## LINK UTILI

- | React Docs  
<https://reactjs.org/docs/getting-started.html>
- | Regole degli Hooks  
<https://reactjs.org/docs/hooks-rules.html>
- | You Don't Know JS Yet  
<https://github.com/getify/You-Dont-Know-JS>

# Salvatore Ventrone

[ventrosky@develer.com](mailto:ventrosky@develer.com)

 Ventrosky

 **develer**

[www.develer.com](http://www.develer.com)