

**Mobile dev 03**

**19/10/2020**

**Paolo Cargnin**

# Struttura del corso

1° lezione

Progressive Web apps

2° lezione

Progressive Web apps

3° lezione

Ionic Angular

4° lezione

# Sommario lezione

- Setup
- Components
- Deploy

# Cos'è Ionic

## IONIC framework

UI COMPONENTS

Stili Adattivi

Navigazione adattiva

Accesso alle funzionalità  
native

Build native binary

Free

## IONIC hub

App managemant

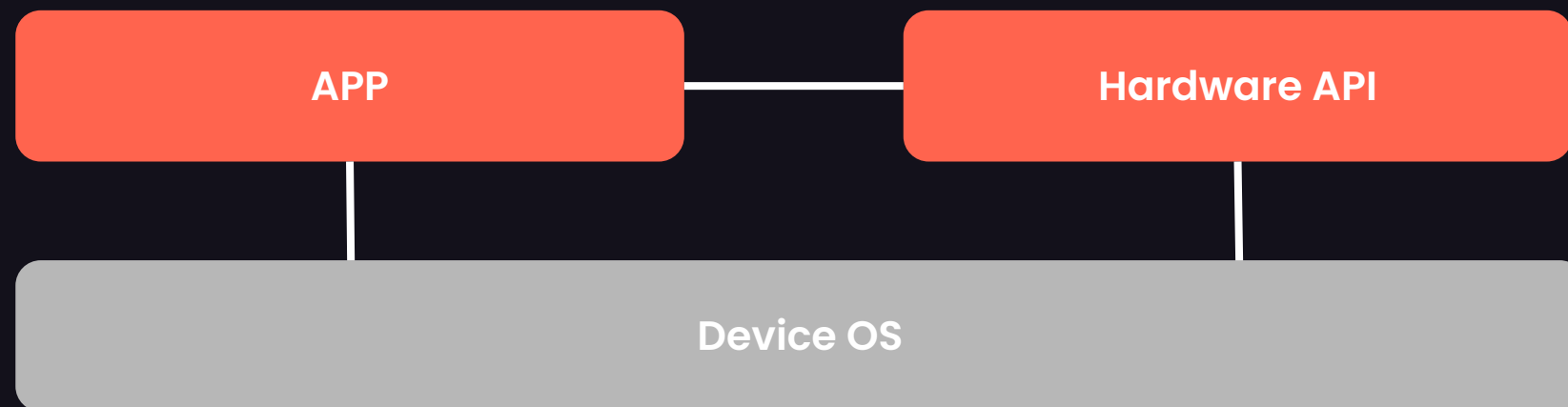
CI / CD

Build native binary

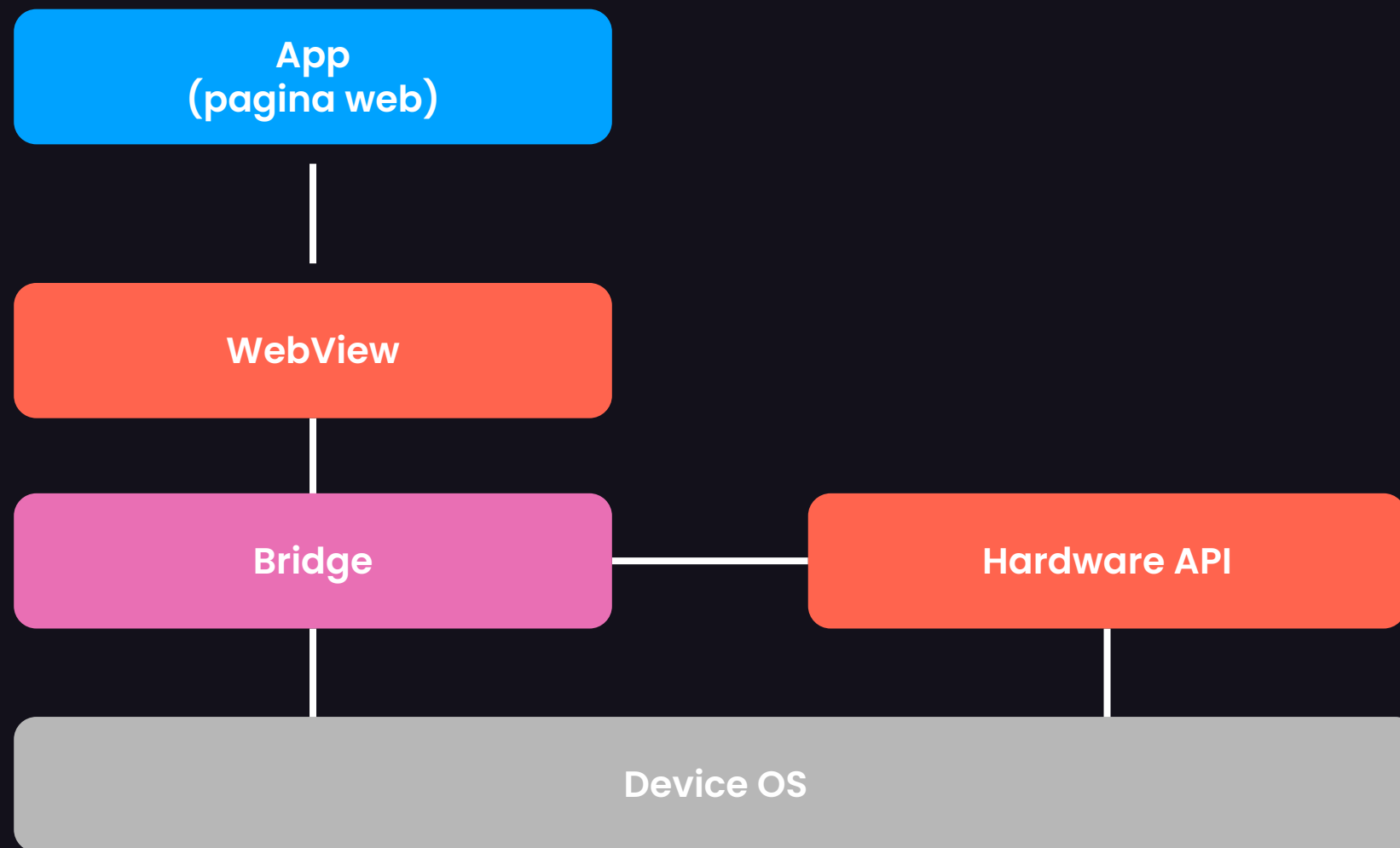
Mobile App Flow

a pagamento\*

# Struttura di una app Nativa



# Struttura di una app IONIC



# Installiamo ionic

```
git npm i -g @ionic/cli@latest cordova-res
```

## Creiamo un account su

<https://ionicframework.com/>

# Creiamo il nostro progetto

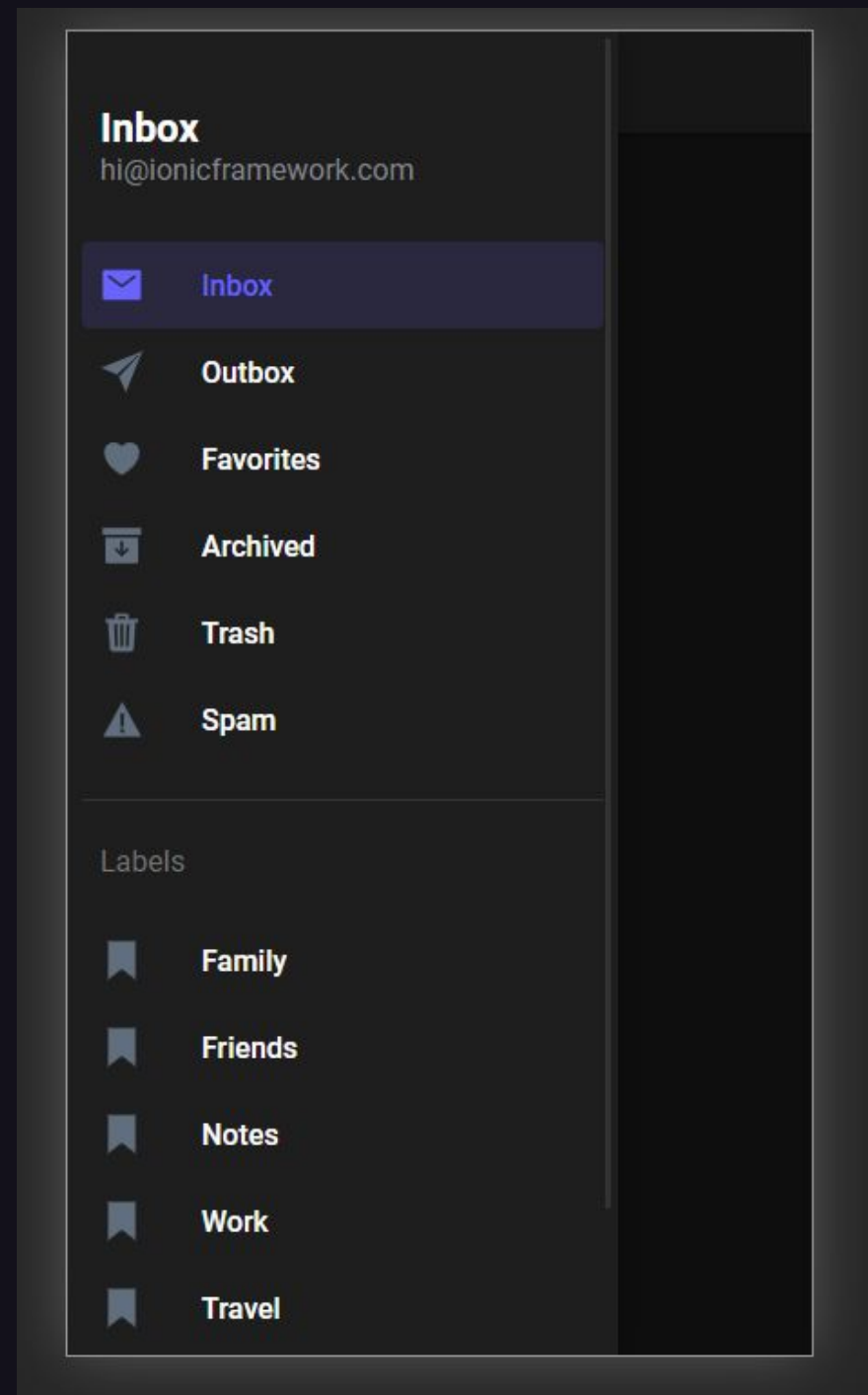
## **Ionic app wizard!**

Collegiamo il nostro account git (github, bitbucket,gitlab) per creare il repository

Cloniamo il repository e eseguiamo  
npm i && ionic serve



# Guardiamo il progetto!



# ionic.config.json

```
{  
  "name": "Our first application",  
  "integrations": {  
    "capacitor": {}  
  },  
  "type": "angular",  
  "id": "ef445d02"  
}
```

# Creiamo il nostro progetto con la CLI

```
ionic start
```

# Creiamo il nostro progetto con la CLI

Angular,

“PokeSchool”,

sidemenu,

Abilitiamo capacitor

# Cosa stiamo creando?

Un gestionale per una scuola di  
pokémon

Per adesso, gestiamo le classi!

# Modifichiamo il progetto

- `ionic generate page poke-class --dry-run`
- Modifichiamo il redirect dell'homepage a `"/poke-class/water"` per mandarlo alla pokeclass
  - Carichiamo la nostra pagina con una ion-card

# Carichiamo le informazioni sul tipo

- Generiamo TypeService:  
`ng generate service type`
- Inseriamo http e creiamo la function get

# Completiamo la “navigazione”!

- Dettagliamo la pagina della classe,
  - Inseriamo la pagina del singolo pokémon