



# Progressive Web Apps

Ejemplo

Para trabajar  
todos juntos

ionicframework(React)  
+  
Mockachino

# Crear una aplicación base

## Ionic

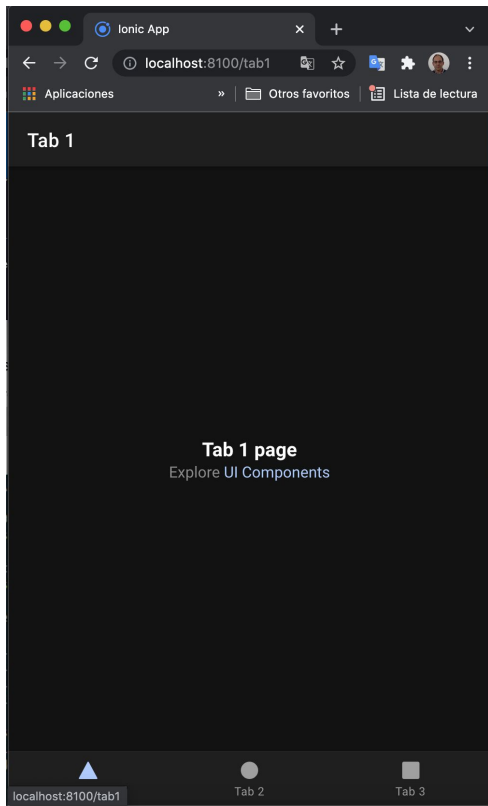
1. Identificarse en <https://ionicframework.com/login> con un usuario de GitHub
2. Crear un proyecto (New - Create New App)
3. Completar los datos de la aplicación (Template: Tabs - Framework: React)
4. Elegir Github como repo
5. Instalar ionic cli `npm install -g @ionic/cli cordova-res`
6. Clonar el repo localmente:

```
git clone https://github.com/fabiancontigliani/um-2021-pwa.git um-2021-pwa
```

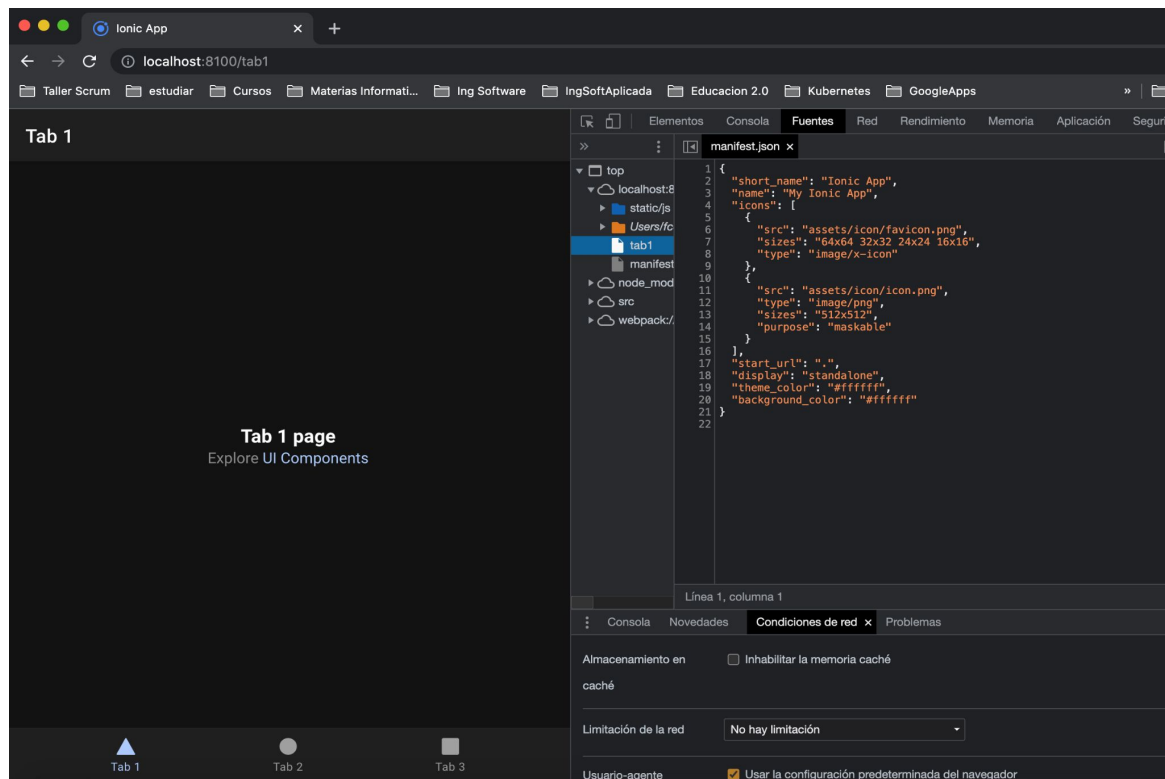
7. Compilar y correr la aplicación

```
cd um-2021-pwa && npm install && ionic serve
```

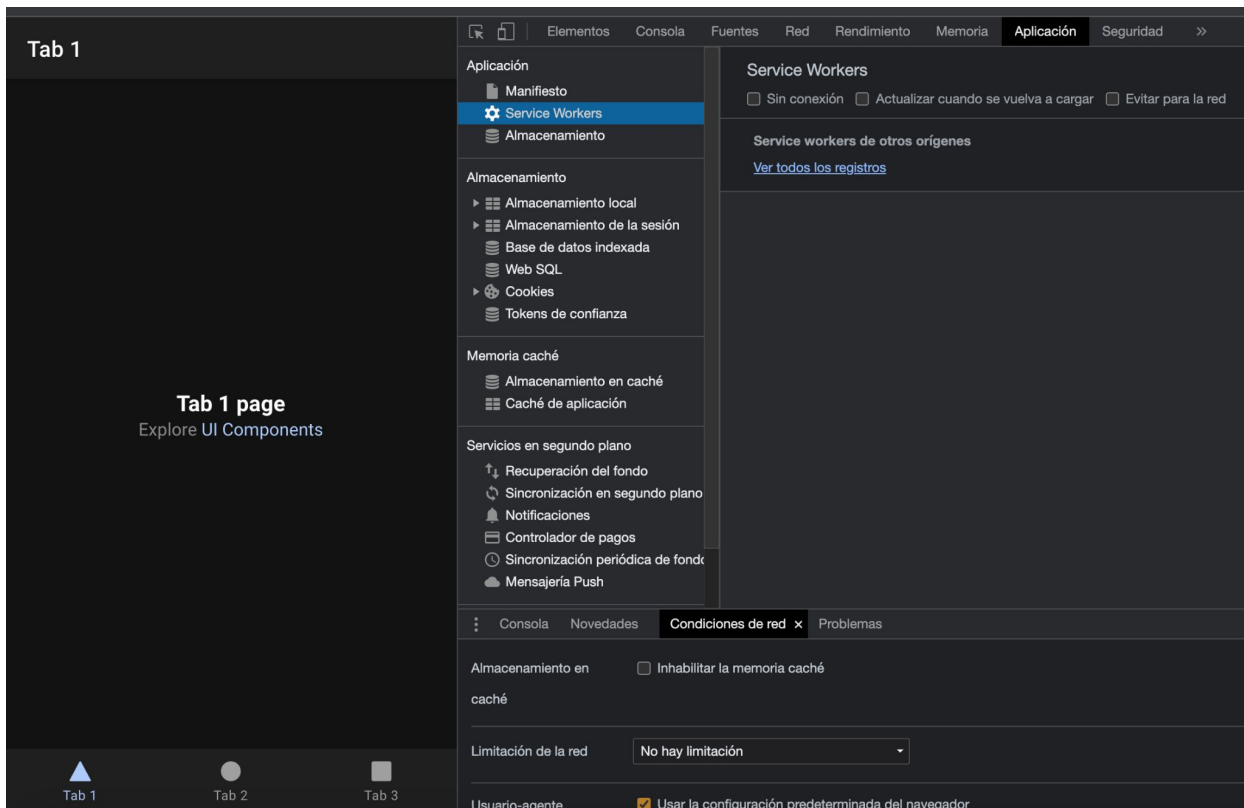
# Vemos la pantalla de la aplicación



# Ya existe el manifest.json



# No tiene Service Workers configurado



# Habilitamos el Service Worker

En el archivo index.tsx cambiamos :

```
//serviceWorkerRegistration.unregister();
```

```
serviceWorkerRegistration.register();
```



# Compilamos el proyecto para producción

- `npm run build`
- `serve -s build`

<https://www.mockachino.com/>

<https://www.mockachino.com/spaces/cfc4d7cc-28e8-40>

Usamos mockachino para simular  
una API que devuelve datos

# Modificamos la cabecera para permitir cross site

"Access-Control-Allow-Origin": "\*"

Ejemplo:

<https://www.mockachino.com/spaces/cfc4d7cc-28e8-40>

# Modificamos el código para llamar a la api y mostrar los datos

```
const getPosts = async () => {
  try {
    const response = await fetch("https://www.mockachino.com/cfc4d7cc-28e8-40/users")
    console.log("llamada");
    if (!response.ok) {
      throw Error(response.statusText)
    }
    const json = await response.json()
    console.log(json.users)

    setUsuarios(json.users)
  } catch (error) {
    console.error("falló")
  }
}
```

# Capturamos los request para gestionar la cache

// Cuando hay una llamada de recuperación responder, si existe, con la cache

```
self.addEventListener('fetch', (event) => {  
  event.respondWith(  
    caches.match(event.request).then((resp) => {  
      console.log("busco en la cache")  
      return resp || fetch(event.request).then((response) => {  
        console.log("no encontré en la cache, busca en la red")  
        return caches.open('v1').then((cache) => {  
          cache.put(event.request, response.clone());  
          console.log("Guarda en la cache", event.request)  
          return response;  
        });  
      });  
    })  
  );  
});
```

# Publicar en firebase

```
npm install firebase
```