

## Exercicio 79: Converter a túa web do tempo nunha app Android con Capacitor JS

**Obxectivo:** Crear unha aplicación para Android (.apk) usando Capacitor JS.

### Requisitos previos:

- Node.js
- npm (ven con Node.js na instalación por defecto)
- O teu proxecto web xa creado (pode ser con Svelte, Astro ou HTML/CSS/JS puro)
- Editor de código (VS Code)

### Pasos a seguir:

#### 1. Inicializa o proxecto con npm

Abre unha **terminal na carpeta raíz** do teu proxecto e **crea unha carpeta www e traslada alí os ficheiros** do teu proxecto e **executa**:

```
npm init -y
```

Isto creará un ficheiro *package.json* necesario para xestionar dependencias.

- Crea na raíz do proxecto un ficheiro ***main.js***
- **En el *package.json* é necesario cambiar a dirección do *main* por *main.js***

A estrutura de carpetas debe quedar así:

```
/www  
main.js  
package-lock.json  
package.json
```

#### 2. Instala Capacitor

Instala as dependencias de Capacitor:

```
npm install @capacitor/core  
npm install @capacitor/cli --save-dev
```

#### 3. Inicializa Capacitor

Lanza o asistente de configuración de Capacitor:

```
npx cap init
```

Preguntarache algúns datos:

- **Name:** O nome da túa aplicación (por exemplo, WeatherApp).
- **Package ID:** Un identificador único en formato de dominio invertido (por exemplo, com.teuusuario.weatherapp).
- **Web directory:** Escribe www (que é onde tes a túa web e que tamén é a opción por defecto).

**No caso de** que a túa aplicación estea feita cun **framework de JS (Svelte, Astro...)** edita o ficheiro *capacitor.config.json* ou *capacitor.config.ts*:

```
{  
  "webDir": "build"  
}
```

Ou

```
{  
  "webDir": "dist"  
}
```

#### 4. Engade a plataforma Android

Instala o paquete de Android e engade a plataforma:

```
npm install @capacitor/android
```



```
npx cap add android
```

me dice que www debe contener un index.html

Isto crea unha carpeta chamada android co proxecto nativo.

#### 5. Copia os ficheiros á plataforma

Cada vez que fagas modificacións na túa webapp ou executes un novo build (no caso dos frameworks de JS) executa:

```
npx cap copy
```

#### 6. Sincroniza os arquivos web coa app nativa

Cada vez que modifiques os ficheiros da túa web, executa:

```
npx cap sync
```

Esto copia el contenido de www al proyecto nativo de Android.

## 7. Abre o proxecto en Android Studio

Abre o proxecto nativo de Android con:

```
npx cap open android
```

Isto abrirá Android Studio. Dende aí podes compilar e executar a túa app nun emulador ou nun dispositivo físico.

## 8. Executa e proba a túa app

En Android Studio:

1. Escolle un emulador ou conecta o teu dispositivo.
2. Preme no botón de "Run" (play verde).
3. Verás a túa web do tempo funcionando como unha app Android.

### Resumo visual do fluxo

1. **npm init -y**
2. **npm install @capacitor/core @capacitor/cli --save-dev**
3. **npx cap init**
4. **npm install @capacitor/android**
5. **npx cap add android**
6. **npx cap copy**
7. **npx cap sync**
8. **npx cap open android**
9. **Executa a app dende Android Studio**

### Consellos finais

- Podes personalizar iconas e *splash* na carpeta `android/app/src/main/res`.
- Consulta a [documentación oficial de Capacitor](#) para engadir funcionalidades nativas.

**Entrega:** Una vez completado todo **subiremos la carpeta *Ejercicio79*** al espacio para entrega de ejercicios que tenemos en el OneDrive de Microsoft Teams.

## Ejercicio 79: Convierte tu web del tiempo en una app Android con Capacitor JS

**Objetivo:** Crear una aplicación para Android (.apk) usando Capacitor JS.

### Requisitos previos:

- Node.js
- npm (viene con Node.js en la instalación por defecto)
- Tu proyecto web ya creado (puede ser con Svelte, Astro o HTML/CSS/JS puro)
- Editor de código (VS Code)

### 1. Inicializa el proyecto con npm

Abre una terminal en la carpeta raíz de tu proyecto (o crea una carpeta www y mueve allí los archivos de tu proyecto) y ejecuta:

```
npm init -y
```

Esto creará un archivo package.json necesario para gestionar dependencias.

- Crea en la raíz del proyecto un fichero **main.js**
- En el **package.json** es necesario cambiar la dirección del **main** por **main.js**

A estructura de carpetas debe quedar así:

```
/www  
main.js  
package-lock.json  
package.json
```

### 2. Instala Capacitor

Instala las dependencias de Capacitor:

```
npm install @capacitor/core  
npm install @capacitor/cli --save-dev
```

### 3. Inicializa Capacitor

Lanza el asistente de configuración de Capacitor:

```
npx cap init
```

Te pedirá algunos datos:

- **Name:** El nombre de tu aplicación (por ejemplo, WeatherApp).
- **Package ID:** Un identificador único en formato de dominio invertido (por ejemplo, com.tuusuario.weatherapp).
- **Web directory:** Escribe `www` (que es donde tienes tu web).
- **En el caso de** que tú aplicación esté hecha con un **framework de JS (Svelte, Astro...)** edita el archivo `capacitor.config.json` o `capacitor.config.ts`:

```
{  
  "webDir": "build"  
}
```

O

```
{  
  "webDir": "dist"  
}
```

#### 4. **Añade la plataforma Android**

Instala el paquete de Android y añade la plataforma:

```
npm install @capacitor/android  
npx cap add android
```

Esto crea una carpeta llamada android con el proyecto nativo.

#### 5. **Copia los archivos a la plataforma**

Cada vez que hagas modificaciones en tu webapp o ejecutes un nuevo *build* (en el caso de los frameworks de JS) ejecuta:

```
npx cap copy
```

#### 6. **Sincroniza los archivos web con la app nativa**

Cada vez que modifiques los archivos de tu web, ejecuta:

```
npx cap sync
```

Esto copia el contenido de `www` al proyecto nativo de Android.

## 7. Abre el proyecto en Android Studio

Abre el proyecto nativo de Android con:

```
npx cap open android
```

Esto abrirá Android Studio. Desde ahí puedes compilar y ejecutar tu app en un emulador o en un dispositivo físico.

## 8. Ejecuta y prueba tu app

En Android Studio:

- Elige un emulador o conecta tu dispositivo.
- Pulsa el botón de "Run" (play verde).
- Verás tu web del tiempo funcionando como una app Android.

## Resumen visual del flujo

1. `npm init -y`
2. `npm install @capacitor/core @capacitor/cli --save-dev`
3. `npx cap init`
4. `npm install @capacitor/android`
5. `npx cap add Android`
6. `npx cap copy`
7. `npx cap sync`
8. `npx cap open android`
9. Ejecuta la app desde Android Studio

## Consejos finales

- Si cambias algo en tu web, recuerda ejecutar `npx cap sync` antes de volver a compilar.
- Puedes personalizar iconos y *splash* en la carpeta `android/app/src/main/res`.
- Consulta la documentación oficial de Capacitor para añadir funcionalidades nativas.

## Entrega:

Una vez completado todo, sube la carpeta **Ejercicio79** al espacio para entrega de ejercicios que tenemos en el OneDrive de Microsoft Teams.