

Taller formativo

# Taller CRUD Básico con Ionic y Vue

---



I.E.S. SERRA PERENXISA



Autor: Sergi García

Actualizado Junio 2024




## Licencia



**Reconocimiento - No comercial - Compartir Igual (BY-NC-SA):** No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se ha de hacer con una licencia igual a la que regula la obra original.

## Nomenclatura

A lo largo de este tema se utilizarán diferentes símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:

 **Importante**

 **Atención**

 **Interesante**

## ÍNDICE

1. ¿Qué es este documento?	3
2. ¿Qué es un CRUD?	3
3. ¿Qué es Ionic?	3
4. ¿Qué es Vue?	3
5. ¿Qué tiene que ver Vue con Ionic?	4
6. ¿Qué es Capacitor?	4
7. ¿Qué necesitamos para poner en marcha este taller?	4
8. Al crear nuestro proyecto hay muchos ficheros y carpetas. ¿Cuáles utilizaremos y para qué sirve cada uno?	4
9. ¿Dónde podemos encontrar información de los componentes existentes en Ionic?	5
10. Repositorio con el proyecto completo comentado	5
11. Testeo de la aplicación y despliegue web y Android	5
12. Comandos más típicos de Ionic	5

## TALLER CRUD SIMPLE CON IONIC Y VUE

### 1. ¿QUÉ ES ESTE DOCUMENTO?

Este documento es una guía introductoria para desarrollar aplicaciones web y móviles multiplataforma utilizando Ionic y Vue.js. La guía cubre los conceptos básicos del desarrollo de aplicaciones CRUD (Crear, Leer, Actualizar, Eliminar) y proporciona instrucciones paso a paso sobre cómo configurar el entorno de desarrollo y comenzar con un proyecto de ejemplo.

### 2. ¿QUÉ ES UN CRUD?

CRUD son las siglas de Create, Read, Update y Delete. Estos son los cuatro tipos básicos de operaciones que se pueden realizar en una base de datos. A continuación se detallan cada uno de estos conceptos:

- **Create (Crear):** Se refiere a la acción de añadir nuevos registros o datos en la base de datos. Esto implica la inserción de nueva información en una tabla específica.
- **Read (Leer):** Se refiere a la acción de recuperar o visualizar los datos existentes en la base de datos. Esta operación permite obtener información de una o varias tablas para ser mostrada o utilizada en la aplicación.
- **Update (Actualizar):** Se refiere a la acción de modificar los datos existentes en la base de datos. Esta operación permite cambiar la información de uno o varios registros, manteniendo la consistencia y precisión de los datos.
- **Delete (Eliminar):** Se refiere a la acción de remover datos existentes de la base de datos. Esta operación permite eliminar uno o varios registros, liberando espacio y manteniendo la base de datos actualizada y relevante.

### 3. ¿QUÉ ES IONIC?

Ionic es un framework de desarrollo de aplicaciones móviles que permite crear aplicaciones multiplataforma utilizando tecnologías web estándar como HTML, CSS y JavaScript. Fue creado para simplificar el proceso de desarrollo de aplicaciones móviles, permitiendo a los desarrolladores escribir una sola base de código que pueda ser desplegada tanto en iOS como en Android, así como en aplicaciones web progresivas (PWA). Las Aplicaciones Web Progresivas (PWA) son aplicaciones web que se comportan como aplicaciones nativas. Utilizan tecnologías modernas para ofrecer una experiencia rápida, confiable y atractiva en cualquier dispositivo.

### 4. ¿QUÉ ES VUE?

Vue.js es un framework progresivo de JavaScript para construir interfaces de usuario. Se centra en la capa de vista (la parte de la aplicación que el usuario ve e interactúa) y es fácil de integrar con otros proyectos o bibliotecas. Vue es conocido por ser:

- **Reactivo:** Reactualiza automáticamente la interfaz de usuario cuando los datos subyacentes cambian.
- **Componentes:** Utiliza una arquitectura basada en componentes, lo que facilita la reutilización de código y la gestión de interfaces complejas.
- **Sencillo y flexible:** Ofrece una curva de aprendizaje suave y puede escalarse para usar en aplicaciones grandes con herramientas adicionales.

## 5. ¿QUÉ TIENE QUE VER VUE CON IONIC?

Ionic, originalmente construido para trabajar con Angular, ha expandido su compatibilidad para incluir otros frameworks modernos, incluyendo Vue.js. Esto significa que puedes utilizar Vue.js como el framework de frontend dentro de una aplicación Ionic. A lo largo de este taller, usaremos Vue junto con Ionic.

Más información de como empezar con Vue en Ionic en:

<https://enappd.com/blog/introduction-to-ionic-vue-beginners-guide/176/>

## 6. ¿QUÉ ES CAPACITOR?

Capacitor es un framework de desarrollo de código abierto desarrollado por el equipo de Ionic, diseñado para facilitar la creación de aplicaciones móviles multiplataforma utilizando tecnologías web estándar como HTML, CSS y JavaScript. A diferencia de su predecesor Cordova, Capacitor ofrece una arquitectura más moderna y flexible, proporcionando acceso simplificado a funcionalidades nativas del dispositivo a través de APIs unificadas. Esto permite a los desarrolladores integrar características como la cámara, el GPS y los sensores de manera eficiente en aplicaciones para iOS, Android y la web.

## 7. ¿QUÉ NECESITAMOS PARA PONER EN MARCHA ESTE TALLER?

Para poner en marcha este taller, necesitamos los siguientes elementos:

- Instalar NodeJS: <https://nodejs.org/en> y si tenéis algún sistema como Ubuntu, podéis usar <https://github.com/nodesource/distributions>
- Instalar Ionic: `npm install -g @ionic/cli`
- Instalar Vite: `npm install -g vite`
- Crear un proyecto de Ionic con la plantilla base Tabs, por ejemplo con el comando: `ionic start MusicaCRUD tabs --type=vue`
- Dentro del proyecto creado, entra e instala Ionic Storage: `npm install @ionic/storage`

## 8. AL CREAR NUESTRO PROYECTO HAY MUCHOS FICHEROS Y CARPETAS. ¿CUÁLES UTILIZAREMOS Y PARA QUÉ SIRVE CADA UNO?

**Los archivos y carpetas clave de nuestro proyecto con los cuales interactuaremos son los siguientes:**

- **src/index.ts**
  - Este archivo es el punto de entrada de la aplicación. En este caso, solo modificaremos los valores iniciales para incluir el fichero “src/storage.ts” que se incluye para cargar “ionic/storage” y para prepararlo al montar la aplicación.
- **src/storage.ts**
  - Fichero utilizado para cargar “ionic/storage”.
- **src/router.ts**
  - src/router.ts es el archivo donde se configuran las rutas de la aplicación. Define qué componente Vue se debe renderizar cuando el usuario navega a una URL específica, permitiendo la navegación entre las diferentes vistas de la aplicación.
- **view/TabsPage.vue**
  - view/TabsPage.vue es el componente que define la estructura de la página principal con pestañas (tabs) de la aplicación. Aquí se configuran y renderizan las diferentes

pestañas que permiten a los usuarios navegar entre las distintas secciones como "Home", "Acerca De" y "Canciones".

- **view/Home.vue**
  - view/Home.vue es el componente que representa la vista principal o de inicio de la aplicación. Generalmente, esta vista proporciona una visión general o una introducción a la aplicación.
- **view/AcercaDe.vue**
  - view/AcercaDe.vue es el componente que contiene información sobre la aplicación, sus desarrolladores, propósito, y cualquier otra información relevante que los usuarios necesiten saber sobre la aplicación.
- **view/Canciones.vue**
  - view/Canciones.vue es el componente dedicado al CRUD de canciones. Aquí se implementan las funcionalidades para crear, leer, actualizar y eliminar canciones, permitiendo a los usuarios gestionar su colección de música dentro de la aplicación.
- **directorio /dist**
  - El directorio /dist es la carpeta donde se generan los archivos de la aplicación optimizados para producción después de ejecutar el proceso de construcción. Contiene todos los archivos HTML, CSS y JavaScript que se necesitan para desplegar la aplicación en un servidor web.

## 9. ¿DÓNDE PODEMOS ENCONTRAR INFORMACIÓN DE LOS COMPONENTES EXISTENTES EN IONIC?

Podemos encontrar información y ejemplos de cada uno de los componentes ya creado por Ionic en la documentación oficial en <https://ionicframework.com/docs/components>.

## 10. REPOSITORIO CON EL PROYECTO COMPLETO COMENTADO

El repositorio completo de este taller, con comentarios de código para facilitar su entendimiento, está disponible en <https://github.com/sergarb1/MusicaCRUD>

## 11. TESTEO DE LA APLICACIÓN Y DESPLIEGUE WEB Y ANDROID

Para probar y generar una aplicación desarrollada con Ionic, ya sea para la web o para dispositivos Android, aquí tienes los pasos básicos:

- **Probar la Aplicación en Desarrollo (Web):** con el comando **"ionic serve"** se iniciará un servidor de desarrollo y abrirá automáticamente tu aplicación en un navegador web.
- **Crear el código distribuible web:** con el comando **"ionic build --pro"**, se construye la aplicación PWA y se almacena en el directorio **"/dist"**.
- **Generar y probar en Android:** con el comando **"ionic capacitor add android"** se añade dentro de tu proyecto, una carpeta **"/android"**. Abre esa carpeta con Android Studio y desde ahí puedes construir el APK con tu aplicación.
  - Cada vez que hagas cambios, construye de nuevo la aplicación con **"ionic build"** y tras ello, sincroniza la aplicación con el proyecto Android con **"ionic capacitor sync"**.

## 12. COMANDOS MÁS TÍPICOS DE IONIC

Ionic CLI (Command Line Interface) proporciona una serie de comandos útiles para crear, desarrollar y desplegar aplicaciones. Aquí se presentan algunos de los comandos más comunes que los desarrolladores utilizan regularmente.

- **Crear un Nuevo Proyecto**
  - `ionic start MyApp tabs --type=vue`
- **Servir la aplicación para probar y depurar**

- ionic serve
- **Construir la aplicación para Producción**
  - ionic build
- **Añadir una plataforma (iOS o Android)**
  - ionic capacitor add ios
  - ionic capacitor add android
- **Sincronizar el proyecto con Capacitor**
  - ionic capacitor sync
- **Abrir la aplicación en Xcode o Android Studio**
  - ionic capacitor open ios
  - ionic capacitor open android