Arquitectura del Sistema - Aplicación Móvil

Sistema VASPA

VASPA Team

Fabricio González





Tabla de contenido

[Introducción 4](#_Toc53082741)

[Arquitectura del Sistema – Diagrama 4](#_Toc53082742)

[Arquitectura del Sistema – Explicación 4](#_Toc53082744)

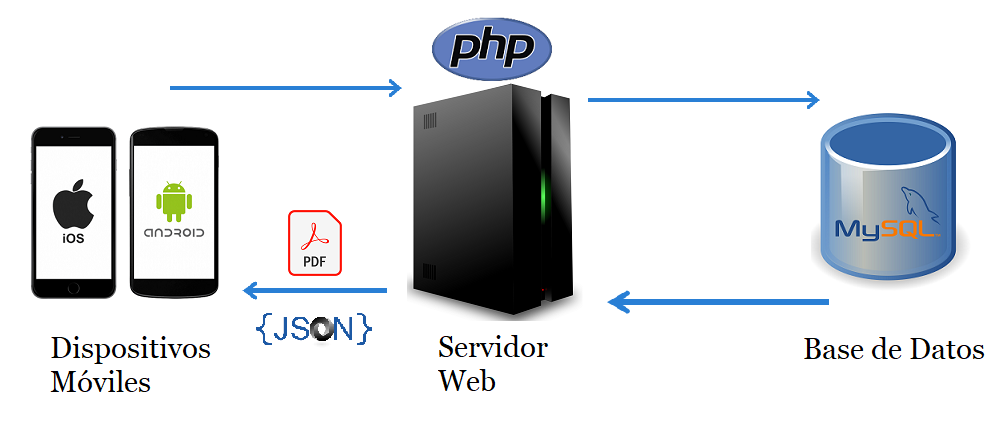
Arquitectura del Sistema - Aplicación Móvil

Introducción

En el presente documento, se exhibirán detalles de implementación de la aplicación móvil que serán muy útiles el día en que el Sistema VASPA se encuentre siendo utilizado en la unidad académica y se deba adecuar la aplicación móvil para que también sea utilizada.

Se presenta un diagrama que refleja de forma sencilla como es el funcionamiento de la aplicación y en la siguiente sección se explicará con más detalle lo que se ha utilizado en el desarrollo, el porqué y cómo funciona.

Arquitectura del Sistema – Diagrama



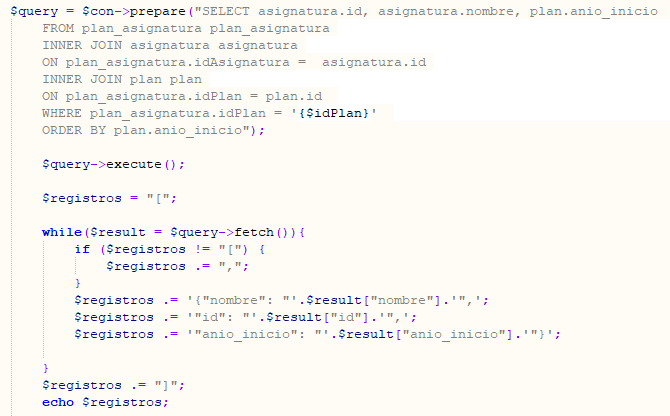
Arquitectura del Sistema – Explicación

La aplicación móvil fue desarrollada en el Framework Ionic en su versión 5 (Magnesium) con el backend en Node.js versión 12.18.3. Se utilizo la última versión disponible de cada componente para que el código fuente desarrollado esté lo más actualizado posible. Recordemos que durante la cursada de la asignatura Laboratorio de Desarrollo de Software en 2018 Ionic se encontraba en su versión 3.

Sin embargo, teniendo sólo lo mencionado la aplicación móvil no funcionaría ya que sólo manejaría datos internos. Por lo tanto, creemos que es necesario explicar cómo se ha llevado a cabo el desarrollo para que la aplicación funcione como lo hace.

Como puede observarse en el diagrama, la aplicación está basada en un servidor web con PHP. En él, se encuentran unos archivos PHP que funcionan como API para conectarse a la Base de Datos del sistema.

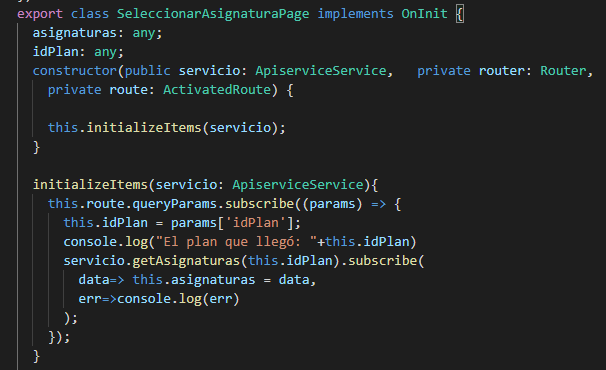
Ejemplo de parte de un archivo PHP que sirve como API para conectar la aplicación con la base de datos:



Puede observarse que se realiza una consulta SQL con algún parámetro y luego esto es devuelto en formato JSON.

Luego, la aplicación móvil con un servicio programado en TypeScript puede decodificar esta información y mostrarla vía HTML en la aplicación móvil (lo que visualiza el usuario).

Por ejemplo, parte de la clase en TypeScript que recupera las asignaturas se ve así:

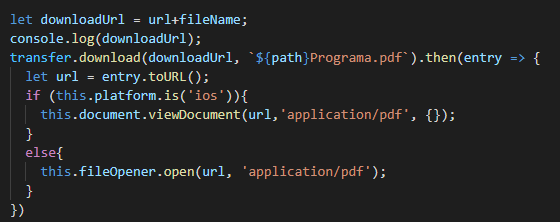


Puede observarse que se realiza una “suscripción” al servicio que obtiene los datos en JSON desde la API desarrollada en PHP. Esto quiere decir que la información es dinámica, por lo cual, si la base de datos del sistema cambia su estado durante la ejecución de la aplicación móvil, esta tendrá información actualizada en tiempo real.

Hasta este punto, se ha explicado todo lo vinculado a la recuperación de la información de la base de datos del sistema. Por lo que restaría explicar la recuperación, descarga y visualización del programa en PDF.

Por como se ha desarrollado el sistema, los programas en PDF estarán almacenados en distintos directorios del sistema, organizados por año y con un estándar en cuanto a los nombres. Por este motivo, la recuperación de este archivo PDF no es muy compleja. Ya que, a medida que el usuario interactúa con la aplicación se envían parámetros entre una pantalla y la siguiente. Luego, con esta información, se puede armar la ruta completa para obtener dicho archivo.

A continuación, se muestra parte del método para la descarga y visualización del programa en PDF:



En este caso, ya se ha armado la ruta del archivo PDF en la variable downloadUrl. Luego, a través del método transfer.download() se realiza la descarga. Y, finalmente, dependiendo de si es un dispositivo móvil con iOS o Android, se llama a distintos métodos para la visualización del programa en PDF.