**UNIVERSIDAD NACIONAL**

**MAYOR DE SAN MARCOS**

****

**Universidad del Perú,**

**DECANA DE AMÉRICA**

Facultad de Ingeniería de Sistemas e Informática

E.A.P: Ingeniería de Software

Documento de Arquitectura

Curso: Gestión de la Configuración y Mantenimiento

Profesora: Lenis Rossi Wong Portillo

Integrantes: GRUPO 6

**Documento de Arquitectura de Software**

1. **Descripción del sistema**

Sistema que se encarga de mostrar un menú de manera digital, mediante una interfaz gráfica amigable para el usuario, creado con tecnologías como Laravel, PHP en la parte del backend, así como Vue.js en el front-end y MySQL para la base de datos.

1. **Funcionalidades**

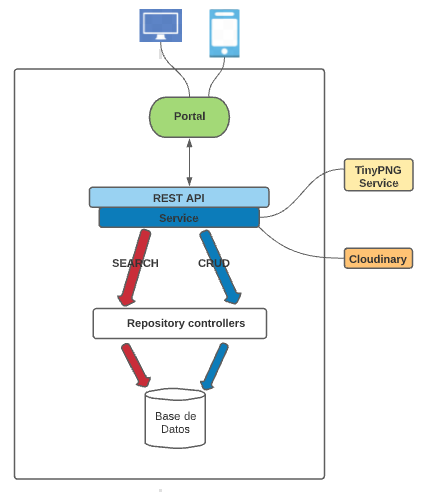
**2.1 Mostrar menúes**

El administrador se encargará de gestionar la carta subida al sistema para ser mostrada a los usuarios. Estos cambios pueden ser hechos únicamente por el administrador del sistema y pueden accionarse cada que éste lo requiera.

**2.2 Manipulación de promociones**

El sistema permitirá al administrador la manipulación de promociones para los diversos platos ofertados en nuestra carta. Esta edición podrá ser hecha únicamente por el administrador.

1. **Arquitectura general del sistema**



Definimos nuestra arquitectura general como cliente/servidor en la cual las siguientes entidades como: administrador se puede conectar a la aplicación ya sea vía web o móvil, para darle una mejor usabilidad.

De lado del cliente se usará el framework Vue.js. El cliente consumirá los servicios en el lado del servidor que estará desarrollado con el Framework Laravel, donde estará alojado nuestros módulos CRUD. El servidor estará conectado a la base de datos MySQL a través de la comunicación vía Eloquent.

**Patrón Arquitectónico**

En este proyecto usaremos el patrón arquitectónico cliente/servidor; sin embargo, las tecnologías que usaremos del lado del cliente(Vue.js) y del servidor(Laravel) usan internamente el patrón MVC.

El objetivo de esta arquitectura es mejorar la usabilidad de la aplicación manteniendo el núcleo funcional independiente de la interfaz de usuario, además de permitir que el proyecto sea escalable.

1. **Tecnologías y servicios a utilizar.**

* Vue.js (Frontend)



* Laravel (Backend)



* Vercel / 23.0.1 / Vercel



* MySQL 5.7



* Api de Tinypng / v1.11.0 / Mapbox

