



DOCUMENTO DE ARQUITECTURA SOFT SOLUTIONS

Jair Felipe Sánchez

Michel Beltrán

Daynover Zapata

UNIMINUTO

2022

Documento de Arquitectura del Software

1. Introducción

En condiciones ideales, el 100% de los bares del país (aparte de poder mantener su modelo de negocio prepandemia (COVID19) tendría la capacidad de comenzar a utilizar estrategias digitales que les permitan vender sus productos, llegando a más clientes potenciales y de no ser dependientes a la concurrencia de público que puedan llegar a perder por diferentes motivos, uno de ellos, la emergencia sanitaria por COVID19, que comenzó en el año 2020 en Colombia. Varias compañías optaron por utilizar el recurso de la tecnología con el fin de mejorar y optimizar sus actividades.

Con el desarrollo de una plataforma de gestión y administración, en donde los bares puedan registrar y realizar un inventario y ventas de productos, además descargar reportes de ventas en un periodo de tiempo, se pretende, en primer lugar, contribuir al avance tecnológico haciendo que los bares comiencen a utilizar y vean las ventajas que trae el uso de las plataformas digitales a sus empresas y, en segundo lugar, evitar que estas ya sea por desconocimiento o desconfianza, cierren por falta de alternativas para automatizar procesos y optimizar costos

2. Alcance

De acuerdo con la ejecución de las reuniones que se realizarán con los usuarios involucrados en los procesos de desarrollo del sistema, y con el patrocinador del proyecto del desarrollo de software, se ajustara el sistema a las necesidades del negocio las cuales se estima permitirán ejecutar los siguientes módulos:

- Login de usuarios
- Gestión de usuarios
- Gestión de reservas
- Gestión de productos
- Gestión de categorías
- Gestión de ventas
- Informe de ventas por fecha

Soft Solutions está diseñado para soportar dos tipos de impresiones: Impresiones a media carta e oficio.

3. Propósito

El documento define los objetivos de la arquitectura, los casos de uso admitidos por el sistema, los estilos arquitectónicos y los componentes seleccionados. El documento proporciona una justificación para las decisiones de arquitectura y diseño tomadas desde la idea conceptual hasta su implementación.

4. Documentos relacionados

Con el fin de manejar y tener conocimientos de los documentos manejados en el campo del desarrollo se realiza la siguiente matriz a continuación:

Título	Fecha	Organización
<i>Buenas Prácticas de Desarrollo</i>	<i>03/10/2022</i>	<i>Team Solution</i>
<i>Diagrama de Gantt</i>	<i>03/10/2022</i>	<i>Team Solution</i>

5. Objetivos

5.1 General

Presentar las especificaciones de diseño o requisitos que deberá cumplir el nuevo sistema de gestión y administración para bares

5.2 Específicos

- Definir los estándares de diseño que se seguirán en el desarrollo de la interfase de usuario de la aplicación.
- Describir las metodologías y tecnologías con las que será desarrollada la aplicación.
- Identificar y documentar el conjunto de casos de uso, que deberán servir como insumo para la fase de construcción del software.

6. Aplicación web

El software se desarrollará bajo una arquitectura MVC (Model-View-Controller), Modelo, Vista, Controlador por sus siglas en inglés. El objetivo principal de esta arquitectura es separar, de la forma más limpia posible las distintas capas de la aplicación, con especial atención a permitir un

modelo de domino limpio, facilidad de mantenimiento y evolución de las aplicaciones. Otros elementos importantes han sido la facilidad del despliegue y el empleo de las tecnologías disponibles en la actualidad. El patrón MVC nos ayuda a desacoplar la lógica de negocios de la interfaz de usuario, MVC separa los conceptos de diseño, y por lo tanto reduce la duplicación de código. MVC también ayuda a los desarrolladores con diferentes habilidades a enfocarse en sus habilidades principales y a colaborar a través de interfaces claramente definidas. Para cada una de las capas se han implementado patrones de diseño específicos, entre los que podemos mencionar.



Figura1. Patrón de diseño MVC

7. Seguridad

El sistema deberá ser seguro, partiendo del principio del acceso mínimo requerido. La aplicación implementará autenticación básica a través de:

- Autenticación: A través de usuario y contraseña para acceder al sistema.
- Autorización: acorde al perfil del usuario.
- Campos de auditoría: Para todas las tablas de movimiento se guardará el usuario y fecha de inserción y modificación de registros.

8. Herramientas utilizadas para el Desarrollo

Durante el desarrollo del proyecto y con el fin de llegar a un final satisfactorio se utilizarán las siguientes herramientas:

NOMBRE	DESCRIPCION
xampp	XAMPP es un paquete de software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script PHP y Perl

phpMyAdmin	phpMyAdmin es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas web, utilizando un navegador web.
GitHub	GitHub es una forja para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador.
Visual Studio Code	Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux, macOS y Web. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código.

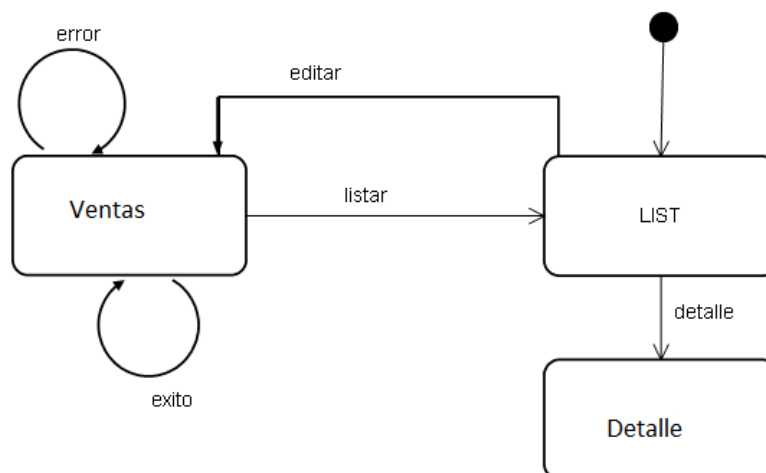
9. Estándares de desarrollo

9.1 Interfaces de usuario

Las interfaces de usuario se desarrollarán utilizando estándares de una tabla u operación CRUD(Create, Read, Update, Delete).

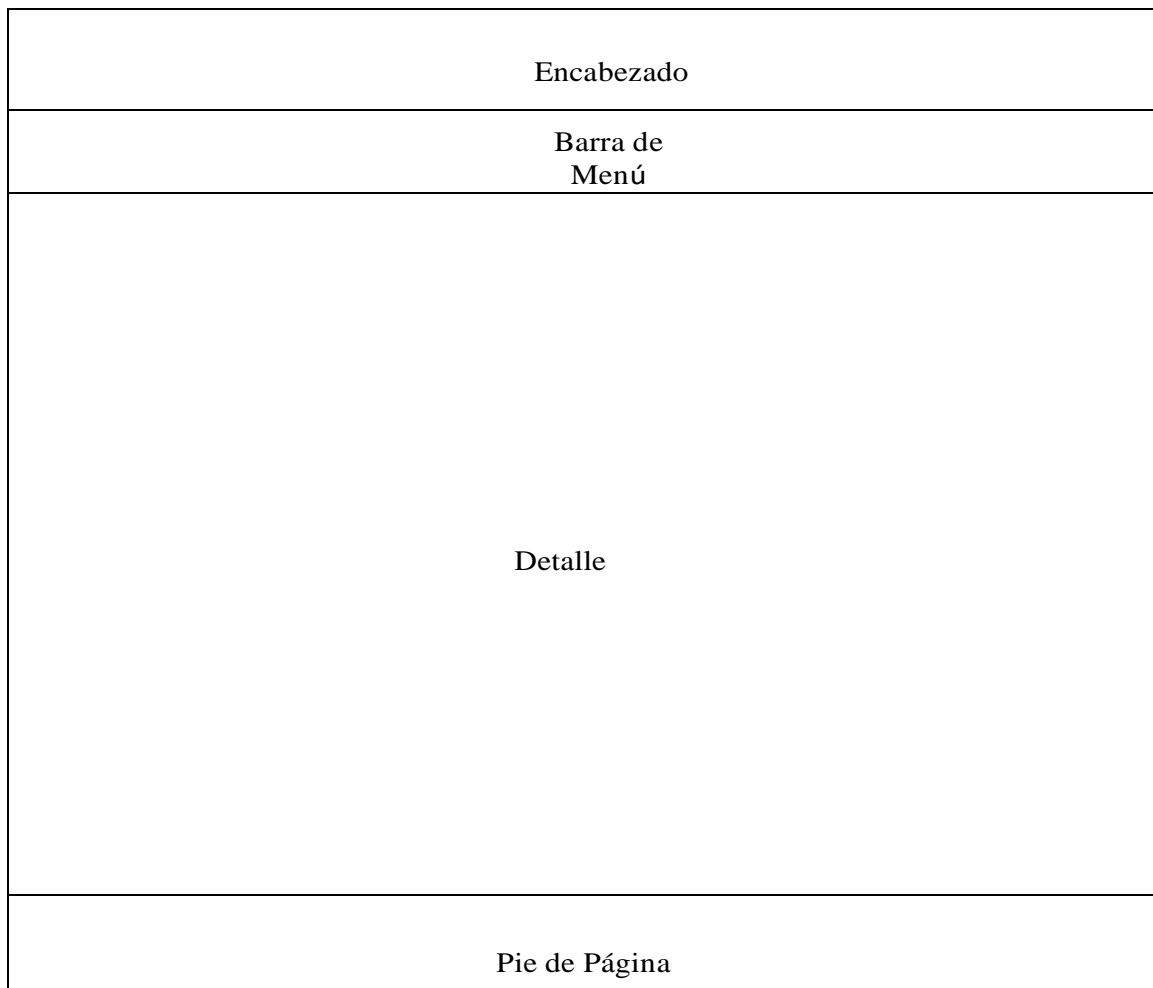
nombreTablaDetalle: Esta página servirá para realizar búsquedas filtradas por los campos que los usuarios consideren necesarios, a partir de ella se podrá ver el detalle de una venta, borrar un registro y cargar un registro en modo de edición al formulario del mantenimiento de la tabla.

nombreTablaVentas: Esta página es utilizada para insertar o modificar una venta o registro en la base de datos, por lo tanto podrá ser accedido en modo de inserción y/o edición,



9.2 Plantilla de sitio

Para el desarrollo de las páginas de todo el sitio, se utiliza el patrón de diseño CompositeView o Vista Compuesta, en el cual se define una plantilla general para el sitio, reutilizando los bloques de código comunes, como el encabezado, el menú, la barra de navegación y pie de página.



10. Casos de uso

- **Iniciar sesión**

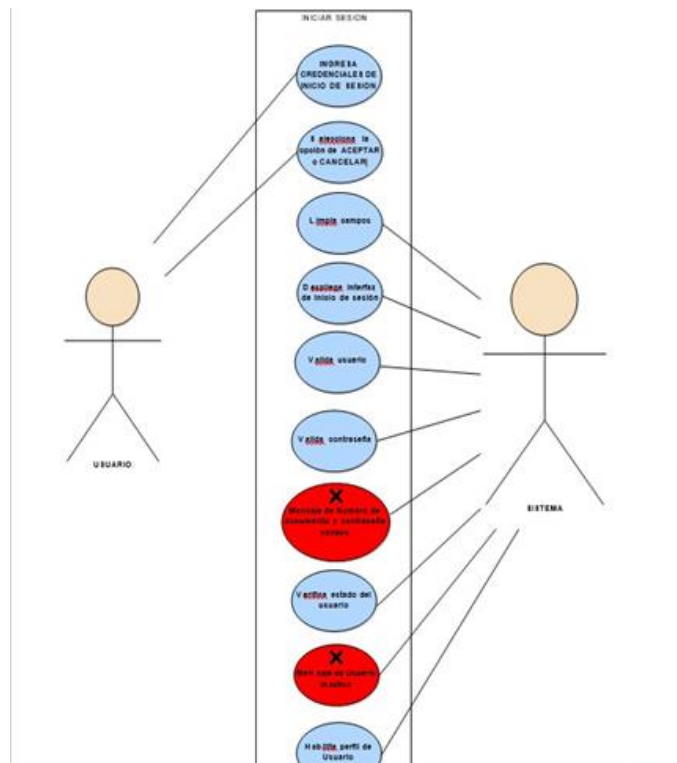


Figura 2. Caso de uso iniciar sesión

- **Validar usuario**

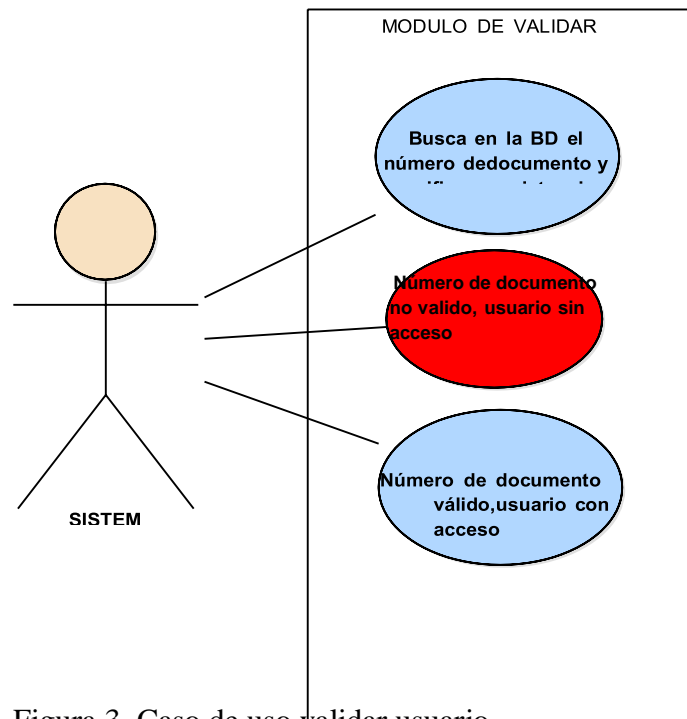


Figura 3. Caso de uso validar usuario

- **Iniciar aplicación**

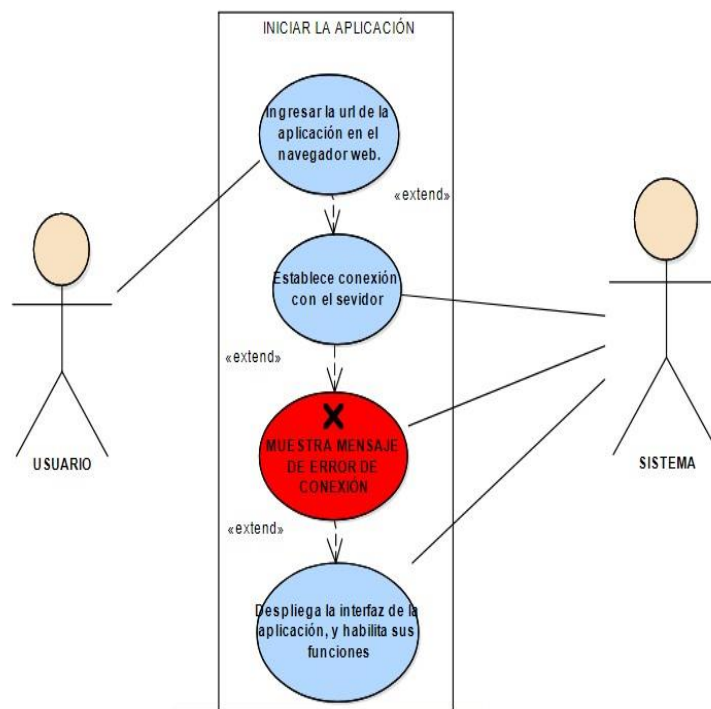


Figura 4. Caso de uso iniciar aplicación

- **Cambiar contraseña**

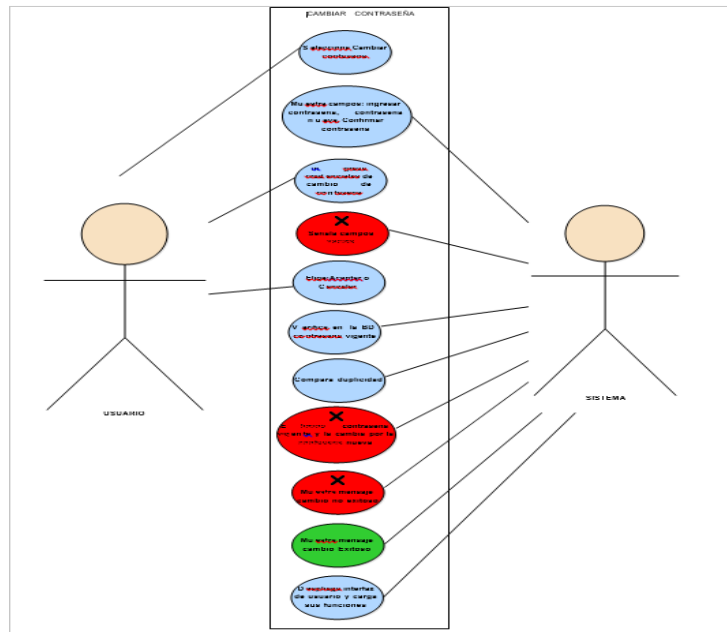


Figura 5. Caso de uso cambiar contraseña

- **Cerrar sesión**

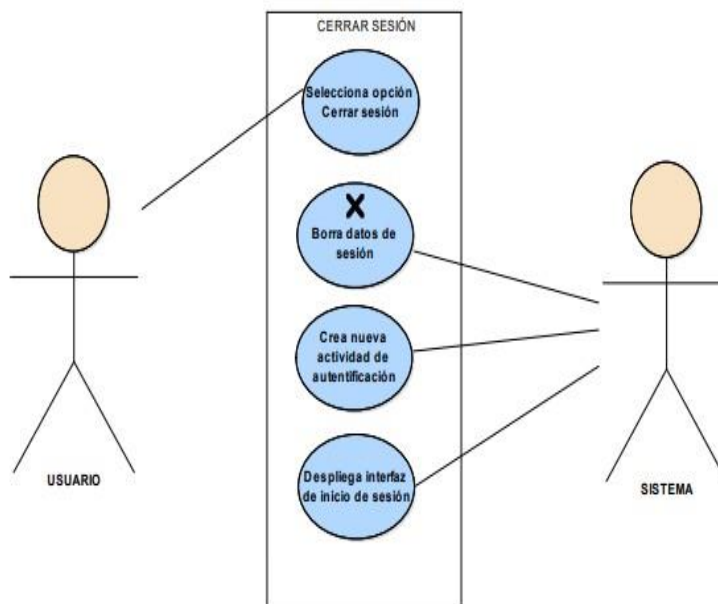


Figura 6. Caso de uso cerrar sesión

11. Descripción del modelo

11.2. Diagrama de entidad relación

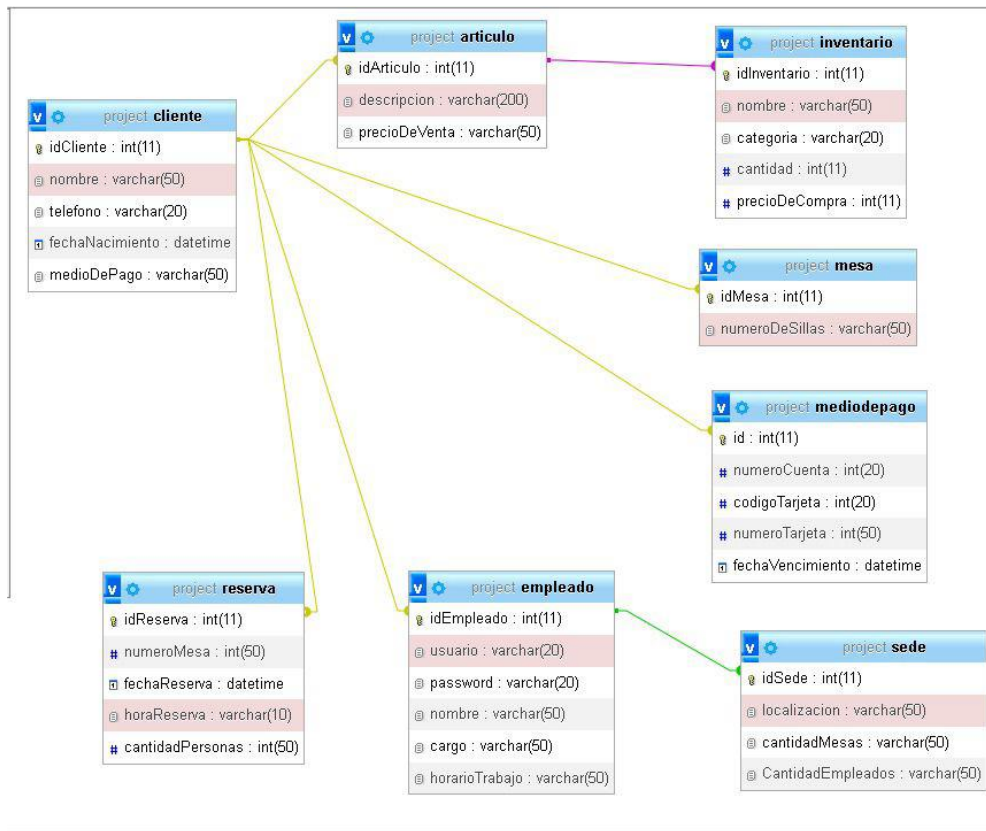


Figura 7. Diagrama de entidad-relación

11.3 Lista de actores

Nombre	Descripción
Cajero	Grupo de usuarios del nivel central con acceso a toda la información de todos los productos, categorías, reservas y ventas
Mesero	Grupo de usuarios asignados a los diferentes clientes. Únicamente tienen acceso a las reservas y realización de ventas.
Administrador	Usuario administrador del sistema. Encargado de toda la gestión de seguridad y administración de catálogos funcionales del sistema.

12. Vista de despliegue

La vista de despliegue presenta aspectos físicos como topología, infraestructura informática e instalación de ejecutables, necesarios para la puesta en marcha e implementación de la aplicación. A continuación, se presenta la vista de despliegue, en el cual se puede observar que es accedido por los usuarios a través de un navegador utilizando HTML.

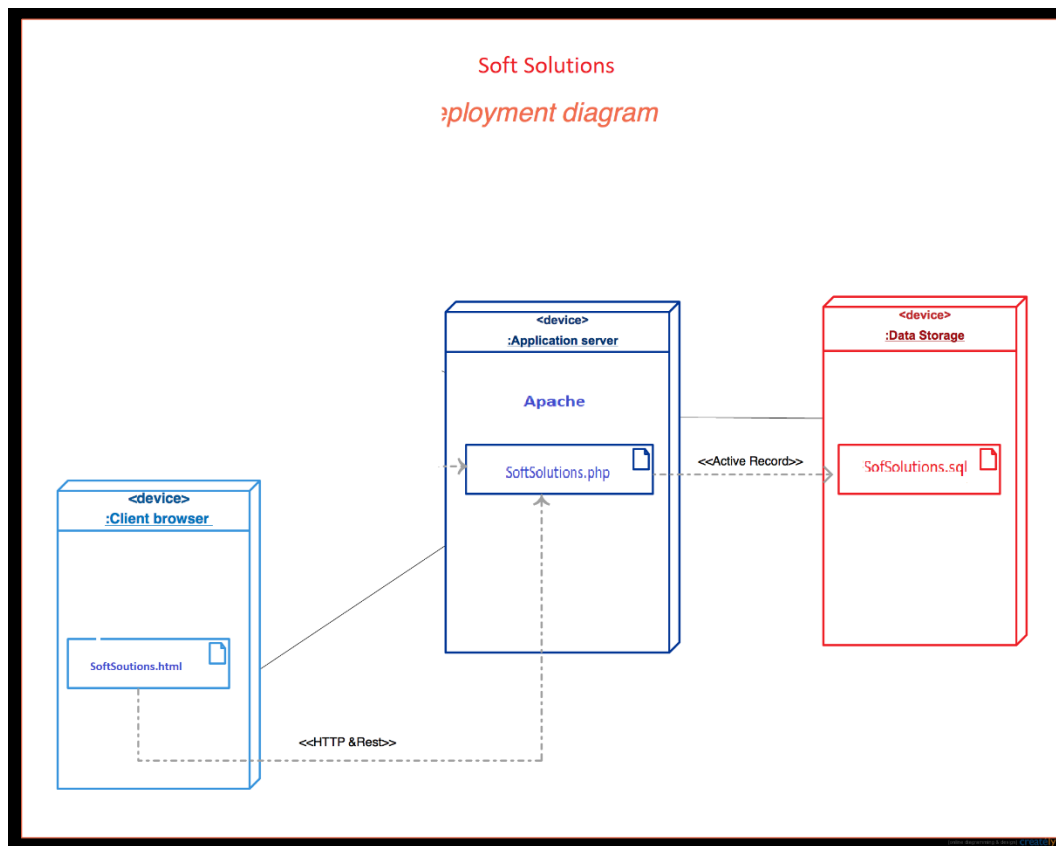


Figura 8. Diagrama de despliegue

12.1. Componentes

Servidor Web

Es un servidor de aplicaciones Apache, en éste residirá el componente web de la aplicación, el componente de servicios y lógica de negocios y el componente de acceso a datos. En esta aplicación se gestionará la lógica de negocios y transacciones, la cual será expuesta a través de servicios de negocios, que serán accedidos por el cliente web, a través del navegador HTML de la PC.

Servidor de Base de Datos

Servidor de base de datos relacional, que será utilizado para darle persistencia a los objetos del dominio que necesiten guardar sus estados, para su posterior consulta o modificación.

13. Vista de implementación

La Vista Implementación, describe los componentes de despliegue construidos y sus interacciones, además de sus dependencias con componentes de terceros utilizados para la construcción de la aplicación.

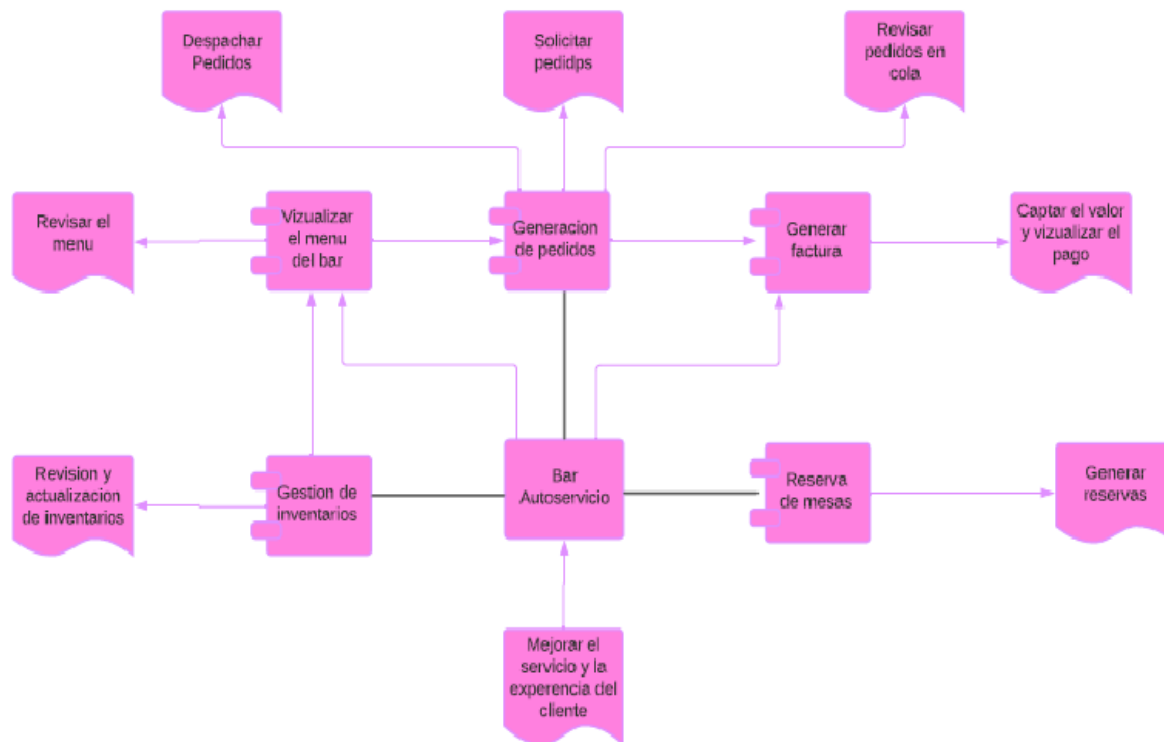
Diagrama de componentes

Figura 9. Diagrama de componentes

14. Metodología de desarrollo**Metodología tradicional**

Las metodologías tradicionales o rígidas en el desarrollo del software son aquellas que establecen una disciplina de trabajo sobre el proceso de desarrollo del software, con el propósito de alcanzar un software más eficiente.

Se caracterizan por definir y establecer total y rígidamente todos y cada uno de los requisitos al inicio de los proyectos de ingeniería de software. Estas metodologías son poco flexibles y no permiten realizar cambios.

El método tradicional funciona aplicando un enfoque lineal donde las etapas del transcurso de desarrollo del software deben complementarse secuencialmente. Es decir, una etapa debe completarse antes de que comience la siguiente, dichas etapas reúnen la recopilación de requisitos y documentación.