***“Año del Bicentenario del Perú: 200 años de Independencia”***

**UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS**

**(UNIVERSIDAD DEL PERÚ, DECANA DE AMÉRICA)**

**FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

***Escuela Profesional Académica de Ingeniería de Software***

**SISTEMA DE PLANIFICACIÓN ALIMENTICIA MEAL PLANNER (MP)**

***Documento de Arquitectura del Software (DAS)***

**EQUIPO 2**

**GESTIÓN DE LA CONFIGURACIÓN DEL SOFTWARE**

**DOCENTE:** Dra. Lenis Rossi Wong Portillo

**CICLO:** 6

**SEMESTRE:** 2021-II

**COORDINADORA:** Torres Talaverano, Luz Elena

**INTEGRANTES:**

Arango Quispe, Esmeralda 19200300

Blas Ruiz, Luis Aaron 19200069

Huarhuachi Ortega, Andrea Mariana 19200267

Palacios Barrutia, Jeanpiere Julian 19200274

Rojas Villanueva, Paula Elianne 19200266

Torres Berlanga, Christian 19200291

Torres Talaverano, Luz Elena 19200294

**LIMA – PERÚ**

**2021**

**HISTORIAL DE CAMBIOS**

| **Versión** | **Autor(es)** | **Descripción** | **Fecha** |
| --- | --- | --- | --- |
| ***1.0*** | **EQUIPO 2**  Palacios Barrutia, Jeanpiere  Torres Berlanga, Christian | Arquitectura de software  - Introducción  - Consideraciones generales | 01.11.21  08.11.21 |
| Rojas Villanueva, Paula | Diseño de software   * Introducción   - Impacto del software por módulos | 01.11.21  08.11.21 |

**ÍNDICE**

[**ARQUITECTURA DEL SOFTWARE**](#_zi0dzsbiwyry) **4**

[INTRODUCCIÓN](#_gdcaibn2k37s) 4

[Propósito](#_m8wj1lx5nkts) 4

[Alcance](#_qhktewmkakhi) 4

[Objetivos](#_6tw5rog8f5lt) 4

[CONSIDERACIONES GENERALES](#_jjovvdcz3u4n) 4

[Definición de la Arquitectura](#_bd46w76uvjvr) 4

[Tecnologías utilizadas](#_ryr4vlz7b6qn) 4

[Capas arquitectónicas](#_frj8t8csfa3a) 6

[**DISEÑO DE SOFTWARE**](#_pn1hql51vz8h) **7**

[INTRODUCCIÓN](#_jk3fuxkcp1vx) 7

[Propósito](#_fvum452ufbqo) 7

[Alcances](#_m6wusg85pu98) 7

[Objetivo](#_4mcr6lvisftu) 7

[IMPACTO DEL SOFTWARE POR MÓDULOS](#_lt6n9iohznci) 7

[Módulo Información de la página](#_52rkh8m4n8z) 8

[Interfaz de Usuario](#_fwn7998bnd7y) 8

[Módulo Autentificación de usuario](#_aq7e6rcdjo8x) 8

[Colecciones](#_9p7ye0js5xrk) 8

[Interfaz de Usuario](#_d0g6u432rtg4) 8

**DOCUMENTACIÓN DE ARQUITECTURA DEL SOFTWARE**

# **ARQUITECTURA DEL SOFTWARE**

## **INTRODUCCIÓN**

### **Propósito**

El presente documento tiene como propósito proporcionar una fuente de referencia para los analistas y diseñadores de la arquitectura de la aplicación.

### **Alcance**

* Definir las tecnologías, frameworks a utilizar en el proyecto Meal Planner.
* Desarrollo de la capa arquitectónica del proyecto Meal Planner.

### **Objetivos**

* Referencia para los analistas y desarrolladores con el fin de discernir si la arquitectura apoyará sus requisitos de la aplicación.
* El documento se actualizará de acuerdo a los cambios técnicos de arquitectura que vayan apareciendo en el transcurso del desarrollo. Se detalla los requerimientos de cambios que el grupo pueda tener.
* Si exigiera cambios en la arquitectura, el diseñador hará las modificaciones apropiadas a este documento.

## **CONSIDERACIONES GENERALES**

### **Definición de la Arquitectura**

Respecto a la arquitectura del software, se decidió trabajar en partes separadas el desarrollo del servidor y el sitio web, permitiendo flexibilidad y mejora de la eficiencia para el equipo de desarrollo.

### **Tecnologías utilizadas**

Las tecnologías usadas en el proyecto son:

* **Bootstrap:**

Es una biblioteca multiplataforma o conjunto de herramientas de código abierto para diseño de sitios web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como extensiones de JavaScript adicionales. A diferencia de muchos frameworks web, solo se ocupa del desarrollo front-end.

Hemos elegido esta herramienta ya que nos brinda muchas facilidades al momento de codear y una estructura bien organizada.

* **Node.js:**

Es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor (pero no limitándose a ello) basado en el lenguaje de programación JavaScript, asíncrono, con E/S de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google.

Se usará node.js porque nos permite reusar el lenguaje de js pero para operaciones con el servidor.

* **Mysql:**

Es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base de datos de código abierto más popular del mundo, y una de las más populares en general junto a Oracle y Microsoft SQL Server, todo para entornos de desarrollo web.

Se escogió este gestor debido a su fácil acceso ya que es de código abirto y a lo ligero que es comparado a los otros gestores.

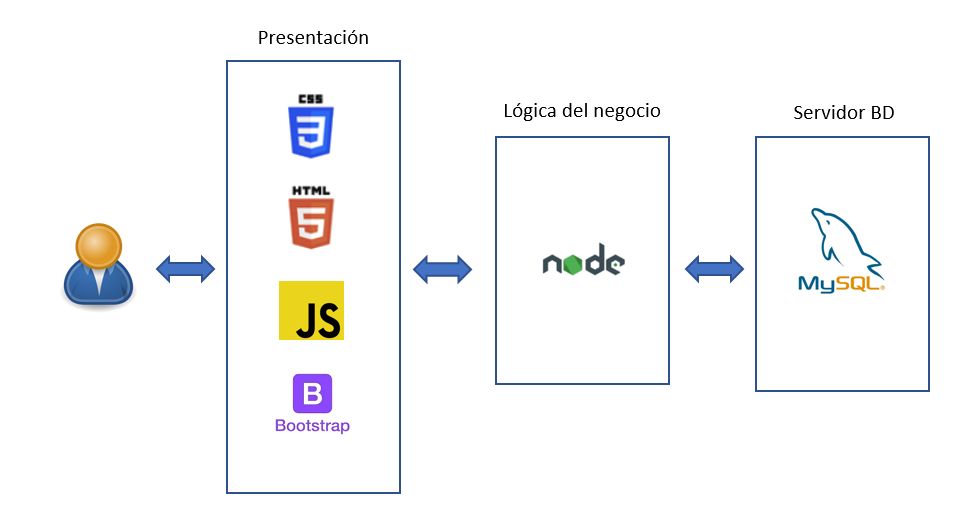
* **GIT:**

Es un software para rastrear cambios en cualquier conjunto de archivos, generalmente utilizado para coordinar el trabajo entre programadores que desarrollan código fuente en colaboración durante el desarrollo del software. Sus objetivos incluyen velocidad, integridad de datos y soporte para flujos de trabajo distribuidos y no lineales (miles de ramas paralelas que se ejecutan en diferentes sistemas).

Se está usando este software en conjunto con Github para el control de versiones de nuestro proyecto debido a su popularidad y facilidad de acceso.

### **Capas arquitectónicas**

El diagrama inicial de la arquitectura del sistema es el siguiente:



Meal Planner estará compuesto por las siguientes capas arquitectónicas:

* **Presentación:** Toda la parte de la presentación se realizará con HTML, CSS y JS. Además se hará uso de framework Bootstrap para facilitar el responsive. Hay que diferenciar entre vistas y componentes; las vistas son las diferentes ventanas de la página web (página principal, la sección de notas, la lista de usuarios…), en cambio, los componentes son pequeños trozos de código que se repiten en las diferentes vistas (barra de navegación, footer…). El script es el controlador, que manejará las distintas acciones (pulsación de botones, peticiones a la API).
* **Lógica de negocios**: El API se ha desarrollado en el entorno de ejecución de NodeJS; usando NestJS como framework para manejar las peticiones (procesamiento de las peticiones, aplicar los middlewares necesarios, etc.) a los diferentes endpoints.
* **Servidor BD:** Mediante MySql se creará la conexión con la base de datos y así, disponer de una interfaz para tratar con esta.

# **DISEÑO DE SOFTWARE** ………… guia del documento **ECI-DDS**

## **INTRODUCCIÓN**

### **Propósito**

El presente documento contiene un resumen de los componentes de diseño y la lista de impactos que tendrá la instalación del producto del proyecto en el sistema.

### **Alcances**

* Listar impactos por módulos y producto en la implementación del proyecto.
* Listar procesos y/o componentes del sistema que se verán afectados al momento de realizar la implementación del proyecto.

### **Objetivo**

Facilitar al área de sistemas para identificar con anticipación los procesos y/o procedimientos que se deben adecuar.

## **IMPACTO DEL SOFTWARE POR MÓDULOS**

Esta sección contiene una lista de procesos y/o componentes del sistema que se verán afectados al momento de realizar la implementación del proyecto. Esta información puede ser utilizada por el área de sistemas para identificar con anticipación los procesos y/o procedimientos que se deben adecuar.

Primero, se realiza un resumen del impacto por módulo y producto en la implementación del proyecto (ver la siguiente Tabla: Impacto por módulo y producto del proyecto Meal Planner)

*Impacto por módulo y producto del proyecto Meal Planner*

| PRODUCTO | MÓDULO | NIVEL DE IMPACTO |
| --- | --- | --- |
| MP | Información de la página | Medio |
| MP | Autenticación de Usuario | Alto |
| MP | Usuario Cliente | Alto |

Los niveles de impacto pueden ser: Alto, Medio y Bajo. Donde Alto indica un mayor impacto y bajo un impacto menor.

### **Módulo Información de la página**

### **Interfaz de Usuario**

Se creará una pantalla para que un usuario visitante puede acceder a información acerca de Meal Planner con esta estructura:

### **Módulo Autentificación de usuario**

#### **Colecciones**

Se creará la colección de usuarios (User document) para catalogar los datos de los usuarios.

| CAMPO | DATA\_TYPE | NULLABLE | COMMENTS |
| --- | --- | --- | --- |
| id | Int | No | id del usuario |
| Email | String | No | Correo del usuario |
| Password | String | No | Contraseña del usuario |
| Name | String | No | Nombre del usuario |
| Last\_name | String | No | Apellido del usuario |
| Phone | Int | No | Teléfono del usuario |
| Age | Int | No | Edad del usuario |
| Genere | String | No | Género del usuario |
| Profile | Object | No | Objeto contenedor de Dni, Name, LastName y Phone. |

#### **Interfaz de Usuario**

**Registro - Log up**

Se tendrá una pantalla para poder registrarse en Meal Planner con la siguiente estructura:

**Inicio sesión - Login**

Se creará una pantalla para que un usuario registrado puede acceder a Meal Planner: