



MATERIA

Ingeniería Web.

ALUMNO

*Fragoso Martínez Vanessa Alin
Santana Islas Gerardo Leonardo*

TEMA

Requerimientos Funcionales y No Funcionales

Grupo: 2TM3

Fecha de entrega:

16/06/23

Ficha del documento

Fecha	Revisión	Autor	Verificado dep. calidad.
16/06/23	2°	Fragoso Martínez Vanessa Alin. Santana Islas Gerardo Leonardo.	



Contenido:

FICHA DEL DOCUMENTO 2

CONTENIDO 2

1 INTRODUCCIÓN 5

- 1.1 Propósito 6
- 1.2 Alcance 6
- 1.3 Personal involucrado 6
- 1.4 Definiciones, acrónimos y abreviaturas 7
- 1.5 Referencias 10
- 1.6 Resumen 11

2 DESCRIPCIÓN GENERAL 12

- 2.1 Perspectiva del producto 12
- 2.2 Funcionalidad del producto 12
- 2.3 Características de los usuarios 13
- 2.4 Restricciones 13
- 2.5 Suposiciones y dependencias 13
- 2.6 Evolución previsible del sistema 14

3 REQUISITOS ESPECÍFICOS 14

- 3.1 Requisitos comunes de los interfaces 15
 - 3.1.1 Interfaces de usuario 15
 - 3.1.2 Interfaces de hardware 15
 - 3.1.3 Interfaces de software 15
 - 3.1.4 Interfaces de comunicación 16
- 3.2 Requisitos funcionales 16
 - 3.2.1 Requisito funcional 1 16
 - 3.2.2 Requisito funcional 2 16
 - 3.2.3 Requisito funcional 3 17



UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENIERIA
Y TECNOLOGIAS AVANZADAS – IPN



Ingeniería web
1er Departamental
Profr. POLANCO MONTELONGO FRANCISCO ANTONIO
Academia de Telemática

3.2.4 Requisito funcional n **¡Error! Marcador no definido.**

3.3	Requisitos no funcionales	17
3.3.1	Requisitos de rendimiento	17
3.3.2	Seguridad	17
3.3.3	Fiabilidad	18
3.3.4	Disponibilidad	18
3.3.5	Mantenibilidad	18
3.3.6	Portabilidad	18



1 introducción

El impacto de la Web ha propiciado la necesidad de utilizar una gran cantidad de tecnologías, librerías, herramientas y estilos para desarrollar una aplicación web de manera óptima.

Es necesario conocer los elementos más importantes que conforman este tipo de proyectos desde un punto de vista de alto nivel para tener una visión global de la programación web.

Existen dos enfoques en el desarrollo de aplicaciones web:

- Creación de aplicaciones web con tecnologías de desarrollo.
- Creación de aplicaciones web con sistemas gestores de contenido. Para realizar proyectos de esta índole, es necesario conocer las diferentes herramientas con las cuales podemos trabajar, al igual que las ventajas y desventajas de cada una, esto debido a que cada software representa una gran gama de funciones y grados de complejidad que pueden proporcionar un ambiente más proporcional.

Para esto debemos conocer las tecnologías para el desarrollo de estas aplicaciones:

- Tecnologías de cliente: Tecnologías que permiten crear interfaces de usuario atractivos y permiten la comunicación con el servidor. Basadas en HTML, CSS y JavaScript.
- Tecnologías de servidor: Tecnologías que permiten implementar el comportamiento de la aplicación web en el servidor: lógica de negocio, generación de informes, compartir información entre usuarios, envío de correos, etc.
- Bases de datos: La gran mayoría de las webs necesitan guardar información. Las bases de datos son una parte esencial del desarrollo web.

Estas tecnologías han proporcionado soluciones a necesidades de empresas las cuales mediante páginas web, dan a conocer sus productos o servicios, al igual que entretenimiento.

Para este proyecto, se proporcionará una página web orientada al servicio gastronómico proporcionado por un restaurante el cual, dará a conocer los platillos de su menú.

Esto se hace con el propósito de realizar un crecimiento del restaurante y obtener mayor clientela al igual que realizar marketing.



1.1 Propósito

El propósito de este proyecto es poner a prueba los conocimientos adquiridos en esta materia al igual que desarrollar la lógica necesaria para utilizar las tecnologías con el fin de resolver las problemáticas que se puedan presentar en el desarrollo de la aplicación web.

El propósito de este documento es dar a conocer el desarrollo del proyecto y establecer los requerimientos del proyecto.

Este documento está dirigido al profesor Polanco Montelongo Francisco Antonio.

1.2 Alcance

El alcance de este proyecto está dirigido un público con problema de organización en sus tareas del día a día, esto causando una solución al proporcionar una herramienta que permita optimizar los tiempos.

Sin embargo, también se planea el uso de esta aplicación en un ámbito empresarial en el cual se puede utilizar para distribuir la carga de trabajo entre los colaboradores.

- Las principales funciones son la creación de tareas.
- La implementación de un CRUD con el cual se realizará la manipulación de las tareas.

1.3 Personal involucrado

Nombre	Fragoso Martínez Vanessa Alin
Rol	Desarrollador Full Stack
Responsabilidades	Desarrollo de Frontend y Backend

Tabla 1. Personal.

Nombre	Santana Islas Gerardo Leonardo
Rol	Desarrollador full-stack
Responsabilidades	Desarrollo de Frontend y Backend

Tabla 2. Personal.

Relación de personas involucradas en el desarrollo del sistema, con información de contacto.

Esta información es útil para que el gestor del proyecto pueda localizar a todos los participantes y recabar la información necesaria para la obtención de requisitos, validaciones de seguimiento, etc.



1.4 Definiciones, acrónimos y abreviaturas

- **WWW:** (*World Wide Web*): red informática mundial. Es la red que contiene las páginas web existentes en Internet.
- **W3C:** (*World Wide Web Consortium*): consorcio WWW. Es el organismo internacional que crea y determina los estándares web. Podríamos considerarlo una especie de agencia normalizadora.
- **RWD:** (*Responsive Web Design*): diseño web adaptable. Permite que las páginas se ajusten a la medida de cualquier pantalla. Especialmente útil utilizando tablets o móviles libres.
- **ISP:** (*Internet Service Provider*): proveedor de servicios de internet: cada una de las empresas que ofrece servicios de alojamiento, alquiler de dominios y demás similares.
- **URL:** (*Uniform Resource Locator*): localizador de recursos uniforme. La forma de asignar nombre a un recurso de Internet para diferenciarlo de todos los demás, es decir, cada dominio, cada página web en particular, cada buzón de correo electrónico, etc.
- **FTP:** (*File Transfer Protocol*): protocolo de transferencia de ficheros. Es el sistema que se utiliza para publicar (también llamado “subir”) en Internet los archivos que componen una web.
- **CMS:** (*Content Management System*): sistema de gestión de contenidos. Se trata de aplicaciones que se ejecutan en Internet (online) y que son utilizadas para crear webs, modificarlas, administrarlas, etc. El ejemplo más conocido de CMS es WordPress.
- **TLD:** (*Top Level Domain*): dominio de nivel superior. Hace referencia al dominio principal de una web, como en el caso de semymas.com, donde ponemos al alcance de las pymes servicios de posicionamiento.
- **HTTP:** (*HyperText Transfer Protocol*): protocolo de transferencia de hipertexto. Es el protocolo de comunicación mediante el cual solicitamos que se abra una web en nuestro navegador y, a su vez, esta nos es proporcionada.
- **HTML:** (*HyperText Markup Language*): lenguaje de marcas de hipertexto en el que se escribe la estructura básica de las webs, así como su contenido. Para el diseño de las diferentes partes, así como de tipografías y colores, se usa CSS.



- **CSS:** (*Cascade Style Sheet*): hojas de estilo en cascada. Es el lenguaje que da forma y color a las páginas web, determinando cómo se mostrarán en pantalla.
- **UI:** (*User Interface*): entorno de usuario. Hace referencia al aspecto visual de los componentes de cualquier sistema, incluidos los sitios web, así como lo interactivos que son con los usuarios.
- **API:** (*Application Programming Interfaces*): entornos de programación de aplicaciones. Son conjuntos de funciones y algoritmos ofrecidos por los programadores para que puedan ser utilizados por otros como base para la creación de nuevas aplicaciones o códigos.
- **DOM:** Document Object Model. Es el mecanismo que usan los navegadores para modelar el conjunto de nodos del HTML de las páginas web. El DOM representa todo el árbol de nodos que tiene una página web.
- **IDE:** Integrated Development Environment. Programa que usan los desarrolladores para escribir código y que, a diferencia de un editor de textos, contiene una serie de herramientas que ayudan al programador a escribir más y mejor código.
- **JSON:** JavaScript Object Notation. JSON es una forma de modelar datos muy habituales en Javascript pero que también se usa para compartir datos entre servicios y aplicaciones. El JSON se modela como un objeto clave-valor con dobles comillas, similar a la de los propios objetos de Javascript.
- **MEAN:** MongoDB, Express.js, Angular, Node.js: MEAN es un stack específico de librerías para construir webs que fue muy popular hace años. La gracia de este stack es que usa para todo el lenguaje de Javascript, el problema es que hoy en día Angular usa Typescript en lugar de Javascript
- **MVC:** Modelo, Vista, Controlador. Es un patrón de diseño muy usado en programación de objetos y que su misión es la de separar la representación de los datos de la propia lógica de estos.
- **REST:** Representational State Transfer. Este término es usado habitualmente con el de API. En concreto hablamos de API REST a una serie de rutas (endpoints) a las que la web (o otros servicios) mandan peticiones para adquirir o modificar datos.
- **WYSIWYG:** What You See Is What You Get. Esto se refiere a los típicos editores que hay en páginas webs y programas en el que puedes redactar texto con formato (enlaces, títulos, listas...) sin



tener que escribir código o etiquetas. Suele ser típico de foros, páginas para escribir blogs, docs tipo Google Drive, etc.

- **GUI:** Graphical User Interface. Interfaz gráfica de usuario. Es la parte gráfica que permite a los usuarios interaccionar con los sistemas de forma visual.
- **CLI:** Command User Interface. A diferencia de GUI esto significa que se interactúa con el sistema usando comandos en una terminal.
- **CTA:** Call To Action: Típicos botones grades que hay en muchas webs para incitar al usuario a hacer click.
- **CRUD:** Create, Read, Update, Delete. Las 4 funciones más básicas que se pueden implementar para una API o una base de datos. La idea es crear función de crear, leer, modificar o borrar un dato.
- **CDN:** Content Delivery Network. Una red de servidores normalmente distribuidos en varios países cuya función es servir lo más rápido posible (gracias a sistemas de caché y minificación) imágenes, vídeos, librerías JS y otro tipo de cosas.
- **ARIA:** Accessible Rich Internet Applications. Etiquetas aria usadas para añadir accesibilidad al contenido de las páginas web.
- **SPA:** Single Page Applications. Es una forma de construir webs en la que cargan, gracias a Javascript, todas las páginas que contienen la web para que navegar luego entre ellas se haga de forma instantánea: Angular o VueJS son ejemplos de librerías para hacer webs SPA.
- **SSR:** Server Side Rendering. Concepto muy de moda últimamente. La idea es crear un servicio intermedio en servidor, entre el frontend y el backend, cuya función sea prerenderizar las páginas en servidor para enviarlas ya cargadas a los usuarios. Esto facilita también el SEO.
- **SEO:** Search Engine Optimization. Conjunto de estrategias a la hora de crear páginas webs para facilitar el trabajo a los robots de los buscadores (como Google por ejemplo) y así tratar de conseguir mejores posiciones en las búsquedas.
- **SSG:** Static Site Generator. En lugar de que el usuario entre en una web y se hagan las peticiones al backend y a la base de datos para montar la página, la idea con SSG es que las páginas se generen una sola vez con los datos ya cargados. Lo malo de esto es que claro, los datos no pueden cambiar muy amenudo porque por cada cambio en los datos hay que generar nuevamente todas las páginas.
- **JAMStack:** Javascript APIs and Markup. Término muy de moda últimamente. La idea con JAMStack es aprovechar las ventajas de



UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENIERIA
Y TECNOLOGIAS AVANZADAS – IPN



Ingeniería web
1er Departamental

Profr. POLANCO MONTELONGO FRANCISCO ANTONIO
Academia de Telemática

SSG para montar una web con datos, pero generada de forma estática. Se suele usar mucho para webs tipo blogs o de noticias.

- **TDD:** Testing Driven Development. Técnica que consiste en crear primeros los tests antes que el código para asegurar que todo el código que escribimos está testeado.
- **LTS:** Long term support: Normalmente se usa para el versionado de librerías o de cualquier software. Se trata de versiones especiales que sacan y que te aseguran que van a tener soporte alargado, normalmente de años.



1.5 Referencias

Referencia	Título	Ruta	Fecha	Autor
1	Top Ten Guidelines for Homepage Usability	http://www.nngroup.com/articles/top-tenguidelines-for-homepage-usability/	2002	Nielsen, J.
2	Specifying Quality Characteristics and Attributes for Web Sites.	Páginas 266-278.	2001	Olsina, L., Lafuente, G. y Rossi, G.
3	Engineering a Better Web Site.	Todo el libro.	2000	Ginige, A.
4	Software Engineering : A Practitioner s Approach – European Adaptation.	Todo el libro	2000	Pressman, R. S.
5	Prospects for an Engineering Discipline of Software.	https://ieeexplore.ieee.org/document/60586	1990	Shaw, M.

Tabla 3. Referencias.

Relación completa de todos los documentos relacionados en la especificación de requisitos de software, identificando de cada documento el título, referencia (si procede), fecha y organización que lo proporciona.



1.6 Resumen

Dentro de este documento observamos que se tiene la descripción general del proyecto con el fin entender la funcionabilidad y las relaciones de los elementos al igual que las características de cada usuario y el desempeño de funciones de estos.

También se mencionan las restricciones con mayor posibilidad de presentarse dentro del desarrollo de la aplicación web, esto puede provocar un problema con ciertos factores de los cuales pueden ser software o hardware, estas problemáticas están consideradas dentro del documento. Como último punto de la descripción de este proyecto, se encuentra el apartado dedicado a la evolución esperada del proyecto, donde se consideran las actualizaciones y posibles mejoras que se puedan realizar en un futuro, pensando en futuras versiones del proyecto con el fin de adaptarlo a las nuevas necesidades.

Posteriormente se establecen los requerimientos del proyecto tales como los requerimientos para el software, hardware, usuario o las interfaces y la comunicación entre los distintos elementos de la aplicación web, con el fin de saber el cómo debe funcionar el sistema, siendo factores ideales y las funciones esperadas.

Al igual que los requerimientos no funcionales, estos nos explicaran el cómo funciona el sistema del proyecto a comparación de los resultados esperados, con estos conocimientos tendremos claro factores fundamentales del sistema tales como la seguridad, portabilidad incluso la fiabilidad.

La organización de este documento busca su fácil comprensión al englobar en los primeros puntos la introducción con los tecnicismos y referencias ocupadas al igual que presentar al equipo responsable al igual que sus funciones, esto con el fin de que el lector tenga el conocimiento para comprender el desarrollo del proyecto.

Posteriormente, se dará una descripción del sistema, al igual que las características de este, esto nos dará la información necesaria para comprender el proyecto, su uso, sus dificultades y puntos de mejora para futuras versiones.



2 descripción general

2.1 Perspectiva del producto

El sistema actúa de manera independiente, no al pertenecer a un sistema mayor como un intermediario que realice la creación, modificación o eliminación de las tareas y muestre en otra categoría las tareas ya completadas, el usuario contara con mayor libertad en el manejo de sus tareas y diseño de la aplicación al igual que el uso de las tecnologías de más confianza.

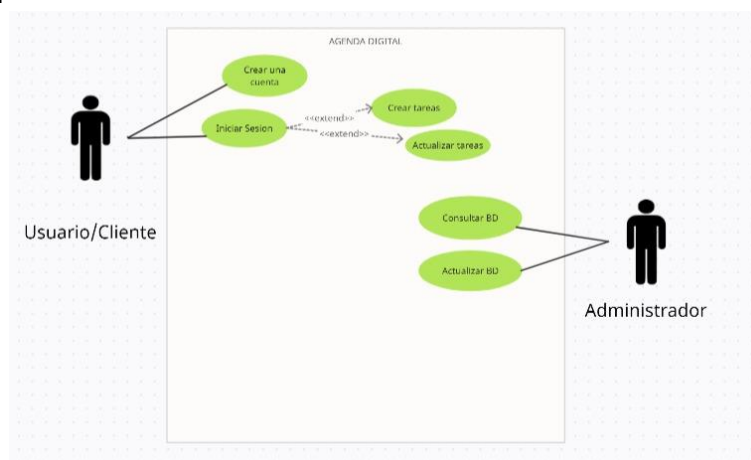


Figura 1. Arquitectura del sistema.

2.2 Funcionalidad del producto

La página web mostrara una pantalla de inicio en la cual se observará un texto que nos permite diferenciar los distintos templates, y una barra de navegación en la parte superior, dentro de esta sección se mostrara una serie de vínculos los cuales nos mostraran una página dependiendo de en cual se entre.

En una sección se mostrarán un formulario el cual nos pedirá ingresar datos para poder registrarse e iniciar sesión, también se observa una opción en la barra de navegación que nos permite iniciar sesión si ya se tiene una cuenta. Una vez iniciada sesión en la barra de navegación se cambiará a los hipervínculos que nos permitirán ver los formularios los cuales nos permitirán crear y modificar las tareas, a su vez también puede completarse las tareas o en caso necesario, eliminarla.



2.3 Características de los usuarios

Tipo de usuario	Administrador.
Formación	Conocimiento de ingeniería web (Full stack).
Habilidades	Modificación total de la página.
Actividades	Administrar todas las funciones de la página web.

Tabla 4. Características del usuario.

2.4 Restricciones

La principal problemática será el uso de las bases de datos y el uso de un lenguaje de programación ya que la conexión entre ambos suele ser un tanto complejo ya que el más pequeño error causará que se pierda la conexión y por ende la posible pérdida de los datos dentro de la base de datos.

Otro problema será encontrar un servicio de hosting confiable el cual nos permita cargar el proyecto en una web.

También se considera la problemática la visualización de la página web en distintos dispositivos tales como computadoras o dispositivos móviles.

Sin embargo, una problemática mas compleja es el uso de un servicio hosting en la nube en el cual es necesario tener una un repositorio en plataformas con github con el cual nos permitirán configurar las variables necesarias para la nube.

2.5 Suposiciones y dependencias

El principal factor que afectaría al sistema seria la modificación de los contenedores los cuales significan un rediseño del Frontend lo que terminaría en retrasos en la entrega, también se considera la suposición del sistema operativo y compilador que se use para la realización de programa, al ser necesario el cambio de hardware o su defecto programar las funciones desde 0 ya que los compiladores y manejadores de lenguajes trabajan de manera distinta.

2.6 Evolución previsible del sistema

Se buscará un implementar más funciones tales como un repositorio de tareas eliminadas a modo de un historial el que nos permitirá tener un mayor control de tareas.

También se buscará un convenio con una empresa con el fin de buscar funcionalidades específicas para la optimización y administración de tareas y objetivos de un proyecto el cual necesite una planificación en base a un periodo de tiempo estimado.



3 requisitos específicos

Esta es la sección más extensa e importante del documento. Debe contener una lista detallada y completa de los requisitos que debe cumplir el sistema a desarrollar. El nivel de detalle de los requisitos debe ser el suficiente para que el equipo de desarrollo pueda diseñar un sistema que satisfaga los requisitos y los encargados de las pruebas puedan determinar si éstos se satisfacen.

Número de requisito	RE1		
Nombre de requisito	Estructura del Proyecto		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Fuente del requisito	Desarrollador		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional

Tabla 5. Requerimiento Especifico 1.

Número de requisito	RE2		
Nombre de requisito	Diseño Adaptable		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Fuente del requisito	Desarrollador		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional

Tabla 6. Requerimiento Especifico 2.

Número de requisito	RF3		
Nombre de requisito	Gestor de Usuarios.		
Tipo	<input type="checkbox"/> Requisito	<input checked="" type="checkbox"/> Restricción	
Fuente del requisito	Desarrollador		
Prioridad del requisito	<input type="checkbox"/> Alta/Eencial	<input checked="" type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional

Tabla 7. Requerimiento Especifico 3.

3.1 Requisitos comunes de los interfaces

3.1.1 Interfaces de usuario

EL uso de una interfaz UI/UX especializada para que la experiencia del usuario sea agradable, rápido, fácil e intuitiva. Realizada con diseño “Material design” y en cada ventana el usuario podrá ver todas las herramientas disponibles, no tendrá necesidad de cambiar entre ventanas para poder acceder a un recurso necesitado. Login y registro



sencillo, la pantalla del login será la pantalla principal en cuanto el usuario ingrese por primera vez al sistema.

3.1.2 Interfaces de hardware

Es de vital importancia disponer de equipos de cómputo en excelente estado, con los periféricos necesarios y que cuente con una conectividad en red para acceder a la página web. ○ Explorador: Principalmente mozilla, Opera o Chrome. ○ Dispositivo: PC, Laptop, Tablet o celular. ○ SO: macOS, Windows 10 o Android.

3.1.3 Interfaces de software

Se integrarán los pagos por medio de Tarjeta de crédito/debito, por lo que será necesario del software especial dedicado para el datafono en uso.

3.1.4 Interfaces de comunicación

Las comunicaciones de los clientes, servidores y aplicaciones se realizarán mediante protocolos estándares de internet, ya sea transfiriendo información, uso de datos sensibles, etc.

3.2 Requisitos funcionales

Definición de acciones fundamentales que debe realizar el software al recibir información, procesarla y producir resultados.

3.2.1 Requisito funcional 1.

Número de requisito	RF1		
Nombre de requisito	Registro usuarios		
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción		
Fuente del requisito	Desarrollador		
Prioridad del requisito Alta/Esencial	<input checked="" type="checkbox"/>	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional

Tabla 8. Requerimiento Funcional 1.

Cada usuario debe registrarse en el sistema, será almacenado en base de datos para poder acceder y usar el sistema, se deberán suministrar datos claves que logre identificar a cada usuario, datos como:

- Nombre.
- ID
- Password



3.2.2 Requisito funcional 2

Número de requisito	RF2
Nombre de requisito	Gestionar Tareas.
Tipo	<input type="checkbox"/> Requisito <input checked="" type="checkbox"/> Restricción
Fuente del requisito	Desarrollador
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

Tabla 9. Requerimiento Funcional 2.

El sistema debe tener un módulo para publicar, editar y eliminar las tareas del día hechas los usuarios y desarrolladores de esta página.

3.2.3 Requisito funcional 3

Número de requisito	RF3
Nombre de requisito	Consulta de información
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Desarrollador
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

Tabla 10. Requerimiento Funcional 3.

Se podrá consultar la información de cada Tarea, ya sea pendiente o completada, esto nos permite saber el nombre, la descripción y saber si tiene importancia o prioridad.

3.3 Requisitos no funcionales

3.3.1 Requisitos de rendimiento

- El sistema debe garantizar un diseño de interfaces que no afecten el desempeño de la base de datos ni perjudiquen el tráfico de internet.
- La creación y modificación de las tareas deben ser con el menor tiempo de proceso al momento de insertar los datos en dentro de la base de datos.
- Optimización entre pestañas, que permitan una navegación fluida capaz de realizar los procesos sin ningún tipo de retraso.



3.3.2 Seguridad

Tanto Django como Flask se consideran soluciones seguras por diferentes razones. Django cuenta con múltiples características de seguridad, como CSRF, SQL y XSS, y tiene el apoyo de un equipo comprometido con la identificación de fallos y errores. Muchas otras características relevantes para la seguridad se añaden y ejecutan automáticamente.

No ocurre lo mismo con Flask. Debido a su código compacto, inicialmente hay menos riesgos de seguridad. De hecho, es más probable que surjan problemas de seguridad al añadir extensiones de terceros. Flask es seguro si estás al quite para hacer todas las actualizaciones de seguridad necesarias y mantienes las bibliotecas al día. A diferencia de Django, Flask no ofrece actualizaciones automáticas, de manera que proteger las aplicaciones y evitar los clásicos errores de Python requiere bastante más esfuerzo en comparación con Django.

3.3.3 Fiabilidad

Es uno de los factores que dará confianza al cliente, para lo cual el sistema está controlando todo tipo de transacción y esta apto a responder todo tipo de incidente.

3.3.4 Disponibilidad

El sistema ha sido desarrollado tomando en cuenta las necesidades, requerimientos, reglas, política, misión, objetivos etc. De la cooperativa, por lo que se encuentra disponible el 90% del tiempo del día tomando en cuenta que el día tiene 24 horas; mientras que el 10% del tiempo es para tareas de almacenes para la actualización de la información de los objetivos.

3.3.5 Mantenibilidad

El sistema cuenta con características parametrizables lo que permitirá futuros mantenimientos. Es decir, cada tres meses se va a realizar un mantenimiento preventivo, encargado de hacerlo están los desarrolladores.

Se realizará el mantenimiento dos veces sin ningún recargo económico, pasados estas dos revisiones tendrán costos adicionales.



3.3.6 Portabilidad

Todos los códigos del marco Django están escritos en Python, que se ejecuta en muchas plataformas. Lo que lleva a ejecutar Django también en muchas plataformas como Linux, Windows y Mac OS.

- Desarrollo rápido.
- Débilmente acoplado.
- Facilidad de modificación.