

Pontificia Universidad Católica del Ecuador

Facultad de Ingeniería

Escuela de Sistemas



Integrantes:

Luis Benavides

Félix Pilamunga

Oscar Fernández

Josué Parra

Tema: “Ideas para cocinar”



2 INDICE DE CONTENIDO

| | | |
|-----|-----------------------------------------------------------------------------------------|----|
| 2 | INDICE DE CONTENIDO | 2 |
| 3 | DEFINICIÓN DE LA PROPUESTA..... | 3 |
| 4 | JUSTIFICACIÓN | 3 |
| 5 | OBJETIVO GENERAL..... | 4 |
| 6 | OBJETIVOS ESPECÍFICOS..... | 4 |
| 7 | ALCANCE | 4 |
| | La aplicación debe permitir al usuario potencial realizar las siguientes acciones:..... | 4 |
| 8 | FUNDAMENTACIÓN TEÓRICA | 5 |
| 8.1 | Servidor Web | 5 |
| 8.2 | Modelo Vista Controlador (MVC) | 5 |
| 8.3 | ORM | 6 |
| 8.4 | Back-end..... | 6 |
| 8.5 | Front-end | 7 |
| 8.6 | API-rest..... | 7 |
| 9 | METODOLOGÍA INVESTIGATIVA..... | 8 |
| 9.1 | Análisis de Requerimientos..... | 8 |
| 9.2 | Análisis Potenciales de Usuarios | 14 |
| 9.3 | Angular | 14 |
| 9.4 | Trabajo en Grupo | 15 |
| 10 | PRINCIPALES RESULTADOS OBTENIDOS | 15 |
| 11 | CONCLUSIONES..... | 16 |
| 12 | RECOMENDACIONES..... | 16 |
| 13 | BIBLIOGRAFÍA..... | 17 |

3 DEFINICIÓN DE LA PROPUESTA

En este proyecto queremos enfocarnos en la realización de un aplicativo web enfocado en el manejo de Recetas de Comida, queremos brindar una alternativa sencilla e innovadora con el objetivo de obtener toda esta información a la mano a través de un clic, nuestro aplicativo web permitirá a cada usuario el manejo de un perfil por el cual se podrá gestionar la creación de recetas de comida, a su vez podrá visualizar un amplio catálogo de recetas a medida que más usuarios se suman a esta propuesta.

4 JUSTIFICACIÓN

La aplicación desarrollada se denomina: “Ideas para cocinar” es una aplicación creada para presentar a sus usuarios registrados y no registrados, información de recetas de comida típica ecuatoriana, a su vez invitándolos a formar parte de un aprendizaje, para cocinar de una manera amigable, interactiva en la que compartirán sus conocimientos, también: registrarse, ver, editar, agregar sus recetas con autoría de una manera amigable e interactiva, pudiendo así agregar imágenes, pequeños tips para el mejoramiento de la sazón de nuestra gastronomía ecuatoriana.

Existe también la opción de almacenar sus recetas de acuerdo a su usuario, es decir darse a conocer sus Id, para sus posibles seguidores en una red social al estar vinculada en Facebook.

Esta aplicación a diferencia de varias de las paginas en internet, son informativos más no una aplicación que se pueda interactuar o generar más conocimiento.

Nuestra aplicación posee una interface amigable, que permite interactuar con otros usuarios y lo más importante generar conocimiento y a su vez aprendizaje.

Como por ejemplo podemos mencionar las siguientes páginas web relacionadas a comidas ecuatorianas.

- <https://cocina-casera.com/comida-platos-tipicos-ecuador/>
- <https://lacomidaecuatorialana.com/cocina-tipica-y-gastronomia-ecuatorialana/>
- <https://www.laylita.com/recetas/platos-fuertes-ecuatorialanos/>

5 OBJETIVO GENERAL

Desarrollar una aplicación web que tenga como finalidad enseñar a preparar recetas de Comida típica ecuatoriana, dando a conocer un poco de su origen de una forma interactiva con el usuario.

6 OBJETIVOS ESPECÍFICOS

- Realizar una búsqueda en el tiempo indicado de las principales recetas y comida típica que encontramos en nuestro país con el fin de almacenar información en la base de datos respectiva.
- Llevar a cabo la codificación de los debidos procesos a considerar necesarios para realizar un producto de software eficaz y eficiente.
- Convocar a una reunión con los integrantes del grupo, en esta junta se deben acordar las acciones que se realizarán, así como el tiempo en que se llevarán a cabo.
- Documentar debidamente cada una de las etapas del proyecto con el fin de entregar un manu al claro para el manejo sin problemas de cada usuario.
- Aplicar las mejoras en las diversas partes desarrolladas, utilizando un método de monitoreo para asegurar que estas se estén llevando a cabo de la forma correcta.
- Después de aplicar las mejoras, tomar las opiniones de varios usuarios para conocer si las estrategias utilizadas, han tenido el impacto que se buscaba.

7 ALCANCE

La aplicación debe permitir al usuario potencial realizar las siguientes acciones:

- Registrar y autenticar usuarios.
- Manejo de perfil de usuario
- Presentar recetas de Comida con información detallada.
- Compartir recetas en Facebook
- Ingresar y definir la privacidad de recetas propias
- Guardar recetas favoritas
- Buscar recetas mediante uno o más parámetros.

8 FUNDAMENTACIÓN TEÓRICA

Realizar una búsqueda académica que den sustento a los conceptos de las tecnologías o herramientas y las metodologías utilizadas en el desarrollo del proyecto.

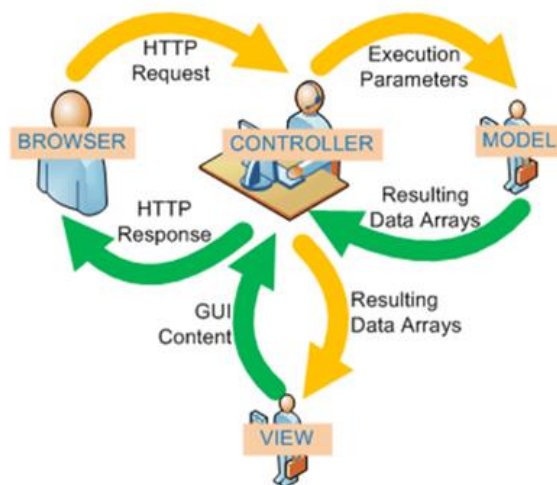
8.1 Servidor Web

Un servidor web es un programa que atiende y responde a las diversas peticiones de los navegadores, proporcionándoles los recursos que solicitan mediante el protocolo HTTP o el protocolo HTTPS. Un servidor web básico tiene un esquema de funcionamiento muy sencillo, ejecutando de forma infinita el bucle siguiente:

- Espera peticiones en el puerto TCP asignado (el estándar para HTTP es el 80).
- Recibe una petición.
- Busca el recurso en la cadena de petición.
- Envía el recurso por la misma conexión por donde ha recibido la petición.
- Vuelve al punto 2.

Un servidor web que sigue el esquema anterior cumpliría los requisitos básicos de los servidores HTTP, aunque, eso sí, sólo podría servir ficheros estáticos. A partir del esquema anterior se han diseñado y construido todos los programas servidores de HTTP que existen, variando sólo el tipo de peticiones (páginas estáticas, CGI, Servlets, etc.) que pueden atender, en función de que sean o no multi-proceso, multi-hilados, etc.

8.2 Modelo Vista Controlador (MVC)



Modelo Vista Controlador (MVC) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

Se trata de un modelo muy maduro y que ha demostrado su validez a lo largo de los años en todo tipo de aplicaciones, y sobre multitud de lenguajes y plataformas de desarrollo.

- El **Modelo** que contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia.
- La **Vista**, o interfaz de usuario, que compone la información que se envía al cliente y los mecanismos interacción con éste.
- El **Controlador**, que actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno.

8.3 ORM

Un ORM es un modelo de programación que permite mapear las estructuras de una base de datos relacional (SQL Server, Oracle, MySQL, etc.), en adelante RDBMS (Relational Database Management System), sobre una estructura lógica de entidades con el objeto de simplificar y acelerar el desarrollo de nuestras aplicaciones.

Las estructuras de la base de datos relacional quedan vinculadas con las entidades lógicas o base de datos virtual definida en el ORM, de tal modo que las acciones CRUD (Create, Read, Update, Delete) a ejecutar sobre la base de datos física se realizan de forma indirecta por medio del ORM.

8.4 Back-end

Es la parte del desarrollo web que se encarga de que toda la lógica de una página web funcione. Se trata del conjunto de acciones que pasan en una web pero que no vemos como, por ejemplo, la comunicación con el servidor.

Algunas de las funciones que se gestionan en la parte del back-end son:

- El Desarrollo de funciones que simplifiquen el proceso de desarrollo.

- Acciones de lógica.
- Conexión con bases de datos.
- Uso de librerías del servidor web (por ejemplo para implementar temas de caché o para comprimir las imágenes de la web).

Además, tiene que velar por la seguridad de los sitios web que gestiona y optimizar al máximo los recursos para que las páginas sean ligeras.

8.5 Front-end

El front-end es a menudo llamado GUI (Graphical User Interface) porque es la interfaz que los visitantes pueden ver y usar. El front-end se utiliza principalmente para mostrar varios tipos de contenido y hacer que la entrada del usuario esté disponible para el back-end. El contenido mostrado incluye la estructura básica del sitio web, como la navegación. El front-end incluye textos, gráficos, vídeos y otros medios. El front-end es a menudo llamado GUI (Graphical User Interface) porque es la interfaz que los visitantes pueden ver y usar. El front-end se utiliza principalmente para mostrar varios tipos de contenido y hacer que la entrada del usuario esté disponible para el back-end. El contenido mostrado incluye la estructura básica del sitio web, como la navegación. El front-end incluye textos, gráficos, vídeos y otros medios.

8.6 API-rest

Una API de transferencia de estado representacional (REST), o API de RESTful, es una interfaz de programación de aplicaciones que se ajusta a los límites de la arquitectura REST.

Una API o interfaz de programación de aplicaciones es un conjunto de definiciones y protocolos que se usa para diseñar e integrar el software de aplicaciones. Suele considerarse como el contrato entre un proveedor de información y un usuario, donde se establece el contenido que se requiere del consumidor (la llamada) y el que necesita el productor (la respuesta).

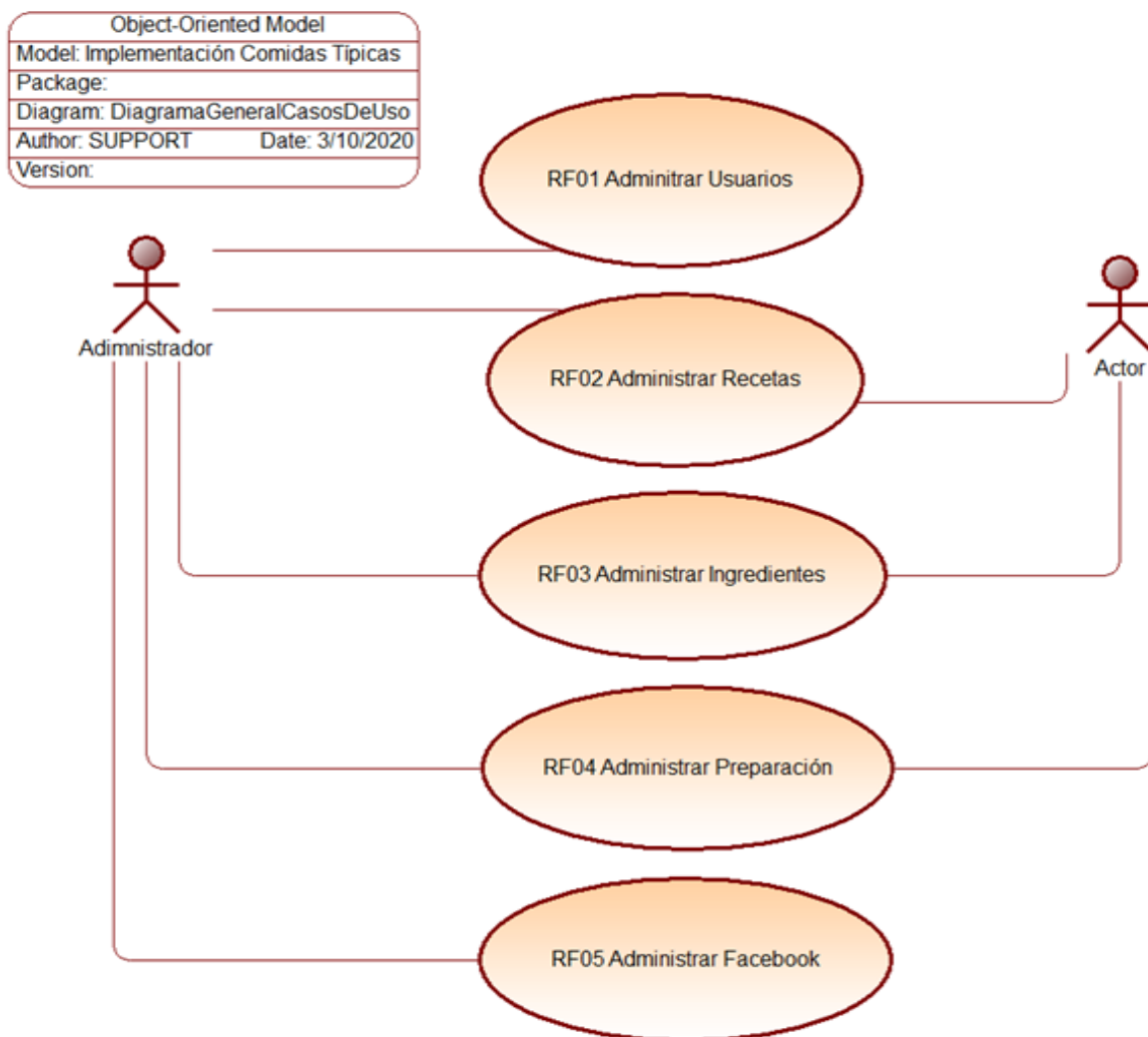
Existe una gran variación en las formas en que se utilizan los API. Las redes sociales, por ejemplo, proporcionan un API que se puede utilizar en otros sitios para recuperar información de una página.

9 METODOLOGÍA INVESTIGATIVA

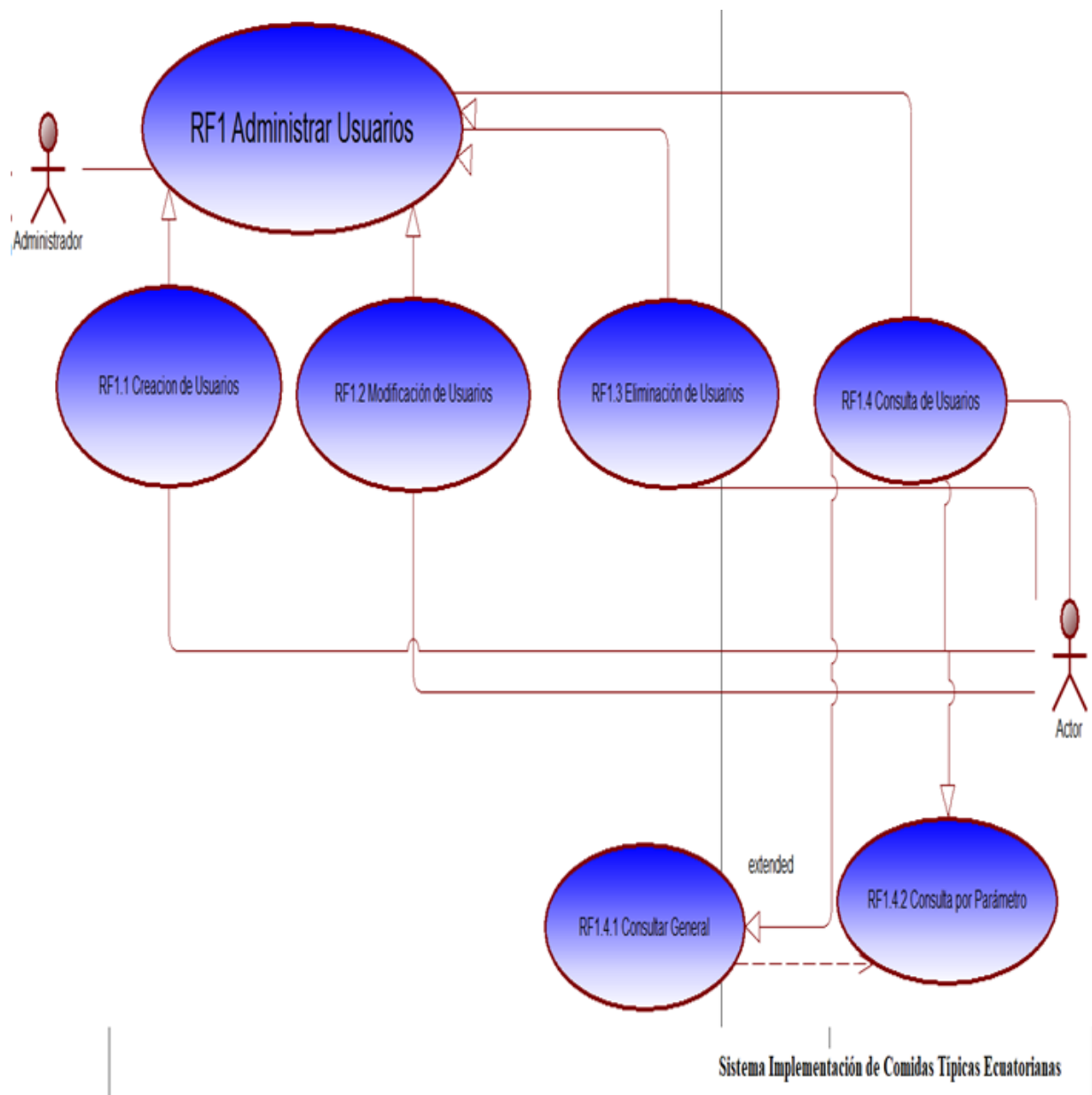
Se indican cuáles fueron las etapas o fases para el desarrollo del proyecto, ejemplos: análisis de requerimientos, análisis de potenciales usuarios, investigación sobre Angular, determinar cómo es el trabajo en grupo (gestor de proyectos, sistema de control de versiones), etc.

9.1 Análisis de Requerimientos

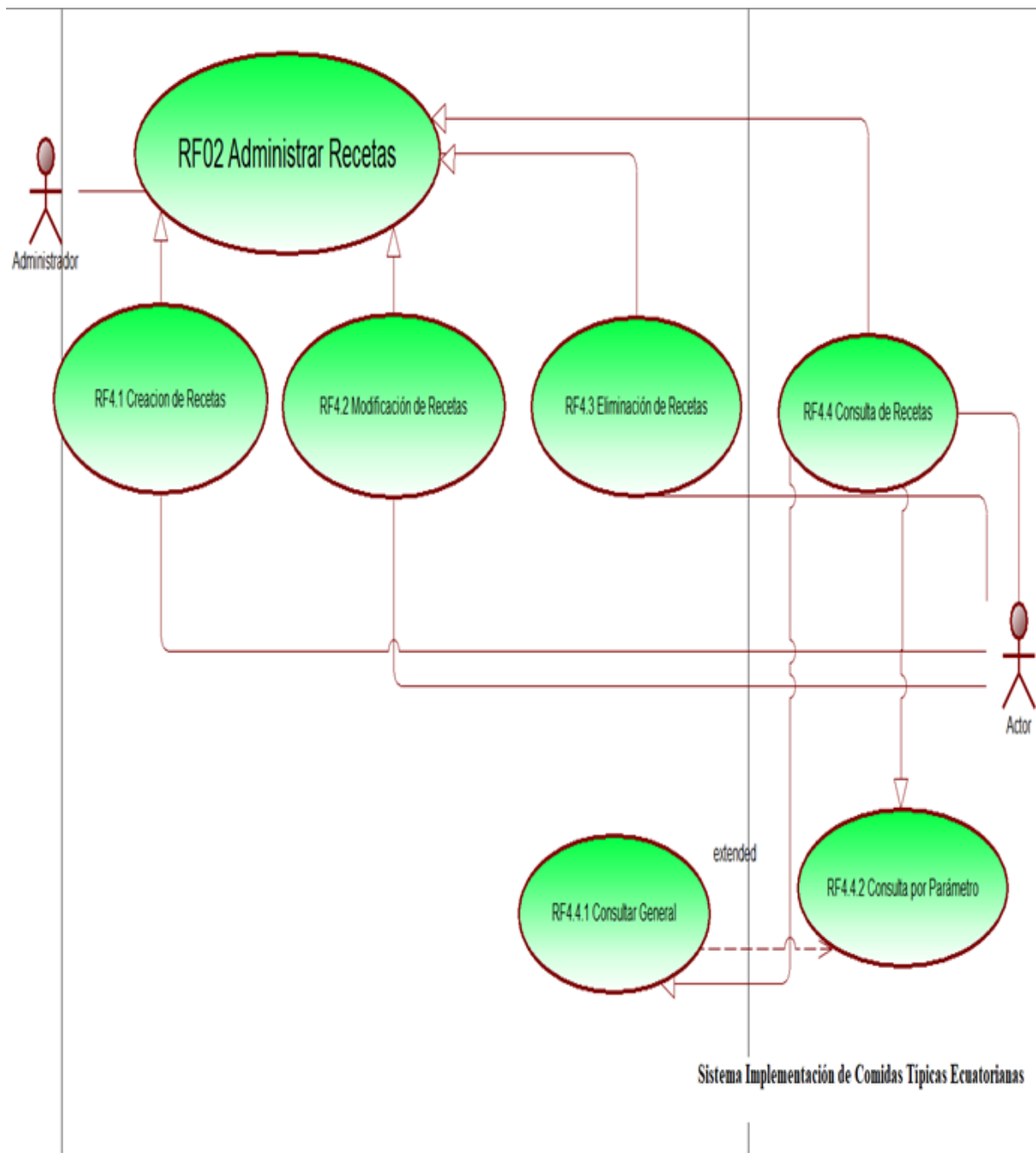
Diagrama General – Casos de uso



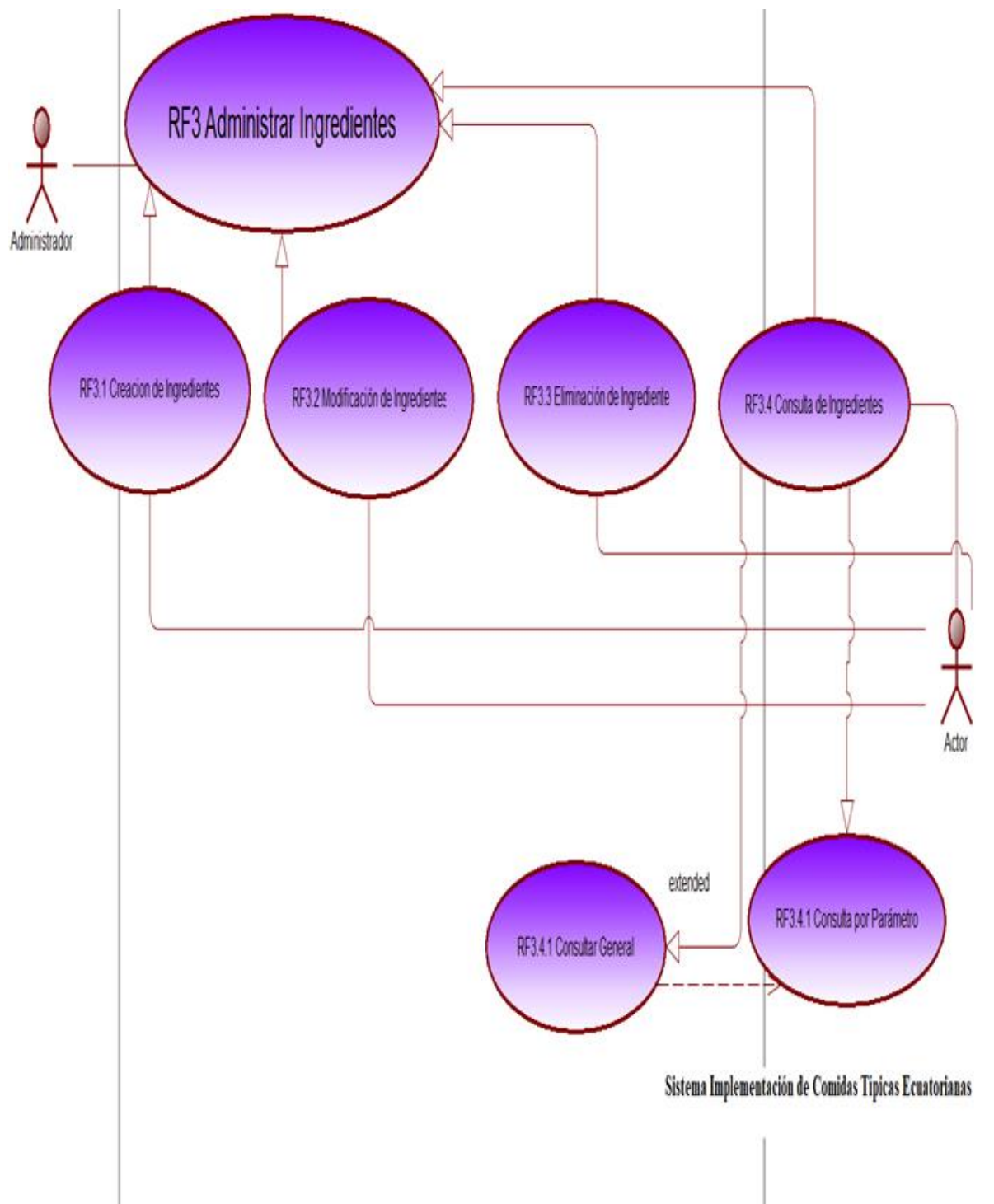
RF01 Caso de Uso Administrar Usuarios



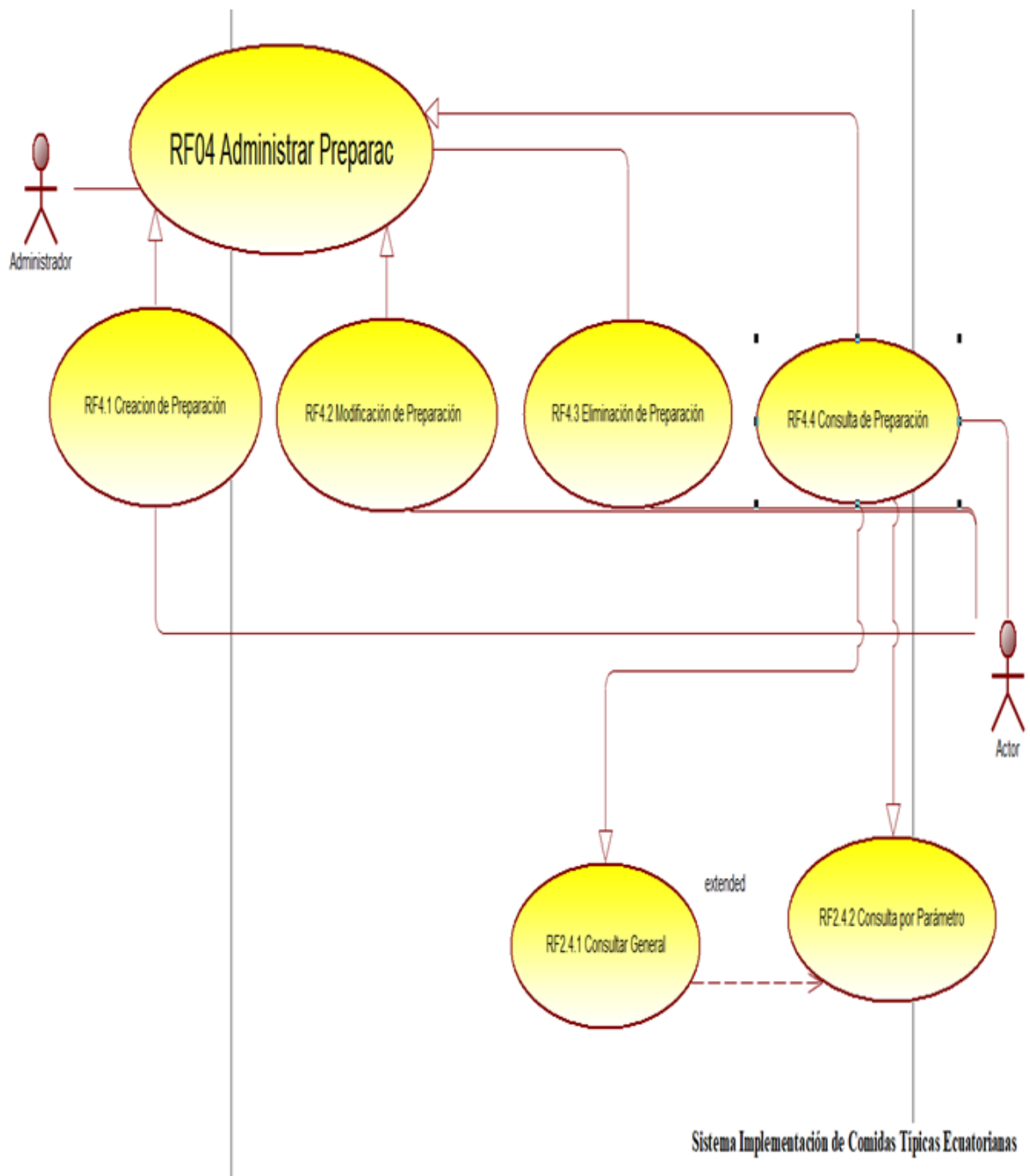
RF02 Caso de Uso Administrar Recetas



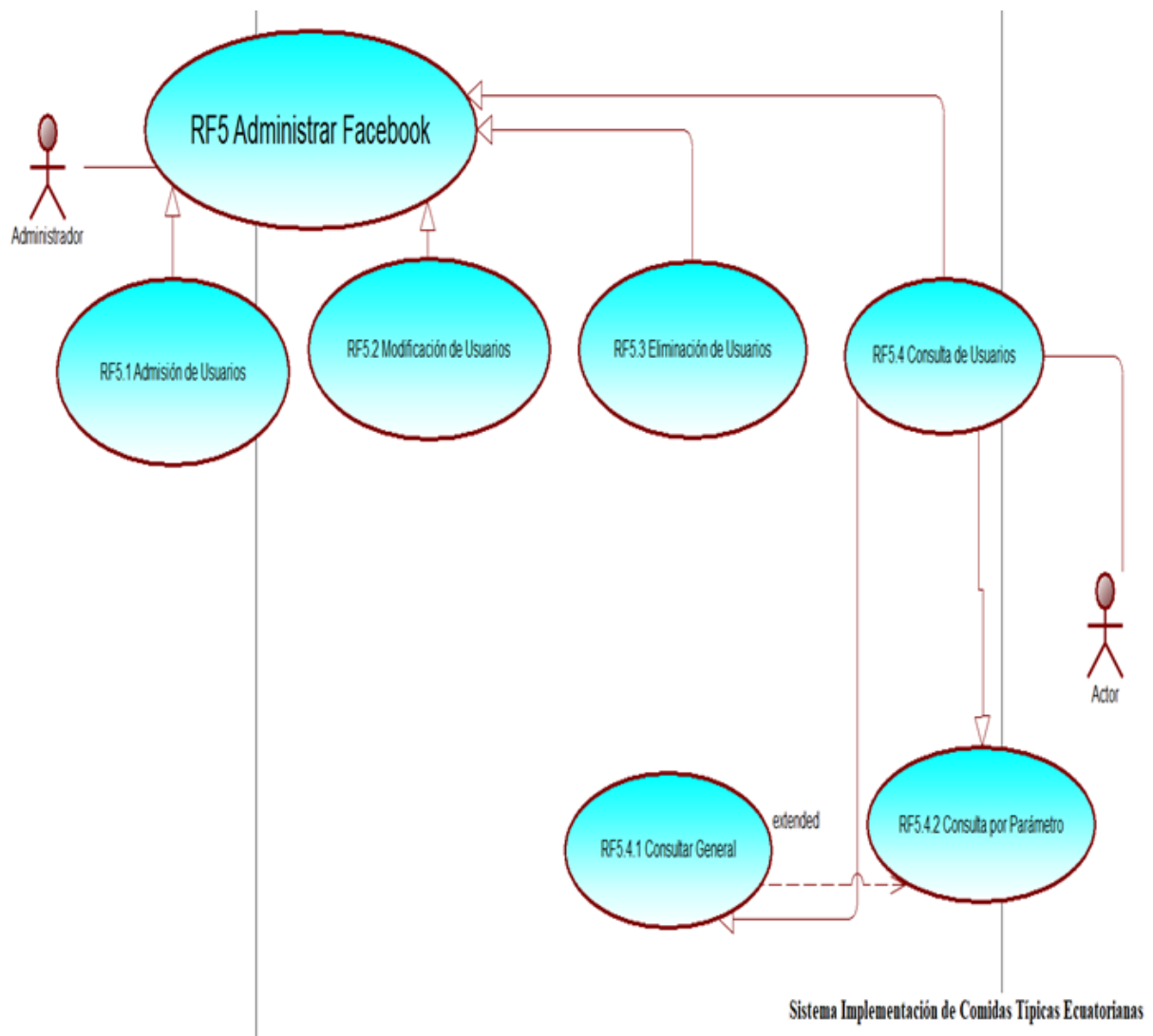
RF03 Caso de Uso Administrar Ingredientes.



RF04 Caso de Uso Administrar Preparación.



RF05 Caso de Uso Administrar Facebook.



9.2 Análisis Potenciales de Usuarios

De acuerdo a un análisis realizado a nuestros usuario potencial, se ha determinado que la aplicación es sencilla de utilizar, la interface es amigable puesto que las letras son de notoria apreciación, sus colores son agradables a la vista de un usuario potencial, su registro es sencillo, su menú es entendible el usuario no tiene opción a perderse en ningún ámbito de la aplicación, una vez registrado el usuario ingresa al menú con dos opciones: recetas, ingredientes, en las cuales puede visualizar sus recetas o agregar recetas.

En la entrevista realizada a un usuario potencial sin conocimientos avanzados en computación a determinado que es totalmente amigable y sencilla nuestra aplicación por tanto es una aplicación con todo lo necesario para formar parte de una red social e interactuar con otros usuarios potenciales.

9.3 Angular

Angular es un framework opensource desarrollado por Google para facilitar la creación y programación de aplicaciones web de una sola página, más conocidas como webs SPA (Single PageApplication).

Angular separa completamente el frontend y el backend en la aplicación ya que evita escribir código repetitivo y mantiene todo más ordenado gracias a su patrón MVC (Modelo-Vista-Controlador) asegurando los desarrollos con rapidez, a la vez que posibilita modificaciones y actualizaciones.

Entre otras ventajas, este framework es modular y escalable adaptándose a nuestras necesidades y al estar basado en el estándar de componentes web, y con un conjunto de interfaz de programación de aplicaciones (API) permite crear nuevas etiquetas HTML personalizadas que pueden reutilizarse.

El lenguaje principal de programación de Angular es Typescript, y así toda la sintaxis y el modo de hacer las cosas en el código es el mismo, lo que añade coherencia y consistencia a la información, permitiendo, por ejemplo, la incorporación de nuevos programadores, en caso de ser necesarios, ya que pueden continuar su trabajo sin excesiva dificultad.

9.4 Trabajo en Grupo

El gestor del proyecto se encargó de gestionar en proyecto desde la fase inicial hasta la final, mediante un plan de proyecto (tarea que puede aportar beneficios al mismo), y de mediar entre los traductores y el cliente ante cualquier problema que pueda surgir, siendo una persona muy organizada y diplomática que cuenta con la habilidad de anticipar y resolver los problemas que se presenten (como el abandono de algún miembro del equipo o un fallo en la lectura de las instrucciones), así como de hacer balance de los riesgos que conllevan ciertas decisiones. Además de todo esto, mantuvo una buena comunicación entre el equipo y los clientes, saber priorizar la importancia de las tareas y ser capaz de automatizarlas.

Un control de versiones es un sistema que registra los cambios realizados en un archivo o conjunto de archivos a lo largo del tiempo, de modo que puedas recuperar versiones específicas más adelante. El sistema de control de versiones nos ayuda a registrar los cambios realizados al código. Aún más, registra quién realizó los cambios y puede restaurar el código borrado o modificado.

Para el sistema de control de versiones usamos GitHub. Antes de continuar, debemos entender a Git como el corazón de GitHub. Git es un sistema de control de versiones desarrollado por Linus Torvalds (el hombre que creó Linux). Si Git es el corazón de GitHub, entonces Hub es su alma. El hub de GitHub es lo que convierte una línea de comandos, como Git, en la red social más grande para desarrolladores.

Además de contribuir a un determinado proyecto, GitHub le permite a los usuarios socializar con personas de ideas afines. Puedes seguir a las personas y ver qué hacen o con quién se conectan.

10 PRINCIPALES RESULTADOS OBTENIDOS

Satisfacción y cumplimiento del producto de software a cabalidad respetando cada uno de los puntos dentro de la documentación.

Flexibilidad, fluidez y originalidad en cada una de las opciones de la aplicación.

Se logró cumplir cada uno de los objetivos planteados desde el inicio limitándonos al alcance Establecido dentro de la documentación.

11 CONCLUSIONES

Las conclusiones son hechos comprobados durante su investigación, no se aceptan opiniones personales ni recomendaciones. No sirve conclusiones como: Angular es una herramienta muy versátil y usada por muchos desarrolladores. (falta indicar porque es versátil y por qué dice que muchos la usan)

- Podemos determinar que nuestro aplicativo web es una herramienta muy sencilla e intuitiva de utilizar.
- Nuestro aplicativo web ofrece una manera segura y privada de administrar recetas en el internet
- El manejo adecuado de usuarios nos permite tener una base de datos amplia de distintas recetas

12 RECOMENDACIONES

- Es recomendable establecer adecuadamente la conexión a la base de datos con el fin de poder filtrar la información de manera segura.
- Al momento de desarrollar el Front-End hay que tomar en cuenta que una combinación adecuada de colores puede determinar el interés del usuario si desea continuar o no dentro de la página.
- Al momento de desarrollar un aplicativo web es importante que como desarrolladores creemos una experiencia para el usuario, eso determinara si el usuario desea pertenecer o no a la comunidad que ofrece nuestra página web.
- Se debe utilizar métodos de encriptación para que no se pueda vulnerar los datos de los usuarios

13 BIBLIOGRAFÍA

Laravel - *The PHP Framework For Web Artisans*. (2020).

Recuperado de: <https://laravel.com/>

Introducción *Angular*. (2020).

Recuperado de: <https://angular.io/>

Laravel, parte 3: *Modelo Vista Controlador* – *Fernando Gaitán*. (2020).

Recuperado de: <https://fernando-gaitan.com.ar/laravel-parte-3-modelo-vista-controlador/>

Si.ua.es. 2020. *Modelo Vista Controlador (MVC)*. *Servicio De Informática ASP.NET MVC 3 Framework*

Recuperado de: <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>

Hostname.cl. 2020. *¿Qué Es Un Servidor Web?* | Hostname.Cl

Recuperado de: <https://www.hostname.cl/blog/servidor-web>

Rafa Arjonilla. 2020. *¿Qué Es El Backend De Una Web Y Por Qué Es Tan Importante?*

Recuperado de: <https://rafarjonilla.com/que-es/backend>

Rock Content. 2020. *API Rest: ¿Qué Es Y Cómo Funciona Ese Recurso?*

Recuperado de: <https://rockcontent.com/es/blog/api-rest>