**Un conjunto de letras blancas en un fondo blanco

Descripción generada automáticamente con confianza media**

**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**

**FACULTAD DE INFORMÁTICA Y ELECTRONICA**

**ESCUELA DE INGENIERIA EN SISTEMAS**

**“DESARROLLO DE UN SISTEMA WEB PARA LA GESTIÓN DEL RECICLAJE EN LA CIUDAD DE RIOBAMBA UTILIZANDO LA TECNOLOGÍA DE BASE DE DATOS NO-SQL Y EL FRAMEWORK EXPRESS”**

**Trabajo de titulación**

Tipo: Proyecto Técnico

Presentado para optar al grado académico de:

**INGENIERO EN SISTEMAS INFORMÁTICOS**

**AUTOR:** FREDDY ORLANDO LEMA MIRANDA.

Riobamba – Ecuador

2021

**Un conjunto de letras blancas en un fondo blanco

Descripción generada automáticamente con confianza media**

**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**

**FACULTAD DE INFORMÁTICA Y ELECTRONICA**

**ESCUELA DE INGENIERIA EN SISTEMAS**

**“DESARROLLO DE UN SISTEMA WEB PARA LA GESTIÓN DEL RECICLAJE EN LA CIUDAD DE RIOBAMBA UTILIZANDO LA TECNOLOGÍA DE BASE DE DATOS NO-SQL Y EL FRAMEWORK EXPRESS”**

**Trabajo de titulación**

Tipo: Proyecto Técnico

Presentado para optar al grado académico de:

**INGENIERO EN SISTEMAS INFORMÁTICOS**

**AUTOR:** FREDDY ORLANDO LEMA MIRANDA.

**TUTOR:** DR. JULIO ROBERTO SANTILLÁN CASTILLO

Riobamba – Ecuador

2021

**© 2021, Freddy Orlando Lema Miranda.**

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

Yo, Freddy Orlando Lema Miranda, declaro que el presente trabajo de titulación es de mi autoría y los resultados del mismo son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor asumo la responsabilidad legal y académica de los contenidos de este trabajo de titulación; El patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo. Riobamba, 29 de enero de 2021

**Freddy Orlando Lema Miranda**

**060502293-8**

CAPITULO I

**ANTECEDENTES**

La industria del reciclaje en la ciudad tiene entre sus principales servicios: compra de papel, cartón, plásticos, metales reciclables y venta de materias primas recicladas para uso industrial. **Otras empresas además de ofrecer los servicios anteriores se encargan del manipuleo, pesaje, transporte, destrucción, clasificación, embalaje y disposición final técnica de los productos reciclables.**

Estos productos reciclables son adquiridos mediante la visita directa de los usuarios que pueden ser personas recicladoras que recorren la ciudad o también pueden ser los mismos propietarios de dichos desechos, estos pueden ir con una frecuencia semanal o diaria según sea el caso a los centros de acopio o plantas procesadoras, algunas empresas cuentan con flotas de vehículos para realizar la compra de materiales a domicilio.

De forma diaria en la ciudad de Riobamba se arrojan 150 toneladas de basura, pero no hay una ordenanza que obligue a los ciudadanos a separar los desechos. Toda la basura, incluidos el papel, el cartón, el plástico, vidrio y latas, se arroja en el mismo contenedor sin diferenciación (El Telégrafo 2019).

Actualmente existen sitios web en los cuales las personas pueden reciclar, ya sea mediante la venta de ésta en forma de chatarra, o iniciativas de empresas de recolección de basura, entre los cuales podemos mencionar los siguientes:

**“GRAHAM RECICLAJE”**

Esta empresa promueve la industria del reciclaje contando con un sitio web para brindar información y además con centros de acopio para que las personas puedan reciclar cualquier tipo de residuo sólido.

**“Cash for electronic scrap USA”**

Esta empresa tiene un sitio web en el cual las personas pueden enviar su basura electrónica a través de empresas de correo en los Estados Unidos como FEDEX y reciben dinero a cambio.

**“EBay”**

A través de esta plataforma de comercio electrónico las personas venden su basura tecnológica como chatarra.

En el Ecuador específicamente en la ciudad de Guayaquil (Bajaña Rodríguez 2015) en su trabajo desarrolló un sistema web para la optimización del proceso de recolección de residuos electrónicos reciclables, específicamente computadoras y sus partes, que les permitirá a las personas reciclar o donar sus computadoras de escritorio, laptops y partes de computadoras. Este sistema tuvo una gran limitación ya que solo se enfocaba al reciclaje de equipos electrónicos mas no al reciclaje de residuos sólidos que es donde se encuentra la mayor cantidad de elementos a reciclar.

En la ciudad de Riobamba los procesos de reciclaje se realizan de manera tradicional, algunas asociaciones cuentan con personas que con el fin de no incomodar a la ciudadanía realicen esta actividad en la noche y madrugada en los diferentes puntos de la ciudad de manera aleatoria, además de que el reciclaje por parte de la ciudadanía es limitado pues todavía no existe una conciencia colectiva sobre la importancia del manejo de residuos, esto ha originado que la industria del reciclaje no pueda identificar la ubicación de los elementos que se quieren reciclar, además de la información en general de los desechos y de las personas que quieran realizar dicho proceso. La oportunidad radica en poder enlazar a la industria del reciclaje con las personas que estén dispuestos a realizar este proceso.

Por lo expuesto anteriormente se propone el Desarrollo de un sistema web para la gestión del reciclaje en la ciudad de Riobamba utilizando la tecnología de base de datos No-SQL y el framework Express, esta herramienta permitirá ayudar a los habitantes y a la industria que realicen este proceso tener un control más organizado y saber exactamente la información en general de las personas y lo residuos sólidos que se quieran reciclar.

**FORMULACIÓN DEL PROBLEMA**

¿Cómo lograr una adecuada usabilidad en la implementación del sistema web para la gestión del reciclaje en la ciudad de Riobamba?

**SISTEMATIZACIÓN DEL PROBLEMA**

* ¿Qué tecnologías se van a utilizar para el desarrollo del sistema web para la gestión del reciclaje?
* ¿Qué módulos se implementarán en el sistema para la gestión del reciclaje?
* ¿La metodología SCRUM permitirá desarrollar el sistema web para la gestión del reciclaje en los tiempos previstos?
* ¿De qué manera se puede evaluar la usabilidad del sistema web para la gestión del reciclaje en la ciudad de Riobamba?

**JUSTIFICACIÓN TEÓRICA**

El presente trabajo de titulación se realiza con el propósito de aportar a la temática de Sistemas de Gestión Ambiental, además de mostrar lo importante que puede ser ya que los resultados que ofrecen a largo plazo pueden ser muy significativos, entre los cuales pueden ser: beneficios económicos para la industria que se dedica a realizar estos procesos, ingresos extras para las personas que desean reciclar, mejorar la gestión de información de los residuos sólidos que se reciclan, contribuir a disminuir el impacto ambiental que producen muchas toneladas de desechos sólidos al día. (Córdova García, Vallejo López, 2013) mencionan que la implementación de un plan de reciclaje puede contribuir a optimizar el uso de los materiales potencialmente reciclables, que podrían ser incorporados a nuevos procesos sin necesidad de ser depositados en el relleno sanitario.

Con el uso sistema por parte del usuario final se podrá obtener información general de las personas que deseen reciclar desde sus inmuebles o lugares de trabajo, los desechos reciclados, las empresas recolectoras, así como también de reportes acerca de cuanto se recicla por parte de cada ciudadano, el perfil de cada uno de los usuarios, la comunicación mediante un sistema de mensajería y por último se tendrá una sección en donde se tendrán las publicaciones y las principales novedades acerca de la temática.

Entre las herramientas que se utilizaran para el desarrollo del sistema, se empleara el framework angular para realizar las interfaces gráficas, y del lado del servidor se empleara el framework Express, además de utilizar Visual Studio Code como IDE de desarrollo de scripts, para la base de datos se utilizará MongoDB, al ser herramientas conocidas y que contienen información detallada por toda la web. Dentro de las ventajas que presentan estas herramientas, es que pueden ser usadas para realizar sistemas en innumerables dispositivos (tablets, smartphones, páginas web, consolas) permitiendo el desarrollo de módulos y la construcción de servicios web, además de que existe por cada herramienta una versión libre que tiene todo lo necesario para el procesamiento de la información, así como, vasta información de la que disponer en la web.

**JUSTIFICACIÓN APLICATIVA**

El desarrollo de sistemas informáticos ayudan a optimizar grandes cantidades de información, permitiendo automatizar los procesos que se llevan a cabo por cada una de las entidades, también permiten mejorar la comunicación ya que un sistema web puede ser accedido desde cualquier lugar del planeta que cuente con acceso a internet, es por ello que con este sistema se podrá gestionar la información de los ciudadanos, desechos sólidos, empresas recicladoras, realizar reportes además de dar a conocer las principales novedades acerca de este proceso, mejorando considerablemente la gestión de la información del proceso de reciclaje de la ciudad con un conjunto de funcionalidades que permitirán satisfacer dichas necesidades.

El sistema web contara con diferentes módulos para los usuarios los cuales son:

**Módulo de Autenticación**

Este módulo contará con los datos necesarios para el ingreso y registro en el sistema.

**Módulo Reciclar**

Este apartado los usuarios que se registraron y escogieron el tipo de usuario que recicla podrán ingresar los datos que corresponderán a los elementos que desean reciclar.

**Módulo Recolectar**

Cualquier usuario registrado en el sitio web puede ser un recolector. Para recolectar debe presionar clic sobre la sección denominada “Recolectar”, la cual presenta una página, la cual tiene información disponible de los elementos reciclados en el sitio web por los usuarios.

**Módulo Perfil ecológico**

Cuando un usuario recicla o recolecta, automáticamente se genera un perfil ecológico, el cual tiene información acerca de los residuos reciclados por el usuario.

**Módulo de mensajería.**

Permitirá el envío y recepción de mensajes entre los usuarios del sistema.

**Módulo de Publicaciones y Novedades.**

Cualquier usuario registrado podrá realizar publicaciones de acuerdo con la temática.

Estos módulos representan básicamente las relaciones que se quieren sistematizar y esquematizar a través del sistema informático, siendo todos los usuarios los encargados de la gestión de la información del reciclaje.

La investigación se encontrará acorde al Plan Nacional de Desarrollo 2017-2021 de la república del Ecuador en el eje “Economía al servicio de la comunidad”, su objetivo número 5 “Impulsar la productividad y competitividad para el crecimiento económico sostenible de manera redistributiva y solidaria.”, en su política 5.4 “Incrementar la productividad y generación de valor agregado creando incentivos diferenciados al sector productivo, para satisfacer la demanda interna, y diversificar la oferta exportable de manera estratégica.”. Así como también se encontrará acorde a las líneas y programas de investigación de la ESPOCH, en el eje de TICs, en la línea de investigación de “Tecnologías de la Información y la Comunicación” en el programa de “Ingeniería de Software.” y acorde a la línea de investigación de la EIS (Escuela de Ingeniería en Sistemas) de “Ciencias” en el programa “Gestión de los sistemas de Información” en el ámbito “Gestión de la información”.

**OBJETIVOS**

**OBJETIVO GENERAL**

Desarrollar un sistema web para la gestión del reciclaje en la ciudad de Riobamba utilizando la tecnología de base de datos No-SQL y el framework Express evaluando la usabilidad mediante la Norma ISO/IEC 9126-3.

**OBJETIVOS ESPECIFICOS**

* Analizar el Framework Express y la tecnología NoSQL para el desarrollo de la aplicación web.
* Desarrollar los módulos de la aplicación web para la gestión del reciclaje.
* Utilizar la metodología SCRUM para el desarrollo de la aplicación web.
* Realizar la evaluación de usabilidad mediante la Norma ISO/IEC 9126-3.

CAPÍTULO II

# MARCO TEÓRICO

## Sistemas web

Los "sistemas Web" o también conocido como "aplicaciones Web" son aquellos que están creados e instalados no sobre una plataforma o sistemas operativos (Windows, Linux). Sino que se alojan en un servidor en Internet o sobre una intranet (red local). Su aspecto es muy similar a páginas Web que vemos normalmente, pero en realidad los 'sistemas Web' tienen funcionalidades muy potentes que brindan respuestas a casos particulares.

Los sistemas Web se pueden utilizar en cualquier navegador Web (Chrome, Firefox, Internet Explorer, etc.) sin importar el sistema operativo. Para utilizar las aplicaciones Web no es necesario instalarlas en cada computadora ya que los usuarios se conectan a un servidor donde se aloja el sistema(Báez, 2012).

(Badal 2017), Menciona algunas ventajas y desventajas de las aplicaciones web:

**Ventajas**

* La Web App utiliza lenguajes muy conocidos entre los programadores como: HTML y CSS.
* A diferencias a la App Nativa, con una sola Web App se llegará a todos los dispositivos. Está programada para su reproducción en cualquier sistema operativo.
* Se ejecutan a través de una simple URL. No necesita una actualización. La modernización la debe sufrir la propia web.
* No se instala desde las tiendas App. Eso significa ahorro.
* Su beneficio más importante, el precio. El desarrollo de una Web App es el más económico.

**Desventajas**

* Con una sola Web App se llegará a todos los dispositivos. Eso sí, la web deberá estar programada para su reproducción en cualquier sistema operativo. Si no es responsive, tendremos problemas al abrirla.
* La inversión que pretendes ahorrarte en el desarrollo de la App, la tendrás que hacer en mejorar tu página web. No sirve de nada hacer una Web App si no adaptas tu página para reproducirla con calidad en cualquier dispositivo.
* Si tu web tiene algún tipo de aprieto, la aplicación tendrá problemas. Además, necesitarás obligatoriamente conexión a internet para su reproducción.
* Al no encontrarla en las diferentes tiendas, pierde visibilidad.
* Tendrás la restricción en el acceso a ciertas características hardware de tu dispositivo.

En este trabajo se ha decidido implementar un sistema web actualmente el internet es uno de los medios tecnológicos más utilizados por las personas; un sistema web hace uso del internet permitiendo el acceso a través de un navegador web debido a que se encuentra alojado en un servidor, ayudando así ahorrar espacio en las computadoras a diferencia de las aplicaciones de escritorio que deben estar previamente instaladas, la característica principal al desarrolla run sistema web es que se puede trabajar desde cualquier lugar es decir sin la necesidad de algún componente externo e independientemente del sistema operativo que se encuentre instalado en la computadora de la cual va acceder.

## Arquitectura de software

Se define como el arte de proyectar, construir y diseñar aplicaciones informáticas, también define la forma de trabajar en un sistema como es la construcción de nuevos módulos pero también implica distinguir una solución estructurada que satisfaga las necesidades todos los requisitos técnicos y operacionales(Martínez, 2016).

Según (Cervantes, 2018), presenta las siguientes características que debe poseer cada arquitectura de acuerdo a la que se vaya a implementar dependiendo del caso de estudio que se vaya a realizar.

* Parte del diseño de software
* Representa alto nivel de la estructura del sistema
* Pueden incluir patrones que supervisen la composición de sus componentes.
* La planificación de proyectos, análisis de riesgos, organización y procesos de desarrollo son los que garantizan la calidad del sistema a desarrollar.
* Los atributos de calidad son parte de los requerimientos no funcionales.
* Una arquitectura errónea puede llevar a problema incontables.

## Arquitectura N capas

El estilo arquitectural en capas se basa en una distribución jerárquica de los roles y las responsabilidades para proporcionar una división efectiva de los problemas a resolver. Los roles indican el tipo y la forma de la interacción con otras capas y las responsabilidades la funcionalidad que implementan(Chininin 2012).

Según (Chininin 2012), esta arquitectura menciona algunas características:

* Descomposición de los servicios de forma que la mayoría de las interacciones ocurre solo entre capas vecinas.
* Las capas de una aplicación pueden residir en la misma máquina o pueden estar distribuidos entre varios equipos.
* Los componentes de cada capa se comunican con los componentes de otras capas a través de interfaces bien conocidos.
* Cada nivel agrega las responsabilidades y abstracciones del nivel inferior.

Arquitectura 3 capas cliente servidor

La arquitectura cliente servidor de 3 capas tiene, la parte del servidor de bases de datos y el servidor de la lógica del negocio y por otro lado la parte de cliente o grupo de clientes donde lo habitual es que un servidor sea una máquina bastante potente con un hardware y software específico que actúa de depósito de datos y funcione como un sistema gestor de base de datos o aplicaciones.

En esta arquitectura el cliente suele ser estaciones de trabajo que solicitan varios servicios al servidor, mientras que un servidor es una máquina que actúa como depósito de datos y funciona como un sistema gestor de base de datos, este se encarga de dar la respuesta demandada por el cliente.(Schiaffarino 2019)

### Componentes de la arquitectura Cliente Servidor

Para entender este modelo (Schiaffarino 2019) define a continuación algunos conceptos básicos que lo conforman.

**Red:** Una red es un conjunto de clientes, servidores y base de datos unidos de una manera física o no física en el que existen protocolos de transmisión de información establecidos.

**Cliente:** El concepto de cliente hace referencia a un demandante de servicios, este cliente puede ser un ordenador como también una aplicación de informática, la cual requiere información proveniente de la red para funcionar.

**Servidor:** Un servidor hace referencia a un proveedor de servicios, este servidor a su vez puede ser un ordenador o una aplicación informática la cual envía información a los demás agentes de la red.

**Protocolo:** Un protocolo es un conjunto de normas o reglas y pasos establecidos de manera clara y concreta sobre el flujo de información en una red estructurada.

**Servicios**: Un servicio es un conjunto de información que busca responder las necesidades de un cliente, donde esta información pueden ser mail, música, mensajes simples entre software, videos, etc.

**Base de datos:** Son bancos de información ordenada, categorizada y clasificada que forman parte de la red, que son sitios de almacenaje para la utilización de los servidores y también directamente de los clientes.

A continuación, (Aimbsor, 2017) lista algunas ventajas y desventajas de esta arquitectura:

**Ventajas**

* Como ya habrás deducido, el modelo cliente/servidor está especialmente indicado en redes medias o grandes que necesiten un alto nivel de fiabilidad.
* Las principales ventajas que ofrece son:
* Administración centrada en el servidor. Los clientes tienen poca trascendencia en el esquema y sus necesidades de administración son menores.
* Centralización de los recursos. Los recursos comunes a todos los usuarios se administran en el servidor. Así se evitan situaciones como la redundancia o inconsistencia de información en las bases de datos.
* Mejora de la seguridad. Al disponer de un mecanismo central de autenticación, las posibilidades de acceso indebido se reducen considerablemente.
* Escalabilidad de la instalación. Se pueden añadir o suprimir clientes sin que el funcionamiento de la red se vea afectado.

**Desventajas**

* Coste elevado. Tanto la instalación como el mantenimiento son más elevados debido al perfil muy técnico del lado servidor.
* Dependencia del servidor. Toda la red está construida al rededor del servidor y si éste deja de funcionar o lo hace con un rendimiento inadecuado, afectará a toda la infraestructura.

## Transferencia de representación de estado (REST)

REST es una interfaz para conectar varios sistemas basados en el protocolo HTTP (uno de los protocolos más antiguos) y nos sirve para obtener y generar datos y operaciones, devolviendo esos datos en formatos muy específicos, como XML y JSON.

El formato más usado en la actualidad es el formato JSON, ya que es más ligero y legible en comparación al formato XML. Elegir uno será cuestión de la lógica y necesidades de cada proyecto. REST se apoya en HTTP, los verbos que utiliza son exactamente los mismos, con ellos se puede hacer GET, POST, PUT y DELETE. De aquí surge una alternativa a SOAP.(Rosa Moncayo 2018)

(Bbvaapimarket, 2016) menciona algunas características:

* Protocolo cliente/servidor sin estado: cada petición HTTP contiene toda la información necesaria para ejecutarla, lo que permite que ni cliente ni servidor necesiten recordar ningún estado previo para satisfacerla. Aunque esto es así, algunas aplicaciones HTTP incorporan memoria caché. Se configura lo que se conoce como protocolo cliente-caché-servidor sin estado: existe la posibilidad de definir algunas respuestas a peticiones HTTP concretas como cacheables, con el objetivo de que el cliente pueda ejecutar en un futuro la misma respuesta para peticiones idénticas. De todas formas, que exista la posibilidad no significa que sea lo más recomendable.
* Las operaciones más importantes relacionadas con los datos en cualquier sistema REST y la especificación HTTP son cuatro: POST (crear), GET (leer y consultar), PUT (editar) y DELETE (eliminar).
* Los objetos en REST siempre se manipulan a partir de la URI. Es la URI y ningún otro elemento el identificador único de cada recurso de ese sistema REST. La URI nos facilita acceder a la información para su modificación o borrado, o, por ejemplo, para compartir su ubicación exacta con terceros.
* Interfaz uniforme: para la transferencia de datos en un sistema REST, este aplica acciones concretas (POST, GET, PUT y DELETE) sobre los recursos, siempre y cuando estén identificados con una URI. Esto facilita la existencia de una interfaz uniforme que sistematiza el proceso con la información.
* Sistema de capas: arquitectura jerárquica entre los componentes. Cada una de estas capas lleva a cabo una funcionalidad dentro del sistema REST.

Entre las ventajas de la tecnología REST que menciona (Bbvaapimarket, 2016) están:

* **Separación entre el cliente y el servidor:** El protocolo REST separa totalmente la interfaz de usuario del servidor y el almacenamiento de datos. Eso tiene algunas ventajas cuando se hacen desarrollos. Por ejemplo, mejora la portabilidad de la interfaz a otro tipo de plataformas, aumenta la escalabilidad de los proyectos y permite que los distintos componentes de los desarrollos se puedan evolucionar de forma independiente.
* **Visibilidad, fiabilidad y escalabilidad:** La separación entre cliente y servidor tiene una ventaja evidente y es que cualquier equipo de desarrollo puede escalar el producto sin excesivos problemas. Se puede migrar a otros servidores o realizar todo tipo de cambios en la base de datos, siempre y cuando los datos de cada una de las peticiones se envíen de forma correcta. Esta separación facilita tener en servidores distintos el front y el back y eso convierte a las aplicaciones en productos más flexibles a la hora de trabajar.
* **La API REST siempre es independiente del tipo de plataformas o lenguajes:** la API REST siempre se adapta al tipo de sintaxis o plataformas con las que se estén trabajando, lo que ofrece una gran libertad a la hora de cambiar o probar nuevos entornos dentro del desarrollo. Con una API REST se pueden tener servidores PHP, Java, Python o Node.js. Lo único que es indispensable es que las respuestas a las peticiones se hagan siempre en el lenguaje de intercambio de información usado, normalmente XML o JSON.

## Lenguajes de Programación para la web

En la actualidad, existe una gran cantidad de lenguajes de programación enfocados al desarrollo web. En sus orígenes, estos lenguajes tenían un formato estático. Sin embargo, con el paso de los años y la evolución a la hora de crear páginas web han evolucionado en lenguajes dinámicos. Estos nuevos avances permiten al usuario interaccionar más con la página y mejorar la experiencia en la navegación, con lo que ya no es un mero tablón que muestra información. (PiensaSolutions, 2017).

**JavaScript**

Se utiliza principalmente del lado del cliente, aunque se puede utilizar del lado del servidor. Actualmente y gracias a tecnologías como AJAX es utilizado para enviar y recibir información del servidor.

Como principales ventajas, tenemos que destacar que es un lenguaje de *scripting*seguro y fiable, cuyos*scripts* tienen capacidades limitadas, debido a la seguridad.

Como desventajas, podríamos mencionar que el código debe descargarse por completo y es visible por cualquier usuario (PiensaSolutions, 2017).

**Lenguaje PHP**

Es un lenguaje enfocado en la creación de webs dinámicas. Sus *scripts* son interpretados por el servidor y genera código HTML. Requiere Apache o IIS con librerías de PHP. Hereda su sintaxis de C, Java y Perl.

Como principales ventajas, hemos de decir que es un lenguaje fácil de aprender y muy rápido. Soporta la orientación a objetos y utiliza un lenguaje multiplataforma. Además, puede conectarse con una gran cantidad de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server… No necesita que se definan los tipos de variables. Uno de sus aspectos más llamativos es que está diseñado con el fin de ser un lenguaje muy seguro para escribir CGI, más que Perl o C.

Es el lenguaje base que utilizan la mayoría de CMS o gestores de contenidos más extendidos como WordPress, PrestaShop, ¡Drupal o Joomla! (PiensaSolutions, 2017).

**Lenguaje Python**

Considerado por muchos el lenguaje más limpio a la hora de programar. El código, al igual que JavaScript, es interpretado y no compilado.

Algo curioso en este lenguaje es que permite a los programadores elegir un estilo de programación concreto (objetos, estructurado, funcional…), debido a que es un lenguaje de programación multiplataforma.

Como ventajas de Python, destacamos que es libre y de fuente abierta, de propósito general. Cuenta con muchas funciones y librerías y es multiplataforma y fácil de programar. Por otro lado, su principal desventaja es que, al ser un lenguaje interpretado, es bastante lento (PiensaSolutions, 2017)..

**Lenguaje Ruby**

Como el anterior, es un lenguaje interpretado y está orientado a objetos. Hereda su sintaxis de Phyton y Perl. El lenguaje puede cargar librerías de extensiones dinámicamente si el sistema operativo lo permite. Además, es un lenguaje portátil (PiensaSolutions, 2017).

En el presente trabajo de titulación se utilizará JavaScript en donde se desarrollarla la Api Rest del sistema gracias al entorno de ejecución NodeJs que permite crear aplicaciones del lado de servidor livianas y eficientes. Además de permitir optimizar los recursos y ser OpenSource.

## Node.js

Es un entorno que trabaja en tiempo de ejecución, de código abierto, multiplataforma, que permite a los desarrolladores crear toda clase de herramientas de lado servidor y aplicaciones en JavaScript. La ejecución en tiempo real está pensada para usarse fuera del contexto de un explorador web (es decir, ejecutarse directamente en una computadora o sistema operativo de servidor). Como tal, el entorno omite las APIs de JavaScript específicas del explorador web y añade soporte para APIs de sistema operativo más tradicionales que incluyen HTTP y bibliotecas de sistemas de ficheros. (Mozilla 2018).

(Blancarte, 2017) menciona ventajas y desventajas de NodeJs las cuales se presentan a continuación:

**Ventajas**

* **OpenSource:** sin duda, hoy el software OpenSource ha tenido la evolución más rápida y con mejor aceptación de los usuarios, además que es totalmente gratis su uso, puede descargar el código fuente para analizarlo y crear tu propio proyecto de Open Source basado en este.
* **Optimización de los recursos:** Dada la naturaleza asíncrona de NodeJS nos permite tener una mucho mejor administración de los recursos.
* **Desarrollos ágiles:** NodeJS permite crear aplicaciones de una forma simple y rápida, a diferencia de otros lenguajes de programación como Java o C#, que se requiere de una gran cantidad de clases para echar a volar un proyecto.
* **FullStack:** Dado que JavaScript se puede ejecutar del lado del servidor y del cliente, es posible crear aplicaciones de BackEnd y FrontEnd con una sola tecnología.
* **Modularidad:** Como ya lo hablamos, NodeJS es extremadamente modular, lo cual permite utilizar únicamente los módulos requeridos sin traernos nada de más.

**Desventajas**

* **Tipos dinámicos:** JavaScript permite enviar como parámetro un objeto que no tiene definición, solo abres y cierres llaves {} y dentro pones lo que sea, lo que hace muy complicado saber que valores puedes enviar. Si eres programador JavaScript seguramente me quedarás colgar vivo diciendo que eso es bueno.
* **Callbacks:** Si bien, como ya analizamos, esto hace que se utilicen mejor los recursos, la verdad es que las callbacks son un dolor de cabeza, ya que tener funciones para cada paso es incómodo y hace muy complicado seguir el código.
* **Refactor:** Si alguna vez has hecho un refactor importante en un proyecto de JavaScript seguro me entenderás, pues casi nunca es posible realizar uno de forma segura, pues el tipeado débil y tipos dinámicos hacen que sea complicado para el IDE determinar cómo refactorizar.
* **Librerías estándar:** Esta es a mi ver una de las más fuertes, es verdad que existen librerías para todos los problemas que se te puedan ocurrir, pero el problema es que existen muchas opciones para lo mismo, haciendo complicado elegir entre una u otra. Adicional que cada una avanza a un ritmo separado en vez de unir fuerzas para mejorar una versión estándar.
* **Arquitectura:** En mi experiencia las aplicaciones NodeJS tiene un nivel de arquitectura muy por debajo de los lenguajes de programación tradicional como Java o C# pues en NodeJS todo se basa en funciones que son enviadas y exportadas de clases. Y es raro ver que se implementan patrones arquitectónicos o patrones de diseño.

## Framework Express

Express es el framework web más popular de Node, y es la librería subyacente para un gran número de otros framework web de Node populares. Proporciona mecanismos para:

* Escritura de manejadores de peticiones con diferentes verbos HTTP en diferentes caminos URL (rutas).
* Integración con motores de renderización de "vistas" para generar respuestas mediante la introducción de datos en plantillas.
* Establecer ajustes de aplicaciones web como qué puerto usar para conectar, y la localización de las plantillas que se utilizan para renderizar la respuesta.
* Añadir procesamiento de peticiones "middleware" adicional en cualquier punto dentro de la tubería de manejo de la petición.

A pesar de que Express es en sí mismo bastante minimalista, los desarrolladores han creado paquetes de middleware compatibles para abordar casi cualquier problema de desarrollo web. Hay librerías para trabajar con cookies, sesiones, inicios de sesión de usuario, parámetros URL, datos POST, cabeceras de seguridad y muchos más. (Mozilla, 2018)

En la **Figura 1-2** se puede observar un ejemplo de la codificación de una aplicación sencilla en NodeJS bajo el framework Express.



**Figura 1-2:** Ejemplo de codificación con el Framework Express.

**Realizado por:** Freddy Lema, 2020

La aplicación se iniciará en el servidor en el puerto 3000 para la conexión. Cuando el usuario visite la URL visualizará “Hello World" Para las solicitudes a la página principal. Para cada URL que no esté definida responderá con un 404 Not Found.

## Framework Angular

También conocido como AngularJS es un framework MVC (Modelo Vista Controlador), desarrollado por Google para el Desarrollo Web Front End que permite crear aplicaciones SPA (Single-Page Applications). Al usar un patrón MVVM (model view view-model) separamos la lógica de la de diseño, pero mantenemos ambas partes conectadas (data binding). De manera que la capa visual no sabe lo que está pasando en la capa lógica, pero manteniendo control sobre el DOM (el cuerpo de la web) y actualizar su contenido como se desee (Galán, 2020).

(Colma, 2020) menciona algunas ventajas y desventajas de usar Angular:

**Ventajas**

* Mantenido por Google
* Gran grupo y ecosistema
* Enlace de datos bidireccional
* Uso de TypeScript
* Soporte para carga diferida
* Inyección de dependencia
* Potente enrutador
* Soporte para Ionic
* Disponibilidad de paquetes

**Desventajas**

* Curva de estudio más pronunciada
* Capacidades limitadas de optimización de motores de búsqueda
* Demasiadas variaciones
* Gran cantidad de código repetitivo

## Bases de datos NoSQL

Las bases de datos NoSQL, también llamadas No Solo SQL, son un enfoque hacia la gestión de datos y el diseño de base de datos que es útil para grandes conjuntos de datos distribuidos. NoSQL, que abarca una amplia gama de tecnologías y arquitecturas, busca resolver los problemas de escalabilidad y rendimiento de big data que las bases de datos relacionales no fueron diseñadas para abordar. NoSQL es especialmente útil cuando una empresa necesita acceder y analizar grandes cantidades de datos no estructurados o datos que se almacenan de forma remota en varios servidores virtuales en la nube.

Contrariamente a las ideas falsas causadas por su nombre, NoSQL no prohíbe el lenguaje estructurado de consultas (SQL). Si bien es cierto que algunos sistemas NoSQL son totalmente no-relacionales, otros simplemente evitan funcionalidades relacionales seleccionadas como esquemas de tablas fijas y operaciones conjuntas. Por ejemplo, en lugar de utilizar tablas, una base de datos NoSQL podría organizar los datos en objetos, pares clave/valor o tuplas. NoSQL se menciona a menudo en combinación con otras herramientas de big data, como el procesamiento paralelo masivo, las bases de datos a base de columnas y las bases de datos como servicio (DaaS). (Rouse, 2015) .

(Espinoza y Fernanda, 2013) mencionan algunas ventajas y desventajas de las bases de datos no relacionales.

**Ventajas**

* **Hay bases de datos no relacionados de código abierto.** Los productos de código abierto aportan beneficios a los desarrolladores como: en precio, no se necesita de un servidor con grandes recursos, no tienen una estructura de datos definida, los datos pueden ser diversos existiendo heterogeneidad.
* **Es de escalamiento sencillo.** Las bases de datos no relacionales buscan una manera de añadir más servidores para manejar más cargas de datos, permitiendo a las empresas una distribución de los equipos dependiendo de las actividades a realizar.
* **Economía**. Las bases de datos no relacionales utilizan servidores de bajo costo para la administración de los datos y el volumen de las transacciones que realicen. El costo por gigabyte por segundo para estas bases puede ser mucho menor que el costo de los RDBMS, lo que le permite almacenar y procesar más datos a un precio mucho más bajo, pudiendo así añadir máquinas según sean las necesidades de la empresa.
* **No generan cuellos de botella**. Las bases de datos relacionales tienen este problema ya que estos tienen que transcribir cada sentencia para ser ejecutadas incluyendo las sentencias complejas, además de un nivel de ejecución más preciso para llevar a cabo, por lo que constituye un punto de entrada común, único y conflictivo en base a rendimiento.

**Desventajas**

* **Soporte.** Las empresas que utilicen estas bases de datos buscan que si su sistema falle por algún motivo tengan el soporte necesario, oportuno y competente de parte de sus vendedores, que en este caso no es el adecuado.
* **Formalismo.** A pesar de que en algunas empresas ya se ha utilizado estas bases, todavía se enfrentan a problemas de credibilidad e inestabilidad.
* **Experiencia.** El corto tiempo de existencia de estas bases de datos implica que no haya un número considerable de desarrolladores y de administradores que conozcan de esta tecnología lo que es un problema encontrar personas conconocimientos técnicos lo suficientemente necesarios para que las empresas se centren en esta base de datos para sus aplicaciones.
* **Administración.** Los objetivos de esta base de datos es proporcionar una solución de administración desde cero pero es un problema que esto se llegue a dar ya que requiere de mucha habilidad de instalar y de un gran esfuerzo para mantener.
* **Compatibilidad.** Estas bases de datos tienen su propio API, las interfaces de consultas son únicas y tienen peculiaridades.

## MongoDB

MongoDB (del inglés humongous, "enorme") es un sistema de base de datos NoSQL orientado a documentos de código abierto y escrito en C++, que en lugar de guardar los datos en tablas lo hace en estructuras de datos BSON (similar a JSON) con un esquema dinámico. Al ser un proyecto de código abierto, sus binarios están disponibles para los sistemas operativos Windows, GNU/Linux, OS X y Solaris y es usado en múltiples proyectos o implementaciones en empresas como MTV Network, Craigslist, BCI o Foursquare.

La razón de esto es que MongoDB, al estar escrito en C++, cuenta con una más que notoria capacidad para aprovechar los recursos de la máquina y, al estar licenciado bajo una licencia GNU AGPL 3.0, es posible adaptarlo a nuestras necesidades (Robledano, 2019).

**Componentes de una base de datos de MongoDB**

Las bases de datos en MongoDB se encuentran conformadas por colecciones. A su vez una colección es un conjunto de varios documentos agrupados en ella.

Los documentos se muestran al usuario en formato JSON para facilitar su lectura ver **Figura 2-2**. La interacción entre el usuario y los documentos JSON, se hace por medio de una terminal que usa el lenguaje de programación JavaScript. Con la ayuda de JavaScript se pueden hacer consultas y modificar los documentos JSON (Galvis, 2015).

Diagrama

Descripción generada automáticamente

Figura 2-2: Componentes de una base de datos de MongoDB.

**Fuente:** (Galvis, 2015, p.11).

Para hacer más eficiente la codificación, decodificación, transporte y el almacenamiento de los diferentes tipos de datos, se usa BSON, es la representa-ción binaria de los documentos JSON, ver la **Figura 3-2**.

Diagrama

Descripción generada automáticamente

**Figura 3-2:** Esquema de comunicación entre una aplicación cliente y un servidor MongoDB.

**Fuente:** (Galvis, 2015, p.11).

**Colecciones y Documentos**

Las colecciones son representaciones de varios documentos bajo un nombre que los representa, MongoDB crea una sola colección después de crear el primer documento, en la **Figura 4-2** se puede ver la representación de una colección junto con el documento con una relación de contención, también los campos o atributos de los documentos.

**Diagrama

Descripción generada automáticamente**

Campos

Documento

Relación de contención

Colección

**Figura 4-2:** Representación de una Colección con relación de contención.

**Realizado por:** Freddy Lema, 2021.

**Diagrama orientado a documentos.**

(Galvis, 2015) en su trabajo de titulación propone una serie de símbolos para representar gráficamente los diferentes tipos de esquemas que se puede hacer en MongoDB. Esto permite modelar el comportamiento estático del esquema de datos y así simplificar y facilitar la comprensión.

Relación de uno a muchos

**Diagrama

Descripción generada automáticamente**

**Figura 5-2:** Representación de relaciones entre colecciones.

**Realizado por:** Freddy Lema, 2021.

## Norma ISO/IEC 9126-3

La ISO, bajo la norma ISO-9126, ha establecido un estándar internacional para la evaluación de la calidad de productos de software el cual fue publicado en 1992 con el nombre de “Information technology –Software product evaluation: Quality characteristics and guidelines for their use”, en el cual se establecen las características de calidad para productos de software. El estándar ISO-9126 establece que cualquier componente de la calidad del software puede ser descrito en términos de una o más de seis características básicas, las cuales son: funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad; cada una de las cuales se detalla a través de un conjunto de subcaracterísticas que permiten profundizar en la evaluación de la calidad de productos de software (Cataldi, 2000).

El proyecto técnico se evaluará con la característica de usabilidad, que es la capacidad del software de ser entendido, aprendido, y usado en forma fácil y atractiva (Diplomado en Gestión de la Calidad del Software 2015).

El criterio de usabilidad contiene 5 atributos los cuales se mencionan a continuación:

* **Entendimiento:** es la manera fácil de utilizar la mayor cantidad de tareas de la aplicación en el menor tiempo posible.
* **Aprendizaje:** es la capacidad que posé la aplicación de trasmitir su uso al usuario (introducciones, manuales, etc.).
* **Operabilidad:** es la facilidad que otorga la aplicación al usuario para su control.
* **Atracción:** es la armonía creada en la aplicación para hacerla más agradable y vistosa al usuario.
* **Conformidad de Uso:** es la capacidad para cumplir las normas y estándares relacionados a la usabilidad (Diplomado en Gestión de la Calidad del Software 2015).

CAPITULO III

# MARCO METODOLÓGICO

En este capítulo se describen las técnicas de investigación, métodos y la metodología que se utilizó para el desarrollo del sistema web que permite gestionar la información del reciclaje en la ciudad de Riobamba.

Para el desarrollo del sistema web se usó el tipo de investigación aplicada, empleando los métodos de investigación deductivo, sintético e inductivo. Además, las técnicas de investigación siguientes: la observación, encuesta y revisión de literatura concerniente al tema. Para concluir en cuanto a la metodología de desarrollo se empleó la metodología ágil denominada Scrum que permite planificar actividades para elaborar entregables y regulares del producto final.

La metodología Scrum cuenta con 3 fases de trabajo las cuales son: planificación, desarrollo y fase de cierre o de finalización.

## Diseño de la investigación

### Tipo de investigación

En el presente trabajo de titulación se hará uso de la investigación aplicada, por que se cuenta con conocimientos que se han ido adquiriendo durante toda la carrera, además, la experiencia en el desarrollo de sistemas web que permitan manejar información. Esto permitirá elaborar un sistema de calidad para el usuario final.

### Métodos de investigación

Los métodos de investigación que permitirán realizar el trabajo de titulación son:

* **Sintético:** Este método nos permite partir de lo esencial hacia un todo, es decir se utilizará para ir realizando cada uno de los módulos que conformarán todo el sistema.
* **Deductivo:** Consiste en ir de lo general a lo particular basándose en el razonamiento y la lógica, se aplicará para reunir información para el marco teórico donde se estudiarán las tecnologías que se utilizan para desarrollar el sistema.
* **Inductivo:** Se emplea este método para realizar el marco de resultados es decir la medición de la usabilidad del sistema.

### Técnicas de investigación

Para recopilar información necesaria para el desarrollo del sistema se hará uso de las técnicas siguientes:

* **Encuesta:** A miembros de la fundación manos que limpian, que se dedican al reciclaje para obtener información para medir la usabilidad en el sistema web.
* **Revisión de literatura:** Permite estudiar y analizar las tecnologias que se van a utilizar para el desarrollo del sistema, como tambien la revision de sistemas similares para tener un concimiento mas general del proyecto de titulacion.

## Revisión de sistema web similares

**CFES USA**

Es una de las corporaciones de reciclaje de metales preciosos más grandes de América del Norte, la compañía maneja tanto volumen que son capaces de ofrecer a sus clientes rendimientos inmejorables en todos sus metales preciosos electrónicos y chatarra de computadora.

Esta empresa tiene un sitio web en el cual las personas pueden enviar su basura electrónica a través de empresas de correo en los Estados Unidos como FEDEX y reciben dinero a cambio.

**Graham Reciclaje**

Es una empresa que fomenta la preservación del planeta, a través cadenas de centros de acopio, cuenta con personal altamente calificado para realizar todas estas tareas.

Esta empresa cuenta con un sitio web que sirve para brindar información acerca de los residuos que reciben para su reciclaje, también da a conocer los servicios que ponen a disposición para la ciudadanía en general.

**Reciclar**

Es una empresa que se encarga de la clasificación, transporte destrucción, embalaje, manipuleo, pesaje, y disposición final técnica de los productos reciclables. Compra estos materiales todo el año.

También cuenta con un sitio web en donde presenta información acerca de, sus servicios, compras, ventas, noticias, contacto. En el apartado para gestionar sus compras y ventas cuentan con un formulario en donde la ciudadanía puede ingresar los datos de lo que quieren comprar o vender.

**Recynter**

Empresa que se encarga del manipuleo y pesaje de todo tipo de residuos en el país beneficiando al cuidado del medio ambiente. Esta empresa cuenta con un sitio web informativo que permite conocer cuales son sus servicios en general, también cuenta con formularios en donde la ciudadanía puede vender sus residuos a esta entidad.

## Análisis e interpretación de la revisión de sistemas web similares

Las empresas mencionadas anteriormente cuentan con plataformas web para dar a conocer sus servicios como recicladores de residuos, estos pueden ser de todo tipo desde residuos como: metales ferrosos, metales no ferrosos, equipos de electrónica, papel, cartón, plástico, etc. Sin embargo, la mayoría de estos sitios web solo se limitan a informar de los servicios y de la información en general de la empresa. O a su vez cuentan con formularios sencillos en donde las personas o empresas pueden llenar la información acerca de los residuos que quieren vender o comprar.

Esto conlleva a concluir que a través de un sistema web se puede presentar información acerca de cualquier empresa que realiza servicios de reciclaje facilitando la comunicación con los clientes que están interesados en el tema, es importante mencionar que también se puede gestionar las compras y ventas iniciales por medio de formularios en donde los interesados pueden llenar la información para su posterior contacto con la empresa.

## Fase de planificación

Para establecer la planificación de este proyecto se implementará la metodología de desarrollo ágil SCRUM, que se basa en establecer el Product Backlog que es una lista en donde se establecen los requisitos que se han definido juntamente con el Product Owner, y el Sprint Backlog que es la planificación realizada por iteraciones que cuentan con historias de técnicas y de usuario que se deberán ir cumpliendo en cada una de ellas.

### Personas y roles involucrados en el proyecto

Los roles descritos en la **Tabla 1-3** son los definidos en base a la metodología SCRUM. Teniendo en total 3 roles, como (Product Owner) tenemos a Graham que participa como el dueño del sistema web. El (Scrum Master) es el director del proyecto y por último el desarrollador que se encargara de la codificación de las funcionalidades del sistema.

**Tabla 1-3:** Roles del proyecto.

|  |  |
| --- | --- |
| **Persona** | **Rol** |
| ASORMALIM | Product Owner |
| Ing. Julio Santillán | Scrum Master |
| Freddy Lema | Desarrollador |

**Realizado por:** Freddy Lema, 2020

Para realizar la estimación de un Sprint se ha utilizado el método denominado T-shirt o talla de la camiseta. Dentro de mencionado método, las tallas S, M, L y XL, en la **Tabla 2-3** están las estimaciones que se utilizarán para dar una medida de duración de los Sprint del proyecto, además mencionaremos que 1 semana equivaldrá a 40 puntos estimados y cada punto estimado equivaldrá a una hora de trabajo.

**Tabla 2-3:** Método de la camiseta.

|  |  |  |
| --- | --- | --- |
| **Talla** | **Puntos estimados** | **Horas de Trabajo** |
| S | 5 | 5 |
| M | 10 | 10 |
| L | 20 | 20 |
| XL | 40 | 40 |

**Realizado por:** Freddy Lema, 2020

### Product Backlog

Para el desarrollo del sistema web, se definieron 14 requisitos funcionales los cuales serán convertidos en historias de usuario identificadas con HU-01, donde el 01 es el número de dicha historia, y las historias técnicas definidas con HT-03, donde 03 es el número de dicha historia técnica. El Product Backlog, representado en la **Tabla 3-3,** contiene las 17 historias de usuario y 05 historias técnicas las cuales se encuentran priorizadas.

**Tabla 3-3:** Product Backlog

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **DESCRIPCIÓN** | **PRIORIDAD** | **PUNTOS ESTIMADOS** | **TALLA** |
| **HT\_01** | Como desarrollador necesito diseñar la base de datos. | ALTA | 40 | XL |
| **HT\_02** | Como desarrollador necesito establecer un estándar de codificación. | ALTA | 20 | L |
| **HT\_03** | Como desarrollador necesito diseñar la arquitectura del sistema. | ALTA | 20 | L |
| **HT\_04** | Como desarrollador necesito diseñar la interfaz de usuario. | ALTA | 40 | XL |
| **HU\_01** | Como desarrollador necesito codificar la Api REST de todas las colecciones del sistema. | ALTA | 40 | XL |
| **HU\_02** | Como usuario público deseo iniciar sesión en el sistema. | ALTA | 20 | L |
| **HU\_03** | Como usuario autenticado requiero agregar una nueva publicación para reciclar. | ALTA | 40 | XL |
| **HU\_04** | Como usuario autenticado requiero modificar una publicación para reciclar. | ALTA | 40 | XL |
| **HU\_05** | Como usuario autenticado requiero eliminar una nueva publicación para reciclar. | ALTA | 40 | XL |
| **HU\_06** | Como usuario autenticado requiero listar mis publicaciones para reciclar. | ALTA | 40 | XL |
| **HU\_07** | Como usuario autenticado requiero ver mis recolecciones asignadas. | MEDIA | 20 | L |
| **HU\_08** | Como usuario autenticado requiero ver mis postulaciones a publicaciones para reciclar. | MEDIA | 20 | L |
| **HU\_09** | Como usuario autenticado requiero eliminar mis postulaciones realizadas a una publicación. | MEDIA | 20 | L |
| **HU\_10** | Como usuario autenticado requiero asignar una postulación a mi publicación para reciclar. | MEDIA | 20 | L |
| **HU\_11** | Como usuario autenticado requiero ver mi perfil ecológico. | MEDIA | 40 | XL |
| **HU\_12** | Como usuario autenticado requiero enviar mensajes a los usuarios registrados en el sistema. | MEDIA | 40 | XL |
| **HU\_13** | Como usuario autenticado requiero ver las publicaciones y novedades de los usuarios registrados en el sistema. | BAJA | 40 | XL |
| **HU\_14** | Como usuario autenticado requiero postular a las publicaciones de reciclaje de otros usuarios. | BAJA | 20 | L |
| **HU\_15** | Como usuario autenticado requiero ver mis notificaciones. | BAJA | 20 | L |
| **HU\_16** | Como usuario autenticado requiero regístrame en el en el sistema. | BAJA | 20 | L |
| **HU\_17** | Como usuario autenticado requiero modificar mi perfil | BAJA | 20 | L |
| **HT\_05** | Como desarrollador necesito elaborar la documentación del sistema. | BAJA | 20 | L |
| **HT\_06** | Como desarrollador necesito realizar el despliegue del sistema en el servidor. | BAJA | 20 | L |
| **TOTAL** | | 620 | | |

**Realizado por:** Freddy Lema, 2020

### Sprint Backlog

El Sprint Backlog se obtiene al realizar la planificación de requisitos, está organizando en iteraciones y considerando el método de talla de camiseta se ha estimado que cada Sprint tendrá como máximo 40 puntos estimados dado que una semana equivale a 40 puntos estimados, así diremos que Sprint se culminará en una semana. La **Tabla 4-3** contiene los detalles del Sprint Backlog organizados por iteraciones, esto es la guía principal del desarrollador que deberá ser codificando en el orden en el que fueron planificadas.

**Tabla 4-3:** Sprint Backlog

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SPRINT | ID | Fecha Inicio | Fecha Fin | PUNTOS ESTIMADOS | TOTAL |
| **1** | HT\_01 | 01/10/2020 | 07/10/2020 | 40 | 40 |
| **2** | HT\_02 | 08/10/2020 | 10/10/2020 | 10 | 40 |
| HT\_03 | 11/10/2020 | 15/10/2020 | 10 |
| HT\_04 | 16/10/2020 | 21/10/2020 | 20 |
| **3** | HU\_01 | 22/10/2020 | 31/10/2020 | 40 | 40 |
| **4** | HU\_02 | 01/11/2020 | 03/11/2020 | 20 | 40 |
| HU\_16 | 04/11/2020 | 07/11/2020 | 20 |
| **5** | HU\_03 | 08/11/2020 | 14/11/2020 | 40 | 40 |
| **6** | HU\_04 | 15/11/2020 | 21/11/2020 | 40 | 40 |
| **7** | HU\_05 | 22/11/2020 | 30/11/2020 | 40 | 40 |
| **8** | HU\_06 | 01/12/2020 | 07/12/2020 | 40 | 40 |
| **9** | HU\_07 | 08/12/2020 | 14/12/2020 | 20 | 40 |
| HU\_08 | 15/12/2020 | 21/12/2020 | 20 |
| **10** | HU\_09 | 22/12/2020 | 31/12/2020 | 20 | 40 |
| HU\_10 | 01/01/2021 | 07/01/2021 | 20 |
| **11** | HU\_11 | 08/01/2021 | 14/01/2021 | 40 | 40 |
| **12** | HU\_12 | 15/01/2021 | 21/01/2021 | 40 | 40 |
| **13** | HU\_13 | 22/01/2021 | 31/01/2021 | 40 | 40 |
| **14** | HU\_14 | 01/02/2021 | 03/02/2021 | 20 | 40 |
| HU\_15 | 04/02/2021 | 07/02/2021 | 20 |
| **15** | HU\_17 | 08/02/2021 | 10/02/2021 | 20 | 40 |
| HT\_05 | 11/02/2021 | 13/02/2021 | 20 |
| **16** | HT\_06 | 14/02/2021 | 17/02/2021 | 20 | 20 |
| **Total** | | | | 620 | 620 |

**Realizado por:** Freddy Lema, 2020

## Fase de desarrollo

En esta fase se llevará a cabo las actividades para el desarrollo del sistema web, el proyecto cuenta con un total de 16 Sprint cada uno planificado con un máximo de 40 puntos estimados o 1 semana a excepción de el último Sprint se trabajará 20 puntos y de los Sprint número 2, número 9 y número 10 que se trabajarán 40 puntos estimados en dos semanas.

### Diseño de la Base de Datos

La Base de Datos para el sistema de gestión del reciclaje será diseñado e implementado con tecnología no SQL especificadamente en MongoDB, por lo cual se establecerá el diagrama que representará el esquema propuesto para la base de datos del sistema gestión del reciclaje.

A continuación, en la **Figura 6-3**, se puede visualizar el diagrama de la base de datos del sistema web.

**Imagen que contiene Texto

Descripción generada automáticamente**

**Figura 6-3:** Diagrama de la base de datos.

**Realizado por:** Freddy Lema, 2020.

La base de datos está compuesta con 6 colecciones que representan a las entidades que conformarán el sistema con sus respectivos documentos que tienen los atributos de cada colección con los que se estarán interactuando para realizar las diferentes funcionalidades del sistema web para la gestión del reciclaje.

### Diccionario de datos

El diccionario de datos de la BD permitirá conocer las características de los tipos de datos que se estarán manejando en el sistema, además de brindar información de respaldo para actualizaciones y mantenimiento por el equipo de desarrollo.

A continuación, en las tablas se describen el diccionario de datos de cada uno de los documentos.

**DOCUMENTO MENSAJE**

**Tabla 5-3:** Diccionario de datos Colección Mensaje.

|  |  |  |
| --- | --- | --- |
| **Campo** | **Tipo** | **Descripción** |
| \_id | ObjectId | Almacena el identificador del mensaje |
| fecha | Date | Almacena la fecha |
| id\_receptor | ObjectId | Almacena el identificador del receptor del mensaje |
| id\_emisor | ObjectId | Almacena el identificador del emisor del mensaje |
| descripcion | String | Almacena la descripción del mensaje |
| leido | Boolean | Almacena si el mensaje fue leído. |

**Realizado por:** Freddy Lema, 2020.

**DOCUMENTO CHAT**

**Tabla 6-3:** Diccionario de datos Colección Chat

|  |  |  |
| --- | --- | --- |
| **Campo** | **Tipo** | **Descripción** |
| \_id | ObjectId | Almacena el identificador del chat |
| fecha | Date | Almacena la fecha |
| id\_usuario | ObjectId | Almacena el identificador del usuario al que pertenece al chat |
| leido | Boolean | Almacena si el mensaje fue leído. |

**Realizado por:** Freddy Lema, 2020.

**DOCUMENTO USUARIO**

**Tabla 7-3:** Diccionario de datos Colección Usuario

|  |  |  |
| --- | --- | --- |
| **Campo** | **Tipo** | **Descripción** |
| \_id | ObjectId | Almacena el identificador del usuario |
| nombres | String | Almacena los nombres |
| apellidos | String | Almacena los apellidos |
| correo | String | Almacena el correo |
| telefono | String | Almacena el teléfono |
| direccion | String | Almacena la dirección |
| password | String | Almacena la contraseña |
| estado | Boolean | Almacena el estado |
| notificaciones | Array | Almacena las notificaciones del usuario |

**Realizado por:** Freddy Lema, 2020.

**DOCUMENTO PUBLICACION**

**Tabla 8-3:** Diccionario de datos Colección Publicación.

|  |  |  |
| --- | --- | --- |
| **Campo** | **Tipo** | **Descripción** |
| \_id | ObjectId | Almacena el identificador de la publicación |
| descripcion | String | Almacena la descripción de la publicación |
| img | String | Almacena la imagen |
| tipo | String | Almacena el tipo |
| cantidad | Double | Almacena la cantidad |
| direccion | String | Almacena la dirección |
| fecha | Date | Almacena la fecha |
| estado | Boolean | Almacena el estado |
| id\_usuario | ObjectId | Almacena el identificador del usuario que hace la publicación |
| residuos | Array | Almacena los residuos de la publicación |

**Realizado por:** Freddy Lema, 2020.

**DOCUMENTO RESIDUO**

**Tabla 9-3:** Diccionario de datos Colección Residuo.

|  |  |  |
| --- | --- | --- |
| **Campo** | **Tipo** | **Descripción** |
| \_id | ObjectId | Almacena el identificador del residuo |
| descripcion | String | Almacena la descripción de la publicación |
| tipo | String | Almacena el tipo |

**Realizado por:** Freddy Lema, 2020.

**DOCUMENTO POSTULACIONES**

**Tabla 10-3:** Diccionario de datos Colección Postulaciones.

|  |  |  |
| --- | --- | --- |
| **Campo** | **Tipo** | **Descripción** |
| \_id | ObjectId | Almacena el identificador de la postulación |
| id\_publicacion | ObjectId | Almacena el identificador de la publicación |
| Id\_usuario | ObjectId | Almacena el identificador del usuario |

**Realizado por:** Freddy Lema, 2020.

### Estándar de codificación

Establecer un estándar de codificación permite seguir un conjunto de buenas prácticas que aumentan la legibilidad por parte de otros desarrolladores aumentando la calidad y mantenibilidad del sistema.

En el desarrollo del sistema se utilizó la notación Camel Case que consiste en unir palabras sin dejar espacios entre ellas. Existen dos notaciones derivadas de Camel Case las cuales son: Uper Camel Case y Lower Camel Case este ultimo será utilizado para la codificación de las funcionalidades del sistema. A continuación, en la **Tabla 11-3** se tiene un ejemplo de aplicación de esta notación.

**Tabla 11-3:** Ejemplo de aplicación estándar Lower Camel Case

|  |  |  |
| --- | --- | --- |
| **Componente** | **Nombre** | **Descripción** |
| Variables | nombreUsuario String | Nombre de la variable con el respectivo tipo de dato |
| Métodos | usuarios Obtener () | Nombre de la clase seguido del método. |
| Clases | usuarioController | Nombre de la clase seguido del módulo al que pertenece. |

**Realizado por:** Freddy Lema, 2020

### Arquitectura del sistema

Definir la arquitectura del sistema es de suma importancia porque permite conocer como esta estructurado el sistema de forma global, evidenciando el funcionamiento que existe entre los diferentes componentes.

A continuación, en la **Figura 7-3** se puede observar el diagrama de como fue implementado.

Diagrama

Descripción generada automáticamente

**Figura 7-3:** Diagrama de la Arquitectura del sistema.

**Realizado por:** Freddy Lema, 2020.

Se basa en la arquitectura cliente servidor de 3 capas que nos permite separar la lógica del sistema, la lógica de los datos y la interfaz que se le presenta al usuario que en este caso sería el cliente. Este patrón permite modificar cada uno de sus componentes si necesidad de afectar a los demás. También permite desarrollar de forma independiente cada uno de los componentes del sistema garantizando la modularidad y la escalabilidad del sistema.

### Diseño de la interfaz de usuario

En este apartado se procedió a diseñar las interfaces de cada una de las funcionalidades que tendrá el sistema para lo cual se utilizó el framework angular que nos permite crear sistemas web de una sola página.

Los colores y el logo de la aplicación fueron escogidos de tal manera que causen un buen impacto de atractividad a la vista del usuario final, además también se escogieron fuentes que permitan la legibilidad con el menor esfuerzo del usuario.

A continuación, se presentan cada una de las interfaces.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Figura 8-3:** Interfaz Iniciar Sesión.

**Realizado por:** Freddy Lema, 2020.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Figura 9-3:** Interfaz Registro.

**Realizado por:** Freddy Lema, 2020.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Figura 10-3:** Interfaz Inicio.

**Realizado por:** Freddy Lema, 2020.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Figura 11-3:** Interfaz Mis Postulaciones.

**Realizado por:** Freddy Lema, 2020.

Interfaz de usuario gráfica, Texto, Aplicación, Teams

Descripción generada automáticamente

**Figura 12-3:** Interfaz Mis Recolecciones.

**Realizado por:** Freddy Lema, 2020.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Figura 13-3:** Interfaz Crear Publicación.

**Realizado por:** Freddy Lema, 2020.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

**Figura 14-3:** Interfaz Mis Publicaciones.

**Realizado por:** Freddy Lema, 2020.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Figura 15-3:** Interfaz Postulaciones de la Publicación.

**Realizado por:** Freddy Lema, 2020.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Figura 16-3:** Interfaz Mis Mensajes.

**Realizado por:** Freddy Lema, 2020.

Interfaz de usuario gráfica, Texto, Aplicación, Teams

Descripción generada automáticamente

**Figura 17-3:** Interfaz Mensajes.

**Realizado por:** Freddy Lema, 2020.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Figura 18-3:** Interfaz Notificaciones.

**Realizado por:** Freddy Lema, 2020.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Figura 19-3:** Interfaz Configuración del Perfil.

**Realizado por:** Freddy Lema, 2020.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Figura 20-3:** Interfaz Perfil Ecológico.

**Realizado por:** Freddy Lema, 2020.

### Historias de Usuario

Las Historias de usuario son una representación de los requerimientos que fueron identificados para las funcionalidades que requieren los usuarios. Cada una de estas cuentan con algunos campos que se mencionan a continuación: Id o Número, Nombre, Usuario, Sprint Asignado, Fecha de Inicio, Fecha Fin, Descripción, Pruebas de Aceptación. Las historias de usuario permiten comprobar que las funcionalidades se cumplan correctamente a través de tareas de ingeniería con sus respectivas pruebas de aceptación.

A continuación, en la **Tabla 12-3,** se presentan un modelo de historia de usuario con su respectiva tarea de ingeniería y prueba de aceptación.

**Tabla 12-3:** Historia de Usuario.

|  |  |
| --- | --- |
| **HISTORIA DE USUARIO** | |
| **Número:** HU01 | **Nombre de la Historia:** Como usuario público deseo iniciar sesión para ingresar al sistema. |
| **Usuario:** Público | **Sprint Asignado:** 4 |
| **Puntos Estimados:** 20 | **Puntos Reales:** 20 |
| **Descripción:** Como usuario deseo iniciar sesión para ingresar al sistema. | |
| **Pruebas de Aceptación:**   * Ingresar a la página de inicio para utilizar las funcionalidades del sistema. | |

**Realizado por:** Freddy Lema, 2020.

**Tabla 13-3:** Prueba de Aceptación.

|  |  |
| --- | --- |
| **PRUEBA DE ACEPTACIÓN** | |
| **Código:** HU01-PA01 | **Nombre de la Historia:** Como usuario deseo iniciar sesión para ingresar al sistema. |
| **Nombre de la Prueba:** Ingresar al sistema como usuario público. | |
| **Responsable:** Freddy Lema | **Fecha:** 07/11/2020 |
| **Descripción:** El usuario debe iniciar sesión en el sistema. | |
| **Condiciones de Ejecución:**   * El correo del usuario debe estar registrado en la base de datos | |
| **Pasos de Ejecución:**   * Ingresar a la página de inicio de sesión * Ingresar el mail del usuario en su respetivo campo * Ingresar la contraseña en su respetivo campo * Clic en iniciar sesión | |
| **Resultado Esperado:** El usuario debe poder ingresar a la pantalla de inicio para utilizar las funcionalidades del sistema. | |
| **Evaluación de la Prueba:** Exitosa. | |

**Realizado por:** Freddy Lema, 2020.

**Tabla 14-3:** Tarea de Ingeniería.

|  |  |
| --- | --- |
| **TAREA DE INGENIERÍA** | |
| **Sprint:** 4 | **Número de Tarea:** HU01-TI01 |
| **Nombre de la Historia:** Como usuario administrador deseo iniciar sesión para gestionar mi información. | |
| **Nombre de la Tarea:** Crear las entidades y clases que permitan autenticarse. | |
| **Responsable:** Freddy Lema | **Tipo de Tarea:** Desarrollo |
| **Fecha Inicio: 04/11/2020** | **Fecha Fin: 07/11/2020** |
| **Descripción:** Se deben crear las clases de acuerdo con las entidades existentes para iniciar sesión. | |
| **Pruebas de Aceptación:**   * Clases y métodos existentes para iniciar sesión. | |

**Realizado por:** Freddy Lema, 2020.

Todas las demás historias de usuario están documentadas de la misma manera, para mas información dirigirse al **anexo A**.

### Manual de Usuario

Para que el usuario se familiarice con el sistema web se ha elaborado el manual de usuario que explica el funcionamiento y caracteristicas básicas, para ver el manual dirigirse al **anexo B.**

## Fase de finalización

En esta etapa se realiza las diferentes actividades que se definieron para la culminación del sistema para la gestión del reciclaje, además se establece el Burndown chart que hace referencia a la gestión del desarrollo del proyecto, es presentado como un gráfico en donde se muestra la velocidad en la que se completaron los requisitos planteados al inicio del desarrollo del sistema. A continuación, en la **Tabla 15-3**, se describe las actividades en la etapa de finalización del proyecto.

**Tabla 15-3:** Actividades de finalización.

|  |  |  |
| --- | --- | --- |
| **ACTIVIDAD** | **DESCRIPCIÓN** | **RESPONSABLE** |
| Documentación del sistema. | Elaborar el manual de usuario y técnico. | Desarrollador |
| Despliegue del sistema en el servidor web. | Instalación y despliegue del sistema en el servidor gratuito heroku. | Desarrollador |

**Realizado por:** Freddy Lema, 2020.

### BurnDown Chart

Al finalizar el desarrollo del Sprint Backlog se presenta a continuación la Figura, que representa el progreso del proyecto a lo largo del tiempo de desarrollo, es parte esencial de cualquier proyecto ágil y es una forma simple y clara de mostrar lo que ocurre en cada sprint.

El **Gráfico 1-3** contiene dos líneas que representan los puntos reales (línea roja) y los puntos estimados (línea azul) de cada sprint, es una comparativa que nos ayudará a comprender como avanzó el desarrollo del proyecto.

**Gráfico 1-3:** Burndown Chart.

**Realizado por:** Freddy Lema, 2020.

**3.6 BIBLIOGRAFIA**

* + 1. BIBLIOGRAFÍA GENERAL

AIMBSOR, 2017. 1.01. Arquitectura cliente/servidor - AIMB SOR. *Aimbsor* [en línea]. [Consulta: 1 diciembre 2020]. Disponible en: https://sites.google.com/site/aimbsor/introduccion-a-los-sor/1-1-arquitectura-cliente-servidor.

BADAL, H., 2017. Ventajas y desventajas de una Web App. *Linkedin* [en línea]. [Consulta: 1 diciembre 2020]. Disponible en: https://es.linkedin.com/pulse/ventajas-y-desventajas-de-una-web-app-hector-badal-mba.

BAEZ, S., 2012. Sistemas Web. *Sistemas Web :: KnowDo* [en línea]. [Consulta: 12 enero 2020]. Disponible en: http://www.knowdo.org/knowledge.php?id=39.

BAJAÑA RODRÍGUEZ, L.A., 2015. *Implementación de un sistema web para el reciclaje y donación de partes de computadoras.* [en línea]. Guayaquil: ESPOL. Disponible en: http://www.dspace.espol.edu.ec/xmlui/bitstream/handle/123456789/39920/D-84920.pdf?sequence=-1&isAllowed=y.

BBVAAPIMARKET, 2016. API REST: qué es y cuáles son sus ventajas en el desarrollo de proyectos. *BBVA API\_Market* [en línea]. [Consulta: 1 diciembre 2020]. Disponible en: https://www.bbvaapimarket.com/es/mundo-api/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos/.

BLANCARTE, O., 2017. Introducción a NodeJS (JavaScript del lado del Servidor). *Oscar Blancarte - Software Architecture* [en línea]. [Consulta: 1 diciembre 2020]. Disponible en: https://www.oscarblancarteblog.com/2017/05/29/introduccion-a-nodejs-2/.

CATALDI, I.Z., 2000. Metodología de diseño, desarrollo y evaluación de software educativo. , pp. 255.

CERVANTES, H., 2018. Arquitectura de Software. *SG Buzz* [en línea]. [Consulta: 12 enero 2020]. Disponible en: https://sg.com.mx/revista/27/arquitectura-software.

CHINININ, Z., 2012. GUIA DE ARQUITECTURA EN N-CAPAS: Estilos Arquitecturales. *GUIA DE ARQUITECTURA EN N-CAPAS* [en línea]. [Consulta: 12 enero 2020]. Disponible en: http://arquitecturancapas.blogspot.com/2012/10/estilos-arquitecturales.html.

COLMA, 2020. Ventajas y desventajas de Angular ⋆ Noticias: arte, viajes, diseño, tecnología. *News: Art, Travel, Design, Technology* [en línea]. [Consulta: 1 diciembre 2020]. Disponible en: https://es.news.colma.do/ventajas-y-desventajas-de-angular.

CÓRDOVA GARCÍA,VALLEJO LÓPEZ, M.F., Jéssica Lorena, 2013. *Determinación y evaluación de las externalidades ambientales del reciclaje de plástico, vidrio y compuestos de celulosa generados en la ciudad de Riobamba.* [en línea]. RIOBAMBA: UNACH. Disponible en: http://dspace.unach.edu.ec/bitstream/51000/778/1/UNACH-EC-IMB-2013-0004..pdf.

DIPLOMADO EN GESTIÓN DE LA CALIDAD DEL SOFTWARE, 2015. Usabilidad. *gestion de la calidad del software - norma iso-9126* [en línea]. [Consulta: 8 enero 2021]. Disponible en: https://diplomadogestioncalidadsoftware2015.wordpress.com/norma-iso-9126/calidad-interna-y-externa/usabilidad/.

EL TELÉGRAFO, 2019. Los recicladores de Riobamba piden ayuda ciudadana. *El Telégrafo - Noticias del Ecuador y del mundo* [en línea]. [Consulta: 26 enero 2020]. Disponible en: https://www.eltelegrafo.com.ec/noticias/regional/1/recicladores-riobamba-ciudadania.

ESPINOZA, C. y FERNANDA, R., 2013. Análisis comparativo entre bases de datos relacionales con bases de datos no relacionales. , pp. 88.

GALÁN, D., 2020. ¿Qué es AngularJS y por qué deberías usarlo? *IfgeekthenEveris* [en línea]. [Consulta: 3 octubre 2020]. Disponible en: https://ifgeekthen.everis.com/es/que-es-angularjs-y-por-que-deberias-usarlo.

GALVIS, J.P.P., 2015. Propuesta de notación gráfica para el modelo orientado a documentos de MongoDB. , pp. 96.

MARTINEZ, C.P., 2016. Arquitectura de software y sus beneficios. *DevExperto* [en línea]. [Consulta: 12 enero 2020]. Disponible en: https://devexperto.com/arquitectura-del-software/.

MOZILLA, 2018. Introducción a Express/Node. *Documentación web de MDN* [en línea]. [Consulta: 3 octubre 2020]. Disponible en: https://developer.mozilla.org/es/docs/Learn/Server-side/Express\_Nodejs/Introduction.

PIENSASOLUTIONS, 2017. Principales lenguajes de programación para el desarrollo web. *Blog Piensa Solutions* [en línea]. [Consulta: 12 enero 2020]. Disponible en: https://blog.piensasolutions.com/principales-lenguajes-programacion-web/.

ROBLEDANO, Á., 2019. Qué es MongoDB y características. *OpenWebinars.net* [en línea]. [Consulta: 4 octubre 2020]. Disponible en: https://openwebinars.net/blog/que-es-mongodb/.

ROSA MONCAYO, J.M., 2018. ¿Qué es REST? Conoce su potencia. *OpenWebinars.net* [en línea]. [Consulta: 3 octubre 2020]. Disponible en: https://openwebinars.net/blog/que-es-rest-conoce-su-potencia/.

ROUSE, M., 2015. ¿Qué es NoSQL (No Solo SQL)? - Definición en WhatIs.com. *SearchDataCenter&nbsp;en&nbsp;Español* [en línea]. [Consulta: 4 octubre 2020]. Disponible en: https://searchdatacenter.techtarget.com/es/definicion/NoSQL-No-Solo-SQL.

SCHIAFFARINO, A., 2019. Modelo cliente servidor: ¿Qué es? Características, Ventajas y Desventajas. *Infranetworking* [en línea]. [Consulta: 1 diciembre 2020]. Disponible en: https://blog.infranetworking.com/modelo-cliente-servidor/.