



UNIVERSIDAD DEL BÍO-BÍO, CHILE

FACULTAD DE CIENCIAS EMPRESARIALES

DEPARTAMENTO DE SISTEMAS DE INFORMACIÓN

Gestión de reciclaje en la comuna de Concepción. Propuesta de un Modelo e Implementación de una herramienta basada en TIC de apoyo a los procesos de reciclaje.

PROYECTO DE TÍTULO PRESENTADO POR FRANCISCO ESTEBAN
HERNÁNDEZ URZÚA.

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN INFORMÁTICA
DIRIGIDA POR SERGIO ARAYA.

CONCEPCIÓN, 2020

Agradecimientos

A mi familia que me apoyó en todo el proceso de mi carrera universitaria.

A mi Madre, Padre y Hermana por estar siempre conmigo y darme su apoyo incondicional.

A mi abuela por su amor incondicional.

Mis amigos que me permitió conocer la universidad, Alejandro, Brayan, Iván y Jorge. Y a ellos les deseo el mejor de los éxitos.

A toda la gente que me ayudo y compartimos momentos gratos durante mi recorrido por la universidad durante todos estos años y me forjaron para convertirme en la persona que soy ahora.

Resumen

Este proyecto se realiza para cumplir con los requisitos exigidos por la Universidad del Bío-Bío para realizar el proceso de titulación de la carrera de Ingeniería Civil en Informática.

El proyecto entrega una propuesta de modelo para el desarrollo de la gestión del reciclaje en la comuna de Concepción y el desarrollo de una herramienta TIC que apoya a los principales procesos que conlleva esta gestión.

A lo largo de este informe se describe porque surgió la idea de este proyecto, el proceso de construcción de un modelo que permite apoyar a la gestión del Reciclaje y también el desarrollo de una herramienta TIC, específicamente una aplicación móvil, con su debida documentación.

El modelo propuesto se centra en permitir una mayor comprensión de los principales procesos que son llevados a cabo en la gestión del reciclaje y las entidades que participan de ello, así como también identificar los distintos flujos de información que se comparten entre los diferentes procesos dentro del modelo, todo esto para ofrecer tanto a la municipalidad de Concepción como a la ciudadanía una ayuda al momento de enfocar esfuerzos y presupuestos a la gestión del reciclaje.

Para la realización de la aplicación móvil, se hacen uso de nuevas tecnologías y arquitecturas de software. Se eligió el marco de trabajo React Native que permite un desarrollo de aplicaciones móviles tanto para Android como IOS utilizando lenguaje Javascript y una base de datos NoSQL utilizando NodeJS para diseñar e implementar un servicio de API REST que permite interactuar a la aplicación con los datos almacenados.

Para la realización de la aplicación se utiliza la metodología cascada, permitiendo sentar cimientos sólidos para desarrollar la aplicación. Además, para obtener la información del funcionamiento del reciclaje en Concepción, se tuvieron conversaciones directamente con encargados del proceso.

Índice

| | |
|--|----|
| Agradecimientos | 2 |
| Resumen | 3 |
| Índice figuras | 9 |
| Índice Tablas | 10 |
| Capítulo 1 Definición de la Empresa | 11 |
| 1.1 Descripción de la empresa | 11 |
| 1.1.1 Organigrama | 11 |
| 1.1.2 Funciones y atribuciones de la Municipalidad | 11 |
| 1.2 Descripción del área de estudio | 12 |
| 1.3 Descripción de la problemática | 16 |
| Capítulo 2 Definición Proyecto | 20 |
| 2.1 Objetivos del proyecto | 20 |
| 2.2 Ambiente ingeniería de software | 20 |
| Capítulo 3 Marco Teórico | 22 |
| 3. 1 Introducción | 22 |
| 3. 2 Reciclaje en Chile | 23 |
| 3.3 Políticas de reciclaje en Chile | 24 |
| 3.4 Economía circular del reciclaje | 24 |
| 3.5 ¿Como funciona la economía circular? | 25 |
| 3.6 Marco de trabajo de desarrollo | 26 |
| 3.7 Javascript | 26 |
| 3.8 Aplicaciones nativas | 27 |
| 3.9 React | 27 |
| 3.10 JSX | 27 |

| | |
|--|----|
| 3.11 Virtual DOM | 28 |
| 3.12 Componentes | 28 |
| 3.13 React Native | 29 |
| 3.14 Redux | 29 |
| 3.14.1 ¿Por qué es necesario Redux? | 29 |
| 3.14.2 ¿Cómo funciona Redux? | 30 |
| 3.14.3 Actions | 30 |
| 3.14.4 Reducers en Redux | 31 |
| 3.14.5 Store en Redux | 32 |
| 3.15 Bases de datos NoSQL | 32 |
| 3.15.1 ¿Qué es NoSQL? | 32 |
| 3.15.2 Tipos de bases de datos NoSQL | 33 |
| 3.16 MongoDB | 35 |
| 3.17 JSON Web Tokens | 35 |
| 3.18 Procesos de Reciclaje | 36 |
| Capítulo 4 Definición de un Modelo | 38 |
| Capítulo 5 Especificación de Requerimientos del Software | 42 |
| 5.1 Alcances | 42 |
| 5.2 Objetivos de Software | 42 |
| 5.2.1 Objetivo General | 42 |
| 5.2.2 Objetivos específicos | 43 |
| 5.4 Requerimientos Específicos | 43 |
| 5.4.1 Requerimientos Funcionales del Sistema | 43 |
| 5.4.2 Interfaces Externas de Entrada | 44 |
| 5.4.2 Requerimientos no Funcionales | 44 |
| Capítulo 6 Factibilidad | 45 |
| 6.1 Factibilidad Técnica | 45 |

| | |
|--|----|
| 6.2 Factibilidad Operativa | 45 |
| 6.3 Factibilidad Económica | 46 |
| 6.4 Conclusión Factibilidad | 46 |
| Capítulo 7 Análisis | 47 |
| 7.1 Actores | 47 |
| 7.2 Diagramas de Casos de Uso | 48 |
| 7.2.1 Jerarquía de usuarios | 48 |
| 7.2.2 Casos de Usos Usuario | 49 |
| 7.2.3 Casos de Uso Administrador | 50 |
| 7.3 Especificación Casos de Uso | 51 |
| 7.3.1 Caso de uso: Ingresar al sistema | 51 |
| 7.3.2 Caso de Uso: Registrarse | 52 |
| 7.3.3 Caso de Uso: Cerrar sesión | 53 |
| 7.3.4 Caso de Uso: Ver Puntos de Reciclaje | 53 |
| 7.3.5 Caso de Uso: Comentar punto de reciclaje | 54 |
| 7.3.6 Caso de Uso: Ver Comentarios Punto de Reciclaje | 55 |
| 7.3.7 Caso de Uso: Registrar Reciclaje Personal | 56 |
| 7.3.8 Caso de Uso: Ver Historial de Reciclaje Personal | 58 |
| 7.3.9 Caso de Uso: Editar Perfil | 58 |
| 7.3.10 Caso de Uso: Ver Perfil de Usuario | 59 |
| 7.3.11 Caso de Uso: Ver Punto de Reciclaje | 60 |
| 7.3.12 Caso de Uso: Ver Tips de Reciclaje | 61 |
| 7.3.13 Caso de Uso: Agregar Punto de Reciclaje | 62 |
| Capítulo 8 Diseño | 63 |
| 8.1 Diseño Físico de la Base de Datos | 63 |
| 8.1.1 Descripción de Documento: Usuario | 63 |
| 8.1.2 Descripción de Documento: Material | 64 |

| | |
|--|----|
| 8.1.3 Descripción de Documento: Comentario | 64 |
| 8.1.4 Descripción de Documento: Punto Limpio | 65 |
| 8.1.5 Descripción de Documento: Recicla | 65 |
| 8.2 Diseño de Arquitectura Funcional | 66 |
| 8.3 Diseño API REST | 66 |
| 8.3.1 Principios de API REST | 67 |
| 8.3.2 API REST Construida | 72 |
| 8.3 Diseño de Interfaz y Navegación | 77 |
| 8.3.1 Navegación | 77 |
| 8.3.2 Pantalla Login | 78 |
| 8.3.3 Pantalla Registro | 79 |
| 8.3.4 Pantalla Inicio | 80 |
| 8.3.5 Pantalla Punto Limpio | 81 |
| 8.3.6 Pantalla Comentario | 82 |
| 8.3.7 Pantalla Reciclar | 83 |
| 8.3.8 Pantalla Historial Reciclaje | 84 |
| 8.3.9 Pantalla Perfil de Usuario | 85 |
| 8.3.10 Pantalla Agregar Punto Limpio | 86 |
| Capítulo 9 Pruebas | 88 |
| 9.1 Pruebas de Caja Negra | 88 |
| 9.2 Resultados Pruebas Aplicación y API | 88 |
| 9.2.1 Prueba 1: Iniciar Sesión | 89 |
| 9.2.2 Prueba 2: Registrar Usuario | 90 |
| 9.2.3 Prueba 3: Registrar Reciclaje Personal | 91 |
| 9.2.4 Prueba 4: Registrar comentario en un punto limpio. | 92 |
| 9.2.5 Prueba 5: Ingresar al sistema con Facebook | 93 |
| 9.2.6 Prueba 6: Registrar comentario utilizando la API. | 94 |

| | |
|------------------------|-----|
| Capítulo 10 Conclusión | 97 |
| Bibliografía | 99 |
| Linkografía | 101 |

Índice figuras

| | |
|--|----|
| Figura 1 Organigrama Municipalidad de Concepción | 11 |
| Figura 2 Organigrama Dirección Medio Ambiente | 13 |
| Figura 3 Economía circular | 25 |
| Figura 4 Pastel estándares tecnologías web | 26 |
| Figura 5 Base de datos clave-valor | 33 |
| Figura 6 Bases de datos documentales | 34 |
| Figura 7 Bases de datos en grafo | 35 |
| Figura 8 JSON Web Token | 36 |
| Figura 9 Ejemplo Modelo Reciclaje | 37 |
| Figura 10 DFD Superior | 40 |
| Figura 11 DFD de Contexto | 41 |
| Figura 12 Jerarquía de actores | 48 |
| Figura 13 Casos de uso Usuario | 49 |
| Figura 14 Casos de uso Administrador | 50 |
| Figura 15 Modelo Base de Datos | 63 |
| Figura 16 Arquitectura Funcional | 66 |
| Figura 17 Diseño Api REST | 67 |
| Figura 18 API Request 1 | 70 |
| Figura 19 API Request 2 | 71 |
| Figura 20 Objeto JSON | 72 |
| Figura 21 Mapa Navegación App | 77 |
| Figura 22 Pantalla de Login | 78 |
| Figura 23 Pantalla Registro | 79 |
| Figura 24 Pantalla Inicio | 80 |
| Figura 25 Pantalla Punto Limpio | 81 |
| Figura 26 Modal Comentario | 82 |
| Figura 27 Pantalla Reciclar | 83 |
| Figura 28 Pantalla Historial Reciclaje | 84 |
| Figura 29 Pantalla Perfil Usuario | 85 |
| Figura 30 Pantalla Crear Punto Limpio | 86 |
| Figura 31 Pantalla Crear Punto Limpio2 | 87 |

Índice Tablas

| | |
|--|----|
| Tabla 1 Requerimientos Funcionales | 43 |
| Tabla 2 Caso de Uso: Ingresar al sistema | 51 |
| Tabla 3 Caso de Uso: Registrarse | 52 |
| Tabla 4 Caso de Uso: Cerrar Sesión | 53 |
| Tabla 5 Caso de Uso: Ver Puntos de Reciclaje | 54 |
| Tabla 6 Caso de Uso: Comentar Punto de Reciclaje | 55 |
| Tabla 7 Caso de Uso: Ver Comentarios Punto de Reciclaje | 56 |
| Tabla 8 Caso de Uso: Registrar Reciclaje Personal | 57 |
| Tabla 9 Caso de Uso: Ver Historial de Reciclaje Personal | 58 |
| Tabla 10 Caso de Uso: Editar Perfil | 59 |
| Tabla 11 Caso de Uso: Ver Perfil de Usuario | 60 |
| Tabla 12 Caso de Uso: Ver Punto de Reciclaje | 61 |
| Tabla 13 Caso de Uso: Ver Tips de Reciclaje | 61 |
| Tabla 14 Caso de Uso: Agregar Punto de Reciclaje | 62 |
| Tabla 15 Descripción Documento: Usuario | 64 |
| Tabla 16 Descripción Documento: Material | 64 |
| Tabla 17 Descripción Documento: Comentario | 65 |
| Tabla 18 Descripción Documento: Punto Limpio | 65 |
| Tabla 19 Descripción Documento: Recicla | 66 |
| Tabla 20 Métodos HTTP | 69 |
| Tabla 21 Endpoint Comentario | 73 |
| Tabla 22 Endpoint Material | 73 |
| Tabla 23 Endpoint Punto Limpio | 74 |
| Tabla 24 Endpoint Recicla | 75 |
| Tabla 25 Endpoint Usuario | 75 |
| Tabla 26 Endpoint Register | 76 |
| Tabla 27 Endpoint Login | 76 |
| Tabla 28 Caso de Prueba: Iniciar Sesión | 89 |
| Tabla 29 Caso de Prueba: Registrar Usuario | 91 |
| Tabla 30 Caso de Prueba: Reciclaje Personal | 92 |
| Tabla 31 Caso de Prueba: Registrar Comentario | 93 |

Tabla 32 Caso de Prueba: Ingresar con Facebook 94

Tabla 33 Caso de Prueba: Registrar Comentario Mediante API 95

Capítulo 1 Definición de la Empresa

1.1 Descripción de la empresa

Según la Constitución las municipalidades son corporaciones autónomas de derecho público, con personalidad jurídica y patrimonio propio, cuya finalidad es satisfacer las necesidades de la comunidad local y asegurar su participación en el progreso económico, social y cultural de las respectivas comunas.

1.1.1 Organigrama

En la figura 1 se muestra el organigrama de la Municipalidad de Concepción.



Figura 1 Organigrama Municipalidad de Concepción

1.1.2 Funciones y atribuciones de la Municipalidad

Según la constitución, las siguientes son las funciones y atribuciones que posee la municipalidad:

- a) Elaborar, aprobar y modificar el plan comunal de desarrollo cuya aplicación deberá armonizar con los planes regionales y nacionales.

- b) La planificación y regulación de la comuna y la confección del plan regulador comunal, de acuerdo con las normas legales vigentes.
- c) La promoción del desarrollo comunitario.
- d) Aplicar las disposiciones sobre transporte y tránsito públicos, dentro de la comuna, en la forma que determinen las leyes y las normas técnicas de carácter general que dicte el ministerio respectivo.
- e) Aplicar las disposiciones sobre construcción y urbanización, en la forma que determinen las leyes, sujetándose a las normas técnicas de carácter general que dicte el ministerio respectivo, y
- f) El aseo y ornato de la comuna. Respecto a los residuos domiciliarios, su recolección, transporte y/o disposición final corresponderá a las municipalidades, con excepción de las que estén situadas en un área metropolitana y convengan con el respectivo gobierno regional que asuma total o parcialmente estas tareas.

1.2 Descripción del área de estudio

Este proyecto se encuentra enfocado en el cuidado del medio ambiente, por lo que a continuación se describe la dirección de medioambiente de la Municipalidad de Concepción.

La dirección de medio ambiente en una unidad de línea, de dependencia directa del alcalde y se encuentra bajo la dirección de Andrea Aste von Bennewitz.

Dependen de la Dirección de Medio Ambiente las Secciones de Recursos Naturales y Cambio Climático, de Gestión de Residuos Orgánicos, de Saneamiento Ambiental, de Educación Ambiental y Participación Ciudadana.

Asimismo, depende directamente de la Dirección de Medio Ambiente la Oficina de Inspección. En la figura 2 se muestra el organigrama de la dirección.

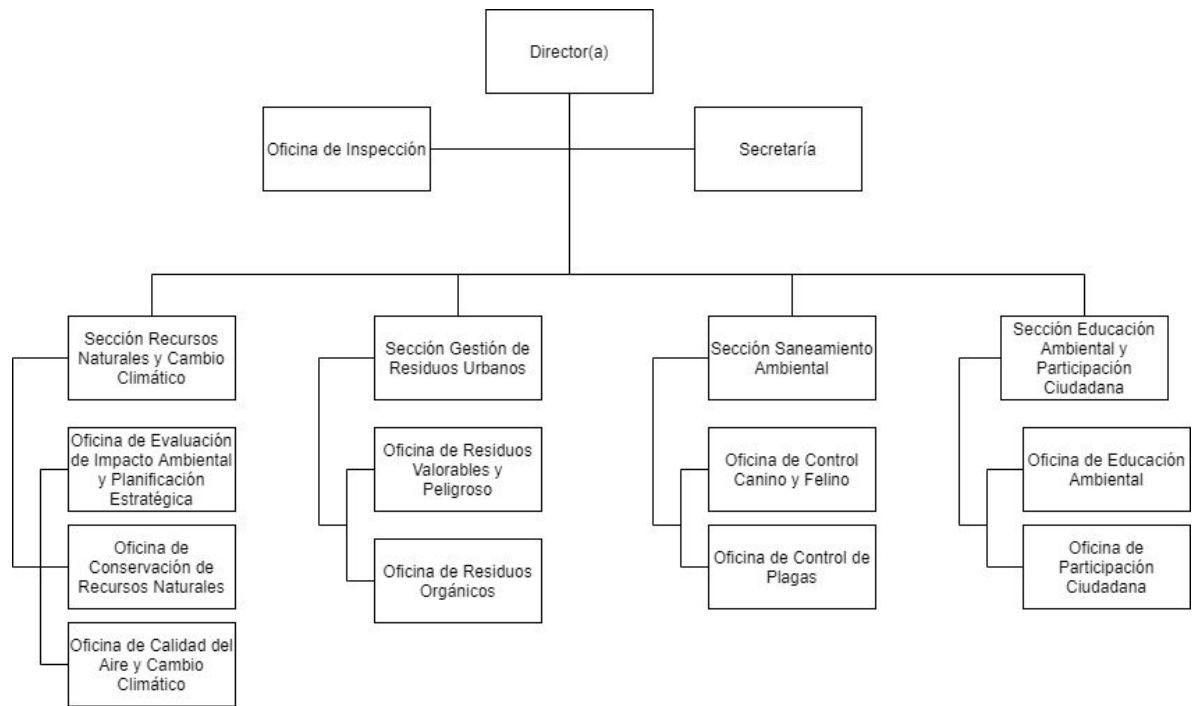


Figura 2 Organigrama Dirección Medio Ambiente

Objetivos

- Incorporar la dimensión ambiental en el diseño y ejecución de políticas, planes y programas municipales, contribuyendo al desarrollo sustentable de la comuna mediante acciones que apunten a la preservación, uso racional y reparación del patrimonio ambiental comunal.

Funciones

- Asesorar al alcalde en materias relativas a la planificación, gestión y control ambiental en el territorio comunal.
- Velar por el cumplimiento de la legislación ambiental vigente en materias relativas a la protección y manejo racional de los recursos naturales: agua, suelo, aire.
- Elaborar estudios y planes estratégicos ambientales en el marco del Plan de Desarrollo Comunal.

- Elaborar proyectos y programas ambientales en conjunto con el Ministerio de Medio Ambiente y otras reparticiones del Estado.
- Participar en la evaluación ambiental de los proyectos o actividades que se emplacen en el territorio comunal, a través del Sistema de Evaluación del Impacto Ambiental.
- Fiscalizar la Ordenanza Medio Ambiental comunal.
- Recepcionar denuncias y activar las coordinaciones correspondientes con la Subsecretaría de Medio Ambiente para su adecuada fiscalización, o con los organismos sectoriales competentes, según corresponda.
- Implementar convenios de cooperación para la ejecución de estudios, proyectos y programa relacionados con la gestión ambiental, en alianza con instituciones públicas y privadas.
- Fomentar la educación ambiental en el Sistema Comunal de Educación.
- Desarrollar y coordinar actividades y programas de capacitación en materias ambientales a las organizaciones sociales y comunitarias, así como a los docentes y alumnos de la comunidad educativa comunal.
- Promover y colaborar con toda entidad pública y privada y organizaciones sociales y comunitarias que deseen desarrollar proyectos e iniciativas en beneficio del medio ambiente local.
- Desarrollar programas permanentes de conservación de lagunas, humedales, bosque nativo y espacios públicos naturales.
- Desarrollar una estrategia de conservación y protección de la biodiversidad sustentada en los ecosistemas naturales que existen en el territorio comunal.
- Prevención y control de plagas naturales y artificiales, en colaboración con otros entes del Estado responsables de esa actividad.
- Contribuir al control de vectores, perros vagos, y en general de todos aquellos agentes biológicos, animales o invertebrados que puedan transmitir enfermedades a los habitantes de la comuna, en coordinación con la autoridad sanitaria correspondiente.
- Otras funciones que la ley señale o la autoridad superior le asigne, de conformidad con la legislación vigente.

A continuación, también se explica el funcionamiento de las secciones y oficinas que tienen relación directa con el desarrollo de este proyecto.

Sección Gestión de Residuos Orgánicos

Objetivo

- Implementar el Plan de Gestión de Residuos Sólidos Municipales y apoyar al Ministerio de Medio Ambiente en la implementación de la política nacional de residuos sólidos.

Oficina de Residuos Valorables y Peligrosos

Funciones:

- Desarrollar e implementar planes y programas para la correcta gestión y disposición final de los residuos peligrosos generados en la comuna.
- Implementar proyectos y programas de reciclaje de los productos establecidos en la “Ley de Responsabilidad Extendida al Productor”.
- Implementar “Puntos Limpios” comunales, para la recolección de residuos reciclables.

Oficina de Residuos Orgánicos

Funciones:

- Desarrollar e implementar planes y programas de gestión integrada de residuos domiciliarios, fomentando el reciclaje y el compostaje.
- Desarrollar proyectos y programas de utilización de compost en iniciativas de huertos familiares y comunitarios para fomentar el reciclaje de residuos orgánicos y alimentación sana.
- Generar proyectos asociados a la gestión de residuos sólidos orgánicos con tecnologías de punta e innovadoras relacionadas a la obtención de energía.

Sección Educación Ambiental y Participación Ciudadana

Objetivo

- Promover la corresponsabilidad ambiental, en los ámbitos comunitario, empresarial, escolar, universitario y en el conjunto de los servicios públicos, para apoyar la gestión estratégica municipal en sustentabilidad comunal e intercomunal.

Oficina de Educación Ambiental

Funciones

- Diseñar y ejecutar procesos de transferencia de información ambiental local y actual, en conjunto con las Secciones de la Dirección, hacia los establecimientos educacionales de la comuna, contribuyendo con la formación de profesores, alumnos, padres y apoderados, para que actúen corresponsablemente en la gestión ambiental estratégica del municipio en el territorio en que se emplazan sus unidades educativas.
- Diseñar y ejecutar procesos de transferencia de información ambiental local y actual, en conjunto con los departamentos de la Dirección del Medio Ambiente, hacia dirigentes y vecinos de organizaciones territoriales y funcionales, para que actúen corresponsablemente con la gestión ambiental estratégica del municipio en sus territorios barriales.

1.3 Descripción de la problemática

Actualmente es más importante que nunca velar por el cuidado del medio ambiente. Que cada persona aporte al planeta con su granito de arena es sumamente relevante.

Una de las maneras que existen para ayudar en el cuidado del medioambiente, y de más fácil acceso a las personas en general, es mediante los puntos limpios o de reciclaje que hay en las diferentes ciudades y comunas. Pero esto a veces se vuelve una tarea complicada, ya que no hay mucha información disponible respecto al lugar en donde se encuentran estos puntos, algunos están no están en condiciones óptimas y el periodo de recogida no es el correcto, lo que hace que los puntos de reciclajes acumulen residuos.

Por estas razones nace la idea de crear una herramienta basada en TIC que permita a los usuarios visualizar los puntos limpios y de reciclaje cercanos a su residencia y ver la calidad de estos.

Además, es de suma importancia el desarrollo de un modelo que permita apoyar el proceso de reciclaje, para gestionar bien los recursos y para entregar un mejor servicio de este tipo a la comunidad. Dado que Concepción es una de las ciudades más importantes de Chile, y que cuenta con una gran cantidad de población resulta relevante efectuar un proyecto de este tipo en dicha ciudad.

Actualmente la comuna de Concepción cuenta con una serie de procesos que ayudan al funcionamiento del reciclaje y a continuación se procederá a explicar como se realiza actualmente dicha labor.

Lo primero que hay que mencionar, es que la Dirección de Medio Ambiente de la Municipalidad es la que se encarga de dirigir todo, desde la recogida, la instalación de puntos limpios, mantención de los puntos limpios y la posterior reutilización de los residuos reciclados.

Para comenzar, los puntos limpios son emplazados en puntos estratégicos escogidos por la Municipalidad. En cada uno de estos puntos se pueden reciclar materiales como plásticos, papeles, desechos orgánicos, cartones, metales como latas de aluminio, conservas, latas de bebida, etc.

Además de los puntos limpios que existen, también hay un solo punto actualmente en la comuna en la que se pueden reciclar residuos peligrosos, como vidrios, aceites y baterías.

La Municipalidad cuenta solo con 5 puntos limpios que son gestionados por ellos y se encargan de su recolección, pero hay una gran variedad de otros puntos de reciclaje que son administrados por privados y no son de mayor conocimiento para la comunidad, por lo que se hace relevante crear un modelo que permita gestionar tanto los puntos limpios municipales como los privados, y que esta información sea entregada a toda la comunidad.

El Municipio también se encarga de la recogida y mantención de los residuos en los puntos limpios, realizando recogidas de forma periódica, así como también la propia limpieza de los lugares. Al momento de recoger los residuos, el Municipio hace una inspección superficial, separando los materiales que realmente se pueden reciclar de los que no, para su posterior recogida y derivación a los centros procesadores de materiales y/o vertederos según corresponda. Los residuos peligrosos recogidos que no se pueden reciclar, son gestionados de forma diferente, ya que son derivados a un vertedero

especialmente diseñado para el control de residuos peligrosos, cuidando los efectos que podrían producir en el entorno y medio ambiente, estos desechos suelen ser aceites, baterías o vidrios, que por un u otro motivo no pueden ser reciclados para volver a ser reutilizados.

Los Centros Procesadores mencionados anteriormente son seleccionados por la Municipalidad y tienen convenios. Actualmente son dos, y el Municipio de encarga de ir a dejar de forma periódica los residuos para que sean procesados y revalorizados. El Municipio solo se encarga de la recogida de residuos, mantención de puntos limpios y la vinculación con la comunidad, entregar la información necesaria a la población y educar, ya que la ciudadanía y su participación es el factor más importante de todo este proceso.

Los Centros Procesadores se encargan de darle nuevo valor a los residuos. Una vez que el Municipio descarga los residuos en los centros, comienza el proceso de separación de los materiales, ya que no todo lo que la ciudadanía desecha en los puntos limpios realmente se puede reciclar. Este es un arduo proceso en el que deben separar todos los residuos y seleccionar los que son aptos para reciclar de los que no lo son. Luego de que se realiza esta gestión, se continua con el proceso de reciclaje, para poder darle nuevo valor en el mercado a los diferentes residuos.

Cuando se separan los materiales que no se pueden reciclar, se avisa a la Municipalidad para que se haga la recogida de los materiales para que posteriormente estos sean llevados al correspondiente vertedero.

La venta de los materiales reciclados son de total beneficio para los centros procesadores, la Municipalidad solo se encarga de entregar los materiales y recogerlos.

Además de todo lo mencionado anteriormente, todos estos procesos de recogida son realizados por un solo camión municipal, especialmente encargado del reciclaje en la comuna. A este camión se le realiza mantención de manera mensual en el propio mecánico municipal.

En general todo el proceso es llevado a cabo por el Municipio de la comuna, salvo por el reciclaje mismo de los materiales y volver a darle valor de mercado a los mismos, pero todos estos procesos no responden a una gestión global. Cada uno de estos procesos son realizados de manera independiente, obviamente con previa coordinación, pero siguen siendo distintos procesos.

En conversaciones con una de las personas encargadas del proceso de gestión de reciclaje en Concepción, indicó que ningún Municipio cuenta con un Modelo de Gestión de Reciclaje que englobe todo el proceso, lo que hace aún más importante el desarrollo de un modelo que pueda relacionar todos estos procesos y así ayudar a la gestión y toma de decisiones, para enfocar los esfuerzos donde sean realmente requeridos y entregar un mejor servicio a la comunidad y ayudar al medio ambiente.

También sería importante contar con una base de datos que contenga todos los puntos limpios, no solo los que son gestionados por el Municipio, sino también los por privados, para así mejorar aún más el proceso y entregar esta información a toda la comunidad.

Por todo lo mencionado anteriormente, se hace de suma relevancia el desarrollo de un modelo que permita identificar y describir los principales procesos que son llevados a cabo y construir una herramienta que invite a la comunidad a reciclar y tener información importante respecto a los diferentes puntos de reciclaje.

Capítulo 2 Definición Proyecto

2.1 Objetivos del proyecto

General:

- Diseñar un modelo del proceso de reciclaje y crear un prototipo de herramienta basada en TIC que permita apoyar la gestión de dicho proceso, en la comuna de Concepción.

Específicos:

- Estudiar conceptos asociados a los procesos de reciclaje a partir de la literatura existente y de experiencias de distintas ciudades.
- Conocer y analizar los procesos de reciclaje desarrollados actualmente en la comuna de Concepción, para comprender su funcionamiento y desarrollo.
- Establecer los subprocesos asociados al proceso de reciclaje, las instancias que participan y las relaciones entre estos, de manera de diseñar un modelo conceptual que los represente.
- Identificar procesos más relevantes que constituyan la base para diseñar y construir el prototipo de herramienta basada en TIC, que permita apoyar la gestión del proceso de reciclaje.
- Construir el prototipo de herramienta basada en TIC que permita apoyar la gestión del proceso de reciclaje.

2.2 Ambiente ingeniería de software

Como metodología para el desarrollo de esta aplicación se utilizará la metodología cascada, lo que permitirá una definición sólida de los requerimientos que serán necesarios para construir esta aplicación basada en el modelo de gestión del reciclaje.

Para el desarrollo de la herramienta TIC se hará uso de tecnologías móviles y de servicios API REST, por lo que se utilizarán las siguientes herramientas para la confección de la aplicación:

- React Native: Para el desarrollo de la aplicación móvil.
- NodeJS: Para la comunicación entre la base de datos y la aplicación.

- MongoDB: Como base de datos de la aplicación.
- Redux: Para el control de la información dentro de la aplicación.
- Postman: Para probar funcionamiento API REST.
- ExpressJS: Marco de trabajo (framework) de NodeJS.
- Visual Studio Code: Para escribir el código del software.
- Expo: conjunto de herramientas para ayudar en el desarrollo de aplicaciones en React Native.

Capítulo 3 Marco Teórico

3.1 Introducción

Existen distintas formas de definir un marco teórico. Sin embargo, el objetivo de este es dar a conocer una serie de ideas o conceptos coherentemente organizados de manera que sean fáciles de comunicar a los demás. Es por esto, que en el presente capítulo se describirán los conceptos más relevantes en el ámbito del desarrollo de aplicaciones móviles, así como también los aspectos más importantes en el proceso de reciclaje.

Se pretende diseñar un modelo que represente la gestión del proceso de reciclaje y a partir de ello, desarrollar un prototipo de una herramienta basada en las tecnologías de la información que represente los procesos más importantes de dicho modelo. Para poder diseñar el modelo se necesita primero comprender como funciona el reciclaje y los procesos que este conlleva, realizar análisis de información existente respecto a políticas de reciclaje en Chile y como es llevado a cabo el proceso a lo largo del país en las diferentes ciudades, especialmente en Concepción.

Se describirán a continuación las características más importantes de las tecnologías utilizadas para el desarrollo de la herramienta basada en las tecnologías de la información, en este caso se optó por la construcción de una aplicación móvil. Las tecnologías utilizadas son, Javascript, React Native que sirve para desarrollar aplicaciones móviles nativas, base de datos NoSQL MongoDB, Redux que permite controlar el estado de la información de un sistema, NodeJS un marco de trabajo basado en el lenguaje Javascript, Express que es un marco de trabajo de NodeJS, API REST y JSON Web Tokens (JWT).

Finalmente, y para concluir este capítulo, se describirá el patrón de diseño utilizado para el desarrollo de esta aplicación móvil.

3. 2 Reciclaje en Chile

Para hablar de reciclaje debemos saber que lo que significa reciclar. Según la real academia española (RAE) reciclar es “someter a un material usado a un proceso para que se pueda volver a utilizar”. El tema de la reutilización de materiales hoy en día se ha vuelto un tema muy relevante para la sociedad, debido al calentamiento global que está sufriendo el planeta. Este efecto básicamente produce un aumento en la temperatura media del planeta.

Los residuos orgánicos, están constituidos por una mezcla heterogénea de materiales, dentro de los cuales se encuentran los residuos biodegradables u orgánicos, papel, cartón, plástico, vidrio, metales, entre otros. Estos residuos, al descomponerse emiten gases de efecto invernadero, especialmente metano. Los gases de efecto invernadero, contribuyen a atrapar el calor generado por los rayos solares en la atmósfera. Aunque estos gases son naturales y fundamentales de la atmósfera, el problema se produce cuando hay una elevada cantidad de estos.

El sector de residuos orgánicos es el responsable de entre 3% y 4% de las emisiones de gases (EPA, 2013). Al ver estos números se podría tener una interpretación errónea y pensar que no es importante, pero en cifras representa aproximadamente 2000 millones de toneladas de dióxido de carbono equivalente al año, igualando a industrias como la de aviación internacional y el transporte marítimo. Además de esto, los vertederos mal gestionados son responsables del 75% de las emisiones de metano a la atmósfera.

Viendo estos datos, es posible afirmar que los residuos producidos por las personas e industrias día a día tienen un impacto real en el medio ambiente, influyendo directamente en el calentamiento global que aqueja al planeta.

En Chile, el 21% de las personas creen que la basura y la suciedad en las calles causan problemas ambientales (CADEM 2015) y especialmente en Concepción un 30% lo cree así. Además, un 34% de las personas en Concepción dicen reciclar residuos en puntos limpios de la comuna y de las personas que no reciclan se encuentran las razones de que “no hay donde reciclar”, “falta de costumbre”, “falta de tiempo”, “comodidad” y “falta de información”, siendo la que más resalta “no hay donde reciclar” con un 35%.

Para realizar el proceso de reciclaje, generalmente las personas se acercan a puntos verdes o puntos de reciclaje. Un punto limpio es una instalación donde se recogen y almacenan temporalmente y de forma gratuita los residuos domésticos que, por su gran

volumen o peligrosidad, no deben arrojarse a la bolsa de la basura ni depositarse en los contenedores de la calle. Cada punto verde define los materiales que ahí se depositan.

3.3 Políticas de reciclaje en Chile

En Chile los encargados de recolectar, transportar y eliminar por los métodos adecuados las basuras, residuos y desperdicios que se depositen o produzcan en la vía urbana son los municipios de cada ciudad, por lo tanto, ellos son los encargados de gestionar los distintos procesos de reciclaje que pudiera haber.

3.4 Economía circular del reciclaje

La economía circular es un concepto más bien económico que se relaciona directamente con la sostenibilidad, y cuyo objetivo es que el valor de los productos, los materiales y los recursos (agua, energía, etc.) se mantengan en la economía durante el mayor tiempo posible, y que se reduzca al mínimo la generación de residuos. Se trata de implementar una nueva economía, circular (no lineal), basada en el principio de cerrar el ciclo de vida de los productos, los servicios, los residuos, los materiales, el agua y la energía.

En realidad, se presenta un nuevo concepto económico, que va mas allá que las famosas 3 R, reducir, reusar y reciclar. Este nuevo sistema se basa en algo que se llama el ecodiseño de productos, teniendo en cuenta procesos de fabricación y el uso de materiales, con objetivo principal que se consuma poca energía y no se generen desechos. Estos productos que se fabrican deben poder extender su vida útil siendo fáciles de reparar y cerrar el círculo siendo reutilizables y fáciles de desmontar.

En la siguiente imagen se explica de una manera sencilla el proceso por el que sigue la economía circular:

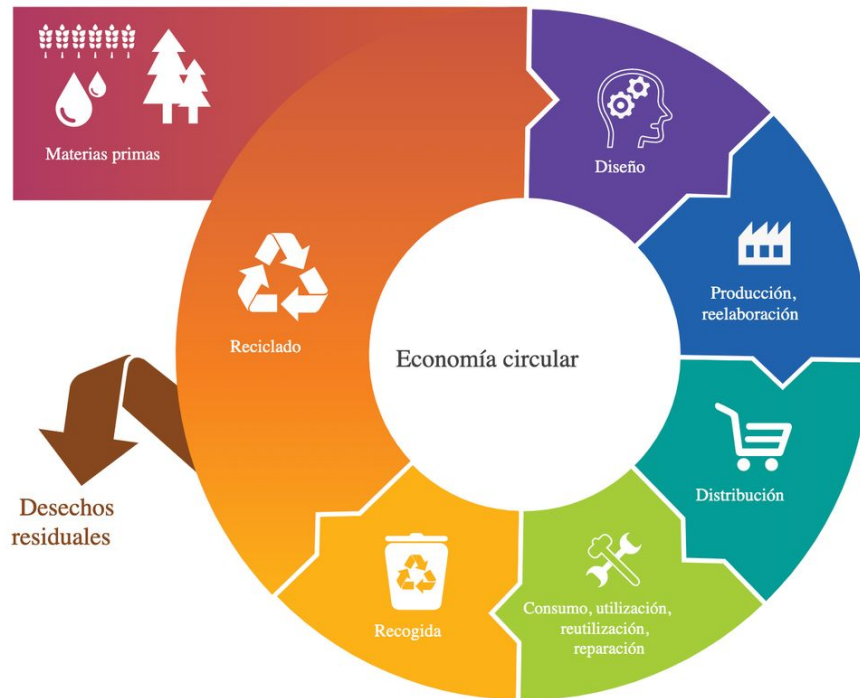


Figura 3 Economía circular

3.5 ¿Como funciona la economía circular?

El principio fundamental de este tipo de economía es la utilización de los residuos de unos como materia prima de otros, lo que reduce de manera drástica tanto la generación de residuos como la extracción de nuevos materiales y materias primas.

Además, dentro de este concepto se engloban otros principios como considerar los impactos ambientales a lo largo de todo el ciclo de vida de los bienes y servicios, priorizar el uso a la posesión y la prestación de un servicio frente a la venta de un bien, la reparación de los productos estropeados y por último la reutilización y el reciclaje.

La puesta en práctica de estos principios conlleva un gran número de beneficios tanto para el medio ambiente como para la sociedad, ya que la economía circular considera también un punto clave, la inclusión social. Entre estos beneficios de la economía circular destacan:

- Disminución del uso de los recursos, reducción de la producción de residuos y limitación del consumo de energía.
- Permite obtener una ventaja competitiva en el contexto de la globalización ya que disminuye la dependencia de unos suministros externos e inciertos.

3.6 Marco de trabajo de desarrollo

Un marco de trabajo de desarrollo, en programación, por lo general es definido como una aplicación o conjunto de funciones que permiten el rápido desarrollo de aplicaciones mediante el uso de librerías y funcionalidades ya creadas. Los marcos de trabajo o también conocidos como frameworks, permite que el desarrollador no esté continuamente “reinventando la rueda”, escribiendo el mismo código una y otra vez, sino que se pueda centrar en el problema a resolver y no en actividades que generalmente son de uso común y ya han sido resueltas por otros programadores.

Además, permite tener orden y control tanto del código como de la organización de los archivos ya que momento de realizar la instalación de alguno de estos frameworks, estos ya poseen una estructura definida.

3.7 Javascript

JavaScript es un lenguaje de programación que permite realizar actividades complejas en una página web cada vez más una página web hace más cosas que sólo mostrar información estática. Es la tercera capa del pastel de los estándares en las tecnologías para la web, dos de las cuales son (HTML y CSS).

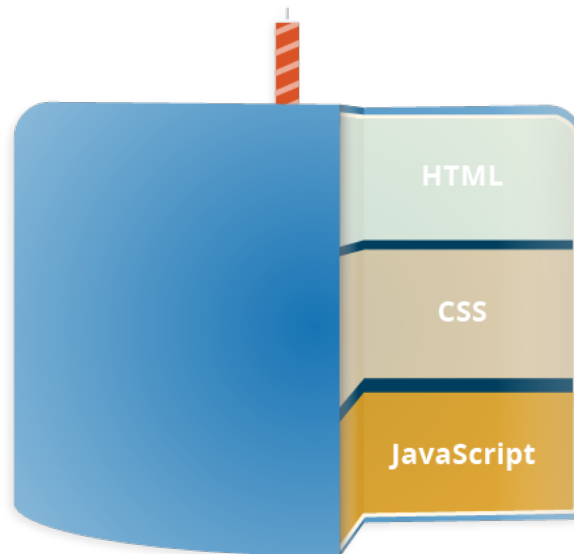


Figura 4 Pastel estándares tecnologías web

- HTML es un lenguaje de marcado que usa la estructura para dar un sentido al contenido web, por ejemplo, define párrafos, cabeceras, tablas, imágenes y vídeos en la página.

- CSS es un lenguaje de reglas en cascada que se utiliza para aplicar un estilo al contenido en HTML, por ejemplo, colocando colores de fondo, fuentes y marginando el contenido en múltiples columnas.
- JavaScript Es un lenguaje de programación que permite crear contenido nuevo y dinámico, controlar archivos de multimedia, crear imágenes animadas y muchas otras cosas más.

3.8 Aplicaciones nativas

Una aplicación móvil nativa, es una aplicación que está codificada en un lenguaje de programación específico para la plataforma que está destinada. Por ejemplo, Java para el desarrollo de aplicaciones en Android. Las aplicaciones nativas proveen buen rendimiento y alto grado de fiabilidad. También tienen acceso a las diferentes características que posee un dispositivo, como lo son la cámara y los contactos. Además, permite que los usuarios puedan usar algunas aplicaciones sin necesidad de internet. Sin embargo, el desarrollo de este tipo de aplicaciones es costoso, ya que debe construirse una para cada tipo de sistemas móviles.

3.9 React

React es una librería Javascript creada por Facebook utilizada en el desarrollo web para diseñar y construir interfaces de usuario de manera simple. El desarrollo en esta librería se basa principalmente en los siguientes aspectos: JSX, Virtual DOM y componentes.

3.10 JSX

React introduce una manera completamente nueva de escribir código para el desarrollo de páginas web. Mezcla código Javascript con HTML para así no tener que separar en diferentes archivos.

Cuando un navegador web carga una página en HTML, este genera un Document Object Model (DOM), una representación de cómo está organizado el sitio web. Los desarrolladores pueden agregar contenido dinámico a los sitios webs modificando el DOM con lenguajes de programación como Javascript.

JSX(Javascript eXtension) es una extensión de React que hace más simple a los desarrolladores modificar el DOM usando simplemente código estilo HTML.

Esto no solo es por un tema de conveniencia, usar JSX para actualizar el DOM de un sitio web provoca mejoras significativas en el rendimiento del sitio y en la eficiencia del desarrollo. Todo esto gracias al Virtual DOM de React.

3.11 Virtual DOM

Si no se usa React, el sitio web usará solo HTML para actualizar el DOM (el proceso que hace a la página “cambiar” en pantalla sin que el usuario tenga que refrescar manualmente la página). Esto funciona bien en páginas simples, páginas estáticas, pero para sitios dinámicos que involucran alta interacción con el usuario puede volverse un problema (debido a que el todo el DOM necesita recargarse cada vez que el usuario clickea una funcionalidad que refresque la página).

Pero, si el sitio utiliza JSX para manipular y actualizar el DOM, React crea una característica llamada Virtual DOM. El Virtual DOM es una copia del DOM del sitio, y React utiliza esta copia para ver cuáles partes del DOM actual necesitan cambiar cuando ocurre algún evento (por ejemplo, un usuario clickea un botón).

Digamos que un usuario ingresa un comentario en un post y presiona el botón “Comentar”. Sin utilizar React, todo el DOM se tendría que actualizar para reflejar este cambio. React, por otro lado, observa todo el Virtual DOM para ver que cambió después de la acción de un usuario y selectivamente actualiza la sección del DOM que corresponde.

Este tipo de actualizaciones selectivas toma menos poder de cómputo y menos tiempo de carga, lo que ayuda mucho en sitios webs complejos donde constantemente el usuario está interactuando y realizando acciones.

3.12 Componentes

El Desarrollo de sitios webs utilizando React se basa en la construcción páginas webs utilizando varios de los llamados componentes. Los componentes son trozos de código que sirven para ser reutilizados. Por ejemplo, un botón. En vez de programar un botón diferente para cada lugar donde se requiera, se diseña un botón que sirva de base para todo el resto, y donde solo cambian las propiedades de este, por ejemplo, el texto que muestra el botón.

3.13 React Native

React Native es un framework creado por Facebook que permite desarrollar aplicaciones móviles nativas utilizando solo código Javascript, tanto para sistemas Android como iOS.

La ventaja de utilizar este framework es que solo se necesita saber Javascript, HTML y CSS para poder desarrollar aplicaciones, lo que facilita el desarrollo para diferentes sistemas operativos, ya que no hay necesidad de aprender Java para desarrollar en Android o Swift para desarrollar en iOS. Se escribe solo código Javascript una vez y funciona para ambos sistemas operativos, lo que ahorra tiempos y costos de desarrollo.

Además, el código escrito en Javascript es transpilado a código nativo dependiendo del sistema operativo, lo que hace que el rendimiento de las aplicaciones no se vea disminuido y fuese como si hubiese sido escrito con código nativo.

3.14 Redux

En palabras simples, Redux es una librería de Javascript que permite manejar el estado de las aplicaciones escritas en el mismo lenguaje. Con estado se hace referencia a las diferentes variables que pueda utilizar el sistema, por ejemplo, los datos de un usuario, su nombre, edad, correo, etc. Redux establece un estado general accesible desde toda la aplicación, lo que permite que desde cualquier sitio de la aplicación se pueda acceder a cualquier variable o dato que se requiera.

3.14.1 ¿Por qué es necesario Redux?

En aplicaciones escritas con librerías Javascript como React, Angular o Vue se codifican muchos componentes que se relacionan entre sí para dar funcionamiento a la aplicación. Cada uno de estos componentes posee su propio estado(variables). En aplicaciones sencillas pasar información de un componente a otro no es una tarea compleja, pero a medida que los sistemas van creciendo en tamaño y complejidad, traspasar información de un componente a otro que no tiene relación directa se vuelve una tarea compleja y a veces enredada.

En el caso de React se recomienda que los componentes solo compartan información entre padre e hijo, y no más que eso para no complejizar el seguimiento del flujo de la información.

Compartir información entre componentes que no están directamente relacionados puede convertirse en una tarea ardua y difícil de predecir el comportamiento que esta tendrá, además de pasar información a componentes que no la necesitan en algunas ocasiones.

Con esto queda claro que el manejo de estados se vuelve complejo a medida que la aplicación crece. Por eso necesario contar con una herramienta de manejo de estados que permita tener un control en la aplicación como Redux.

3.14.2 ¿Cómo funciona Redux?

Redux establece un almacén (store en inglés) que mantiene el estado global de la aplicación. Cada componente puede acceder a este almacén desde cualquier parte, sin tener que comunicarse con otros componentes para poder acceder a cierta información.

Redux posee tres conceptos fundamentales en los que basa su funcionamiento. Actions, Reducers y Store. A continuación se explicarán cada uno de estos conceptos y como se relacionan entre sí.

3.14.3 Actions

En palabras simples, las acciones son distintos eventos que pueden ocurrir en la aplicación. Las actions son la única forma en la que se puede enviar datos desde la aplicación al store de Redux. La información puede provenir de las interacciones de los usuarios, llamadas a APIs, o incluso envíos de formularios.

Las actions son enviadas usando el método `dispatch()`. Las acciones son objetos de Javascript, y deben tener un tipo de propiedad para indicar el tipo de acción que debe ser llevada a cabo. También las actions deben poseer una variable llamada "payload" que contiene la información que debe ser trabajada procesada por la action. Las actions son creadas a través de un action creator.

A continuación, se muestra un ejemplo de una action que podría ser despachada al loguearse en una aplicación.

```
{
  type: "LOGIN",
  payload: {
    username: "foo",
    password: "bar"
  }
}
```

El siguiente es un ejemplo de su action creator.

```
const setLoginStatus = (name, password) => {
  return {
    type: "LOGIN",
    payload: {
      username: "foo",
      password: "bar"
    }
  }
}
```

Como se explico anteriormente, las actions deben contener la propiedad tipo y un payload para ser almacenado.

3.14.4 Reducers en Redux

Reducer son funciones puras que toman el actual estado de una aplicación, realizan una acción, y retornan un nuevo estado. Esos estados son guardados como objetos, y especifican como el estado de una aplicación cambia en respuesta a la acción enviada al store.

A continuación, se muestra un ejemplo de cómo funcionan los reducers en Redux:

```
const LoginComponent = (state = initialState, action) => {
  switch (action.type) {
    // Este reducer procesa la acción de tipo "LOGIN"
    case "LOGIN":
      return state.map(user => {
        if (user.username !== action.username) {
          return user;
        }

        if (user.password === action.password) {
          return {
            ...user,
            login_status: "LOGGED IN"
          }
        }
      });
    default:
      return state;
  }
};
```

Como función pura, los reducers no cambian la información del objeto que reciben, ni tampoco producen algún efecto colateral en la aplicación. Dado un objeto igual, deben siempre producir el mismo resultado.

3.14.5 Store en Redux

El store mantiene todo el estado de la aplicación. En Redux hay solo un store. Se puede acceder al state almacenado y actualizar el estado.

Para crear un store en una aplicación Redux, se utiliza la siguiente línea de código:

```
const store = createStore(LoginComponent);
```

Las actions realizadas en el estado siempre retornan un nuevo estado, lo que hace que el state sea fácil y predecible.

3.15 Bases de datos NoSQL

La utilización de bases de datos NoSQL se hace más y más presente en la arquitectura del desarrollo de aplicaciones y sistemas. Cuando recién aparecieron las bases de datos SQL estas fueron una revolución. Cumplen con su objetivo y lo hacen bien, pero la velocidad con la que avanzan las nuevas tecnologías ha hecho que las bases de datos SQL posean algunas limitaciones.

Esta es una de las razones que dio origen a las bases de datos NoSQL. La revolución de la informática en la sociedad y la interacción masiva entre usuarios que producen y almacenan contenido a cada segundo, exigen una forma más flexible de almacenar, ordenar y captar una gran cantidad de datos.

3.15.1 ¿Qué es NoSQL?

Estas siglas hacen referencia a Not Only SQL. Plantea una forma de hacer modelos de datos más flexibles que se adaptan rápidamente a los requisitos de aplicaciones modernas.

Por lo tanto, hablar de bases de datos NoSQL es hablar de estructuras que nos permiten almacenar información en aquellas situaciones en las que las bases de datos relacionales generan ciertos problemas debido principalmente a problemas de escalabilidad y rendimiento de las bases de datos relacionales donde se dan cita miles de usuarios concurrentes y con millones de consultas diarias. Además de lo comentado anteriormente, las bases de datos NoSQL son sistemas de almacenamiento de información que no cumplen con el esquema entidad-relación. Tampoco utilizan una

estructura de datos en forma de tabla donde se van almacenando los datos, sino que para el almacenamiento hacen uso de otros formatos como clave-valor, mapeo de columnas, documentales o grafos.

3.15.2 Tipos de bases de datos NoSQL

3.15.2.1 Bases de datos clave-valor

Son el modelo de base de datos NoSQL más popular, además de ser la más sencilla en cuanto a funcionalidad. En este tipo de sistema, cada elemento está identificado por una llave única, lo que permite la recuperación de la información de forma muy rápida, información que habitualmente está almacenada como un objeto binario (BLOB). Se caracterizan por ser muy eficientes tanto para las lecturas como para las escrituras. Algunos ejemplos de este tipo son Cassandra, BigTable o HBase.

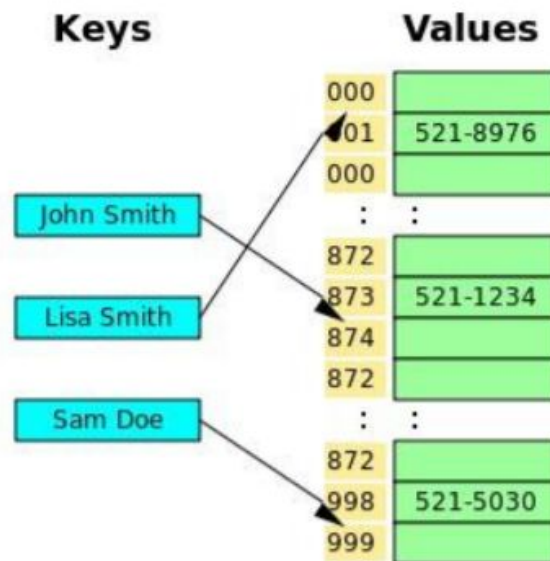


Figura 5 Base de datos clave-valor

3.15.2.2 Bases de datos documentales

Este tipo almacena la información como un documento, generalmente utilizando para ello una estructura simple como JSON o XML y donde se utiliza una clave única para cada registro. Este tipo de implementación permite, además de realizar búsquedas por clave-valor, realizar consultas más avanzadas sobre el contenido del documento. Son las bases de datos NoSQL más versátiles. Se pueden utilizar en gran cantidad de proyectos, incluyendo muchos que tradicionalmente funcionarían sobre bases de datos relacionales. Algunos ejemplos de este tipo son MongoDB o CouchDB.

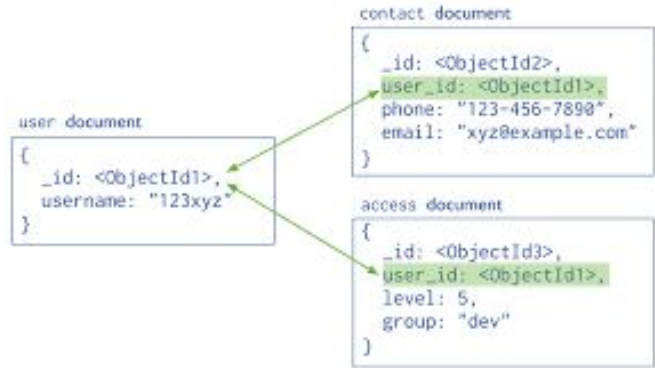


Figura 6 Bases de datos documentales

3.15.2.3 Bases de datos en grafo

En este tipo de bases de datos, la información se representa como nodos de un grafo y sus relaciones con las aristas de este, de manera que se puede hacer uso de la teoría de grafos para recorrerla. Para sacar el máximo rendimiento a este tipo de bases de datos, su estructura debe estar totalmente normalizada, de forma que cada tabla tenga una sola columna y cada relación dos. Este tipo de bases de datos ofrece una navegación más eficiente entre relaciones que en un modelo relacional. Algunos ejemplos de este tipo son Neo4j, InfoGrid o Virtuoso

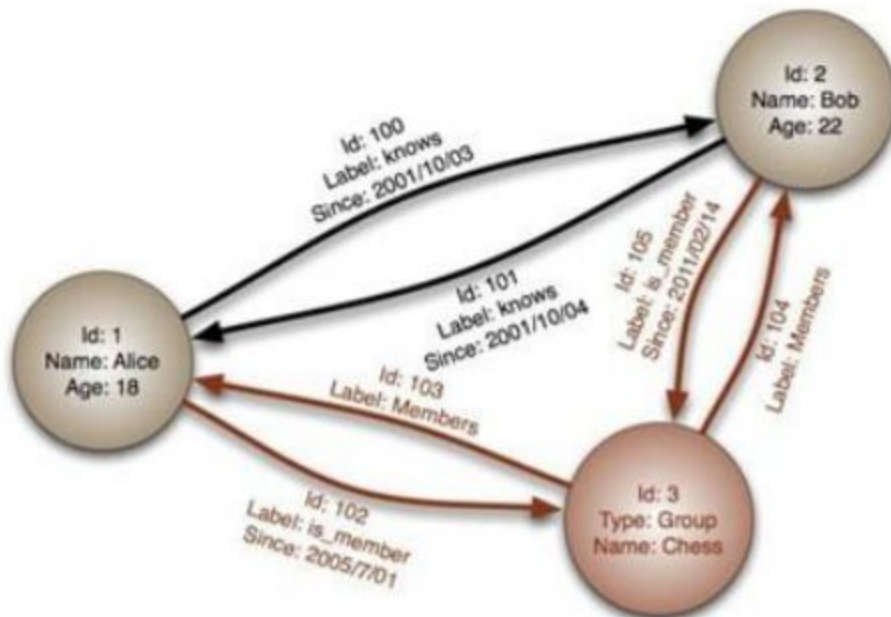


Figura 7 Bases de datos en grafo

3.15.2.4 Bases de datos orientada a objetos

En este tipo, la información se representa mediante objetos, de la misma forma que son representados en los lenguajes de programación orientada a objetos (POO) como ocurre en JAVA, C# o Visual Basic .NET. Algunos ejemplos de este tipo de bases de datos son Zope, Gemstone o Db4o.

3.16 MongoDB

MongoDB es una base de datos orientada a documentos. Esto quiere decir que, en lugar de guardar los datos en registros, guarda los datos en documentos. Estos documentos son almacenados en BSON, que es una representación binaria de JSON.

Una de las diferencias más importantes con respecto a las bases de datos relacionales, es que no es necesario seguir un esquema. Los documentos de una misma colección - concepto similar a una tabla de una base de datos relacional -, pueden tener esquemas diferentes.

3.17 JSON Web Tokens

Los JSON Web Tokens son una cadena de texto que tiene tres partes codificadas en Base64, cada una de ellas separadas por un punto, como la que vemos en la imagen siguiente:

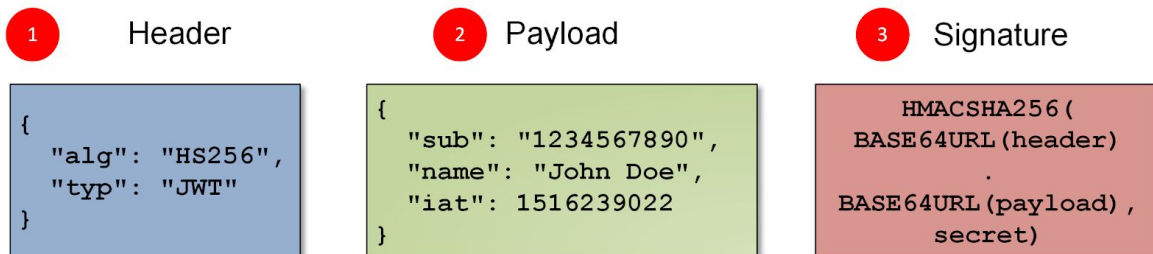
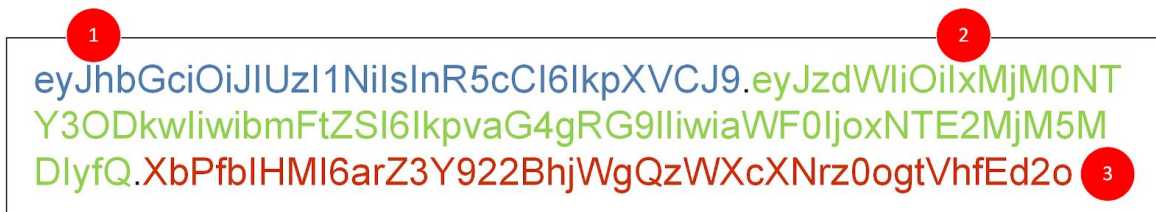


Figura 8 JSON Web Token

Como se ilustra en la imagen un JSON Web Token consta con tres partes.

- Header: encabezado dónde se indica, al menos, el algoritmo y el tipo de token, que en el caso del ejemplo anterior era el algoritmo HS256 y un token JWT.
- Payload: donde aparecen los datos de usuario y privilegios, así como toda la información que queramos añadir, todos los datos que creamos convenientes.
- Signature: una firma que permite verificar si el token es válido.

Los JSON Web Tokens son utilizados para autenticar a algún cliente al momento de hacer alguna petición al servidor. La información que tiene encriptada se utiliza para identificar quien hace la petición y si es un usuario válido, además mediante la firma se confirma la autenticidad de este token, lo que permite realizar verificaciones de autenticidad en las peticiones de una manera rápida y sencilla implementando este tipo de objetos.

3.18 Procesos de Reciclaje

En el proceso general del reciclaje, las principales entidades participantes son las empresas generadoras de residuos, las personas que generan residuos y/o llevan los residuos a centros de acopio de residuos para reciclar y las empresas valorizadoras. Es de suma importancia la participación de estas entidades y como interactúan entre sí.

Primero los hogares o personas, en conjunto con las grandes empresas que generan residuos, en un sistema centrado en el reciclaje, son llevados a puntos de reciclaje, para ser recogidos por quien sea que administre aquellos puntos y luego ser llevados a las empresas valorizadoras o centros procesadores, que se encargan de separar los residuos y seleccionar aquellos que se pueden reciclar de los que no, para volver a darles valor y estos puedan volver a entrar en el mercado.

En la siguiente figura se muestra un ejemplo de como sería el funcionamiento de un proceso administrado por un municipio.

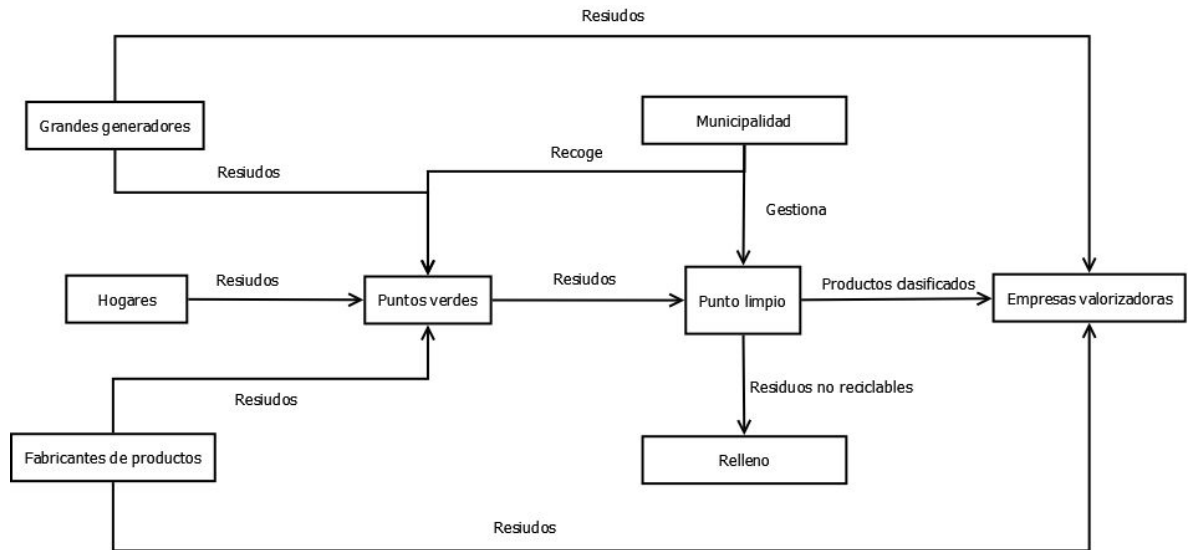


Figura 9 Ejemplo Modelo Reciclaje

Capítulo 4 Definición de un Modelo

Con todos los antecedentes mencionados en el capítulo 1 de este proyecto, a continuación, en la figura 9 se muestra la propuesta de un modelo para la Gestión del Reciclaje en la comuna de Concepción representado en un diagrama de flujo de datos, de aquí en adelante DFD.

Se decidió realizar un DFD porque permite visualizar y entender de manera rápida y sencilla como es llevado a cabo un proceso, sin deber tener mayores conocimientos previos para poder comprender como es el funcionamiento de un proceso modelado.

Las entidades que participan en esta gestión son:

- Municipalidad.
- Empleados.
- Puntos de Reciclaje.
- Vertedero.
- Mecánico Municipal.
- Centros Procesadores
- Camiones
- Usuarios

Municipalidad

Es la que se encarga de gestionar y controlar el proceso en general. Entrega los lineamientos y presupuestos para que el proceso pueda ser llevado a cabo. Así como también define los puntos de reciclaje, se encarga de la relación con los centros procesadores y entrega información a la comunidad.

Empleados

Los empleados son quienes se encargan de limpiar los puntos de reciclaje, conducir camiones, administrar los vertederos, entregar información a la comunidad, administrar ciertos procesos, etc.

Puntos de Reciclaje

Estos son los puntos donde se llevan los residuos para reciclar que se encuentran distribuidos por toda la comuna. Necesitan de requerimientos para poder funcionar y ser controlados periódicamente.

Vertedero

El vertedero es donde se llevan todos los residuos que no se pueden reciclar, así como también los residuos peligrosos. Estos deben cumplir con ciertos requerimientos entregados por la municipalidad e informar periódicamente el estado en el que se encuentran.

Mecánico Municipal

Es quien se encarga de reparar y hacer mantención a los camiones encargados de la recogida de los residuos.

Centros procesadores

Son los encargados de hacer que los residuos vuelvan a tener valor de mercado y hacerlos comercializables nuevamente, además de entregar un informe a la Municipalidad que contenga los materiales que pudieron y no pudieron ser reciclados.

Camiones

Son los encargados de realizar la recogida de los residuos tanto en los puntos de reciclaje como de los centros procesadores.

Usuarios

Son los que dan el motor a todo este proceso, llevan los residuos a los puntos de reciclajes y dan retroalimentación respecto al funcionamiento de los mismos.

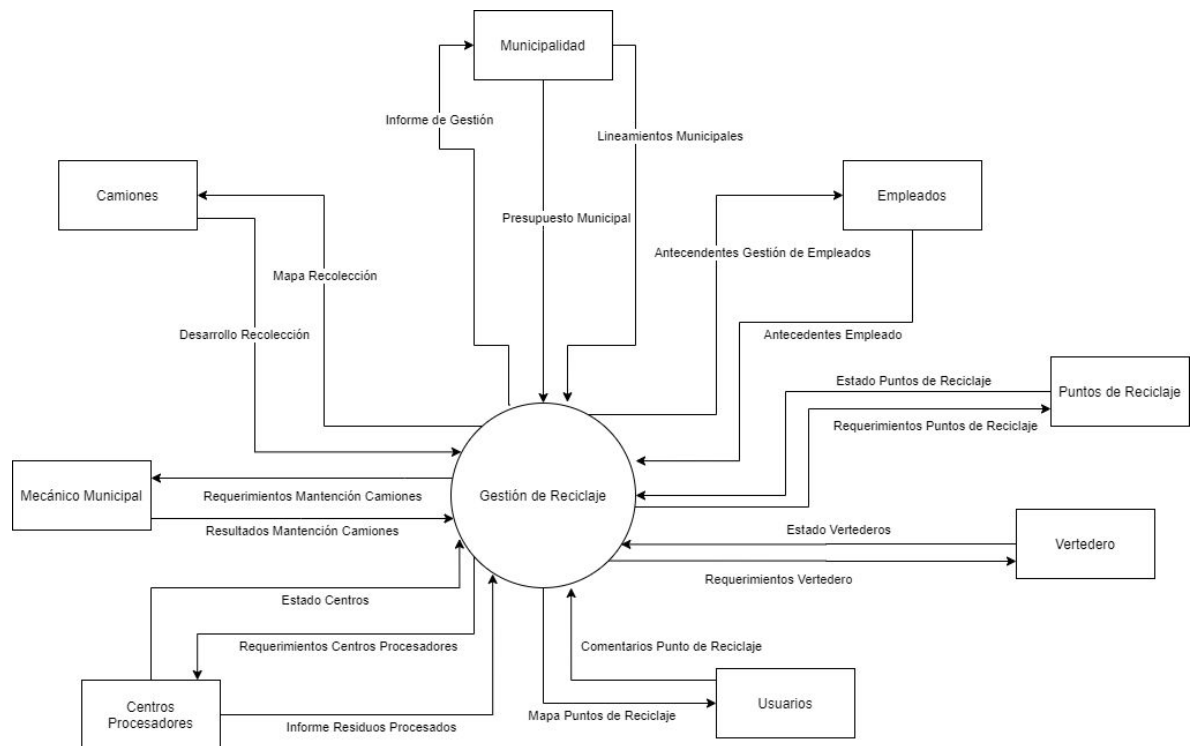


Figura 10 DFD Superior

Para tener un mejor entendimiento se procedió a modelar un DFD de contexto que permite entender de mejor manera como es que funciona el proceso. Este DFD se muestra en la figura 10.

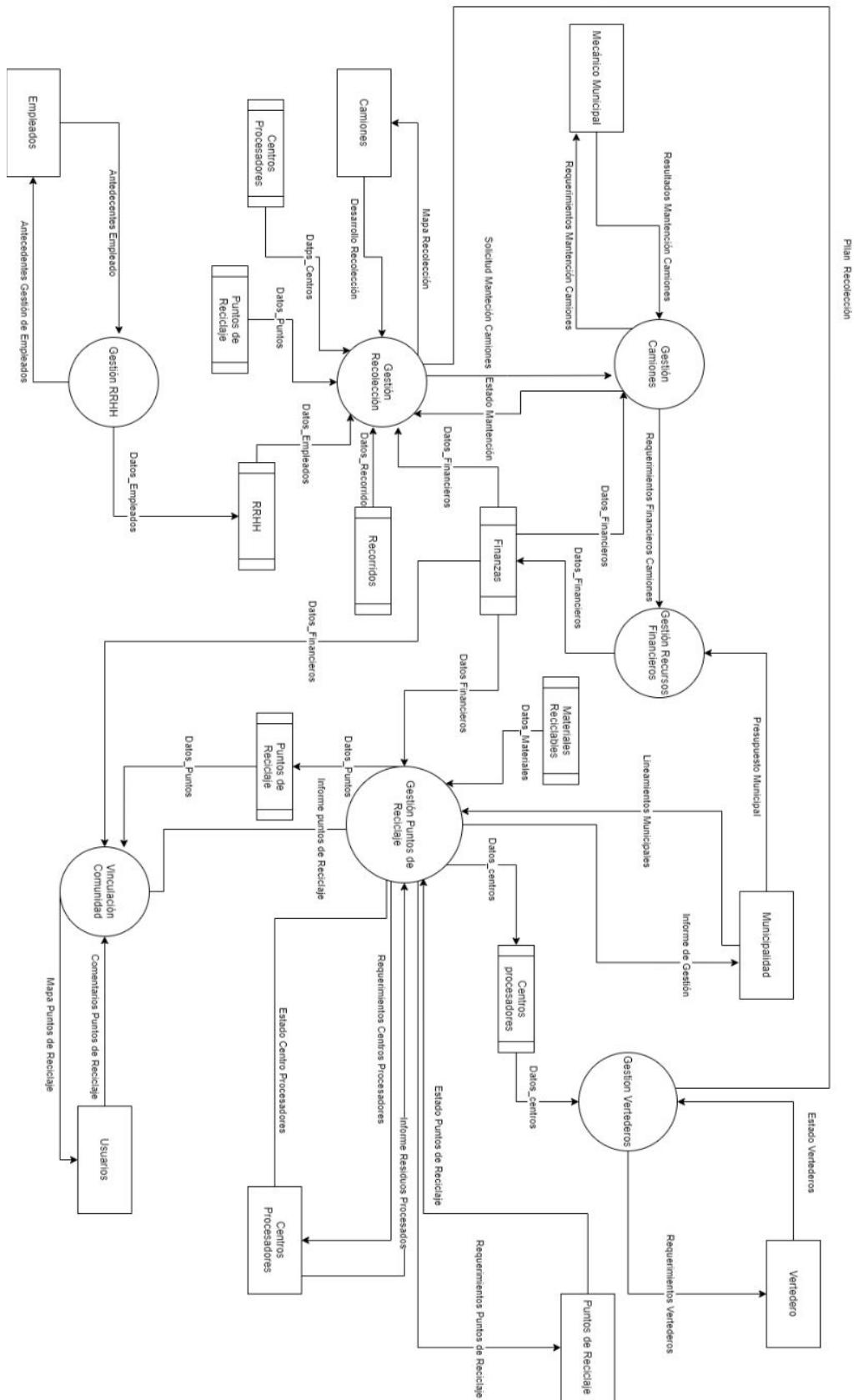


Figura 11 DFD de Contexto

Capítulo 5 Especificación de Requerimientos del Software

5.1 Alcances

- El software que se construirá será un prototipo, que será tan solo una parte de todo el sistema.
- El software permitirá al usuario ver los puntos de reciclaje en un mapa. Esto quiere decir que el usuario al entrar a la aplicación verá un mapa con todos los diferentes puntos de reciclaje en la comuna.
- El software permitirá seleccionar cada punto de reciclaje para ver información de cada uno de ellos. Información como qué tipo de material se recicla, puntuación del punto de reciclaje, comentarios, horario de atención, etc.
- El usuario podrá puntuar los diferentes puntos de reciclaje. Al seleccionar algún punto de reciclaje para poder ver su información, se le dará la opción al usuario de puntuar el punto de reciclaje y realizar un comentario.
- El software mostrará información sobre el reciclaje y sus procesos. Habrá una sección dentro de la aplicación donde se entregará información sobre que es el reciclaje y los diferentes procesos que conlleva el reciclaje, además del impacto que genera cada persona al reciclar.
- El software mostrará tips de reciclaje. La aplicación tendrá una sección en la que mostrará a los usuarios consejos para reciclar, como separar los residuos, porque reciclar, etc.

5.2 Objetivos de Software

5.2.1 Objetivo General

- Desarrollar un prototipo de aplicación que sirva de apoyo al proceso de Vinculación Comunidad descrito en el modelo propuesto. Permitirá tener un registro de las opiniones de la comunidad respecto al proceso de reciclaje en general y de los reciclajes realizados.

5.2.2 Objetivos específicos

- El sistema permite que los usuarios obtengan información en tiempo y real y de manera rápida de los diferentes puntos de reciclaje, de esta manera la municipalidad ahorra costos y tiempo en la entrega de información relacionada a los puntos de reciclaje.
- El sistema permite a los usuarios entregar opiniones respecto a la labor realizada en los puntos de reciclaje de una manera simple, permitiendo a la municipalidad ahorrar esfuerzos y eliminar la necesidad de realizar encuestas ciudadanas.
- El sistema entrega información respecto a la valoración que tienen los usuarios de los puntos de reciclaje, permitiendo a la municipalidad saber rápidamente en que lugares deben priorizar sus esfuerzos.
- El sistema permite a los usuarios una rápida forma de registro ofreciendo ingreso con redes sociales.
- El sistema permite a los usuarios llevar registro personal de reciclaje.

5.4 Requerimientos Específicos

5.4.1 Requerimientos Funcionales del Sistema

| ID | Nombre | Descripción |
|-------|-------------|---|
| RF_01 | Ingreso | El sistema permite un fácil acceso, ofreciendo a los usuarios distintas formas de autenticarse y registrarse. |
| RF_02 | Comentar | Permitir al usuario realizar comentarios en los diferentes puntos de reciclaje. |
| RF_03 | Perfiles | Ciertas funcionalidades del sistema deben estar restringidas según el tipo de usuario. |
| RF_04 | Mapa | Se debe permitir visualizar un mapa con todos los puntos de reciclaje. |
| RF_05 | Información | Brindar información al usuario respecto a diferentes temas relacionados con el reciclaje. |
| RF_06 | Reciclaje | Permitir al usuario llevar un registro de los reciclajes personales realizados. |

Tabla 1 Requerimientos Funcionales

5.4.2 Interfaces Externas de Entrada

| ID | Nombre | Detalle de Datos contenidos en ítem |
|--------|--------------------|--|
| IEE_01 | Autenticación | CORREO, CONTRASEÑA |
| IEE_02 | Registro | NOMBRE, APELLIDO, CORREO, CONTRASEÑA, CONFIRMAR_CONTRASEÑA |
| IEE_03 | Comentario | CALIFICACIÓN, DESCRIPCIÓN |
| IEE_04 | Nuevo Punto Limpio | LATITUD, LONGITUD, NOMBRE, DESCRIPCIÓN, MATERIALES, DIRECCIÓN, HORARIO |

5.4.2 Requerimientos no Funcionales

- El sistema debe asegurar que los datos estén protegidos del acceso no autorizado.
- La presentación de los datos debe ser simple para los usuarios.
- El sistema debe ser compatible tanto con Android como IOS.
- El sistema debe entregar mensajes de alertas o validaciones claros, haciendo énfasis en los datos ingresados por el usuario.
- El sistema debe ser diseñado de manera que sea escalable.

Capítulo 6 Factibilidad

6.1 Factibilidad Técnica

La factibilidad técnica se refiere a los recursos necesarios como herramientas, conocimientos, habilidades, experiencia, etc., que son necesarios para efectuar las actividades o procesos que requiere el proyecto. Generalmente se hace referencia a

elementos tangibles (medibles). El proyecto debe considerar si los recursos técnicos actuales son suficientes o deben complementarse.

Para desarrollar esta aplicación se cuentan con los conocimientos necesarios en las herramientas que se utilizarán, además de las herramientas necesarias para llevar a cabo esta aplicación, notebook, dispositivo móvil y acceso a internet. Esto permitiría realizar sin mayor problema los procesos en la herramienta de software que se desea implementar, haciendo de la solución propuesta, una solución técnicamente factible.

6.2 Factibilidad Operativa

Debido a que el proyecto desarrollado no es para una entidad en específico, sino que será utilizable por la comunidad, la factibilidad operativa se realiza en base a impacto esperado y a estimaciones.

Al momento de desarrollar la aplicación, será importante la usabilidad de la aplicación. Que sea sencilla, intuitiva y fácil de usar, ya que es una aplicación que será útil para la comunidad. El software será diseñado de tal manera que sea fácil de utilizar y amigable, que no requerirá de conocimientos de nivel alto por parte de los usuarios para usar la aplicación, solamente nivel básico de uso del celular y acceso a internet, algo hoy en día muy normalizado.

Además, la base de datos del sistema será construida de manera que sea escalable y fácil de modificar, esto para que en el caso que la aplicación se siga desarrollando, no sea compleja de entender o manipular.

Descritas ya las características que poseerá este software y quienes lo utilizarán, el software es factible operativamente.

6.3 Factibilidad Económica

Esta aplicación está pensada como una ayuda para la comunidad, por lo que no generará ingresos económicos, pero sí tendrá beneficios sociales de impacto positivo en la sociedad. Ayudará a la comunidad en los procesos de reciclaje, la ubicación de los puntos de reciclaje e información de cada uno de ellos. Además, servirá a la municipalidad de Concepción, ya que la aplicación generará estadísticas como puntos de reciclaje

mejores valorados, comentarios de la comunidad, etc. Lo que ayudará a la municipalidad en definir donde concentrar sus esfuerzos y recursos para mejorar la gestión del proceso de reciclaje en Concepción.

6.4 Conclusión Factibilidad

Habiendo analizado cada uno de los puntos que se evalúan al momento de definir si un proyecto es viable o no, se concluye que la implementación y desarrollo de la herramienta TIC propuesta es factible.

Capítulo 7 Análisis

7.1 Actores

Administrador

- **Rol o funciones dentro de la organización:** Es administrador, la persona que tiene conocimiento de cómo es el funcionamiento de los diferentes puntos de reciclaje y las características de estos.

- **Nivel de conocimientos técnicos requeridos:** Este actor requiere de conocimientos técnicos de nivel medio, debido a que debe manejar información específica dentro del sistema.
- **Nivel privilegio en el sistema y las funcionalidades del software a las cuales tiene acceso:** Este actor tiene permiso a todas las funcionalidades del software en su plenitud, puede utilizar todas las características de la aplicación.

Usuario

- **Rol o funciones dentro de la organización:** Es el usuario común, los ciudadanos que llevan sus residuos hasta los puntos de reciclaje y/o está interesado en saber el estado de los diferentes acopios de residuos para reciclar.
- **Nivel de conocimientos técnicos requeridos:** Este actor requiere de conocimientos técnicos de nivel bajo, solo debe saber utilizar un dispositivo móvil.
- **Nivel privilegio en el sistema y las funcionalidades del software a las cuales tiene acceso:** Este actor puede acceder a todas las funcionalidades del software, excepto por poder gestionar los puntos de reciclaje.

7.2 Diagramas de Casos de Uso

7.2.1 Jerarquía de usuarios

Jerarquía de usuarios



Figura 12 Jerarquía de actores

7.2.2 Casos de Usos Usuario



Figura 13 Casos de uso Usuario

7.2.3 Casos de Uso Administrador

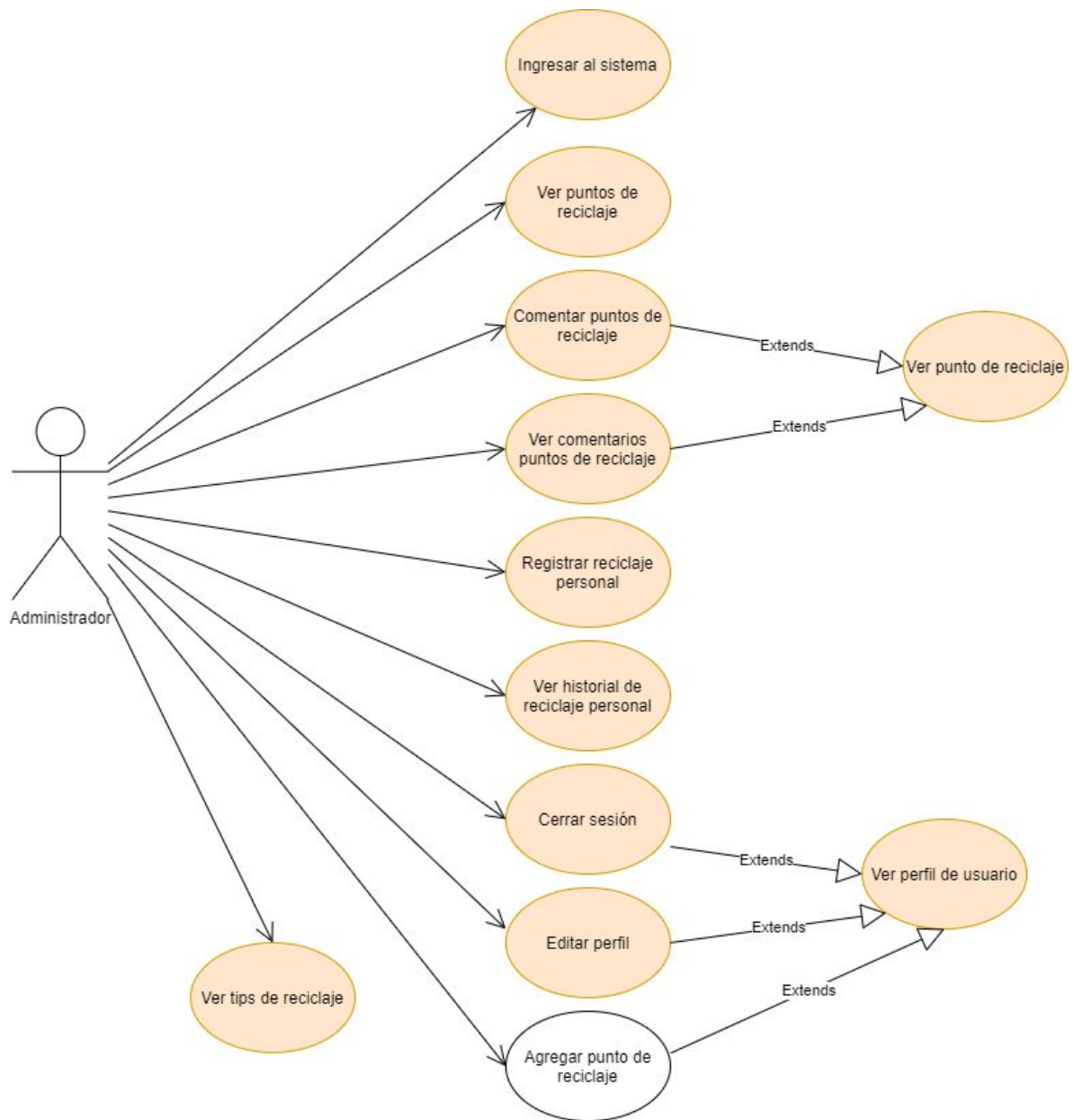


Figura 14 Casos de uso Administrador

Nota: El color naranja de los casos de uso se debe a que todos ellos también pueden ser realizados por el actor Usuario.

7.3 Especificación Casos de Uso

7.3.1 Caso de uso: Ingresar al sistema

- Descripción: El usuario podrá ingresar al sistema para hacer uso de este.
- Actores: Cualquier actor puede iniciar el evento.
- Precondiciones:
 - El sistema se encuentra disponible.
- Flujo de eventos básico:

Tabla 2 Caso de Uso: Ingresar al sistema

| Actor | Sistema |
|---|--|
| | 1. El sistema despliega un formulario con los campos "email" y "contraseña" |
| 2. El usuario completa el formulario con sus credenciales. | 3. El sistema verifica que los datos ingresados sean válidos |
| | 4. El sistema determina las operaciones permitidas por el perfil del usuario y las despliega donde corresponden. |
| | 5. El sistema redirige al usuario a la pantalla de inicio. |

- **Cursos Alternativos:**
 - 2b. Los datos ingresados por el usuario no son válidos. El sistema muestra en pantalla indicando la causa del error y se vuelve al paso 1.
- **Post condiciones:**
 - El usuario se encuentra autenticado para hacer uso del sistema.

7.3.2 Caso de Uso: Registrarse

- Descripción: El usuario podrá crear una cuenta para poder ingresar al sistema.
- Actores: Usuario.
- Precondiciones:

- El sistema se encuentra disponible.
- Flujo de eventos básico:

| Actor | Sistema |
|--|---|
| 1. Este caso de uso da inicio cuando un usuario desea registrarse en el sistema para poder hacer uso de él. | |
| | 2. El sistema despliega un formulario de registro con los campos nombre, apellido, correo, contraseña y edad. |
| 3. El usuario completa el formulario ingresando los campos correspondientes. | |
| | 4. El sistema valida los datos ingresados. |
| | 5. Sistema válida la existencia previa del correo introducido. |
| | 6. El sistema registra al usuario en el sistema. |
| | 7. El sistema redirige al usuario a la pantalla de inicio. |

Tabla 3 Caso de Uso: Registrarse

- **Cursos Alternativos:**
 - 4b. Los datos ingresados por el usuario no son válidos. El sistema muestra en pantalla un mensaje indicando la causa del error y el usuario es enviado al paso 2.
 - 5b. El sistema informa mediante un mensaje la existencia del correo ingresado y redirige al usuario al paso 2.
- **Post condiciones:**
 - El usuario se encuentra autenticado dentro del sistema.

7.3.3 Caso de Uso: Cerrar sesión

- Descripción: El usuario podrá salir del sistema.
- Actores: Usuario y Administrador.
- Precondiciones:
 - El sistema se encuentra disponible
 - El usuario está autenticado dentro del sistema.
- Flujo de eventos básico:

| Actor | Sistema |
|-------|---------|
|-------|---------|

| | |
|--|--|
| 1. Este caso de uso de inicio cuando un usuario quiere cerrar la sesión en la que se encuentra. | |
| | 2. El sistema llama al caso de uso 'Ver perfil de usuario' |
| | 3. El sistema muestra un botón que, al ser presionado, cierra la sesión del usuario. |
| 4. El usuario presiona el botón cerrar sesión. | |
| | 5. El sistema elimina los datos almacenados en el dispositivo del usuario actual. |
| | 6. El sistema redirige al usuario a la pantalla de inicio de sesión. |

Tabla 4 Caso de Uso: Cerrar Sesión

- **Post condiciones:**
 - El usuario se encuentra fuera del sistema.

7.3.4 Caso de Uso: Ver Puntos de Reciclaje

- Descripción: El usuario podrá visualizar en un mapa todos los puntos de reciclaje de la comuna.
- Actores: Usuario y Administrador.
- Precondiciones:
 - El usuario está autenticado dentro del sistema.
 - El sistema se encuentra disponible.
 - El usuario concedió los permisos al sistema para que pueda acceder a su ubicación.
- Flujo de Eventos Básico:

| Actor | Sistema |
|---|--|
| 1. Este caso de uso da inicio cuando el usuario presiona el mapa dentro de la aplicación. | |
| | 2. El sistema despliega un mapa con indicadores que representan los puntos de reciclaje. |
| 3. El usuario puede visualizar los diferentes puntos de reciclaje presentes en el mapa y seleccionarlos para ver su información. | |

Tabla 5 Caso de Uso: Ver Puntos de Reciclaje

- Post condiciones:
 - Usuario es capaz de ver un mapa con los puntos de reciclaje.

7.3.5 Caso de Uso: Comentar punto de reciclaje

- Descripción: El usuario puede comentar y calificar puntos de reciclaje.
- Actores: Usuario y Administrador.
- Precondiciones:
 - El usuario está autenticado dentro del sistema.
 - El usuario concedió acceso a la localización del dispositivo.
- Flujo de Eventos Básico:

| Usuario | Sistema |
|---|--|
| | 1. El sistema llama al caso de uso Ver Punto de Reciclaje |
| | 2. El sistema muestra un botón que permite comentar un punto de reciclaje en específico. |
| 3. El usuario presiona el botón comentar. | |
| | 4. El sistema despliega un formulario en donde se ingresa el cuerpo del comentario y permite seleccionar cantidad de estrellas para calificar. |
| 5. Usuario ingresa cuerpo del comentario y calificación. | |
| | 6. El sistema valida los campos y guarda el comentario. |
| | 7. Sistema redirige al usuario a la pantalla de punto de reciclaje en la que se encontraba. |

Tabla 6 Caso de Uso: Comentar Punto de Reciclaje

- Post condiciones:

- Punto de reciclaje posee un nuevo comentario en la base de datos y la calificación del punto es actualizada.

7.3.6 Caso de Uso: Ver Comentarios Punto de Reciclaje

- Descripción: Usuario visualiza una lista con los comentarios y calificaciones del punto de reciclaje.
- Actores: Usuario y Administrador.
- Precondiciones:
 - Usuario seleccionó un punto de reciclaje en el mapa.
 - El sistema se encuentra disponible.
- Flujo de Eventos Básico:

| Actor | Sistema |
|---|--|
| 1. Este caso de uso de inicio cuando el usuario selecciona un punto de reciclaje en específico en el mapa. | |
| | 2. El sistema llama al caso de uso Ver Punto de Reciclaje |
| | 3. Sistema despliega una lista con todos los comentarios y calificaciones realizados, así como la fecha en la que fueron realizados. |
| 4. El usuario es capaz de visualizar la lista de comentarios con su respectivo detalle. | |

Tabla 7 Caso de Uso: Ver Comentarios Punto de Reciclaje

- **Post condiciones:**
 - Usuario puede visualizar a través de la aplicación los diferentes comentarios realizados por otros usuarios.

7.3.7 Caso de Uso: Registrar Reciclaje Personal

- Descripción: Permite al usuario registrar los reciclajes que realiza de manera personal para poder llevar un registro de estos.
- Actores: Usuario y Administrador.
- Precondiciones:
 - El sistema se encuentra disponible.
 - Usuario se encuentra en la pantalla de reciclaje personal.
- Flujo de Eventos Básico:

| Actor | Sistema |
|--|---|
| 1. Este caso de uso da inicio cuando el usuario quiere hacer registro de un reciclaje. | |
| | 2. Sistema muestra una lista con los diferentes materiales que se pueden reciclar, así como también unos botones por cada uno de estos materiales que permite aumentar o disminuir la cantidad de material a reciclar, además un botón que permite registrar la cantidad de materiales seleccionados. |
| 3. El usuario selecciona la cantidad de materiales que va a reciclar de cada material y da al botón reciclar. | |
| | 4. El sistema almacena el reciclaje realizado por el usuario. |
| | 5. El sistema despliega un mensaje que confirma que el reciclaje se ha realizado con éxito. |
| | 6. Cantidades de materiales son reseteadas a 0. |

Tabla 8 Caso de Uso: Registrar Reciclaje Personal

- Post condiciones:
 - Es registrado un reciclaje nuevo en la base de datos perteneciente al usuario de la sesión actual.

7.3.8 Caso de Uso: Ver Historial de Reciclaje Personal

- Descripción: El usuario puede ver un registro de los reciclajes que ha realizado en todo el tiempo.

- Actores: Usuario y Administrador.
- Precondiciones:
 - El sistema se encuentra disponible.
 - Usuario se encuentra en la pantalla de reciclaje personal.
- Flujo de Eventos Básico:

| Actor | Sistema |
|---|---|
| 1. El caso de uso de inicio cuando el usuario desea ver el historial de sus reciclajes realizados. | |
| | 2. El sistema verifica que el usuario haya realizado por lo menos un reciclaje. |
| | 3. El sistema muestra un gráfico con las cantidades de cada material reciclado, cada material representado por un color diferente, además de una lista que muestra la fecha del reciclaje y los materiales que fueron reciclados. |

Tabla 9 Caso de Uso: Ver Historial de Reciclaje Personal

- Flujo de Eventos Alternativo:
 - 2b. El usuario no ha realizado ningún reciclaje y es enviado a la pantalla que le permite realizar reciclajes.
- Post condiciones:
 - Usuario es capaz de visualizar información respecto a su reciclaje.

7.3.9 Caso de Uso: Editar Perfil

- Descripción: El usuario puede realizar cambios en su información personal, correo, nombre y edad.
- Actores: Usuario y Administrador.
- Precondiciones:
 - El sistema está disponible.
 - Usuario se encuentra en la pantalla de perfil de usuario.
- Flujo de Eventos Básico:

| Actor | Sistema |
|-------|---------|
|-------|---------|

| | |
|--|---|
| 1. Este caso de uso da inicio cuando el usuario desea actualizar su información personal. | |
| | 2. El sistema llama al caso de uso Ver Perfil de Usuario. |
| | 3. Sistema despliega formulario con la información del usuario, nombre, correo y edad. |
| 4. Usuario completa los campos que desee y presiona el botón actualizar información. | |
| | 5. Sistema valida los datos. |
| | 6. Sistema actualiza información del usuario. |
| | 7. El sistema entrega mensaje de que se ha realizado con éxito la actualización de los datos. |

Tabla 10 Caso de Uso: Editar Perfil

- Flujo de Eventos Alternativo:
 - 4b. Datos ingresados no son válidos y se muestra un mensaje con el error al usuario.

7.3.10 Caso de Uso: Ver Perfil de Usuario

- Descripción: El usuario es capaz de ver su información personal.
- Actores: Usuario y Administrador.
- Precondiciones:
 - El sistema está disponible.
 - El usuario seleccionó la pantalla del Perfil de Usuario.
- Flujo de Eventos Básico:

| Actor | Sistema |
|--|---|
| 1. El caso de uso da inicio cuando el usuario desea ver la información personal de su cuenta. | |
| | 2. El sistema muestra información personal del usuario. Nombre, correo, edad y botón para actualizar información. |

Tabla 11 Caso de Uso: Ver Perfil de Usuario

- Postcondiciones:

- El usuario es capaz de visualizar su información personal.

7.3.11 Caso de Uso: Ver Punto de Reciclaje

- Descripción: El usuario puede visualizar información detallada de un punto de reciclaje.
- Actores: Usuario y Administrador.
- Precondiciones:
 - Usuario seleccionó un punto de reciclaje en el mapa.
 - El sistema está disponible.
- Flujo de Eventos Básico:

| Actor | Sistema |
|--|--|
| 1. El caso de uso da inicio cuando el usuario desea ver información de un punto de reciclaje en específico. | |
| | 2. El sistema despliega información referente al punto de reciclaje seleccionado. Calificación, comentarios, imágenes del punto de reciclaje, horario de funcionamiento, materiales que recibe, dirección y una descripción. |

Tabla 12 Caso de Uso: Ver Punto de Reciclaje

7.3.12 Caso de Uso: Ver Tips de Reciclaje

- Descripción: El usuario puede visualizar diferentes tips de reciclaje e información de como reciclar los diferentes materiales.
- Actores: Usuario y Administrador.
- Precondiciones:
 - El sistema se encuentra disponible.
 - El usuario seleccionó la pantalla de información de reciclaje.
- Flujo de Eventos Básico.

| Actor | Sistema |
|---|---|
| 1. El caso de uso de inicio cuando el usuario quiere ver información relacionada con el reciclaje. | |
| | 2. El sistema despliega una lista con diferentes tips e información referente al reciclaje. |

Tabla 13 Caso de Uso: Ver Tips de Reciclaje

- Postcondiciones:
 - El usuario puede visualizar tips e información de reciclaje.

7.3.13 Caso de Uso: Agregar Punto de Reciclaje

- Descripción: El usuario puede agregar nuevos puntos de reciclaje que serán visibles en el mapa.
- Actores: Administrador.
- Precondiciones:
 - El sistema se encuentra disponible.
 - Usuario se encuentra autenticado como Administrador.
- Flujo de Eventos Básico:

| Actor | Sistema |
|---|---|
| 1. Este caso de uso da inicio cuando el usuario desea agregar un nuevo punto de reciclaje. | |
| | 2. El sistema llama al caso de uso Ver Perfil de Usuario. |
| | 3. El sistema despliega un formulario en donde el usuario debe ingresar nombre, descripción, dirección, horario, materiales que se reciclan y seleccionar un punto en el mapa que se despliega en pantalla. |
| 4. Usuario completa los campos y presiona botón guardar. | |
| | 5. Sistema valida datos ingresados. |
| | 6. El sistema guarda en su base de datos un nuevo punto de reciclaje. |
| | 7. El sistema devuelve al usuario a la pantalla de perfil de usuario. |

Tabla 14 Caso de Uso: Agregar Punto de Reciclaje

Capítulo 8 Diseño

8.1 Diseño Físico de la Base de Datos

La Figura 11 a continuación, representa los documentos que serán utilizados como base de datos. Es importante recalcar que son colecciones de documentos y no tablas, ya que es una base de datos no relacional, en este caso, MongoDB.

Tampoco hay relaciones entre las tablas, solo campos dentro de cada documento que hace referencia a otro documento dentro de la base de datos, lo que permite imitar el funcionamiento de las bases de datos relacionales con clave foráneas.

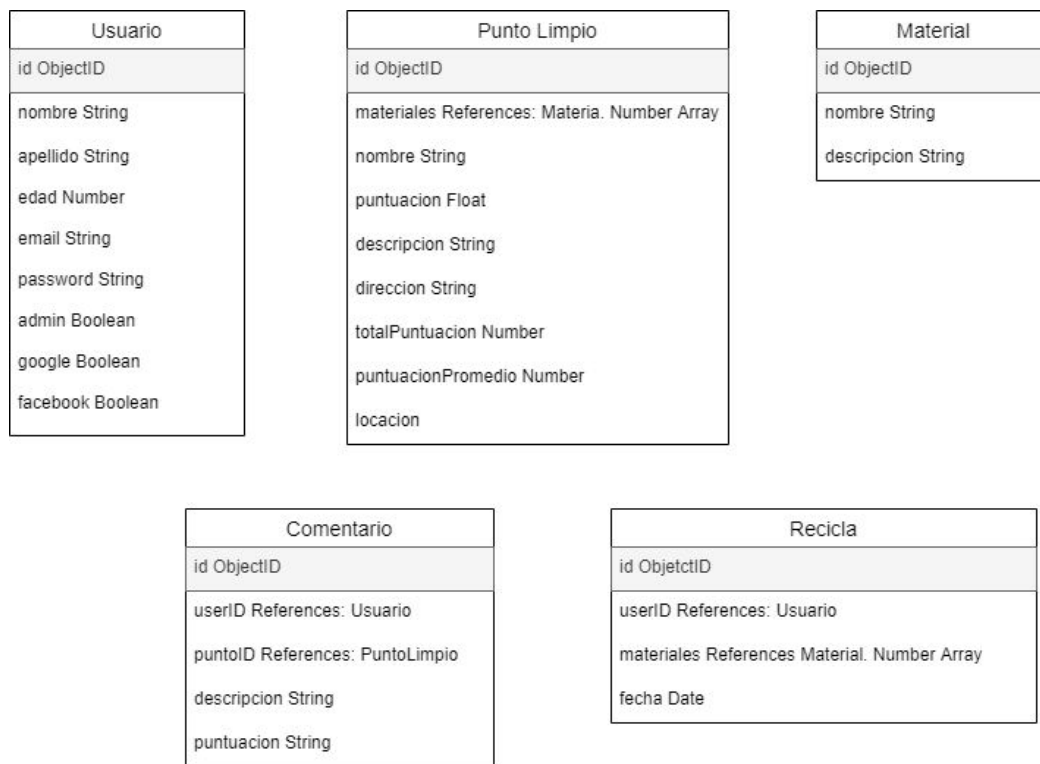


Figura 15 Modelo Base de Datos

8.1.1 Descripción de Documento: Usuario

El documento Usuario, tiene como objetivo almacenar todos los usuarios del sistema.

| Contenido | Tipo de Dato | Descripción |
|-----------|--------------|-------------|
|-----------|--------------|-------------|

| | | |
|-----------------|----------|--|
| id | ObjectID | Identificador Usuario. |
| nombre | String | Nombre del usuario. |
| apellido | String | Apellido del usuario. |
| edad | Number | Edad del usuario. |
| email | String | Correo electrónico del usuario. |
| password | String | Contraseña del usuario. |
| admin | Boolean | Corresponde a si el usuario es o no un administrador. |
| google | Boolean | Corresponde a si el usuario se registró o no con una cuenta de Google. |
| facebook | Boolean | Corresponde a si el usuario se registró o no con una cuenta de Facebook. |

Tabla 15 Descripción Documento: Usuario

8.1.2 Descripción de Documento: Material

El documento material se utiliza para almacenar todos los tipos de materiales que se puedan reciclar.

| Contenido | Tipo de Dato | Descripción |
|--------------------|--------------|--|
| id | ObjectID | Identificador del material reciclable. |
| nombre | String | Nombre del material reciclable. |
| descripcion | String | Descripción del material reciclable. |

Tabla 16 Descripción Documento: Material

8.1.3 Descripción de Documento: Comentario

El documento de comentarios almacena los comentarios que los usuarios realizan en los diferentes puntos limpios. Tiene referencias con el documento usuario y punto limpio.

| Contenido | Tipo de Dato | Descripción |
|--------------------|--------------|--|
| id | ObjectID | Identificador del comentario. |
| userID | ObjectID | Referencia al usuario que realizó el comentario. |
| puntoID | ObjectID | Referencia al punto limpio en el cual fue realizado el comentario. |
| descripcion | String | Cuerpo del comentario realizado. |
| puntuacion | Number | Calificación referente al punto limpio que se comenta. |

Tabla 17 Descripción Documento: Comentario

8.1.4 Descripción de Documento: Punto Limpio

La colección de puntos limpios se utilizará para almacenar los diferentes puntos limpios que haya. Se almacena un arreglo de documentos de materiales que hacen referencia a la colección de materiales.

| Contenido | Tipo de Dato | Descripción |
|---------------------------|-----------------|--|
| id | ObjectID | Identificador del punto limpio. |
| materiales | Array ObjectID | Arreglo con todos los id que hacen referencia a el documento material, que se pueden reciclar. |
| nombre | String | Nombre del punto limpio. |
| puntuacion | Number | Suma de toda las puntuaciones realizadas en el punto limpio. |
| descripcion | String | Correo electrónico del usuario. |
| direccion | String | Contraseña del usuario. |
| totalPuntuaciones | Number | Cantidad de puntuaciones realizadas al punto limpio. |
| puntuacionPromedio | Number | Puntuación promedio del punto limpio. |
| location | Location Object | Objeto que contiene la latitud y longitud en la que se encuentra emplazado el punto limpio. |

Tabla 18 Descripción Documento: Punto Limpio

8.1.5 Descripción de Documento: Recicla

El documento recicla se utiliza para guardar un registro personal de reciclaje que realicen los usuarios.

| Contenido | Tipo de Dato | Descripción |
|-------------------|----------------|--|
| id | ObjectID | Identificador del reciclaje. |
| userID | ObjectID | Referencia al documento usuario que realiza el reciclaje. |
| materiales | Array ObjectID | Arreglo de referencia y cantidad de materiales reciclados por el usuario, al documento material. |
| fecha | String | Fecha en la que fue realizado el reciclaje. |

Tabla 19 Descripción Documento: Recicla

8.2 Diseño de Arquitectura Funcional

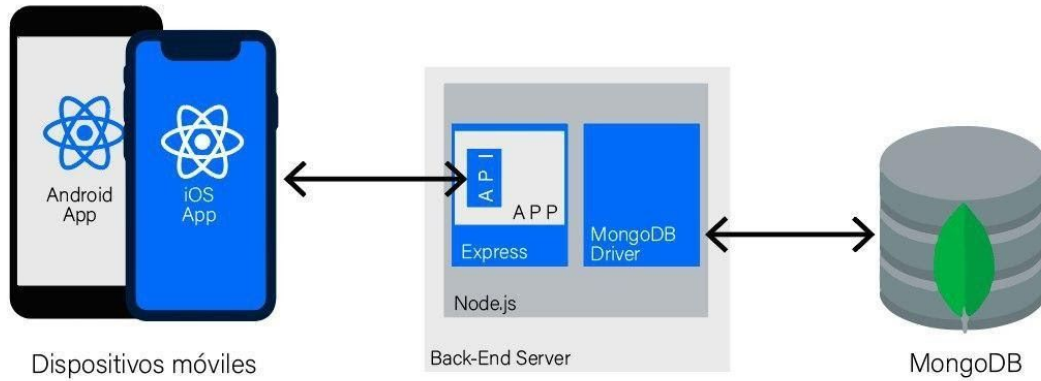


Figura 16 Arquitectura Funcional

La arquitectura de la aplicación está construida en base a como son construidas hoy en día las aplicaciones móviles modernas.

Cuenta con una base de datos NoSQL que es administrada por NodeJS y a la cual se puede acceder y manipular mediante la API REST construida.

Esta API REST a su vez es consumida por el cliente final que serán los dispositivos móviles, permitiendo interactuar a la aplicación con la base de datos.

8.3 Diseño API REST

API REST es un acrónimo en inglés para representational state transfer (transferencia de estado representacional). Es un estilo de arquitectura para sistemas distribuidos de hipertexto. Es una interfaz de aplicación de programa que utiliza peticiones HTTP para obtener (GET), insertar (POST), actualizar (PUT) y eliminar (DELETE).

Este tipo de arquitectura permite la comunicación entre dos o más softwares sin importar el lenguaje o forma en la que estén construidos, es creada en el lado del servidor para que múltiples clientes puedan interactuar con él de manera directa.

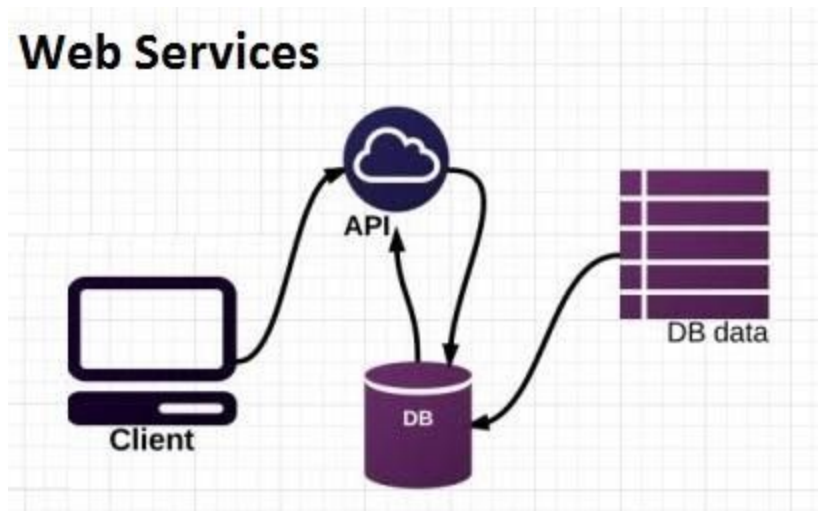


Figura 17 Diseño Api REST

8.3.1 Principios de API REST

- **Ciente-servidor:** Separando la interfaz de usuario de la lógica del almacenamiento de datos, se mejora la portabilidad de la interfaz del usuario para múltiples plataformas y mejorar la escalabilidad simplificando los componentes del servidor.
- **Sin estado:** Cada petición del cliente al servidor debe contener toda la información necesaria para procesar la petición, y no puede aprovechar de ningún tipo de información guardada en el servidor. Por lo tanto, el estado de la sesión, por ejemplo, se mantiene completamente en el cliente.
- **Almacenable en caché:** Las restricciones de caché requieren que los datos dentro de una respuesta a una petición estén etiquetados implícita o explícitamente como almacenables en caché o no. Si una respuesta se puede almacenar en caché, la caché de un cliente tiene derecho a reutilizar esos datos de respuesta para solicitudes posteriores equivalentes.
- **Sistema de capas:** El estilo del sistema de capas permite que una arquitectura se componga de capas jerárquicas al restringir el comportamiento de los componentes de manera que cada componente no pueda “ver” más allá de la capa inmediata a la que están interactuando.

También es importante saber de qué están compuestas las peticiones que se hacen al servidor. Principalmente poseen cuatro cosas:

1. La ruta raíz (en inglés endpoint y se llamara así de aquí en lo que resta de informe).
2. El método.
3. Las cabeceras.
4. Los datos (o body).

Endpoint

El endpoint, es la URL a la que se hacen peticiones. Siguen la siguiente estructura:

`https://sitio.com/endpoint`

El endpoint es el punto de entrada de la API a la que se hacen peticiones. Por ejemplo, el endpoint de la API de Github es `https://api.github.com`.

Otro punto importante es el camino (path en inglés y se llamará así de ahora en adelante). El path determina el recurso o dato que se está solicitando en una petición. Se puede acceder a los paths de igual forma en como se acceden a ciertas paginas en sitios web. Por ejemplo, para obtener el perfil de un usuario de Facebook se navega a la página:

`https://www.facebook.com/nombre-usuario/`

Para saber que paths están disponibles para los usuarios, los sitios cuentan con documentación de su API.

Generalmente los paths siguen el siguiente formato

`api/users/:id`

El path de arriba permite, por ejemplo, obtener la información de un usuario que tenga la id ingresada por la URL, donde “:id” debe ser reemplazada por una id existente en el sistema. Además de esto se debe indicar el método con el que se realizará una petición, definido en la siguiente sección.

Método

Los métodos hacen referencia a las acciones que se pueden realizar a través de una API, los comunes son GET, POST, PUT y DELETE.

| Nombre Método | Significado Petición |
|---------------|----------------------|
|---------------|----------------------|

| | |
|--------------------|---|
| GET | Esta petición es usada para obtener un recurso desde el servidor. Si se realiza una petición GET, el servidor busca la información que se solicitó y la devuelve. En otras palabras, la petición GET realiza una operación de lectura. Este es el método por defecto. |
| POST | Esta petición es usada para crear un nuevo recurso en el servidor. Si se realiza una petición POST, el servidor crea una nueva entrada en la base de datos y responde si la acción fue exitosa o no. En otras palabras, una petición POST realiza la operación de crear. |
| PUT y PATCH | Estas dos peticiones son utilizadas para actualizar recursos en el servidor. Si se realiza una petición PUT o PATCH, el servidor actualiza una entrada en la base de datos y responde si fue realizada de manera exitosa o no. En otras palabras, las peticiones PUT o PATCH, realizan operaciones de actualizar. |
| DELETE | Esta petición elimina un recurso del servidor. Si se realiza una petición DELETE, el servidor elimina la entrada que se le dice de la base de datos y responde si la acción fue realizada de manera exitosa o no. |

Tabla 20 Métodos HTTP

Cabeceras

También es importante mencionar las cabeceras. Estas sirven para proveer de información tanto al cliente como al servidor. Pueden ser utilizadas para muchos propósitos, como autenticación y proveer información del contenido de una petición.

Las cabeceras HTTP son una pareja propiedad-valor, separadas por dos puntos. A continuación, se muestra un ejemplo de una cabecera HTTP:

"Content-Type: application/json"

Esta cabecera indica que el contenido de la petición realizada a la API es de formato JSON.

Datos o Body

Los datos (body de aquí en adelante) contienen información que se quiere enviar al servidor. Esta opción solo es utilizada con peticiones POST, PUT y DELETE.

En la figura 16 y 17 a continuación se muestra un ejemplo real de una petición POST, mostrando como se utilizan cada uno de los conceptos mencionados para realizar una petición a la API.

En este ejemplo se estaría ingresando una notificación a la base de datos, y el cuerpo de esta, está ilustrado en la figura 16.

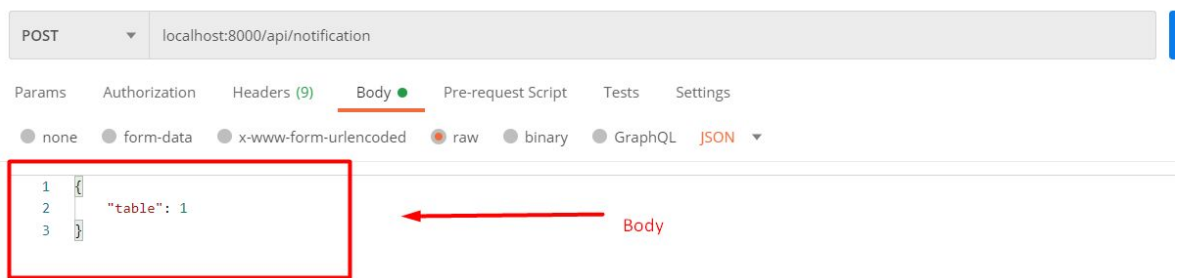


Figura 18 API Request 1

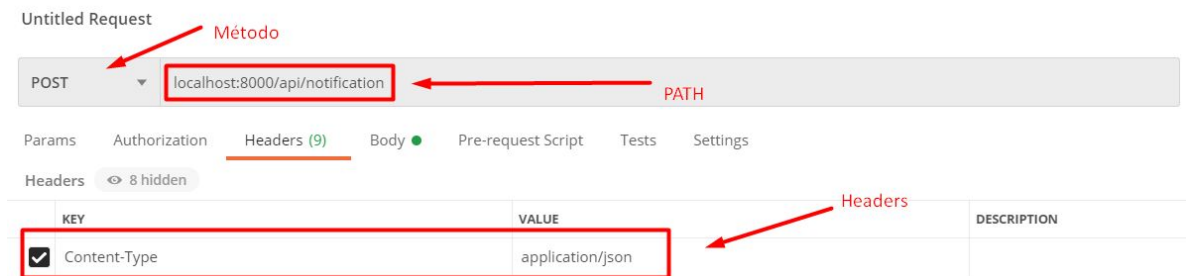


Figura 19 API Request 2

Cuando una petición es realizada, la API responde con un código de respuesta, este es un estándar HTTP que indica códigos de estado y mensajes de error.

- Código 200+ significa que la petición se ha realizado con éxito.
- Código 300+ significa que la petición ha sido redirigida a otra URL.
- Código 400+ significa que ha ocurrido un error en la petición del lado del cliente.

- Código 500+ significa que ha ocurrido un error en la petición del lado de servidor.

JSON

JSON es el acrónimo en inglés para JavaScript Object Notation (notación de objeto de Javascript) es un formato de texto que permite el intercambio de datos de manera sencilla. Se origino como un tipo de objeto propio del lenguaje JavaScript, pero se volvió tan popular que se volvió un formato independiente del lenguaje.

Estos tipos de objetos son los más utilizados en peticiones HTTP a las API, ya que tienen una amplia versatilidad que permite que sean fáciles de manipular y procesar.

En la figura número 18 se muestra un ejemplo de un objeto JSON.

```

{
  "0": {
    "id": 4,
    "name": "abhishek",
    "user_id": 6,
    "created_at": "2015-06-04 23:33:59",
    "updated_at": "2015-06-02 23:33:59",
    "users": {
      "id": 6,
      "name": "abhishek",
      "last_name": "deshkar",
      "email": "abhi.alone@ymail.com",
      "cell_no": "9979559210",
      "gender": "male",
      "country": 91,
      "primary_nick": "abhishek",
      "created_at": "2015-06-04 23:33:59",
      "updated_at": "2015-06-15 00:02:40"
    }
  }
}

```

Figura 20 Objeto JSON

8.3.2 API REST Construida

En el presente apartado se especificará cada uno de los endpoints de la API, especificando sus métodos y el contenido que debe tener la petición.

La simbología utilizada {} hace referencia a un objeto tipo JSON, si el objeto aparece de la siguiente forma {} significa que el body debe ir vacío.

Endpoint para la colección de comentarios

| Endpoint | Método | Body | Cabecera |
|-----------------|--------|---|---|
| /comentario | GET | {} | Accept:application/json |
| /comentario | POST | { userId: ObjectID, puntold: ObjectID, userName: String, userLastName: String, descripción: String, puntuación: Number, fecha: Date } | Content-type:application/json Authorization: {token Usuario} |
| /comentario/:id | PUT | { userName: String, userLastName: String, descripcion: String, puntuacion: Number, fecha: Date } | Content-type:application/json |
| /comentario/:id | DELETE | {} | Content-type:application/json |

Tabla 21 Endpoint Comentario

Endpoint para la colección de materiales

| Endpoint | Método | Body | Cabecera |
|---------------|--------|--|---|
| /material | GET | {} | Accept:application/json |
| /material | POST | { nombre: String, descripcion: String } | Content-type:application/json Authorization: {token Usuario} |
| /material/:id | PUT | { nombre: String, descripción: String } | Content-type:application/json |
| /material/:id | DELETE | {} | Content-type:application/json |

Tabla 22 Endpoint Material

Endpoint para la colección de puntos limpios

| Endpoint | Método | Body | Cabecera |
|------------------|--------|--|---|
| /puntolimpio | GET | {} | Accept:application/json |
| /puntolimpio | POST | { nombre: String, puntuacin: Number, descripcion: String, direccion: String, materiales:ObjectID Array, location: Location Object } | Content-type:application/json Authorization: {token Usuario} |
| /puntolimpio/:id | PUT | { Nombre: String, puntuacion: Number, descripcion: String, direccion: String, materiales: ObjectID Array, location: Location Object } | Content-type:application/json |
| /puntolimpio/:id | DELETE | {} | Content-type:application/json |

Tabla 23 Endpoint Punto Limpio

Endpoint para la colección de reciclajes personales

| Endpoint | Método | Body | Cabecera |
|--------------|--------|--|---|
| /recicla | GET | {} | Accept:application/json |
| /recicla | POST | { userId: ObjectID, materiales: ObjectID Array, fecha: Date } | Content-type:application/json Authorization: {token Usuario} |
| recicla/:id | PUT | { materiales: ObjectID Array, fecha: Date } | Content-type:application/json |
| /recicla/:id | DELETE | {} | Content-type:application/json |

Tabla 24 Endpoint Recicla

Endpoint para la colección de usuarios

| Endpoint | Método | Body | Cabecera |
|--------------|--------|--|-------------------------------|
| /usuario | GET | {} | Accept:application/json |
| /usuario | POST | { nombre: String, apellido: String, email: String, password: String, edad: Number, admin: Boolean, Google: Boolean, Facebook: Boolean } | Content-type:application/json |
| /usuario/:id | PUT | { nombre: String, apellido: String, email: String, password: String, edad: Number, admin: Boolean, } | Content-type:application/json |
| /usuario/:id | DELETE | {} | Content-type:application/json |

Tabla 25 Endpoint Usuario

Endpoint para el registro de usuarios

| Endpoint | Método | Body | Cabecera |
|-----------|--------|--|-------------------------------|
| /register | POST | { nombre: String, apellido: String, email: String, password: String, edad: Number, admin: Boolean } | Content-type:application/json |

Tabla 26 Endpoint Register

Endpoint para la autenticación de usuarios

| Endpoint | Método | Body | Cabecera |
|---------------|--------|------|-------------------------------|
| /login | POST | {} | Accept:application/json |
| /login/google | POST | { | Content-type:application/json |

| | | | |
|-----------------|------|--|-------------------------------|
| | | email: String, password: String } | |
| /login/facebook | POST | { email: string, password: String } | Content-type:application/json |

Tabla 27 Endpoint Login

8.3 Diseño de Interfaz y Navegación

En este capítulo se mostrarán las diferentes pantallas de la aplicación y las diferentes acciones que se pueden realizar en cada una de ellas, además de ilustrar como está diseñada la navegación de esta para ser utilizada por los usuarios.

8.3.1 Navegación

La imagen de la figura 12 muestra el diseño de navegación de la aplicación.

En las aplicaciones desarrolladas en React Native, la navegación se realiza creando contenedores, los cuales poseen las diferentes pantallas y formas de navegación. Los navegadores en la figura 12 hacen referencia a estos contenedores.

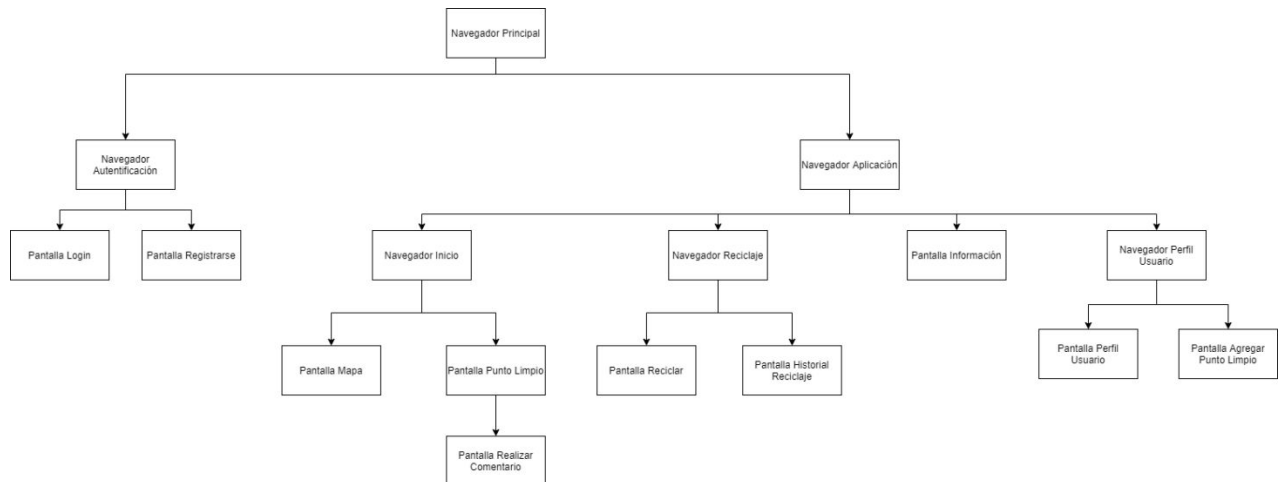


Figura 21 Mapa Navegación App

8.3.2 Pantalla Login

Esta es la primera pantalla que se despliega al abrir la aplicación. Aquí el usuario debe ingresar sus credenciales para poder ingresar a la aplicación y luego presionar el botón iniciar sesión. De ser exitosa la autenticación, será redireccionado a la pantalla de Inicio ilustrada en la figura 22.

También está la opción de que el usuario ingrese con su cuenta de Facebook o Google si así lo desea.

Si el usuario no posee una cuenta, debe presionar el texto Registrarme en la parte inferior de la pantalla, que lo llevará a la pantalla de Registro ilustrada en la Figura 23.

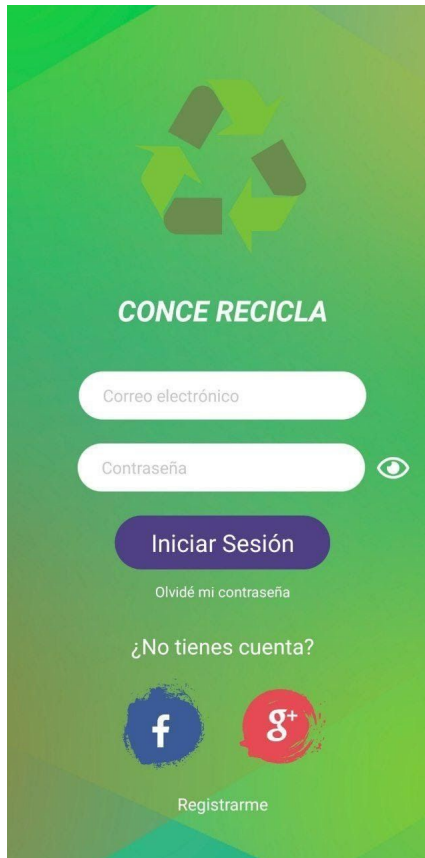
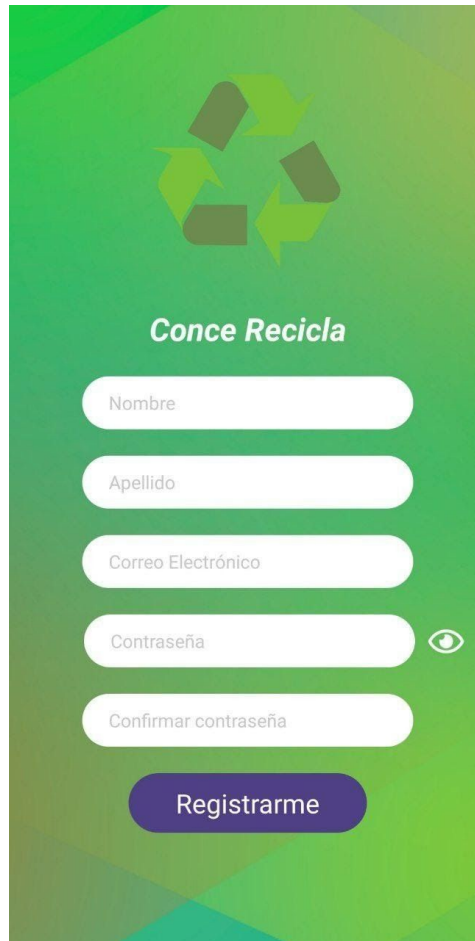


Figura 22 Pantalla de Login

8.3.3 Pantalla Registro

En esta pantalla el usuario puede crear una nueva cuenta para ingresar a la aplicación. Luego de llenar los campos debe presionar el botón Registrarme para poder validar los campos y así crear la cuenta. De tener éxito el usuario será derivado a la pantalla de Inicio de la aplicación que se muestra en la Figura 24.




Conce Recicla

Nombre

Apellido

Correo Electrónico

Contraseña 

Confirmar contraseña

Registrarme

Figura 23 Pantalla Registro

8.3.4 Pantalla Inicio

Esta es la pantalla de bienvenida luego que el usuario se autentifica. Aquí se muestran los puntos limpios en el mapa pertenecientes a la comuna de Concepción. Aquí el usuario podrá mover el mapa para ver donde se encuentran los diferentes puntos limpios. Cada punto rojo representa un punto limpio, estos al ser presionados abrirán una nueva pantalla, ilustrada en la figura 25.

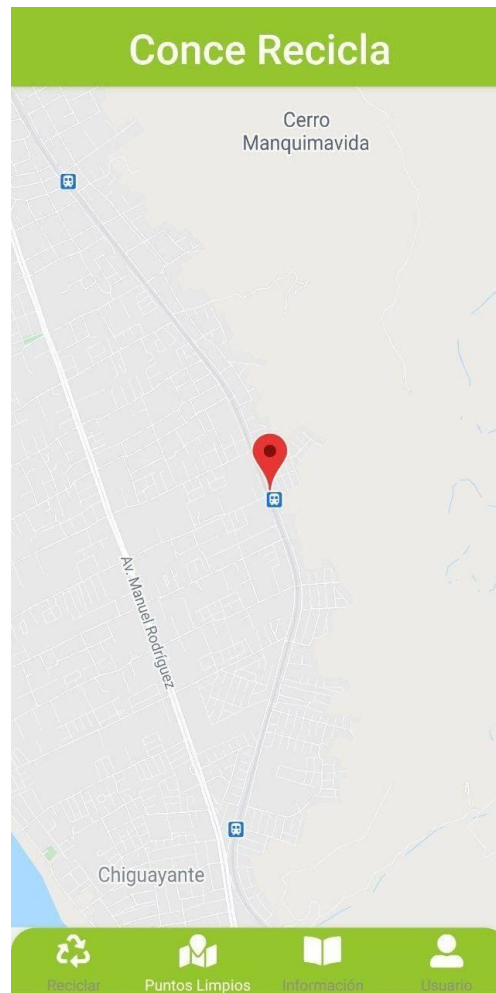


Figura 24 Pantalla Inicio

8.3.5 Pantalla Punto Limpio

En esta pantalla, los usuarios podrán ver toda la información correspondiente al punto limpio, así como también una calificación y una lista de los comentarios y valoraciones hechas por otros usuarios.

También está el botón valorar, que desplegará un modal para realizar comentarios y calificaciones sobre el punto limpio actual. Este modal se ilustra en la figura 26.



Figura 25 Pantalla Punto Limpio

8.3.6 Pantalla Comentario

En este modal, el usuario podrá ingresar un comentario y seleccionar una cantidad de estrellas específica que corresponderá a la calificación que el usuario le otorgará al punto limpio actual en el que se encuentra.

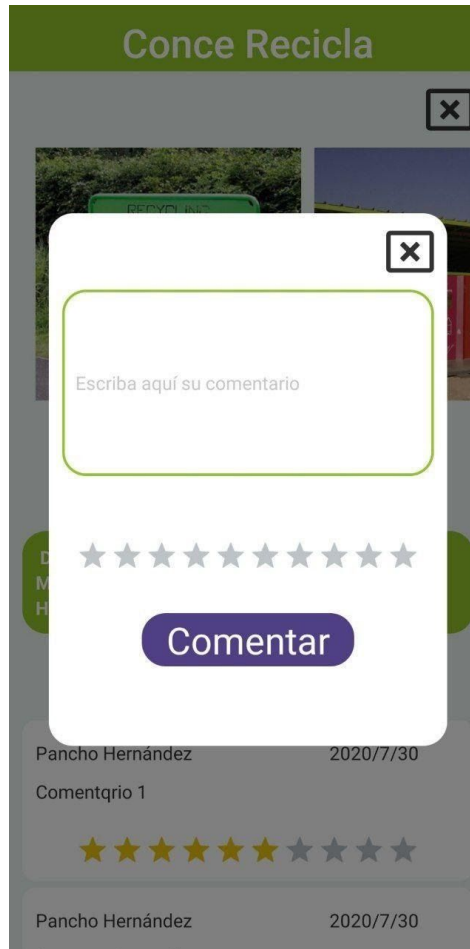


Figura 26 Modal Comentario

8.3.7 Pantalla Reciclar

En esta pantalla el usuario podrá realizar un registro de los diferentes materiales que recicle. Selecciona la cantidad de cada uno de los materiales y guarda este registro presionando el botón Reciclar.

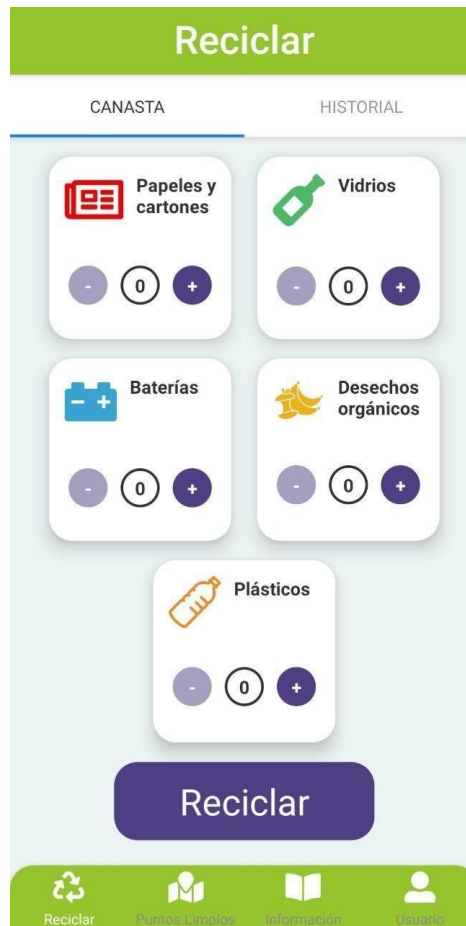


Figura 27 Pantalla Reciclar

8.3.8 Pantalla Historial Reciclaje

En esta pantalla el usuario podrá visualizar el registro histórico de todos sus materiales, y un gráfico con la cantidad total reciclada de cada material representada por un color específico.



Figura 28 Pantalla Historial Reciclaje

8.3.9 Pantalla Perfil de Usuario

En esta pantalla el usuario podrá visualizar y actualizar su información personal, además de poder cerrar la sesión actual.

Perfil Usuario

Nombre
Pancho

Apellido
Hernández

Edad
0

Guardar

Cerrar Sesión

Reciclar Puntos Limpios Información Usuario

Figura 29 Pantalla Perfil Usuario

8.3.10 Pantalla Agregar Punto Limpio

En esta pantalla se permite al administrador del sistema, agregar un punto limpio nuevo a la base de datos, seleccionando un punto en el mapa y llenando los campos correspondientes.



Figura 30 Pantalla Crear Punto Limpio



Figura 31 Pantalla Crear Punto Limpio2

Capítulo 9 Pruebas

La etapa, fase, o tarea de realizar pruebas es algo tan importante como la realización del análisis, diseño o construcción de un sistema. Específicamente en el desarrollo de un producto software, se recomienda destinar una buena parte del tiempo a realizar pruebas de lo construido. La realización de pruebas exhaustivas disminuye el riesgo de estar desarrollando un software que no cumpla o satisfaga las necesidades para las que fue solicitado.

9.1 Pruebas de Caja Negra

Las pruebas de caja negra, también denominadas, pruebas de comportamiento, se enfocan en verificar el cumplimiento de los requisitos funcionales del software. Es decir, permiten al ingeniero de software (o al equipo encargado de las pruebas) diseñar un conjunto de condiciones de entrada que buscan ejercitar o ejecutar por completo la totalidad de los requisitos funcionales de un programa. Las pruebas de caja negra, permiten enfocarse en el real funcionamiento del sistema y tienen un alto porcentaje de probabilidades de descubrir errores.

A continuación, se muestran los resultados de la ejecución de las pruebas de caja negra que fueron realizadas al proyecto. Estas pruebas son realizadas la aplicación móvil y también desde Postman para el caso de la API diseñada, hasta la manipulación de estos en la base de datos del sistema.

9.2 Resultados Pruebas Aplicación y API

Para un mejor entendimiento de las pruebas realizadas, estas serán organizadas en tablas, lo que permite una mejor visualización de los pasos a seguir y de los resultados de cada una de las pruebas. Cada una de las tablas representa una prueba a una funcionalidad específica de la aplicación.

9.2.1 Prueba 1: Iniciar Sesión

Propósito

Comprobar que los usuarios pueden iniciar sesión en la aplicación utilizando sus credenciales, correo y contraseña.

| | |
|-------------------------|---|
| Prerrequisitos | La cuenta de usuario ya debe existir en el sistema y no debe se debe estar ya autenticado en el sistema |
| Datos Correctos | Correo: franciscohernandez.u@gmail.com Contraseña: 12345678 |
| Datos Incorrectos | Correo: fcohernandez@gmail Contraseña: asd123 |
| Pasos Por Seguir | <ol style="list-style-type: none"> 1. Abrir aplicación. 2. Ingresar correo. 3. Ingresar contraseña. 4. Presionar el botón Iniciar Sesión |
| Resultados Esperados | <p>Si los datos ingresados son correctos y se siguen los pasos, la aplicación redirige al usuario a la pantalla de inicio.</p> <p>Si los datos son incorrectos, la aplicación enviará al usuario un mensaje de alerta indicando que los datos ingresados son incorrectos.</p> |
| Resultados Obtenidos | <p>Al ingresar los datos correctos, el usuario es redirigido a la pantalla de inicio de la aplicación.</p> <p>Al ingresar usuarios incorrectos, la aplicación muestra una alerta con un mensaje indicando que la contraseña y/o correo son incorrectos.</p> |
| Evaluación de la prueba | El sistema respondió en ambos casos probados, de la manera esperada. |

Tabla 28 Caso de Prueba: Iniciar Sesión

9.2.2 Prueba 2: Registrar Usuario

| | |
|-------------------|--|
| Propósito | Comprobar si es posible que los usuarios se registren en la aplicación de forma autónoma. |
| Prerrequisitos | Usuario no debe estar autenticado dentro de la aplicación y las credenciales de inicio de sesión no deben existir. |
| Datos Correctos | Nombre: Juan Apellido: Pérez Correo: juanito@gmail.com Contraseña: juan123 Confirmar contraseña: juan123 |
| Datos Incorrectos | Nombre: Juan2 Apellido: 23 Correo: juanito@gmail.com Contraseña: juan123 |

| | |
|-------------------------|---|
| | Confirmar contraseña: juan123 |
| Pasos Por Seguir | <ol style="list-style-type: none"> 1. Usuario se dirige a la pantalla de registro. 2. Usuario completa los campos para registrarse. 3. Usuario presiona el botón de Registrarse. |
| Resultados Esperados | <p>Si el usuario ingresa datos correctos, el usuario es creado y se redirige al usuario a la pantalla de inicio.</p> <p>Si el usuario ingresa datos incorrectos, la aplicación muestra un mensaje indicando que ha ocurrido un error al crear el usuario.</p> |
| Resultados Obtenidos | <p>Cuando el usuario ingreso datos correos, se redirige automáticamente a la pantalla de inicio.</p> <p>Con los datos incorrectos, la aplicación muestra una alerta que indica que ha ocurrido un error al crear la cuenta.</p> |
| Evaluación de la prueba | El sistema ha respondido con los resultados que se esperaban. |

Tabla 29 Caso de Prueba: Registrar Usuario

9.2.3 Prueba 3: Registrar Reciclaje Personal

| | |
|-------------------|---|
| Propósito | Comprobar el correcto funcionamiento del registro de reciclaje personal. |
| Prerrequisitos | El usuario debe estar autenticado dentro de la aplicación y en la pantalla de reciclaje. |
| Datos Correctos | <p>Papeles: 1 Vidrios: 0 Residuos Orgánicos: 0 Baterías: 0 Plásticos: 1</p> |
| Datos Incorrectos | <p>Papeles: 0 Vidrios: 0 Residuos Orgánicos: 0 Baterías: 0 Plásticos: 0</p> |
| Pasos Por Seguir | <ol style="list-style-type: none"> 1. Usuario selecciona la cantidad de cada uno de los materiales a reciclar. |

| | |
|-------------------------|---|
| | 2. Usuario presiona el botón Reciclar. |
| Resultados Esperados | <p>Si los datos ingresados son correctos, se muestra al usuario un mensaje que indique que la acción se ha realizado de manera exitosa.</p> <p>Si los datos ingresados son incorrectos, la aplicación muestra un mensaje al usuario indicando que ocurrió un error.</p> |
| Resultados Obtenidos | <p>Cuando se ingresaron datos correctos, la aplicación despliega una alerta indicando que el registro del reciclaje se ha realizado de manera exitosa.</p> <p>Con los datos incorrectos, la aplicación muestra al usuario una alerta indicando que ha ocurrido un error al registrar el registro del reciclaje.</p> |
| Evaluación de la prueba | El sistema respondió de la manera esperada tanto cuando se ingresan datos incorrectos como datos correctos. |

Tabla 30 Caso de Prueba: Reciclaje Personal

9.2.4 Prueba 4: Registrar comentario en un punto limpio.

| | |
|----------------------|---|
| Propósito | Verificar que los usuarios puedan registrar comentarios y valorarlos de manera correcta. |
| Prerrequisitos | El usuario debe estar autenticado en el sistema y estar en la pantalla de algún punto limpio. |
| Datos Correctos | Comentario: ¡Este punto limpio funciona perfectamente! Calificación: 5 |
| Datos Incorrectos | Comentario: ¡Este punto limpio funciona perfectamente! Calificación: 0 |
| Pasos Por Seguir | <ol style="list-style-type: none"> 1. Presionar el botón de Valorar en la pantalla de punto limpio. 2. Ingresar comentario y seleccionar cantidad de estrellas. 3. Presionar botón Comentar. |
| Resultados Esperados | Si los datos ingresados son correctos, el usuario es redirigido a la pantalla del mapa de los puntos limpios con un mensaje que indique que el comentario ha sido realizado con éxito. |

| | |
|-------------------------|---|
| | Si los datos ingresados son incorrectos se muestra un mensaje indicando que ha ocurrido un error al registrar el comentario. |
| Resultados Obtenidos | <p>Cuando los datos ingresados son correctos, la aplicación muestra una alerta indicando que el comentario ha sido registrado con éxito y se redirige a la pantalla del mapa de los puntos limpios.</p> <p>Cuando los datos ingresados son incorrectos, se despliega una alerta indicando que ha ocurrido un error al momento de registrar el comentario en el punto limpio actual y se redirige al usuario a la pantalla del mapa de puntos limpios.</p> |
| Evaluación de la prueba | El sistema respondió, en ambos casos, de la forma esperada. |

Tabla 31 Caso de Prueba: Registrar Comentario

9.2.5 Prueba 5: Ingresar al sistema con Facebook

| | |
|----------------------|--|
| Propósito | Verificar que el usuario puede ingresar al sistema con su cuenta de Facebook. |
| Prerrequisitos | El usuario debe tener una cuenta de Facebook y no estar autenticado dentro de la aplicación. |
| Datos Correctos | Correo: juancho727@outlook.com Contraseña: asdw123! |
| Datos Incorrectos | Correo: juancho727@outlook.com Contraseña: 12345 |
| Pasos Por Seguir | <ol style="list-style-type: none"> 1. El usuario presiona el botón para ingresar con Facebook. 2. Ingresar datos de la cuenta de Facebook. 3. Presionar botón ingresar. |
| Resultados Esperados | <p>Si los datos ingresados son correctos, la aplicación redirige a la pantalla de inicio.</p> <p>Si los datos ingresados son incorrectos, la aplicación redirige a la pantalla de login indicando que hubo un error.</p> |
| Resultados Obtenidos | Con los datos correctos, la aplicación redirige al usuario a la pantalla de inicio. |

| | |
|-------------------------|---|
| | Cuando los datos son incorrectos la aplicación redirige a la pantalla de login mostrando una alerta indicando que ocurrió un error al tratar de ingresar. |
| Evaluación de la prueba | La aplicación responde con los resultados esperados. |

Tabla 32 Caso de Prueba: Ingresar con Facebook

9.2.6 Prueba 6: Registrar comentario utilizando la API.

| | |
|----------------------|--|
| Propósito | Comprobar que, sin un token de acceso, no es posible comentar dentro de la aplicación. |
| Prerrequisitos | Obtener un token de acceso de algún usuario. |
| Datos Correctos | Token: Token Comentario: 'Comentario nuevo' Calificación: 3 |
| Datos Incorrectos | Token: 'espacio en blanco' Comentario: 'Comentario nuevo' Calificación: 5 |
| Pasos Por Seguir | <ol style="list-style-type: none"> 1. Se ingresa la url de la API en Postman. 2. Se ingresan los datos en el cuerpo de la petición, token, comentario y calificación. 3. Se realiza la petición. |
| Resultados Esperados | <p>Si los datos ingresados son correctos, el comentario es registrado en la base de datos y devuelve un mensaje de éxito.</p> <p>Si los datos ingresados son incorrectos, es devuelto un mensaje de error indicando que no está autorizado para realizar esta acción.</p> |
| Resultados Obtenidos | <p>Cuando los datos con correctos, la API devuelve un mensaje de todo ha funcionado de manera correcta y el comentario ha sido registrado.</p> <p>Con los datos incorrectos la API devuelve un mensaje indicando que la acción no se puede realizar porque el token ingresado es inválido.</p> |

| | |
|-------------------------|--|
| Evaluación de la prueba | La API ha respondido con los resultados esperados. |
|-------------------------|--|

Tabla 33 Caso de Prueba: Registrar Comentario Mediante API

Nota: Este es el token utilizado como dato correcto, se pone aquí abajo porque al ser muy grande, el formato de la tabla se ve alterado.

Token: “eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VhcmIvREliOnsiZ29vZ2xlljp0cnVILCJmYWNIYm9vayl6ZmFsc2UsIl9pZCI6IjVIZDU3OWUwY2VjYjQ0MjE2ODk1ZWVmZSIsIm5vbWJyZSI6IktZyYW5jaXNjbyIsImFwZWxsaWRvIjoisiSGVyb3OhbmRleilslmVtYWIsIjoiznJhbmNpc2NvaGVybmFuZGV6LnVAZ21haWwuY29tliwicGFzc3dvc mQiOiJ4ZCIsIl9fdil6MH0sImIhdCI6MTU5MTEzMTA0MSwiZXhwIjozNTkxMTMyNDgxf Q.Akf8es4lhN-WTrSEGaMDRJSrmct_4vODWufybsZtFak”

Capítulo 10 Conclusión

Al finalizar este proyecto se puede afirmar haber cumplido con la totalidad de objetivos generales y específicos, desarrollando un modelo de gestión de reciclaje y a partir de este un prototipo de aplicación móvil que permite el apoyo a dicho proceso.

Las aplicaciones móviles hoy en día y la interacción que permiten con los usuarios, además de lo útil que son como herramienta para diferentes procesos u organizaciones es algo increíble y que cada día toma más fuerza en un mundo digitalizado, comprender como son diseñadas y construidas es clave para afrontar futuros desafíos, ya que hacia esa dirección es hacia donde apunta el futuro del desarrollo de software, sin dejar de lado los sistemas web.

Relacionado con el prototipo de aplicación construida, las tecnologías utilizadas permiten que en un futuro esta crezca y puedan añadirse más funcionalidades de una manera sencilla, siendo altamente escalable, no solo en su interfaz sino también desde el lado del servidor, gracias a la flexibilidad que poseen las bases de datos NoSQL. Pero para poder dar paso a la construcción de esta aplicación, se tuvo que identificar los principales procesos y entidades que participan en la gestión de algo tan importante hoy en día como lo es el reciclaje.

Para identificar estos procesos y a sus participantes, se tuvieron conversaciones con encargados directos de la gestión en la Municipalidad, lo que permitió tener un gran entendimiento y facilitó el desarrollo del modelo y su conexión con cómo funciona el proceso realmente. Definir este modelo de una buena manera es de suma importancia no solo para este proyecto, si no para cualquier proyecto en general, sentar sólidas bases en el diseño del sistema o proceso al que se quiere prestar apoyo es clave para el posterior desarrollo del software, desde la interfaz de usuario hasta la base de datos dependen de una buena definición de requerimientos y de esto depende gran parte del éxito que tendrá.

Definidos el modelo y el prototipo de la aplicación terminado, se consigue mediante este sistema entregar una herramienta que permite a la Municipalidad ver la evaluación que tiene la ciudadanía respecto al proceso de reciclaje, específicamente en el proceso relacionado de vinculación con la comunidad, que tiene que ver con la información entregada a la población y como esta la percibe. Este proceso es clave ya que sin la participación ciudadana todo el proceso de reciclaje casi no tiene razón de ser.

Desde el punto de vista de lo académico, los conocimientos adquiridos durante el transcurso de la carrera sirvieron para forja una sola base que permitió afrontar el desarrollo de este proyecto con profesionalismo.

En el ámbito personal se lograron adquirir conocimientos muy importantes en nuevas tecnologías, que ya han abierto puertas en el desarrollo como profesional y personal, se mejoró la capacidad para analizar y diseñar distintos sistemas y modelos de flujos de datos, perfeccionando habilidades que permitirán sin duda seguir mejorando y creciendo como profesional para afrontar cualquier desafío que venga por delante.

Bibliografía

Comisión Nacional del Medio Ambiente, (2005). *Política de gestión integral de residuos sólidos*.

(Delía et al., 2013). *Un análisis experimental de tipo de aplicaciones para dispositivos móviles*.

Delía, L., Galdamez, N., Thomas, P., & Pesado, P. (2013). Un Análisis Experimental de Tipo de Aplicaciones para Dispositivos Móviles. *Instituto de Investigación En Informática LIDI*, 766–776.

Feitó Cespón, M., Castro, R. C., & Rubio Rodríguez, M. A. (2016). Modelos de optimización para el diseño sostenible de cadenas de suministros de reciclaje de múltiples productos Optimization model to sustainable design of multiple-products recycling supply chain. *Revista Chilena de Ingeniería*, 24(1), 135–148.

González, A. C. (2001). Una aproximación monetaria. *Costos y Beneficios Ambientales Del Reciclaje En México*, 58, 17–26. Retrieved from <http://www.redalyc.org/articulo.oa?id=53905802>

Ministerio del Medio Ambiente. (2014). *Informe final “Primera Encuesta Nacional de Medio Ambiente: Opiniones, Comportamientos y Preocupaciones.”*

Osorio, D. A. P., & Pineda Osorio, D. A. (2012). *Modelo para la gestión de reciclaje de residuos electrónicos*. Retrieved from <http://repositorio.uchile.cl/handle/2250/112257>
<http://www.repositorio.uchile.cl/handle/2250/112257>

Pavez, I., León, C., & Triadú, V. (2016). Jóvenes universitarios y medio ambiente en Chile: Percepciones y comportamientos. *Revista Latinoamericana de Ciencias Sociales*,

Niñez y Juventud, 14(2), 1435–1449.

<https://doi.org/10.11600/1692715x.14237041215>

Rojas, J. P. (2014). Residuos sólidos y calentamiento global – Parte 1. *Cegesti*, (254), 1–3. Retrieved from http://www.cegesti.org/exitoempresarial/publicaciones/publicacion_254_310314_es.pdf

Vásquez, O. (2005). Modelo de simulación de gestión de residuos sólidos domiciliarios en la Región Metropolitana de Chile. *Revista de Dinámica de Sistemas*, 1(Septiembre), 27–52.

Linkografía

Facebook (2020). *Documentación React*. Recuperado de: <https://es.reactjs.org/>

Facebook (2020). *Documentación React Native*. Recuperado de:
<https://facebook.github.io/react-native/>

Redux (2020). *Documentación Redux* Recuperado de: <https://redux.js.org/>

MongoDB Atlas (2020). *Documentación MongoDB* Recuperado de:
<https://docs.mongodb.com/manual/>