



Diseño y Programación de Software Multiplataforma

Proyecto de Cátedra

StockApp – G01T

Presentado por:

Bonilla Avilés, David Alejandro – BA181927
Cruz González, José Roberto – CG181933
Garay Alvarado, Bryan Walberto – GA181935

UNIVERSIDAD DON BOSCO

Enlace a repositorio: https://github.com/Garay01/DPS_StockApp

Tabla de contenido

INTRODUCCIÓN	3
DISEÑO UX/UI	4
MOCK UPS VISTA ESCRITORIO	4
MOCK UPS VISTA DISPOSITIVO MÓVIL	8
SOLUCIÓN A REALIZAR	10
DESCRIPCIÓN DEL PROYECTO	10 ^Z
LÓGICA DE RESOLUCIÓN DEL PROBLEMA - BACKEND	10
LÓGICA DE RESOLUCIÓN DEL PROBLEMA - FRONTEND	12
TECNOLOGÍAS A UTILIZAR	17
TECNOLOGÍAS USADAS EN EL BACKEND	17
TECNOLOGÍAS USADAS EN EL FRONTEND	17
PRESUPUESTO	18
COSTOS DE MANO DE OBRA ETAPA DE EJECUCIÓN	18
COSTOS DE MATERIALES A UTILIZAR ETAPA DE EJECUCIÓN	19
PRESUPUESTO ESTIMADO	19
BIBLIOGRAFÍA.....	20

Introducción

El proyecto a realizar en la materia de Diseño y Programación de Software Multiplataforma, busca la automatización así como la digitalización del proceso de control de inventario realizado dentro del departamento de informática en la alcaldía de Ilopango, los requerimientos presentados buscan solventar el manejo y toma de datos de los repuestos adquiridos por dicho departamento, al igual que sus respectivas descargas a las demás áreas dentro de la alcaldía.

El presente documento, pretende evidenciar las formas, diseño y métodos a realizar para no solamente cumplir con los requerimientos, sino también, construir una aplicación que pueda ser utilizada desde diferentes dispositivos sin importar el tamaño o resolución de este, satisfaciendo el objetivo de la materia.

Diseño UX/UI

Mock Ups vista escritorio

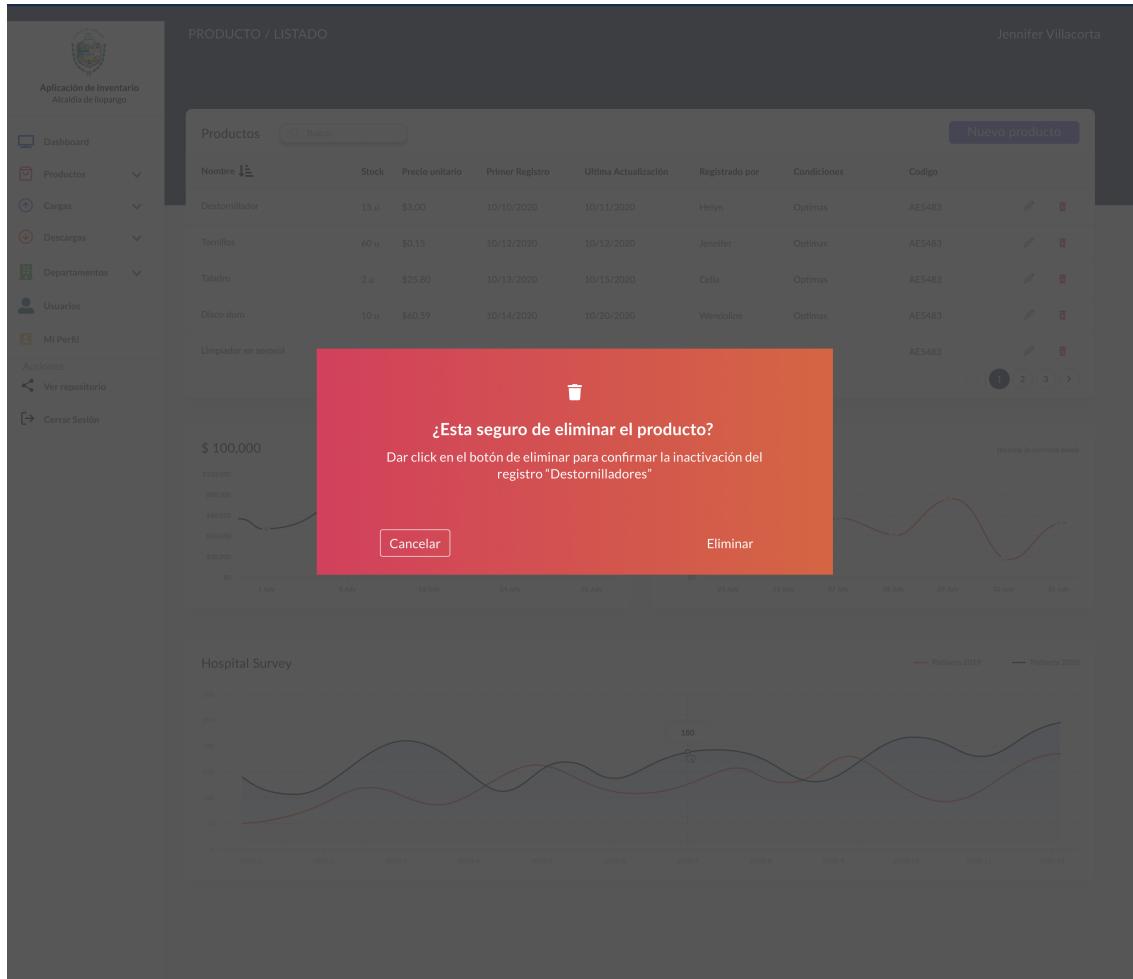
Registro de productos

The mockup shows a dark-themed desktop application window titled 'PRODUCTO / LISTADO'. In the top right corner, the name 'Jennifer Villacorta' is displayed. On the far left, a vertical sidebar contains the application's logo and navigation menu items: 'Dashboard', 'Productos' (selected), 'Cargas', 'Descargas', 'Departamentos', 'Usuarios', and 'Mi Perfil'. Below the menu, there are two buttons: 'Acciones' and 'Ver repositorio'. At the bottom of the sidebar is a 'Cerrar Sesión' button. The main content area is a table titled 'Productos' with a search bar above it. The table has columns: Nombre, Stock, Precio unitario, Primer Registro, Ultima Actualización, Registrado por, Condiciones, and Código. The table lists five products:

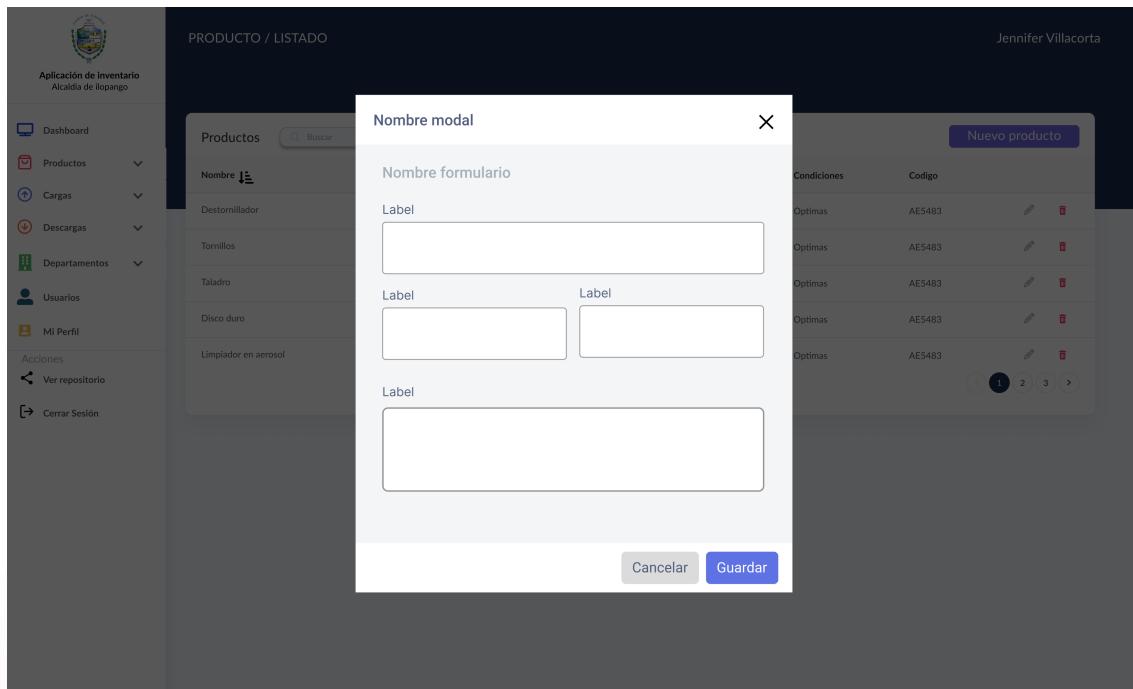
Nombre	Stock	Precio unitario	Primer Registro	Ultima Actualización	Registrado por	Condiciones	Código
Destornillador	15 u	\$3.00	10/10/2020	10/11/2020	Helyn	Opcionales	AE5483
Tornillos	60 u	\$0.15	10/12/2020	10/12/2020	Jennifer	Opcionales	AE5483
Taladro	2 u	\$25.80	10/13/2020	10/15/2020	Celia	Opcionales	AE5483
Disco duro	10 u	\$60.59	10/14/2020	10/20/2020	Wendoline	Opcionales	AE5483
Limpiajador en aerosol	5 u	\$4.65	10/15/2020	10/30/2020	Celia	Opcionales	AE5483

At the bottom right of the table are navigation icons for page 1, 2, 3, and a next arrow. A purple 'Nuevo producto' (New product) button is located at the top right of the table header.

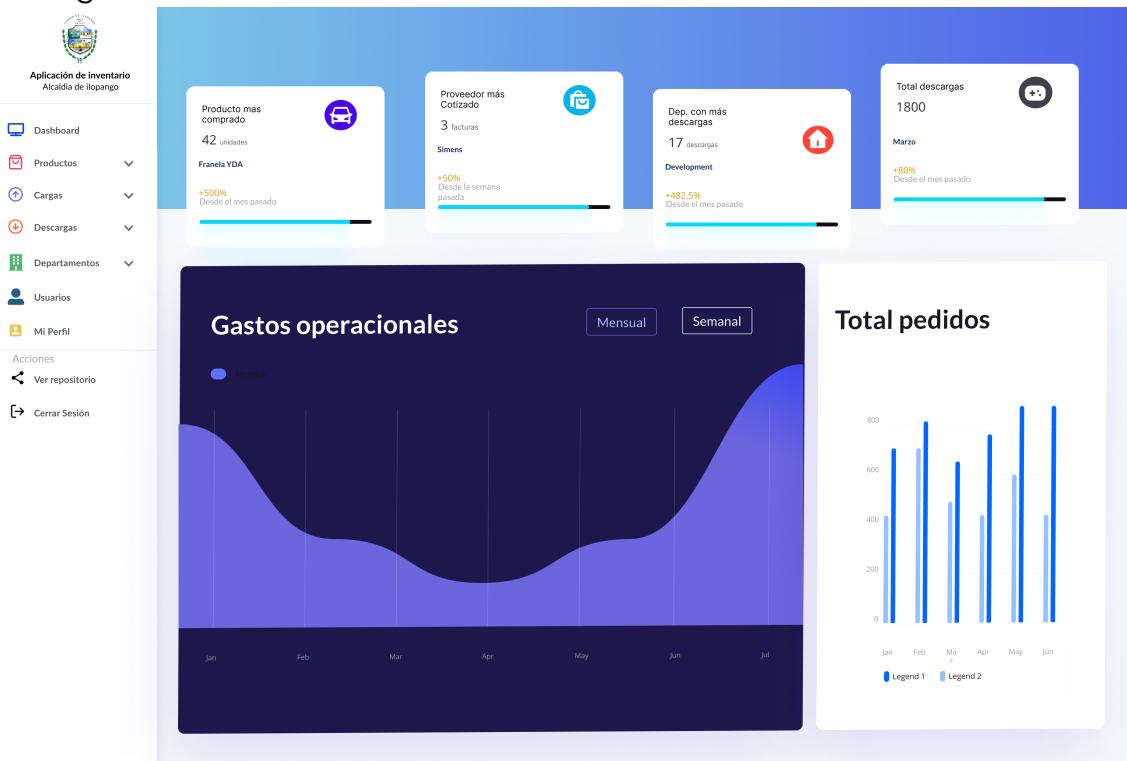
Vista modal de información



Vista formulario en ventana modal



Vista graficas



Vista inicio de sesión

The screenshot shows the login page for the inventory system. It features a teal header with the city coat of arms and the text "Sistema de inventario" and "Departamento de informática". Below the header is a "Login" button. The main area contains a "Iniciar sesión" form with fields for "Correo electrónico" and "Contraseña", and a "Entrar" button. At the bottom of the form is a link "Olvido su contraseña?".

Vista formulario de reportes

DESCARGA / REPORTE

Jennifer Villacorta

Reporte - Descarga reporte

Filtros de reportes

Fecha inicio:

Fecha finalización:

Aceptar

Aplicación de inventario
Alcaldía de Ilopango

- Dashboard
- Productos
- Cargas
- Descargas
- Departamentos
- Usuarios
- Mi Perfil

Acciones

- Ir repository
- Cerrar Sesión

Vista Correo electrónico

Titulo del correo StockApp



Contenido



Mock Ups vista dispositivo móvil

Vista tabla de registros y Menú

StockApp

PRODUCTO / LISTADO Jennifer Villacorta

Productos

Nombre	Stock	Precio unitario	Primer Registro	Registrado por
Destornillador	15 u	\$3.00	10/10/2020	Helyn
Tornillos	60 u	\$0.15	10/12/2020	Jennifer
Taladro	2 u	\$25.80	10/13/2020	Celia
Disco duro	10 u	\$60.59	10/14/2020	Wendol
Limpiador en aerosol	5 u	\$4.65	10/15/2020	Celia

© 2021 Alcaldía municipal de Ilopango
Desarrolladores: Garay Alvarado José Crúz
David Bonilla
Licencia Creative Commons

Aplicación de inventario
Alcaldía de Ilopango

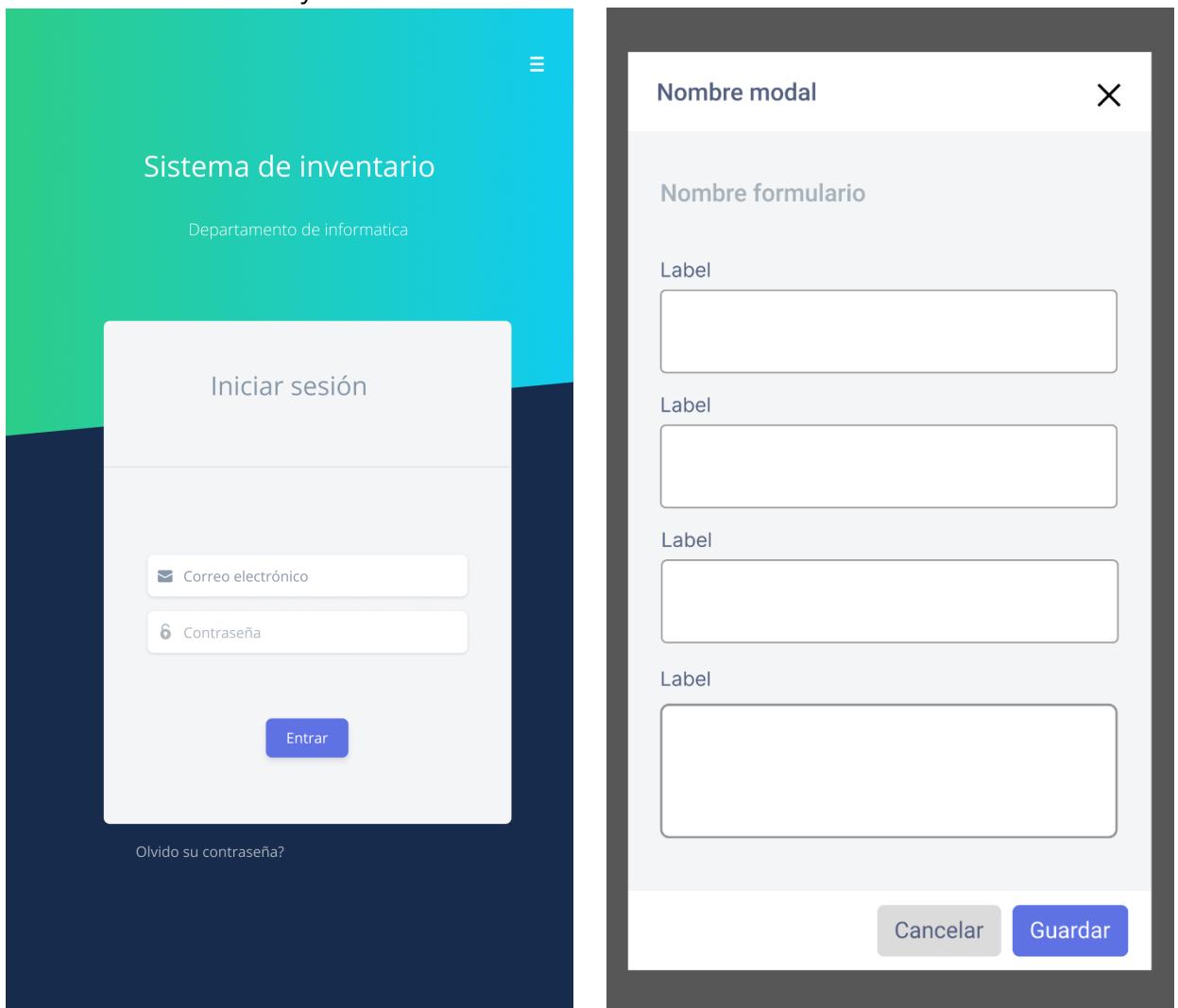
- Dashboard
- Productos
- Cargas
- Descargas
- Departamentos
- Usuarios
- Mi Perfil

Acciones

- Ver repositorio
- Cerrar Sesión

© 2021 Alcaldía municipal de Ilopango
Desarrolladores: Garay Alvarado José Crúz
David Bonilla
Licencia Creative Commons

Vista inicio de sesión y formulario en una ventana modal



Solución a realizar

Descripción del proyecto

Este proyecto de servicio social pretende agilizar y mejorar los procesos de la alcaldía municipal de Ilopango, específicamente en el departamento de tecnología el cual requiere automatizar un proceso interno que actualmente se

lleva de forma manual "Control de inventario de repuestos y accesorios", a través de la creación de una aplicación web que gestione los inventarios.

La aplicación web contara con las siguientes características:

- Creación, eliminación y modificación de usuarios
- Creación, eliminación, modificación y asignación de repuestos adquiridos por el departamento de tecnología
- Creación, eliminación, modificación e impresión de facturas
- Creación e impresión de bitácoras pertenecientes al repuesto
- Descargas de inventario

Lógica de resolución del problema - Backend

Para la base de datos a utilizar se ha optado por MySql, esto por el modelo relacional que ofrece este tipo de BD's, y se requiere un modelo relacional debido a que las inserciones o actualizaciones de datos no son masivas ni a gran escala, y mucho menos las estructuras de datos planteadas requiere de relaciones complejas, el esquema utilizado para concluir en lo anteriormente descrito es el siguiente:

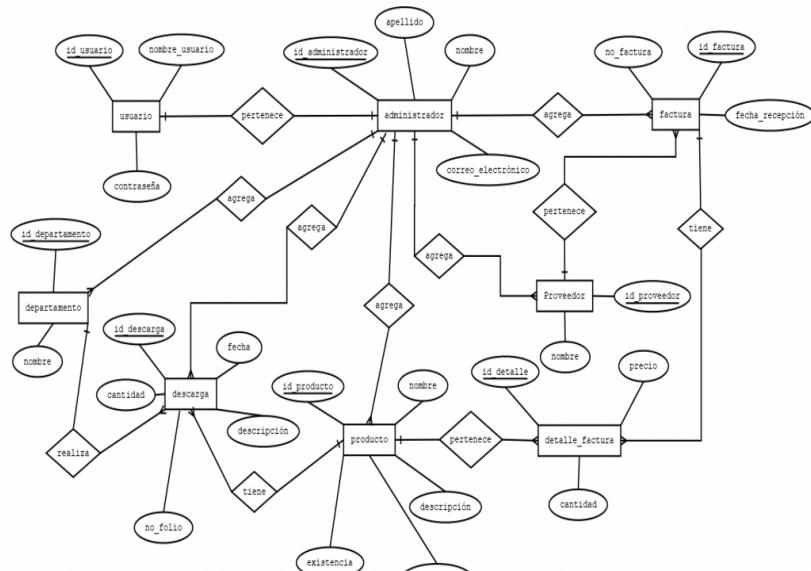


Ilustración 1 - Diagrama ER

Como datos finales o recopilación de datos necesarios a partir del diagrama anterior se puede notar que las entidades involucradas dentro del sistema serían las siguientes:

- Usuarios
- Administradores
- Facturas
- Detalles
- Proveedores
- Productos
- Descargas
- Departamentos

Pasando estos datos recopilados por el diagrama "ER" se diseñó un diagrama relacional para previsualizar la estructura final o prototipo de la base de datos:

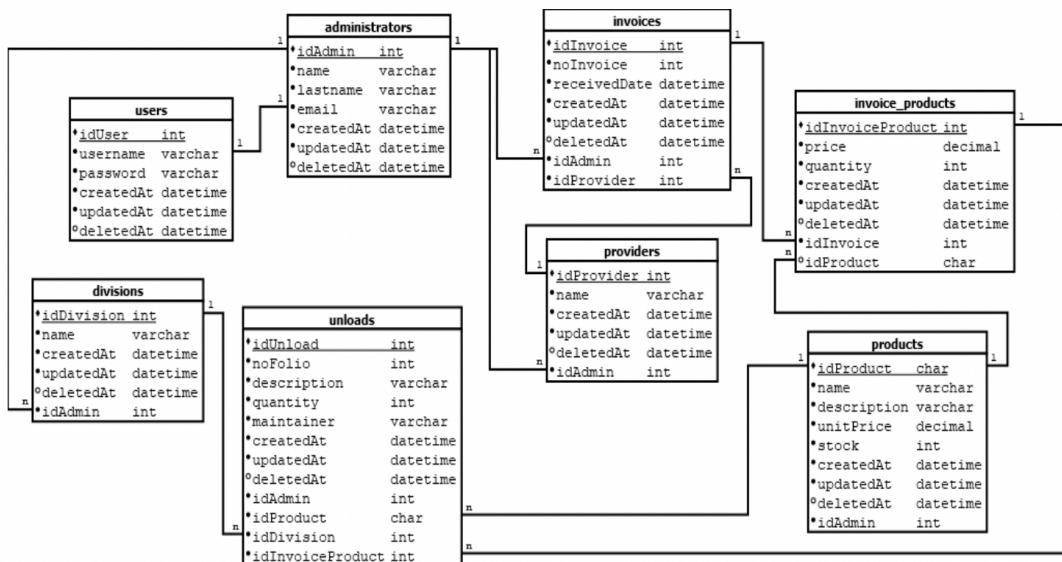


Ilustración 2 - Modelo Relacional

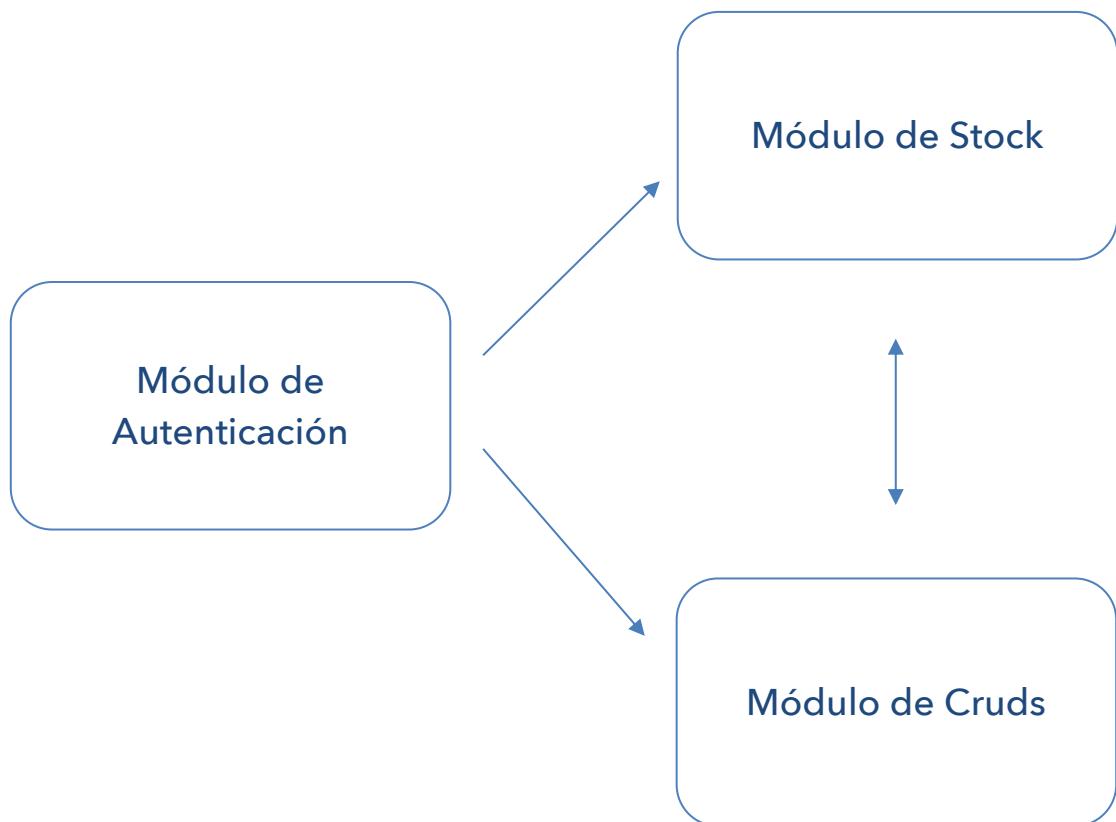
Luego de haber definido la lógica de resolución a implementar en la base de datos, se buscó la manera de consumir estos datos por medio de un servidor, a su vez, se planteó por medio de los encargados de la Alcaldía Municipal de Ilopango que dicho servidor o servicio debía servir la vista o aplicativo, para que no tenga que ser instalada individualmente en ninguna máquina, por lo tanto, dicho servicio no solo tenía que servir los datos, sino también los recursos por medio de los cuales deberá cargarse la aplicación web.

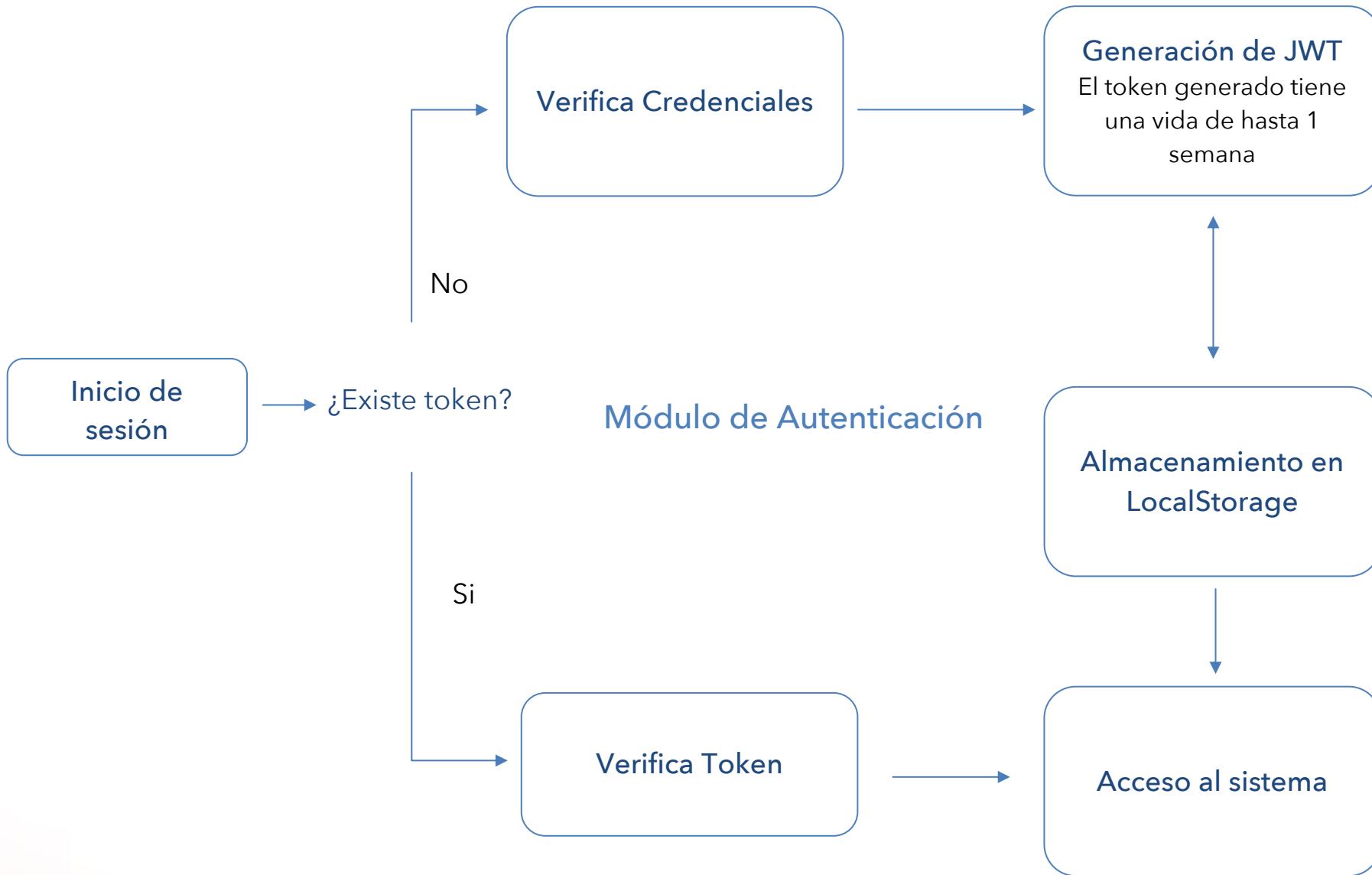
Con el objetivo de cumplir con el propósito de la materia se decidió utilizar la tecnología de GraphQL para el servidor, el cuál es una tecnología utilizada por Facebook para la entrega dinámica de datos a todas sus aplicaciones, ya

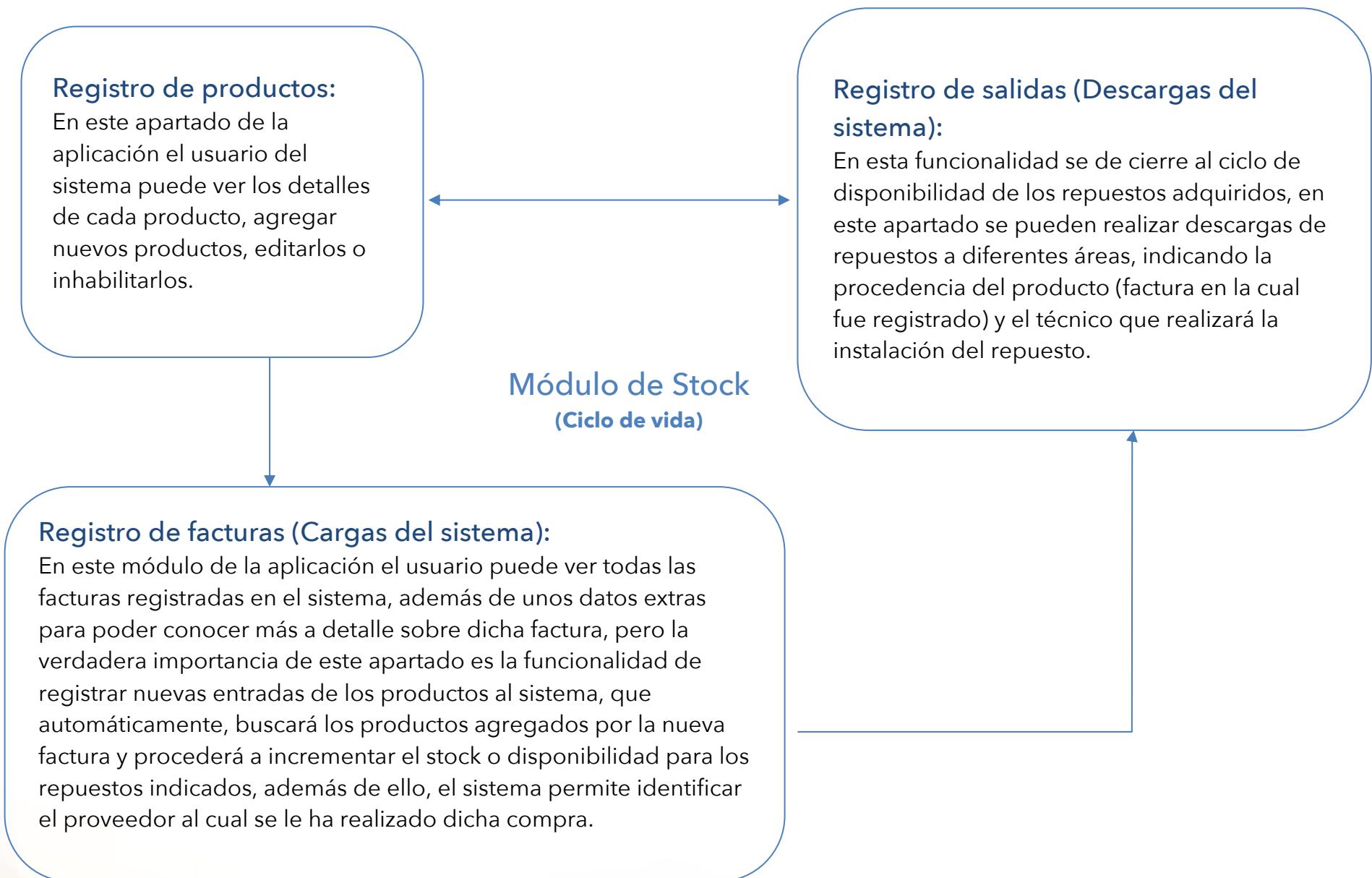
sea móviles o web. Y para poder servir la aplicación por medio del mismo servidor se hará uso de Apollo - Express server y de esta manera poder servir recursos del aplicativo además de la API - GraphQL que servirá los datos (Cada una de estas tecnologías serán descritas con mayor detenimiento en los siguientes apartados).

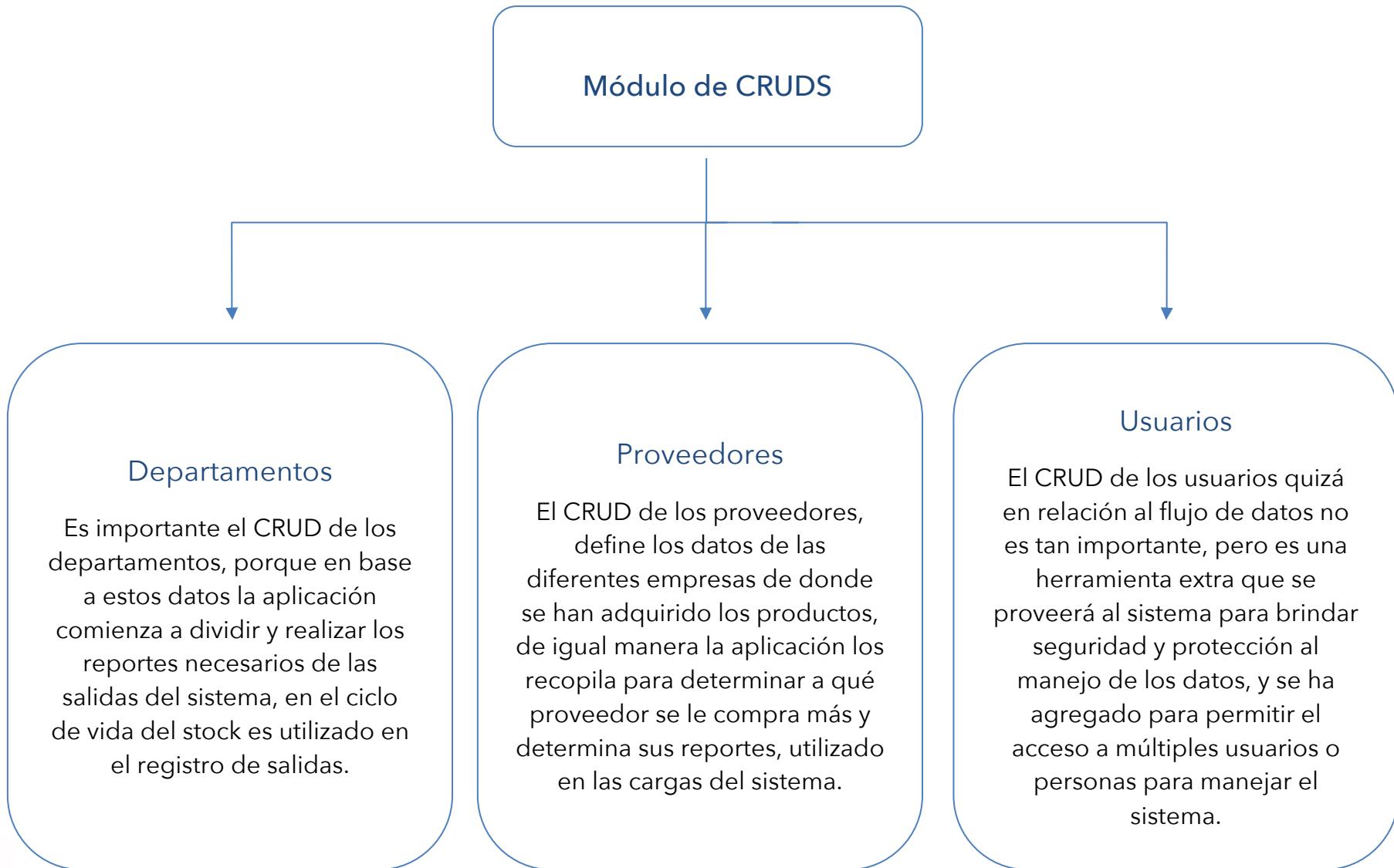
Lógica de resolución del problema - Frontend

Ya habiendo definido cómo se deberá servir el aplicativo y la manera en la que se podrán consumir los datos, la mejor tecnología para trabajar el proyecto en el frontend se ha decidido que sea React de esta manera podría consumir sin mayor problema la API - GraphQL y puede ser servido como recursos por medio del servidor, se han planteado los siguientes módulos de la aplicación:









Módulo de Mensajería

La aplicación empleará un servidor SMTP para poder realizar notificaciones de registro por medio de correo electrónico, al igual que en el proceso de recuperación de contraseñas, de esta manera la aplicación tiene mayor interacción con los usuarios del mismo.

Herramientas y Funcionalidades extra

Módulo de Reportería

Por último, todos estos datos que maneja el sistema tienen su importancia en esta sección donde la aplicación los ordenará, filtrará y expondrá con el fin de dar información concreta por medio de la creación de Reportes y estadísticas que ayuden a la lectura de estos datos.

Tecnologías a utilizar

A manera de resumen se detallan a continuación las tecnologías y herramientas a utilizar durante el desarrollo:

Tecnologías usadas en el backend

- **NodeJs:** Entorno de ejecución de código abierto y multiplataforma que permite ejecutar código JavaScript fuera del navegador, se utilizará para realizar todo el código para el manejo de datos entre el cliente y la base de datos.
- **GraphQL:** Lenguaje de consultas para API (Interfaz de programación de aplicaciones) se utilizará para el control de peticiones de parte del cliente, quien realizará las consultas y este deberá ir a traer los datos necesarios a MySql.
- **Apollo server express:** Herramienta de código abierto que permite crear servidores GraphQL, se utilizará como motor de la API GraphQL, y que pueda establecer la comunicación entre el cliente y el servidor.
- **MySQL:** Sistema de gestión de base de datos relacionales, donde se creará la base de datos anteriormente descrita para almacenar y gestionar los datos del sistema.
- **Sequelize:** Herramienta que permite mapear bases de datos relacionales y poder comunicarse con ella, se utilizará para establecer la comunicación entre el servidor y la base de datos, además, realizar cada consulta por medio de su lenguaje de consultas.

Tecnologías usadas en el frontend

- **Javascript:** Lenguaje de programación interpretado por el navegador, permite la ejecución del programa lo más limpia y suave posible, es la base de las demás herramientas a utilizar dentro del frontend.
- **React:** Librería de manejo del DOM de javascript, permite versatilidad y respuesta a cambios en la vista de manera ágil y reactiva, donde se diseñarán cada una de las interfaces y pantallas del diseño, utilizada

por el requerimiento de no instalación en dispositivos específicos que presentaron los encargados del proyecto, pero que de igual manera se contemplará el uso de la aplicación en dispositivos móviles.

- **Apollo client:** Cliente de Apollo para poder leer y ejecutar mutaciones de graphql desde la vista, permitirá el control del estado de la aplicación durante todo el uso de ella, almacena los datos y permite el uso de las entradas y salidas para mostrar dichos datos en cada una de las pantallas (similar a Redux).
- **TypeScript:** Agrega tipado a Javascript, fundamental durante el desarrollo de la aplicación para errores de tipo o acceso a datos inexistentes, utilizado para una mayor y mejor reutilización del código, evitando errores de sintaxis o redundancias en él.

Presupuesto

La aplicación es realizada sin ningún fin de lucro, pero se ha colocado la siguiente estimación con el objetivo de valorar en términos de gastos y costos el proyecto:

Costos de mano de obra etapa de ejecución

Integrante	Hora de trabajo	Horas totales de trabajo previstas	Total, por trabajador
Cruz González, José Roberto	\$5	240	\$1200
Bonilla Avilés, David Alejandro	\$5	240	\$1200
Garay Alvarado, Bryan Walberto	\$5	240	\$1200
Total			\$3600

Tabla 1 Costo de mano de obra, etapa de ejecución

Costos de materiales a utilizar etapa de ejecución

Recurso o Servicio	Cantidad / Calidad	Tipo	Costo/uso	Tiempo de uso	Total
Internet	3 contratos / 5 MB	Costo	\$24 / mes	1.5 mes	\$108.00
Electricidad	3 facturas	Costo	\$0.1242/k Wh	360 horas	\$130.14
Licencia Office 365	3	Costo	Versión gratuita	6 semanas	\$0.00
Computadoras	3	Costo	\$5 / mes (Depreciación)	1.5 mes	\$22.50
Sistemas Operativos (Basados en Linux)	5	Costo Software	\$0	Indefinido	\$0.00
Servidor Bases de Datos	2	Costo Software	\$0	Indefinido	\$0.00
Visual Studio Code	3	Herramienta	\$0	1.5 mes	\$0.00
Total, en Materiales					\$260.64

Tabla 2 Costos en materiales de producción

Presupuesto estimado

	Costos	Total
Mano de Obra		\$3600
Materiales para utilizar		\$260.64
Ganancia		30%
Total, Presupuesto		\$3860.64
Total, Valor del Producto		\$5018.83

Tabla 3 Presupuesto estimado

Bibliografía

OPENWEBINARS.NET. 2021. QUÉ ES NODEJS Y PARA QUÉ SIRVE. [ONLINE] AVAILABLE AT: <[HTTPS://OPENWEBINARS.NET/BLOG/QUE-ES-NODEJS/](https://openwebinars.net/blog/que-es-nodejs)> [ACCESSED 6 SEPTEMBER 2021].

EDTEAM - NO TE DETENGAS. 2021. AGILIZA TU DESARROLLO EN NODEJS CON EL ORM SEQUELIZE. [ONLINE] AVAILABLE AT: <[HTTPS://ED.TEAM/BLOG/AGILIZA-TU-DESARROLLO-EN-NODEJS-CON-EL-ORM-SEQUELIZE](https://ed.team/blog/agiliza-tu-desarrollo-en-nodejs-con-el-orm-sequelize)> [ACCESSED 6 SEPTEMBER 2021].

MEDIUM. 2021. INTRODUCCIÓN A APOLLO CLIENT CON REACT PARA GRAPHQL. [ONLINE] AVAILABLE AT: <[HTTPS://MEDIUM.COM/@KHRIZTIANMORENO/INTRODUCCI%C3%B3N-A-APOLLO-CLIENT-CON-REACT-PARA-GRAHQL-97FC919B9DEF](https://medium.com/@khriztianmoreno/introducci%C3%B3n-a-apollo-client-con-react-para-graphql-97fc919b9def)> [ACCESSED 6 SEPTEMBER 2021].

MEDIUM. 2021. 3 WAYS TO IMPLEMENT RESPONSIVE DESIGN IN YOUR REACT APP. [ONLINE] AVAILABLE AT: <[HTTPS://ITNEXT.IO/3-WAYS-TO-IMPLEMENT-RESPONSIVE-DESIGN-IN-YOUR-REACT-APP-BCB6EE7EB424](https://itnext.io/3-ways-to-implement-responsive-design-in-your-react-app-bcb6ee7eb424)> [ACCESSED 6 SEPTEMBER 2021].

ZAGROBELNA, M., 2021. CÓMO HACER UN MOCKUP EN 5 FÁCILES PASOS. [ONLINE] DESIGN & ILLUSTRATION ENVATO TUTS+. AVAILABLE AT: <[HTTPS://DESIGN.TUTSPLUS.COM/ES/TUTORIALS/HOW-TO-MAKE-A-MOCKUP-IN-5-EASY-STEPS--CMS-32316](https://design.tutsplus.com/es/tutorials/how-to-make-a-mockup-in-5-easy-steps--cms-32316)> [ACCESSED 6 SEPTEMBER 2021].