

UNIVERSIDAD LUTERANA SALVADOREÑA

FACULTAD DE CIENCIAS DEL HOMBRE Y LA NATURALEZA

LICENCIATURA EN CIENCIAS DE LA COMPUTACIÓN



PROYECTO

Api con JavaScript Inventario de Telas

EQUIPO DESARROLLADOR

APELLIDOS, NOMBRES

CARNET

| | |
|---------------------------------|-------------|
| Monge Cerón Liliana de Ángeles | MC01136242 |
| Aguilar Sánchez Henry David | AS011334801 |
| Fabian Martínez Elvis Henry | FM01134934 |
| García Flores Luis Roberto | GF01135913 |
| Fernández Obispo Marlon Ernesto | FO01135128 |

CÁTEDRA: Nuevas Tendencias de Programación

HORARIO: Sábado de 07:00 a 08:40am

FACILITADOR: Licenciado Irwin Guardado

SAN SALVADOR, 9 DE DICIEMBRE DE 2022

INTRODUCCIÓN

Una de las principales situaciones por las que pasan a diario las empresas u organizaciones, es que el mundo se va globalizando, buscando ser más competitivas entre sí, es por ello que sufren de pérdidas ya sea en productos o capital, y por tal razón algunas empresas han optado en cerrar sus establecimientos, por lo tanto para poder lograr un mejor rendimiento es necesario implementar herramientas informáticas para la gestión administrativa.

Los sistemas de inventarios son un proceso básico para todo tipo de empresas y en especial para aquellas que gestionan un número muy variado de productos, este proceso permite conocer la localización exacta de las productos o artículos con que se cuenta, se puede evitar errores y retrasos en las tareas de almacenaje y preparación de pedidos implementando este sistema.

Este documento describe el trabajo realizado en el proyecto a presentar para la materia de Nuevas tendencias de programación. El proyecto consiste en el desarrollo de una api basada en Javascript desde donde se podrá gestionar el Stock de telas de un almacén además se podrá modificar los datos como sucursales disponibles y dichos datos se actualizarán de forma inmediata ya que utilizando este tipo de API se puede obtener datos desde el servidor sin tener que volver a cargar toda la página o aplicación logrando que estas tengan una sensación de rapidez y agilidad donde se podrán enviar y modificar datos desde cualquier lugar o cualquier momento.

En cuanto a los usuarios, la API tiene dos tipos. Por un lado, están los usuarios registrados previamente con un Id, que únicamente pueden acceder a información general sobre las existencias pero solo visualizarla y están los usuarios con permiso de administrador distinguidos con otro Id, que pueden acceder a una mayor funcionalidad dentro de la API pueden gestionar la base de datos pueden agregar y eliminar productos y sucursales.

índice

| | |
|--|----|
| INTRODUCCIÓN..... | 2 |
| ANÁLISIS DEL PROBLEMA | 4 |
| TEMA: Sistema de Inventario de Telas..... | 4 |
| PROBLEMÁTICA A RESOLVER..... | 4 |
| OBJETIVOS | 5 |
| OBJETIVO GENERAL:..... | 5 |
| OBJETIVOS ESPECÍFICOS: | 5 |
| ANÁLISIS DE BENEFICIOS DEL PROYECTO | 6 |
| BENEFICIOS AL EMPRESARIO:..... | 6 |
| BENEFICIOS PARA LOS CLIENTES:..... | 6 |
| BENEFICIOS PARA LOS TRABAJADORES: | 6 |
| JUSTIFICACIÓN | 7 |
| METODOLOGÍA Y CICLO DE VIDA UTILIZADOS: | 8 |
| REQUERIMIENTOS FUNCIONALES..... | 8 |
| REQUERIMIENTOS NO FUNCIONALES..... | 9 |
| MODELO DE CICLO DE VIDA: | 10 |
| ANÁLISIS:..... | 11 |
| DIAGRAMA DE FLUJO DE DATOS (DFD) | 11 |
| DIAGRAMAS DE CASOS DE USO..... | 12 |
| DIAGRAMA DE CLASES..... | 13 |
| DIAGRAMA DE BASE DE DATOS | 14 |
| DISEÑO: | 15 |
| DESARROLLO | 17 |
| A) TÉCNICAS DE PROGRAMACIÓN APLICADAS | 17 |
| B) HERRAMIENTAS DE DESARROLLO UTILIZADAS..... | 17 |
| C) LENGUAJE DE PROGRAMACIÓN Y GESTORES DE BASES DE DATOS U OTRAS HERRAMIENTAS UTILIZADAS EN EL DESARROLLO DEL PROYECTO. | 19 |
| CONCLUSIONES..... | 22 |
| RECOMENDACIONES | 23 |
| BIBLIOGRAFÍA:..... | 24 |
| ANEXOS..... | 25 |
| MANUAL DE USUARIO:..... | 25 |
| MANUAL DE DESARROLLADOR..... | 38 |

ANÁLISIS DEL PROBLEMA

TEMA: Sistema de Inventario de Telas.

PROBLEMÁTICA A RESOLVER

El proyecto que está siendo presentado de un sistema de Inventario de telas, desarrollado en la colonia Esperanza, se ha observado la problemática que no cuentan con un sistema de inventario actualizado para poder llevar el control de las salidas y entradas de los productos . Lo tradicional está quedando atrás, hoy en día la mayoría de personas se están actualizando con las tecnologías porque estas van innovándose día a día, para facilitar el trabajo a las personas.

Estos negocios prefieren realizar la gestión de sus recursos de manera manual y sin una organización adecuada para su realización, lo cual requiere mucho tiempo invertido para registrar las transacciones, en el cual las empresas o negocios sufren de pérdidas en sus rendimientos y por lo tanto generar un mayor costo, además que los datos no son exactos y verídicos para la generación de los reportes que permitan la toma de mejores decisiones estratégicas que beneficien el crecimiento de la empresa Sky.

El sistema de inventario de telas será llevado de forma virtual para brindar un mejor beneficio, el sistema tendrá un mejor control de los productos que se irán adquiriendo para ofrecer a las personas, así dando una mejor efectividad al inventario registrando los artículos y clasificación de estos. Las empresas y negocios están optando por la implementación de bases de datos para la organización y gestión de sus sistemas. Sin embargo, la mayoría de negocios pequeños o en crecimiento no cuentan con estos beneficios para su organización y gestión.

Además del retraso en la atención a los clientes por la falta de un sistema automático de control de inventario y registró también la poca capacidad de los métodos actuales que no benefician para la optimización del tiempo de operación en el proceso de ventas.

OBJETIVOS

OBJETIVO GENERAL:

Presentar una API REST para la gestión de inventario de un almacén de telas. Aplicando los conocimientos obtenidos en clase para llevar a cabo el desarrollo de una API REST analizando e interpretando el código para su elaboración con la ayuda de herramientas para hacerlo más atractivo.

OBJETIVOS ESPECÍFICOS:

1. Mostrar la funcionalidad de API REST basada en los requerimiento para poder gestionar el inventario de tela.
2. Especificar las herramientas utilizadas para la programación de una Api rest.
3. Detallar cada una de las funciones de Api Rest en manuales para que el usuario pueda conocerlas.

ANÁLISIS DE BENEFICIOS DEL PROYECTO

El sistema tendrá muchos beneficios tanto para el empresario, el dueño de la empresa, los clientes así como también para los trabajadores, lo cual se detalla de forma clara a continuación:

BENEFICIOS AL EMPRESARIO:

- Mejora el flujo de trabajo.
- Gestión de inventarios.
- Agilidad en los procesos administrativos.
- Ayuda a organizar a tu equipo.
- Aumentar los ingresos.
- Mantener sincronizadas con la misma información.
- Gestionar los datos de forma segura.

BENEFICIOS PARA LOS CLIENTES:

- Reduce el tiempo de atención a clientes.
- Acceso a la información en tiempo real
- Servicio al cliente más eficiente.
- Mejor satisfacción de clientes.
- Reducción de quejas y reclamos

BENEFICIOS PARA LOS TRABAJADORES:

- Automatizar las tareas del día a día.
- Optimiza el control de inventario.
- Simplifica en la contabilidad.
- Mejora la relación con los clientes

JUSTIFICACIÓN

La tecnología ayuda a resolver diversos problemas o en todo caso ayuda a facilitar la solución a los problemas, de igual manera aporta grandes beneficios a la sociedad. Esta razón es la que impulsa a dar una solución al problema surgido en una empresa con inventario de Telas; Ya que esta empresa ha reconocido la importancia de contar con un Sistema que les ayude a facilitar su inventario, ya que presentan dificultades con el procesamiento y manejo de la información que se realiza para el control del inventario, actualmente lo manejan con herramientas ofimáticas como Word, Excel o documentos físicos, detallando que la empresa no cuenta con un sistema digital de inventario que permita controlar los productos en stock de forma eficiente y oportuna.

Por lo que esto genera la necesidad notable de contar con una herramienta informática, para mantener un estricto control de inventario; El presente proyecto busca modernizar, facilitar el control y actualización de datos de productos existentes.

La herramienta propuesta resguardará la información relevante de los productos en una base de datos en MYSQL lo que permitirá a los usuarios acceder a la información disponible del inventario. Ya que hoy en día la mayoría de negocios está tomando en cuenta los beneficios que trae el uso de la tecnología ya que ayuda al crecimiento laboral, permite brindar un mejor servicio a sus clientes. Otro aspecto importante, es que dicho sistema permitirá resguardar la información de forma exacta y confiable, la cual estará disponible para su respectivo análisis y contribuir a la toma de decisiones del negocio.

METODOLOGÍA Y CICLO DE VIDA UTILIZADOS:

REQUERIMIENTOS FUNCIONALES

1. El usuario podrá ingresar datos a los diferentes campos de las tablas.

1.1 El sistema tendrá que mostrar la tabla de usuario.

1.2 El sistema tendrá que mostrar la tabla de ancho.

1.3 El sistema tendrá que mostrar la tabla de calidad

1.4 El sistema tendrá que mostrar la tabla de roll

1.5 El sistema tendrá que mostrar la tabla de sucursal.

1.6 El sistema tendrá que mostrar la tabla de tela.

1.7 El sistema debe de mostrar la tabla unidad de medida.

2. El usuario podrá ingresar datos.

2.1 El sistema tendrá que procesar los datos.

2.2 El sistema mostrará los datos almacenados.

2.3 El sistema tendrá la funcionalidad de actualizar y eliminar datos existentes.

3. El usuario podrá actualizar los registros almacenados.

3.1 El sistema tendrá que mostrar los datos actuales.

3.2 El sistema tendrá que ocupar la función de PUT para poder actualizar los datos.

4. El usuario podrá eliminar los registros almacenados.

4.1 El sistema tendrá que mostrar los datos actuales para poder eliminar el dato.

4.2 El sistema tendrá que mostrar los datos almacenados y seleccionar el id para eliminar el registro deseado.

4.3 El sistema tendrá que ocupar la función delete para poder eliminar los datos solicitados.

REQUERIMIENTOS NO FUNCIONALES

1. El sistema tendrá que procesar los datos ingresados por el usuario.

2. El sistema tendrá una interfaz sencilla.

3. El sistema tendrá que permitir la actualización de datos existentes.

4. El sistema permite la conexión con la base de datos.

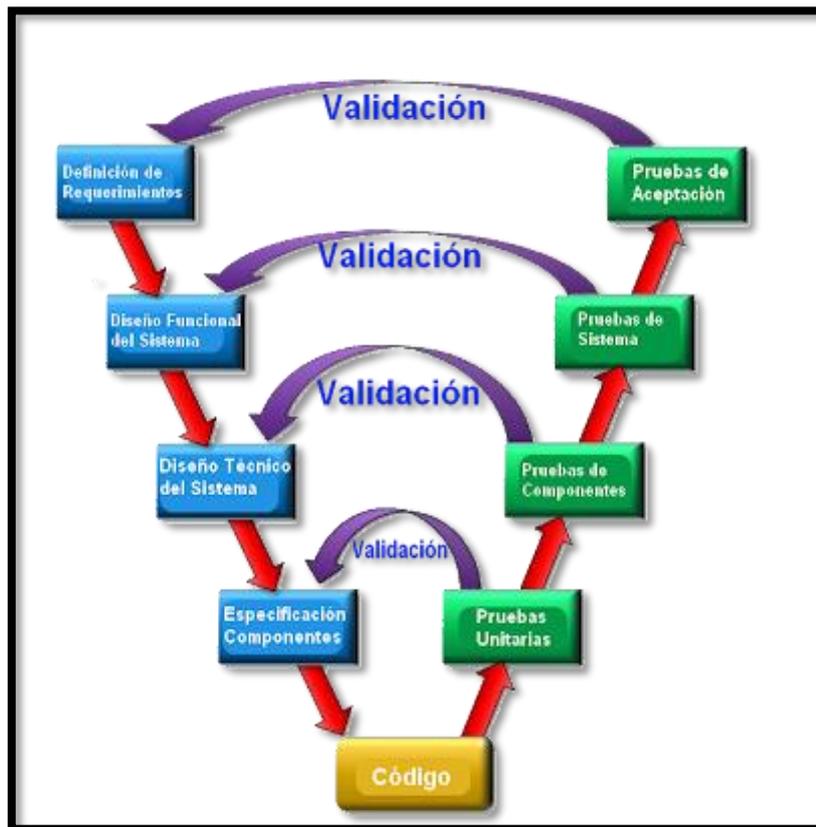
5. El sistema deberá procesar todas las consultas que se soliciten dentro de ella.

6. El sistema funciona en el sistema operativo Windows.

MODELO DE CICLO DE VIDA:

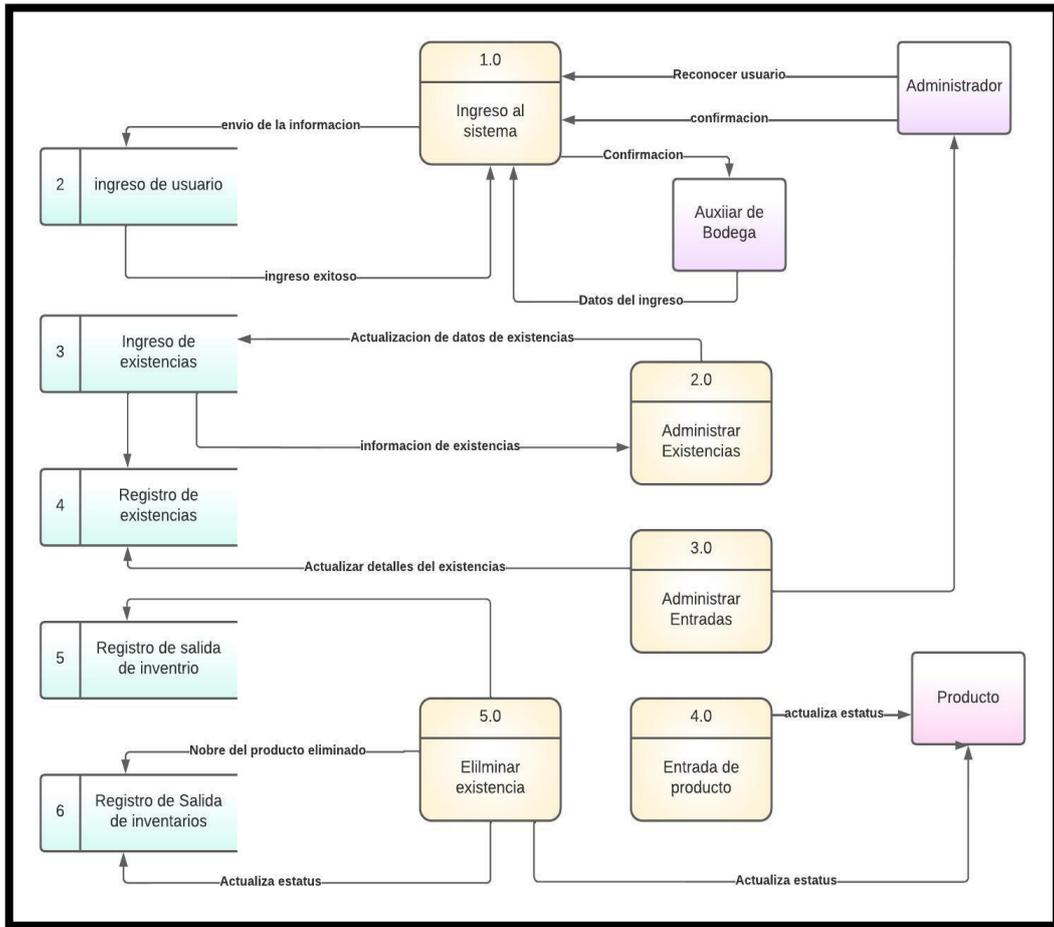
Para el desarrollo de nuestro proyecto usamos el modelo V, el cual nos permite ir actualizando nuestro sistema e ir haciendo pruebas para facilitar la localización de posibles fallos dentro del sistema. Las ventajas que ofrece son:

- La relación entre las etapas de desarrollo y los distintos tipos de pruebas facilitan la localización de fallos.
- Es un modelo sencillo y de fácil aprendizaje.
- Hace explícito parte de la iteración y trabajo que hay que revisar.
- Especifica bien los roles de los distintos tipos de pruebas a realizar.
- Involucra al usuario en las pruebas.



ANÁLISIS:

DIAGRAMA DE FLUJO DE DATOS (DFD)



DIAGRAMAS DE CASOS DE USO

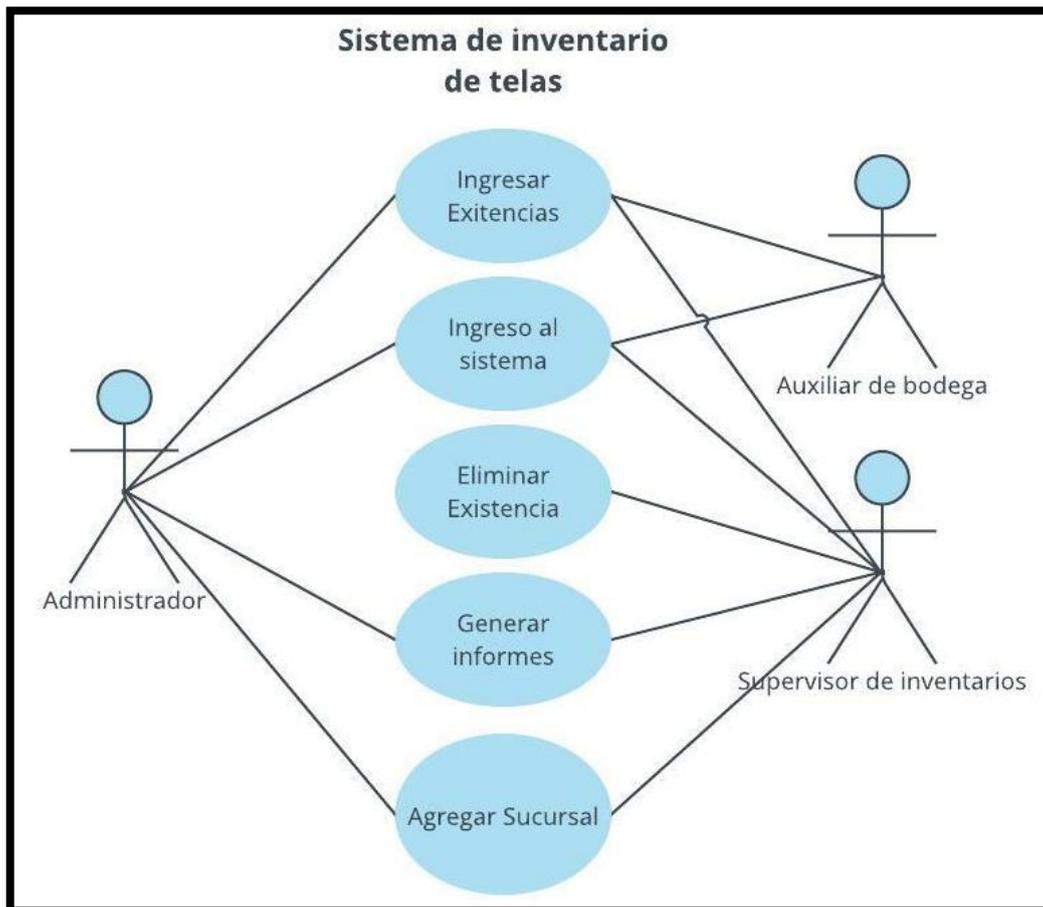


DIAGRAMA DE CLASES

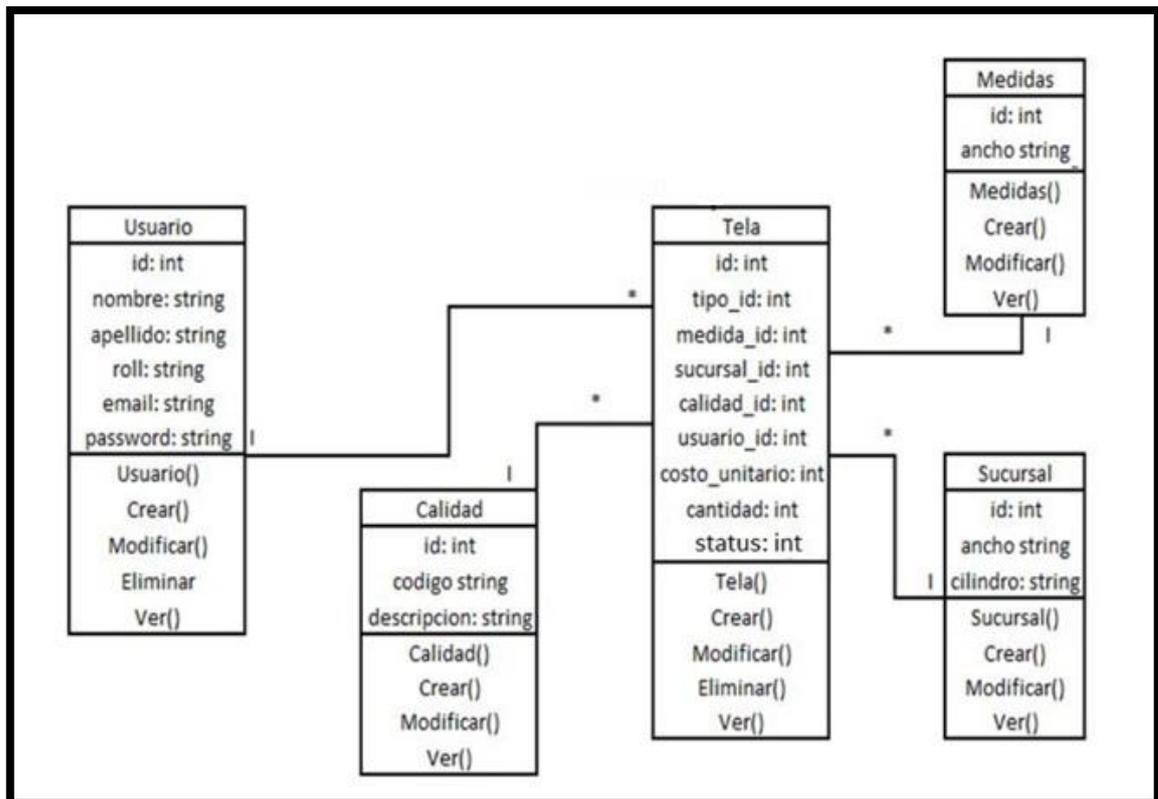
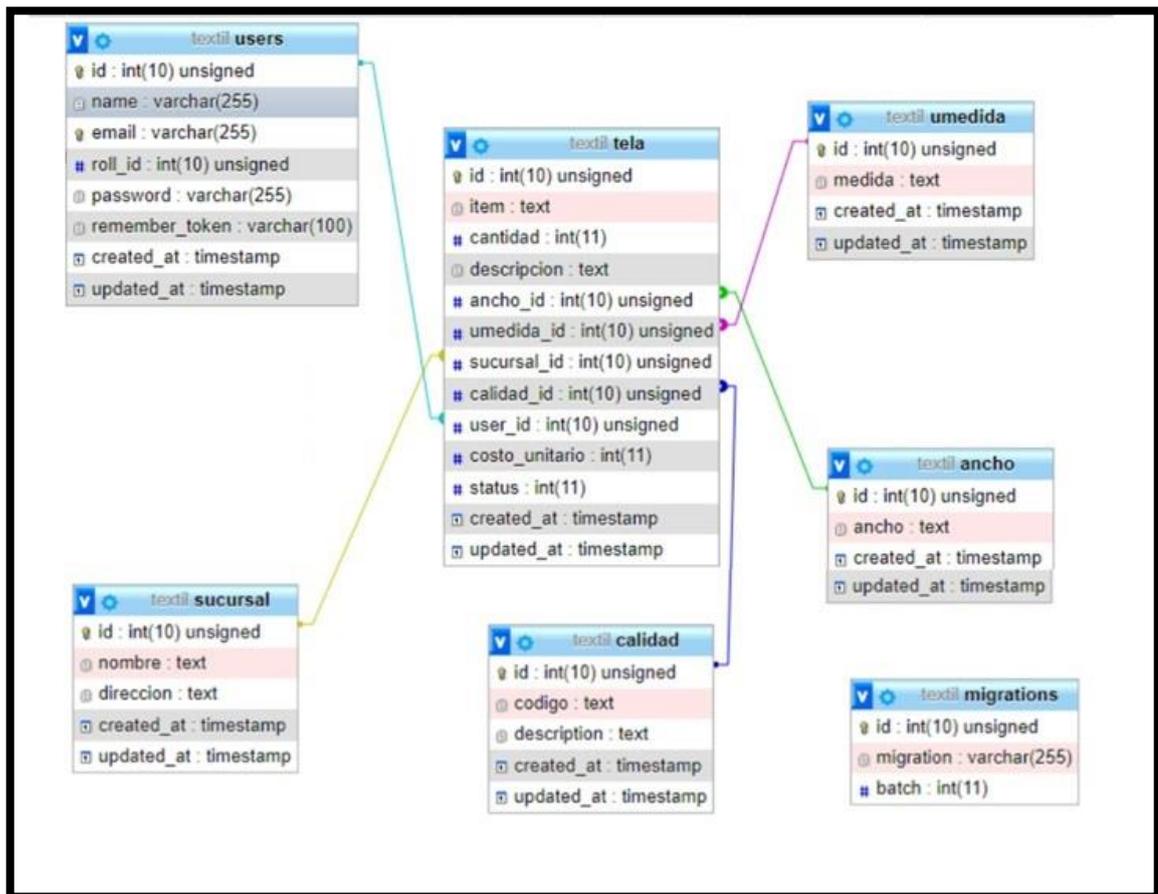
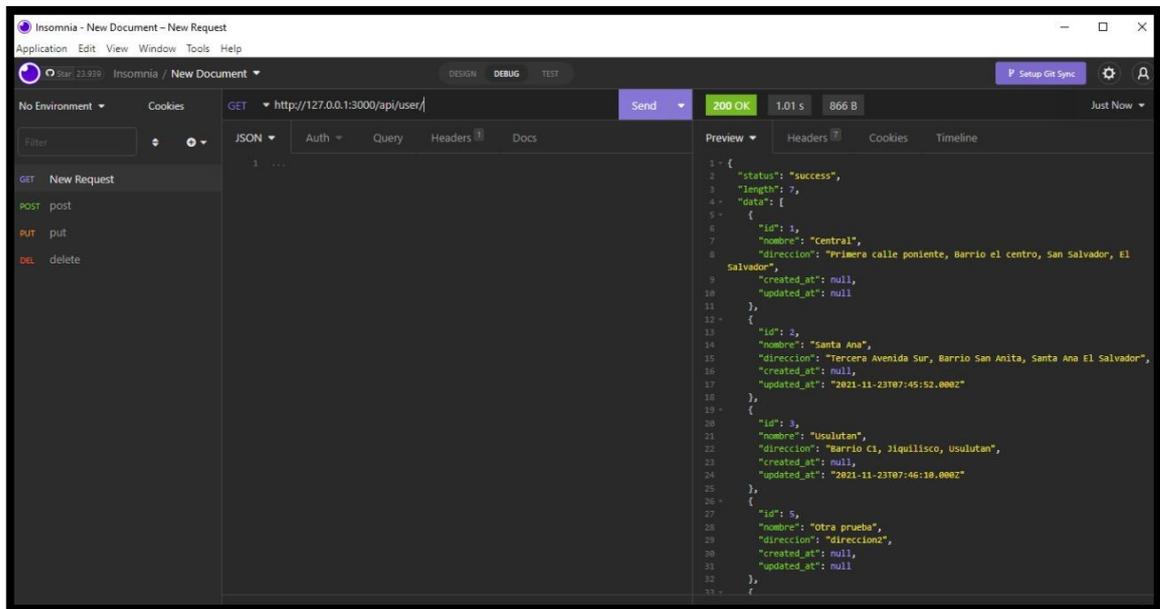


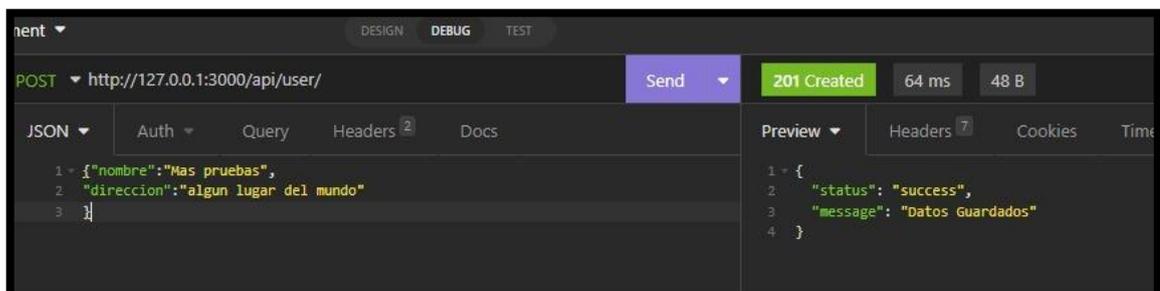
DIAGRAMA DE BASE DE DATOS



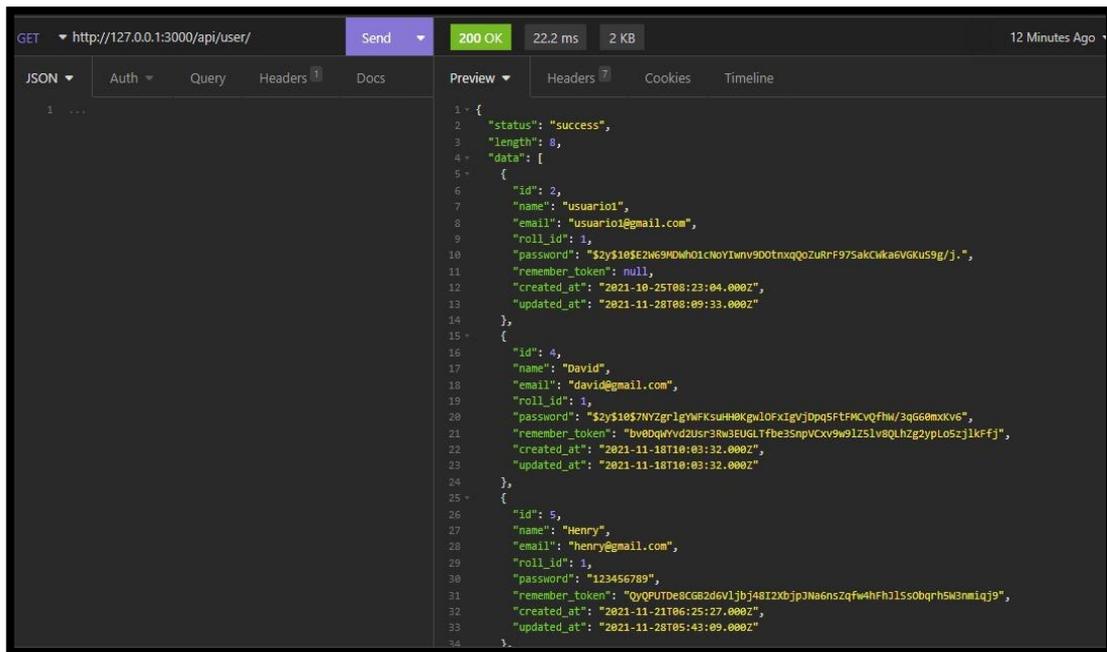
DISEÑO:



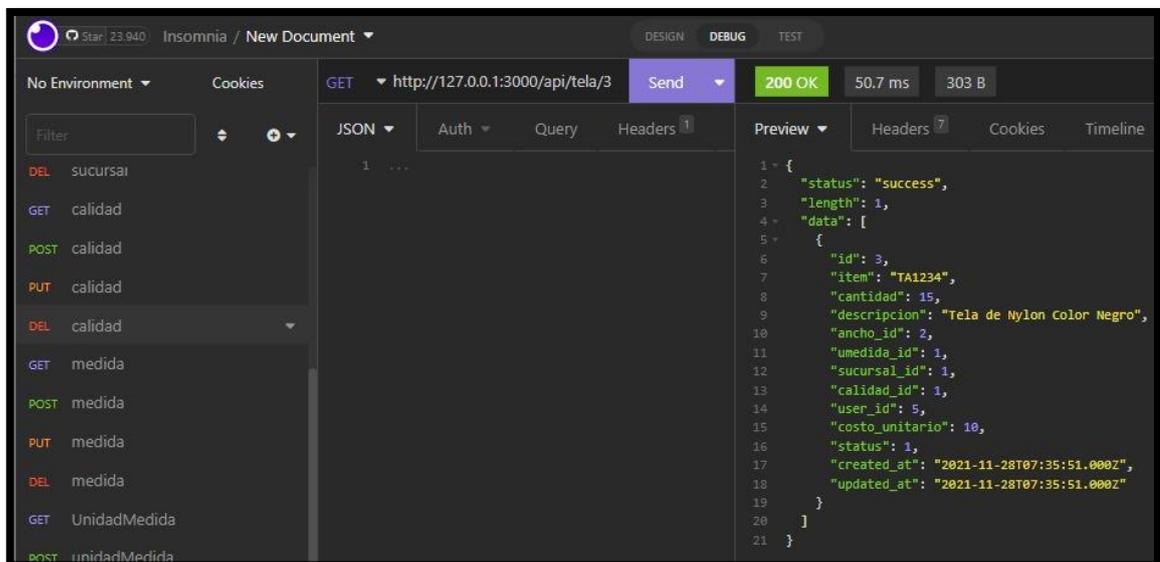
Vista Sucursal: muestra el detalle de todas las sucursales que se han registrado, se puede mostrar todo el registro o únicamente seleccionar uno por Id.



Vista usuario: En la pantalla anterior se muestra el formulario para registrar sucursal, del lado izquierdo se encuentran los datos de entrada para posteriormente guardar el registro



Vista Usuario: En esta pantalla se muestra el detalle de los usuarios registrados en la base de datos, se puede mostrar de manera individual o consultado todos los registros.



Vista Tela: En esta pantalla se visualiza el inventario de tela registrado, se muestran datos como código de tela, cantidad, sucursal, descripción, entre otros.

DESARROLLO

A) TÉCNICAS DE PROGRAMACIÓN APLICADAS

En nuestro Proyecto se emplea un patrón de diseño de software (MVC) para implementar interfaces de usuario, datos y lógica de control. Todo esto enfatizado a la mejor visualización tomando en cuenta que proporciona una buena distribución de trabajo y posterior mantenimiento de la Api.

En el modelo definimos los datos que contiene la aplicación, en vista se muestran los datos y en el controlador contiene la lógica en respuesta a la entrada de los usuarios.

B) HERRAMIENTAS DE DESARROLLO UTILIZADAS

Para el desarrollo del sistema se utilizaron diferentes herramientas, las cuales son:

1. Computadora: Máquina electrónica capaz de almacenar información y tratarla automáticamente mediante operaciones matemáticas y lógicas controladas por programas informáticos.



2. Acceso a internet.

La conexión que nos permite acceder desde nuestra computadora con el objetivo de navegar en busca de información que nos facilite el desarrollo de nuestro proyecto.



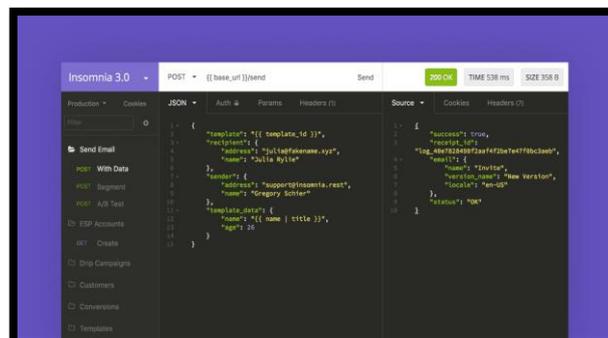
3. Visual Studio Code: Es un editor de código fuente desarrollado por Microsoft para Windows, Linux, macOS y Web. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. La versión utilizada es la última versión estable la 1.62.3 que se dio el 18 de noviembre de 2021.



4. Xampp: Es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor Web Apache y los intérpretes para lenguajes.



5. Insomnia: Es un cliente REST multiplataforma, con una interfaz clara y sencilla. Con funcionalidades que facilitan enormemente el trabajo. Es altamente configurable y personalizable, para que se adapte a nuestras necesidades.



C) LENGUAJE DE PROGRAMACIÓN Y GESTORES DE BASES DE DATOS U OTRAS HERRAMIENTAS UTILIZADAS EN EL DESARROLLO DEL PROYECTO.

1. **Mysql:** Es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base de datos.



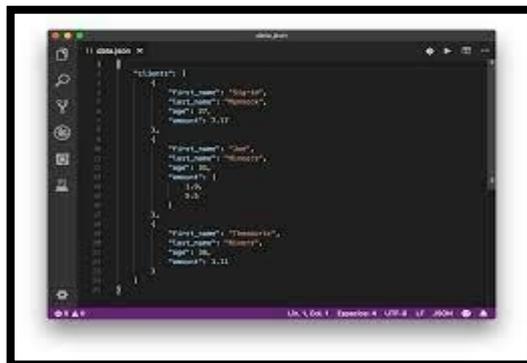
2. **JavaScript:** JavaScript abreviado como Js. Es un lenguaje ligero, interpretado y orientado a objetos con funciones de primera clase, y mejor conocido como el lenguaje de programación para las páginas Web, pero también se utiliza en muchos entornos que no son de navegador. Es un lenguaje de scripts dinámico, multi paradigma, basado en prototipos y admite estilos de programación orientados a objetos, imperativos y funcionales.



3. Node.js: Es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor (pero no limitándose a ello) basado en el lenguaje de programación JavaScript.



4. Package.json: Es un archivo generado automáticamente cuando se instalan paquetes o dependencias en el proyecto. Contiene la información importante sobre el proyecto así como metadatos funcionales(número de versión, dependencias, etc.).



5. Express.js: Es un framework web transigente, escrito en JavaScript y alojado dentro del entorno de ejecución NodeJS.



6. Vue: (vista) es un marco de JavaScript para construir interfaces de usuario. Se basa en HTML, CSS y JavaScript estándar y proporciona un modelo de programación declarativo y basado en componentes que lo ayuda a desarrollar interfaces de usuario de manera eficiente, ya sean simples o complejas.



7. Vuetify: Es una biblioteca de interfaz de usuario de Vue que ofrece componentes de Material Design Framework.



CONCLUSIONES

- -La finalización de dicho proyecto, es un sistema de Inventario de Tela que se adapta básicamente a las necesidades de la empresa ya que podrán tener un mejor manejo o facilidad a la hora de que se realicen los procesos de dicho producto.
- Se asume con seguridad, que la implementación del proyecto, es de suma importancia para el desarrollo tecnológico de la empresa Sky, tomando en cuenta que el viejo modelo con el que cuentan quedaría atrás; tomando así el entorno web como una alternativa viable.
- Con los resultados finales se llegó a realizar un sistema de inventario donde se pueda realizar todas las necesidades requeridas en la empresa, de una manera que permita llevar un mejor gestionamiento entre inventario y sucursales, así mismo también agilizando las actividades de los encargados de administrar el sistema, aprovechando la capacidad de productiva de la herramienta y proporcionando un manual en el que comprenda todo lo requerido.

RECOMENDACIONES

A continuación se presentan una serie de recomendaciones para que las personas beneficiarias de dicho proyecto puedan verificar los beneficios que este brinda:

- Los usuarios pueden hacer uso del sistema de Inventario de Tela ya que este les facilita el trabajo administrativo.
- La principal ventaja que el proyecto ofrece es brindar a los clientes y usuarios un método factible y útil permitiendo interacción entre ellos y el sistema.
- El sistema presentado es fácil manipularlo a la hora gestionarlo y sobre todo es más seguro ya que se trabaja online.
- Se recomienda darle seguimiento y soporte al sistema de inventario para ir innovando ya sea agregando secciones o herramientas que faciliten aún más el trabajo.
- Aprovechar de manera adecuada el uso de la herramienta para mejor desempeño y gestión de la misma.

BIBLIOGRAFÍA:

Wikipedia (05 de octubre de 2020), editor de texto Visual Studio Code, obtenido de:
https://es.wikipedia.org/wiki/Visual_Studio_Code

Álvaro Perdiz (18 de Abril de 2020), Información básica sobre el packet.json, obtenido de: <https://alvaroperdiz.com/javascript/node-is/informaci%C3%B3n-basica-package-json/>

Atareao (20 de agosto de 2018), insomnia, obtenido de:

<https://atareao.es/software/programacion/insomnia-un-productivo-cliente-rest/#:~:text=Insomnia%20es%20una%20aplicaci%C3%B3n%20altamente.Atajos%20de%20teclado>

ANEXOS

MANUAL DE USUARIO:

En el presente manual de usuario se pretende dar una ilustración entendible a los usuarios del sistema de gestión de inventario de textil, para la hora de usarlo puedan hacerlo de la mejor manera y se les facilite la ejecución de cada método para modificar las existencias de telas en bodegas.

OBJETIVOS

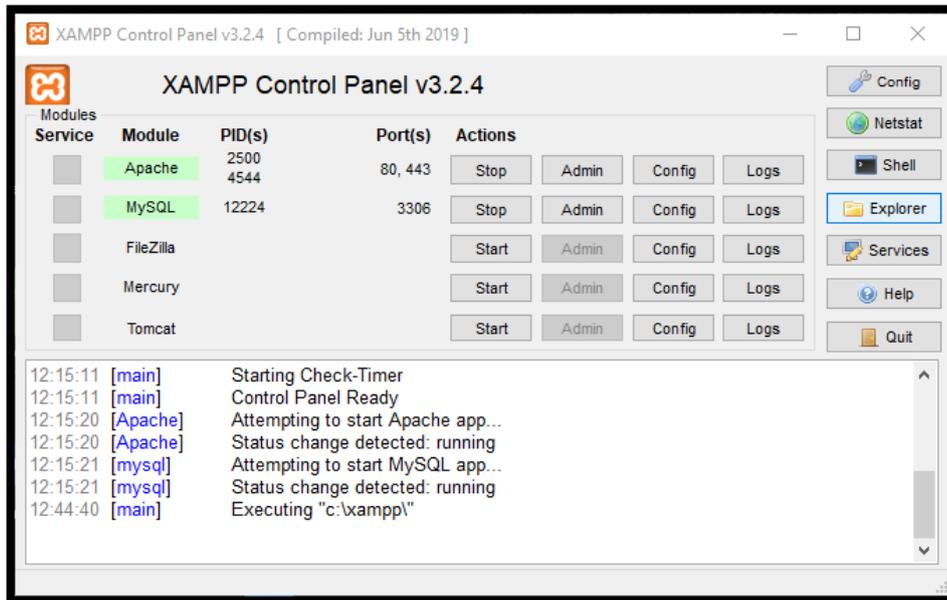
Objetivo general:

Mostrar los pasos que se deben seguir para poder utilizar el sistema de inventario de telas y así aprovechar en su totalidad sus funciones disponibles

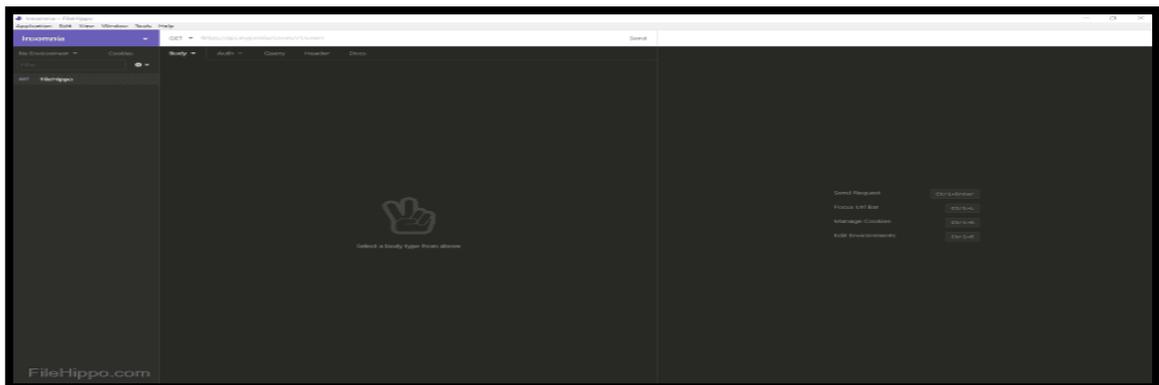
Objetivos específicos:

- Presentar cada una de las herramientas que habilitan la conexión a la base de datos donde se alojara la información
- El desarrollo del manual de usuario proporcionará mayor claridad en la ejecución de los pasos.

Para acceder a la base de datos y poder gestionar el inventario dentro de las bodegas se debe contar con los servicios de Xampp y para eso se debe instalar y configurar previamente

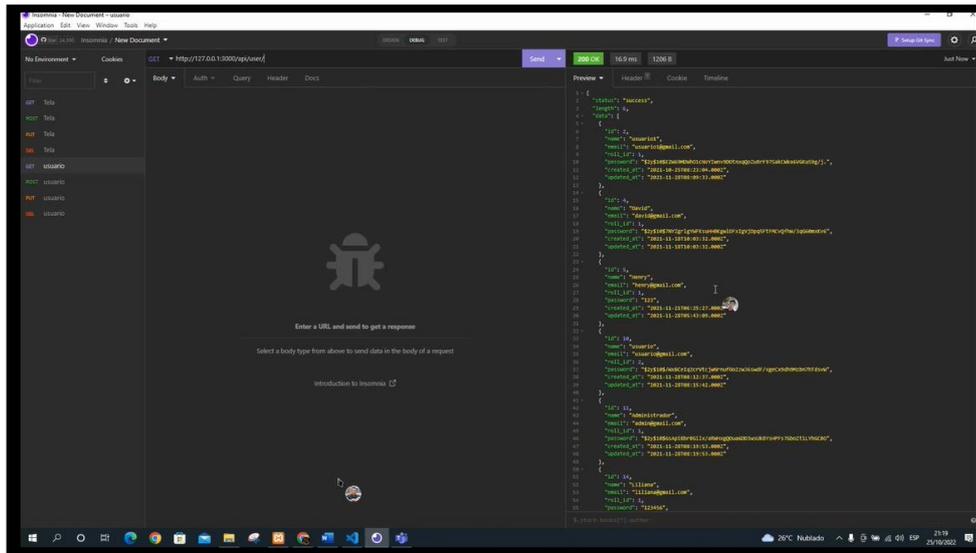


Para poder enviar datos entre la base de datos y el servidor se debe hacer por medio de insomnia, se debe descargar e instalar el programa.

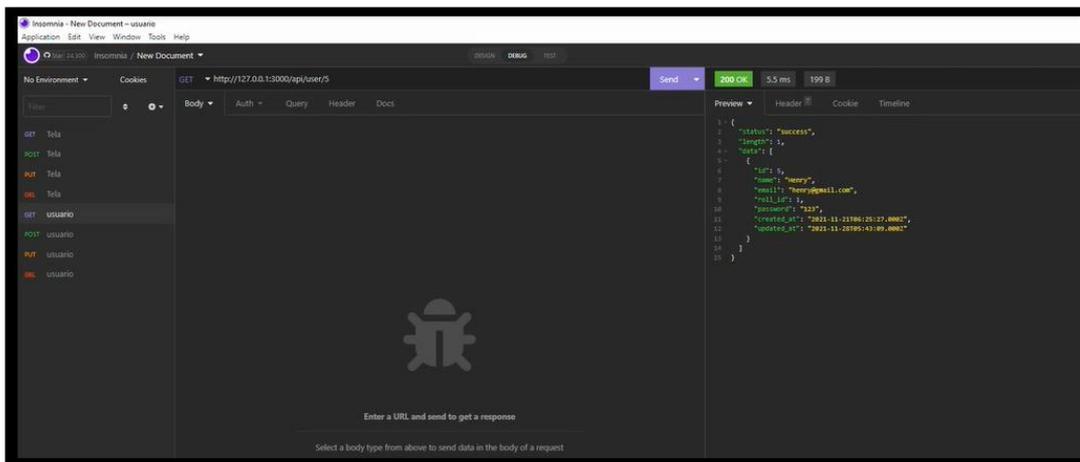


los pasos para poder ingresar existencias son :

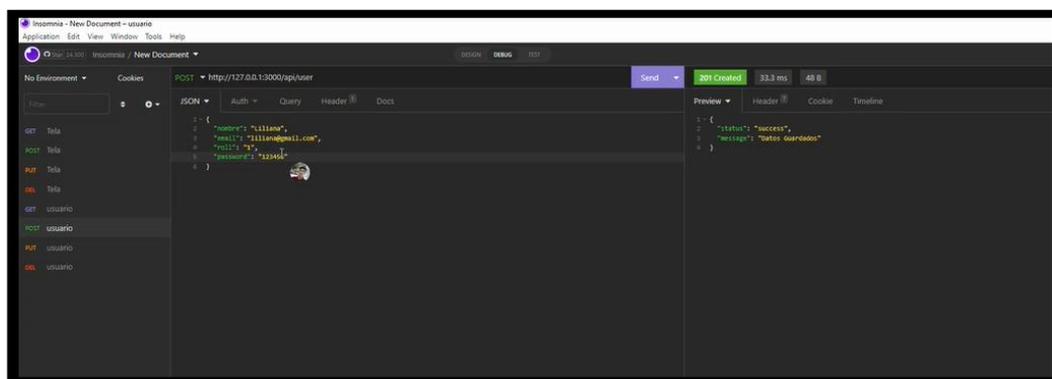
Para ver todos los registros de la tabla se debe poner la ruta de la base de datos sin ID
Con el método GET



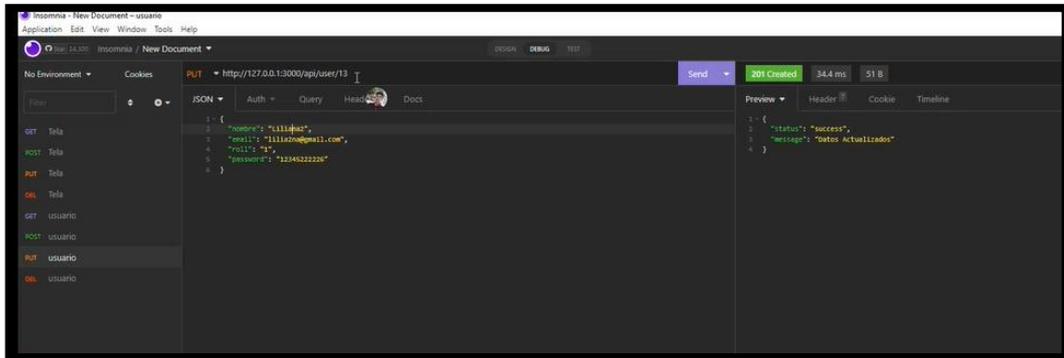
Si se desea recoger los datos de un solo registro se debe colocar el ID del registro **Con el método GET**



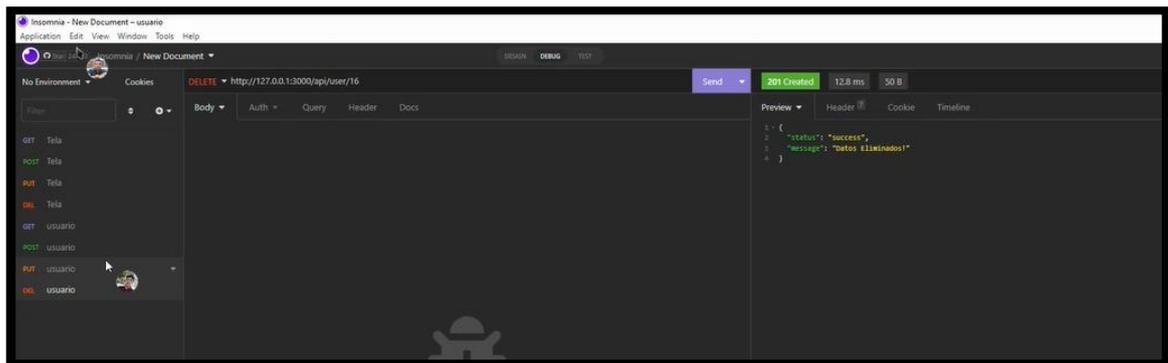
Para ingresar datos se coloca la dirección de la base de datos, especificando la tabla donde se agregará el registro **Con el método POST.**



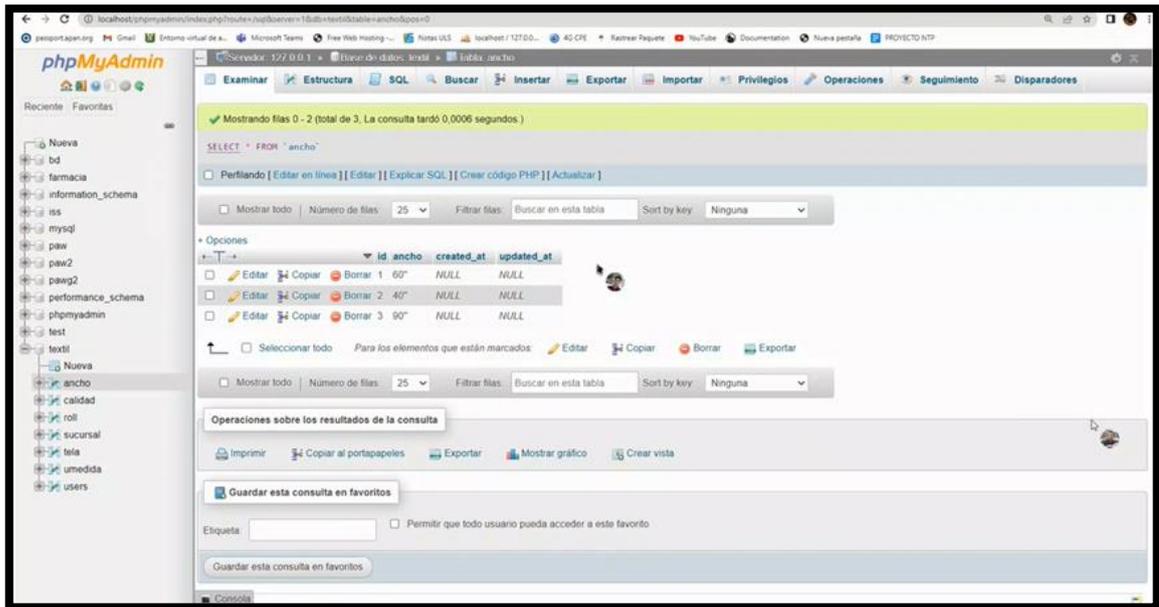
Para modificar los datos de un solo registro se debe colocar el ID del registro que será afectado **Con el método PUT**



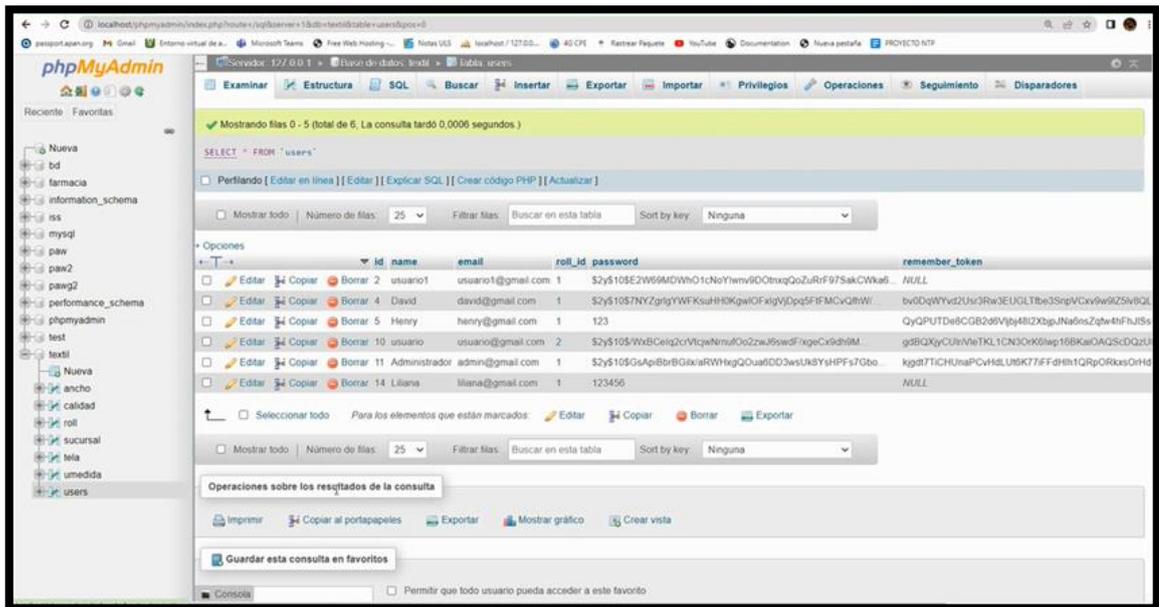
Para eliminar los datos de un solo registro se debe colocar el ID del registro que será afectado **Con el método DELETE**



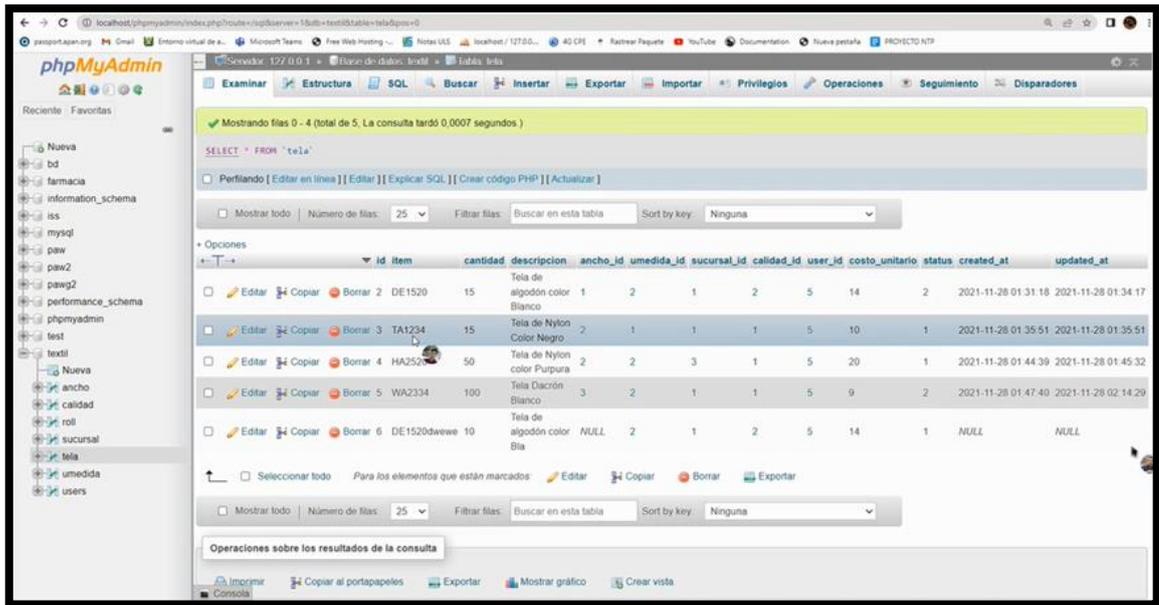
Base de datos tabla ancho: Esta tabla contiene la clasificación de las medidas y serán solo tres clasificaciones las cuales están en pulgadas y estas irán referenciadas en la tabla tela y serán representadas con el ID



Base de datos tabla usuarios: Esta tabla contiene la información relevante y cada ingreso tendrá un ID que será único en cada usuario como requisitos para ingresarlo están el nombre, correo y una contraseña.

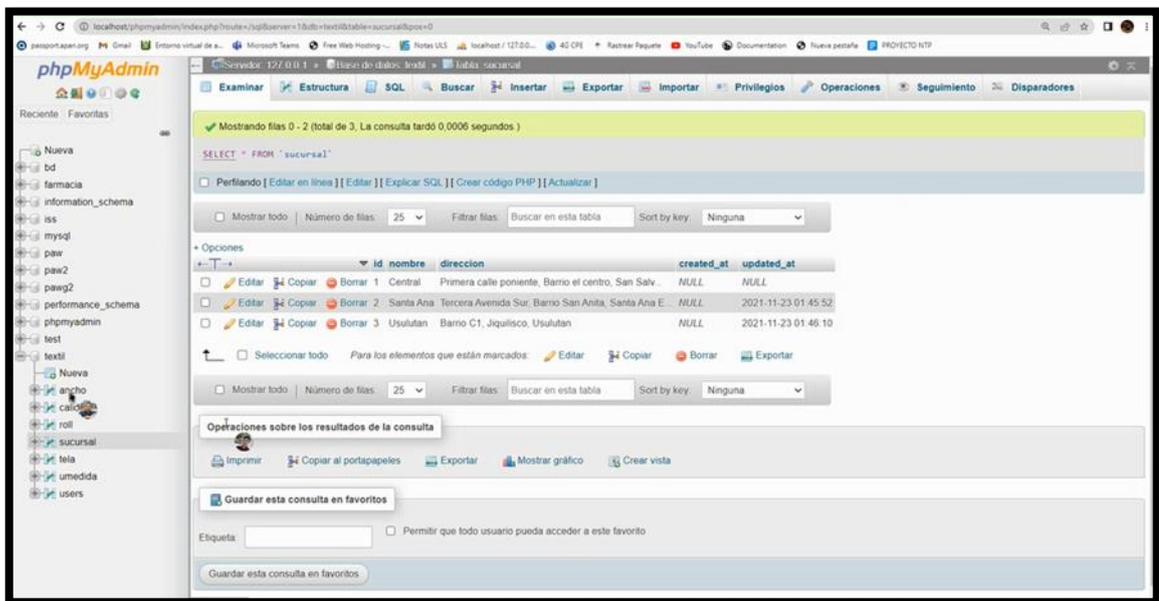


Base de datos tabla tela: En esta tabla se registran las existencias y al momento de registrarlas se ingresan los como cantidad en yardas o peso , descripción o texto explicativo el ancho, sucursal y calidad y automáticamente se asignará dependiendo del usuario que lo ingrese.

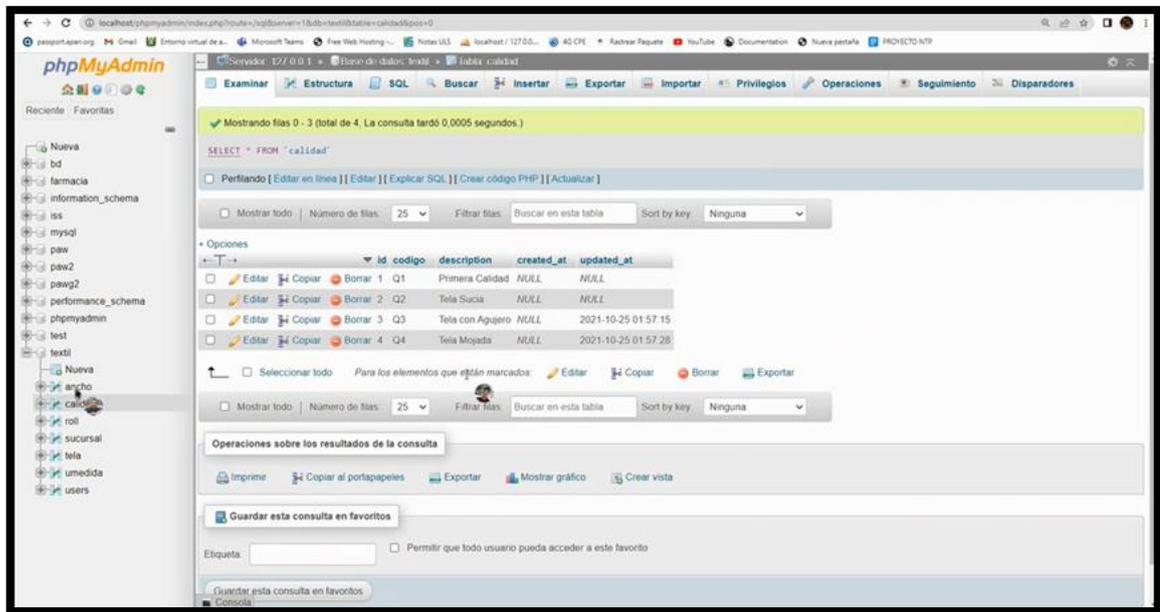


Base de datos tabla sucursal.

En esta opción ya quedan definidas las sucursales que existirán y el usuario administrador podrá agregar más sucursales de ser requerido.

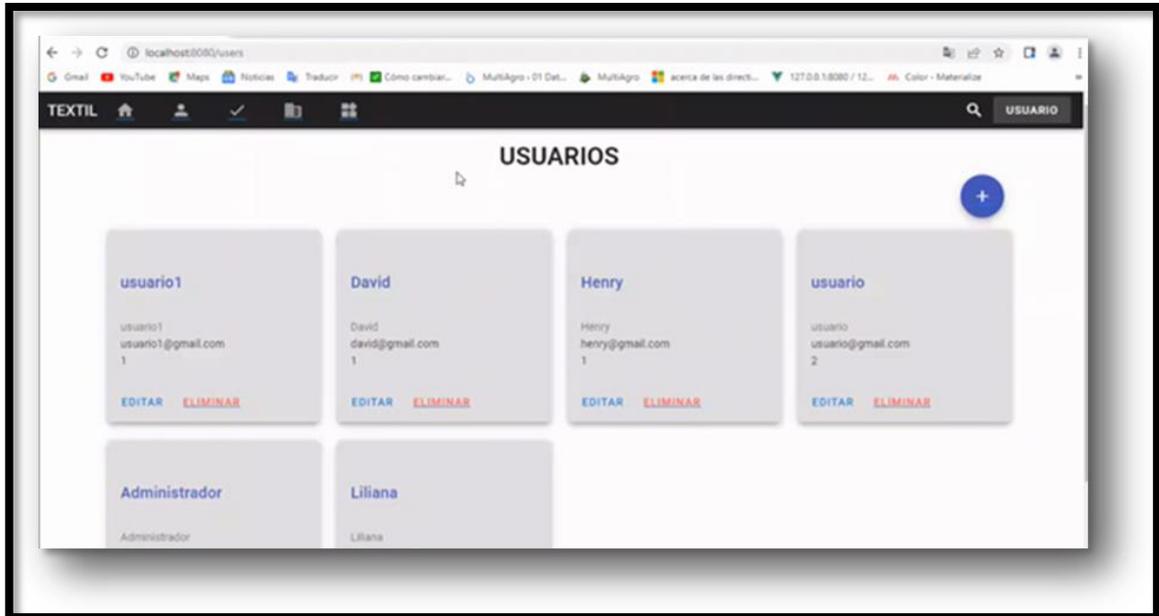


Base de datos tabla calidad: Esta tabla contiene las diferentes clasificaciones de calidad y serán solo tres las cuales están Q1 Primera calidad, Q2 Tela sucia, Q3 Tela con agujeros y Q4 Tela mojada y estas irán referenciadas en la tabla tela y serán representadas con el ID.

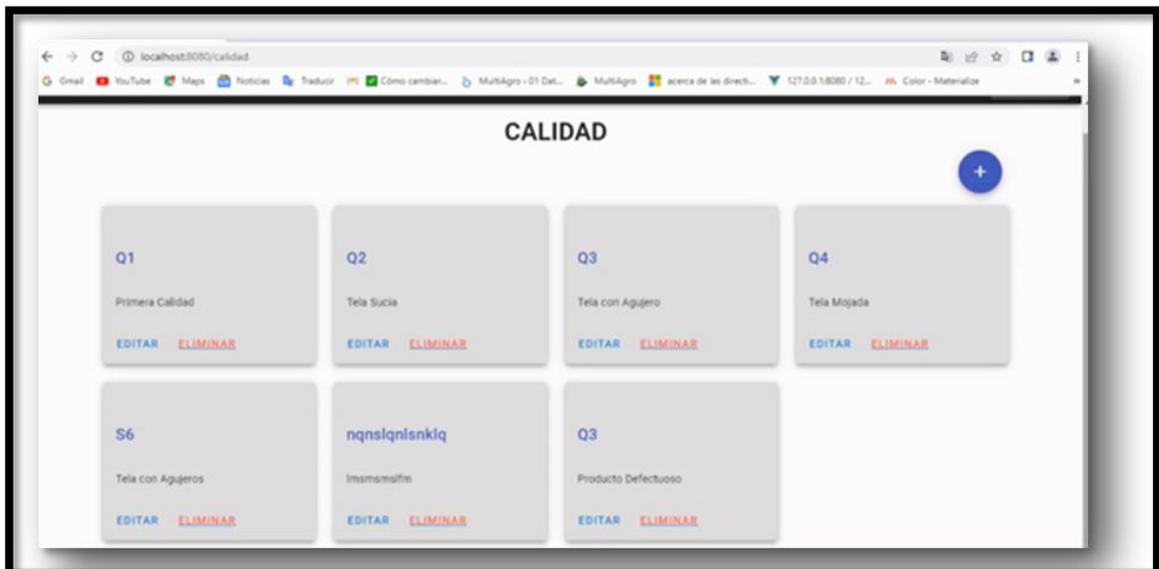


Front End :

Al ingresa con las dos rutas asignadas: - Local: <http://localhost:8080/>
- Network: <http://192.168.1.6:8080/> se podrán visualizar los usuarios registrados y dentro de esta misma vista se puede agregar los usuarios y definir sus roles.



La vista de calidad contiene las distintas clasificaciones de calidad de las telas estas serán definidas previamente para poder asignarlas a los distintos productos que se ingresarán .





localhost:3000/newCalidad

TEXTIL USUARIO

New Calidad

Q3

| I

GUARDAR

Para poder ingresar una nueva clasificación de calidad se seleccionará el botón de + y se completarán los siguientes datos:

- Clave de calidad
- Especificación a evaluar

Si se desea eliminar una clasificación de calidad se presiona en la opción ELIMINAR

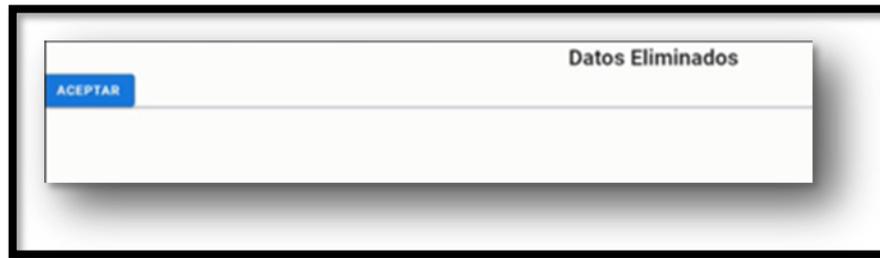
localhost:3000/calidad

CALIDAD

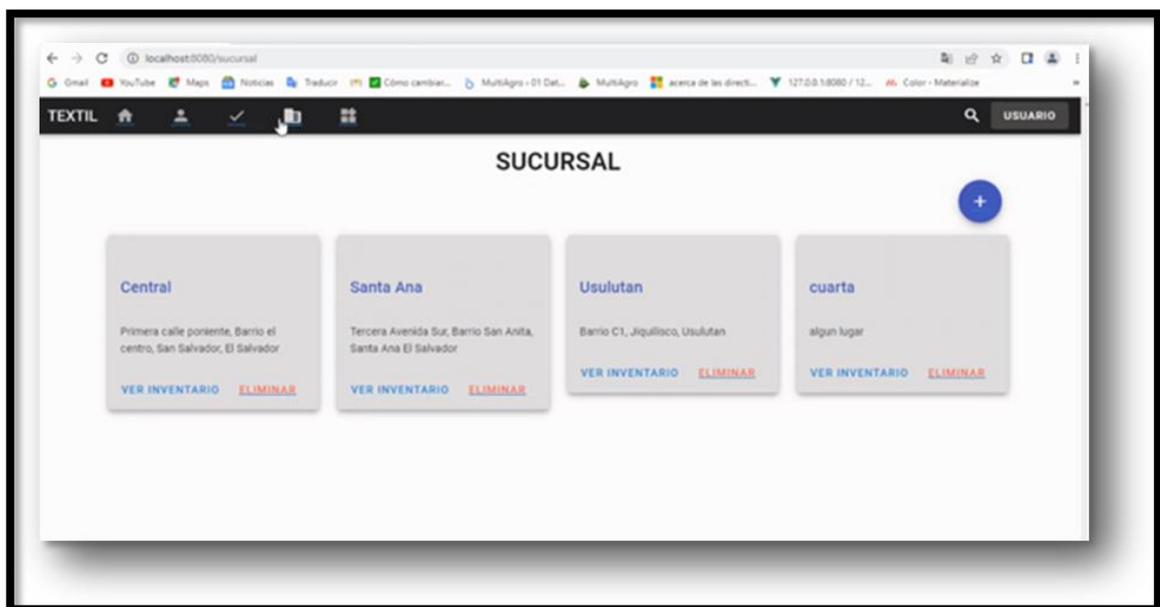
+

| | | | |
|--|---|--|--------------------------------------|
| Q1 Primera Calidad EDITAR ELIMINAR | Q2 Tela Sucia EDITAR ELIMINAR | Q3 Tela con Agujero EDITAR ELIMINAR | Q4 Tela Mojada EDITAR ELIMINAR |
| S6 Tela con Agujeros EDITAR ELIMINAR | nqnsiqnlsnklq Insomamallm EDITAR ELIMINAR | Q3 Producto Defectuoso EDITAR ELIMINAR | |

Aparecerá la siguiente notificación al ser eliminado el registro



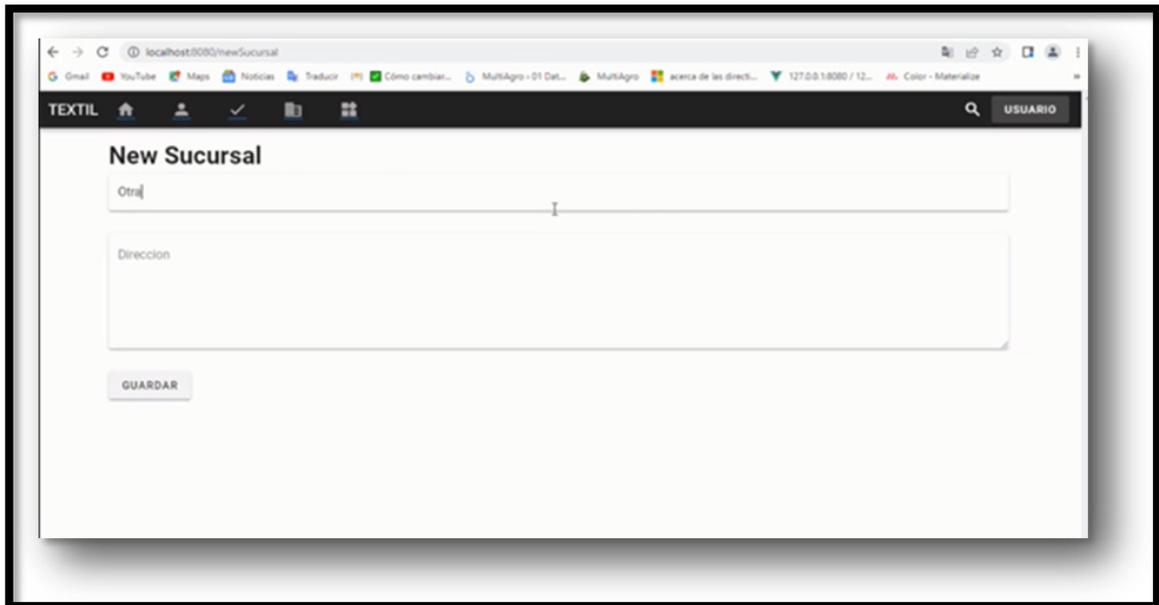
La vista de Sucursal contiene las distintas sucursales estas serán definidas previamente porque desde esa ubicación se ingresarán los inventarios y solo existirán desde el momento que sea ingresada hasta que el inventario finalice.



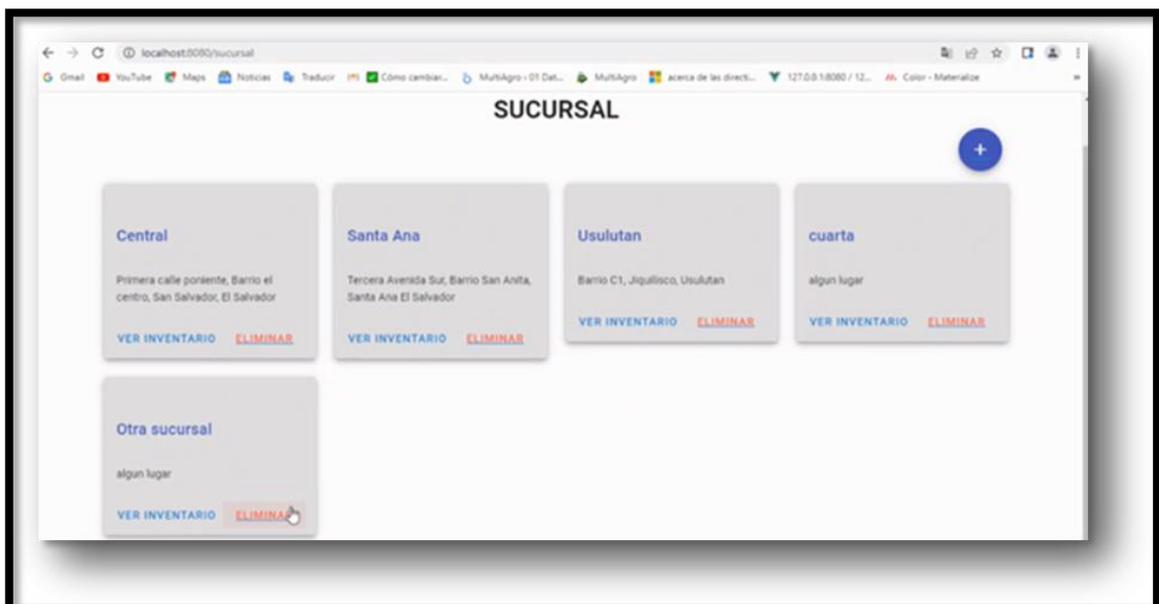
Para poder ingresar una nueva sucursal se seleccionará el botón de + y se completarán los siguientes datos:

Nombre de la sucursal, un nombre general

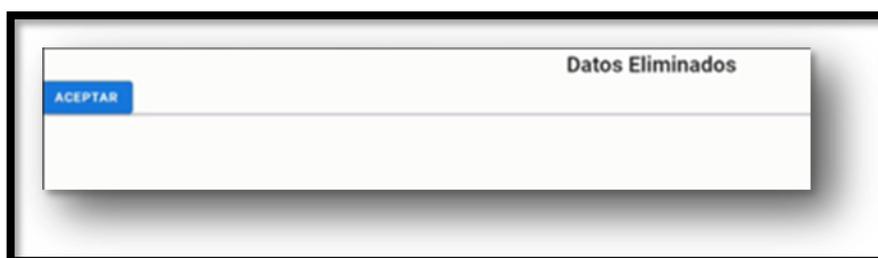
Dirección exacta de la sucursal



Si se desea eliminar el registro se presiona en la opción ELIMINAR



Aparecerá la siguiente notificación al ser eliminado el registro



La vista de Telas contiene los inventarios y solo existirán desde el momento que sea ingresada hasta que el inventario finalice.

The screenshot shows a web browser window with the URL localhost:3000/telas. The page title is "Telas". There is a search bar and a "USUARIO" button in the top right. A blue circular button with a white plus sign is in the top right corner of the table area. The table has the following data:

| Item | Descripcion | Sucursal | Calidad | Unidad de Medida | Costo Unitario | Cantidad | Acciones |
|-------------|------------------------------|----------|---------|------------------|----------------|----------|-----------------|
| DE1520 | Tela de algodón color Blanco | 1 | 2 | 2 | 14 | 15 | [Edit] [Delete] |
| TA1234 | Tela de Nylon Color Negro | 1 | 1 | 1 | 10 | 15 | [Edit] [Delete] |
| HA2520 | Tela de Nylon color Purpura | 3 | 1 | 2 | 20 | 50 | [Edit] [Delete] |
| WA2334 | Tela Dacrón Blanco | 1 | 1 | 2 | 9 | 100 | [Edit] [Delete] |
| DE1520dewew | Tela de algodón color Bla | 1 | 2 | 2 | 14 | 10 | [Edit] [Delete] |
| www | 333 | 1 | 1 | 1 | 2 | 2 | [Edit] [Delete] |
| sdids | sdids | 1 | 1 | 1 | 2 | 12 | [Edit] [Delete] |
| hdhhdhdhdh | jkckckckkbcbs | 1 | 1 | 1 | 345 | 1 | [Edit] [Delete] |



Para poder ingresar un nuevo inventario y una existencia se seleccionará el botón de + y se completarán los siguientes datos:

- ITEM
- CANTIDAD
- DESCRIPCIÓN
- SUCURSAL
- ANCHO
- UNIDAD DE MEDIDA
- CALIDAD

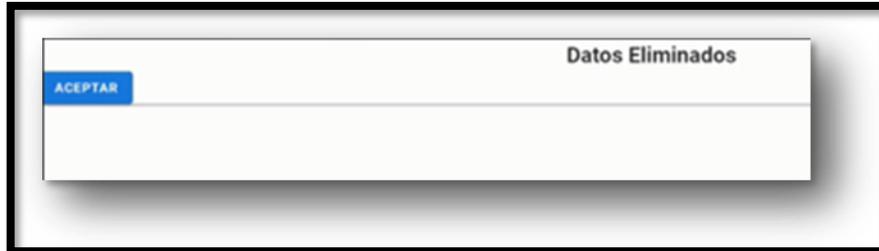
The screenshot shows a web browser window with the URL localhost:3000/newTela. The page title is "New Tela". The form has the following fields:

- Item:
- Cantidad:
- Descripcion:
- Sucursal:
- Ancho:
- Unidad de Medida:
- Calidad:

Si se desea eliminar una existencia o un ingreso se presiona el siguiente icono



Aparecerá la siguiente notificación al ser eliminado el registro



MANUAL DE DESARROLLADOR

INTRODUCCIÓN

Hoy en día las empresas tienen una mayor participación en el mercado, los sistemas de inventario se han convertido en un elemento crítico e importante en el crecimiento de las empresas o negocios; dicho sistema ha sido diseñado con el propósito de llevar el control interno de los movimientos u operaciones. Elegir un software y hardware apropiado para una empresa puede ofrecer muchos beneficios tales como llevar un control y encontrar rentabilidad; por lo que es importante no esperar para explorar las opciones valiosas. Dado los objetivos propuestos para el funcionamiento del sistema, este se convierte en una herramienta de fácil acceso para sus usuarios.

OBJETIVOS

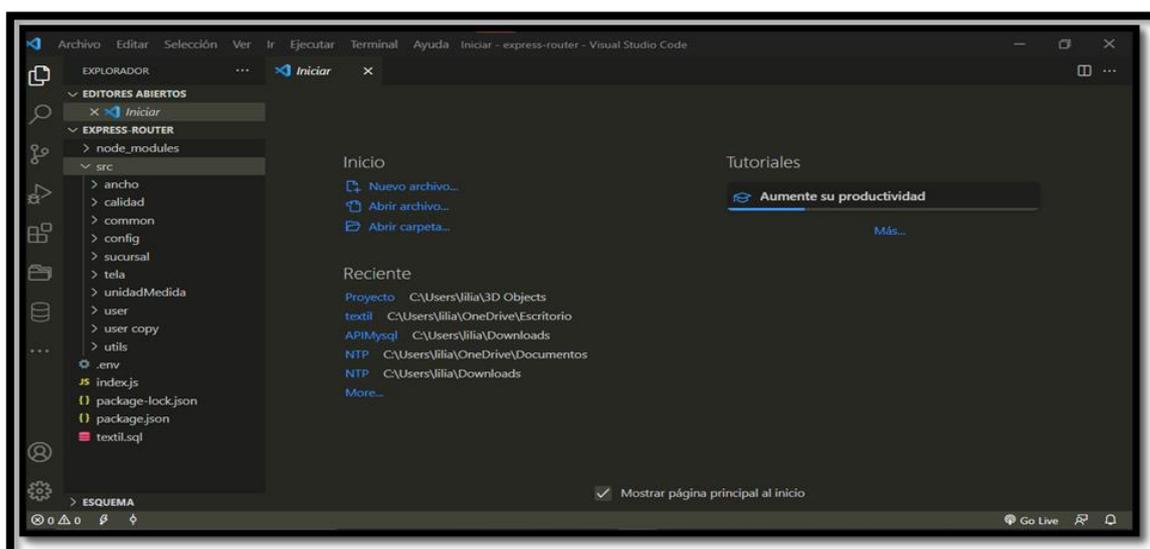
Objetivo general: Mostrar las diferentes instalaciones y configuraciones de librerías que fueron realizadas para el correcto funcionamiento del sistema de Inventario.

Objetivos específicos:

- Presentar cada una de las librerías instaladas para el sistema de inventario de tela.
- El desarrollo del manual de desarrollador, instruirá en el uso adecuado de las configuraciones del sistema de inventario, para cualquier actualización a futuro, también para el acceso dentro de él.

Manual de desarrollador Back End

1. Creación de la estructura de los archivos para el diseño del sistema de inventario de telas. Se encuentra una carpeta general llamada express-router y dentro de esta se encuentran alojadas otras carpetas como: ancho, calidad, common, config, sucursal, tela, unidad de medida, user. En cada una de ellas, se encuentran todos los archivos o los códigos fuentes que han sido creados para realizar dicho proyecto.



2. El comando `npm init -y` se utiliza para instalar paquetes y para inicializar el proyecto. Al ejecutar este comando se crea un archivo `package.json`, el cual incluye el nombre del proyecto, el tipo de licencia, la versión utilizada.

```
PS C:\Users\lilia\Music\express-router> npm init -y
Wrote to C:\Users\lilia\Music\express-router\package.json:

{
  "name": "express-router",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "dependencies": {
    "abbrev": "^1.1.1",
    "base64-js": "^1.5.1",
    "anymatch": "^3.1.2",
    "accepts": "^1.3.8",
    "better-js-class": "^0.1.3",
    "balanced-match": "^1.0.2",
    "basic-auth": "^2.0.1",
    "array-flatten": "^1.1.1",
    "brace-expansion": "^1.1.11",
    "body-parser": "^1.20.0",
    "bluebird": "^3.7.2",
    "bignumber.js": "^9.0.0",
    "binary-extensions": "^2.2.0",
    "bson": "^4.7.0",
    "buffer": "^5.7.1",
```

3. El comando `npm install node-mysql` es un controlador que se utiliza junto con `node.js`. Esta librería se utiliza para realizar la conexión a la base de datos.

```
PS C:\Users\lilia\Music\express-router> npm install node-mysql
changed 1 package, and audited 136 packages in 2s

13 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Users\lilia\Music\express-router> █
```

4. El comando `npm install mysql` es utilizado para la descargar la librería `mysql`.

```
PS C:\Users\lilia\Music\express-router> npm install mysql
changed 1 package, and audited 136 packages in 11s

13 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Users\lilia\Music\express-router> █
```

5. El comando `npm install express`, es una función de `middleware` integrada en `Express`. Analiza las solicitudes entrantes con cargas `JSON` y se basa en `body-parser`. También se utiliza para levantar los servicios de `node`.

```
PS C:\Users\lilia\Music\express-router> npm install express
changed 3 packages, and audited 136 packages in 1s

13 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Users\lilia\Music\express-router> █
```

6. El comando `npm install http-errors` es una librería utilizada para mostrar el error que se esté generando en el momento.

```
PS C:\Users\lilia\Music\express-router> npm install http-errors
changed 1 package, and audited 136 packages in 913ms

13 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Users\lilia\Music\express-router> █
```

7. El archivo `package.json` aloja todas las librerías descargadas que se han utilizado para el sistema de inventario al igual que sus respectivas versiones.

```
package.json X
package.json > {} dependencies > node-mysql
1  {
2  "name": "express-router",
3  "version": "1.0.0",
4  "description": "",
5  "main": "index.js",
6  "type": "module",
7  Depuración de >
8  "scripts": {
9    "test": "echo \"Error: no test specified\" && exit 1",
10   "dev": "nodemon index.js"
11 },
12 "keywords": [],
13 "author": "",
14 "license": "ISC",
15 "dependencies": {
16   "cors": "^2.8.5",
17   "dotenv": "^16.0.2",
18   "express": "^4.18.1",
19   "http-errors": "^2.0.0",
20   "morgan": "^1.10.0",
21   "mysql": "^2.18.1",
22   "node-mysql": "^0.4.2",
23   "nodejs-mysql": "^0.1.3"
24 },
25 "devDependencies": {
  "nodemon": "^2.0.19"
```

8. El archivo index.js es el archivo principal. Ya que contiene las importaciones de cada una de las carpetas utilizadas para crear el sistema de inventario.

```
JS index.js x
JS index.js > ...
1  import express from "express";
2
3  ///importaciones de los routers
4  import {apiUser} from './src/user/index.js';
5  import {apiSucursal} from './src/sucursal/index.js';
6  import {apiCalidad} from './src/calidad/index.js';
7  import {apiMedida} from './src/ancho/index.js';
8  import {apiUnidadMedida} from './src/unidadMedida/index.js';
9  import {apiTela} from './src/tela/index.js';
10 // Importacion del archivo
11 //import { Config } from './src/config/config.js';
12
13
14 const app = express();
15
16 // configuracion de puerto
17 //const puerto = Config.port ? Config.port : 4000;
18
19
20 // middleware
21 app.use(express.json());
22
23
24 // uso de rutas
25 apiUser(app);
26 apiSucursal(app);
```

```
JS index.js x
JS index.js > ...
22
23
24 // uso de rutas
25 apiUser(app);
26 apiSucursal(app);
27 apiCalidad(app);
28 apiMedida(app);
29 apiUnidadMedida(app);
30 apiTela(app);
31
32
33 // Ejecutar el servicio
34 app.listen(3000, () => {
35   console.log("Server is running***");
36 })
37
```

9. Conexión a base de datos con mysql , se realiza la configuración para la base de datos llamada textil y que ésta reciba los datos que se le ingresen. De igual manera se le pasa el parámetro de error para advertir si existe una mala conexión o si esta es exitosa.

```
JS db.js x
src > config > JS db.js > ...
1 //import { config } from './config.js';
2
3 import { createConnection } from 'mysql';
4 const connection = createConnection({
5   host: 'localhost',
6   user: 'root',
7   password: '',
8   database: 'textil'
9 })
10
11 connection.connect(
12   (err)=>{
13     if (!err) {
14       console.log("Conexion Establecida");
15     } else {
16       console.log("Conexion Fallida");
17     }
18   }
19 )
20
21
22 //connection.end()
23
24 //exports.default = connection
25
26 export {connection};
```

10. El archivo response.js contiene el constructor de error, este código indica cuando ha ocurrido una falla en el proceso de carga de archivos.

```
JS db.js JS Response.js x
src > common > JS Response.js > ...
1 import createError from "http-errors";
2
3 const Response = {
4   success:(res, status = 200, message= "ok", body={})=>{
5     res.status(status).json({message, body});
6   },
7   error:(res,error=null)=>{
8     const {statusCode, message}=error
9     ? error
10    :new createError.InternalServerError();
11    res.status(statusCode).json({message});
12  }
13 };
14 export {Response };
15
```

11. Configuración de la base de datos. A través del servidor phpmyadmin se crea la base de datos con el nombre textil. Dentro de ella, se establecen las tablas a utilizar:

Ancho: guarda los distintos tipos de ancho que puede medir la tela.

Calidad: contiene cuatro clasificaciones: primera calidad, sucia, agujero y mojada.

Roll: se definen que función está desempeñando si es administrador o usuario.

Sucursal: contiene las sucursales que han sido establecidas en este caso son 3: central, Santa Ana y Usulután.

Tela: Es la que contiene la información general de las demás tablas.

Umedida: esta tabla contiene la unidad de medida pueden ser en libras o yardas.

Users: se alojan los nombres de los usuarios que han sido registrados en la base de datos y que tipo de función desempeña si es usuario o administrador.

| Tabla | Acción | Filas | Tipo | Cotejamiento | Tamaño | Residuo a depurar | Juego de caracteres | Comentario | Creación | Última actualización | Última revisión |
|-----------------------------------|--|-------|--------|--------------------|--------------------|-------------------|---------------------|------------|---------------------------|----------------------|-----------------|
| <input type="checkbox"/> ancho | Examinar Estructura Buscar Insertar Vaciar Eliminar | 3 | InnoDB | utf8mb4_unicode_ci | 16,0 KB | - | utf8mb4 | | 25-10-2022 a las 20:32:33 | - | - |
| <input type="checkbox"/> calidad | Examinar Estructura Buscar Insertar Vaciar Eliminar | 4 | InnoDB | utf8mb4_unicode_ci | 16,0 KB | - | utf8mb4 | | 25-10-2022 a las 20:32:32 | - | - |
| <input type="checkbox"/> roll | Examinar Estructura Buscar Insertar Vaciar Eliminar | 2 | InnoDB | utf8mb4_unicode_ci | 16,0 KB | - | utf8mb4 | | 25-10-2022 a las 20:32:32 | - | - |
| <input type="checkbox"/> sucursal | Examinar Estructura Buscar Insertar Vaciar Eliminar | 3 | InnoDB | utf8mb4_unicode_ci | 16,0 KB | - | utf8mb4 | | 25-10-2022 a las 20:32:32 | - | - |
| <input type="checkbox"/> tela | Examinar Estructura Buscar Insertar Vaciar Eliminar | 4 | InnoDB | utf8mb4_unicode_ci | 96,0 KB | - | utf8mb4 | | 25-10-2022 a las 20:32:32 | - | - |
| <input type="checkbox"/> umedida | Examinar Estructura Buscar Insertar Vaciar Eliminar | 2 | InnoDB | utf8mb4_unicode_ci | 16,0 KB | - | utf8mb4 | | 25-10-2022 a las 20:32:32 | - | - |
| <input type="checkbox"/> users | Examinar Estructura Buscar Insertar Vaciar Eliminar | 5 | InnoDB | utf8mb4_unicode_ci | 48,0 KB | - | utf8mb4 | | 25-10-2022 a las 20:53:11 | - | - |
| 7 tabla(s) Número de filas | | | 23 | InnoDB | utf8mb4_general_ci | 224 KB | 0 | utf8mb4 | 25-10-2022 a las 20:32:33 | - | - |

Front End

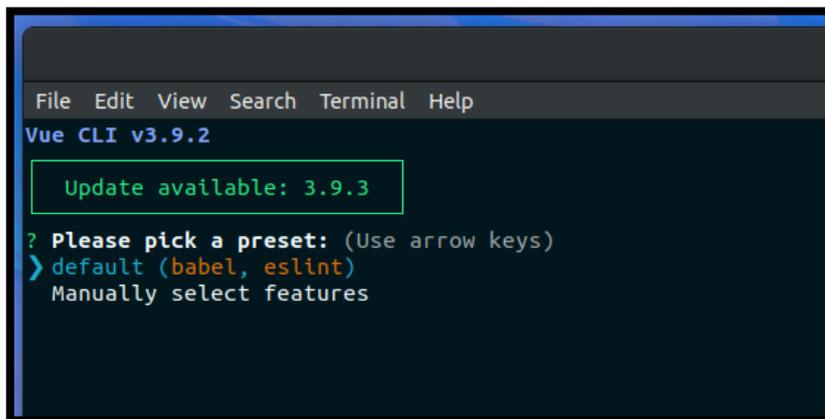
12. Instalación de VUE CLI:

Es una herramienta para línea de comandos que sirve para gestionar proyectos Vue. Para usarla se debe descargar en el equipo desde su paquete NPM con el siguiente comando:

```
npm install -g @vue/cli
```

Para que el vue cli se encargue de crear toda la estructura inicial del proyecto ejecutar:
vue create nombre-de-tu-proyecto

Cuando se ejecuta este comando, la terminal preguntará por la elección de las opciones por defecto o personalizado



```
File Edit View Search Terminal Help
Vue CLI v3.9.2
Update available: 3.9.3
? Please pick a preset: (Use arrow keys)
> default (babel, eslint)
   Manually select features
```

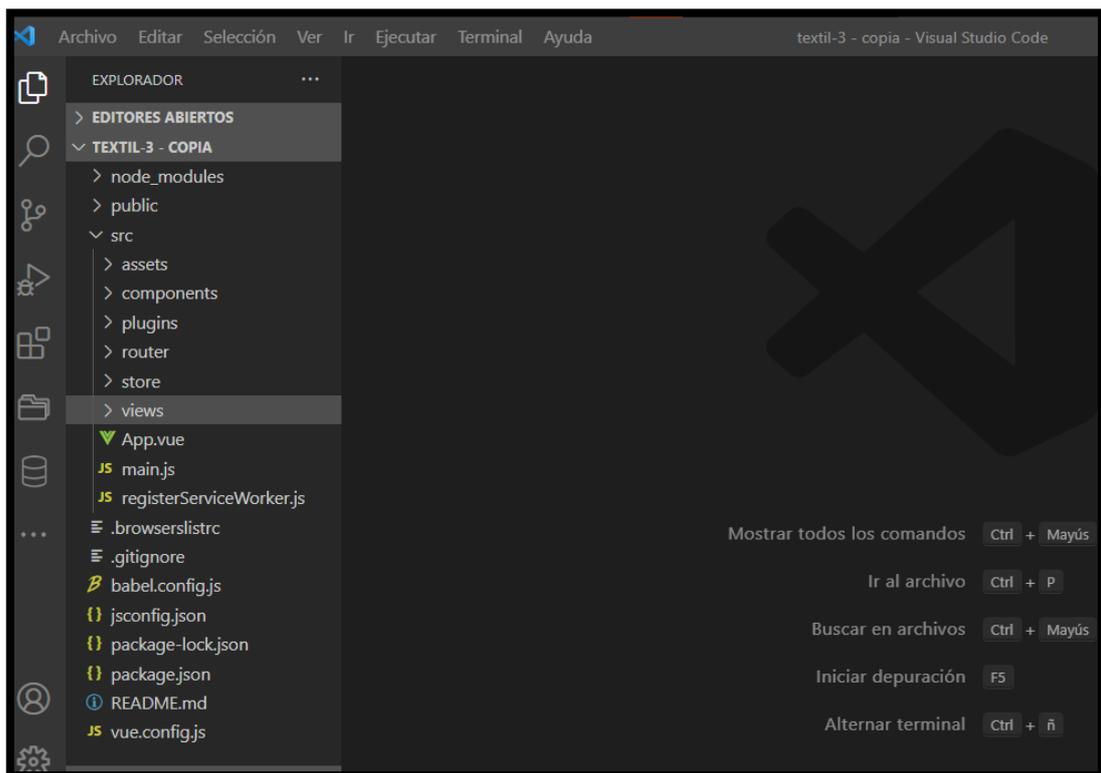
En cuanto se elija la opción de preferencia se creará el proyecto
Importando la librería de Vue

La otra forma de hacerlo consiste en usar directamente el JavaScript compilado de Vue de forma local o mediante CDN:

```
<script src="https://cdn.jsdelivr.net/npm/vue@2.6.10/dist/vue.js"></script>
```

Una vez preparado todo se ejecuta el proyecto en una terminal de visual studio code y luego al navegador con la IP generada para el proyecto.

La creación de la estructura de los archivos para el diseño (Front End) del sistema de inventario de telas .Se encuentra una carpeta general llamada Textil-3-copia y dentro de esta se encuentran alojadas otras carpetas como: node_modules, public, src, components, router, views. En cada una de ellas, se encuentran todos los archivos o los códigos fuentes que han sido creados para realizar dicho proyecto.



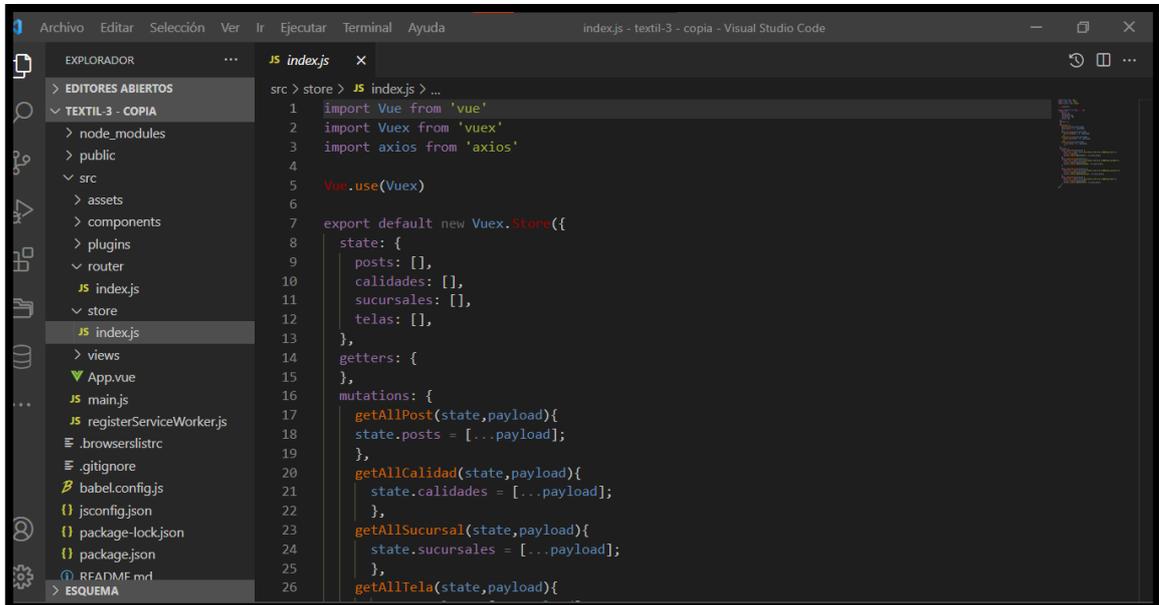
13. En la carpeta components se encuentra un archivo llamado Menu.vue, este archivo contiene el menú de la aplicación con cada uno de los nombres de los módulos.

```
1 <template>
2   <div>
3     <v-app-bar
4       color="#212121"
5       dense
6       dark
7     >
8     <v-toolbar-title>TEXTIL</v-toolbar-title>
9     <router-link to="/" >
10      <v-btn class="ma-1" plain> <v-icon >mdi-home</v-icon> </v-btn>
11    </router-link>
12    <router-link to="/calidad" >
13      <v-btn class="ma-1" plain> <v-icon >mdi-check</v-icon> </v-btn>
14    </router-link>
15    <router-link to="/sucursal" >
16      <v-btn class="ma-1" plain> <v-icon >mdi-domain</v-icon> </v-btn>
17    </router-link>
18    <router-link to="/tela" >
19      <v-btn class="ma-1" plain> <v-icon >mdi-widgets</v-icon> </v-btn>
20    </router-link>
21    <v-spacer></v-spacer>
22    <router-link to="/login" v-if="!user">
23      <v-btn class="ma-1" plain> Login </v-btn>
24    </router-link>
25    <v-btn icon>
26      <v-icon>mdi-magnify</v-icon>
```

14. en la carpeta router, se encuentra el archivo index.js donde están guardadas las rutas para navegar en la aplicación.

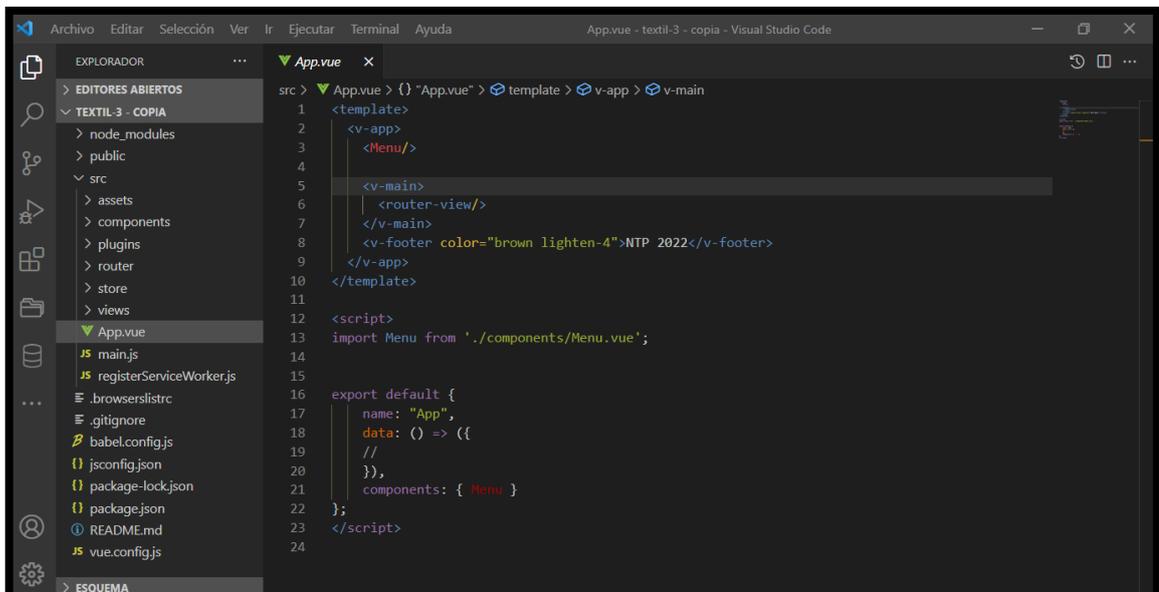
```
1 import CalidadRoutes from '@/views/calidad/router/index.js'
2 import sucursalRoutes from '@/views/sucursal/router/index.js'
3 import telaRoutes from '@/views/telas/router/index.js'
4 import Vue from 'vue'
5 import VueRouter from 'vue-router'
6 import userRoutes from '../views/users/router/index.js'
7
8
9 Vue.use(VueRouter)
10
11 const routes = [
12   {
13     path: '/',
14     name: 'home',
15     component: () => import("../views/HomeView.vue")
16   },
17   {
18     path: '/calidad',
19     name: 'calidad',
20     component: () => import("../views/CalidadView.vue")
21   },
22 ],
23 ...userRoutes,
24 ...CalidadRoutes,
25 ...sucursalRoutes,
26 ...telaRoutes,
```

15. En la carpeta store se encuentra el archivo indes.js este archivo contiene las consultas de datos que se hacen a las diferentes tablas creadas mediante el método get para seleccionar los registros del módulo de calidad, sucursal, tela.



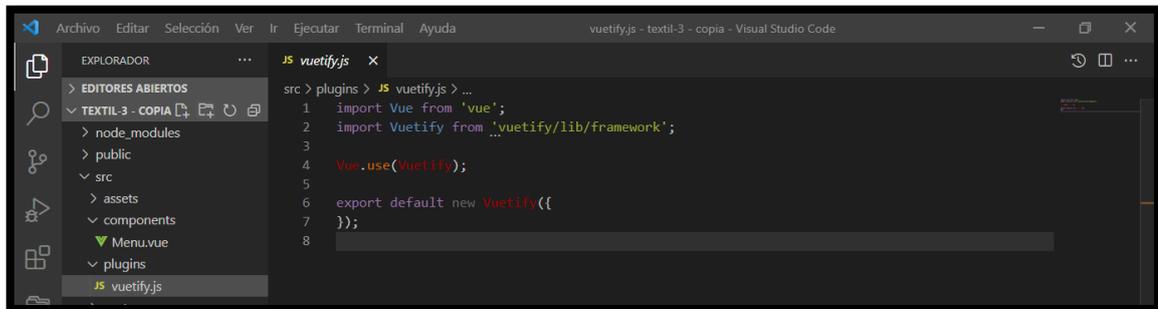
```
src > store > JS index.js > ...
1  import Vue from 'vue'
2  import Vuex from 'vuex'
3  import axios from 'axios'
4
5  Vue.use(Vuex)
6
7  export default new Vuex.Store({
8    state: {
9      posts: [],
10     calidades: [],
11     sucursales: [],
12     telas: [],
13   },
14   getters: {
15   },
16   mutations: {
17     getAllPost(state,payload){
18       state.posts = [...payload];
19     },
20     getAllCalidad(state,payload){
21       state.calidades = [...payload];
22     },
23     getAllSucursal(state,payload){
24       state.sucursales = [...payload];
25     },
26     getAllTela(state,payload){
```

16. En la carpeta views se encuentra un único archivo denominado App.vue este es el archivo principal de la aplicación que hace la función de ejecutar todo el Front End



```
src > App.vue > {} "App.vue" > template > v-app > v-main
1  <template>
2    <v-app>
3      <Menu/>
4
5      <v-main>
6        <router-view/>
7      </v-main>
8      <v-footer color="brown lighten-4">NTP 2022</v-footer>
9    </v-app>
10 </template>
11
12 <script>
13 import Menu from './components/Menu.vue';
14
15
16 export default {
17   name: "App",
18   data: () => ({
19     //
20   }),
21   components: { Menu }
22 };
23 </script>
24
```

17. en la carpeta Plugins se encuentra el archivo vuetify.js, es una biblioteca de Vue.js destinada a la creación de interfaces de usuario que implementa componentes de usabilidad listos para usar con la combina de Vue.js y Material Design.



The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left and the Code Editor on the right. The Explorer sidebar shows the file structure of a project, with the 'plugins' directory expanded to show the 'vuetify.js' file. The Code Editor displays the content of 'vuetify.js', which is a JavaScript file that imports Vue and Vuetify, uses Vuetify, and exports a new Vuetify instance.

```
src > plugins > JS vuetify.js > ...
1  import Vue from 'vue';
2  import Vuetify from 'vuetify/lib/framework';
3
4  Vue.use(Vuetify);
5
6  export default new Vuetify({
7  });
8
```

CONCLUSIÓN

Está claro que un sistema de inventario es un elemento moderno muy importante dentro de cualquier empresa que realice dicha función. Por tanto, el proyecto realizado en la empresa antes mencionada, es una herramienta que cumple con todos los estándares que demanda. Se puede deducir, que en el presente manual de desarrollador se presenta la comunicación técnica, que este tendrá con el sistema; con el cual se busca facilitar el trabajo a los usuarios. De igual forma, se busca que el usuario tenga una interacción más cercana con el sistema, utilizando como base las capturas que se presentan en este documento. El objetivo principal es que el usuario conozca cómo preparar los datos para obtener los resultados en el sistema. Además, este debe servir como herramienta de aprendizaje y referencia para el personal administrativo de dicha empresa.