Tecnológico de Costa Rica

Campus Tecnológico Local San Carlos

Unidad de Computación

Bases de datos I

Entrega modelo de aplicación web

Integrantes:

Jose Ignacio Alfaro Solano

Warner Fidel Hurtado Laguna

Profesora:

Marlen Treviño Villalobos

I semestre 2020

01 de agosto del 2020

Tabla de contenido

[Introducción 3](#_Toc47146774)

[Análisis del problema 3](#_Toc47146775)

[Solución del problema 3](#_Toc47146776)

[Repositorio del frontend 4](#_Toc47146777)

[Repositorio del backend 5](#_Toc47146778)

[Manejo de la base de datos 6](#_Toc47146779)

[Análisis de los resultados 7](#_Toc47146780)

[Diferentes vistas de la aplicación 7](#_Toc47146781)

[Reflexiones personales 9](#_Toc47146782)

[Jose Ignacio 9](#_Toc47146783)

[Warner 10](#_Toc47146784)

[Conclusiones 10](#_Toc47146785)

[Recomendaciones 11](#_Toc47146786)

[Referencias bibliográficas 11](#_Toc47146787)

[*MARK OTTO, A. B. C.* 11](#_Toc47146788)

[*CHENKIE, R*. 11](#_Toc47146789)

[*ANGULAR* 11](#_Toc47146790)

# Introducción

El objetivo de este proyecto es realizar una aplicación web para el manejo de viáticos la cual se conecte a la base de datos anteriormente creada, para así tener un panorama más amplio del funcionamiento de una base de datos en un entorno real de producción. Esta aplicación permitirá el manejo de eventos, viáticos y así como también la posibilidad de habilitar y deshabilitar información en las tablas para el manejo de la aplicación, así como sus restricciones planteadas para el flujo del programa.

Además, se pretende dejar en evidencia la construcción de la aplicación con sus respectivos comentarios en el controlador de versiones utilizado, en este caso Git, así se puede tener una mejor perspectiva de lo que comprende la creación de una aplicación que consuma los datos de una base de datos.

# Análisis del problema

El problema por resolver es crear una base de datos y una aplicación que permita el manejo de viáticos y eventos de una empresa, esta misma debe permitir ingresar, editar, eliminar y desactivar cualquier tipo de información en sus tablas respectivas, además, se manejará mediante triggers los cambios en tablas que dependan de otras para así hacer los cambios en los respectivos estados, por último, se generan sus restricciones y mensajes al cliente que se encuentre utilizando la aplicación.

El enfoque de esta aplicación va a ser de tipo web ya que el fácil acceso de esta misma desde cualquier dispositivo y lugar hará que su valor crezca considerablemente a cambio de que sea una aplicación de tipo desktop, ya que el acceso externo a una aplicación de este tipo es un poco más complicado.

# Solución del problema

Se procedió a utilizar el framework de Google llamado Angular en su versión 9+ como parte del frontend de la aplicación ya que como se explicó este mismo es de fácil acceso y utilización desde cualquier dispositivo ya sea desde una laptop o un smartphone ya que este framework se trabaja bajo el estilo de componentes responsive, además, para el manejo de la información que va a llegar al sistema de base de datos se utilizó un backend que se encuentra programado bajo la tecnología de Node Js que trabaja con el lenguaje de JavaScript al igual que Angular.

Se quiso dar ese enfoque web ya que tanto la validación de datos como el manejo de cualquier tipo de información se hace estrictamente en el frontend y en el backend, por lo que reduce algún tipo de error a la hora de su llegada a las tablas de la base de datos, ya que si lo comparamos con un lenguaje desktop como Java se tendría que hacer muchas más validaciones así como triggers en la base de datos para que se dispare si existe algún tipo de error, lo que el manejo de estas mismas tablas se haría un poco más compleja y también aumentaría en tamaño en cantidad de funciones de la base de datos.

Un dibujo de una cara con ojos y boca

Descripción generada automáticamente con confianza baja

Ilustración Lenguaje de programación utilizado

Ilustración Node Js + Javascript como servidor

Ilustración Framework del Frontend

## Repositorio del frontend

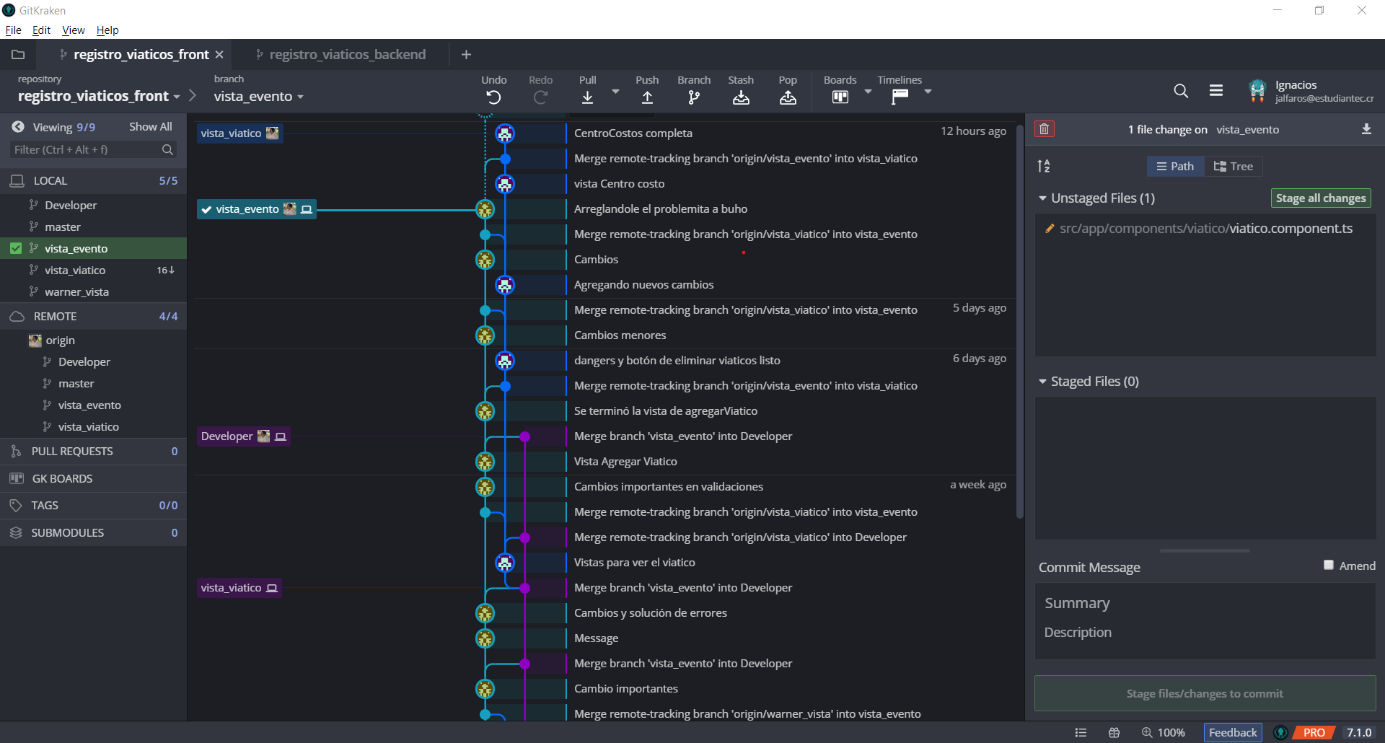
Para el manejo de las tareas en el frontend se procedió a manejar mediante el controlador de versiones Git bajo la utilización de GitKraken, para que de esta manera primeramente tener respaldos por si hubiera existido cualquier tipo de error y segundo al ser un trabajo que se hace cada uno desde su computadora así sea más simple el manejo de las etapas de desarrollo en la que se encuentran la aplicación.

Ilustración Repositorio del Frontend

Se manejó cada vista mediante ramas y luego cuando se encontraban estables se procedían a subir esos cambios a una rama llamada Developer, para así ahí manejar la última versión estable de la aplicación y proceder a realizar los respectivos merge a las ramas locales.

## Repositorio del backend

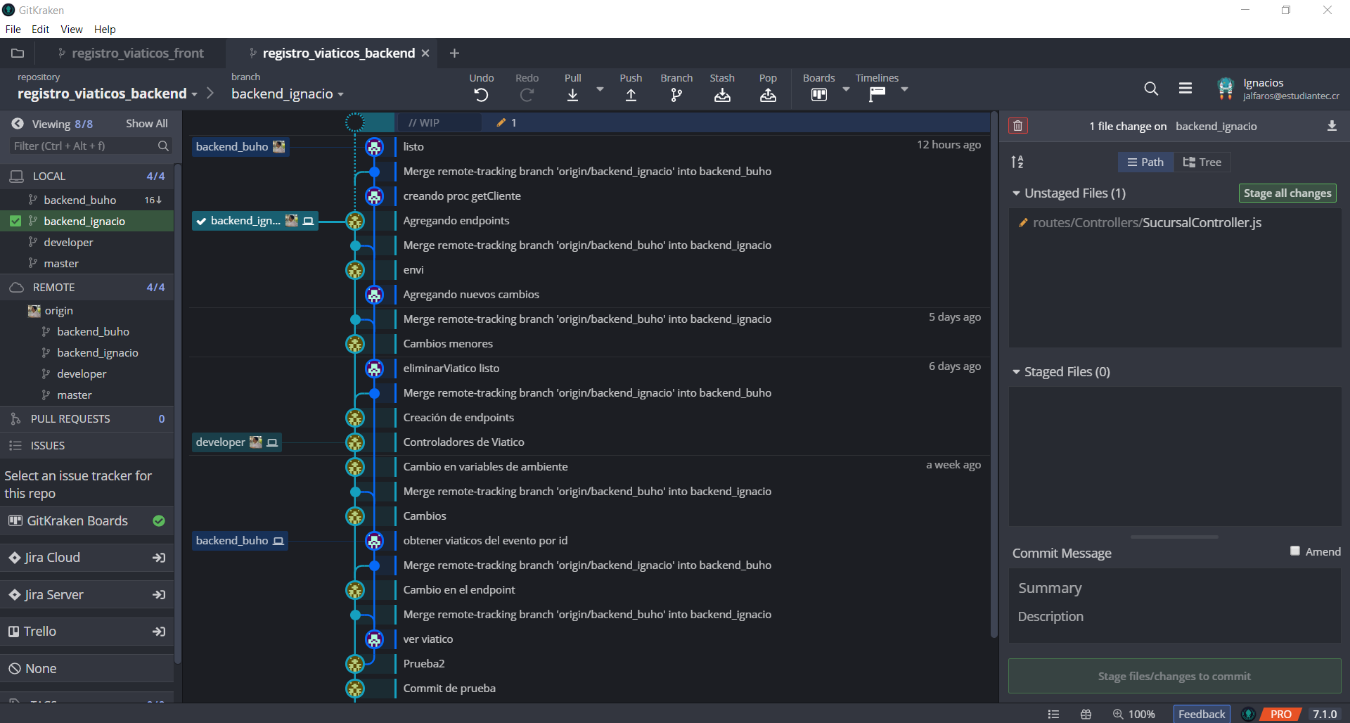
Al igual que la parte del frontend se contó con un repositorio para así cada uno pudiera hacer localmente sus respectivos controladores y sus endpoints que iban a ser consumidos por el framework Angular.

Ilustración Repositorio de Backend

Al utilizar estos repositorios se pudo tener un mayor orden con respecto al proyecto, dónde nos asignábamos tareas las cuales después de realizadas se unen a la rama principal, esto facilita el trabajo ya que por medio de los commits se puede ver quién y que trabajo se realizó. Además de ayudar a tener un mejor orden tenemos un respaldo el cual podemos en algún momento volver a un cambio anterior si por alguna circunstancia se necesitaría devolverse a un estado anterior.

## Manejo de la base de datos

Para el manejo de la base de datos no se conoció de alguna manera en la cual poderlo manejar mediante un repositorio, por lo que acudimos a enviarnos al chat de WhatsApp los scripts de los diferentes procesos almacenados y los otros recursos que se fueron programando en la misma, por lo que cada uno posee una versión exactamente igual de la base de datos.

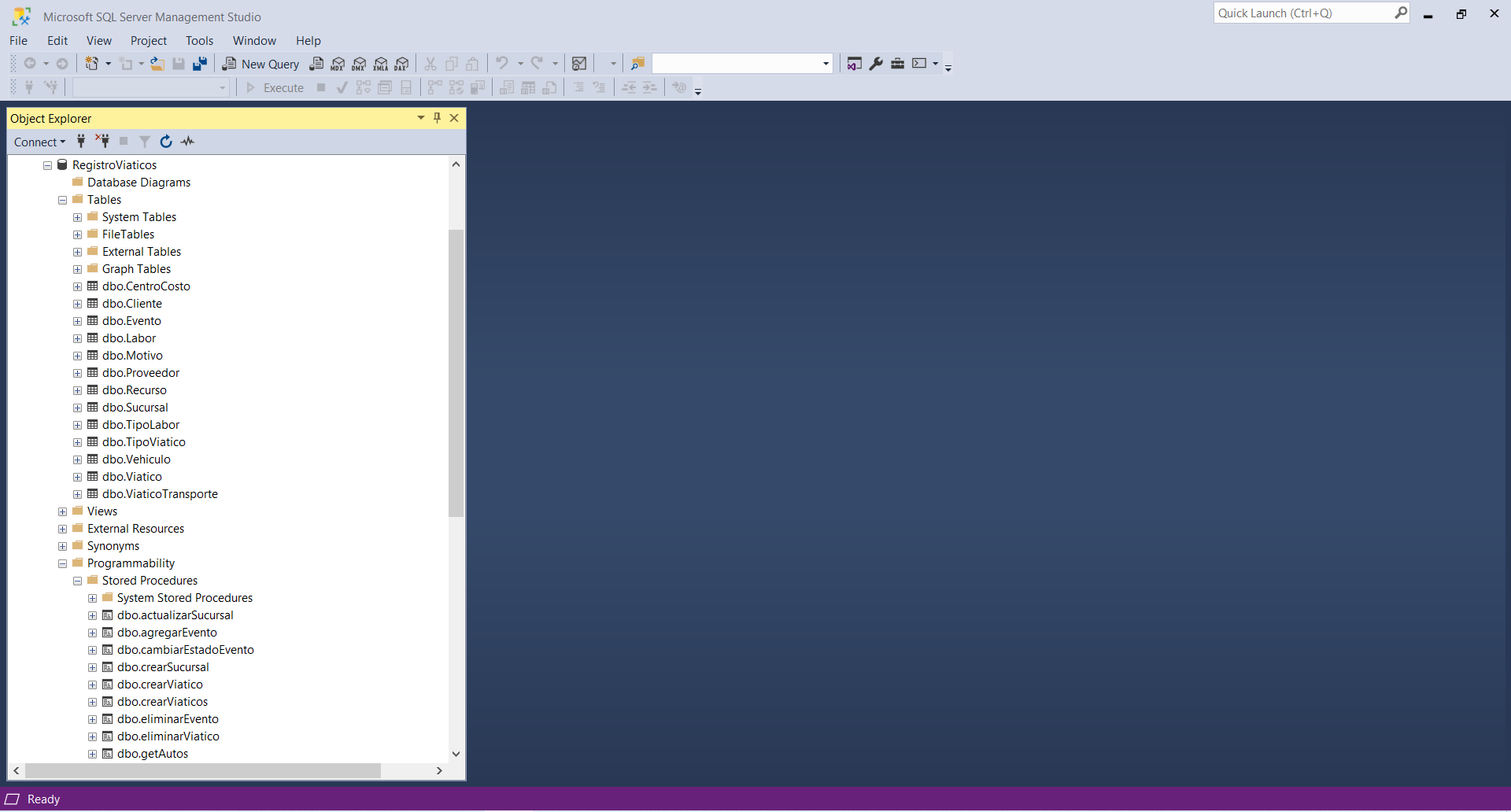


Ilustración Base de datos e información

En esta imagen se puede ver la cantidad de tablas utilizadas en la creación de esta aplicación, además de algunos procesos almacenados que son consumidos por el backend.

# Análisis de los resultados

Primeramente, se logró completamente el objetivo deseado ya que se desarrolló una aplicación totalmente funcional donde en la base de datos se puede hacer cualquier tipo de manejo de información que la aplicación, el usuario o el administrador requiera.

Por otra parte, existen muchas más mejoras que se pueden aplicar a este frontend, principalmente en las tablas de información algunas de ellas son: que no corte el texto cuando es más largo de la celda y otra sería agregar la paginación para que no muestre tantas filas de evento, sino que, el usuario desee cuantas quiere ver por página y tenga la posibilidad de decidir cuantas quiere manejar.

## Diferentes vistas de la aplicación

En una reunión sobre el estilo que se pensaba darle a la aplicación procedimos a llegar a la convención de tener vistas simples, separadas y totalmente intuitivas para el usuario, ahora bien, dejando de lado la parte estética, la aplicación cuenta las diferentes alertas en sus diferentes vistas, además de que cada formulario se encuentra validado con advertencias para que así independientemente de lo que arroje la aplicación el usuario siempre sepa qué hacer, por otra parte, la aplicación cuenta con un navbar para que la navegación a través de las distintas vistas sea de forma más simple.

Además, se cuenta con protección en las rutas, por lo que cualquier intento de acceder a una ruta que no exista mediante el URL se procederá a hacer su respectivo redireccionamiento a su vista principal que es la vista de ver los eventos.

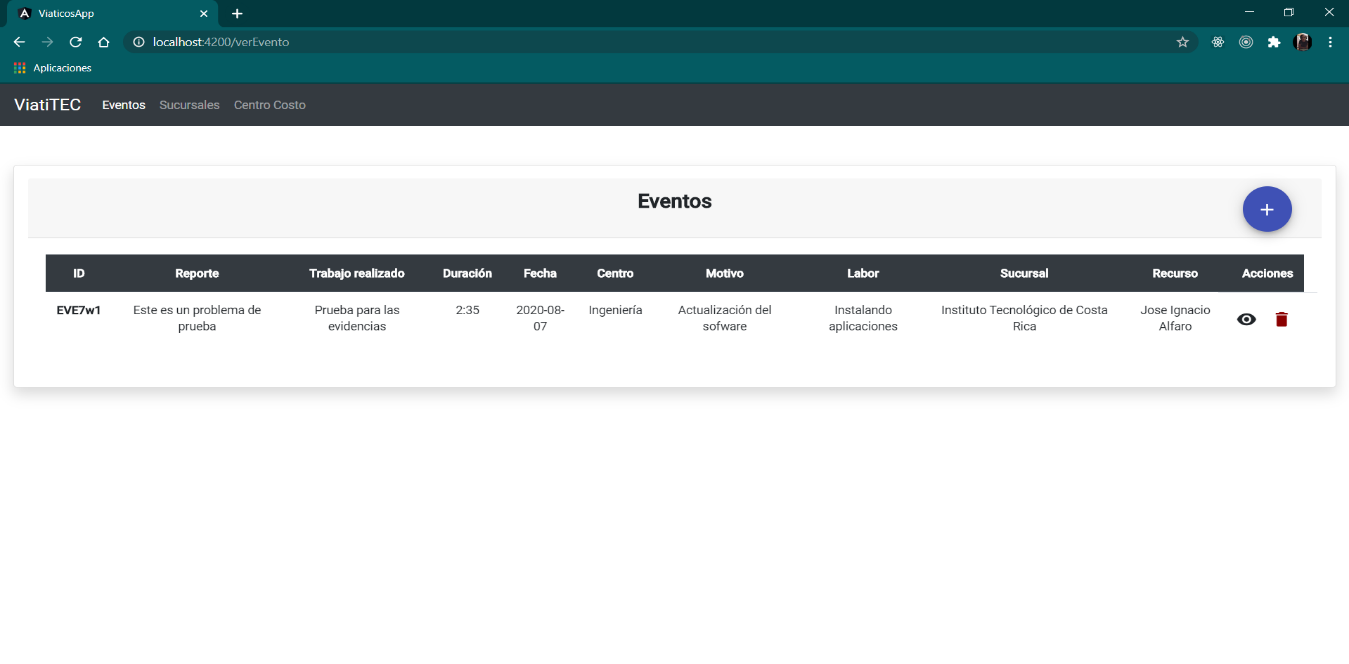


Ilustración Ventana principal de la página, vista de Eventos

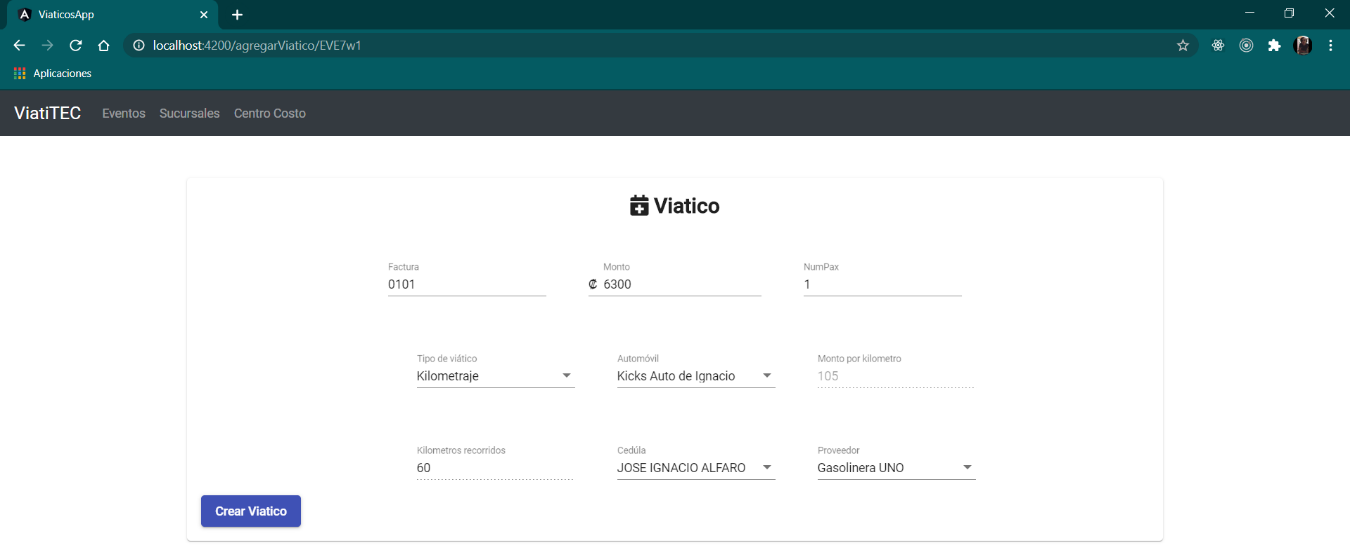


Ilustración Formulario de agregar un viático

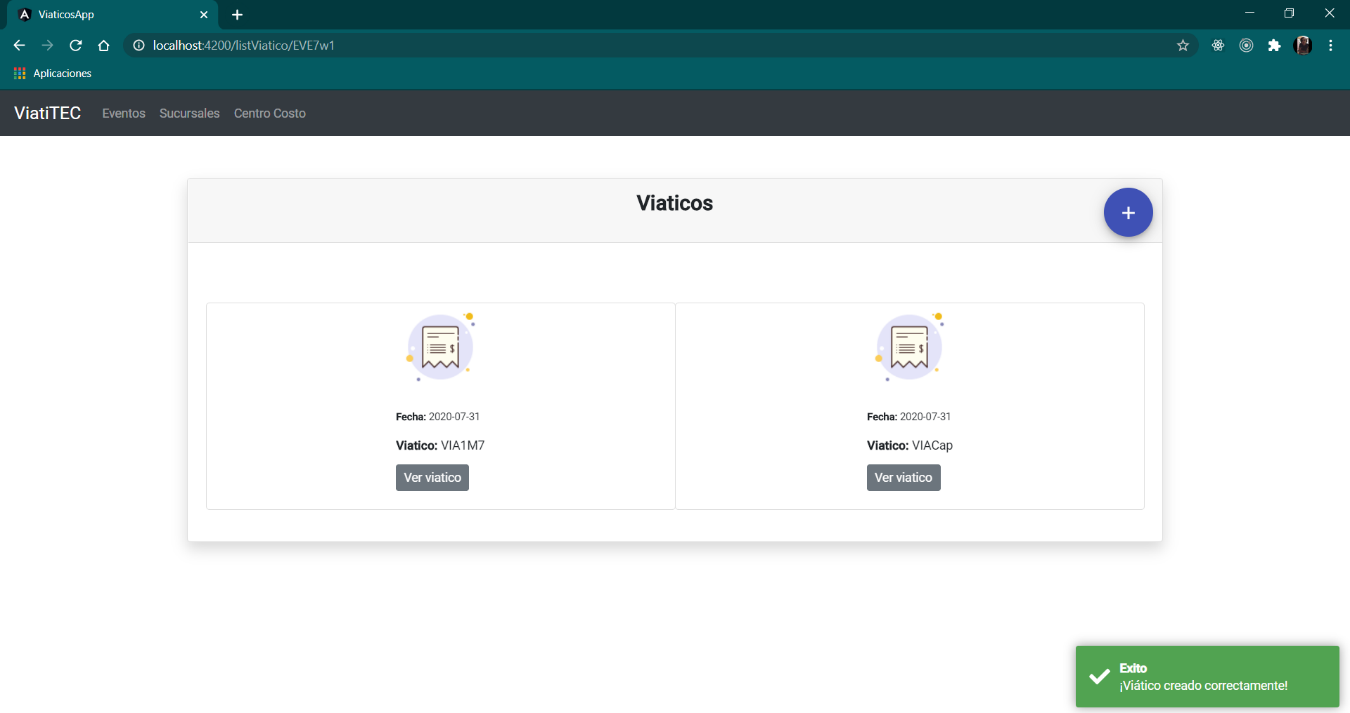


Ilustración Vista de viáticos que posee un evento

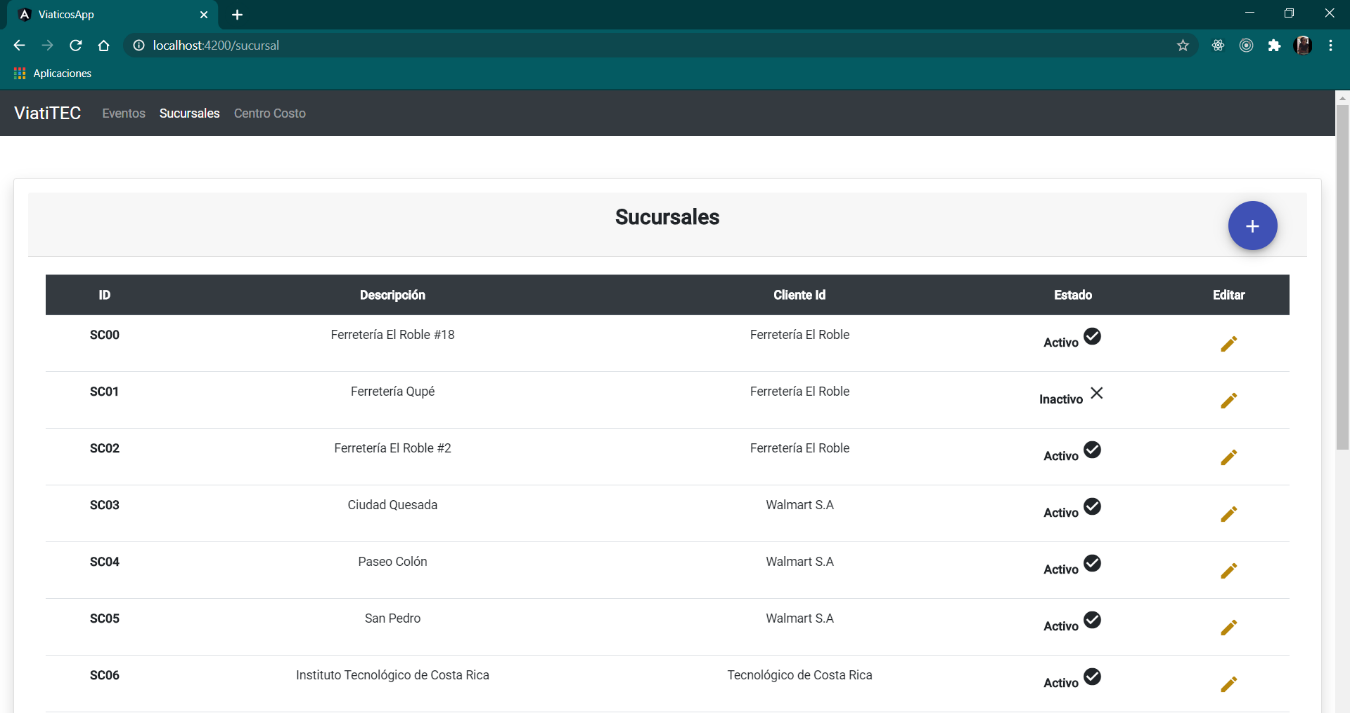


Ilustración Una de las vistas que permite manejar los estados de las entidades, así como su nombre

# Reflexiones personales

## Jose Ignacio

Me parece que primeramente el saber cómo son almacenados los datos en unas tablas con distintas relaciones entre ellas es algo realmente maravilloso ya que me hace tener una noción más cercana de cómo funciona en realidad las páginas web o aplicaciones que utilizamos día con día. Aunque por otra parte me hace pensar que en realidad mucha de nuestra información personal se encuentra almacenada en alguna tabla de alguna aplicación a la cual nos hayamos registrado, tanto así como contraseñas e inclusive los números de tarjeta de crédito, por lo que yo como desarrollador de software me hace pensar en la gran responsabilidad que tiene manejar este tipo de información ya que estamos trabajando con la confianza de alguna o muchas personas que están confiando en darnos información personal para que así ellos puedan consumir algún tipo de servicio que se está prestando, entonces me hace pensar que primeramente antes de ser desarrollador se debe de aprender de ética en el trabajo ya que al manejar cierta información que tal vez no es valiosa para uno como desarrollador, otras personas pueden estar bien interesadas en estas.

## Warner

Sin duda alguna las bases de datos vinieron a mejorar nuestras vidas, tanto para nosotros como para las empresas, gracias a ellas se puede almacenar con orden mucha información disminuyendo el trabajo que se realizaba en el pasado. Al trabajar con bases de datos tenemos que poner en práctica la ética, ya que muchos de estos datos que están en nuestras manos y nosotros como programadores manejamos son de mucha importancia y de total discreción para sus propietarios, ya que si se filtran estos datos puede que afecte de manera negativa una empresa porque estos archivos o datos puede que sean sobre información de cuentas u clientes. Por ende, se debe manejar esta información con mucha responsabilidad y cautela, tratando de poner en práctica los conocimientos adquiridos sobre la seguridad en los gestores de bases de datos, donde podemos dar muchos tipos de permisos sobre el manejo de la información tanto al usuario como al colaborador de la construcción de la base de datos. De esta manera y muchas otras podemos dar seguridad a la información, una de ellas sería quitar accesos a algunos datos con los que se esté trabajando en los cuales no es necesario que otros empleados tengan el acceso a ellos. Finalmente quiero declarar que este curso ha sido de gran ayuda y aprendizaje, he aprendido cosas que me van a ayudar como una gran base tanto personalmente como para mi futuro debido a que he aprendido mucho acerca de muchos temas en los cuales no tenía mucho conocimiento y otros que totalmente desconocía, los cuales siempre tendré presente y usaré para el desarrollo de aplicaciones, por lo tanto, seguiré estudiando y descubriendo nuevos temas acerca de los gestores de bases de datos.

# Conclusiones

En conclusión se puede ver cómo el utilizar algún tipo de base de datos para la construcción de nuestras aplicaciones es de suma importancia, ya que nos permite solo manejar la información que queremos mostrar en la parte de la aplicación en la que el usuario se encuentra, además nos brinda seguridad a la aplicación ya que toda esta información que la aplicación se encuentra manejando está completamente separada del framework o del lenguaje que se está utilizando para hacer la aplicación en la parte visual, agregando a esto último el hecho de tener la bases de datos conectada a un servidor nos brinda más beneficios ya que la mayor parte de la validación que se hace de la información se hace principalmente en el frontend y en el servidor por lo que podemos afirmar que la información que deseemos editar o agregar en nuestras tablas lleguen completamente “limpias” y así nos reducimos de recursos en la base de datos y le dejamos ese trabajo al servidor y frontend.

# Recomendaciones

Primeramente, se debería de agregar un sistema de autenticación para que así solo personas con diferentes roles tengan acceso a solo una parte de la aplicación y de esta manera sacarle un gran provecho a la seguridad de enrutamiento que Angular tiene disponible.

Las validaciones de formularios pueden ser aún más estrictas como en los inputs que solo se activen cuando se dispara algún evento y que se tenga validaciones como por ejemplo en los viáticos de transporte, que estos solo se puedan agregar únicamente de una cantidad delimitada por la empresa hacia arriba.

Se pueden tener vistas ocultas solo para los desarrolladores o personas de mantenimiento de la aplicación para que así solo las personas con los privilegios suficientes puedan hacer el uso de los servicios como activar o desactivar entidades.

# Referencias bibliográficas

### MARK OTTO, A. B. C.

#### **Introduction**

[1]a. Mark Otto, "Introduction", *Getbootstrap.com*, 2020. [Online]. Available: https://getbootstrap.com/docs/4.1/getting-started/introduction/. [Accessed: 01- Aug- 2020].

### 

### CHENKIE, R.

#### **Angular Authentication: Using Route Guards**

[3]R. Chenkie, "Angular Authentication: Using Route Guards", *Medium*, 2020. [Online]. Available: https://medium.com/@ryanchenkie\_40935/angular-authentication-using-route-guards-bf7a4ca13ae3. [Accessed: 01- Aug- 2020].

### 

### ANGULAR

***Getting started***

[3]"Angular", *Angular.io*, 2020. [Online]. Available: https://angular.io/docs. [Accessed: 01- Aug- 2020].