

Instituto Tecnológico de Costa Rica
Área de Ingeniería en Computadores
Bases de Datos – CE3101

Profesor:

Marco Rivera Meneses

Documentación Externa

Estudiante:

Brandon Gomez Gomez

Erick Madrigal

Marco Rivera Serrano

Kevin Rodriguez Lanuza

Grupo:

1

II Semestre

2021

Descripción de los métodos implementados

La lista de los métodos implementados es la siguiente

Para los roles:

- Get (general, obtiene todos los roles)
- Get (individual, obtiene un rol individual)
- Post
- Put

Para los clientes:

- Get (general, obtiene todos los roles)
- Get (individual, obtiene un rol individual)
- Post
- Put

Para las tarjetas:

- Get (general, obtiene todos los roles)
- Get (individual, obtiene un rol individual)
- Post
- Put

Para los usuarios:

- Get (general, obtiene todos los roles)
- Get (individual, obtiene un rol individual)
- Post
- Put

Para las cuentas:

- Get (general, obtiene todos los roles)
- Get (individual, obtiene un rol individual)
- Post
- Put

Descripción de las estructuras de datos desarrolladas

Para el manejo de la información a nivel interno del API, se utilizaron listas. Estas almacenan un ítem o record del json asignado y a partir de este se puede retornar toda la lista, get general; un elemento en específico, get individual; agregar un ítem nuevo, método post; o editar algún récord ya registrado, método put.

Descripción detallada de los algoritmos desarrollados

Los métodos entre los tipos (roles, clientes, tarjetas y demás), son repetitivos así que solo se realizará una explicación por método implementado.

- Obtener todos los datos:

Para este caso, no se recibe ningún parámetro de entrada, dado que se quiere obtener toda la información. Para esto, primero se toma la variable que contenga la ruta al archivo json

especificado. Luego, se lee toda esta información del json y se guarda en la variable response que será devuelta como respuesta en formato json.

- Obtener un dato en específico:

De la misma manera, se intenta leer el archivo json, pero ahora el json se transforma en una lista de alguna clase. Luego, utilizando los métodos de las listas, se busca un ítem que coincida con el parámetro. Finalmente, se realiza la respuesta, si el objeto no fue encontrado, se devuelve un null.

- Para agregar un ítem:

Se lee el json especificado y se transforma en una lista de alguna clase. Utilizando los métodos de las listas, se agrega un nuevo elemento y se asigna cada atributo según la clase de la lista. Luego, esta misma se sobrescribe en el json y se devuelve un mensaje de confirmación.

- Para actualizar un ítem:

Se lee el json especificado y se transforma en una lista de alguna clase. Utilizando un foreach loop, se busca el ítem respectivo según el parámetro y se actualizan los valores. Luego, el archivo es sobrescrito y se envía un mensaje de confirmación.

Problemas conocidos

No se pudo implementar el método Delete a ninguna clase debido a un error en la lectura del json. Otra posible explicación es que sea una barrera de protección en el archivo que impida la eliminación de información de este.

Se tuvieron problemas para implementar la lógica en angular, por tanto, la aplicación web quedó como una vista, con poca funcionalidad. Hasta la escritura de este documento no se pudo arreglar este problema, y se continúa con la poca funcionalidad de las vistas administrador y cliente.

Problemas encontrados

Error con angular, mala instalación

Al elaborar el app web, al momento de codificar y probar las actualizaciones hubieron varios problemas, que tomaron mucho tiempo, pues no se sabía cuál era, sin embargo entre prueba y error, se observó que solo era necesario volver a instalar angular.

Lecturas internas del Json

Descripción detallada:

No se podía acceder al contenido del json si no era leído como un string. Esto dificulta la búsqueda dado que pueden existir cadenas similares y realizar la computación de estos lleva mucho esfuerzo de procesamiento.

Soluciones encontradas:

Existe una librería interna en Visual Studio llamada Newtonsoft.Json que permite manipular estos archivos con más facilidad. Específicamente, los métodos llamados "DeserializeObject" y "SerializeObject". Estos métodos transforman el contenido de un json a una lista de cierta clase y permite que la obtención, búsqueda, actualización y sobreescritura sea una tarea sencilla.

Recomendaciones:

Investigar las librerías que puedan servir de ayuda sin tener que realizar código extra para una función tan sencilla y así, poder aprovechar el tiempo en completar el o los requerimientos.

Conclusiones:

Saber qué tipo de archivos se utilizarán para agregar una librería a la solución del proyecto y así, simplificar el trabajo.

Envío de datos entre ventanas

Descripción detallada:

Al cargar el listview de la lista de compras de tarjeta se obtenía el dato de la tarjeta de la ventana anterior, pero la carga de estos datos no se realizaba a la hora de rellenar la lista de compras por lo que aparecía un error de que se apuntaba a un nulo.

Intentos de solución:

Se realizó un delay para tratar de darle tiempo a la carga de los datos de la otra ventana, también se intentó poniendo la carga de los datos a la hora de crear el view y rellenar la tabla hasta que se mostrará la ventana pero tampoco funcionó.

Soluciones encontradas:

Los datos se cargan en el momento en que la ventana es creada y se implementó un botón el cual rellena la lista al momento de presionarlo.

Trabajo en equipo

Plan de trabajo

Objetivo	Tiempo estimado	Propietario(s)	Entrega
Desarrollo en angular	6 días	<ul style="list-style-type: none">• Brandon• Kevin• Erick	23/08/21
Desarrollo en Bootstrap	6 días	<ul style="list-style-type: none">• Brandon• Kevin• Erick	23/08/21
Desarrollo en HTML	6 días	<ul style="list-style-type: none">• Brandon• Kevin• Erick	23/08/21
Desarrollo en C#	10 días	<ul style="list-style-type: none">• Marco	26/08/21
Creación de métodos	5 días	<ul style="list-style-type: none">• Marco	31/08/21
Creación de jsons	3 días	<ul style="list-style-type: none">• Marco	31/08/21
Completar métodos	7 días	<ul style="list-style-type: none">• Marco	9/14/21
Conexión	5 días	<ul style="list-style-type: none">• Brandon• Kevin• Erick• Marco	15/09/21

Seguimiento

- Lunes 23 de agosto del 2021:

Para esta fecha, ya se tenían las bases para el desarrollo de la página web, aplicación móvil y el REST API. Sin embargo, la investigación a profundidad para desarrollar el API no había concluido y se necesitaba más tiempo.

- Lunes 30 de agosto del 2021:

Para esta fecha, ya se tenían las bases del rest api y se mejoraron características propias de las vistas. Además, se realizó una correcta distribución de los atributos para ajustar la base de datos.

- Lunes 06/09/2021:

Para esta fecha, se tenían 2 métodos realizados (get general y post) y los jsons ya estaban confiscados y listos para el uso del api. Todavía se está trabajando en la conexión del API y la página web.

- Lunes 13/09/2021:

Para esta fecha, se tenían los métodos get individual y put. Existe una conexión entre la página web y el API utilizando el método get. Falta implementar los otros métodos.

- Miércoles 15/09/2021:

Entrega del proyecto:

Bitácora

Asignación del trabajo, 18 de agosto

Brandon:

- 26/8/2021: Configuración de ambiente Angular, Bootstrap, Node, JQuery.
- 1/09/2021: Creación de NavBar y dropdown, manejo de rutas, creación de formularios.
- 1/09/2021: Actualización con angular material, creación de dropdown en formulario.
- 4/09/2021: Manejo interno de formularios y envío de informacion local
- 6/09/2021: Completado manejo de tarjetas desde frontEnd, linkeado con API
- 13/09/2021: Añadido opción de borrar y todas las demás clases.
- 15/09/2021: Bases de datos y FrontEnd, editar exitoso.

Erick:

- 25/8/2020-27/8/2021 Investigación y realización de pruebas en Android studio
- 28/8/2021: Creación: de la base de una Ventana de login
- 31/08/2021-3/08/2021: Creación de las páginas principales de la app y la navegación entre las mismas
- 08/09/2021: Se implementa el envío de datos entre pantallas y se empieza a realizar la pantalla que muestra las compras de las tarjetas
- 09/09/2021: Se termina la pantalla de donde se muestran las compras de las tarjetas
- 10/09/2021: Se crea la ventana de transferencias y de movimientos
- 11/09/2021: Se cambia la ventana de préstamos para realizar los pagos ordinarios y extraordinarios
- 12/09/2021: Se termina la ventana de préstamo y de tarjetas
- 14/09/2021: Se terminan las validaciones del login

Marco:

- Se me asignó realizar el REST API, así que se buscará información para la creación de este en C# y .NET.
- Logré crear un API sencillo que devuelve dos valores por defecto.
- Cree los archivos jsons para ser usados como bases de datos. Todavía existe la posibilidad de que se realicen cambios.
- Investigué e implementé un método para manipular archivos jsons desde C#.
- Realice el método get para obtener todos los datos. Falta investigar cómo obtener un solo dato.
- Logré el método post de manera satisfactoria.

- Codifiqué el método get para un solo valor.
- Terminé el método put y ahora se pueden utilizar 3 métodos en todo el api.
- Intenté crear el método Delete, pero se generan errores que impiden la eliminación de un récord en el json.

Kevin:

- 25/08/2021 - 27/08/2021 Investigación sobre angular, bootstrap, css y HTML.
- 09/09/2021 Creación de la vista de Login, pago de cuentas y movimientos.
- 10/09/2021 Creación para la vista Tarjetas asociadas.
- 11/09/2021 Creación de la vista para préstamos.
- 22/08/2021 - 14/08/2021 Intento de implementación de lógica en angular.
- 15/09/2021 Terminar Documentación.

Conclusiones

El manejo de información debe tomarse con mucho cuidado. Nosotros como programadores podemos llegar a manejar información muy delicada, privada o sensible de cientos de miles de personas y está en nuestras manos mantener la seguridad de esta información evitando filtraciones de datos.

Una interfaz amigable con el usuario le permite con muchísima facilidad acceder al sistema y poder realizar todas las funciones que desea. En cambio, si al usuario se le presenta una aplicación tosca, le resultará complicado navegar y no podrá realizar las acciones que quiera. Además, no será una experiencia gratificante para este.

Recomendaciones del proyecto

Realizar búsquedas exhaustivas acerca de los componentes o partes del proyecto. Existe una inmensa cantidad de tutoriales y videos que permiten tomar los problemas presentados de una manera sencilla y poder aprovechar el tiempo en perfeccionar el proyecto.

Mantener una fecha límite para las entregas por parte de cada compañero, permite una mejor organización del proyecto y permite que se avance rápido y elimina la mala costumbre de hacer días u horas antes. También hay que recordar ser flexibles con las fechas límites por si se presenta alguna dificultad a los compañeros de trabajo.

Bibliografía

- Developers, G. (n.d.). *Android Developers*. obtenido de: <https://developer.android.com/docs?hl=es-419>
- Art of Engineer (1 de agosto del 2020). *Learn Angular 10, Web API & SQL Server by Creating a Web Application from Scratch*. Youtube. <https://www.youtube.com/watch?v=4a9VxZjnT7E>
- Mairegger, M. (12 de Octubre de 15). *Stackoverflow*. Obtenido de json add new object to existing json file C#:

<https://stackoverflow.com/questions/33081102/json-add-new-object-to-existing-json-file-c-sharp>