



Nombres: Ricardo Paredes
Richard Rocha

Fecha: 14/07/2023
Grupo: N

Diseño de Base de Datos del Proyecto "Money-Saver"

Para el proyecto "MoneySaver", se propone utilizar una base de datos no relacional, específicamente Firebase, como backend para almacenar los datos de gastos de forma segura y permitir su sincronización en diferentes dispositivos. A continuación, se justifica esta elección:

1. **Escalabilidad:** Firebase es una plataforma de desarrollo de aplicaciones en la nube que ofrece una base de datos en tiempo real y escalable. Permite manejar grandes volúmenes de datos y el aumento de usuarios sin comprometer el rendimiento. A medida que la cantidad de usuarios y transacciones aumente, Firebase podrá adaptarse de manera eficiente.
2. **Sincronización en tiempo real:** Firebase proporciona sincronización en tiempo real, lo que significa que cualquier cambio realizado en un dispositivo se reflejará automáticamente en todos los demás dispositivos conectados. Esto es especialmente relevante para una aplicación de seguimiento de gastos, donde es crucial mantener los datos actualizados en todos los dispositivos del usuario.
3. **Facilidad de desarrollo:** Firebase ofrece una interfaz intuitiva y una amplia documentación, lo que facilita su implementación en el proyecto. Además, proporciona SDKs (kits de desarrollo de software) para diversas plataformas, incluyendo Android, lo que simplifica la integración con la aplicación móvil.
4. **Almacenamiento seguro:** Firebase ofrece seguridad integrada para proteger los datos almacenados. Proporciona reglas de seguridad personalizables que permiten controlar el acceso y las operaciones permitidas en la base de datos. Esto garantiza que los datos de gastos personales de los usuarios estén protegidos y solo sean accesibles para ellos.
5. **Estructura flexible:** Una base de datos no relacional, como Firebase, permite una estructura flexible de los datos. En el contexto de la aplicación MoneySaver, donde los usuarios pueden registrar gastos con diferentes categorías y detalles, una base de datos no relacional brinda la flexibilidad para adaptarse a los cambios en los requisitos y agregar nuevos campos o categorías de forma más sencilla que en una base de datos relacional.
6. **Integración con la aplicación móvil:** Firebase proporciona bibliotecas y herramientas de desarrollo específicas para Android, lo que simplifica la integración con la aplicación móvil desarrollada en Kotlin. Esto permite un flujo de trabajo más eficiente y una comunicación fluida entre la aplicación y la base de datos. [1][2][3]

Un modelo para la base de datos en Firebase, específicamente diseñado para el proyecto "MoneySaver":

- Colección "users":
 - Documento por cada usuario registrado.
 - Campos:
 - **user_id:** ID único del usuario (generado automáticamente por Firebase).
 - **name:** Nombre del usuario.
 - **email:** Correo electrónico del usuario (utilizado como identificador único de inicio de sesión).
 - **password:** Contraseña del usuario (almacenada de forma segura, preferiblemente en forma de hash).
 - **budgets** (subcolección):
 - Documento por cada presupuesto establecido por el usuario.



- Campos:
 - **budget_id**: ID único del presupuesto.
 - **category**: Categoría a la que se aplica el presupuesto.
 - **amount**: Monto asignado al presupuesto.
- **expenses** (subcolección):
 - Documento por cada transacción de gasto registrada por el usuario.
 - Campos:
 - **expense_id**: ID único de la transacción de gasto.
 - **category**: Categoría del gasto.
 - **description**: Descripción del gasto.
 - **amount**: Monto del gasto.
 - **date**: Fecha en la que se registró el gasto.
 - **timestamp**: Marca de tiempo para ordenar y filtrar los gastos por fecha.

A continuación, se muestra un ejemplo,

Colección "users":

- Documento "user_123":
 - Campos:
 - **user_id**: "user_123"
 - **name**: "John Doe"
 - **email**: "john.doe@example.com"
 - **password**: [contraseña almacenada de forma segura]
 - **budgets** (subcolección):
 - Documento "budget_1":
 - Campos:
 - **budget_id**: "budget_1"
 - **category**: "Comida"
 - **amount**: 200
 - Documento "budget_2":
 - Campos:
 - **budget_id**: "budget_2"
 - **category**: "Transporte"
 - **amount**: 100
 - **expenses** (subcolección):
 - Documento "expense_1":
 - Campos:
 - **expense_id**: "expense_1"
 - **category**: "Comida"
 - **description**: "Almuerzo en restaurante"
 - **amount**: 25.50
 - **date**: "2023-07-10"
 - **timestamp**: [marca de tiempo correspondiente a la fecha]
 - Documento "expense_2":
 - Campos:
 - **expense_id**: "expense_2"
 - **category**: "Transporte"
 - **description**: "Billete de autobús"
 - **amount**: 2.75
 - **date**: "2023-07-11"
 - **timestamp**: [marca de tiempo correspondiente a la fecha]



En este ejemplo, tenemos un usuario con ID "user_123" llamado "John Doe" y su dirección de correo electrónico es "john.doe@example.com".

users	[-] users { }	
	user_123	[-] user_123 { }
		user_id user_123
		name John Doe
		email john.doe@example.com
		password contraseña123
		budgets [+] budgets { }
		expenses [+] expenses { }

Ilustración 1. Estructura gráfica de la base de datos no relacional al ingresar un usuario

El usuario ha establecido dos presupuestos: uno para la categoría "Comida" con un monto de 200 dólares y otro para la categoría "Transporte" con un monto de 100 dólares.

budgets	[-] budgets { }	
	budget_1	[-] budget_1 { }
		budget_id budget_1
		category Comida
		amount 200
	budget_2	[-] budget_2 { }
		budget_id budget_2
		category Transporte
		amount 100

Ilustración 2. Estructura gráfica de la base de datos no relacional al ingresar un presupuesto

Además, el usuario ha registrado dos transacciones de gasto. La primera transacción ("expense_1") se realizó en la categoría "Comida" por un monto de 25.50 dólares, con una descripción de "Almuerzo en restaurante" y fecha de registro "2023-07-10". La segunda transacción ("expense_2") se realizó en la categoría "Transporte" por un monto de 2.75 dólares, con una descripción de "Billete de autobús" y fecha de registro "2023-07-11".

expenses	[-] expenses { }	
	expense_1	[-] expense_1 { }
		expense_id expense_1
		category Comida
		description Almuerzo en restaurante
		amount 25.5
		date 2023-07-10
		timestamp [marca de tiempo correspondiente a la fecha]
	expense_2	[-] expense_2 { }
		expense_id expense_2
		category Transporte
		description Billete de autobús
		amount 2.75
		date 2023-07-11
		timestamp [marca de tiempo correspondiente a la fecha]

Ilustración 3. Estructura gráfica de la base de datos no relacional al digitar un gasto



Anexos

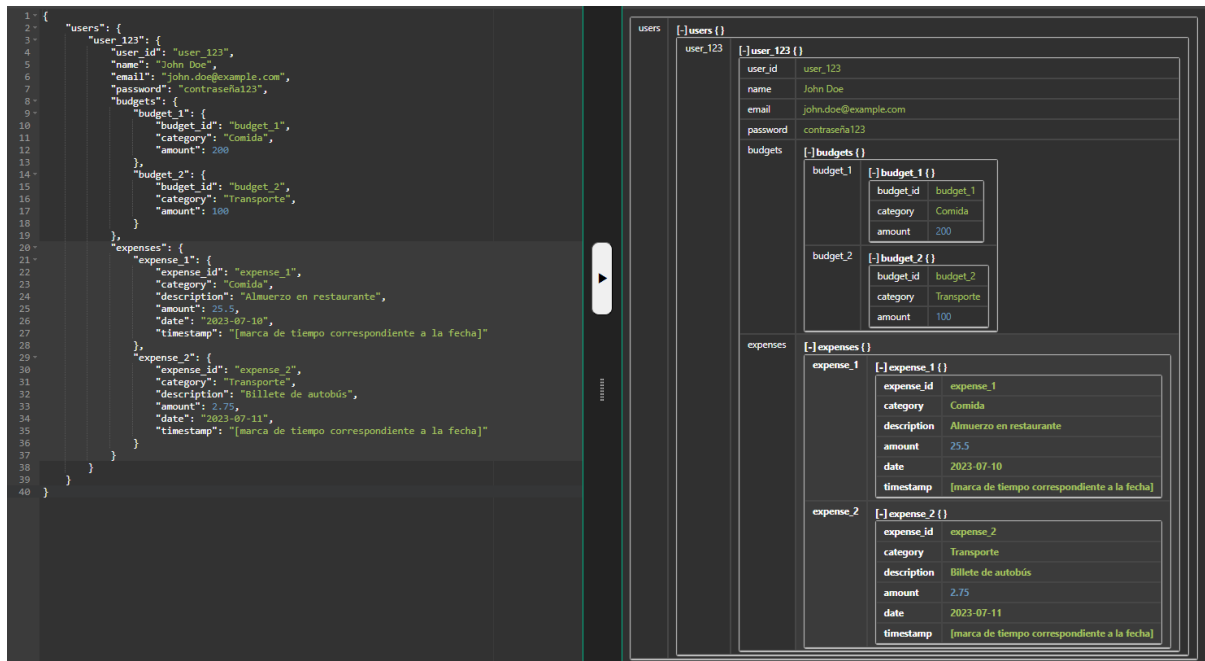


Ilustración 4. Representación gráfica de un ejemplo de nuestra base de datos

Código de ejemplo de la base de datos

```
{
  "users": {
    "user_123": { "user_id": "user_123",
                  "name": "John Doe",
                  "email": "john.doe@example.com",
                  "password": "contraseña123",
                  "budgets": {
                    "budget_1": { "budget_id": "budget_1",
                                  "category": "Comida",
                                  "amount": 200
                                },
                    "budget_2": { "budget_id": "budget_2",
                                  "category": "Transporte",
                                  "amount": 100
                                }
                  },
                  "expenses": {
                    "expense_1": { "expense_id": "expense_1",
                                   "category": "Comida",
                                   "description": "Almuerzo en restaurante",
                                   "amount": 25.5,
                                   "date": "2023-07-10",
                                   "timestamp": "[marca de tiempo correspondiente a la fecha]"
                                 },
                    "expense_2": { "expense_id": "expense_2",
                                   "category": "Transporte",
                                   "description": "Billete de autobús",
                                   "amount": 2.75,
                                   "date": "2023-07-11",
                                   "timestamp": "[marca de tiempo correspondiente a la fecha]"
                                 }
                  }
                }
  }
}
```



}

Conclusiones:

- La elección de una base de datos no relacional, como Firebase, para el proyecto "MoneySaver" ofrece ventajas significativas en términos de escalabilidad, sincronización en tiempo real, facilidad de desarrollo y almacenamiento seguro.
- El modelo básico propuesto para la base de datos en Firebase se adapta a las necesidades del proyecto, permitiendo el registro de usuarios, presupuestos y transacciones de gasto de manera eficiente.
- Firebase proporciona una integración sólida con la plataforma Android, lo que facilita el desarrollo y la sincronización de datos entre dispositivos.
- La flexibilidad de una base de datos no relacional permite una estructura adaptable y la capacidad de agregar nuevos campos o categorías según sea necesario.

Recomendaciones:

- Al implementar la base de datos, asegúrate de establecer reglas de seguridad adecuadas en Firebase para proteger los datos de los usuarios y evitar accesos no autorizados.
- Considera realizar pruebas exhaustivas para garantizar el correcto funcionamiento y la sincronización de los datos en todos los dispositivos.
- Mantén un enfoque iterativo y colaborativo con el equipo de desarrollo y los stakeholders, realizando revisiones y ajustes periódicos en el modelo de la base de datos según las necesidades y los comentarios recibidos.
- Si el proyecto crece en tamaño y complejidad, evalúa regularmente el rendimiento de la base de datos y considera ajustes o mejoras para mantener la escalabilidad y la eficiencia.
- Documenta claramente la estructura y el funcionamiento de la base de datos, incluyendo los esquemas, las reglas de seguridad y cualquier consideración especial para futuras referencias y colaboración en el proyecto.

Bibliografía

- [1] Marrero, Luciano, et al. "Un estudio comparativo de bases de datos relacionales y bases de datos nosql." *XXV Congreso Argentino de Ciencias de la Computación (CACIC)(Universidad Nacional de Río Cuarto, Córdoba, 14 al 18 de octubre de 2019)*. 2019.
- [2] Córdova Espinoza, Rosa Fernanda, and Bernardo Esteban Cuzco Sarango. *Análisis comparativo entre bases de datos relacionales con bases de datos no relacionales*. BS thesis. 2013.
- [3] Valverde, Vanessa, Narcisa Portalanza, and Paulina Mora. "Análisis descriptivo de base de datos relacional y no relacional." *Revista Atlante: Cuadernos de Educación y Desarrollo* 3 (2019).