

Actividad 2

23 de septiembre del 2021

Objetivos

El objetivo de la presente actividad es desarrollar una aplicación que controle los gastos de los empleados de la empresa Tech Fashion. Se realice una gestión de deudas de cada usuario a través de varios algoritmos y se cancelen y simplifiquen según el sistema crea conveniente.

El desarrollo del software debió estar acompañado en todo momento por el diagrama UML especificado al inicio de la actividad.

Requerimientos

Características iniciales

- La app deberá poder agregar y eliminar usuarios.

Deudas y Pagos

- **Si una persona paga algo en una reunión**, el sistema carga el gasto indicando quien pagó, cuánto, quienes participaron del encuentro y la app reparte los gastos.
- Cuando una persona va a una reunión y no es la que paga se le genera una **deuda** con el que pagó
 - Los gastos se dividen de dos maneras
 - **Partes iguales**
 - **Modo cumpleaños:** si una persona cumple años el día de la reunión, no paga nada.
 - A futuro se van a agregar más formas de dividir gastos

Modos de deudas y liquidaciones

- Los gastos pueden ser **urgentes**, si una persona adeuda gastos urgentes no podrá ir a reuniones hasta saldar esa deuda.

- A futuro se van a agregar más tipos de deudas
- **Simplificación de deudas**
- Se deben poder **liquidar deudas**, es decir, pagar lo que se debe.

Base de Datos

- Se deben mostrar los últimos eventos producidos (deudas y pagos)
- Si se selecciona un usuario se podrán ver todas sus deudas.
- Toda la info se guarda en una base de datos local
 - Se van a agregar más métodos de almacenamiento en un futuro
- Cada 1 minuto se hace backup de toda la base de datos en un txt llamado backup

Criterios del Cliente

- Se debe incluir una librería proporcionada por jay p walls en el sistema

Entrevista

- Deberá guardarse información de usuarios: nombre, apellido, dni.
- Por ahora una sola persona realiza el gasto.
- El historial debe mostrar todos los eventos ocurridos en la aplicación, incluyendo la generación de un nuevo gasto, las liquidaciones de deuda y cualquier información que consideremos pertinente. A la hora de simplificar deudas, podemos mostrar lo que creamos conveniente
- Si el cumpleaños tiene deuda urgente puede participar de gastos
- El usuario debe guardar su cumpleaños.
- Por ahora solo se evita pagar si el cumpleaños cumple años el mismo día que se realiza el gasto.
- Se guarda todo en la base de datos (usuarios, deudas, gastos)
- Se pueden cancelar deudas de manera parcial y total.
- Ni bien se registran los eventos estos se guardan en la base de datos.

Funcionalidades

1. **Agregar y eliminar usuarios.**

2. **Simplificación de deudas** en forma de bucle.

- a. Ej: Agustín -20\$-> Franco -30\$-> Johann -40\$-> Agustín Deberá borrarse la deuda de 20\$ y restarle 20\$ a las demás deudas eliminando el bucle Franco -10\$-> Johann -20\$-> Agustín

3. **Liquidación de deudas:**

- a. Se pueden liquidar deudas de manera parcial.
- b. Se pueden liquidar deudas de manera total.
- c. Si se liquida con precio mayor a la deuda, se liquida de manera total y se devuelve el vuelto al usuario.

4. **Creación de deudas a través de tres algoritmos:**

- a. Gasto empresarial
 - i. El usuario "Tech Fashion" paga el 50% del gasto.
 - ii. Para el resto del gasto, se aplica el algoritmo "Partes Iguales".
 - iii. No importan los cumpleaños, todos pagan por igual.
 - iv. Si algún participante posee una deuda urgente, se cancela el gasto.
- b. Partes Iguales
 - i. Se dividen los gastos por igual entre la cantidad de participantes.
 - ii. Si algún participante posee deudas urgentes:
 - 1. Si es su cumpleaños, participa del gasto pero no paga su parte.
 - 2. Si no es su cumpleaños, se cancela la reunión.
 - iii. Si algún participante cumple años, no pagará el gasto.

- iv. Si todos los participantes cumplen años, se cancela la reunión (ya que al menos un participante debe abonar el gasto)
- c. IVA
 - i. Se aplica un algoritmo de división provisto por la empresa.
 - ii. Si algún participante posee una deuda urgente, se cancela el gasto.
- 5. Se muestran **carteles de error** con la mayor representatividad posible intentando no vulnerar detalles de la implementación del software.
- 6. **Historial**
 - a. Se muestran todos los eventos ocurridos en la vida útil de la aplicación.

7. Botón Compartir

- a. Se agrega un botón compartir en distintas vistas de la interfaz para poder compartir información relevante hacia cualquier aplicación del dispositivo.

Links a Diagramas:

UML:

https://drive.google.com/file/d/1mzKra7xuX_NoqxqNcxVmjqG_XWBZInit/view?usp=sharing

UML simplificado:

<https://drive.google.com/file/d/1M7BEYjG6O7vpkfWwBdYJ6xXOdpcK0Zqp/view?usp=sharing>