Proyecto de Sistemas de Bases de Datos I

Sistema de administración de finanzas para el hogar

**Sistemas de Bases de Datos I**

##### Primer Termino 2021-2022

**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**Facultad de ingeniería en electricidad y computación**

**Índice**

[Integrantes 3](#_Toc73716941)

[Titulo del Proyecto 3](#_Toc73716942)

[Objetivo del Proyecto 3](#_Toc73716943)

[Objetivos Específicos 3](#_Toc73716945)

[Descripción General 3](#_Toc73716947)

[Descripciones Funcionales](#_Toc73716947) 4

[Modelo Conceptual 4](#_Toc73716949)

[Diccionario de Datos 4](#_Toc73716951)

[Modelo Lógico 4](#_Toc73716952)

[Flujo de Navegación 4](#_Toc73716954)

# Integrantes

* GUSTAVO ADOLFO CASTRO LINO
* RONALD XAVIER CABRERA PALACIOS
* FAUSTO SEBASTIAN JACOME ARGOTI

# Titulo del Proyecto

Sistema de administración de finanzas para el hogar: App FamilyWallet

**Etapa de Desarrollo: Análisis del Sistema**

# Objetivo del Proyecto

Desarrollar un sistema de bases de datos relacionado a administración de las finanzas de un hogar, de fácil aprendizaje enfocada en gestionar y planificar las finanzas de las familias a través del registro de datos como ingresos, gastos, presupuestos a corto y largo plazo.

### Etapa de Desarrollo: Análisis del Sistema

# Objetivos Específicos

- Desarrollar un balance de cuentas para que el cliente pueda mantenerse al día con sus finanzas.

- Mostrar al cliente en donde están gastando sus ingresos con el fin de que puedan corregir sus fugas de dinero.

- Graficar de diferentes formas los datos recibidos en el sistema para que el cliente pueda realizar un mejor análisis respecto a sus finanzas.

### Etapa de Desarrollo: Análisis del Sistema

# Descripción General

El siguiente proyecto de Sistema de Bases de datos busca crear una app que permita a un grupo de usuarios manejar sus finanzas de forma correcta y adecuada. La problemática se relaciona con el mal manejo del dinero y el bajo conocimiento respecto a educación financiera que existe entre las personas. Además, actualmente se vive en una era del consumismo donde lo que se piensa primordialmente es en gastar en lugar de ahorrar.

Este sistema va dirigido principalmente a las familias que no ahorran, ni planifican sus finanzas; por lo que necesitan ayuda para controlar sus ingresos y gastos. También, este sistema es de utilidad para aquellas personas que siguen controlando sus finanzas de modos cotidianos (cuaderno, notas). En la app se desea implementar múltiples opciones que permitan al usuario manejar las finanzas no solo del jefe del hogar, sino de todos los miembros de la familia y que todos sean capaces de observar sus balances.

Con todo esto, se espera hacer conciencia sobre el manejo del dinero, el cual es un tema muy importante hoy en día. Con la cantidad de problemas que existen en todo el mundo, la economía global se está viendo afectado, por lo que es recomendable controlar la economía del hogar y prever gastos futuros.

Roles de usuario:

Desarrollador del sistema, conocimiento pleno de la aplicación.

Jefe del hogar, administrador de la aplicación, tendrá funciones desbloqueadas que otros usuarios no tienen.

Hijo/Familiar, usuario general, tiene funciones limitadas dentro de la aplicación.

Especificaciones del Sistema: almacenamiento local.

Sistemas externos: Money Pro - Finanzas Personales, Fintonic, Monefy, Mobills.

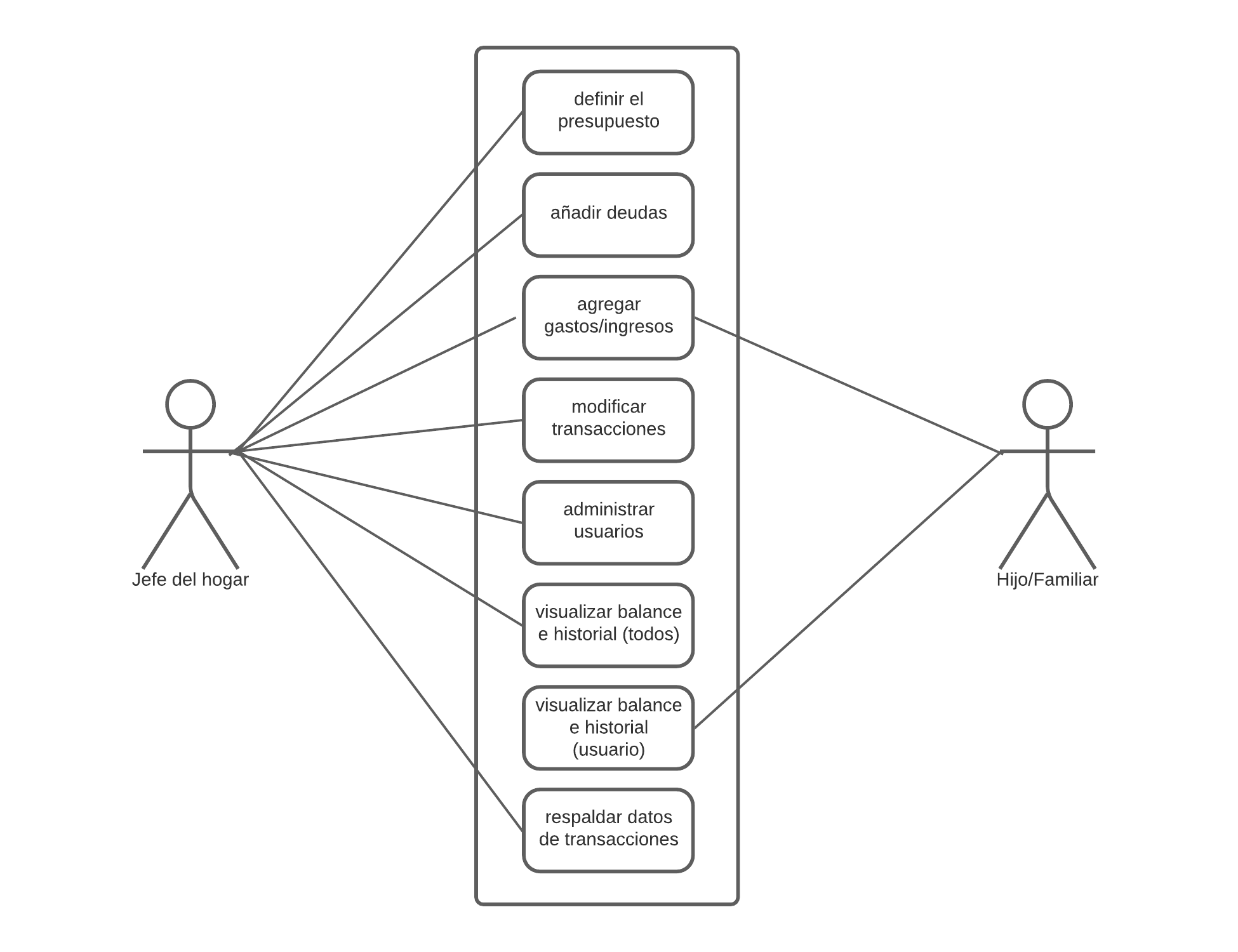
Tipos de seguridad (segundo parcial)

Transacciones críticas y frecuentes (segundo parcial)

### Etapa de Desarrollo: Análisis del Sistema

# Descripciones Funcionales

Diagrama de Caso de Uso



1. Definir los procesos especificados en el desarrollo del proyecto.

* Definir el presupuesto
* Añadir deudas
* Agregar gastos/ingresos
* Modificar transacciones
* Administrar usuarios
* Visualizar balance e historial
* Respaldar datos de transacciones
* añadir cuentas

1. Identificar cada uno de los diferentes roles de las personas y sistemas externos involucrados en el proceso.

* Jefe de hogar (tiene todas las funciones desbloqueadas de la app): es el administrador de la app, quien está encargado de definir el presupuesto familiar y asignar mensualidades a los hijos. También podrá tomar decisiones basado en los balances para restructurar el presupuesto familiar según sus objetivos o posibilidades.
* Hijo/familiar (tiene funciones limitadas de la app): podrá usar la mesada, recibir ingresos y hacer gastos, cuyas transacciones serán tomadas en cuenta para el análisis del uso del presupuesto familiar

1. Detallar cada uno de los procesos (que no se escape nada) siga las siguientes definiciones.
2. **Nombre:** definir el presupuesto

**Descripción:** este proceso permite al cliente definir su presupuesto con diferentes opciones (diario, semanal, mensual o anual).

**Entrada:** elegir plazo (diario, semanal, anual, mensual) -> cantidad de presupuesto

**Salida:**

mensaje de éxito: “Se ingresó correctamente su presupuesto”

1. **Nombre:** añadir deudas

**Descripción**: este proceso permite al cliente agregar deudas, definir sus plazos y la cantidad a pagar en cada plazo; además de la deuda total.

**Entrada**: deuda total, nombre de la deuda, plazo de la deuda, cantidad a pagar en cada plazo, interés.

**Salida**:

mensaje de éxito: “Se ingresó correctamente su deuda”  
mensaje de error: “La cantidad a pagar en plazos no debe ser mayor a la deuda total”

mensaje de advertencia: “El nombre de la deuda ya existe. ¿Quisiera cambiar el nombre?”

1. **Nombre:** agregar gastos/ingresos

**Descripción:** este proceso permite al cliente agregar gastos/ingresos, además de dividirlo en tipos (transporte, trabajo, comida, etc.).

**Entrada:** elegir gasto/ingreso -> nombre de la transacción, cantidad de la transacción, fecha de la transacción

**Salida:**

Mensaje de éxito: “Se ingresó correctamente su gasto”

Mensaje de éxito: “Se ingresó correctamente su ingreso”

1. **Nombre:** añadir cuentas

**Descripción:** este proceso permite al cliente añadir sus cuentas bancarias si tienen.

**Entrada:** usuario, cédula, correo, número de cuenta.

**Salida:**

Mensaje de éxito: “Se ingresó correctamente su cuenta”

1. **Nombre:** modificar transacciones

**Descripción:** este proceso permite al cliente modificar algún gasto de las transacciones ya ingresadas.

**Entrada:** elegir transacción -> cambiar nombre, cambiar cantidad

**Salida:**

Mensaje de éxito: “Su transacción ha sido modificada correctamente”

1. **Nombre:** administrar usuarios

**Descripción:** este proceso permite al cliente agregar, quitar, modificar a los usuarios dentro del sistema.

**Entrada:** nombre usuario, cargo (jefe de hogar, hijo/familiar), edad

**Salida:**

Mensaje de éxito: “Se agregó correctamente el usuario”

Mensaje de error: “Ya existe un usuario con ese nombre”

Mensaje de advertencia: “Ya existe un usuario con el cargo Jefe de Hogar. ¿Está seguro de agregar otro?”

1. **Nombre:** visualizar balance e historial

**Descripción:** este proceso permite al cliente visualizar su balance e historial. En el caso del jefe del hogar (administrador) podrá visualizar todos los balances de todos los usuarios, mientras que el del hijo/familiar, solo visualizará el de él mismo.

**Entrada:** nombre usuario a consultar, rango de fechas a consultar

**Salida:** se generará una tabla que contenga la siguiente información. Además, se presentará un reporte gráfico de su balance a lo largo del tiempo

|  |  |  |  |
| --- | --- | --- | --- |
| transacción | tipo | cantidad | fecha |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. **Nombre:** respaldar datos de transacciones

**Descripción:** este proceso permite al cliente guardas los datos de sus transacciones dentro del sistema de almacenamiento de su dispositivo o compartirlo a un almacenamiento en la nube.

**Entrada:** nombre usuario a respaldar, rango de fechas a respaldar

**Salida:**

Mensaje de éxito: “Sus transacciones han sido exportadas correctamente a la raíz de su dispositivo”

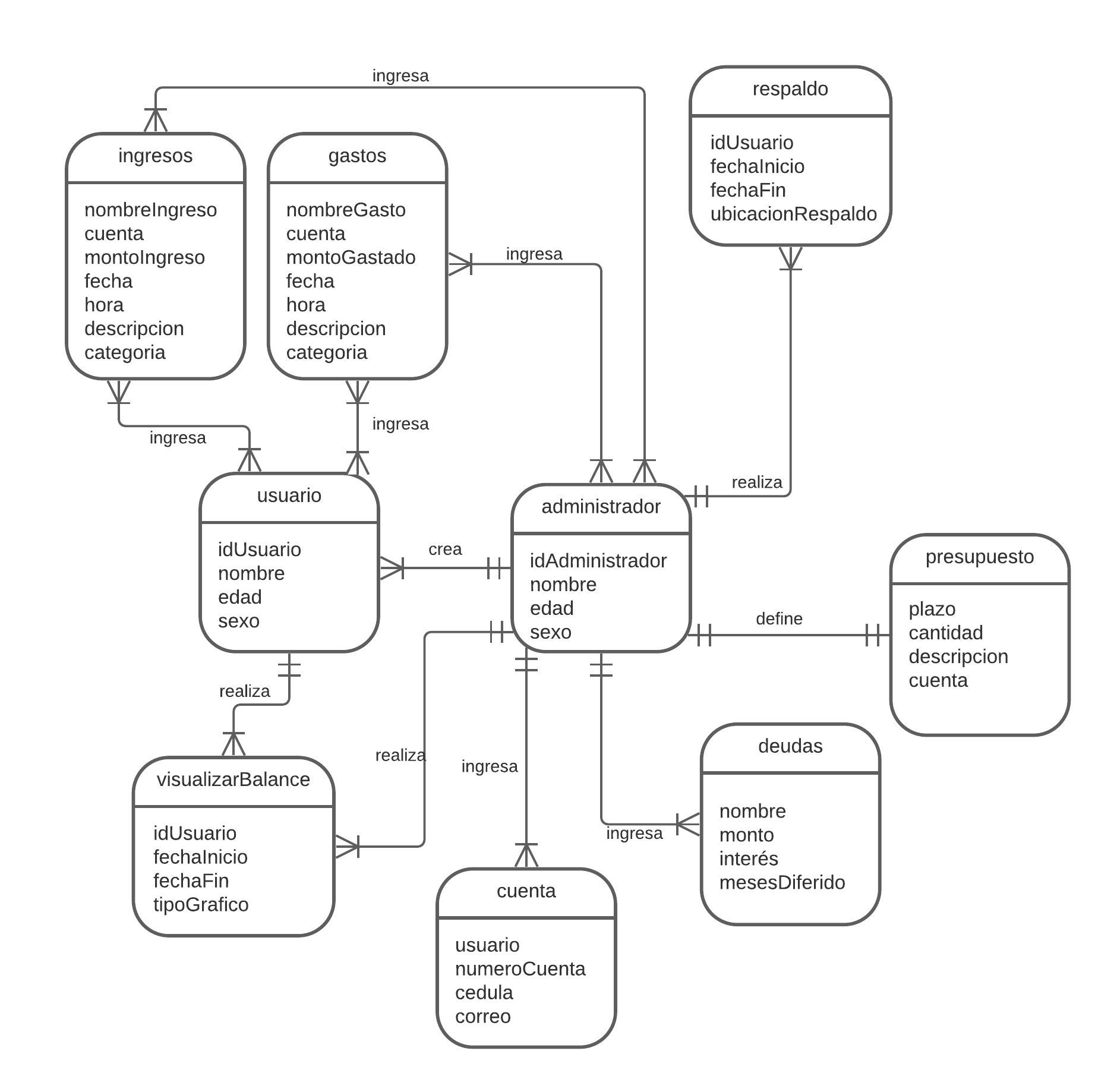
Mensaje de error: “Sin conexión a internet. No se pudo exportar sus transacciones a la nube”

### Etapa de Desarrollo: Análisis del Sistema

# Modelo Conceptual

Presentación del Modelo Conceptual(diagrama)

### Etapa de Desarrollo: Diseño del Sistema



# Diccionario de Datos

Se debe de detallar cada uno de los diferentes elementos del modelo lógicos, siga las siguientes definiciones.

**Tabla:** Concepto del termino en el proceso

**Columna 1:** Especificación del uso/Dominio de los valores

**Columna 2:** Especificación del uso/Dominio de los valores

|  |
| --- |
| **Tabla Administrador** |
| Atributo | Tipo de dato | Descripción | Dominio de los valores |
| idAdministrador | String | Contiene caracteres alfanuméricos | Identificador único del administrador para el ingreso al sistema |
| nombre | String | Contiene caracteres de la A-Z y espacios | Contiene los nombres y apellidos del administrador |
| edad | int | Contiene caracteres numéricos tamaño 2 | Contiene la edad del administrador |
| sexo | String | Contiene caracteres de la A-Z y espacios | Contiene el sexo del administrador |

|  |
| --- |
| **Tabla Usuario** |
| Atributo | Tipo de dato | Descripción | Dominio de los valores |
| idUsuario | String | Contiene caracteres alfanuméricos | Identificador único del usuario para el ingreso al sistema |
| idAdministrador | String | Contiene caracteres alfanuméricos | Identificador único del administrador para el ingreso al sistema |
| nombre | String | Contiene caracteres de la A-Z y espacios | Contiene los nombres y apellidos del usuario |
| edad | int | Contiene caracteres numéricos tamaño 2 | Contiene la edad del usuario |
| sexo | String | Contiene caracteres de la A-Z y espacios | Contiene el sexo del usuario |

|  |
| --- |
| **Tabla Cuenta** |
| Atributo | Tipo de dato | Descripción | Dominio de los valores |
| idUsuario | String | Contiene caracteres alfanuméricos | Identificador único del usuario para el ingreso al sistema |
| numeroCuenta | String | Contiene caracteres numéricos tamaño 16 | Contiene el número de cuenta del usuario |
| cedula | String | Contiene caracteres numéricos tamaño 10 | Contiene la cedula del usuario |
| correo | String | Contiene caracteres alfanuméricos | Contiene el correo del usuario propietario de la cuenta |

|  |
| --- |
| **Tabla Deuda** |
| Atributo | Tipo de dato | Descripción | Dominio de los valores |
| idDeuda | String | Contiene caracteres alfanuméricos | Identificador único de la deuda. |
| idAdministrador | String | Contiene caracteres alfanuméricos | Identificador único del administrador para el ingreso al sistema |
| numeroCuenta | String | Contiene caracteres numéricos tamaño 16 | Contiene el número de cuenta del usuario |
| nombre | String | Contiene caracteres de la A-Z y espacios | Contiene el nombre de la deuda |
| monto | float | Números decimales que representan dinero | Contiene el monto mensual a pagar por la deuda |
| interes | int | Contiene caracteres numéricos tamaño 2 | Contiene el interés a pagar por mes atrasado |
| mesesDiferido | int | Contiene caracteres numéricos tamaño 2 | Contiene los meses a los que se difirió la deuda |

|  |
| --- |
| **Tabla Presupuesto** |
| Atributo | Tipo de dato | Descripción | Dominio de los valores |
| idPresupuesto | String | Contiene caracteres alfanuméricos | Identificador único del presupuesto. |
| idAdministrador | String | Contiene caracteres alfanuméricos | Identificador único del administrador para el ingreso al sistema |
| numeroCuenta | String | Contiene caracteres numéricos tamaño 16 | Contiene el número de cuenta del usuario |
| plazo | String | Contiene caracteres de la A-Z y espacios | Define el plazo del presupuesto (anual, mensual, semanal) |
| cantidad | float | Números decimales que representan dinero | Contiene la cantidad presupuestada |
| descripcion | String | Contiene caracteres de la A-Z y espacios | Contiene la descripción del presupuesto |
| cuenta | int | Contiene caracteres numéricos tamaño 10 | Contiene el número de cuenta del usuario que hace el presupuesto |

|  |
| --- |
| **Tabla Respaldo** |
| Atributo | Tipo de dato | Descripción | Dominio de los valores |
| idRespaldo | String | Contiene caracteres alfanuméricos | Identificador único del respaldo |
| idUsuario | String | Contiene caracteres alfanuméricos | Identificador único del usuario para el ingreso al sistema |
| fechaInicio | Date | Formato fecha | Contiene la fecha desde donde se realizará el respaldo |
| fechaFin | Date | Formato fecha | Contiene la fecha hasta la que se realizará el respaldo |
| ubicacionRespaldo | String | Contiene caracteres alfanuméricos | Contiene la ubicación de guardado del respaldo |

|  |
| --- |
| **Tabla VisualizarBalance** |
| Atributo | Tipo de dato | Descripción | Dominio de los valores |
| idBalance | String | Contiene caracteres alfanuméricos | Identificador único del visualizador de balance |
| idUsuario | String | Contiene caracteres alfanuméricos | Identificador único del usuario para el ingreso al sistema |
| fechaInicio | Date | Formato fecha | Contiene la fecha desde donde se realizará la consulta |
| fechaFin | Date | Formato fecha | Contiene la fecha hasta la que se realizará la consulta |
| tipoGrafico | String | Contiene caracteres alfanuméricos | Contiene el tipo de gráfico donde se expondrán los datos |

|  |
| --- |
| **Tabla Ingresos** |
| Atributo | Tipo de dato | Descripción | Dominio de los valores |
| idIngreso | String | Contiene caracteres alfanuméricos | Identificador único de la transacción al ingresar dinero |
| nombreIngreso | String | Contiene caracteres de la A-Z y espacios | Contiene el nombre del gasto |
| cuenta | int | Contiene caracteres numéricos tamaño 16 | Contiene el número de cuenta del usuario |
| montoIngreso | float | Números decimales que representan dinero | Contiene el monto que se ingresará |
| fecha | Date | Formato fecha | Contiene la fecha de la transacción |
| hora | Date | Formato hora | Contiene la hora de la transacción |
| descripción | String | Contiene caracteres de la A-Z y espacios | Contiene la descripción de la transacción |
| categoria | String | Contiene caracteres de la A-Z y espacios | Contiene la categoría de la transacción |

|  |
| --- |
| **Tabla Gastos** |
| Atributo | Tipo de dato | Descripción | Dominio de los valores |
| idGasto | String | Contiene caracteres alfanuméricos | Identificador único de la transacción al realizar un gasto |
| nombreGasto | String | Contiene caracteres de la A-Z y espacios | Contiene el nombre del gasto |
| cuenta | int | Contiene caracteres numéricos tamaño 16 | Contiene el número de cuenta del usuario |
| montoGasto | float | Números decimales que representan dinero | Contiene el monto que se restará |
| fecha | Date | Formato fecha | Contiene la fecha de la transacción |
| hora | Date | Formato hora | Contiene la hora de la transacción |
| descripción | String | Contiene caracteres de la A-Z y espacios | Contiene la descripción de la transacción |
| categoria | String | Contiene caracteres de la A-Z y espacios | Contiene la categoría de la transacción |

****

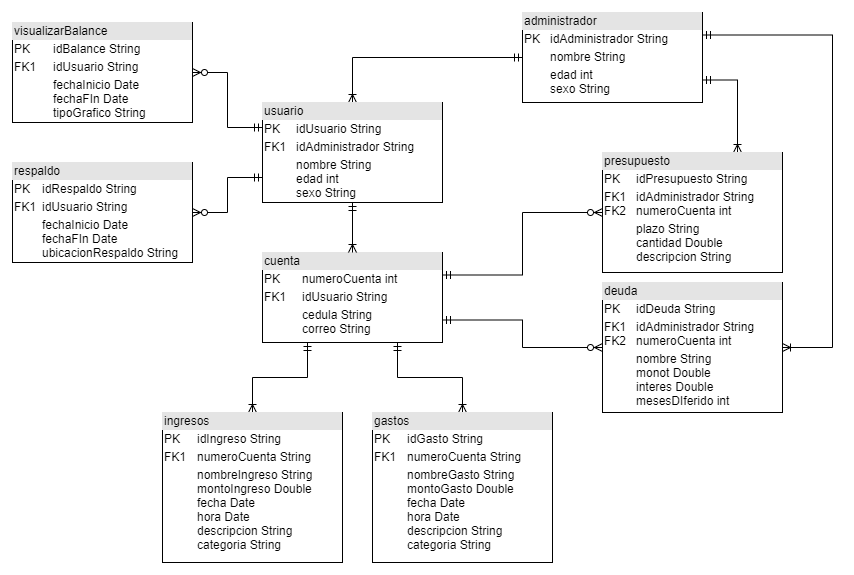
Ejemplo del libro

#### Etapa de Desarrollo: Diseño del Sistema

# Modelo Lógico

Presentación del modelo Lógico(diagrama)

### Etapa de Desarrollo: Diseño del Sistema



# Consultas del sistema

* **Buscar por categoría los ingresos que se han hecho y mostrar el nombre del usuario, el número de cuenta y el monto del ingreso que se realizó.**

π nombre, numeroCuenta, montoIngreso (σ categoría = *“categoría ingresada”* (usuario ⨝ (ingresos ⨝ cuenta)))

* **Buscar por categoría los gastos que se han hecho y mostrar el nombre del usuario, el número de cuenta y el monto del gasto que se realizó.**

π nombre, numeroCuenta, montoGasto (σ categoría = *“categoría ingresada”* (usuario ⨝ (gastos ⨝ cuenta)))

* **Buscar entre un rango de fechas los ingresos que se han realizado y mostrar el nombre, número de cuenta, categoría y monto del ingreso.**

π nombre, numeroCuenta, categoría, montoIngreso (σ fecha = *“202x-xx-xx”* ∧ fecha = *“202x-xx-xx”* (usuario ⨝ (ingresos ⨝ cuenta)))

* **Buscar entre un rango de fechas los gastos que se han realizado y mostrar el nombre, número de cuenta, categoría y monto del gasto.**

π nombre, numeroCuenta, categoría, montoGasto (σ fecha = *“202x-xx-xx”* ∧ fecha = *“202x-xx-xx”* (usuario ⨝ (gastos ⨝ cuenta)))

* **Mostrar el ingreso total que una cuenta adquirió en un rango de fechas.**

T1 = γ numeroCuenta; sum(montoIngreso) → totalIngreso (ingresos)

π totalIngreso (σ fecha = *“202x-xx-xx”* ∧ fecha = *“202x-xx-xx”* ∧ numeroCuenta = xxxxxxxxxx (T1))

* **Mostrar el gasto total que una cuenta realizó en un rango de fechas.**

T1 = γ numeroCuenta; sum(montoGasto) → totalGasto (gastos)

π totalGasto (σ fecha = *“202x-xx-xx”* ∧ fecha = *“202x-xx-xx”* ∧ numeroCuenta = xxxxxxxxxx (T1))

* **Mostrar el nombre, monto, interés y los meses en los que se ha diferido las deudas de un determinado número de cuenta.**

π nombre, monto, interés, mesesDiferido (σ numeroCuenta = xxxxxxxxxx (deuda))

* **Mostrar el nombre, monto, interés, los meses en los que se ha diferido y el número de cuenta de la deuda con el interés más alto.**

T1 = γ max(interes) → interesMax (deuda)

π nombre, numeroCuenta, monto, interés, mesesDiferido (σ interés = interesMax (deuda ⨝ T1))

* **Mostrar el nombre, monto, interés, los meses en los que se ha diferido y el número de cuenta de la deuda con el interés más bajo.**

T1 = γ min(interes) → interesMin (deuda)

π nombre, numeroCuenta, monto, interés, mesesDiferido (σ interés = interesMin (deuda ⨝ T1))

* **Mostrar los números de cuenta de los usuarios que tienen exactamente 18 años.**

π numeroCuenta (σ edad = 18 (usuario ⨝ cuenta))

* **Mostrar el nombre, la descripción y el monto de los ingresos mayores a 100$.**

π nombreIngreso, descripción, montoIngreso (σ montoIngreso > 100 (ingresos))

* **Mostrar el nombre, la descripción y el monto de los gastos mayores a 100$.**

π nombreGasto, descripción, montoGasto (σ montoGasto > 100 (gastos))

* **Mostrar el id y el nombre del usuario al que le pertenece un determinado número de cuenta.**

π idUsuario, nombre (σ numeroCuenta = xxxxxxxxxx (usuario ⨝ cuenta))

* **Mostrar la cantidad de ingresos que se hicieron en un mismo día por parte de una cuenta determinada.**

T1 = γ numeroCuenta, fecha; count(idIngreso) → numeroIngresos (ingresos)

π numeroIngreso (σ numeroCuenta = xxxxxxxxxx ∧ fecha = *“202x-xx-xx”* (T1))

* **Mostrar la cantidad de gastos que se hicieron en un mismo día por parte de una cuenta determinada.**

T1 = γ numeroCuenta, fecha; count(idGasto) → numeroGastos (gastos)

π numeroGastos (σ numeroCuenta = xxxxxxxxxx ∧ fecha = *“202x-xx-xx”* (T1))

**Calificación (19/25)**

Descripción general **(1/3)**

Procesos **(4/5)**

Modelo conceptual **(3/4)**

Modelo lógico **(6/7)**

Diccionario de datos **(2.5/3)**

Redacción y ortografía **(2.5/3)**

La descripción debía ser como un modelo de negocio. Eso lo dijo la miss, y también se lo dijo en la retroalimentación de proyectos. Incluso hubieron grupos que pidieron hacer otra reunión.

Han llenado la información al revés en el diccionario de datos. Lo de Descripción va en Dominio y lo de Dominio en Descripción.

En la nota de Redacción y ortografía también influye la presentación, eso incluye el hecho de no borrar información innecesaria. Recordar que esto es un documento formal. Además se han encontrado 3 tipos de letras diferentes.

Aún les falta identificar procesos de eliminación, y modificación para presupuesto, deudas, cuenta.

Los procesos deben ser más específicos, no traten de contener varios procesos en uno solo. Por ejemplo: Los que se encuentran en Administrar usuario deben estar separados ya que no todos requerirán las mismas entradas ni tendrán las mismas salidas.

Por último, teniendo en cuenta las tablas de sus diagramas, veo que hay algunas tablas que están basadas en VENTANAS de la aplicación, eso no es así. El diagrama/modelo relacional se enfoca en datos que van a guardarse en la base de datos, y a partir de estos por ejemplo crear el balance. Ya que hay cosas que se establecen directamente en la aplicación, pero lo que necesitan poner en el diagrama son los datos que utilizarán para hacerlo. Me hubiese gustado explicarles todo esto en la retroalimentación, lo ideal era que vayan todos, pero si no se organizaban, por lo menos uno que sea responsable y asista para darle un buen rumbo la lógica de sus modelos.

**Modelo conceptual (10/10)**

**Tablas (3/5)**

**Relaciones (2.5/3)**

**Cardinalidades (1/2)**

Al administrador, a pesar de requerirse su información y tenerlo en los procesos casi no se lo tiene que tomar en cuenta para las RELACIONES del diagrama, puesto que ingresar/actualizar/eliminar es una acción externa.

Si quisieran relacionar al Administrador/es se necesitaría información de contacto en caso de que exista algún fallo con su app y se podría decir que la cardinalidad de administrador y usuario es N:N.

La información de cedula y correo puede almacenarse en Usuario en vez de Cuenta. La cardinalidad de Usuario y Cuenta sería 1:N por ende la FK estaría en usuario. Y en la tabla Cuenta podrían quedar los atributos num\_cta, tipo (corriente/ahorro), id\_Banco. Ya que necesitaríamos también una lista de nombres de Bancos en una tabla aparte.

En vez de llamarles nombres ponerles id para que se entienda mejor, ya que nombreIngreso se podría pensar que hablan de su descripción.

Si ponen el atributo cuenta en la tabla Ingresos y Gastos deberían relacionarlo con la tabla Cuenta.

La tabla Respaldo debería estar relacionada directamente con Usuario, lo mismo para Deudas.

**Modelo lógico (12/15)**

**Tablas (5/8)**

**Relaciones (2.5/3)**

**Cardinalidades (3/4)**

Presentado atrasado.

Tener en cuenta la retro alimentación del modelo conceptual antes descrita. presupuesto, deuda, ingresos, gastos son de usuario no de cuenta.

También tener en cuenta que el Administrador no es el Sistema como tal. Así que las cosas que sucedan automáticamente son del programa en sí y eso no se pone en el diagrama.

Por otro lado las ventanas tampoco se ponen en una base de datos, como Visualizar Balance.

Los nombres de las tablas deben ser sustantivos no verbos.