Proyecto de Sistemas de Bases de Datos I

Sistema de administración de finanzas para el hogar

**Sistemas de Bases de Datos I**

##### Primer Término 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 8](#_Toc73716949)

[Diccionario de Datos 9](#_Toc73716951)

[Modelo Lógico 11](#_Toc73716952)

# Integrantes

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

# Título del Proyecto

Sistema de administración de finanzas para el hogar: App FamilyWallet

# 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.

# 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.

# Descripción General

La mayoría de personas de nuestra sociedad tiene un escaso conocimiento con respecto a educación financiera. Además, actualmente se vive en una era del consumismo donde lo que se piensa primordialmente es en gastar en lugar de ahorrar. Es por ello que se da la creación de este sistema, el cual permitirá a las familias controlar sus finanzas de una mejor forma.

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.

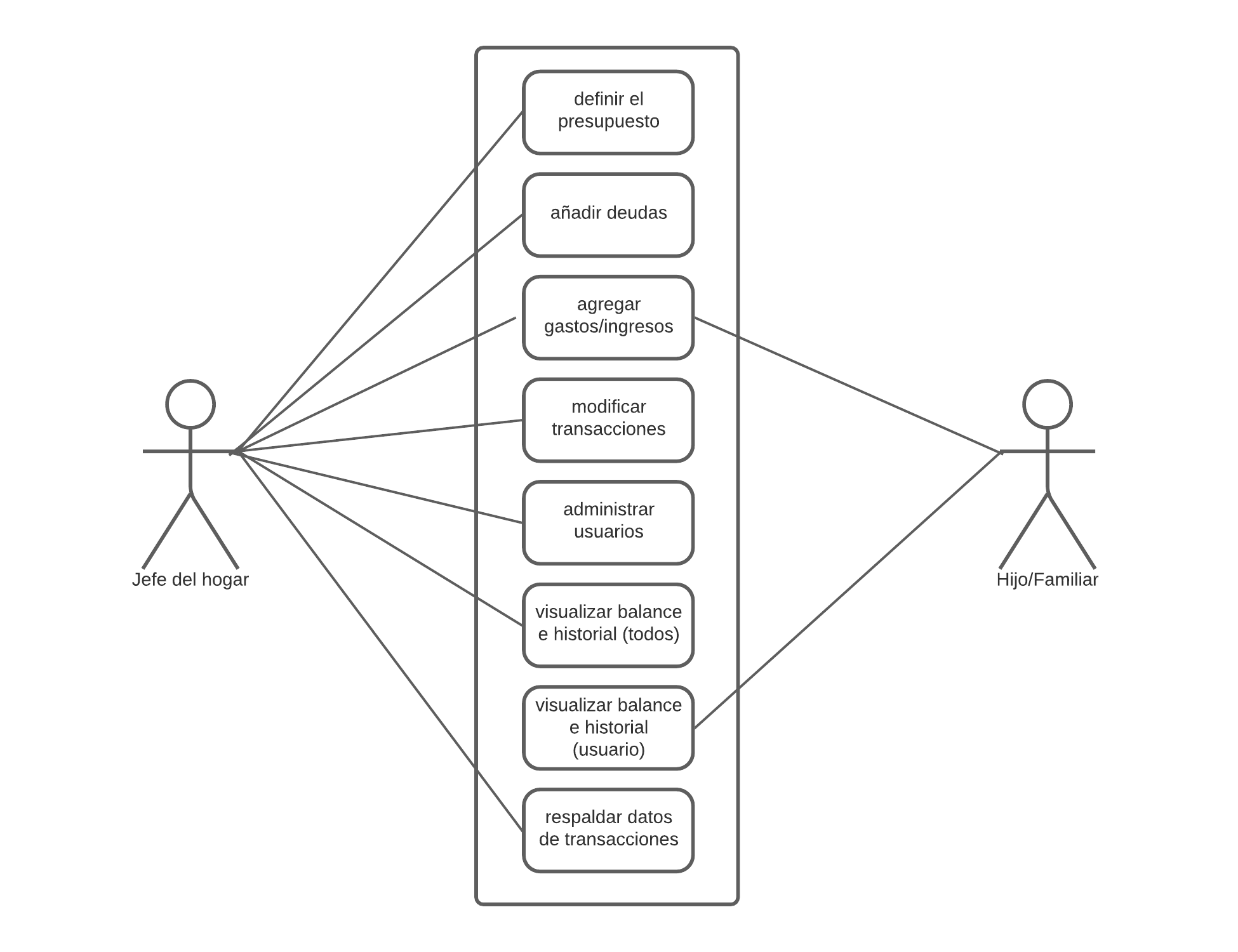
Hijo/Familiar, usuario general, tiene funciones limitadas dentro de la aplicación.

Especificaciones del Sistema: almacenamiento local.

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

# 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 los presupuestos

**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:** modificar los presupuestos

**Descripción:** este proceso permite al cliente modificar su presupuesto que ya ha sido creado

**Entrada:** seleccionar presupuesto -> cantidad de presupuesto

**Salida:**

mensaje de éxito: “Se modificó correctamente su presupuesto”

mensaje de error: “ID de presupuesto no existe”

1. **Nombre:** quitar presupuesto

**Descripción:** este proceso permite al cliente eliminar un presupuesto que ya ha sido creado

**Entrada:** seleccionar presupuesto -> eliminar el presupuesto

**Salida:**

mensaje de éxito: “Se eliminó correctamente su presupuesto”

mensaje de error: “ID de presupuesto no existe”

1. **Nombre:** definir las 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:** modificar las deudas

**Descripción:** este proceso permite al cliente modificar sus deudas que ya han sido creadas

**Entrada:** seleccionar deuda -> cantidad de deuda

**Salida:**

mensaje de éxito: “Se modificó correctamente su deuda”

mensaje de error: “ID de deuda no existe”

1. **Nombre:** quitar deuda

**Descripción:** este proceso permite al cliente eliminar una deuda que ya ha sido creada

**Entrada:** seleccionar deuda -> eliminar la deuda

**Salida:**

mensaje de éxito: “Se eliminó correctamente su deuda”

mensaje de error: “ID de deuda no existe”

1. **Nombre:** agregar gastos

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

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

**Salida:**

Mensaje de éxito: “Se ingresó correctamente su gasto”

1. **Nombre:** agregaringresos

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

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

**Salida:**

Mensaje de éxito: “Se ingresó correctamente su ingreso”

1. **Nombre:** modificar transacciones

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

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

**Salida:**

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

Mensaje de error: “ID gasto no existe”

Mensaje de error: “ID ingreso no existe”

1. **Nombre:** eliminar transacciones

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

**Entrada:** elegir transacción -> eliminar transacción

**Salida:**

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

Mensaje de error: “ID gasto no existe”

Mensaje de error: “ID ingreso no existe”

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:** eliminar cuentas

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

**Entrada:** elegir cuenta -> eliminar cuenta

**Salida:**

Mensaje de éxito: “Se eliminó correctamente su cuenta”

1. **Nombre:** agregarusuario

**Descripción:** este proceso permite al cliente agregar 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:** modificar usuario

**Descripción:** este proceso permite al cliente modificar usuarios dentro del sistema.

**Entrada:** elegir usuario -> cambiar nombre, cambiar cargo

**Salida:**

Mensaje de éxito: “Su usuario ha sido modificado correctamente”

Mensaje de error: “ID usuario no existe”

1. **Nombre:** eliminar usuario

**Descripción:** este proceso permite al cliente eliminar usuarios dentro del sistema.

**Entrada:** elegir usuario -> eliminar usuario

**Salida:**

Mensaje de éxito: “Su usuario ha sido eliminado correctamente”

Mensaje de error: “ID usuario no existe”

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”

# Modelo Conceptual



# Diccionario de Datos

|  |  |  |
| --- | --- | --- |
| **Tabla Usuario** |  | |
| Atributo | Tipo de dato | Descripción | | Dominio de los valores |
| idUsuario | String | Identificador único del usuario para el ingreso al sistema | | Contiene caracteres alfanuméricos |
| numeroCuenta | String | Contiene el número de cuenta del usuario | | Contiene caracteres numéricos tamaño 16 |
| nombre | String | Contiene los nombres y apellidos del usuario | | Contiene caracteres de la A-Z y espacios |
| cedula | String | Contiene la cedula del usuario | | Contiene caracteres numéricos tamaño 10 |
| correo | String | Contiene el correo del usuario propietario de la cuenta | | Contiene caracteres alfanuméricos |
| edad | int | Contiene la edad del usuario | | Contiene caracteres numéricos tamaño 2 |

|  |  |  |
| --- | --- | --- |
| **Tabla Cuenta** |  | |
| Atributo | Tipo de dato | Descripción | | Dominio de los valores |
| numeroCuenta | String | Contiene el número de cuenta del usuario | | Contiene caracteres numéricos tamaño 16 |
| tipoCuenta | String | Contiene el tipo de cuenta del usuario | | Contiene caracteres de la A-Z |
| idBanco | String | Identificador único del banco para el ingreso al sistema | | Contiene caracteres alfanuméricos |

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Tabla Presupuesto** |  | |  |
| Atributo | Tipo de dato | Descripción | | Dominio de los valores |
| idPresupuesto | String | Identificador único del presupuesto. | | Contiene caracteres alfanuméricos |
| idUsuario | String | Identificador único del usuario para el ingreso al sistema | | Contiene caracteres alfanuméricos |
| monto | float | Contiene el monto del presupuesto | | Números decimales que representan dinero |
| plazo | int | Contiene el plazo del presupuesto | | Contiene caracteres de la A-Z |
| descripcion | String | Contiene la descripción de la deuda | | Contiene caracteres de la A-Z y espacios |

|  |  |  |
| --- | --- | --- |
| **Tabla Banco** |  | |
| Atributo | Tipo de dato | Descripción | | Dominio de los valores |
| idBanco | String | Identificador único del banco | | Contiene caracteres alfanuméricos |
| nombre | String | Contiene el nombre del banco | | Contiene caracteres de la A-Z y espacios |

|  |  |  |
| --- | --- | --- |
| **Tabla Respaldo** |  | |
| Atributo | Tipo de dato | Descripción | | Dominio de los valores |
| idRespaldo | String | Identificador único del respaldo | | Contiene caracteres alfanuméricos |
| idUsuario | String | Identificador único del usuario para el ingreso al sistema | | Contiene caracteres alfanuméricos |
| numeroCuenta | String | Contiene el número de cuenta del usuario | | Contiene caracteres numéricos tamaño 16 |
| fechaInicio | Date | Contiene la fecha desde donde se realizará el respaldo | | Formato fecha |
| fechaFin | Date | Contiene la fecha hasta la que se realizará el respaldo | | Formato fecha |
| ubicacionRespaldo | String | Contiene la ubicación de guardado del respaldo | | Contiene caracteres alfanuméricos |

|  |  |  |
| --- | --- | --- |
| **Tabla Ingresos** |  | |
| Atributo | Tipo de dato | Descripción | | Dominio de los valores |
| idIngreso | String | Identificador único de la transacción al ingresar dinero | | Contiene caracteres alfanuméricos |
| idUsuario | String | Identificador único del usuario para el ingreso al sistema | | Contiene caracteres alfanuméricos |
| monto | float | Contiene el monto que se ingresará | | Números decimales que representan dinero |
| fecha | Date | Contiene la fecha de la transacción | | Formato fecha |
| hora | Date | Contiene la hora de la transacción | | Formato hora |
| descripción | String | Contiene la descripción de la transacción | | Contiene caracteres de la A-Z y espacios |
| categoria | String | Contiene la categoría de la transacción | | Contiene caracteres de la A-Z y espacios |

|  |  |  |
| --- | --- | --- |
| **Tabla Gastos** |  | |
| Atributo | Tipo de dato | Descripción | | Dominio de los valores |
| idGasto | String | Identificador único de la transacción al gastar dinero | | Contiene caracteres alfanuméricos |
| idUsuario | String | Identificador único del usuario para el ingreso al sistema | | Contiene caracteres alfanuméricos |
| monto | float | Contiene el monto que se restará | | Números decimales que representan dinero |
| fecha | Date | Contiene la fecha de la transacción | | Formato fecha |
| hora | Date | Contiene la hora de la transacción | | Formato hora |
| descripción | String | Contiene la descripción de la transacción | | Contiene caracteres de la A-Z y espacios |
| categoria | String | Contiene la categoría de la transacción | | Contiene caracteres de la A-Z y espacios |

# 

# Modelo Lógico



# 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, numCuenta, monto (σ categoría = *“categoría ingresada”* (usuario ⨝ ingreso))

* **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, numCuenta, monto (σ categoría = *“categoría ingresada”* (usuario ⨝ gasto))

* **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, numCuenta, categoría, monto (σ fecha = *“202x-xx-xx”* ∧ fecha = *“202x-xx-xx”* (usuario ⨝ ingreso))

* **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, numCuenta, categoría, monto (σ fecha = *“202x-xx-xx”* ∧ fecha = *“202x-xx-xx”* (usuario ⨝ gasto))

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

T1 = γ numeroUsuario; sum(monto) → totalIngreso (ingreso)

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

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

T1 = γ numeroUsuario; sum(monto) → totalGasto (gasto)

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

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

π nombre, monto, interés, plazo (σ idUsuario = xxxx (deuda))

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

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

π descripcion, monto, interés, plazo (σ interés = interesMax (deuda ⨝ T1))

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

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

π descripcion, monto, interés, plazo (σ interés = interesMin (deuda ⨝ T1))

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

π numCuenta (σ edad = 18 (usuario))

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

π idIngreso, descripción, monto (σ monto > 100 (ingreso))

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

π idGasto, descripción, monto (σ monto > 100 (gasto))

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

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

* **Mostrar la cantidad de ingresos que se hicieron en un mismo día por parte de un usuario determinado.**

T1 = γ idUsuario, fecha; count(idIngreso) → numeroIngresos (ingreso)

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

* **Mostrar la cantidad de gastos que se hicieron en un mismo día por parte de un usuario determinado.**

T1 = γ idUsuario, fecha; count(idGasto) → numeroGastos (gasto)

π numeroGasto (σ idUsuario = xxxx ∧ fecha = *“202x-xx-xx”* (T1))