

# Universidad ORT Uruguay

## Licenciatura en Sistemas

### Diseño de Aplicaciones I - Obligatorio I



Micaela Felder 212322



Paula Hernandez 201886



Sara Vila 163497

Repositorio:

<https://github.com/ORT-DA1/212322-201886-163497-oblig1.git>

# Índice

Descripción general del trabajo y del sistema.....	3
Funcionalidades no implementadas.....	3
Errores conocidos.....	3
Descripción y justificación de diseño:.....	4
Diagrama de paquetes: Dominio, Interfaz de usuario,Excepciones.....	4
Diagramas de clases.....	4
Dominio.....	4
Interfaz de usuario.....	5
Excepciones.....	5
Explicación de los mecanismos generales y descripción de las principales decisiones de diseño tomadas.....	6
Abstraccion.....	6
Encapsulamiento.....	6
Modularización.....	6
Herencia.....	6
Interacción Interfaz de Usuario-Dominio.....	7
Manejo de Excepciones y Errores.....	7
Breve análisis de los criterios seguidos para asignar responsabilidades.....	7
Cobertura de pruebas unitarias con su debido análisis y justificaciones.....	8

## 1. Descripción general del trabajo y del sistema.

El proyecto trata de un sistema de finanzas personales, el cual es recomendado por expertos financieros. Este sistema facilita llevar un registro controlado, gráfico y detallado de los gastos fijos en cada mes, gastos realizados en cada mes. Así mismo, permite categorizarlos, ordenarlos por fecha entre varias funcionalidades. Es ideal para organizar la economía de una persona, también el de una familia ya que permite llevar un análisis resumido de lo invertido en distintas cuestiones, como por ejemplo la cuota del mes del gimnasio, la cuenta del supermercado, el gasto del agua, entre otras. A su vez, es intuitivo y fácil de utilizar para que cada usuario pueda modificarlo y adaptarlo a sus necesidades sin problemas.

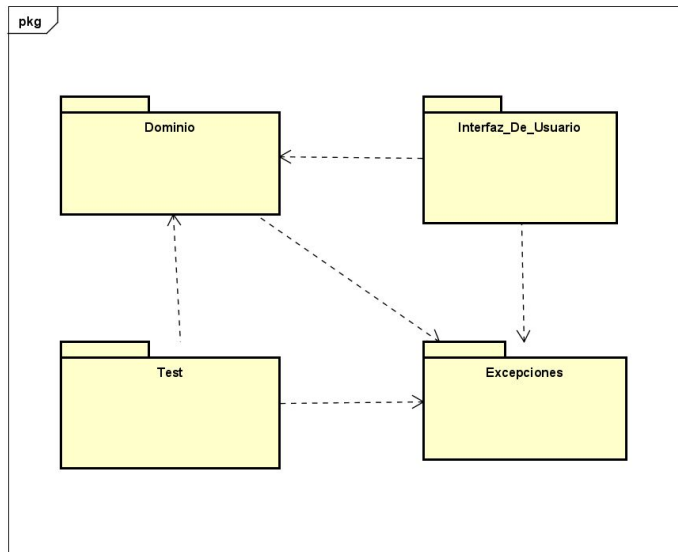
### 1.1 Funcionalidades no implementadas

### 1.2 Errores conocidos:

- En un primer comienzo no tuvimos un buen manejo de git, por lo tanto trabajamos en una rama sobre más de una feature. Por ejemplo, en la rama Gasto creamos y desarrollamos las clases Gasto y Categoría. A pesar de esto, en la práctica aprendimos a manejarlo correctamente en el transcurso y consideramos que los errores son parte del aprendizaje, lo que no impidió lograr el desarrollar un buen sistema.
- Realizamos merge a la rama *master* por un error de tipeo. Reconocemos que hacer merge a master cuando el proyecto no está terminado es una mala práctica. Ya que en master debe quedar la versión final (la que está en funcionamiento) y se debe trabajar en la rama *develop* que contiene el historial del proyecto con ramas features.
- Hay commits en develop por error de manejos en git, como explicamos en el primer punto, al no tener suficiente experiencia práctica en git, si bien sí teórica, al principio nos equivocamos en los pasos a seguir al hacer merge a develop. Posteriormente aprendimos que desde la rama feature se debe realizar merge con la rama develop y para resolver los conflictos (si hubiera), una vez listo el documento se realiza un pull request y se une la rama a develop.
- Al iniciar el programa desde Visual Studio se debe establecer el paquete Interfaz De Usuario como proyecto de inicio ya que debido a la configuración de Git Ignore no hemos logrado dejar esa configuración establecida.

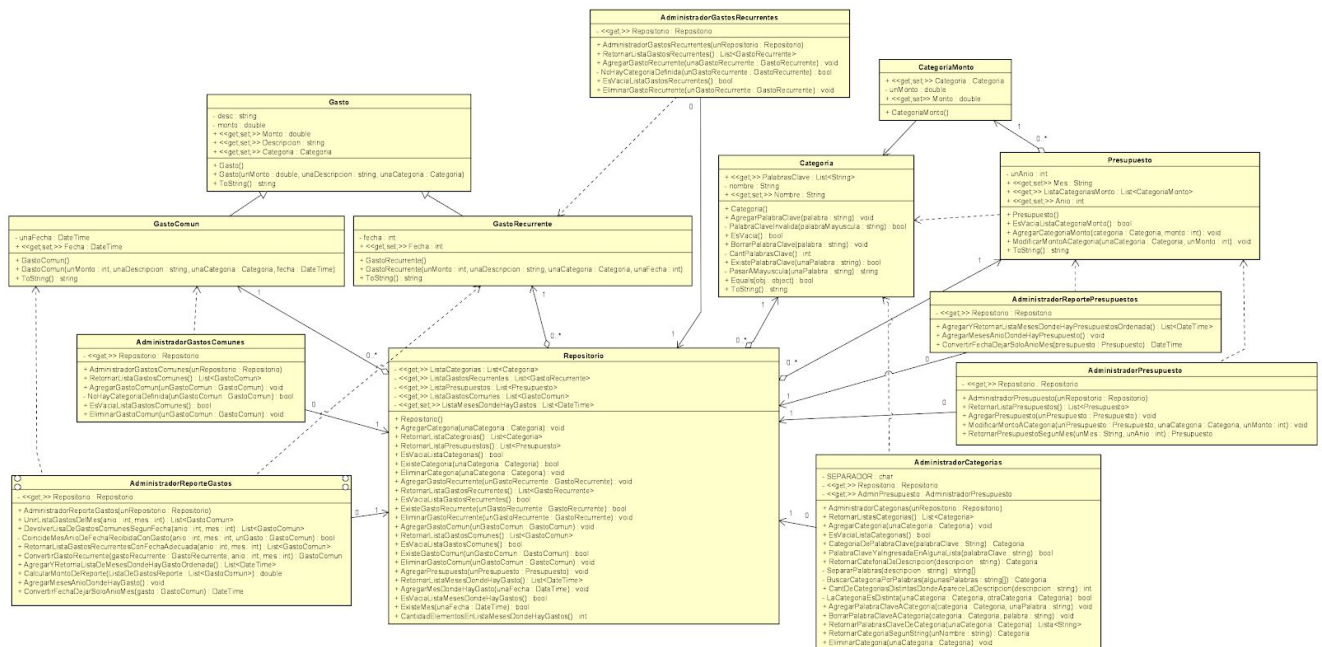
## 2. Descripción y justificación de diseño:

### a. Diagrama de paquetes



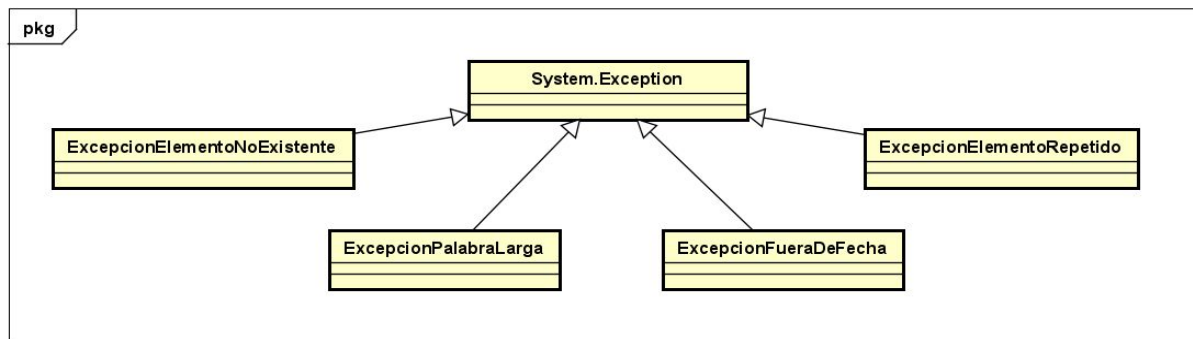
### b. Diagramas de clases

- Diagrama de Clases del Dominio.

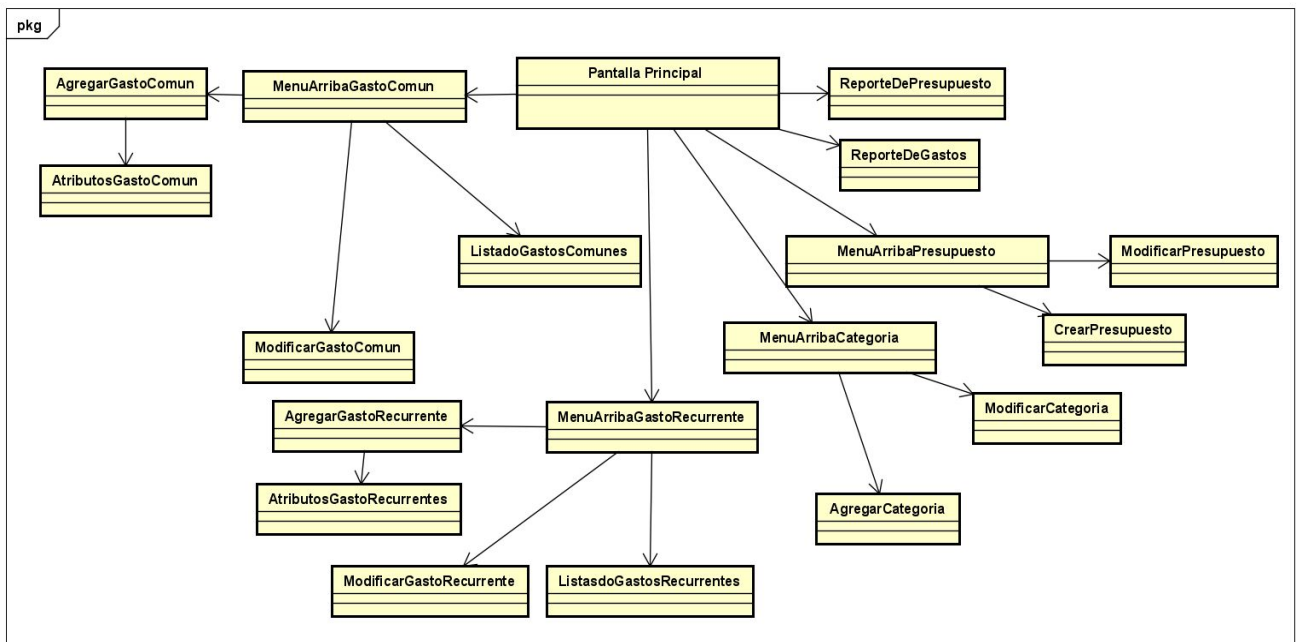


### Anexo I - imagen ampliada del Diagrama de Clases del Dominio.

- Excepciones



- Interfaz de Usuario



### **c. Explicación de los mecanismos generales y descripción de las principales decisiones de diseño tomadas.**

Para diseñar el sistema nos basamos en los principios fundamentales de la programación orientada a objetos.

#### **Abstracción:**

Procuramos que los objetos tengan las características relevantes acorde al objetivo de nuestro sistema, evitando características innecesarias con el fin de ser fáciles de usar, modificar, construir y comprender.

#### **Encapsulamiento e Information Hiding:**

A su vez, nos basamos en el principio de encapsulamiento para garantizar al máximo la integridad de los datos. Por ejemplo, las listas de la clase Repositorio son privadas y accedemos a ellas mediante métodos. A través del proceso de aprendizaje nos dimos cuenta que en las clases Categoría y CategoriaMonto, las listas también deberían ser privadas con métodos que permitan acceder a ellas. En las clases de Administradores existen varios métodos públicos que también deberían ser privados, ya que la interfaz solo utiliza solo algunos de ellos. Los demás métodos que no son utilizados exteriormente deberían ser métodos internos o privados. Hemos aprendido que se deben hacer privadas las variables o métodos innecesarios para tratar el objeto exteriormente, pero necesarias para su funcionamiento.

#### **Modularización**

Según lo visto en clase, es una buena práctica crear un diseño descompuesto en un conjunto de módulos poco acoplados, minimizando al máximo sus dependencias entre ellos, con el fin de reducir el impacto debido a cambios externos. Para esto creamos clases administradoras relativas a cada entidad del dominio, para dividir las responsabilidades y manejar dichos módulos por separado.

#### **Herencia**

Sirve para expresar los elementos comunes entre las clases y a su vez definir métodos nuevos en la subclase. Si las clases estuvieran separadas repetiríamos parte del código, lo que se evita con herencia. Por eso decidimos utilizar la herencia entre GastoRecurrente y GastoComun de la clase Gasto ya que Gasto tiene todos los atributos comunes mientras que GastoComun tiene una fecha de 2018 a 2030 y GastoRecurrente sólo almacena un día del 1 al 28. Como el atributo fecha tiene distintos requerimientos según el tipo de gasto, decidimos separarlo en dos subclases con el atributo fecha adecuado para cada una.

## **Interacción Interfaz-Dominio:**

Interfaz → Administradores → Repositorio

La interfaz se comunica con los métodos de los distintos administradores. Estos se encargan de crear, modificar y almacenar los objetos en Repositorio. Los administradores son el punto de acceso al Dominio.

### **Manejo de errores y excepciones**

Para capturar las excepciones utilizamos una biblioteca de excepciones de .NET.

A su vez, identificamos los tipos de excepciones más recurrentes en nuestro sistema como por ejemplo cuando el elemento no existe y creamos nuestras propias excepciones para estos casos:

- ExcepcionElementoNoExistente
- ExepcionElementoRepetido
- ExcepcionFueraDeFecha
- ExcepcionPalabraLarga

A través de las excepciones capturamos los errores, estos ayudan a interactuar con el usuario, y que pueda recibir el feedback adecuado.

## **d. Breve análisis de los criterios seguidos para asignar las responsabilidades.**

El dominio se puede dividir en tres secciones de clases, cada una con una responsabilidad bien definida:

En primer lugar, en las clases encargadas de modelar las entidades del sistema: Gasto (con sus herencias Gasto Común y Gasto Recurrente), Presupuesto y Categoría.

A su vez, un conjunto de clases Administrador, que tienen los métodos asociados a las funcionalidades pedidas y son el punto de entrada al dominio ya que la interfaz se comunica con ellas. Estos métodos están encapsulados en diferentes clases con el fin de modularizar el sistema.

Por último, la clase Repositorio. Este tiene una única instancia, y es donde se definen y almacenan las estructuras de objetos necesarias para operar (lista de Gastos Comunes, lista de Gastos Recurrentes, lista de Presupuestos, lista Categorías).

### 3. Cobertura de pruebas unitarias con su debido análisis y justificaciones:

#### a.

Para realizar las pruebas unitarias Microsoft.VisualStudio.TestTools.UnitTesting. Aprender a programar con TDD nos ha enseñado a pensar el funcionamiento antes de codificar. Si bien tuvimos que invertir tiempo practicando debido a la curva de aprendizaje, logramos avanzar rápidamente y mantener el código en funcionamiento. A medida que fue creciendo el proyecto pudimos agregar sin perjudicar lo que ya estaba en funcionamiento. Gran parte de esto fue gracias al cumplimiento de las reglas básicas de TDD y hacer refactor adecuadamente.

Cada clase del dominio fue generada a partir de un Test. Nos ayudó a poder evaluar si el funcionamiento de cada uno de los métodos de la clase se comporta como se espera. Se usaron dos métodos de testeo:

[TestClass] para indicar que es una clase de tipo Test.

[TestInitialize] Indica que lo anotado se debe ejecutar antes que todos los métodos

[TestMethod]. A su vez [ExpectedException] para probar las excepciones.

**Tenemos una cobertura de código de 96,78%**

Resultados de la cobertura de código				
Sara Vila_DESKTOP-I6PFV7F 2020-10-29 22				
Jerarquía	No cubiertos (bloques)	No cubiertos (% de bloques)	Cubiertos (bloques)	Cubiertos (% de bloques)
▲ Sara Vila_DESKTOP-I6PFV7F 2...	64	4.19%	1464	95.81%
▶ excepciones.dll	18	75.00%	6	25.00%
▶ obligatorio1.dll	24	3.22%	721	96.78%
▶ {} Dominio	24	3.22%	721	96.78%
▶ AdministradorCate...	14	9.66%	131	90.34%
▶ AdministradorGast...	0	0.00%	26	100.00%
▶ AdministradorGast...	2	7.69%	24	92.31%
▶ AdministradorPres...	5	9.09%	50	90.91%
▶ AdministradorRena...	0	0.00%	138	100.00%
Lista de errores Explorador de pruebas Salida Resultados de la cobertura de código				

Utilizamos GitFlow para poder tener un control adecuado de versiones de nuestro proyecto. En git flow se debería haber un único commit final, una rama develop sin commits y varias ramas feature en las cuales se trabaja cada funcionalidad y luego se lleva a develop lo deseado. En un comienzo tuvimos inconvenientes en aprender a utilizar esta metodología, como explicamos anteriormente.<sup>1</sup>

<sup>1</sup> Explicación en sección 1.2.



b. Para la funcionalidad de alta, baja, modificación y listado de Gastos.

Id	Prueba	Dato a Probar	Resultado Esperado	Resultado Obtenido
1	Ingresar una descripción	<b>Descripción</b> válida: Hogar	Se muestra mensaje: "No hay una categoría para sugerirle" y se muestran los demás campos para llenar.	Ok
2	Ingresar una descripción	(Se ingresa previamente una categoría: Auto con palabra clave Patente) <b>Descripción:</b> Patente	Se muestran los demás atributos del gasto y a su vez se selecciona la categoría Auto	Ok
3	Ingresar una descripción	(Se ingresa previamente una categoría: <b>Entretenimiento</b> con palabras claves: <b>Salida, Cine</b> ). <b>Descripción:</b> Salida al cine	Se despliegan los demás atributos de gasto y se sugiere la categoría entretenimiento.	Ok
4	Ingresar una descripción	(Se ingresa previamente una categoría: <b>Entretenimiento</b> con palabra clave: <b>Cine</b> y otra Categoría: Gastronomía con palabra clave: <b>Salida</b> ). <b>Descripción:</b> Salida al cine	El sistema encuentra 2 categorías a la cual pertenece la descripción entonces se muestra mensaje: "No hay una categoría para sugerirle". Y el campo categoría queda vacío.	Ok
5	Ingresar una descripción	<b>Descripción</b> invalida: A	Se muestra mensaje: "Descripción demasiado corta o larga".	Ok
6	Ingresar una descripción	<b>Descripción</b> invalida: Cine, comer papas fritas a un restaurante más helado	Se muestra mensaje: "Descripción demasiado corta o larga".	Ok
7	Ingresar una	<b>Descripción vacía</b>	Se muestra	Ok

	descripción		mensaje:"Descripción demasiado corta o larga".	
8	Ingresar un monto	<b>Monto:</b> 300	Automáticamente se modifica el número con 2 decimales. 300.00	Ok
9	Ingresar un monto	<b>Monto negativo:</b> -300	La interfaz no lo permite y modifica a 0.00	Ok
10	Ingresar un monto	(Se llenan los demás campos correctamente) <b>No se modifica el campo Monto.</b>	Permanece el monto 0.00 y el sistema permite agregarlo con ese monto.	Ok
11	Ingreso de una fecha	Ingresar una <b>fecha anterior al 01-01-2018</b>	El sistema muestra el calendario hasta el 2018	Ok
12	Ingreso de una fecha	Ingresar una <b>fecha posterior al 31-12-2030</b>	El sistema muestra el calendario hasta el 2030	Ok
13	Categoría	Se <b>despliega el combo box</b> Categorías	Se ve la lista de categorías ingresadas previamente	Ok
14	Categorías	Se <b>selecciona una categoría</b> de la lista	La misma permanece seleccionada	Ok
15	Categorías	(Ingresar campos correctamente llenados) No se selecciona ninguna categoría. Se presiona botón Agregar	Se muestra mensaje:"La categoría no puede quedar vacía". No se agrega el gasto.	Ok
16	Categorías	(Agregar categoría Gastronomía y Repetir pasos prueba 3) <b>Se modifica opción seleccionada automáticamente por Categoría Comida</b>	Se permite cambiar la categoría seleccionada automáticamente por el sistema.	Ok
17	Agregar Gasto	Descripción: Cine Monto: 400 Fecha: 28 Octubre	Se muestra el mensaje "El gasto ha sido creado con éxito"	Ok

		2020 Categoría: Entretenimiento		
18	Botón agregar descripción	Descripción: Cine Monto: 400 Fecha: 28 Octubre 2020 Categoría: Entretenimiento Se presiona botón: Agregar descripción	Se reinicia la operación	Ok
19	Modificar Gasto	Seleccionar gasto a modificar	Se despliega la lista de gastos comunes ingresado previamente	Ok
20	Eliminar	Seleccionar gasto a eliminar y presiona botón eliminar	Se elimina el gasto seleccionado	Ok
21	Modificar	Seleccionar gasto a modificar. Descripción: Entretenimiento Monto: 400 Fecha: 28 Octubre 2020 Categoría: Entretenimiento	Se modifican los atributos del gasto y se muestra el mensaje: "El gasto se ha modificado con éxito"	Ok
22	Listado de gastos	Se presiona el botón Listado de Gastos	Se muestra el listado de gastos ingresados	Ok

#### **Capturas de pantalla ANEXO 3b.**

#### **ANEXO 3b.**

1.

mis finanzas personales

Categoría

Gasto Comun

Gasto Recurrente

Presupuesto

Reporte de Gastos

Reporte de Presupuesto

Agregar Gasto Comun

Modificar Gasto

Ver Listado de Gastos

### Agregar Gasto Comun

Descripcion:

No se ha encontrado una categoria para sugerirle, seleccione una.

2.

mis finanzas personales

Categoría

Gasto Comun

Gasto Recurrente

Presupuesto

Reporte de Gastos

Reporte de Presupuesto

Agregar Gasto Comun

Modificar Gasto

Ver Listado de Gastos

### Agregar Gasto Comun

Descripcion:

Monto:

Fecha:

Categoria:

El gasto ha sido creado con exito

3.

mis finanzas personales

Categoría

Gasto Comun

Gasto Recurrente

Presupuesto

Reporte de Gastos

Reporte de Presupuesto

Agregar Gasto Comun

Modificar Gasto

Ver Listado de Gastos

### Agregar Gasto Comun


Descripcion:

Monto:

Fecha:

Categoria:

4.



Categoría

Gasto Comun

Gasto Recurrente

Presupuesto

Reporte de Gastos

Reporte de Presupuesto

Agregar Gasto Comun

Modificar Gasto

Ver Listado de Gastos


### Agregar Gasto Comun

Descripción
 
Agregar

No se ha encontrado una categoría para sugerirle, seleccione una.

Aceptar

5.



Categoría

Gasto Comun

Gasto Recurrente

Presupuesto

Reporte de Gastos

Reporte de Presupuesto

Agregar Gasto Comun

Modificar Gasto

Ver Listado de Gastos


### Agregar Gasto Comun

Descripción
 
Agregar

Descripción demasiado corta o larga

Aceptar

6.



Categoría

Gasto Comun

Gasto Recurrente

Presupuesto

Reporte de Gastos

Reporte de Presupuesto

Agregar Gasto Comun

Modificar Gasto

Ver Listado de Gastos


### Agregar Gasto Comun

Descripción
 
Agregar

Descripción demasiado corta o larga

Aceptar

7.



Categoría

Gasto Comun

Gasto Recurrente

Presupuesto

Reporte de Gastos

Reporte de Presupuesto

Agregar Gasto Comun

Modificar Gasto

Ver Listado de Gastos

### Agregar Gasto Comun

Descripción  Agregar

×

Descripción demasiado corta o larga

Aceptar

8.



Categoría

Gasto Comun

Gasto Recurrente

Presupuesto

Reporte de Gastos

Reporte de Presupuesto

Agregar Gasto Comun

Modificar Gasto

Ver Listado de Gastos

### Agregar Gasto Comun

Descripción  Agregar

Monto

Fecha

Categoría

Agregar

9.



Categoría

Gasto Comun

Gasto Recurrente

Presupuesto

Reporte de Gastos

Reporte de Presupuesto

Agregar Gasto Comun

Modificar Gasto

Ver Listado de Gastos

### Agregar Gasto Comun

Descripción  Agregar

Monto

Fecha

Categoría

Agregar

10.



Categoría

Gasto Comun

Gasto Recurrente

Presupuesto

Reporte de Gastos

Reporte de Presupuesto

Agregar Gasto Comun

Modificar Gasto

Ver Listado de Gastos

### Agregar Gasto Comun

Descripción

Agregar


Monto

Fecha

Categoría

Agregar

11.



Categoría

Gasto Comun

Gasto Recurrente

Presupuesto

Reporte de Gastos

Reporte de Presupuesto

Agregar Gasto Comun

Modificar Gasto

Ver Listado de Gastos

### Agregar Gasto Comun

Descripción

Agregar

Monto

Fecha

Categoría

12.



Categoría

Gasto Comun

Gasto Recurrente

Presupuesto

Reporte de Gastos

Reporte de Presupuesto

Agregar Gasto Comun

Modificar Gasto

Ver Listado de Gastos

### Agregar Gasto Comun

Descripción

Agregar

Monto

Fecha

Categoría

13.

Mis Finanzas Personales

mis finanzas personales

Agregar Gasto Comun Modificar Gasto Ver Listado de Gastos

Categoria

Gasto Comun

Gasto Recurrente

Presupuesto

Reporte de Gastos

Reporte de Presupuesto

**Agregar Gasto Comun**

Descripcion Cine Agregar

Monto 300.00

Fecha jueves, octubre 29, 2020

Categoria Entreteneimiento Gastronomía Agregar

14.

Mis Finanzas Personales

mis finanzas personales

Agregar Gasto Comun Modificar Gasto Ver Listado de Gastos

Categoria

Gasto Comun

Gasto Recurrente

Presupuesto

Reporte de Gastos

Reporte de Presupuesto

**Agregar Gasto Comun**

Descripcion Cine Agregar

Monto 300.00

Fecha jueves, octubre 29, 2020

Categoria Entreteneimiento Agregar

15.

Mis Finanzas Personales

mis finanzas personales

Agregar Gasto Comun Modificar Gasto Ver Listado de Gastos

Categoria

Gasto Comun

Gasto Recurrente

Presupuesto

Reporte de Gastos

Reporte de Presupuesto

**Agregar Gasto Comun**

Descripcion Cine Agregar

Monto 300.00

Fecha jueves, octubre 29, 2020

Categoria

La Categoria no puede quedar vacia

Aceptar

Agregar



16.

Mis Finanzas Personales

17.

Mis Finanzas Personales

18.

Mis Finanzas Personales

19.

Mis Finanzas Personales

20.

Mis Finanzas Personales

21.

Mis Finanzas Personales

22.

Mis Finanzas Personales

**mis finanzas personales**

Categoria

Gasto Comun

Gasto Recurrente

Presupuesto

Reporte de Gastos

Reporte de Presupuesto

Agregar Gasto Comun

Modificar Gasto

Ver Listado de Gastos

### Listado de gastos

Monto: 200	Descripcion: Nueva Descripcion	Categoria: Entret.	Fecha: 10/29/2020
Monto: 2000	Descripcion: Super	Categoria: Hogar	Fecha: 10/29/2020

23.

Mis Finanzas Personales

**mis finanzas personales**

Categoria

Gasto Comun

Gasto Recurrente

Presupuesto

Reporte de Gastos

Reporte de Presupuesto

Agregar Gasto Comun

Modificar Gasto

Ver Listado de Gastos

Seleccione un gasto a eliminar

Aceptar

Eliminar

Modificar Gasto Seleccionado:

Descripcion

Monto: 0.00

Fecha: jueves, octubre 29, 2020

Categoria

Modificar

Anexo I - imagen ampliada del Diagrama de Clases del Dominio.

