BankerBot

# Tareas:

1. Registrar gastos.
2. Clasificar los gastos.
3. Obtener resúmenes de gastos por rubros.
4. Establecer objetivos de ahorro.
5. Mandar alertas sobre el dinero (utilización).
6. Administrar las cuentas bancarias. (saber y modificar balance)

# Clases:

1. Egreso: compuesto por:
   1. Moneda: Clase abstracta para permitir agregar más, en un principio va a ser $ y U$S
   2. Origen: Lugar del que sale el egreso, ej: cuenta bancaria, billetera, etc.
   3. Tipo de cambio: tiene que ser actualizado al momento de la operación.
   4. Monto: total del egreso.
   5. Concepto: Clasificación del gasto.
   6. Fecha: fecha en la que se realizó
2. Ingreso: compuesto por:
   1. Moneda: Clase abstracta para permitir agregar más, en un principio va a ser $ y U$S
   2. Destino: Lugar del que sale el egreso, ej: cuenta bancaria, billetera, etc.
   3. Tipo de cambio: tiene que ser actualizado al momento de la operación.
   4. Monto: total del ingreso.
   5. Concepto: Clasificación del ingreso.
   6. Fecha: fecha en la que se realizó
3. Entrada: compuesto por:
   1. Fuente de información: Planilla Excel, HTML o el formato de entrada que sea.
4. Alertas: compuesto por:
   1. Alerta por mucho gasto. (% de ingresos x medida de tiempo gastado) (ej: 20% de los ingresos x semana)
   2. Alerta por poco dinero disponible.
   3. Alerta por objetivos de ahorro (total alcanzado).
   4. Alerta de ahorro: Monto a ahorrar por plazo estipulado.
5. Moneda (Interfaz o Superclase): compuesto por:
   1. Constructor.
   2. Saber su cotización al día.
6. Conversión compuesta por:
   1. Método de conversión de moneda a moneda.
7. Cuenta bancaria: compuesto por:
   1. Moneda: tipo de moneda
   2. Banco: Institución
   3. Saldo: monto disponible para operar
8. Billetera: compuesto por:
   1. Moneda: que tipos tengo.
   2. Saldo por moneda: total por moneda
   3. Saldo total: “tenés {saldoxmoneda} + “y” + {saldoxmoneda} disponibles.”
9. Tarjeta de crédito: compuesto por:
   1. Moneda: Tipo de moneda
   2. Límite: valor fijado
   3. Saldo: Cantidad disponible para usar
   4. Gasto: total de movimientos en el mes
10. Chanchita de ahorro: compuesto por:
    1. Moneda: que tipos tengo.
    2. Origen: Múltiples orígenes.
    3. Objetivo de ahorro: Conocer el valor del objetivo.
    4. Total de ahorro:
11. Transferencia interna: compuesto por:
    1. Origen: Conocer todo sobre el origen (cuál es y qué tipo de moneda acepta)
    2. Destino: Conocer todo sobre el destino (cuál es y qué tipo de moneda acepta)
    3. Importe: saber cuánto se está transfiriendo.
    4. Moneda: saber que moneda.
    5. Concepto: motivo de la transferencia.
    6. Fecha: fecha en la que se realizó
12. Lista de egresos: compuesto por:
    1. Tipos de rubros en los que pueda gastar.
    2. Listar tipos de rubros existentes.
    3. Brindar información del gasto en un rubro.
    4. Brindar información de los gastos en todos los rubros.
13. Rubro: compuesto por:
    1. Tipo del rubro.
    2. Líneas de gastos
    3. Calcular el total de gastos por el rubro.

# Trabajo para hacer por el equipo

Tarjetas CRC:  
Formato para hacerlas ahora: <https://creately.com/diagram/example/hmmj51oa2/CRC%20cards>

Diagrams para tarjetas crc:

<https://drive.google.com/file/d/1XxotCEOreO0XN3uR30Z_3pbthmp7faWK/view?usp=sharing>

Sonia: Egreso, Entrada, cuenta bancaria, chanchita de ahorro, rubro

Mato: Ingreso, Moneda, billetera, transferencia interna

Andrés Alertas, conversión, tarjeta de crédito, lista de egresos

# Bibliografía consultada:

<http://www.cs.utsa.edu/~cs3443/uml/uml.html>

<https://medium.com/the-innovation/uml-class-diagram-arrows-guide-37e4b1bb11e>

<https://i.stack.imgur.com/O6Itp.gif>

<https://loufranco.com/wp-content/uploads/2012/11/cheatsheet.pdf>

<https://holub.com/uml/>

# Desafíos encontrados:

1. Terminar de entender las flechas de relaciones en UML.

# Hito 3:

1. Verificar los análisis, chequear que estén convirtiendo las monedas cuando hacen el cálculo total (poner un if para chequear si están todas en dólares).
2. Levantar el telegramBot y empezar a implementar el sistema.
3. CurrencyExchange es por un bloc de notas con los valores de la moneda, hay que dejar las abstracciones prontas como para que se cambie el source de donde se lee esta información.
4. Exportar los datos a HTML.
5. Armar las planillas de HTML acorde a la información que queremos que arme.
   1. Saldos de los medios de pago.
   2. Gastos por rubro.
   3. Alertas configuradas.
6. Mensajes de usuario por telegram hacia la UserInterface (que, en realidad, tienen como destino la consola, desde la consola los lee UserInterface.)
7. Hacer el program.
8. Hacer la documentación de DocFX.
9. Actualizar el Readme.
10. Videos demostrativos de la utilización.

# División de tareas:

1. Mensajes de telegram con el programa (integración de la TelegramAPI de nuestro código).
2. HTML APi
3. DocFX y Readme.
4. Demo en vivo.

Patrón visitor es lo que utiliza la library

|  |  |
| --- | --- |
| Resumen de cuentas (y ahorro) al día {DateTime.Now} | |
| Método de pago | **Saldo** |
| BankAccount | 5000 |
| CreditCard | 15000 |
| SubWallet 1 | 200 |
| SubWallet 2 | 3000 |
| Total |  |

|  |  |
| --- | --- |
| Resumen de gastos por rubro (expensetype) al día {Datetime.Now} | |
| Gasto por rubro | **Total por rubro** |
| Supermercado | 5000 |
| Alquiler | 15000 |
| Transporte | 200 |
| Cuentas | 3000 |
| Total |  |

|  |  |  |
| --- | --- | --- |
| Resumen de alertas fijadas por el usuario al día {DateTime.Now} | | |
| Tipo de Alerta | **Nivel fijado** | **Nivel actual** |
| Objetivo de ahorro | 30.000 | 10.000 |
| Fondos mínimos | 5.000 | 10.000 |
| Gasto en Supermercado | 8.000 | 5.000 |
| Gasto en Transporte | 1.000 | 200 |
|  |  |  |