

**INSTITUTO POLITÉCNICO DE VIANA DO CASTELO**  
**ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO**  
**LICENCIATURA EM ENGENHARIA INFORMÁTICA**

JACKSON JÚNIOR

**PROGRAMAÇÃO II**  
**Desenvolvimento de um Gestor de Ativos Financeiros**

VIANA DO CASTELO  
JANEIRO 2021

## Sumário

Resumo.....	3
Abstract .....	3
Introdução .....	4
Decisões de Projeto .....	4
Escolha de tipo de dados.....	4
Teste unitário .....	4
Arquitetura .....	4
Encapsulamento .....	5
Ordenação .....	5
Criptografia.....	5
Experiência do usuário .....	5
Engenharia de Requisitos .....	6
Documentação dos Requisitos .....	6
Requisitos Funcionais .....	6
Requisitos Não-Funcionais .....	7
Modelagem .....	10
Modelo de Domínio.....	10
Glossário .....	10
Diagrama de Domínio.....	11
Diagrama Entidade-Relacionamento .....	12
Diagrama de Classes.....	13
Bibliografia.....	14
Anexos .....	16
Arquitetura da informação das telas.....	16
Modelagem digital de telas - Usuário .....	18
Modelagem digital de telas - Admin .....	23

## Resumo

O presente trabalho apresenta uma aplicação de gestão e ativos financeiros implementada sob o paradigma de orientação a objetos e com uso de interface gráfica, utilizando-se, ainda, de técnicas de engenharia de requisitos, modelagem de processos e modelagem de dados orientados a objetos. Fazendo uso das linguagens UML, Java.

Contando com a orientação científica do Prof. Dr. Ricardo Castro, Prof. Dr. António Miguel Ribeiro dos Santos Rosado da Cruz, e do Prof. Dr. <sup>o</sup> Luís Filipe Guimarães Teófilo.

Palavras-Chave: BPMN, UML.

## Abstract

The present work presents a management and financial assets application implemented under the object-oriented paradigm and using a graphical interface, also using requirements engineering techniques, process modeling and object-oriented data modeling. Making use of UML, Java languages.

Having the scientific guidance of Prof. Dr. Ricardo Castro, Prof. Dr. António Miguel Ribeiro dos Santos Rosado da Cruz, and Prof. Dr. <sup>o</sup> Luís Filipe Guimarães Teófilo.

Keywords: BPMN, UML.

## Introdução

Este projeto consiste no desenvolvimento de um sistema de rastreio dos ativos financeiros pessoais, tais como depósitos a prazo, imóveis arrendados, juros de produtos financeiros de risco (ações, fundos, etc.) através do qual o utilizador pode acompanhar a sua evolução patrimonial, recolhimento de impostos entre outras funcionalidades. Conta ainda com suporte a diferentes níveis de permissão.

## Decisões de Projeto

### Escolha de tipo de dados

Os tipos primitivos de ponto flutuantes não são recomendados para representar números decimais, ou seja, números na base 10. Esta não é uma particularidade do Java, mas sim de qualquer linguagem que use o ponto flutuante sobre base numérica 2. Tendo em conta que a representação monetária limita-se a múltiplos de  $10^{-2}$ , os únicos números que podem ser representados da forma correta como ponto flutuante binário[1] são os do conjunto {0; 0,25; 0,5; 0,75; 1}, todos os outros sofrem variação em um pequeno valor, que é mais do que suficiente para causar muitos problemas. Para contornar esta situação, no Java, optou-se por utilizar a classe `BigDecimal`[2] que comporta todas as necessidades dos cálculos monetários.

### Teste unitário

Utilizou-se o `JUnit`[3] para implementar a metodologia de desenvolvimento orientada a testes, segundo a escola de Detroit (*Chicago Scholl*)[4], que proporcionou que todos os requisitos funcionais estivessem totalmente implementados nos primeiros momentos de desenvolvimento.

### Arquitetura

A arquitetura escolhida foi o MVC[5], que considerou-se proporcionar uma melhor manutenibilidade do código. O trabalho nas classes de visualização também fez uso de outros *designer patterns*, como o *Observer*[6] nas *tableViews* e na estrutura por detrás da manipulação de eventos dos *inputs*. Outro padrão utilizado foi o *Singleton*[7] para classes de login, sessão e logs, com o pensamento de manter toda a operação *thread-safe*, ainda que a aplicação tenha sido desenvolvida em *single-thread* e não se tenha utilizado a implementação do *singleton* em enumerador, que é mais segura por evitar o uso de *reflection* e serialização.

## Encapsulamento

Objetivando garantir um dos pilares da orientação a objetos implementou-se o método `clone` em todas as classes, bem como seus “companheiros” `equals` e `hashCode`. Desta forma teve-se a garantia de encapsulamento, já que o mesmo foi utilizado em seus métodos `gets` e `setters`.

## Ordenação

Implementou-se a interface `Comparable`[8], com a sobrescrita dos métodos `compareTo` para fornecer a possibilidade de ordenação, através dos métodos nativos de ordenação de coleções do Java.

## Criptografia

Utilizou-se a técnica de `salt` para dar mais robustez a senha, evitando ataques de dicionário. O `salt` foi obtido com a classe `SecureRandom` que fornece números aleatórios criptograficamente fortes[9]. Já para obter o `hash` da senha com o `salt` contou-se com a classe `PBKESpec`[10] que permite configurar a quantidade de iterações e o tamanho da chave, dificultando ainda mais o ataque de força bruta.

## Experiência do usuário

Dedicou-se especial tempo e atenção a modelagem da interface gráfica do projeto, seja com a apresentação dos dados no formato habitual, como datas e valores monetários, como com a taxinomia e arquitetura de informação, partindo de traços a lápis no papel para a modelagem de telas no computador.

# Engenharia de Requisitos

## Documentação dos Requisitos

### Requisitos Funcionais

A prioridade dos requisitos está definida em uma escala de três níveis, nomeadamente:

1 – alta, 2 – média e 3 – baixa.

#	Descrição	Prioridade
1	A aplicação deve permitir a criação de uma conta de utilizador.	1
2	A aplicação deve condicionar o acesso ao sistema através da realização de login.	1
3	A aplicação deve permitir, ao utilizador, inserir, apagar e atualizar seus ativos financeiros.	1
4	A aplicação deve permitir, no caso dos fundos de investimento, a atribuição de uma taxa de juros específica a cada mês.	1
5	A aplicação deve permitir ao utilizador pesquisar por ativos financeiros através de nome e tipo.	1
6	A aplicação deve permitir ao utilizador, <b>no caso de fundos de investimento e depósitos</b> , filtrar a pesquisa por montante aplicado.	1
7	A aplicação deve permitir ao utilizador listar <b>todos</b> os <b>ativos financeiros</b> que estão ativos na data atual, ordenados decrescentemente pelo valor inicialmente aplicado neste ativo.	1
8	A aplicação deve gerar um relatório com todos os <b>ativos financeiros</b> que estavam ativos no intervalo de duas datas. O relatório deverá conter o <b>lucro total</b> (antes e depois de impostos) e o lucro mensal médio (antes e depois de impostos) para todos os ativos desse período.	1
9	A aplicação deve gerar um relatório que lista todos os pagamentos de impostos e respetivo ativo (ao nível mensal) para os ativos financeiros disponíveis em determinado intervalo de tempo.	1

#	Descrição	Prioridade
10	A aplicação deve gerar um relatório, a um administrador, com o valor total depositado em cada banco (depósito a prazo) e o montante que o banco pagou em juros aos utilizadores, no intervalo entre duas datas,	1

## Requisitos Não-Funcionais

### Requisitos de Produto

A prioridade dos requisitos está definida em uma escala de três níveis, nomeadamente:

1 – alta, 2 – média e 3 – baixa.

#	Descrição	Prioridade
1	A aplicação deve manter a interoperabilidade característica da linguagem Java, nomeadamente sendo suportada no Linux, Mac e Windows.	1
2	A aplicação deve contar com comentários no formato JavaDoc.	2
3	A aplicação deve estar coberta com testes unitários.	3
4	A aplicação deve utilizar o JUnit para realização dos testes unitários.	3
5	Quando a aplicação inicia pela primeira vez (ainda sem utilizadores), deverá automaticamente criar um único utilizador do tipo Admin com um nome de utilizador e password conhecidos (admin/admin).	1
6	Todos os dados da aplicação deverão ser persistidos em ficheiro(s).	1
7	A aplicação deverá ter uma interface (com menus) que funciona através da linha de comandos. A aplicação deverá receber como argumento o(s) ficheiro(s) com os dados anteriormente gravados ou o nome de um novo ficheiro.	1
8	Implementação de interface gráfica (substitui a implementação de uma interface baseada em consola).	2
9	A aplicação deve ser desenvolvida com a utilização de versionamento de código.	2

### *Requisitos de Segurança*

A prioridade dos requisitos está definida em uma escala de três níveis, nomeadamente:

1 – alta, 2 – média e 3 – baixa.

#	Descrição	Prioridade
1	A aplicação deve ter 3 níveis de autenticação: <b>User</b> (pode aceder aos seus registos ou a registos consigo partilhados); <b>UserManager</b> (pode criar novos utilizadores, editar permissões na aplicação e visualizar todos os registos); <b>Admin</b> (pode realizar qualquer ação no sistema; criar/editar utilizadores (de todos os tipos), alterar permissões e visualizar todos os registos disponíveis no sistema).	1
2	Os dados introduzidos na aplicação devem ser validados.	1
3	A aplicação deve manter um registro de logs, com o histórico de ações ou comandos realizados no sistema, indicando o nome da ação realizada, o utilizador que a realizou e data/hora.	3
4	A aplicação deve permitir a cada utilizador consultar o seu histórico, sendo que os Admins podem consultar todo o histórico de utilização da aplicação.	3
5	As passwords deverão ser encriptadas	3



### *Requisitos Organizacionais*

A prioridade dos requisitos está definida em uma escala de três níveis, nomeadamente:

1 – alta, 2 – média e 3 – baixa.

#	Descrição	Prioridade
1	No caso de depósito a prazo, os juros (anuais) são pagos mensalmente, ou seja, o valor do depósito aumenta a cada mês.	1
2	As taxas de juros podem ser negativas.	1
3	Os impostos são devidos a cada ano.	1
4	Os impostos só são devidos no caso de os ativos financeiros terem dado lucro.	1
5	Formatação dos números decimais: os números decimais apresentados nos relatórios deverão ser arredondados às centésimas com separadores de milhares (exemplo: 1.234,56).	1

## Modelagem

### Modelo de Domínio

A análise de domínio permite identificar situações problemas comuns em aplicações similares, mas não propriamente iguais, objetivando a economia, celeridade e qualidade do projeto.

Do resultado desta análise, depreende-se os padrões de reutilização, taxinomia de classes, linguagens do domínio e modelos funcionais.

O domínio deste projeto abarcar nomeadamente:

1. Gestão de ativos financeiros;
2. Cálculo de juros *pro rata*;
3. Sistema de log;
4. Sistema de login;

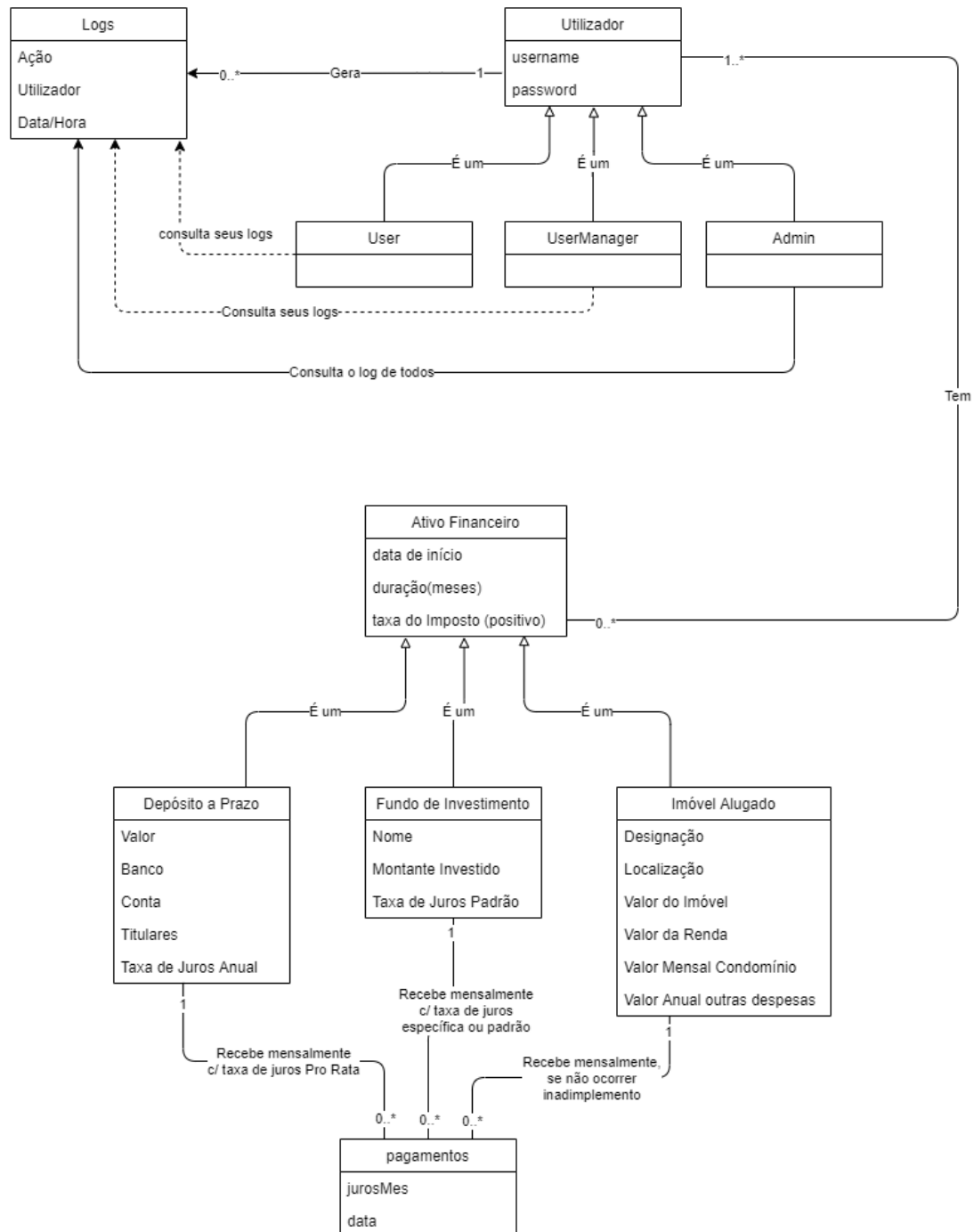
### Glossário

**Fundo de Investimento:** É um instrumento financeiro que resulta da captação de capital junto de diversos investidores, constituindo o conjunto desses montantes um património autónomo, gerido por especialistas que o aplicam numa variedade de ativos.[11]

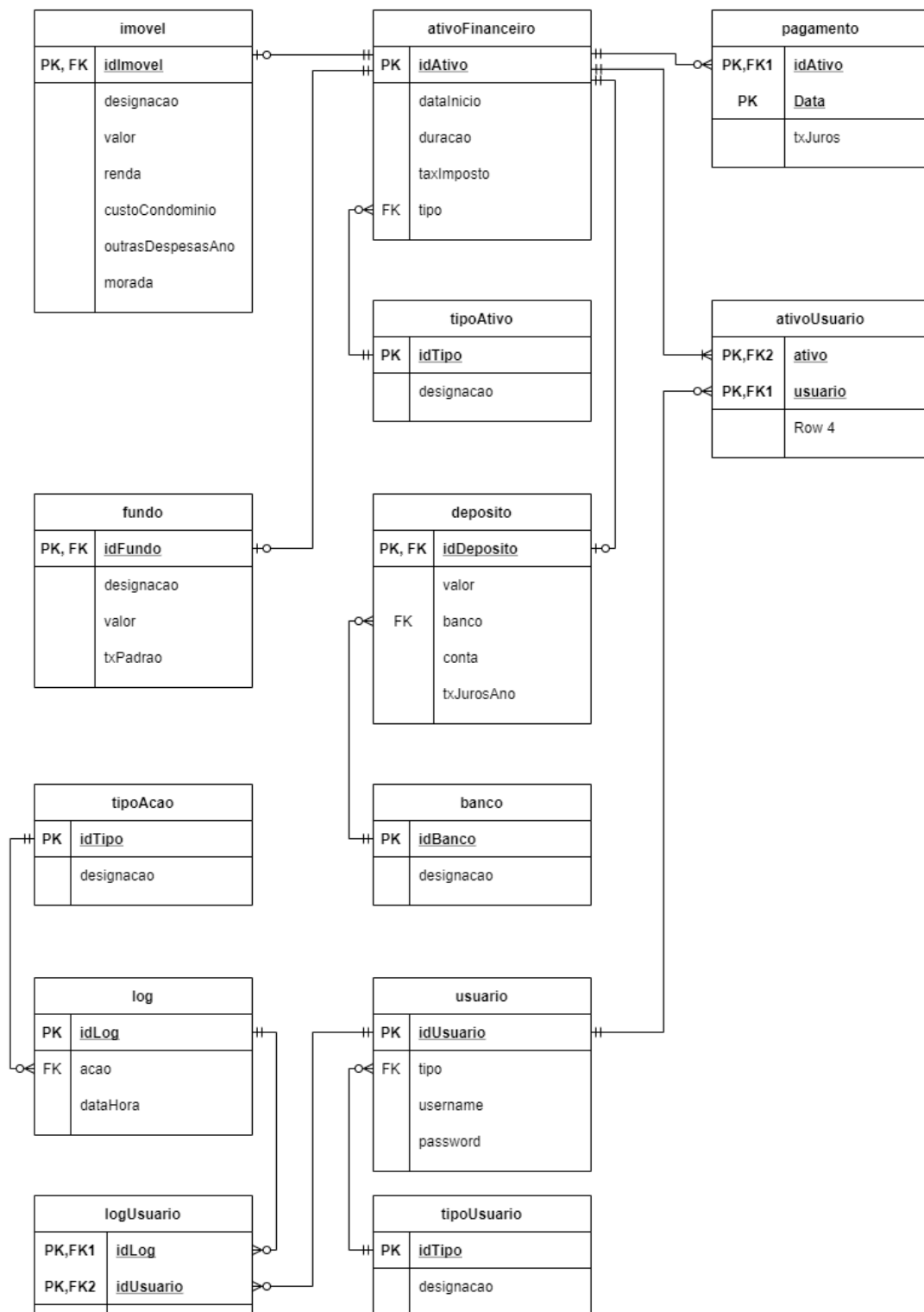
**Depósito a Prazo:** É um produto bancário que pressupõe a entrega de fundos a uma instituição de crédito, que fica obrigada a restituir esses fundos no final de um período acordado e ao pagamento de uma remuneração, designada de juro.[12]

**Pro Rata:** É uma expressão latina que tem um sentido de divisão, podendo ser traduzida livremente como algo que é "medido proporcionalmente". O cálculo de juros proporcionais serve para definir o pagamento de juros ao contratar um serviço, ou o recebimento de juros em um investimento. De maneira simples, essa medição pode ser feita com o valor por inteiro, dividido pela proporção que seja preciso. Se um investimento paga juros anuais, os juros *pro rata* podem ser calculados mensalmente ou semanalmente, por exemplo.[13]

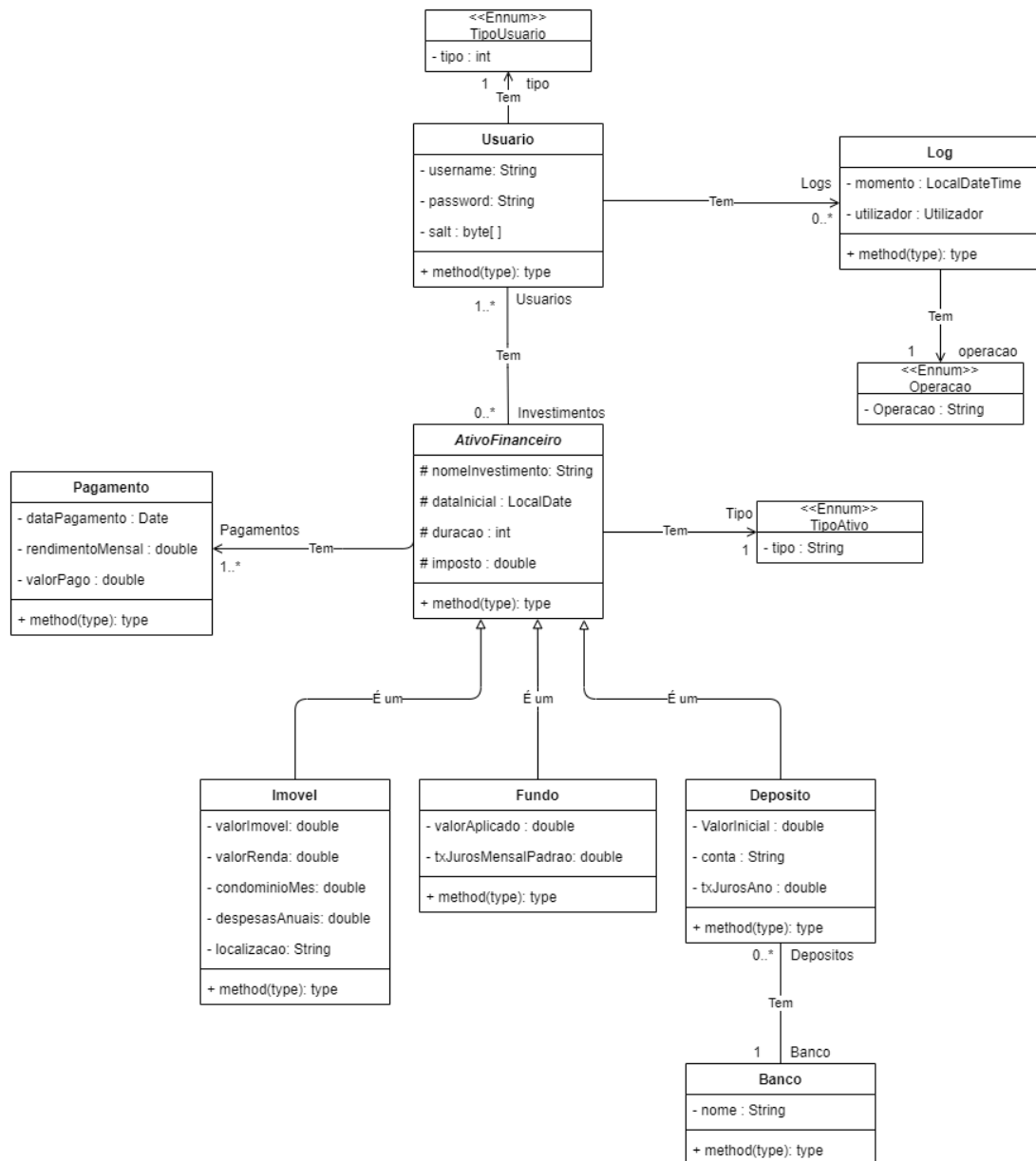
## Diagrama de Domínio



## Diagrama Entidade-Relacionamento



## Diagrama de Classes



## Bibliografia

- [1] "IEEE 754 - Wikipedia." [https://en.wikipedia.org/wiki/IEEE\\_754](https://en.wikipedia.org/wiki/IEEE_754) (accessed Jan. 31, 2021).
- [2] "BigDecimal (Java Platform SE 7)." <https://docs.oracle.com/javase/7/docs/api/java/math/BigDecimal.html> (accessed Jan. 30, 2021).
- [3] "Assertions (JUnit 5.0.1 API)." <https://junit.org/junit5/docs/5.0.1/api/org/junit/jupiter/api/Assertions.html> (accessed Jan. 31, 2021).
- [4] "London vs Chicago TDD | DevLead.io." <https://devlead.io/DevTips/LondonVsChicago> (accessed Jan. 31, 2021).
- [5] "Java SE Application Design With MVC." <https://www.oracle.com/technical-resources/articles/javase/mvc.html> (accessed Jan. 31, 2021).
- [6] "Observer design pattern in java - Java2Blog." <https://java2blog.com/observer-design-pattern-in-java/> (accessed Jan. 31, 2021).
- [7] "Java Singleton Design Pattern Example Best Practices - JournalDev." <https://www.journaldev.com/1377/java-singleton-design-pattern-best-practices-examples> (accessed Jan. 31, 2021).
- [8] "Object Ordering (The Java™ Tutorials > Collections > Interfaces)." <https://docs.oracle.com/javase/tutorial/collections/interfaces/order.html> (accessed Jan. 31, 2021).
- [9] "SecureRandom (Java Platform SE 8)." <https://docs.oracle.com/javase/8/docs/api/java/security/SecureRandom.html> (accessed Jan. 31, 2021).
- [10] "PBEKeySpec (Java Platform SE 7)." <https://docs.oracle.com/javase/7/docs/api/javax/crypto/spec/PBEKeySpec.html> (accessed Jan. 31, 2021).
- [11] "Fundos de investimento | Todos Contam." <https://www.todoscontam.pt/pt-pt/fundos-de-investimento> (accessed Dec. 06, 2020).
- [12] "Depósitos a prazo | Todos Contam." <https://www.todoscontam.pt/pt-pt/depositos-prazo> (accessed Dec. 06, 2020).

- [13] “O que é Pro Rata: significado e conceito - Dicionário Financeiro.”  
<https://www.dicionariofinanceiro.com/o-que-e-pro-rata/> (accessed Dec. 06, 2020).

## Arquitetura da informação das telas





Reed

- celebration of promise  
to be born for us  
to be born for us

→

✓ 107 10/10/2000

On

- route

in Bremen

Frage 2 von 5

33,000,00  
27,000,00

Worms  
Sandra  
Tina (Sandra)

Types: (Scrap)

202 ~~202~~ )

2

7

1. C. D. D. D.

→ 1000000

2100

25

2000

22

---

## Modelagem digital de telas - Usuário

**Asset ManagerY**

HOME

NOVO ATIVO

RELATÓRIO ^

IMPOSTO PAGO

FINANCEIRO




LISTAR TODOS ATIVOS

CONSULTAR LOGS

LOG OUT

Q PROCURAR

FILTROS v



### Cadastrar novo ativo

TIPO DE ATIVO

☐ Depósito a termo ☒ Imóvel arrendado ☐ Fundo de investimento

NOME DO ATIVO Prédio aluguel casa 2					
INICIO	20/12/2020	DURAÇÃO	12 meses v	VALOR DO IMÓVEL	€ 170.000,00
IMPOSTO	10%	CONDOMÍNIO	€ 170	RENTA MENSAL	€ 1.700,00
NOME DO ATIVO Endereço rua 123 legal				OUTRAS DESPESAS	€ 170

CADASTRAR

**Asset ManagerY**

HOME

NOVO ATIVO

RELATÓRIO ^

IMPOSTO PAGO

FINANCEIRO




LISTAR TODOS ATIVOS

CONSULTAR LOGS


LOG OUT

Q PROCURAR

FILTROS v



### Configuração

EDITAR CONTA 

NOME DE USUÁRIO Usuário123	
SENHA *****	CONFIRMAR SENHA *****

SALVAR

- HOME
- NOVO ATIVO
- RELATÓRIO ^
- IMPOSTO PAGO
- FINANCEIRO
- LISTAR TODOS ATIVOS
- CONSULTAR LOGS

LOG OUT

PROCURAR

FILTROS v

### Consultar logs

DATA	REGISTRO	
20/12/2020 14:55:32	LOG OUT	+
20/12/2020 13:55:32	EDIÇÃO ATIVO "CASA DE CAMPO"	+
20/12/2020 13:50:32	CRIAÇÃO ATIVO "CASA DE CAMPO"	+
20/12/2020 13:48:32	LOGIN	+
19/12/2020 14:55:32	LOG OUT	+
19/12/2020 14:40:32	ALTERAÇÃO DE SENHA	+
19/12/2020 13:55:32	EDIÇÃO ATIVO "DISNEY"	+
19/12/2020 13:50:32	LOGIN	+

- HOME
- NOVO ATIVO
- RELATÓRIO ^
- IMPOSTO PAGO
- FINANCEIRO
- LISTAR TODOS ATIVOS
- CONSULTAR LOGS

LOG OUT

PROCURAR

FILTROS ^

☐ até o valor
 ☒ a partir do valor
 ☐ valor exato
 

€100.000,00

## Seja bem vindo ao Asset Manager Y

user123

MEU PATRIMÔNIO

€ 155.000,00

DEPÓSITO A TERMO (50%)  
€ 70.000,00

IMÓVEL ARRENDADO (50%)  
€ 55.000,00

FUNDO DE INVESTIMENTO (20%)  
€ 30.000,00

19

- HOME
- NOVO ATIVO
- RELATÓRIO ^
- IMPOSTO PAGO
- FINANCEIRO**
- LISTAR TODOS ATIVOS
- CONSULTAR LOGS

LOG OUT

PROCURAR

FILTROS

## Financeiro

CONSULTE SEUS LUCROS

MÊS INICIAL: OUTUBRO / 2019

MÊS FINAL: OUTUBRO / 2020

	NOME	LUCRO LÍQUIDO	LUCRO BRUTO	LUCRO LÍQUIDO MÉDIO MENSAL	LUCRO LÍQUIDO MÉDIO MENSAL
	POUPANÇA DISNEY	€ 200,00	€ 220,00	€ 220,00	€ 220,00
	FUNDO PETRO 4	€ 35,50	€ 45,50	€ 45,50	€ 45,50
	CASA DE CAMPO	€ 90,00	€ 100,00	€ 100,00	€ 100,00

- HOME
- NOVO ATIVO
- RELATÓRIO ^
- IMPOSTO PAGO
- FINANCEIRO**
- LISTAR TODOS ATIVOS
- CONSULTAR LOGS

LOG OUT

PROCURAR

FILTROS

## Financeiro

CONSULTE SEUS IMPOSTOS

MÊS PARA CONSULTA

POUPANÇA DISNEY

10/10/19

€ 200,00

10/09/19

€ 200,00

10/08/19

€ 200,00

10/07/19

€ 200,00

10/06/19

€ 200,00

10/05/19

€ 200,00

10/04/19

€ 200,00

MAIS INFORMAÇÕES

+

+

+

- HOME
- NOVO ATIVO
- RELATÓRIO ^
- IMPOSTO PAGO**
- FINANCEIRO
- LISTAR TODOS ATIVOS
- CONSULTAR LOGS

LOG OUT

PROCURAR

FILTROS v

## Imposto Pago

CONSULTE SEUS IMPOSTOS

MES PARA CONSULTA

OUTUBRO / 2019

v

NOME	TOTAL INVESTIDO	MAIS INFORMAÇÕES
POUPANÇA DISNEY	€ 200,00	
FUNDO PETRO 4	€ 35,50	
CASA DE CAMPO	€ 90,00	

- HOME
- NOVO ATIVO
- RELATÓRIO ^
- IMPOSTO PAGO
- FINANCEIRO**
- LISTAR TODOS ATIVOS
- CONSULTAR LOGS

LOG OUT

PROCURAR

FILTROS v

## Listar todos ativos

EDITAR ATIVO

NOME DO ATIVO

Prédio aluguel casa 2

INICIO	20/12/2020	DURAÇÃO	12 meses v	VALOR DO IMÓVEL	€ 170.000,00
IMPOSTO	10%	CONDOMÍNIO	€ 170	RENTA MENSAL	€ 1700,00
				OUTRAS DESPESAS	€ 170

NOME DO ATIVO

Endereço rua 123 legal

RENDIMENTOS TOTAL

LUCRO LÍQUIDO	LUCRO BRUTO	LUCRO LÍQUIDO MÉDIO MENSAL	LUCRO LÍQUIDO MÉDIO MENSAL
€ 200,00	€ 220,00	€ 20,00	€ 22,00

CANCELAR

SALVAR

- HOME
- NOVO ATIVO
- RELATÓRIO ^
- IMPOSTO PAGO
- FINANCEIRO
- LISTAR TODOS ATIVOS
- CONSULTAR LOGS

LOG OUT

FILTROS v

## Listar todos ativos

CONSULTE SEUS ATIVOS

	NOME	TOTAL INVESTIDO	MAIS INFORMAÇÕES
	POUPANÇA DISNEY	€ 2.000,00	
	FUNDO PETRO 4	€ 350,50	
	CASA DE CAMPO	€ 600.000,00	
	POUPANÇA CARRO	€ 5.000,00	
	FUNDO PETRO 5	€ 200,50	
	CASA DE LISBOA	€ 700.000,00	

- HOME
- NOVO ATIVO
- RELATÓRIO ^
- IMPOSTO PAGO
- FINANCEIRO
- LISTAR TODOS ATIVOS
- CONSULTAR LOGS

LOG OUT

FILTROS v

Seja bem vindo ao

## Asset Manager Y

user123

MEU PATRIMÔNIO

€ \*\*\*\*\*

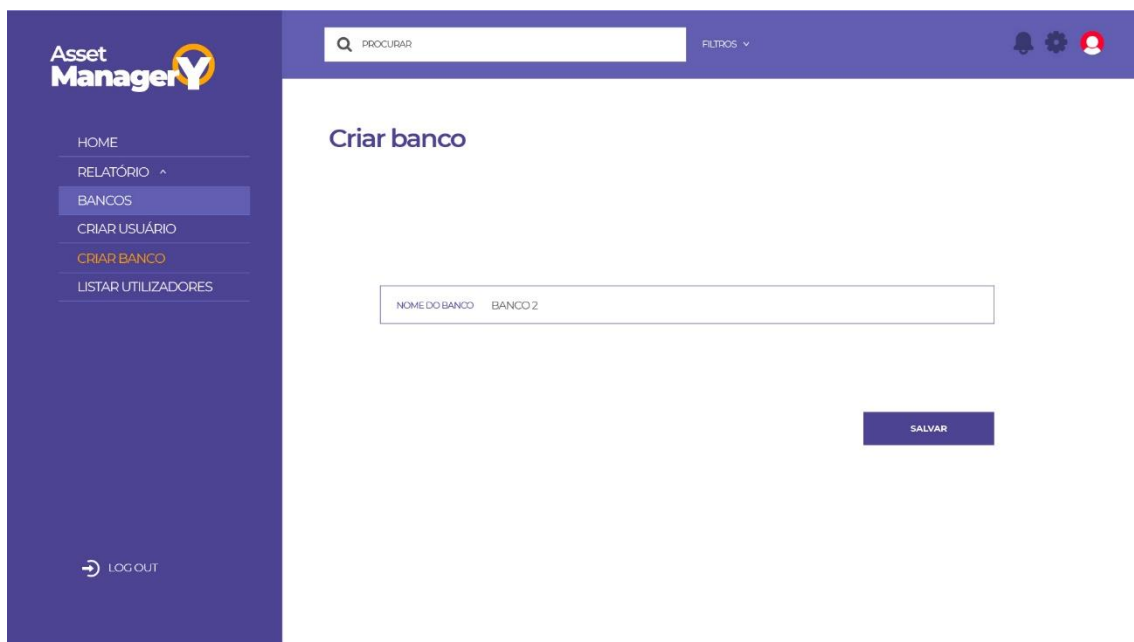
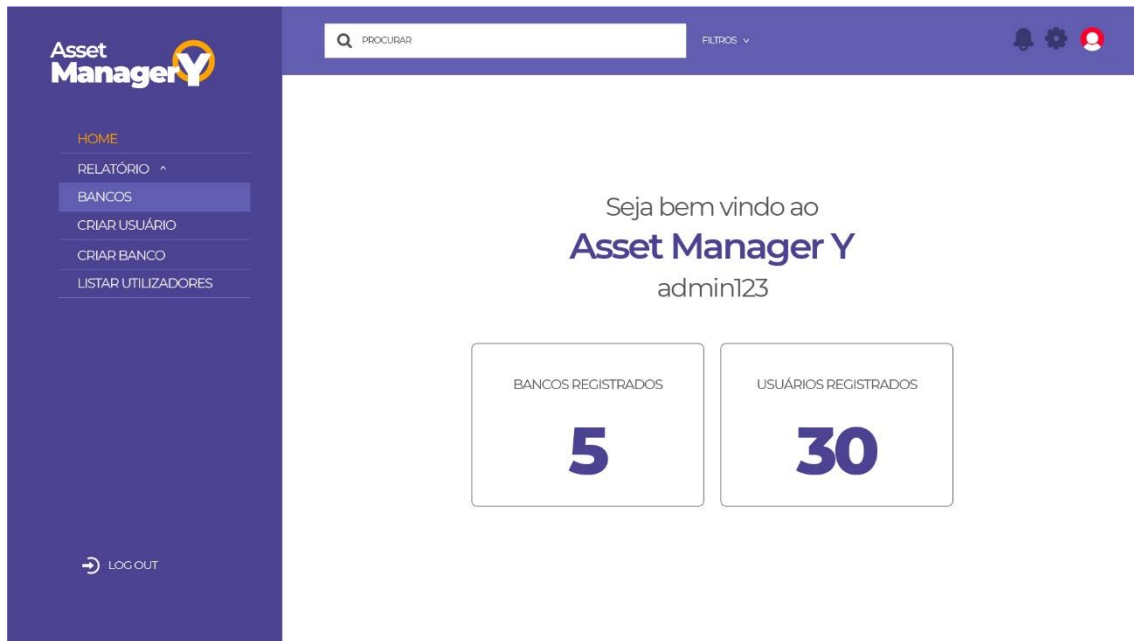
DEPÓSITO A TERMO (50%)  
€ \*\*\*\*\*

IMÓVEL ARRENDADO (30%)  
€ \*\*\*\*\*

FUNDO DE INVESTIMENTO (20%)  
€ \*\*\*\*\*

22

## Modelagem digital de telas - Admin



Asset  
ManagerY

HOME

RELATÓRIO ^

BANCOS

criar usuário




criar banco

listar utilizadores

LOG OUT

Q PROCURAR

FILTROS v



### Criar usuário

Nome de usuário: Usuário 123

SENHA \*\*\*\*\*

CONFIRMAR SENHA \*\*\*\*\*

TIPO DE USUÁRIO: COMUM v

SALVAR

Asset  
ManagerY

HOME

RELATÓRIO ^

BANCOS

criar usuário




criar banco

listar utilizadores

LOG OUT

Q PROCURAR

FILTROS v



### Editar usuário

Nome de usuário atual: Usuário 123

ALTERAR INFORMAÇÕES

Nome de usuário: Usuário 123

SENHA \*\*\*\*\*

CONFIRMAR SENHA \*\*\*\*\*

TIPO DE USUÁRIO: COMUM v

VOLTAR

SALVAR



- HOME
- RELATÓRIO ^
- BANCOS**
- CRIAR USUÁRIO
- CRIAR BANCO
- LISTAR UTILIZADORES

LOG OUT

PROCURAR

FILTROS v

## BANCOS

CONSULTE OS RENDIMENTOS DOS BANCOS

MES INICIAL

OUTUBRO / 2019

v

MES FINAL

OUTUBRO / 2020

v

NOME	TOTAL ACACULADO	EDITAR BANCO
BANCO 1	€ 20.000,00	
BANCO 2	€ 30.000,00	
BANCO 3	€ 40.000,00	

- HOME
- RELATÓRIO ^
- BANCOS
- CRIAR USUÁRIO
- CRIAR BANCO
- LISTAR UTILIZADORES**

LOG OUT

PROCURAR

FILTROS v

## Listar utilizadores

NOME			EDITAR USUÁRIO
USER 1	VERIFICAR LOGS	LISTAR TODOS OS ATIVOS	
USER 2	VERIFICAR LOGS	LISTAR TODOS OS ATIVOS	
USER 3	VERIFICAR LOGS	LISTAR TODOS OS ATIVOS	
USER 4	VERIFICAR LOGS	LISTAR TODOS OS ATIVOS	
USER 5	VERIFICAR LOGS	LISTAR TODOS OS ATIVOS	
USER 6	VERIFICAR LOGS	LISTAR TODOS OS ATIVOS	
ADMIN 1	VERIFICAR LOGS	LISTAR TODOS OS ATIVOS	
ROOT	VERIFICAR LOGS	LISTAR TODOS OS ATIVOS	