Instituto Politécnico de Viseu
Escola Superior de Tecnologia e Gestão de Viseu
Departamento de Informática
Engenharia Informática

Relatório do trabalho de ES1

-16833- Paulo Alexandre
-16813- Francisco Morais
-16798- Pedro Vieira
-16847- Ruben Almeida

Link github: https://github.com/ffmorais99/Manuten-oLivros

Viseu, 2019

Índice

Introdução

Foi nos solicitado, no âmbito da Unidade Curricular de Engenharia de Software I, a criação de um conjunto de funcionalidades que se pudessem inserir no contexto de uma Biblioteca.

Deste modo, o tema escolhido é a manutenção de livros com as funcionalidades de verificação dos livros, analise do relatório, orçamento, pedido de manutenção, receção.

Propusemo-nos a formalizar um sistema que permitirá a Biblioteca, mais propriamente a um funcionário, fazer uma revisão dos livros que, na sua opinião, necessitam de manutenção. De seguida, poderá ainda procurar orçamentos mais adequados e desse modo proceder ao pedido de manutenção desses mesmos livros. Para isso iremos criar algumas classes adicionais.

Esperemos completar com os objetivos propostos

Casos de uso e User Stories

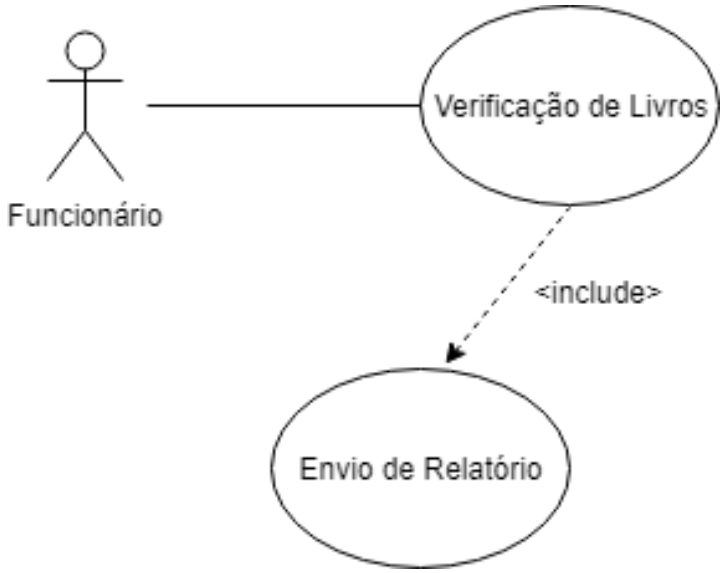

Figura 1 - Caso de uso: Verificação dos Livros

Story id: 1

Titulo: Verificação dos Livros
Enquanto: Funcionário
Quero: Verificar estado dos Livros e realizar relatório
Para: decidir se o livro precisa de arranjo

Confirmação funcional:
O funcionário bibliotecário elabora o relatório que permite identificar quais livros necessitam de reparo.

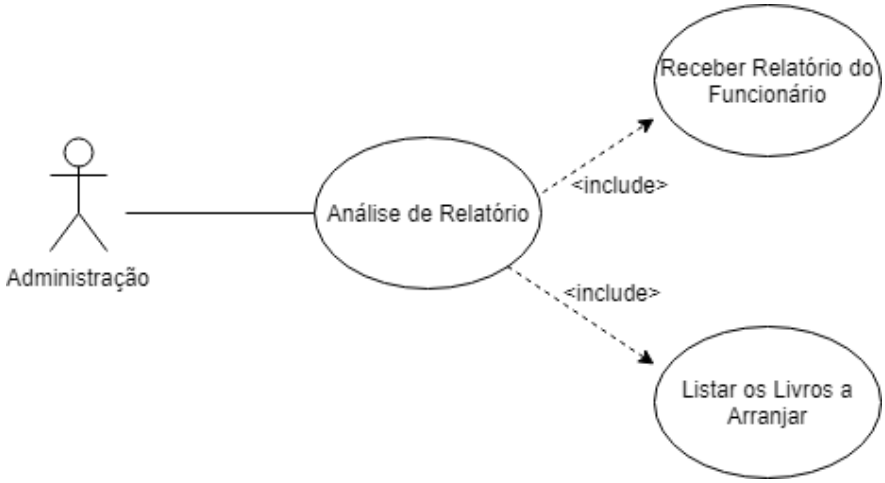

Figura 2 - Caso de uso: Análise de Relatório

Story id: 2

Titulo: Análise de Relatório
Enquanto: Administração
Quero: Analisar o Relatório e escolher os livros a reparar
Para: Sabes quais livros realmente necessitam de reparação e preparar

Confirmação funcional:
O funcionário submete o relatório para sugerir à empresa que livros devem ser reparados A administração averigua esta sugestão aceitando os itens que concorda em existir necessidade de reparação.

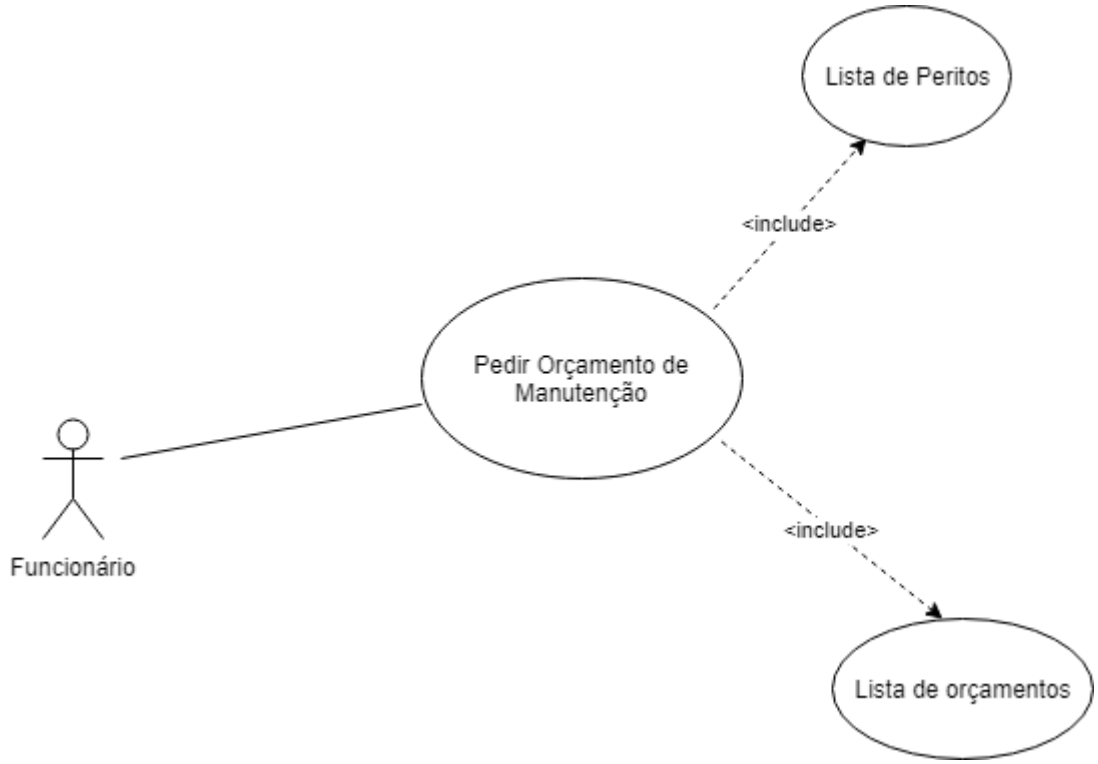

Figura 3 - Caso de uso: Pedir Orçamento

Story id: 3

Titulo:Pedir Orçamento
Enquanto: Funcionário
Quero: Pedir orçamentos aos Peritos Disponiveis
Para: Selecionar Orçamento mais adequado

Confirmação funcional:
O funcionário consulta a lista de peritos ativos.
O funcionário solicita um orçamento a vários Peritos.
O funcionário recebe posteriormente uma lista de orçamentos de todos os peritos contactados.

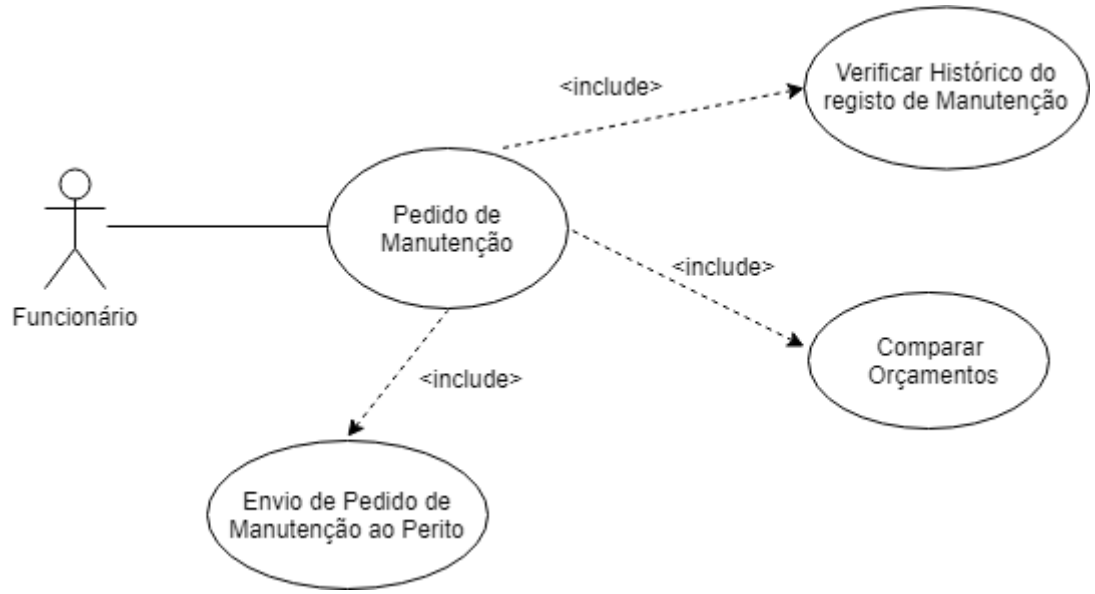

Figura 4 - Caso de uso: Pedido de Manutenção

Story id: 4

Titulo: Pedido de Manutenção
Enquanto: Funcionário
Quero: Selecionar um Orçamento e fazer o pedido de manutenção
Para: Efectuar o pedido de manutenção e ter os livros compostos

Confirmação funcional:
O funcionário verifica o histórico de reparações daqueles livros.
O Funcionário compra todos os orçamentos obtidos.
O funcionário seleciona um orçamento e avança com o pedido de manutenção.

Figura 5 - Caso de uso: Receção do Livro

Story id: 5

Titulo: Receção do livro
Enquanto: Administração
Quero: Registar a receção do livro arranjado
Para: Registar a receção do livro e atualizar o stock disponível na Biblioteca

Confirmação funcional:
O agente bibliotecário atualiza o registo de manutenção do livro se ao verificar a integridade do resultado.
O agente bibliotecário atualiza o stock da biblioteca selecionando os livros recebidos.
O agente bibliotecário envia a fatura para a secção da contabilidade.

Diagrama de atividades

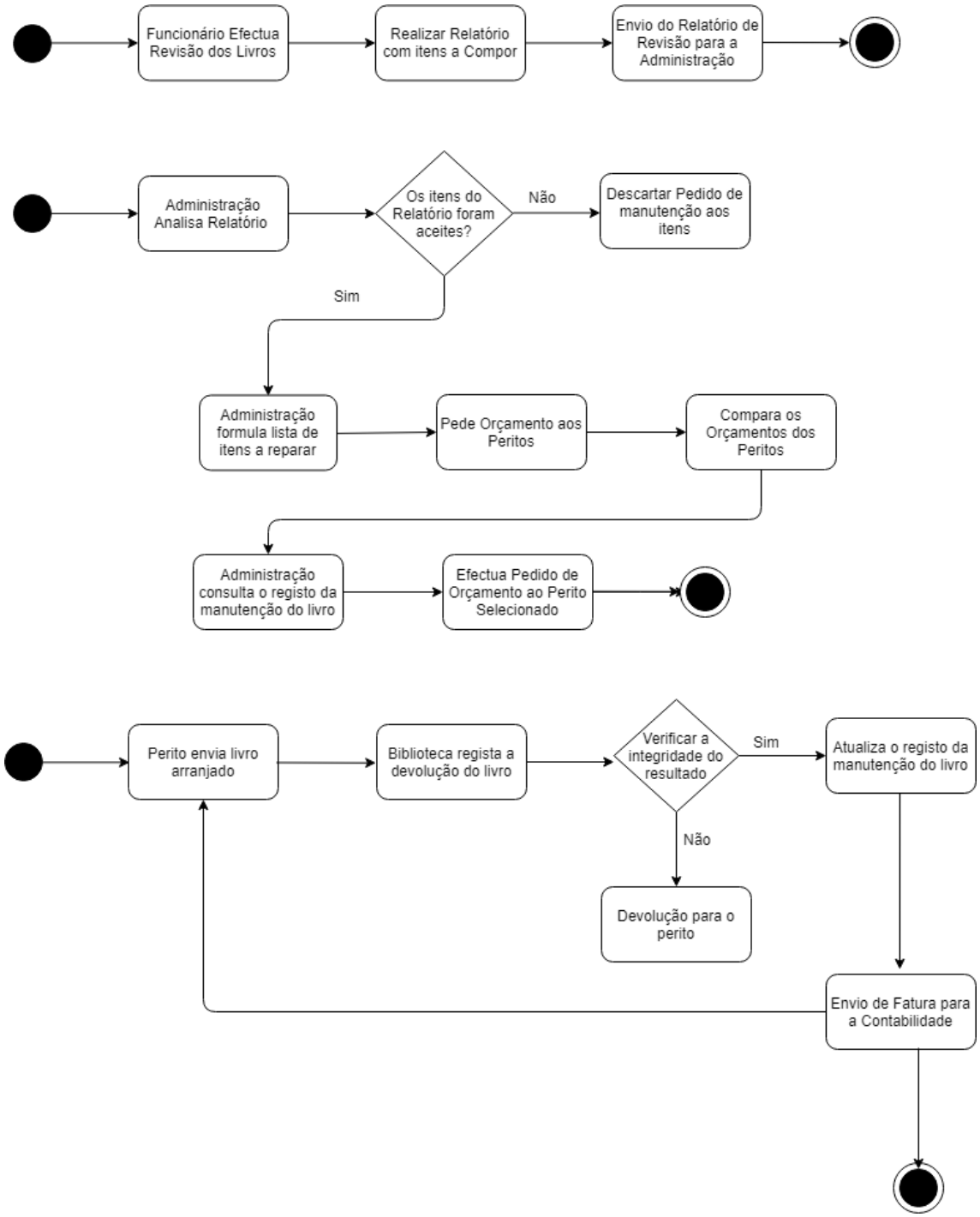

Figura 6 - Diagrama de Atividades

Diagrama de Classes

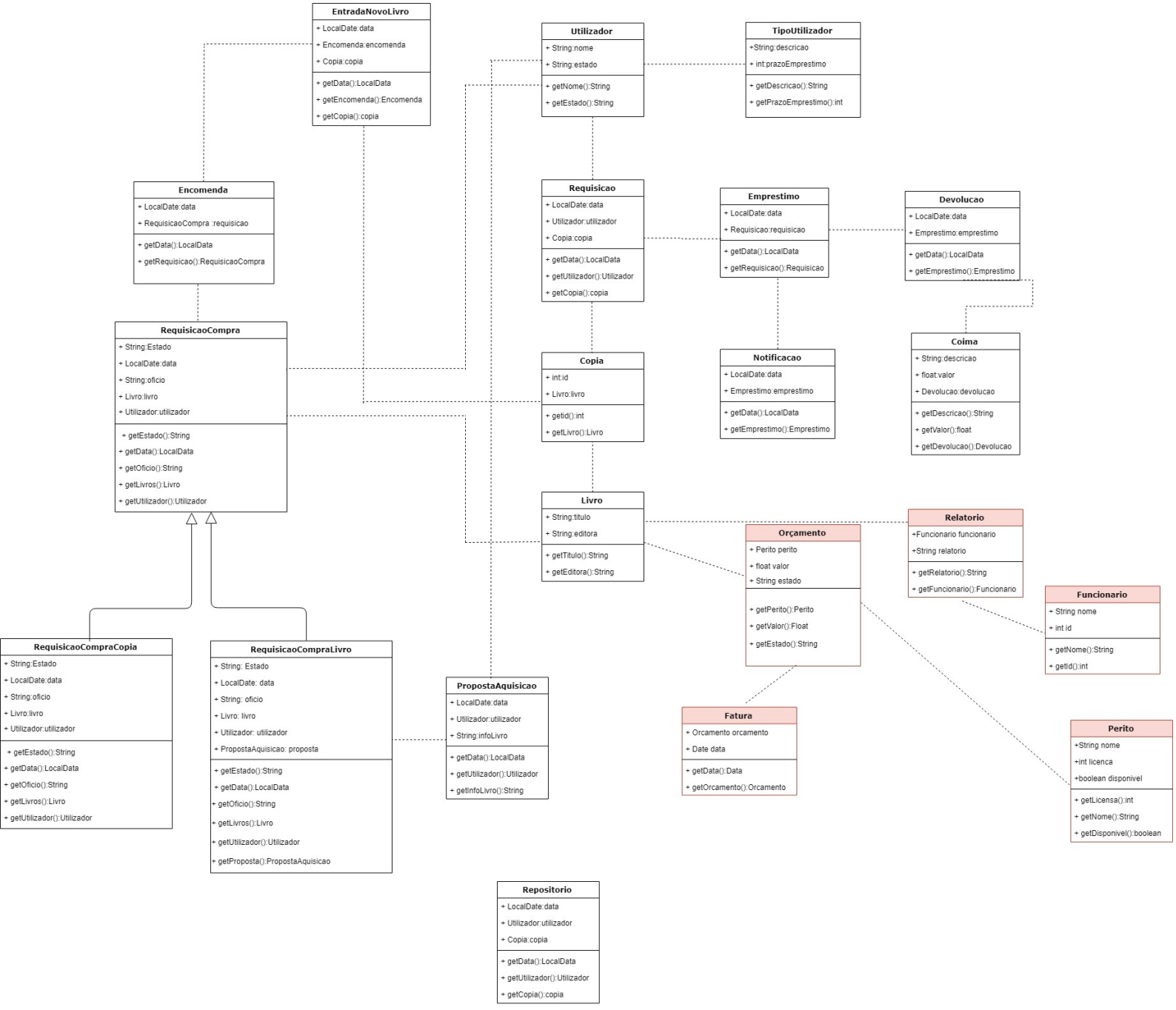

Figura 7 - Diagrama de Classes

Classe Fatura

Figura 8 - Classe Fatura

Fatura.java ESrequisicao1.iml

```
import java.util.Date;

public class Fatura {
    Orcamento orcamento;
    Date data;

    public Fatura(Orcamento orcamento, Date data) {
        this.orcamento = orcamento;
        this.data = data;
    }

    public Orcamento getOrcamento() { return orcamento; }

    public void setOrcamento(Orcamento orcamento) { this.orcamento = orcamento; }

    public Date getData() { return data; }

    public void setData(Date data) { this.data = data; }

}
```

Figura 9-Código da Class fatura

A classe Fatura (Figura 8) está presente devido à funcionalidade de cada Orçamento, aceite pela administração, ter associado a ele uma fatura que contempla o orçamento a que se refere, bem como a data em que foi faturada.

De seguida esta fatura será enviada para a contabilidade, uma lista de faturas que manterá um registo de todas as faturas relativas à manutenção de itens da Biblioteca.

Classe Orçamento

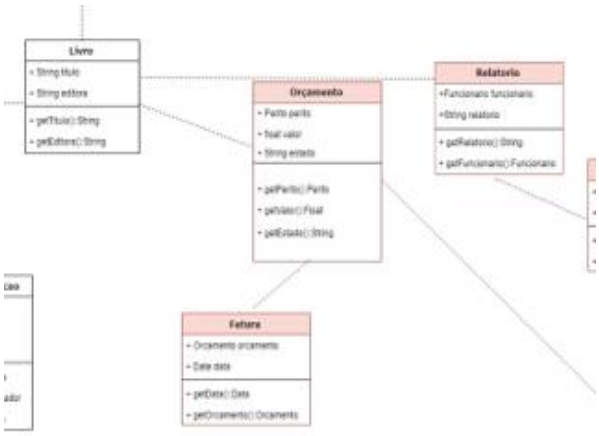

Figura 10 - Classe Orçamento

Orcamento.java | ESrequisicao1.iml

```java
import java.util.ArrayList;

public class Orcamento {
    Perito perito;
    ArrayList<Livro> livros;
    float valor;
    String estado;

    public Orcamento(Perito perito, ArrayList<Livro> livros, float valor, String estado) {
        this.perito = perito;
        this.livros = livros;
        this.valor = valor;
        this.estado = estado;
    }

    public Perito getPerito() { return perito; }

    public void setPerito(Perito perito) { this.perito = perito; }

    public ArrayList<Livro> getLivros() { return livros; }

    public void setLivros(ArrayList<Livro> livros) { this.livros = livros; }

    public float getValor() { return valor; }

    public void setValor(float valor) { this.valor = valor; }

    public String getEstado() { return estado; }

    public void setEstado(String estado) { this.estado = estado; }


}
```

Figura 11-Codigo da class Orcamento

A classe Orçamento (Figura 10) está presente devido à funcionalidade de cada Perito oferecer o seu orçamento após o funcionário solicitar orçamentos aos Peritos da sua lista de Peritos Ativos.

Desse modo a classe Orçamento está relacionado com certo relatório de manutenção, elaborado por um funcionário, e diz respeito a um conjunto de livros a compor, bem como a um Perito de origem.

Classe Relatório

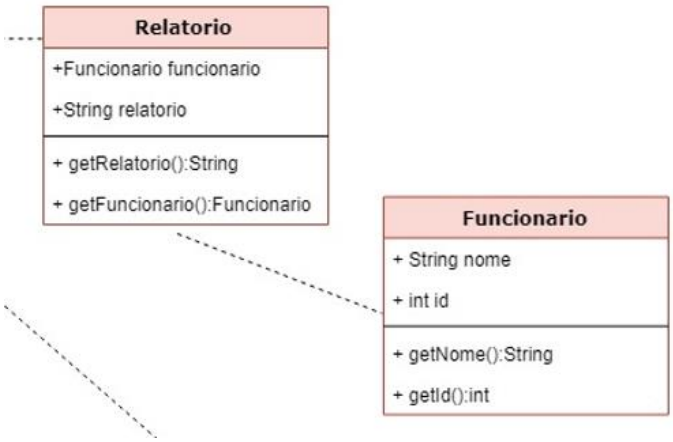

Figura 12 - Classe Relatório

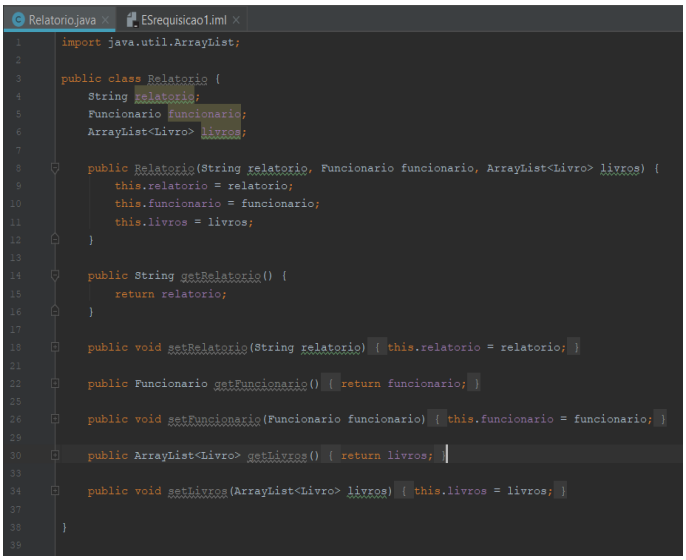

Figura 83-Codigo da class Relatorio

A classe Relatório (Figura 12) está presente devido à funcionalidade de cada pedido de Manutenção que é levado a cabo, ter por base um relatório elaborado por um funcionário da biblioteca.

Este, por sua vez, será aceite pela administração da Biblioteca e posteriormente encaminhado para um Perito.

Classe Funcionário

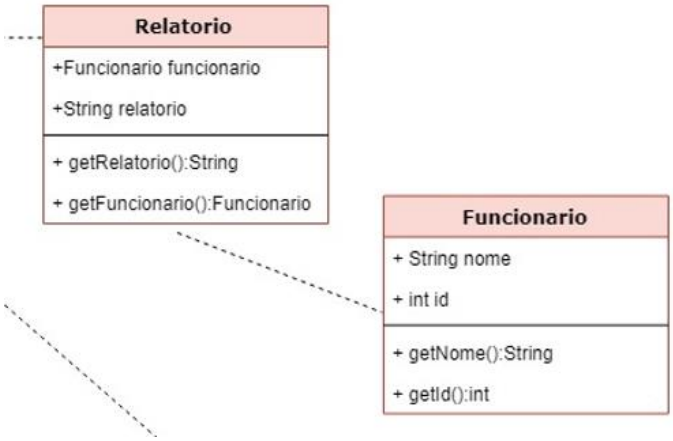

Figura 14 – Classe Funcinário

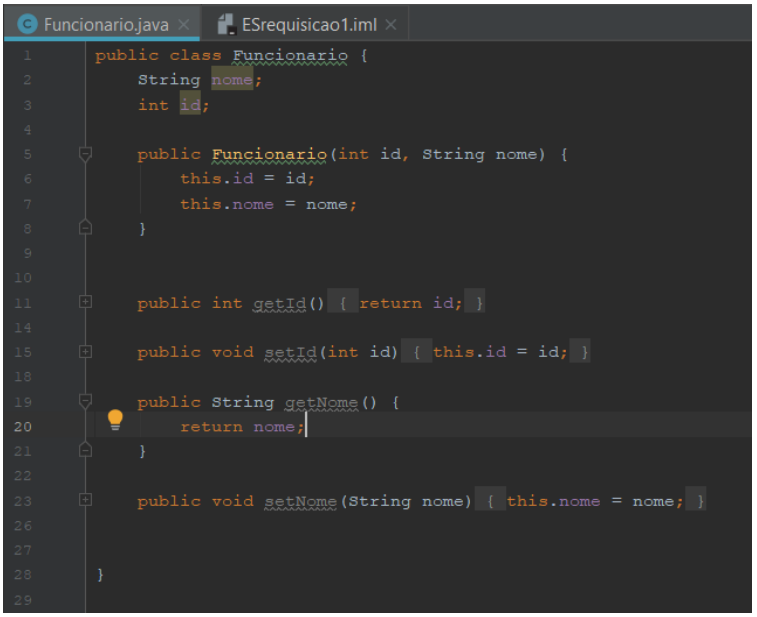

Figura 95-Codigo da class Funcionario

A classe Funcionário (Figura 14) está presente devido à funcionalidade de um Biblioteca ter vários Funcionários e que um funcionário será responsável pela revisão de um conjunto de livros e pela elaboração de um relatório que seja corretamente relacionado com o relatório por este elaborado.

Dessa maneira a administração sabe sempre que Funcionário realizou certo relatório.

Classe Perito

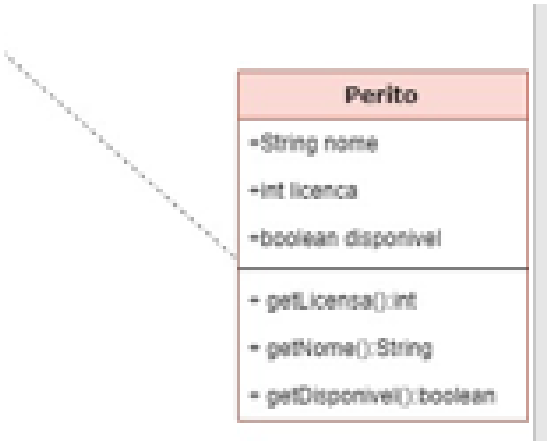

Figura 16 - Classe Perito

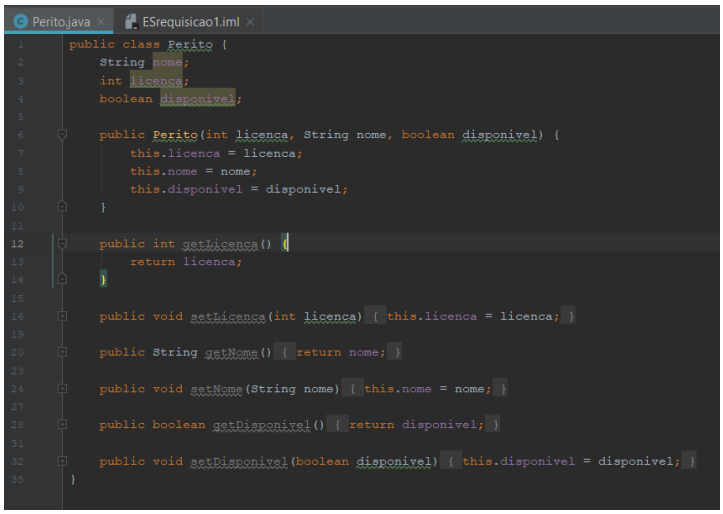

Figura 17- Codigo da class Perito

A classe Perito (Figura 16) está presente devido à funcionalidade de cada Orçamento ter como origem um Perito.

Este Perito contém vários atributos como a sua licença, e um atributo que o identifica como disponível ou não. Os disponíveis poderão ser consultados por funcionários para elaborarem um orçamento para certo pedido de manutenção

Diagrama de Sequencia

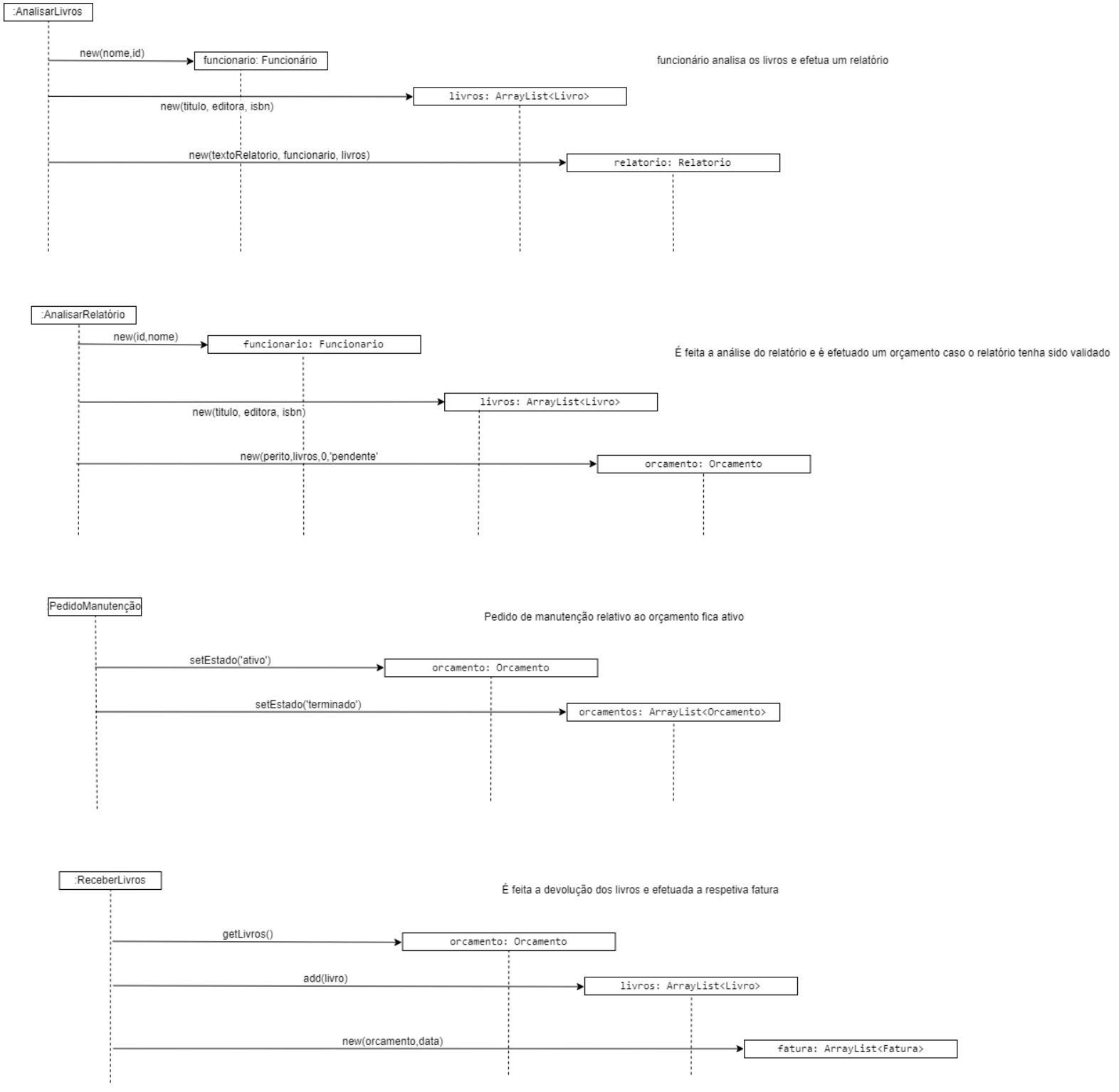

Figura 18 - Diagrama de Sequência

Diagrama de Empacotamento

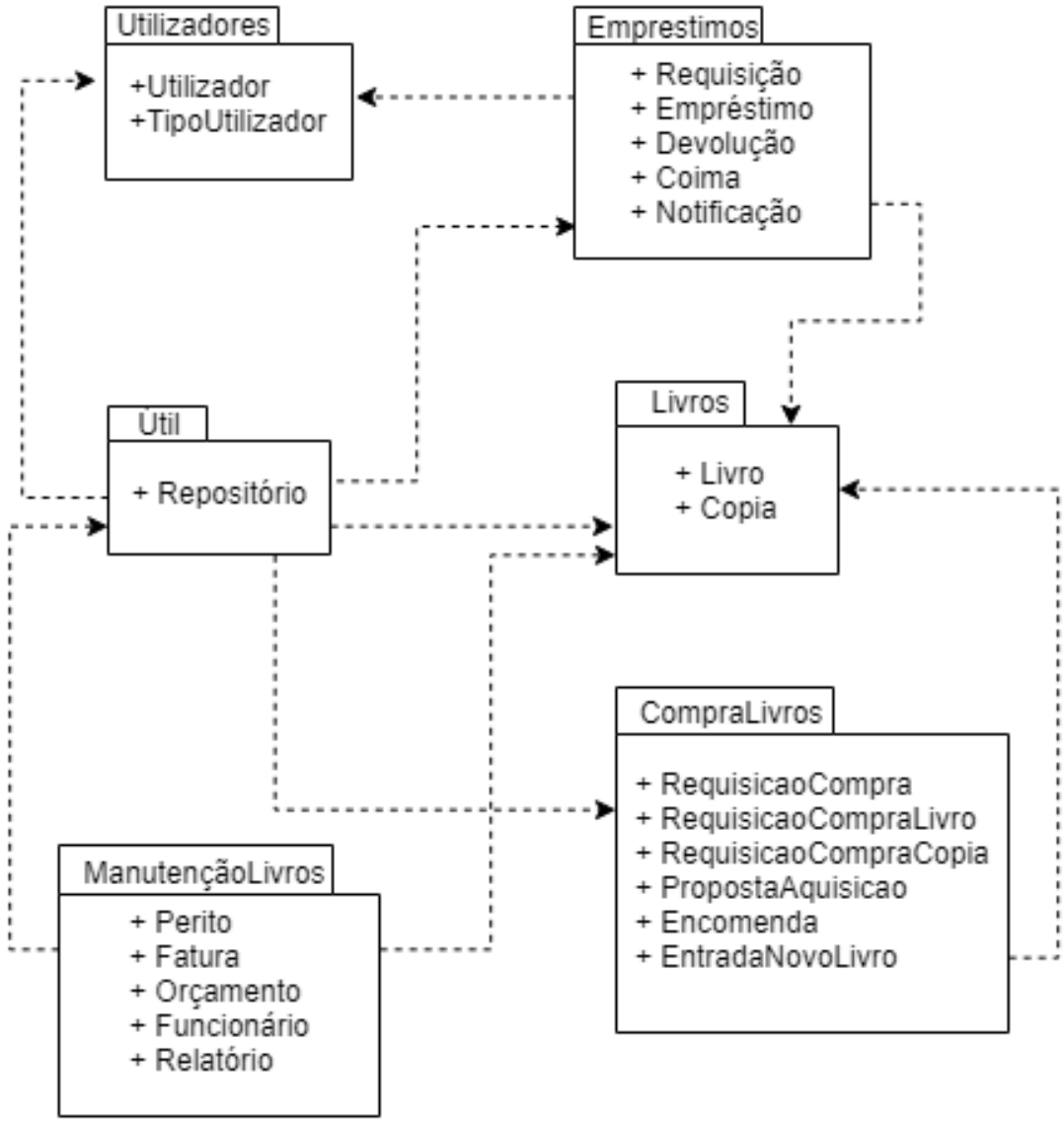

Figura 19 - Diagrama de Empacotamento

Conclusão

Concluindo, este trabalho serviu para aperfeiçoar os nossos conhecimentos sobre engenharia de software nomeadamente user stories, casos de uso, diagramas de atividade, diagramas de sequencias, diagramas de classes, entre outros.

Na realização deste trabalho sentimos algumas dificuldades na elaboração dos diagramas de empacotamento conseguindo, depois de alguma pesquisa, superar essas lacunas. Usámos no planeamento e desenvolvimento deste trabalho a metodologia scrum que aprendemos nas aulas teóricas de engenharia de software I.