Faculdade de Engenharia da Universidade do Porto



*Bike Sharing* – Sistema de Partilha de Bicicletas



**Conceção e Análise de Algoritmos – ??????**

**Elementos do Grupo:**

André Filipe Pinto Esteves 2016???? up2016????@fe.up.pt

Francisco Manuel Canelas Filipe 2016???? up2016????@fe.up.pt

Pedro Miguel Sousa Fernandes 201603846 [up201603846@fe.up.pt](mailto:up201603846@fe.up.pt)

???? de abril de 2018

Índice

[Tema do trabalho 3](#_Toc502940966)

[Solução implementada 3](#_Toc502940967)

[Diagrama de classes UML 4](#_Toc502940968)

[Lista de casos de utilização 5](#_Toc502940969)

[Principais dificuldades 5](#_Toc502940970)

[Contribuição dos elementos 6](#_Toc502940971)

Tema do trabalho

O trabalho tem como objetivo continuar a desenvolver o sistema já criado anteriormente, o software da Imprensa Nacional – Casa da Moeda, e implementar as funcionalidades adicionais explicitamente descritas ao longo deste relatório. Nomeadamente a criação de uma árvore binaria de pesquisa onde guarda as encomendas de publicações, a criação de uma fila de prioridades das publicações por loja e finalmente a conceção de uma tabela de dispersão onde serão guardadas as encomendas suspensas.

Solução implementada

A nossa solução baseia-se nas instruções fornecidas inicialmente, sendo que fomos aplicando as alterações que achamos necessárias para uma melhor estrutura do programa. Assim sendo, o projeto revolve à volta das classes principais e dos seus métodos: Empresa, Publicação (Livro ou Revista), Loja, Empregado e Pedido.

Além destas classes que tiveram de ser revistas chegamos a conclusão que tínhamos de remover a classe *Collection* e as suas funções derivadas pois complicavam desnecessariamente o desenvolvimento do programa. Continuamos a utilizar as estruturas anteriormente implementadas para as datas o que ajudou consideravelmente na implementação das novas funcionalidades:

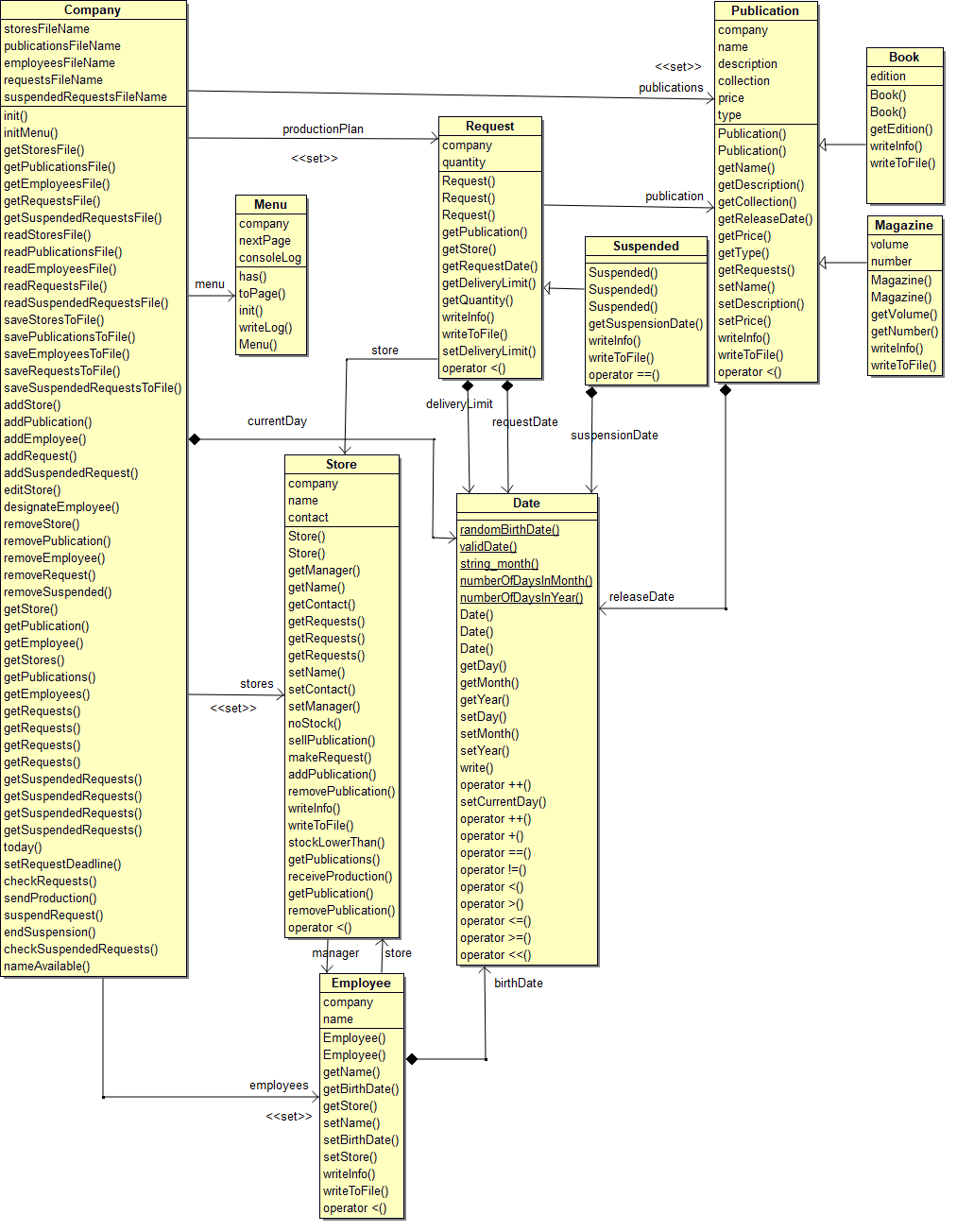
* Árvore binária de pesquisa
* Fila de prioridade
* Tabela de dispersão

Para suportar estas estruturas foram alteradas as classes existentes, mas também criadas novas: *LocalPublication*, que guarda as publicações nas lojas, e *Suspended,* classe derivada de *Request*, para gerir as encomendas suspensas na tabela de dispersão.

Além destas funcionalidades adicionais pedidas o nosso trabalho tivemos de completar certos objetivos que tínhamos desde o projeto anterior.

Quanto à interface principal adicionamos vários painéis adicionais baseados nos métodos já existentes, para que o utilizador pudesse ter uma interação fluida com o programa, sendo apresentado todas as instruções necessárias para a utilização correta da aplicação. É também mantido um registo de atividades, que pode também ser mostrado no ecrã.

Diagrama de classes UML



Lista de casos de utilização

O programa desenvolvido permite efetuar as seguintes operações, que complementam as do projeto passado:

* Editar lojas (nome, contacto, designar responsáveis);
* Gerir encomendas:
  + Listar (podendo-se filtrar por loja ou publicação);
  + Criar e remover;
  + Suspender e terminar suspensão.
  + Alterar data limite de execução.
* Gerir publicações:
  + Identificar publicações com stock inferior a um valor dado pelo utilizador.

Apesar de não estar diretamente relacionado com a interface, foram implementados outros métodos que permitem:

* Verificar, no inicio do programa, o estado dos pedidos, tanto dos ativos como suspensos. Desta forma, caso um pedido esteja concluído (se tiver chegado o dia limite), é entregue à loja a produção. Também, caso se conclua que um pedido esteve suspenso durante demasiado tempo, este é permanentemente cancelado.
* Ordenar crescentemente as publicações na loja pela sua quantidade em stock, e atualizá-lo conforme uma produção é recebida.

Principais dificuldades

Um dos nossos problemas que encontramos no desenvolvimento do nosso programa foi o facto de termos criado anteriormente uma classe *Collection* (Coleção), que continha duas subclasses *BookCollection* (Livro) e *MagazineCollection* (Revista), que complicava o programa, pois tínhamos uma visão futura diferente da que foi proposta. Com base nisto decidimos retirar essa mesma classe e funções derivadas, mas essa mesma decisão fez com que tivéssemos de rever diversos aspetos que a essa classe abrangia. Além disso também o trabalho foi concebido num período de ferias onde ninguém vivia na mesma cidade o que complicou o desenvolvimento do programa o que nos fez trabalhar mais no mês de janeiro o que se traduz num período mais curto para o desenvolvimento do trabalho.

Contribuição dos elementos

O Bruno tratou da remoção da classe Coleções e a nova implementação de métodos para o normal funcionamento do programa. Trabalhamos todos no *debugging* do programa ao longo do seu desenvolvimento, mas o Pedro destacou se neste aspeto do projeto. O Pedro foi também responsável por mapear as estruturas dos novos métodos a ser implementados, para que todos os elementos do grupo pudessem trabalhar em uníssono, já que enquanto isso o Christopher se ocupou com o relatório, diagrama de classes e documentação *doxygen*.