

## Projeto LP2 - Relatório

### 1 - Introdução

O projeto se trata de um sistema de gerenciamento de restaurante. Assim como alguns restaurantes possuem aparelhos para auto-atendimento de usuários (os “totens”), nossa aplicação simula um desses aparelhos para ajudar o usuário (cliente ou funcionário) na realização e pagamento de pedidos, para clientes, e na visualização de dados sobre o restaurante, para o funcionário. Esse tipo de sistema é bastante relevante para facilitar a organização do estabelecimento e segue de acordo com a tendência de automação e auto-atendimento, que também pode ser vista mais recentemente em supermercados.

### 2 - Metodologia de Desenvolvimento

Primeiramente desenvolvemos um diagrama de classes inicial para ajudar a termos um norte e construir os primeiros passos do projeto. Após as concepções iniciais, começamos a desenvolver as classes de modelo, depois as classes de controle. Após já ter caminhado com o desenvolvimento dessas classes, começamos a criar as telas que compõem a interface gráfica, utilizando JavaFX e os arquivos .fxml, cada um com sua devida classe controladora.

Todas as nomeações (classes, variáveis, etc.) seguem o padrão Camel Case.

A organização estrutural foi baseada no modelo MVC (modelo, visão e controle), com a adição do pacote “dao”.

- Classes que servem de modelo para objetos e outras classes estão no pacote “modelo”.
- Classes que são responsáveis por armazenar instâncias de objetos ou manipular informações específicas de cada classe estão no pacote “dao”.
- Classes responsáveis por gerenciar/controlar as telas da interface gráfica estão no pacote “controle”.
- Os arquivos .fxml que definem as telas da interface gráfica estão no pacote “visao”.

O diagrama de classes está presente no arquivo **Diagrama\_de\_Classes\_Final.pdf** presente na pasta do projeto. As classes controladoras da interface gráfica, presentes no pacote “controle”, não foram incluídas no diagrama de classes.

### 3 - Dificuldades Encontradas

Algumas ideias originais, como a implementação de uma fila para a cozinha onde o cliente retira o pedido e esse pedido retirado sai da fila, foram descartadas por inviabilizar o escopo do projeto dentro do prazo. Na sua implementação final, a fila da cozinha (FilaDaCozinha.java) funciona como um banco de dados para todos os pedidos feitos, e serve para mostrar para o funcionário um histórico de todos os pedidos feitos desde que o programa foi aberto.

Uma dificuldade encontrada foi como tornar o ato de realizar o pedido intuitivo e de acordo com a definição de que o pedido aceita repetições (pode-se pedir mais de uma unidade do item), para isso chegamos na estrutura de um CheckBox e um TextField para cada item do cardápio, onde a CheckBox estar marcada significa que você quer pedir aquilo e o número no TextField representa a quantidade desejada. Todas as devidas checagens de situações inválidas (como quantidade ausente ou menor que 1) foram tratadas.

Outra dificuldade encontrada foi na utilização da função sleep(), nativa do Java, que seria usada para só abrir a tela seguinte com um certo atraso, e isso serviria para o usuário ter tempo de ler o que foi exibido na tela após o clique no botão. Como a função não estava funcionando corretamente, mudamos a maneira como a informação era exibida (desatrelando-a do botão que abre a próxima tela) e não foi mais necessário utilizá-la.

Por fim, encontramos bastante dificuldade para conseguir executar o programa diretamente por um arquivo .jar executável, para abrir a aplicação de forma mais intuitiva. Sem encontrar solução, deixamos de lado a ideia. Mais detalhes estão em README.txt.

### 4 - Conclusão

De maneira geral, o projeto alcançou seu objetivo inicial, entregando um sistema que permite ao usuário, seja cliente ou funcionário, executar ações relevantes ao ambiente de um restaurante (pedir, pagar, ver histórico de pedidos, etc.). Desde o início até a sua conclusão, trabalhamos em conjunto para que o desenvolvimento do trabalho ocorresse da melhor forma possível, dividindo algumas responsabilidades entre os integrantes, para que o projeto funcionasse sem nenhum problema.

Entrementes, os conhecimentos e práticas adquiridas são de grande valia para ambos os integrantes.

### Referências

Não utilizamos nenhuma referência externa além do conteúdo da própria disciplina.