

Nome do grupo:

1. Caio Giovanni Pereira Vasconcelos
2. Caio Luna Farias

Nome do projeto:

Coffe

Repositório de código no GITHUB:

<https://github.com/Kingcondor/Cafeteria.git>

Descrição geral do projeto

Coffe é um programa que visa ser um banco de dados das operações para o estabelecimento, visando o uso do consumidor e do administrador de maneiras distintas, sendo seus modos de operar divididos em três grupos de usuários:

- Gerente: Organizar o registro de funcionários e modificá-los se assim desejar (Demissão e contratação de funcionários, registro de pagamento, registro de promoção salarial, registro de comparecimento dos funcionários, estatísticas detalhadas com relação a atividade diária dos clientes e dos funcionários, a quantidade de produtos no estoque, emissão de alertas a todos os usuários (De modo geral e individual), poder de invalidação do pedido mediante erro de solicitação de qualquer usuário);
- Funcionários: Com a finalidade de definir um registro de sua atividade diária, registro do pagamento salarial (Alerta) pessoal do usuário, registro de promoção salarial (Alerta) pessoal do usuário, registro de comparecimento pessoal do usuário, estatísticas de trabalho diário (Histórico de vendas), quantidade de produtos no estoque, recebimento de alerta do gerente, recebimento do requerimento do pedido feito pelo cliente, poder de invalidação do pedido mediante erro de solicitação apenas de transações realizadas pelo usuário (funcionário). Além disso a classe de funcionários é dividida em dois grupos, os atendentes com acesso a todas as funcionalidades supracitadas e os comuns com acesso apenas a consulta de dados pessoais (sem direito a realização de transação ou interação com a classe cliente);
- Clientes: Tem acesso ao cardápio completo de produtos podendo fazer o pedido pelo próprio e marcando seu registro de pagamento, apenas para ser pago mediante a apresentação do mesmo, após a realização do pedido no aplicativo um atendente será designado para atendê-lo;

Como dito nos tópicos acima, a Cafeteria tem três entidades gerais, podendo elas serem modificadas pelo usuário no caso de ter sua classe permitida:

- Pagamento: Está entidade apenas pode ser modificada pelo usuário gerente, se refere a transação do cliente, e ao histórico de vendas;
- Estoque: Entidade que contabiliza a quantidade de produtos disponíveis para venda;
- Produto: Objeto a ser vendido pela classe funcionário para a classe cliente, esta classe tem um valor, um nome e uma descrição;

Requisitos do projeto

REQ1. O sistema será separado em três tipos de usuários: Gerente, cliente e funcionários (atendentes e comuns);

REQ2. O sistema terá como entrada para o usuário a utilização de login e de uma senha registrados pelo usuário no momento do cadastro, podendo ser modificado;

REQ3. O sistema deve permitir a venda de produtos cadastrados no sistema, de quantidade positiva em estoque, salvar no histórico dos três usuários de forma individual;

REQ4. O sistema deve permitir o gerenciamento de usuários dos tipos: cliente e funcionário através do usuário do tipo gerente;

REQ5. O sistema deve registrar as transações referentes a interação entre os usuários do tipo cliente e a entidade produto;

REQ6. O sistema deve registrar o debito de produtos compradas pelo usuário do tipo cliente;

REQ7. O sistema deve permitir que a usuário do tipo gerente emita alertas de caráter textual, que serão mostrados, no tipo de usuário (ou usuário específico) designado pelo gerente, no momento da entrada do usuário alvo;

REQ8. O sistema deve mostrar ao usuário do tipo gerente, se solicitado, a estatística diária e mensal registrada, referente a venda dos produtos e as informações pessoais dos funcionários;

REQ9. O sistema deve permitir que o usuário do tipo funcionário verifique sua atividade diária e dê sua atividade de comparecimento do dia atual como verificada;

REQ10. O sistema deve permitir que o usuário do tipo cliente selecione o produto desejado e solicite a entrega para o funcionário do tipo atendente;

REQ11. O sistema deve permitir que o usuário do tipo atendente tenha acesso ao registro de transações com o usuário do tipo cliente;

Arquitetura

O programa começa na classe Main que instancia um ControladorFacade, que em caso de falhas nas funções básicas do programa um erro é exibido e o programa termina. Caso não haja erros, mas a GUI tiver algum, o modo de Terminal é iniciado no lugar.

Tanto o Terminal quanto a GUI chamam os métodos do ControladorFacade durante a comunicação com o usuário. Essa facade une todos os controladores, que têm acesso aos repositórios que foram construídos seguindo o *design pattern* Singleton. Os repositórios implementam uma interface cada um, garantindo que determinados métodos sempre vão estar disponíveis independentemente da implementação de cada repositório.

As informações de que tipo de usuário (Conta) está sendo tratado é sempre transmitida com o *enum* Tipo para restringir as opções apenas às disponíveis.

A GUI envia a própria classe Main para cada Controller iniciado, isso permite que ele tenha acesso ao ControladorFacade e um dos métodos `showOnBorder` que se sobrepõe ao outro tendo o parâmetro *tittle* como diferencial. Esse método é responsável por criar pequenas janelas (uma nova *Scene*) por cima da janela principal para exibir mensagens ou obter informações do usuário, enquanto o `showOnBorder` sem o parâmetro *tittle* é responsável pela janela principal.



Introdução à Programação II - Projeto
Entrega 01 – Grupo, descrição geral e requisitos

Os beans fazem parte do funcionamento interno do programa, sendo que a view não tem conhecimento de quais beans são utilizados (se comunicando através do *enum* Tipo, que dificilmente mudará pois depende da organização de uma cafeteria, não do funcionamento interno do programa).