



# GRUPO 15

---

Gabriel Henrique de Souza Nunes

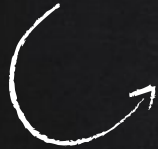
Lucas Leandro Ribeiro

Maíla Ferreira Silva

Samuel Fantini Braga



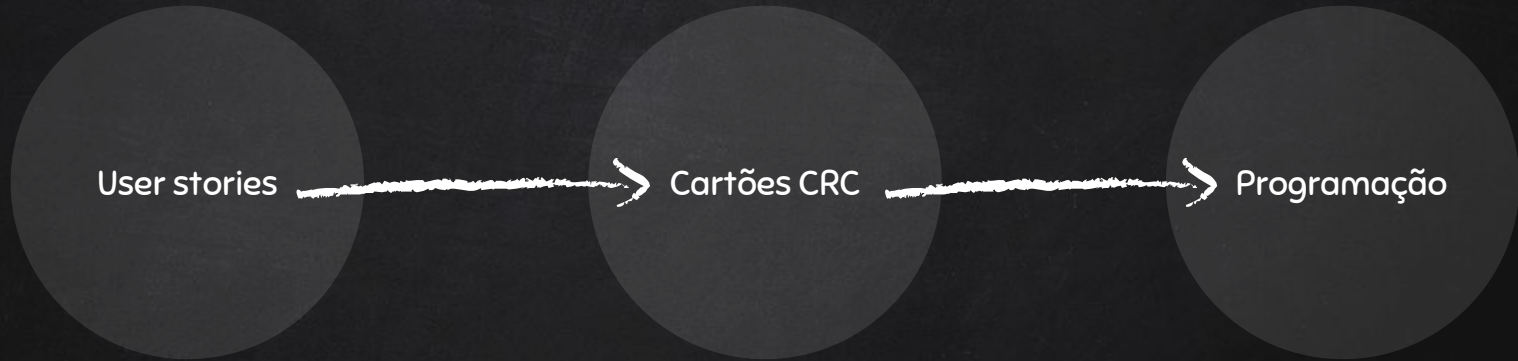
# PROGRAMAÇÃO E DESENVOLVIMENTO DE SOFTWARE II: FRENTE DE LOJA



Gerenciamento e controle de estoque de uma loja de pequeno/ médio porte



## DESENVOLVIMENTO DO PROJETO

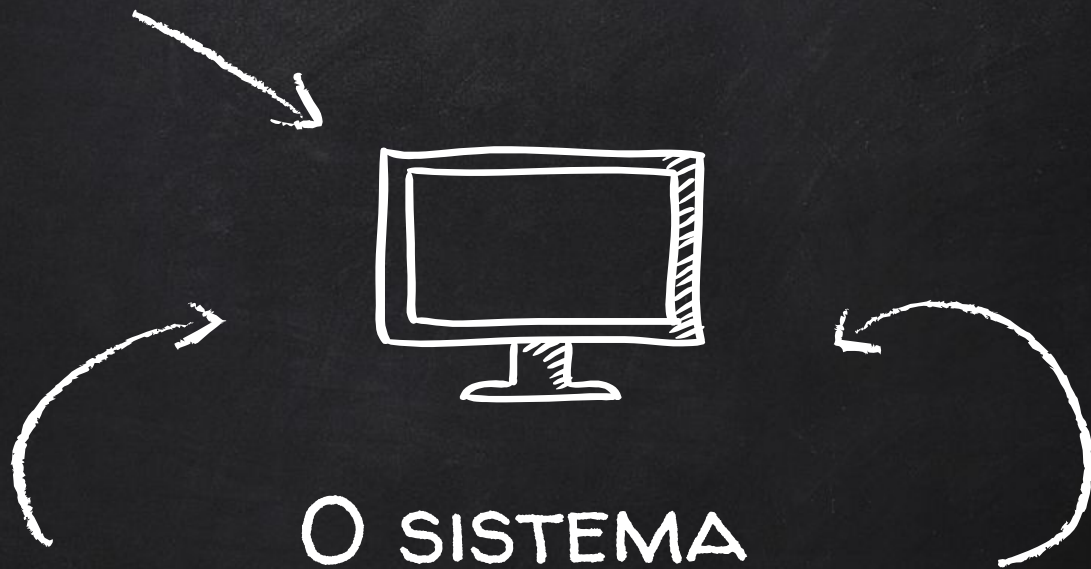




Como funcionário da loja eu  
quero vender e, assim, retirar  
produtos do estoque para que  
conste a venda



Como gerente da loja eu quero  
ter uma visão geral do estoque  
para gerenciar a quantidade de  
produtos e poder tomar  
melhores decisões de compra

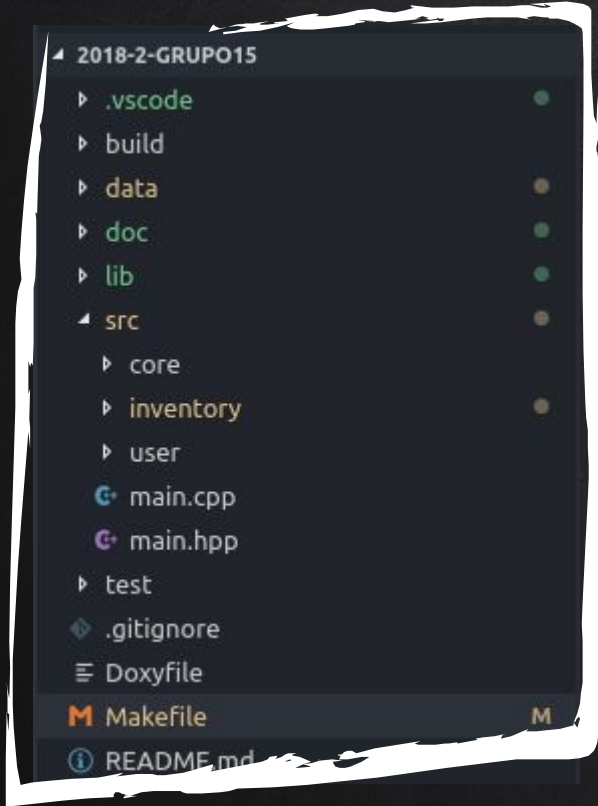






## A ESTRUTURA DO CÓDIGO

- ✗ Uma pasta por namespace
- ✗ Pasta lib
- ✗ Pasta para os arquivos de dados
- ✗ Uma pasta para testes
- ✗ Makefile e README





```
class Session {
private:
    Person* _currentUser;
    Stock* _stock;
    Team* _team;
public:
    /**
     * Construtor que define os dados iniciais da sessão.
     * Aqui são inicializados os dados de estoque e da equipe
     * a partir dos arquivos '.data'.
     * O '_currentUser' é inicializado com 'nullptr'.
     */
    Session();
    /**
     * Retorna o usuário logado.
     * @return o valor do '_currentUser', que será 'nullptr' caso
     * o usuário não esteja logado.
     */
    Person* getCurrentUser();
    Stock* getStock();
    Team* getTeam();
    void setCurrentUser(Person*);
    void logout();
};
```

## SESSION

- ✗ Armazena o estado do sistema e gerencia a autenticação do usuário





```
/**
 * Classe para manipulação dos arquivos de dados.
 *
 * Nesta classe é feito o "parsing" dos arquivos "*.data" e "*.idx.data"
 * que armazenam os dados dos funcionários, das vendas e dos produtos.
 */
class DataFile {
private:
    const string delimiter = "=";
    string _filePath;
    map<string, string> _parameters;
    void load();
public:
    /**
     * Construtor que define o caminho do arquivo a ser manipulado.
     *
     * @param filePath caminho para o arquivo a ser lido ou escrito
     * @param loadData caso seja 'true', o conteúdo atual do arquivo será
     *                processado e carregado no map '_parameters'
     */
    DataFile(string filePath, bool loadData = true);
    /**
     * Retorna o valor de um parâmetro.
     *
     * @param key chave do parâmetro
     * @return valor do parâmetro ou uma string vazia
     */
    string getParameter(string key);
};
```

## DATAFILE

x É responsável pela  
leitura e escrita dos arquivos  
que armazenam os dados do  
sistema



THANKS!

Dúvidas?