

Roteiro do Vídeo Final

Sistema VinilStore – Projeto de Programação 3

Equipe de Desenvolvimento – UPE/POLI

1 Introdução Geral

2 Introdução

Este documento reúne o roteiro utilizado para a gravação do vídeo final do Projeto de Programação 3. O software implementa o gerenciamento completo de uma loja, incluindo cadastros, controle de acesso, fluxo de login, navegação entre telas, manipulação de estoque, carrinho e integração com banco de dados.

O objetivo deste roteiro é apresentar, de forma clara e organizada, como cada parte do sistema funciona e como cada integrante contribuiu para sua construção, garantindo uma apresentação coerente e completa durante o vídeo final.

3 Gabriel – Introdução e Ecossistema do Projeto

3.1 Ecossistema do Sistema

Gabriel explica o ecossistema completo do projeto, baseado na estrutura real do repositório:

- Apresentação das pastas e arquivos: .cpp, .hpp, .ui.
- Como o Qt Creator organiza as telas.
- Função de cada módulo:
 - Cadastro
 - Estoque
 - Controle da Loja
 - Gerente
 - Primeiro Login
 - Registro

- Tela Principal (cliente)
- Explicação sobre como cada tela é independente, mas totalmente conectada.

3.2 Tela de Seleção Inicial

Gabriel mostra a tela com os vários botões (Ajuda, Cadastro, Estoque, Registro etc.) e explica que ela funciona como um hub de desenvolvimento e navegação.

4 Igor – Primeiro Acesso (Registro Inicial)

4.1 Lógica do Primeiro Login

É explicado que:

- A tela de **Primeiro Acesso / Primeiro Login** só aparece caso o banco ainda não tenha o dono da loja cadastrado.
- O sistema verifica a tabela `dono_da_loja`.
- Caso exista dono cadastrado, a tela não aparece mais.

4.2 Fluxo Completo

- Usuário abre o sistema pela primeira vez.
- É direcionado para a tela de criar conta.
- Após clicar em *Create Account*:
 - o registro é salvo no banco;
 - a tela fecha automaticamente;
 - o sistema abre a tela de login normal;
 - o primeiro acesso nunca mais aparece.

4.3 Funções Técnicas

- Função que verifica se o primeiro acesso deve aparecer.
 - Função que impede múltiplos cadastros de dono.
 - Fechamento da janela e redirecionamento automático.
-

5 Vinícius – Tela Principal do Cliente

5.1 Descrição da Interface

Vinícius mostra a tela do cliente, onde aparecem:

- Lista de discos disponíveis.
- Abas “Discos Usados” e “Discos Novos”.
- Botão **Adicionar** para enviar ao carrinho.
- Botão **Carrinho**.

5.2 Fluxo do Cliente

- Seleciona o disco.
 - Adiciona ao carrinho.
 - Visualiza o carrinho.
 - Prossegue para pagamento.
-

6 Juliana – Fluxo do Código e Identificação do Usuário

6.1 Visão Geral do Funcionamento Interno

Essa parte consiste em explicar **como o código identifica automaticamente quem está fazendo login no sistema**. Esse é o núcleo lógico do fluxo do programa, pois define para onde o usuário será redirecionado e quais permissões ele terá.

O sistema VinilStore possui três perfis que podem realizar login:

- **Cliente**
- **Funcionário**
- **Gerente** (um funcionário com permissões superiores)

Toda a autenticação é feita consultando as informações salvas no banco SQLite.

6.2 Onde os Cadastros São Feitos

Juliana explica que:

- A tela do **Gerente** é responsável por cadastrar:
 - novos funcionários,
 - novos clientes,
 - novos gerentes.
- Os dados são salvos nas tabelas:
 - **clientes**
 - **funcionarios** (onde também ficam os gerentes)

Assim, quando alguém tenta logar, o sistema só reconhece porque ele foi previamente registrado através da tela de Gerente.

6.3 Processo de Login – Análise e Conexão com o Banco

Quando o usuário insere login e senha:

1. O programa captura os valores digitados.
2. Realiza consultas SQL nas tabelas correspondentes.
3. Analisa o nível de acesso com base nos dados encontrados.

A ordem da verificação é a seguinte:

- Primeiro verifica na tabela **funcionarios**.
 - Depois verifica na tabela **clientes**.
 - Se não encontrar em nenhuma, acusa erro.
-

6.4 Critérios de Identificação

A comparação no banco funciona assim:

- **Usuário encontrado em funcionários:**

- Se o campo `cargo ==` Gerente → abre a tela do gerente.
- Se `cargo ==` Funcionário → abre a tela de funcionário.

- **Usuário encontrado em clientes:**

- Abre a tela principal do cliente.

- **Usuário não encontrado:**

- Exibe mensagem de erro.
-

6.5 Redirecionamento Automático

Após identificar quem é o usuário, o sistema:

1. Fecha a tela de login.
2. Abre automaticamente a tela correspondente ao perfil.

Isso é feito com as funções do Qt:

- `hide()`
 - `show()`
 - `close()`
-

6.6 Importância da Lógica

Juliana finaliza mostrando que:

- Esse fluxo garante que cada usuário só acesse o que deve acessar.
 - Ele impede que clientes abram telas de funcionários.
 - Ele impede que funcionários abram telas de gerente.
 - É o coração da segurança e da estrutura do sistema.
-

7 Diego – Chamando Funções no Arquivo .ui

7.1 Explicação Técnica

Diego demonstra:

- Como conectar botões e widgets usando `signals` e `slots`.
- Como associar cada botão às suas funções no `.cpp`.
- Como tratar eventos de clique, abrir janelas e passar dados entre telas.

7.2 Casos Práticos

- Botão “Cadastro” abrindo a tela Cadastro.
 - Botão “Gerente” abrindo a tela Gerente.
 - Botão “Estoque” abrindo a tela Estoque.
-

8 Anne – Debug

8.1 Procedimentos de Debug

Anne explica o processo de debug:

- Teste de abertura de telas.
- Conferência das tabelas carregadas.
- Teste de navegação entre telas.
- Verificação das conexões Qt.

8.2 Problemas Corrigidos

- Botões sem função.
 - Caminhos incorretos do banco.
 - Telas que não abriam.
-

9 Anthony – Banco de Dados (SQL)

9.1 Tabelas do Sistema

Segundo o banco real, as tabelas relevantes são:

- clientes
- funcionarios
- dono_da_loja
- vendas
- vinis

9.2 Função de Cada Tabela

- vinis: lista de discos usados/novos.
- vendas: compras realizadas.
- clientes: usuários finais.
- funcionarios: gerentes e funcionários.
- dono_da_loja: controla o primeiro acesso.

9.3 Integração com o Sistema

Onde explica:

- Como o sistema lê e grava os dados;
 - Como são feitas as queries SQL;
 - Como as telas exibem os dados usando Qt.
-

10 Christian – Estudo dos Arquivos Header (.hpp)

10.1 Função dos Arquivos .hpp no Projeto

Christian apresenta a análise dos arquivos .hpp, responsáveis por declarar classes, funções, atributos e estruturas usadas nos .cpp. Eles garantem a separação entre interface e implementação e estruturam o funcionamento interno do sistema.

Os headers são essenciais para:

- organizar a arquitetura do código;
 - permitir reutilização entre módulos;
 - padronizar classes de cada tela;
 - conectar a interface gerada pelo Qt às funções do sistema.
-

10.2 Conteúdo Analisado nos Headers

Os .hpp contêm:

- declarações das classes de cada janela (Cadastro, Gerente, Principal, Estoque, Login);
 - ponteiros para objetos gerados pelo Qt Creator (Ui::NomeDaTela);
 - métodos públicos essenciais das telas;
 - atributos privados relacionados a dados internos;
 - slots e sinais usados para interações e eventos.
-

10.3 Relação dos Headers com o Fluxo do Sistema

É explica que:

- as telas são conectadas instanciando classes definidas nos headers;
 - os slots declarados ali são implementados nos .cpp;
 - headers determinam como telas se comunicam;
 - headers administrativos (como `gerente.hpp`) concentram regras de controle.
-

10.4 Importância do Estudo

A análise dos headers ajudou a equipe a:

- compreender toda a estrutura interna do projeto;
 - identificar relações entre classes e telas;
 - manter nomenclatura e organização consistentes;
 - integrar corretamente interface e banco de dados.
-

11 Conclusão

O vídeo final apresentará o sistema em funcionamento completo: telas, fluxo de navegação, login, permissões e integração com o banco de dados, demonstrando de forma clara toda a estrutura e lógica do vNewStore.