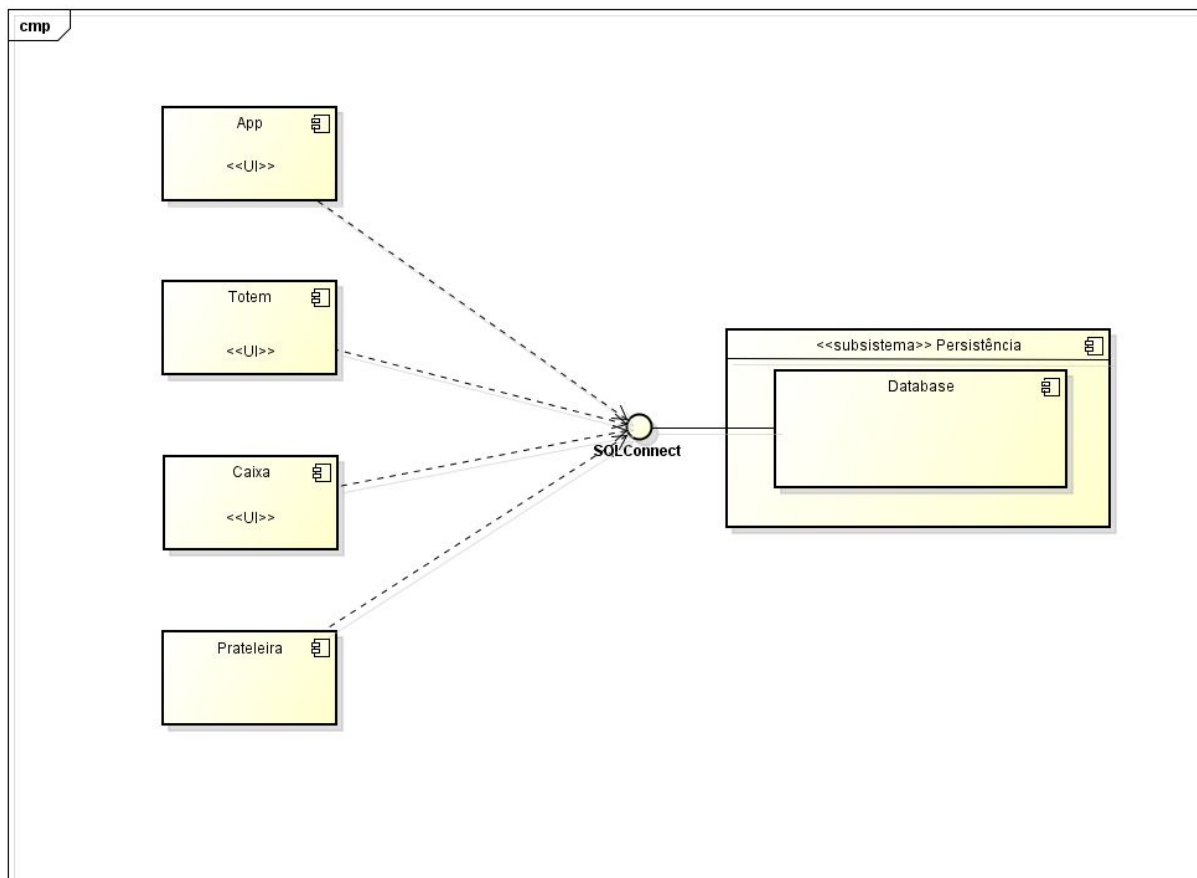


**Membros:** Eduardo Spitzer Fischer <[efischer@outlook.com.br](mailto:efischer@outlook.com.br)>  
Gabriel Lando <[gabriel@lando.net.br](mailto:gabriel@lando.net.br)>  
Rodrigo Paranhos Bastos <[ro.p.bastos@gmail.com](mailto:ro.p.bastos@gmail.com)>

**Produto de Software:** Smart Retail

**Product Owner:** G03

## 1. Modelo Arquitetural



Modelo Arquitetural

O diagrama foi alterado para que se indicasse o subsistema de persistência.

## 2. Projeto Detalhado

### 2.1 Projeto da Interface com o Usuário

O software consiste de 3 interfaces de usuário, chamadas de App, Caixa e Totem e uma interface chamada Prateleira. A interface App é utilizada por um gerente ou fornecedor para realizar ações definidas a eles, como cadastrar novos produtos, reabastecer estoque, cadastrar ou remover fornecedores e gerentes, entre outros. Já a interface Caixa é responsável por identificar quando um cliente entra ou sai da loja. Se o cliente já possuir cadastro, sua entrada é liberada, senão uma tela com um formulário pedindo os dados do cliente deve aparecer. Caso o cliente esteja de saída, essa interface deve concluir a compra e faturar no cartão de crédito do cliente, informando o valor total da compra e os itens adquiridos. A última interface de usuário deve servir como auxílio para o cliente enquanto está dentro da loja, nela o usuário teria acesso a algumas ofertas que combinam com seu perfil, além de listar e exibir o valor atual dos itens contidos em seu carrinho. A interface Prateleira simula o funcionamento das prateleiras da loja, é ela que é responsável por detectar se um cliente adiciona ou remove produtos em seu carrinho.

Na *figura 1* é possível ver como seria a tela de descanso de cada uma das interfaces. Enquanto o App exige que o usuário digite suas credenciais para fazer login, as outras interfaces estão aguardando um cliente passar em frente a câmera do seu sistema de reconhecimento facial para poder iniciar a sua função.

Já na *figura 2*, pode-se ver no App a tela de cadastro de ofertas, enquanto no caixa está sendo processada uma venda de um cliente, onde está sendo exibido os itens em seu carrinho e o valor total da compra. Na interface Totem, pode-se ver como seria a tela de auxílio ao cliente, mostrando seu carrinho e algumas ofertas recomendadas. Por fim, na tela da Prateleira, observa-se um produto sendo adicionado no carrinho de um cliente.



Figura 1: Interfaces de usuário aguardando algum comando.

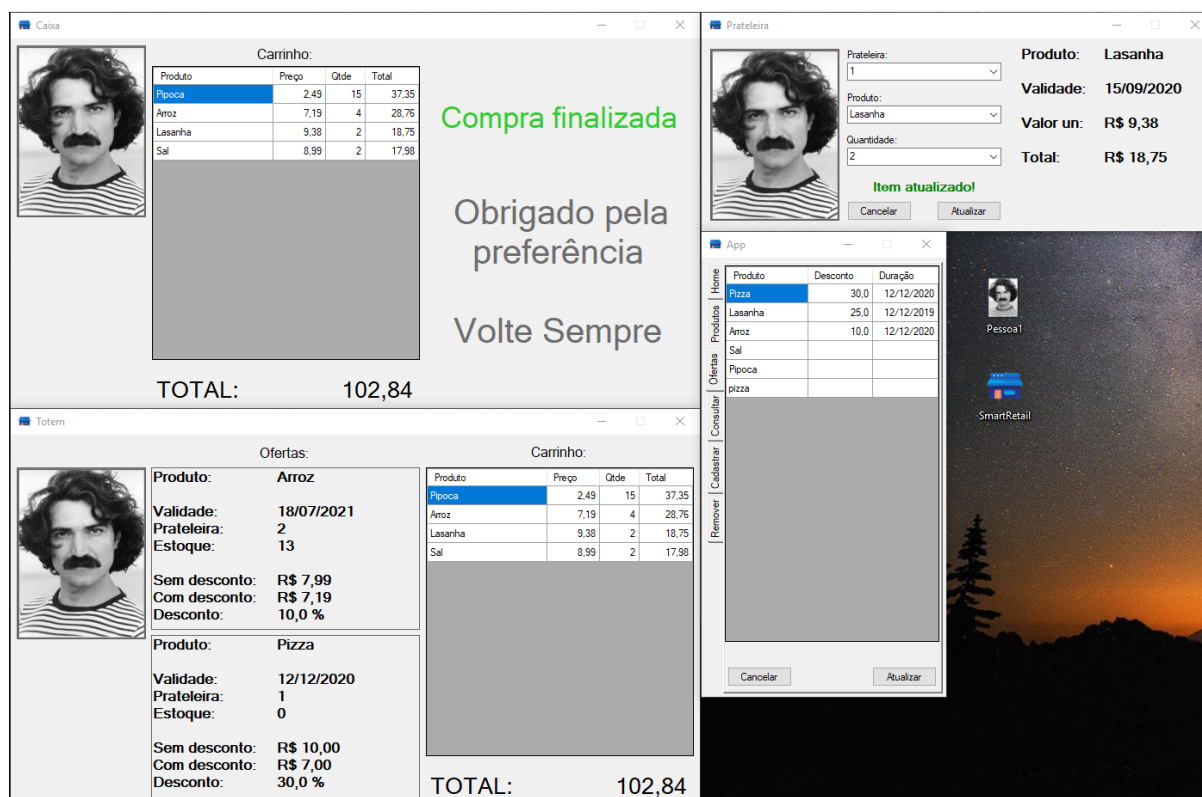
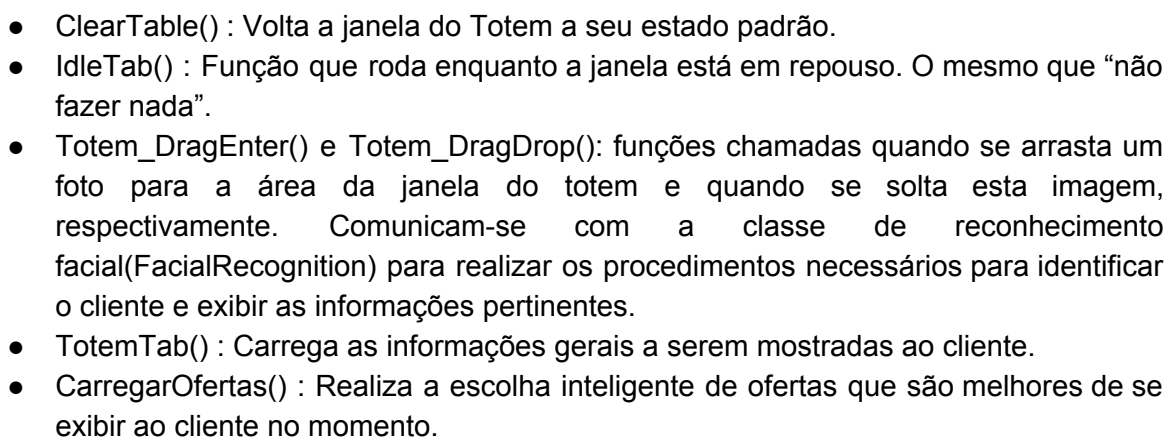


Figura 2: Interfaces de usuário sendo utilizadas.

## Totem



- `LoadImage()`: recebe um arquivo que deve ser a imagem de um cliente, verifica se o arquivo está em um formato válido (isto é, é de fato uma imagem) e, em caso positivo, carrega a imagem para uso posterior.
- `OnDragEnter()` e `OnDragDrop()`: funções chamadas pelas funções "DragEnter" (imagem arrastada para área de janela) e "DragDrop" (imagem solta em

janela) das outras classes que fornecem as informações necessárias para a validação de clientes no dispositivos.

- RetFilename(): Retorna o nome do arquivo de imagem reconhecido anteriormente.

## **App**

- CadastraOfertas(): Insere as ofertas configuradas por um gerente no banco de dados.
- CarregaOfertas(): Recupera as ofertas do banco de dados e as exibe no app.
- CarregaProdFornecedor(): Recupera do banco de dados todos os produtos fornecidos por determinado fornecedor.
- CarregaProdutos(): Carrega os dados de todos os produtos fornecidos no estabelecimento.
- VerificaSemEstoque(): Verifica a quantidade estocada dos produtos cadastrados e alerta os respectivos fornecedores quando essas quantidades ficam abaixo de um mínimo.
- ClearTable(): Limpa os dados em uma tela do app.
- Métodos "Btn\_Click": tratam eventos decorrentes da interação com a UI por meio de cliques em botões. Aqueles cujos nomes começam com "Ger" e "For" referem-se a ações realizadas em telas de gerentes e fornecedores, respectivamente.
- Métodos "Box\_Click": Tratam da interação com caixas de texto na tela de login do app.

## **InfoBasica**

- ValidaEmail(): verifica se o email é válido através de expressões regulares.

## **Prateleira**

- Métodos "Btn\_Click": tratam eventos decorrentes da interação com a UI por meio de cliques em botões.
- Métodos "Box\_Click": Tratam da interação com caixas de texto na janela da prateleira.
- IdleTab(): Função que roda enquanto a janela está em repouso. O mesmo que "não fazer nada".
- Prateleira\_DragEnter() e Prateleira\_DragDrop(): funções chamadas quando se arrasta um foto para a área da janela da prateleira e quando se solta esta imagem, respectivamente. Comunicam-se com a classe de reconhecimento facial(FacialRecognition) para realizar os procedimentos necessários para identificar o cliente e exibir as informações pertinentes.
- PrateleiraTab() : Carrega as informações gerais a serem mostradas ao cliente identificado.

## **Caixa**

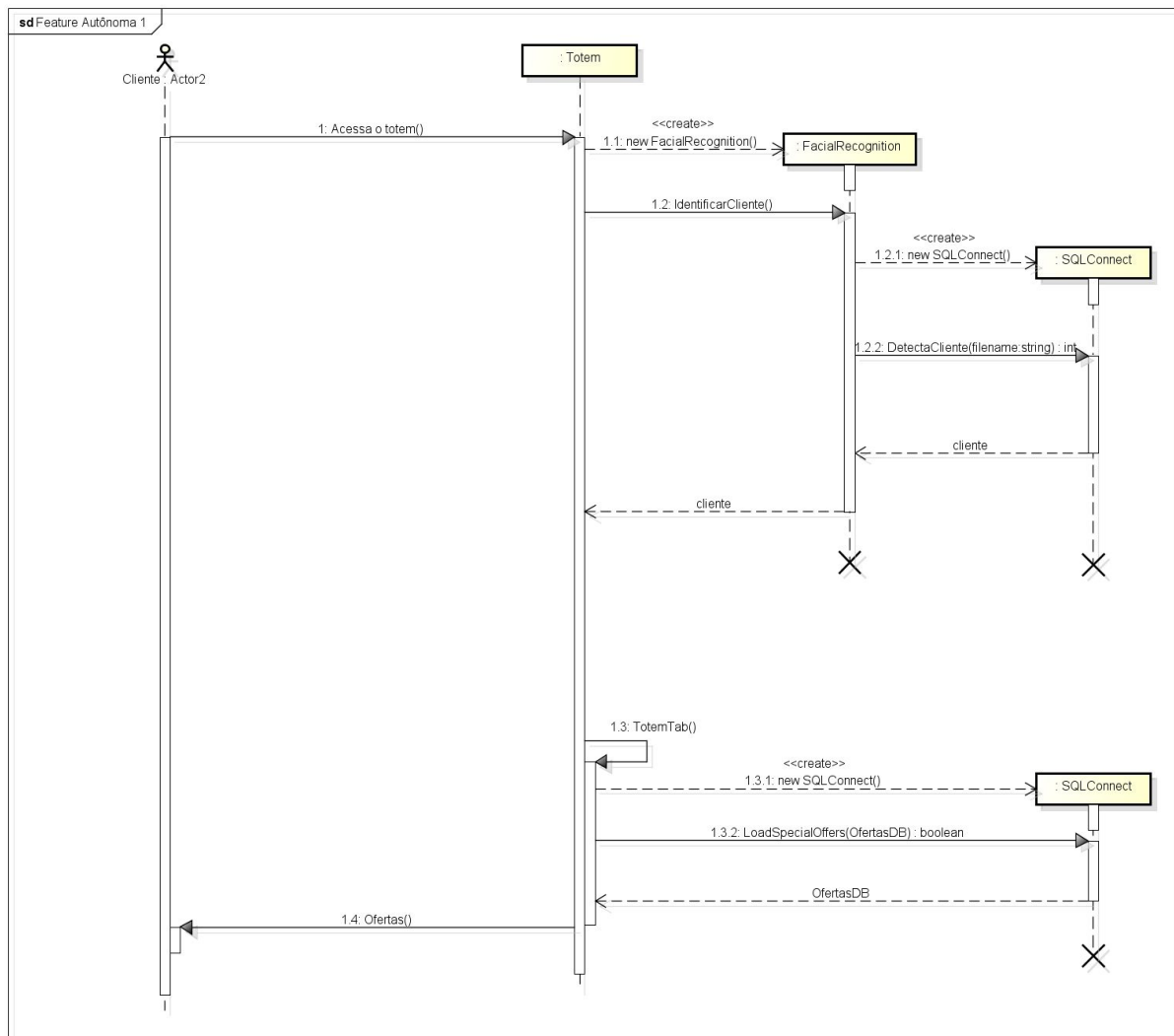
- Métodos "Btn\_Click": tratam eventos decorrentes da interação com a UI por meio de cliques em botões.
- Métodos "Box\_Click": Tratam da interação com caixas de texto na janela do caixa.
- IdleTab(): Função que roda enquanto a janela está em repouso. O mesmo que "não fazer nada".
- ClearTab(): Limpa os dados de tabelas do caixa.

- Caixa\_DragEnter() e Caixa\_DragDrop(): funções chamadas quando se arrasta um foto para a área da janela do caixa e quando se solta esta imagem, respectivamente. Comunicam-se com a classe de reconhecimento facial(FacialRecognition) para realizar os procedimentos necessários para identificar o cliente e exibir as informações pertinentes.
- SaidaTab(): Exibe as informações que um cliente vê quando deixa a loja: sua compra realizada e mensagem de despedida.

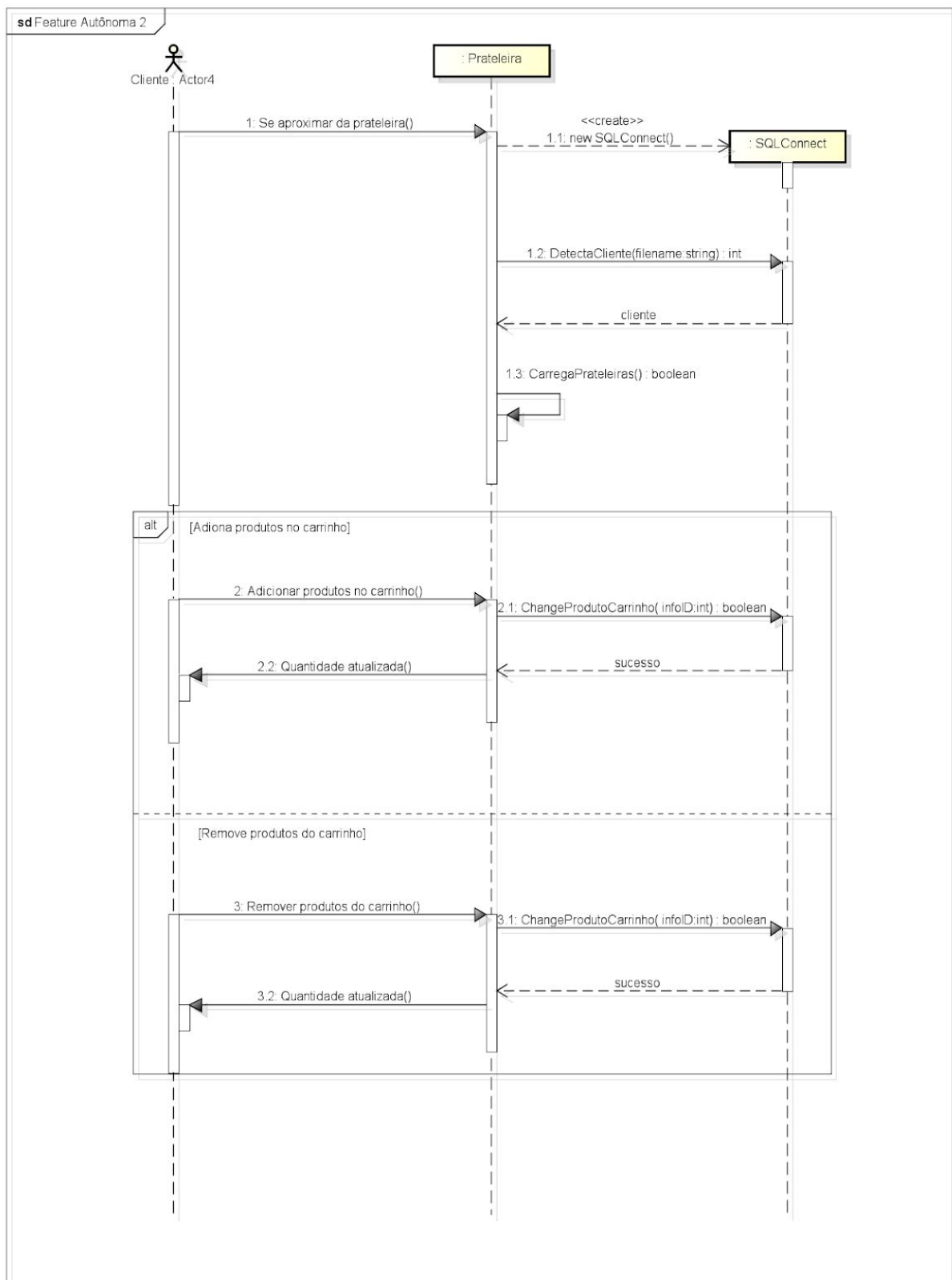
### **SQLConnect**

- Inicializar(): inicializa a conexão com o banco de dados.
- AbrirConexão(): abre um canal de comunicação com o banco de dados.
- FecharConexão(): fecha um canal de comunicação com o banco de dados para que outro possa ser aberto, se for o caso.
- ChangeProdutoCarrinho(): muda os produtos associados a um carrinho no banco de dados.
- DetectaCliente(): verifica se cliente fornecido está cadastrado no banco de dados.
- InserirCliente(): insere cliente no banco de dados.
- InserirFornecedor(): insere fornecedor no banco de dados.
- InserirGerente(): insere gerente no banco de dados.
- InsertProductsFornecedor(): insere produtos no banco de dados, associados a um fornecedor.
- LoadAllOffers(): Carrega todas as ofertas cadastradas no banco de dados.
- LoadAllProducts(): carrega todos os produtos cadastrados no banco de dados.
- LoadSomeProducts(): carrega os dados de produtos identificados por IDs que a função recebe. Serve para ler os dados dos produtos mostrados em ofertas.
- LoadSpecialOffers(): carrega ofertas associadas a um cliente.
- RemoveProductsFornecedor(): desassocia um determinado produto de um fornecedor.
- RemoverUsuário(): Remove cadastro de usuário do banco de dados.
- ReturnProductsFornecedor(): retorna todos os produtos vinculados a um fornecedor.
- ReturnProductsSacola(): Retorna todos os produtos de um carrinho associado a um cliente e o valor total do carrinho.
- SQLConnect(): construtor da classe.
- UpdateProductsFornecedor(): atualiza os produtos associados a um fornecedor.
- ValidaCliente(): Retorna o estado de um cliente codificado em um inteiro. Valor retornado: -1, cliente novo, que deve ser cadastrado; 0, cliente já cadastrado entrando na loja; id do cliente, no caso de cliente saindo do estabelecimento.
- ValidarLogin(): informa se as credenciais inseridas são válidas e qual o tipo de usuário. Valor 1 corresponde a fornecedor e valor 2 a gerente.
- GenerateNF(): gera nota fiscal da venda.

## 2.3 Diagramas de Interação

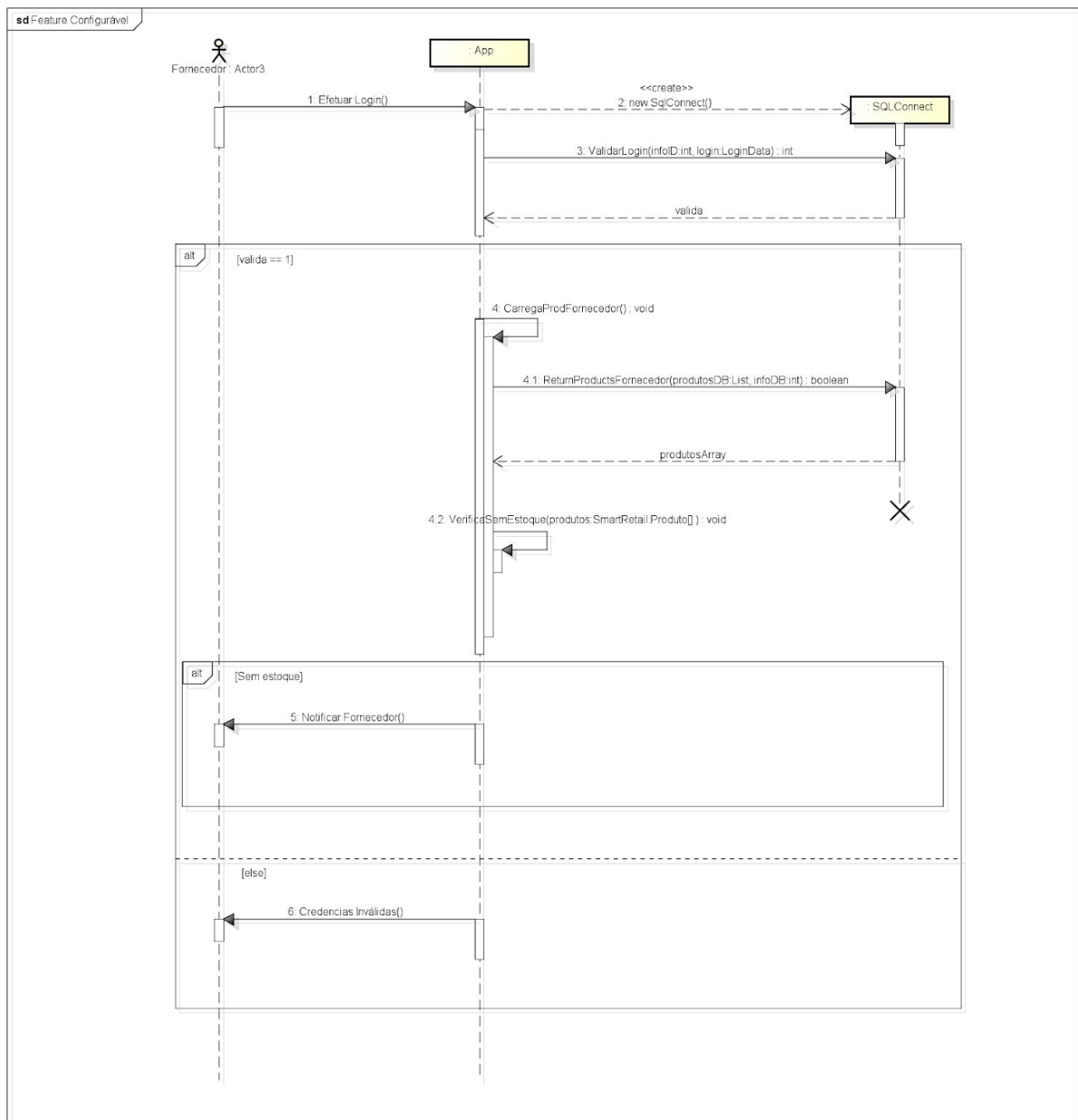


O módulo de reconhecimento facial(classe FacialRecognition) recebe as informações de imagem coletadas pelo Totem, gerando informações necessárias para buscar os dados do cliente no banco de dados através da classe SQLConnect. Uma vez com os dados do cliente, o totem acessa diretamente o banco de dados e escolhe as melhores ofertas para aquele cliente.



A prateleira opera sobre o banco de dados com base em suas interações com o cliente, atualizando as quantidades em seu carrinho.





O app valida as credenciais fornecidas junto ao banco de dados. Em caso de login de fornecedor, recupera as quantidades em estoque dos produtos associados ao mesmo. Em caso de produtos em quantidade abaixo da mínima o fornecedor é avisado na tela de home do app.