# **UNIVERSIDADE CATÓLICA SC**

# **ENGENHARIA DE SOFTWARE**

# **Implementação de um sistema financeiro com verificação facial**

# **Vinicius Gabriel Ochner**

Joinville-SC

04/04/2024

Resumo

O trabalho apresenta o desenvolvimento de um sistema financeiro em um banco virtual que emprega o reconhecimento facial para autenticação de usuários. O projeto visa aprimorar a segurança e a conveniência das transações online, substituindo métodos de autenticação tradicionais como senhas e códigos por tecnologia biométrica. Utilizando o Amazon Rekognition, o sistema oferece um método de autenticação rápido e seguro, capaz de garantir que apenas usuários autorizados acessem suas contas e executem operações financeiras, como transferências e consultas de saldo.

Além do foco em segurança reforçada e facilidade de uso, o sistema proposto é desenvolvido com uma arquitetura moderna e escalável, incorporando tecnologias como GoLang e React, e hospedado na AWS. Utiliza-se o ECS para gerenciamento de containers e o RDS para gestão de bases de dados. A integração com o Amazon Rekognition permite o gerenciamento eficiente de dados biométricos e fortalece a segurança contra fraudes, utilizando análises de imagem baseadas em aprendizado de máquina para verificar a identidade dos usuários. Este projeto atende às necessidades atuais do mercado financeiro digital, oferecendo uma solução robusta e moderna para o ambiente virtual.

1. Introdução

Com o avanço da tecnologia e a crescente popularidade das transações financeiras online, a segurança e a conveniência tornaram-se prioridades fundamentais para consumidores e instituições financeiras. Nesse contexto, surge a necessidade de soluções inovadoras que combinem proteção robusta contra fraudes e facilidade de uso. Uma dessas soluções é um sistema de pagamento em um banco virtual utilizando reconhecimento facial para validação de transações e login.

O reconhecimento facial, uma das formas mais seguras e avançadas de autenticação biométrica, oferece uma maneira eficaz de garantir que apenas usuários autorizados possam acessar suas contas e realizar transações financeiras.

Ao integrar essa tecnologia com os serviços de um banco virtual, é possível proporcionar uma experiência de usuário que alia segurança e praticidade, eliminando a necessidade de senhas e outros métodos de autenticação tradicionais.

Este estudo explora a implementação de um sistema de pagamento em um banco virtual que utiliza o reconhecimento facial não apenas para login, mas também para validar transações financeiras. Com isso, busca-se aumentar a segurança contra fraudes e invasões, ao mesmo tempo em que se oferece uma interface intuitiva e eficiente para os usuários. A introdução dessa tecnologia visa atender às demandas do mercado financeiro atual, oferecendo uma solução moderna e eficaz para as transações financeiras online.

2. Descrição do Projeto

**Descrição Detalhada**

O projeto destaca-se pela concepção de um banco virtual que se distingue não somente pela aprimorada proteção do usuário, mas também pela sua enfática abordagem nas transações internas.

A criação de conta neste ambiente inovador e seguro é fundamental, conferindo aos clientes uma experiência ágil e confiável desde o momento inicial de interação. Uma característica singular reside na implementação da autenticação por reconhecimento facial, proporcionando uma camada adicional de segurança contra atividades fraudulentas e transações não autorizadas.

É de suma relevância salientar que esta sofisticada forma de autenticação facial será incorporada em todas as transações realizadas, garantindo, assim, uma segurança aprimorada em cada interação financeira.

**Funcionalidades Básicas**

O banco virtual oferecerá diversas funcionalidades, como:

* Abertura de contas
* Transferências internas
* Consulta de saldos e extratos
* Autenticação por reconhecimento facial

**Autenticação por Reconhecimento Facial**

A autenticação por reconhecimento facial se destaca como a principal característica do banco virtual, proporcionando:

* **Processo de autenticação rápido e seguro:** A captura da imagem do rosto do usuário e a comparação com a imagem cadastrada no banco de dados garantem um processo de autenticação rápido e seguro, minimizando o risco de fraudes e acessos indevidos.
* **Maior dificuldade de falsificação da identidade do usuário:** A tecnologia de reconhecimento facial dificulta a falsificação da identidade do usuário, reforçando a segurança das transações online.
* **Experiência do usuário aprimorada:** A autenticação por reconhecimento facial torna o acesso à conta mais rápido e conveniente, proporcionando uma experiência do usuário aprimorada.

**Segurança Complementar**

Para reforçar a segurança, o banco virtual contará com:

* **Criptografia de dados:** A criptografia robusta protege os dados dos usuários durante transações e armazenamento, garantindo a confidencialidade das informações.

**Aspectos Técnicos e Implementação**

O desenvolvimento do banco virtual seguirá rigorosos critérios:

* **Arquitetura tecnológica moderna e escalável:** A arquitetura tecnológica do banco virtual será projetada com tecnologias modernas e escaláveis para suportar um grande volume de transações e usuários.
* **Testes e validação abrangentes:** Testes abrangentes serão realizados em todas as etapas do desenvolvimento para garantir que o banco virtual atenda aos requisitos dos usuários e às expectativas de segurança

3. Especificação Técnica

3.1. Requisitos de Software

**Requisitos Funcionais (RF)**

**Gerenciamento de Contas e Transações:**

* Permitir a abertura de contas.
* Realizar transferências internas.
* Permitir a consulta de saldos, extratos e histórico de transações.

**Segurança das Transações:**

* Implementar autenticação facial para validar as transações, garantindo maior segurança e praticidade.
* Adotar criptografia robusta para proteger os dados financeiros dos usuários durante transações e armazenamento.

**Requisitos Não-Funcionais (RNF)**

**Segurança e Confiabilidade:**

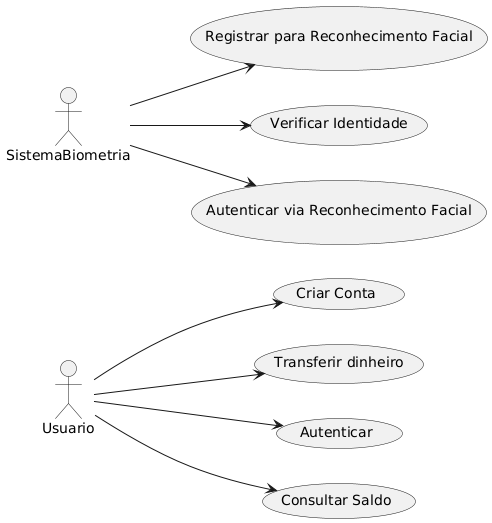
* Garantir alta segurança e criptografia dos dados financeiros dos usuários, utilizando protocolos e tecnologias confiáveis.
* Manter alta disponibilidade do sistema, minimizando interrupções e indisponibilidades.
* Implementar testes rigorosos para garantir a confiabilidade e robustez do sistema.

**Desempenho e Escalabilidade:**

* Garantir o bom desempenho do sistema, mesmo com um grande volume de usuários e transações.
* Arquitetar o sistema de forma escalável para suportar o crescimento da base de usuários e da demanda por serviços.
* Monitorar o desempenho do sistema em tempo real para identificar e solucionar gargalos.
* Implementar mecanismos de automação para otimizar o processamento de transações.

**Usabilidade e Acessibilidade:**

* Fornecer uma interface amigável e intuitiva, fácil de usar para qualquer pessoa, independentemente de seus conhecimentos técnicos.]



### 

### 

3.2. Considerações de Design

#### **Discussão sobre as escolhas de design**

Ao projetar o sistema financeiro com autenticação facial, foram consideradas diversas alternativas para assegurar a melhor solução possível. Após uma análise minuciosa, optou-se pela utilização de uma arquitetura monolitico.

Esta escolha foi motivada pela simplicidade e facilidade de gerenciamento que a arquitetura monolítica oferece, centralizando todas as funcionalidades em um único sistema coeso. Essa abordagem facilita o desenvolvimento inicial, a integração e o teste, reduzindo a complexidade de gerenciamento de múltiplos serviços. Embora os microserviços tenham sido avaliados, concluiu-se que a robustez e o menor custo de manutenção da arquitetura monolítica são mais adequados para o contexto atual do projeto, garantindo alta disponibilidade e segurança em um ambiente de transações financeiras.

**Visão Inicial da Arquitetura**

A arquitetura do sistema financeiro é composta por um monolito centralizado, onde todos os componentes do sistema estão integrados em uma única aplicação. Essa abordagem simplifica a comunicação interna e elimina a necessidade de uma API Gateway, mantendo os processos de forma coesa e eficiente.

Além disso, a arquitetura foi projetada com uma clara separação entre o front-end e o back-end dentro do monolito. Essa separação lógica permite maior flexibilidade no desenvolvimento de interfaces de usuário, facilitando a introdução de novas funcionalidades e melhorias na interface sem interferir diretamente na lógica central do sistema.

**Padrões de Arquitetura**

Para o back-end, é utilizado o padrão monolítico, que centraliza todas as funcionalidades do sistema em uma única aplicação integrada. Essa abordagem simplifica o gerenciamento e reduz a complexidade associada à comunicação entre serviços, tornando o sistema mais fácil de desenvolver e manter. Embora não tenha a resiliência de isolamento de falhas dos microserviços, a arquitetura monolítica é robusta e eficiente para o contexto atual.

Para o front-end, adota-se o modelo Model-View-Controller (MVC), que organiza a aplicação em camadas bem definidas. O MVC facilita a manutenção e atualização do sistema, garantindo uma estrutura de código clara e gerenciável, essencial para evoluir e adaptar um sistema integrado de maneira eficiente.

#### 

#### 

#### 

#### 

#### 

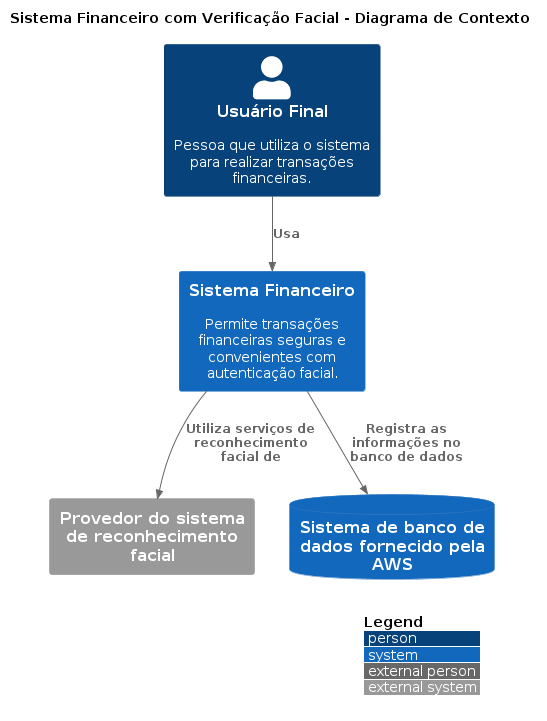
#### 

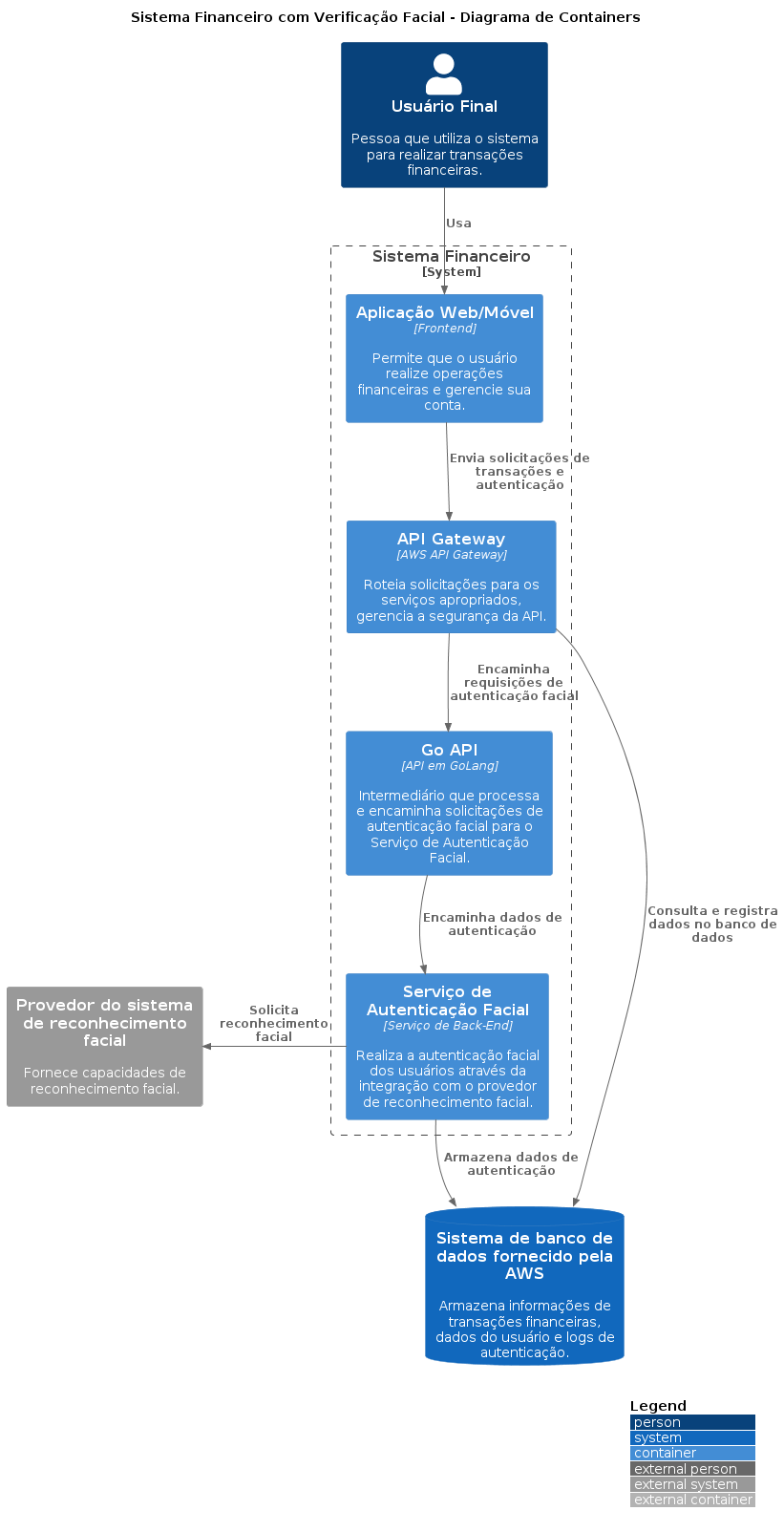
#### 

#### 

#### 

#### **Modelos C4**





3.3. Stack Tecnológica

#### **Linguagens de Programação**

Optamos pelo GoLang para os serviços backend devido à sua alta performance, excelente suporte a concorrência e robustez em operações de rede, características essenciais para sistemas financeiros de alta demanda. Sua simplicidade e eficiência tornam-no ideal para garantir estabilidade e segurança.

#### **Frameworks e Bibliotecas**

Para o desenvolvimento da API em Go, utilizaremos o framework **Gin**, conhecido por seu roteamento ágil e suporte eficiente a middlewares, proporcionando alta performance e facilidade de uso. No frontend, o **React** será mantido para criar interfaces de usuário interativas, escaláveis e altamente responsivas, garantindo uma experiência superior ao usuário final.

#### **Ferramentas de Desenvolvimento e Gestão de Projeto**

Adotaremos o **Git** para controle de versão, permitindo rastreamento e colaboração eficiente no código, e o **JIRA** para gestão de projetos, com foco em organização e produtividade. Além disso, o **Docker** será utilizado para a containerização dos serviços, garantindo portabilidade, uniformidade e facilidade na implantação e escalabilidade na AWS.

#### **Infraestrutura Cloud**

A implementação será realizada na AWS, utilizando serviços como ECS para gerenciamento de contêineres e hospedagem de servidores virtuais, RDS para bases de dados relacionais com alta disponibilidade e ECR para armazenamento e versionamento de imagens Docker. Adicionalmente, o Amazon Rekognition será empregado para análise e reconhecimento facial, garantindo a integração com o sistema de segurança.

A escolha pela AWS baseia-se em sua ampla gama de serviços integrados, segurança robusta e escalabilidade eficiente.

3.4. Considerações de Segurança

#### **Análise de Possíveis Questões de Segurança**

As principais preocupações incluem a interceptação de dados sensíveis e a falsificação de identidade. Para mitigar esses riscos, implementamos criptografia de ponta a ponta para proteger os dados em trânsito, garantindo sua confidencialidade. Além disso, utilizaremos ferramentas de IA, como o Amazon Rekognition, para reconhecimento facial, reforçando a autenticação e minimizando a probabilidade de fraudes e acessos não autorizados.

4. Próximos Passos

1. **Autenticação multifator (MFA)**: Implementar um segundo fator de autenticação, como códigos enviados por aplicativo ou biometria, para aumentar a segurança no acesso às contas.
2. **Detecção e bloqueio de fraudes**: Criar um sistema de monitoramento que identifique e impeça transações suspeitas em tempo real, protegendo os usuários contra atividades maliciosas.
3. **Notificações de segurança personalizadas**: Permitir que os clientes configurem alertas para eventos como transferências acima de valores específicos ou tentativas de login não reconhecidas.
4. **Histórico detalhado de atividades**: Disponibilizar um registro completo e transparente das movimentações na conta, incluindo horários e dispositivos utilizados.
5. **Bloqueio emergencial da conta**: Implementar um recurso que permita ao cliente bloquear sua conta instantaneamente em caso de suspeita de comprometimento.
6. **Gestão de dispositivos autorizados**: Fornecer uma funcionalidade para visualizar e gerenciar dispositivos que têm acesso à conta, permitindo a remoção de dispositivos desconhecidos.

5. Referências

<https://aws.amazon.com/pt/rekognition/>  
<https://www.docker.com/>  
<https://aws.amazon.com/pt/ecs/>  
<https://aws.amazon.com/pt/s3/>  
<https://go.dev/>  
<https://dev.to/nikl/how-to-build-a-containerized-microservice-in-golang-a-step-by-step-guide-with-example-use-case-5ea8>  
<https://gin-gonic.com/>  
<https://kubernetes.io/pt-br/>