

**INF**  
INSTITUTO DE  
INFORMÁTICA



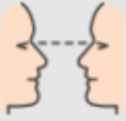
**UFG**  
UNIVERSIDADE  
FEDERAL DE GOIÁS

**UNIVERSIDADE FEDERAL DE GOIÁS  
CURSO ENGENHARIA DE SOFTWARE  
DOMÍNIOS DE SOFTWARE**

**Adriel Lenner Vinhal Mori  
Igor Moreira Pádua  
Marcos Vinícius de Moraes  
Paulo Roberto Vieira**

**Projeto IV/UFG Identificação: sistema de identificação visual de candidatos  
TRABALHO FINAL DE DOMÍNIO DE SOFTWARE**

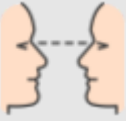
**Goiânia  
2023**



## Documento geral do projeto

### Sumário

<b>1. INTRODUÇÃO</b>	<b>3</b>
<b>2. MAPA DE ATORES</b>	<b>3</b>
<b>3. HISTÓRIA DE USUÁRIO</b>	<b>3</b>
<b>4. REQUISITOS FUNCIONAIS</b>	<b>5</b>
<b>5. REQUISITOS NÃO FUNCIONAIS</b>	<b>6</b>
<b>6. RESTRIÇÕES</b>	<b>6</b>
<b>7. ARQUITETURA DE 3 CAMADAS</b>	<b>7</b>



## Documento geral do projeto

### 1. INTRODUÇÃO

A proposta deste documento é juntar todas as informações presentes no projeto com uma abordagem geral do que é o software, os seus principais desafios para a construção e o porquê da arquitetura escolhida.

### 2. PROBLEMA

O problema no qual o Projeto IV/UFG Identificação: sistema de identificação visual de candidatos quer resolver é a criação de uma IA que consiga identificar um candidato por uma foto já cadastrada no sistema.

### 3. DESAFIOS

1. Identificar um candidato por duas imagens diferentes;
2. Identificar um candidato em tempo real com imagem já registrado no sistema;
3. Ter uma acurácia de no mínimo de 80%.

### 4. MAPA DE ATORES

Listas de stakeholders principais, diretos e indiretos:

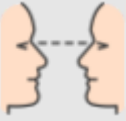
- Principais
  - Operador do sistema.
- Diretos
  - Pessoa que está registrada no sistema;
  - Profissional que mantém o sistema.
- Indiretos
  - Empresa que utiliza o sistema.

Lista de critérios para priorização de stakeholders:

1. Praticidade;
2. Usabilidade;
3. Redução de erros.

### 5. HISTÓRIA DE USUÁRIO

Os requisitos de stakeholders foram coletados no formato de histórias de usuários, com a apresentação do responsável pela escrita do relato dos stakeholders, prioridade, tipo de stakeholder e cenário de usabilidade.



## Documento geral do projeto

### HU01 Confrontar duas imagens

**Responsável:** Igor

**Prioridade:** alta

**Como** operador do sistema,

**Quero** que o sistema confronte duas imagens

**Para** que possa ser validado se existe ou não no banco de dados.

**Cenário 1:** Identificar uma imagem

**Dado** que existem imagens cadastradas no sistema,

**Quando** inserimos uma nova imagem,

**Então** o sistema identifica se a nova imagem corresponde com alguma existente no banco de dados do sistema e retorna mensagem validando ou não.

### HU02 Apresentar Dados

**Responsável:** Paulo

**Prioridade:** alta

**Como** operador do sistema,

**Quero** que o sistema apresente os dados de uma pessoa

**Para** que possa ser validado as informações

**Cenário 1:** Validar dados de uma pessoa cadastrada

**Dado** que uma imagem foi inserida.

**Quando** houver o reconhecimento com alguma existente no sistema

**Então** o sistema deve apresentar os dados da pessoa que teve imagem reconhecida e validada pelo sistema.

### HU03 Status do Candidato

**Responsável:** Adriel Lenner Vinhal Mori

**Prioridade:** alta

**Como** operador do sistema,

**Quero** adicionar um status referente a aplicação e realização da prova ao candidato que está realizando-a.

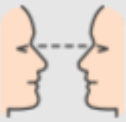
**Para** sinalizar candidatos que finalizaram a prova, faltantes e que tiveram a realização da prova interrompida por conta de força maior.

**Cenário 1:** Adicionar status ao candidato.

**Dado** que um candidato tenha sido previamente identificado,

**Quando** a identificação do candidato for validada

**Então** poderá ser adicionado um status referente a participação do candidato na realização da prova - iniciado, indeferido, interrompido, concluído. Assim, poderá



## Documento geral do projeto

ser adicionada a súmula da aplicação geral da prova na sessão em que está sendo aplicada.

### HU04 Conferência de Sala

**Responsável:** Marcos Vinícius de Moraes

**Prioridade:** alta

**Como** operador do sistema,

**Quero** poder adicionar observações sobre o candidato identificado

**Para** ter um maior controle de situações atípicas que venham a acontecer.

**Cenário 1:** adicionar observações ao candidato.

**Dado** que um candidato tenha sido previamente identificado.

**Quando** a identificação do candidato for validada

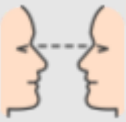
**Então** podem ser feitas anotações acerca do candidato e de qualquer irregularidade ou imprevisibilidade decorrida.

## 6. REQUISITOS FUNCIONAIS

Para estabelecer a prioridade dos requisitos foram adotadas as denominações “essencial”, “importante” e “desejável”. A prioridade dos requisitos é utilizada no gerenciamento do escopo das etapas do projeto e na definição das prioridades durante o desenvolvimento do sistema.

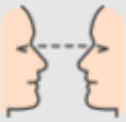
- **Alta (1):** Requisito essencial cujo fracasso em sua implementação significa que o sistema não irá atender as necessidades do cliente. Imprescindível que seja atendido pelo sistema, condição fundamental para o sucesso do projeto.
- **Média (2):** Requisitos importantes para a eficácia ou eficiência do sistema. Sua não implementação afeta a satisfação do usuário e/ou o valor agregado do produto e o não atendimento não determina o fracasso do projeto.
- **Baixo(3):** Requisitos úteis, porém menos críticos, sendo usados menos frequentemente. Não possui muito significado para a satisfação do usuário e pode deixar de ser atendida.

Prioridade	Identificador	Nome e Descrição do Requisito	Interfaces e Dependências
------------	---------------	-------------------------------	---------------------------



## Documento geral do projeto

[1 / 2 / 3]	[TIPSEQ: de acordo com Plano de Gerenciamento de Requisitos.]	[Nome do requisito]: [Detalhes levantados sobre o requisito. Se possível definir o representante do cliente que forneceu a informação.]	Interfaces: [Identificador da interface]  Dependências: [Identificador do requisito]
1	RF1	Captura de imagem: o sistema deve ser capaz de capturar imagens de faces e processá-las para reconhecimento.	
1	RF2	Reconhecimento facial: o sistema deve ser capaz de comparar imagens capturadas com imagens armazenadas em seu banco de dados e identificar a pessoa correspondente.	
1	RF3	Uma vez que a pessoa tenha sido identificada, o sistema possibilita informar o status do candidato - presente, indeferido, concluído.	
1	RF4	Uma vez que a pessoa tenha sido identificada, o sistema permite que seja adicionado uma observação acerca do status da pessoa.	

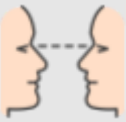


## Documento geral do projeto

1	RF5	Atualização de informações de usuários: o sistema deve permitir a atualização das informações de usuários, incluindo fotos de rosto, nomes e informações de contato.	
1	RF6	Integração com outros sistemas: o sistema deve ser capaz de se integrar com outros sistemas, como aplicativos mobile.	

## 7. REQUISITOS NÃO FUNCIONAIS

Prioridade	Identificador	Nome e Descrição do Requisito	Interfaces e Dependências
[1 / 2 / 3]	[TIPSEQ:  de acordo com Plano de Gerenciamento de Requisitos.]	[Nome do requisito]:  [Detalhes levantados sobre o requisito. Se possível definir o representante do cliente que forneceu a informação.]	Interfaces: [Identificador da interface]  Dependências: [Identificador do requisito]
1	RNF1	Confiabilidade: o sistema deve ser capaz de reconhecer com precisão faces em diferentes condições de iluminação, ângulos e resoluções.	
	RNF2	Segurança: as informações de reconhecimento facial devem ser protegidas contra acesso não	



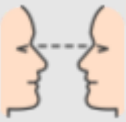
## Documento geral do projeto

		autorizado e violações de segurança.	
1	<b>RNF3</b>	O sistema deverá se comunicar com as dependências do SQL Server.	
1	<b>RNF4</b>	Privacidade: o sistema deve cumprir regulamentos de privacidade e proteger as informações pessoais dos usuários.	
1	<b>RNF5</b>	Escalabilidade: o sistema deve ser capaz de lidar com grandes quantidades de dados e usuários, mantendo a precisão do reconhecimento facial.	
1	<b>RNF6</b>	Usabilidade: o sistema deve ser fácil de usar e compreender para os usuários, com instruções claras e intuitivas.	
1	<b>RNF7</b>	Disponibilidade: o sistema deve estar disponível ao longo do tempo, sem interrupções significativas.	

## 8. RESTRIÇÕES

1. O sistema deve ter um banco de dados contendo as informações dos usuários;
2. O sistema deve ter acesso ao Wi-Fi para sincronizar os dados;
3. O sistema deve conseguir ter acesso a uma câmera para conseguir identificar as pessoas.





### 9. ARQUITETURA

O sistema de identificação visual de candidatos é uma Rede Neural Convolutiva (CNN), já que ela trabalha com imagens e é preciso que se atinja um alto desempenho em tarefas visuais complexas.

Uma Arquitetura de Software para Inteligência Artificial pode incluir vários componentes, tais como:

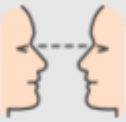
Para a decisão da nossa arquitetura precisamos pensar nas seguintes problemáticas:

- Tipo de modelo: Escolha entre modelos de aprendizado supervisionado, não supervisionado, semi-supervisionado, entre outros.
- Tipo de rede neural: Escolha entre redes neurais feedforward, recorrentes, convolucionais, entre outros.
- Algoritmo de otimização: Escolha entre algoritmos de otimização como Gradient Descent, Adam, entre outros.
- Tamanho do conjunto de treinamento: Escolha do tamanho do conjunto de treinamento e do número de épocas de treinamento.
- Tipo de dados de entrada: Escolha do tipo de dados de entrada, como imagens, áudio, texto, entre outros.
- Tipo de saída: Escolha do tipo de saída, como previsões, classificação, reconhecimento de fala, entre outros.
- Arquitetura de armazenamento: Armazenamento como bancos de dados relacionais ou não relacionais.
- Arquitetura de computação: Como computação em nuvem ou computação local.
- Mecanismo de explicação: Tais como LIME, SHAP, entre outros.

Essas decisões afetam o Desempenho, a Escalabilidade, e a Facilidade de manutenção do sistema AI.

Além disso a arquitetura de uma IA pode incluir vários componentes, tais como:

1. Camada de entrada: Responsável por coletar e processar os dados de entrada para o sistema, como imagens, áudio, ou texto.
2. Camada de modelo: Responsável por armazenar e aplicar o modelo de aprendizado de máquina treinado. Ele pode incluir redes neurais, algoritmos de aprendizado supervisionado, entre outros.
3. Camada de saída: Responsável por produzir a saída do sistema, como previsões, reconhecimento de fala, ou classificação de imagens.
4. Camada de gerenciamento: Responsável por gerenciar as operações do sistema, como monitoramento de desempenho, gerenciamento de recursos, e manutenção.
5. Camada de usuário: Responsável por prover a interface do usuário com o sistema, como uma interface web ou aplicativo móvel.
6. Camada de armazenamento: Responsável por armazenar os dados de entrada, saída, e modelos do sistema.



## Documento geral do projeto

Vale ressaltar que essas camadas são interligadas e trocam informações entre si.

As decisões arquiteturais não para nisso, é preciso também que cada componente do sistema tenha apenas uma responsabilidade e não se sobreponha a outras, tais componentes são:

1. Escalabilidade: o sistema deve ser projetado para escalar facilmente à medida que o volume de tráfego aumenta.
2. Manutenção: o sistema deve ser fácil de manter e atualizar, com componentes que podem ser substituídos ou atualizados sem afetar o funcionamento geral do sistema.
3. Testabilidade: o sistema deve ser projetado de tal forma que os componentes possam ser testados de forma isolada para garantir a confiabilidade do sistema.
4. Reutilização: os componentes do sistema devem ser projetados de forma que possam ser reutilizados em outros projetos.

Existem muitas arquiteturas de software diferentes que se adaptam para diferentes tipos de sistemas e aplicações, incluindo arquitetura monolítica, arquitetura de microserviços, arquitetura de eventos, arquitetura baseada em componentes e muito mais.

Para atender a todas as decisões arquiteturais e dos requisitos que foram propostas nesse documento, o grupo decidiu que a arquitetura baseada em componentes é a que melhor atende às necessidades do cliente.

Essa escolha se deve por quanta que o sistema é dividido em componentes reutilizáveis que podem ser combinados para a criar soluções de IA mais complexas como o nosso sistema.

### 9.1 Arquitetura de 3 camadas

Para ser mais fácil a visualização da arquitetura do sistema, o grupo criou uma arquitetura em 3 camadas.

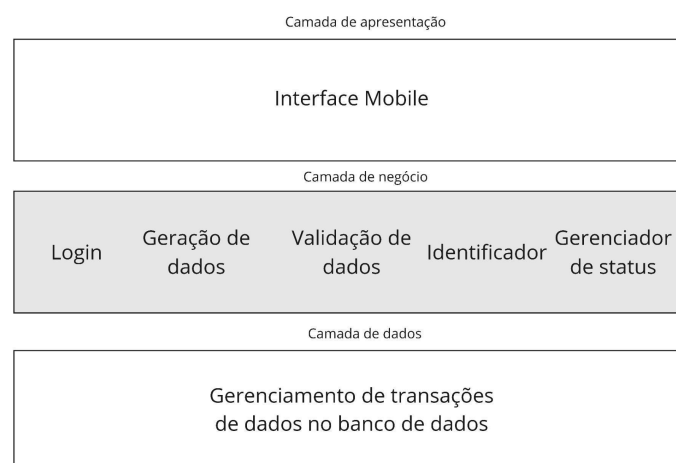


Imagem 1: Representação da arquitetura do projeto em 3 camadas