

**MINISTÉRIO DA EDUCAÇÃO**  
**SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA**  
**INSTITUTO FEDERAL GOIANO – CAMPUS IPORÁ**  
**TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**RELATÓRIO DE ESTÁGIO**

**DOUGLAS RAMOS MARTINS**

**IPORÁ, GO**

**2024**

**MINISTÉRIO DA EDUCAÇÃO**  
**SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA**  
**INSTITUTO FEDERAL GOIANO – CAMPUS IPORÁ**  
**TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**RELATÓRIO DE ATIVIDADES DESENVOLVIDAS: ESTÁGIO**  
**MACHINE LEARNING COM AWS**

**DOUGLAS RAMOS MARTINS**

**ME. WESLEY FLÁVIO DE MIRANDA**  
**Orientador**

Relatório de Estágio apresentado ao Instituto Federal Goiano – *Campus* Iporá, como requisito parcial para conclusão do Curso de Tecnologia em Análise e Desenvolvimento de Sistemas.

**IPORÁ, GO**  
**Dezembro/2024**

## SUMÁRIO

1. INTRODUÇÃO .....	01
2. CARACTERIZAÇÃO DA EMPRESA E DADOS DO ESTÁGIO .....	02
3. ATIVIDADES DESENVOLVIDAS .....	04
4. CONSIDERAÇÕES FINAIS .....	18
REFERÊNCIAS .....	19

## 1 - INTRODUÇÃO

Este relatório de conclusão de estágio descreve os projetos realizados durante o programa de bolsas *Machine Learning* com AWS, no período de novembro de 2023, até abril de 2024, abordando as metodologias utilizadas e as atividades desenvolvidas.

AWS ou Amazon Web Services é uma plataforma de serviços que permite o desenvolvimento de aplicações por meio de computação em nuvem, ou seja, com os recursos alocados em um ambiente com infraestrutura fornecida, sendo acessados via internet pelos desenvolvedores e usuários. Wittig e Wittig (2016, p. 25)

A Amazon Web Services (AWS) é uma plataforma de *web services* que oferece soluções de computação, armazenamento e rede em diferentes camadas de abstração. Você pode usar os serviços para hospedar sites, executar aplicações empresariais e minerar grandes quantidades de dados. O termo *web service* significa que os serviços podem ser controlados por uma interface web. Esta interface pode ser usada por máquinas ou por pessoas por meio de uma interface gráfica de usuário.

Com foco na aplicação prática de técnicas de aprendizado de máquina e desenvolvimento de soluções em nuvem, o estágio proporcionou experiência com tecnologias utilizadas no desenvolvimento e disponibilização em produção dessas soluções.

O estágio é fundamental para a formação acadêmica e o início da carreira profissional devido a grande demanda de profissionais com experiência comprovada para empregos de desenvolvimento de software. Além de fornecer esta primeira experiência com o mercado de trabalho o estágio também possibilita a execução dos conceitos aprendidos durante a graduação.

A seguir no capítulo dois, apresentarei a caracterização da empresa e os dados do estágio, no capítulo três detalharei as atividades desenvolvidas no decorrer estágio e no capítulo quatro trarei minhas considerações finais sobre este período.

## 2 – CARACTERIZAÇÃO DA EMPRESA E DADOS DO ESTÁGIO

Em conformidade com a cordial solicitação de privacidade e sigilo dos dados, aqui trataremos apenas como Empresa, não usando diretamente sua marca, razão social ou qualquer meio identificador direto.

O estágio foi realizado no período de 13 de novembro de 2023 até 10 de abril de 2024. Com carga horária diária de 4 horas, na modalidade *home office*, no período de 09h da manhã até 13h da tarde, totalizando 400 horas de estágio. O supervisor do estágio pela referida instituição de ensino foi o professor Wesley Flávio de Miranda.

### 2.1-Metodologia utilizada

O estágio foi dividido em dez diferentes *sprints*, um período de tempo fixo onde é trabalhada uma quantidade específica de tarefas, cada uma com duração média de 14 dias. Seguindo o modelo de *framework* Scrum, que é um conjunto de ferramentas e estratégias utilizadas durante o desenvolvimento de um projeto, se caracteriza pela sua alta adaptabilidade. A equipe de desenvolvimento define quais tarefas serão realizadas e sua ordem de acordo com mudanças nos requisitos, permitindo a auto-organização sem perder o foco no produto. Sutherland (2016, p. 9) descreve o framework como:

O Scrum acolhe a incerteza e a criatividade. Cria um alicerce para o aprendizado, permitindo que as equipes avaliem o que já criaram e de que forma o criaram, o que é igualmente importante. A estrutura do Scrum procura aproveitar a maneira como as equipes de fato trabalham, fornecendo ferramentas para se auto-organizarem e otimizarem em pouco tempo a rapidez e a qualidade do trabalho.

Durante cada *sprint* foram disponibilizadas videoaulas na plataforma de cursos online UdemY Business seguindo os conteúdos planejados da trilha de estudo, introduzindo novas tecnologias e propondo o desenvolvimento de um pequeno projeto de software durante o período de 14 dias.

Com exceção da primeira *sprint*, os participantes do programa de bolsas foram divididos em equipes de três ou quatro desenvolvedores para cada projeto, nunca repetindo a mesma configuração de membros.

A definição das tarefas a serem realizadas e o tipo de aplicação ficava sob responsabilidade do time de desenvolvimento, podendo recorrer aos instrutores para solução de dúvidas a respeito dos conteúdos das videoaulas e também do projeto.

Diariamente às 10h15min da manhã era realizada uma reunião, com duração máxima de quinze minutos, no modelo *Daily Scrum*. Durante a reunião os participantes informavam, de forma resumida, o trabalho concluído, o que está planejado para o dia e os desafios encontrados, sempre que possível instrutores e colegas se colocavam a disposição logo após a reunião para ajudar a superar os desafios relatados.

### 3 – ATIVIDADES DESENVOLVIDAS

A seguir, apresentarei uma descrição dos projetos de software realizados em cada *sprint* do programa de bolsas.

#### 3.1 – Sistema para verificação de PIN em Javascript

No primeiro projeto do estágio foi solicitada a criação de um sistema que recebesse um código mínimo de quatro dígitos pelo usuário e verificasse se este número é maior ou menor que o código pré-definido no sistema.

A aplicação que desenvolvi, gera um código aleatório de quatro dígitos, não limitando a quantidade de tentativas que o usuário pode realizar e adiciona ao tratamento de resultados a palavra “muito”, caso a diferença fosse maior ou menor que mil unidades. Exemplo: Uma tentativa com o número 1234 para o código 566, informava na tela: “O valor é muito menor que o código”.

O código foi desenvolvido utilizando Javascript, “[...] Javascript é uma linguagem de alto nível, dinâmica, interpretada e não tipada, conveniente para estilos de programação orientados a objetos e funcionais” (Flanagan, 2012, p. 7), aplicando a linguagem para geração de um código aleatório e sua comparação com o valor inserido pelo usuário.

Utilizei a *HyperText Markup Language* (HTML), que serve para “[...] a criação de documentos estruturados em títulos, parágrafos, listas, links, tabelas, formulários e em muitos outros elementos nos quais podem ser incorporadas imagens e objetos[...]” (Flatschart, 2011, p. 9), para exibição da aplicação e recepção dos valores das tentativas do usuário. Também utilizei o *Cascading Style Sheets* (CSS), “[...] linguagem de estilo, responsável pela formatação e apresentação do conteúdo: layout, cores, fontes.” (Flatschart, 2011, p. 10), para formatação da tela da aplicação.

Durante o desenvolvimento, por ainda ser inexperiente na linguagem de programação Javascript, encontrei dificuldades para realizar integração de funções em arquivos diferentes e enviar seus resultados para o *front-end* da aplicação, que é a interface de interação com o usuário.



uma função para identificar os diferentes tipos de dados e retorná-los com suas devidas extensões para o HTML.

### **3.3 – Subindo uma aplicação utilizando Docker:**

Durante a terceira *sprint* fomos avaliados na construção de um sistema em NodeJS que, de forma semelhante ao projeto da sprint 2, realizava consultas a partir de uma API pública, mas que deveria ser disponibilizado com execução em Docker dentro da AWS Cloud.

Gomes (2017, p. 23) caracteriza Docker como:

Docker é uma plataforma aberta, criada com o objetivo de facilitar o desenvolvimento, a implantação e a execução de aplicações sem ambientes isolados. Foi desenhada especialmente para disponibilizar uma aplicação da forma mais rápida possível.

A API selecionada para o projeto foi a Weatherstack, para retornar informações sobre o clima em qualquer localidade que o usuário inserir, por meio de chamadas assíncronas com rotas da biblioteca Express, comunicando com o Docker pela porta 3000.

Quanto ao código, a lógica de pesquisa precisava apenas receber uma *string* informada pelo usuário no *front-end* e depois enviá-la junto da chave de API que nosso grupo criou. Para comunicação com a API utilizamos a biblioteca Axios, permitindo a requisição ser feita de forma assíncrona.

Ressalto que encontramos dificuldades na hora de hospedar o arquivo Docker contendo nosso código dentro da AWS, porque nossas contas não tinham permissão para realizar tal operação e, para contornar o problema, foi escolhida a plataforma Dockerhub para receber o arquivo Docker e enviar uma versão do arquivo que nos permitiu subir uma versão com o AWS Elastic Beanstalk.

### **3.4 – Aplicação em produção no AWS Elastic Beanstalk**

Neste projeto nos foi proposta a criação de uma aplicação com três rotas do tipo GET, a primeira deveria retornar um texto simples com o nome da equipe e da aplicação junto do código de estado 200. A segunda rota deve consumir a API chamada chucknorris.io, que contém uma coleção de piadas sobre o ator americano Chuck Norris e ao ser acessada retornar os dados de uma piada aleatória. Já a terceira rota, deve consumir a API chamada boredapi

que fornece dados de atividades do dia-a-dia para pessoas entediadas, retornando os dados de uma atividade aleatória.

Foi requisitado que o retorno das rotas 2 e 3 seria em JavaScript *Object Notation* (JSON), definido por Lindsay Bassett (2019) como formato para intercâmbio de dados utilizado em vários sistemas de software, e em ambas as rotas era necessário um campo identificador no formato *Globally Unique Identifier* (GUID) gerado aleatoriamente.

Na implementação do código, foi utilizado a biblioteca uuid da linguagem NodeJs para realizar a criação dos campos identificadores e, de forma semelhante aos projetos anteriores, conversamos com as APIs selecionadas para o projeto utilizando as bibliotecas Express e Axios. Após isso, configuramos a aplicação para rodar na porta 8080 e assim permitir o acesso dos dados na rede virtual privada que criamos na AWS e então enviamos o arquivo comprimido para o AWS Elastic Beanstalk. A Figura 2 mostra o retorno formatado dos dados obtidos na consulta a API.

```
1 {
2   "data_atualizacao": "05-01-2020",
3   "data_criacao": "05-01-2020",
4   "icone": "https://assets.chucknorris.host/img/avatar/chuck-norris.png",
5   "id": "98399F99-7815-47AC-91DF-9F298A50B800",
6   "piada": "Mother Teresa said CHUCK NORRIS was the only man she ever wanted to marry.",
7   "referencia": "https://api.chucknorris.io/jokes/_eLD9PRxS9-hNbuTF7-oA"
8 }
```

Figura 2 - Exemplo do retorno da rota 2

### 3.5 – Treinamento de Dados – Reservas de hotéis

Durante a quinta sprint recebemos uma base de dados de reservas de quartos de hotéis e treinamos um modelo de inteligência artificial para prever o preço de um quarto com base no seu tamanho, quantidade de camas, banheiros e nas características da reserva, como dia da semana, se o hotel tem estacionamento, etc. Na figura 3 temos a representação de uma pequena parte da base de dados recebida, os valores e suas colunas estão separados de acordo com sua cor.

```

1 Booking_ID,no_of_adults,no_of_children,no_of_weekend_nights,no_of_week_nights,type_of_meal_plan,required_car_
2 INN00001,2,0,1,2,Meal Plan 1,0,Room_Type 1,224,2017,10,2,Offline,0,0,0,65,0,Not_Canceled
3 INN00002,2,0,2,3,Not Selected,0,Room_Type 1,5,2018,11,6,Online,0,0,0,106.68,1,Not_Canceled
4 INN00003,1,0,2,1,Meal Plan 1,0,Room_Type 1,1,2018,2,28,Online,0,0,0,60,0,Canceled
5 INN00004,2,0,0,2,Meal Plan 1,0,Room_Type 1,211,2018,5,20,Online,0,0,0,100,0,Canceled
6 INN00005,2,0,1,1,Not Selected,0,Room_Type 1,48,2018,4,11,Online,0,0,0,94.5,0,Canceled
7 INN00006,2,0,0,2,Meal Plan 2,0,Room_Type 1,346,2018,9,13,Online,0,0,0,115,1,Canceled
8 INN00007,2,0,1,3,Meal Plan 1,0,Room_Type 1,34,2017,10,15,Online,0,0,0,107.55,1,Not_Canceled
9 INN00008,2,0,1,3,Meal Plan 1,0,Room_Type 4,83,2018,12,26,Online,0,0,0,105.61,1,Not_Canceled
10 INN00009,3,0,0,4,Meal Plan 1,0,Room_Type 1,121,2018,7,6,Offline,0,0,0,96.9,1,Not_Canceled
11 INN00010,2,0,0,5,Meal Plan 1,0,Room_Type 4,44,2018,10,18,Online,0,0,0,133.44,3,Not_Canceled
12 INN00011,1,0,1,0,Not Selected,0,Room_Type 1,0,2018,9,11,Online,0,0,0,85.03,0,Not_Canceled
13 INN00012,1,0,2,1,Meal Plan 1,0,Room_Type 4,35,2018,4,30,Online,0,0,0,140.4,1,Not_Canceled
14 INN00013,2,0,2,1,Not Selected,0,Room_Type 1,30,2018,11,26,Online,0,0,0,88,0,Canceled
15 INN00014,1,0,2,0,Meal Plan 1,0,Room_Type 1,95,2018,11,20,Online,0,0,0,90,2,Canceled
16 INN00015,2,0,0,2,Meal Plan 1,0,Room_Type 1,47,2017,10,20,Online,0,0,0,94.5,2,Not_Canceled
17 INN00016,2,0,0,2,Meal Plan 2,0,Room_Type 1,256,2018,6,15,Online,0,0,0,115,1,Canceled

```

Figura 3 - Modelo da Base de dados

O modelo foi construído com a linguagem Python dentro do serviço AWS Sagemaker, plataforma que permite criação e treinamento de modelos de *Machine Learning*, nele foi realizado um tratamento inicial dos dados para encontrar quais colunas da base de dados causavam mais impacto no preço de um quarto. Também construímos uma matriz de correlação para escolher quais colunas seriam removidas antes de iniciar o treinamento dos dados, essa matriz está representada na figura 4.

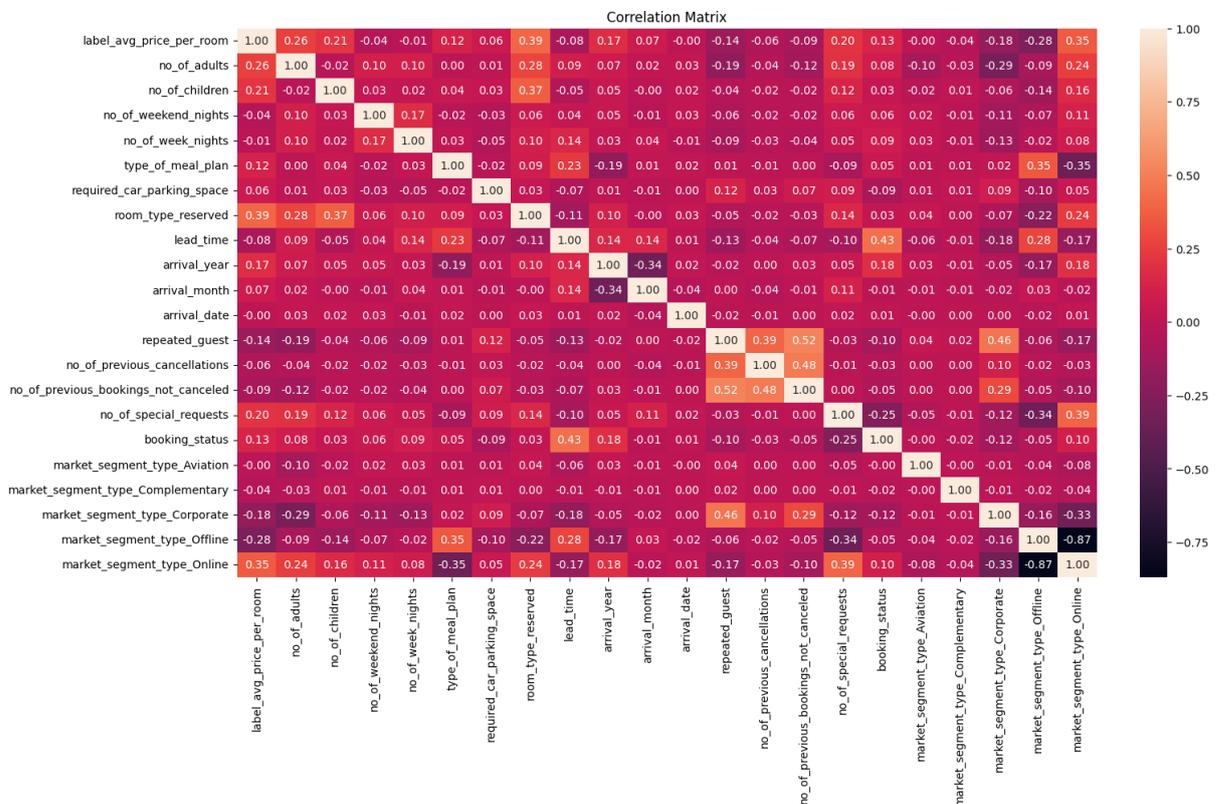


Figura 4 - Matriz de correlação

Depois o modelo foi treinado utilizando 72% da base de dados, para prever os preços de 10% da base de dados, sem a coluna de valor, e recebeu os 18% restantes para validar a eficiência em prever os valores dos quartos.

Para apresentação do modelo foi enviada uma imagem Docker, do projeto para o serviço AWS Elastic Beanstalk, onde foi possível realizar testes no modelo a partir da interface gráfica que criamos.

### 3.6 – Conversão de texto para áudio com AWS Polly

Na sexta sprint construímos uma página em HTML que recebe uma frase digitada pelo usuário e aciona via API uma função lambda, que aciona o serviço AWS Polly para criar um áudio *text-to-speech* dessa frase. Tal código pode acionar três rotas diferentes para tratar esta conversão

Na rota v1 é enviada uma postagem da frase inserida no HTML direto para a AWS Polly e o áudio gerado é salvo dentro de um *bucket Simple Storage Service (S3)*, container para armazenar objetos dentro do serviço AWS S3, que criamos. A figura 5 ilustra a arquitetura do fluxo da rota V1.

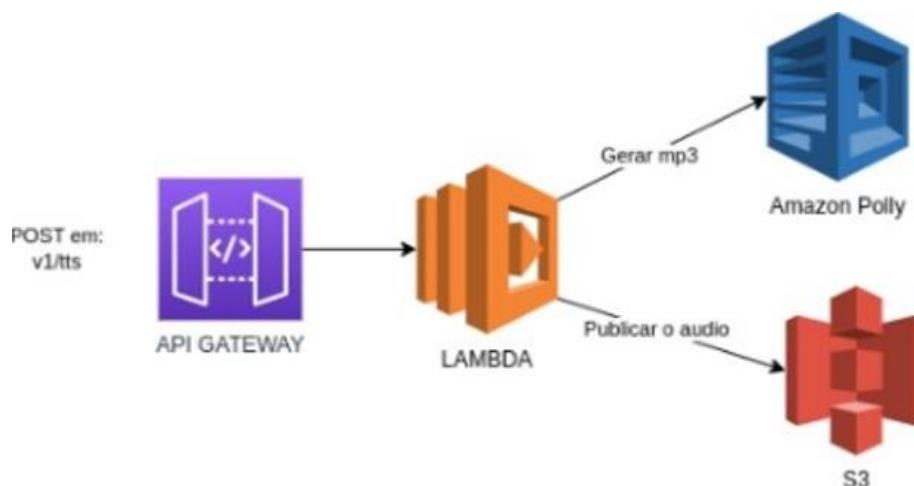


Figura 5 - Arquitetura da rota v1

Na rota v2 são realizadas as operações da rota v1 e também é gerado um identificador único (id) para a frase recebida. O id, o texto da frase e o link de onde o áudio foi gravado pela AWS Polly são enviados para uma tabela não sequencial hospedada no serviço AWS DynamoDB. O Dynamo DB, como descreve Tanmay Deshpande (2015) é um serviço de

banco de dados NoSQL, totalmente gerenciado e caracterizado por sua escalabilidade infinita, alta avaliabilidade e segurança. A Figura 6 ilustra a arquitetura do fluxo da rota v2.

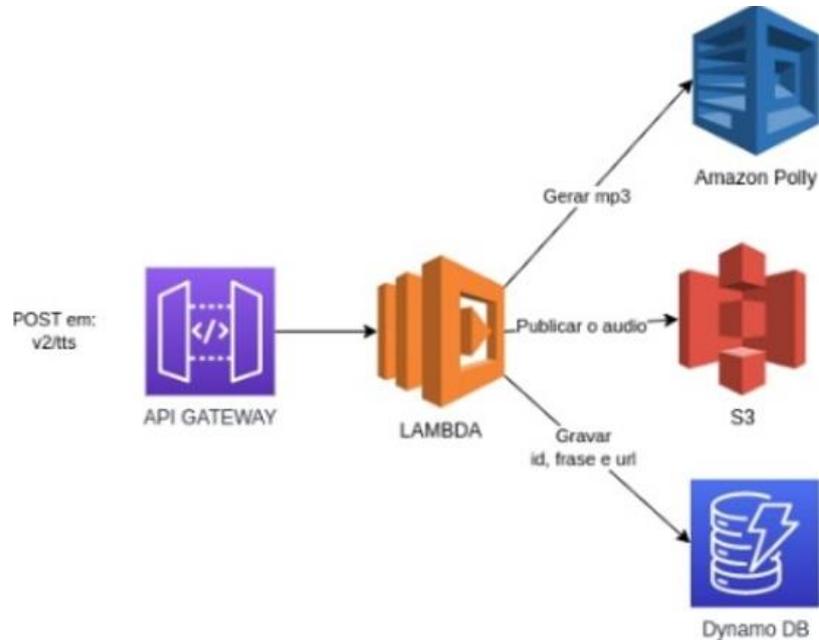


Figura 6 - Arquitetura da rota v2

Na rota v3, ocorre uma verificação dentro da tabela contida no DynamoDB, comparando o id único da frase criada com os outros ids no banco de dados. Caso já exista uma transcrição para esta frase, é retornado ao usuário a *Uniform Resource Locator* (URL) onde pode-se acessar o áudio já existente, caso não exista, são realizadas as operações da rota v2. A figura 7 ilustra a arquitetura do fluxo da rota v3.

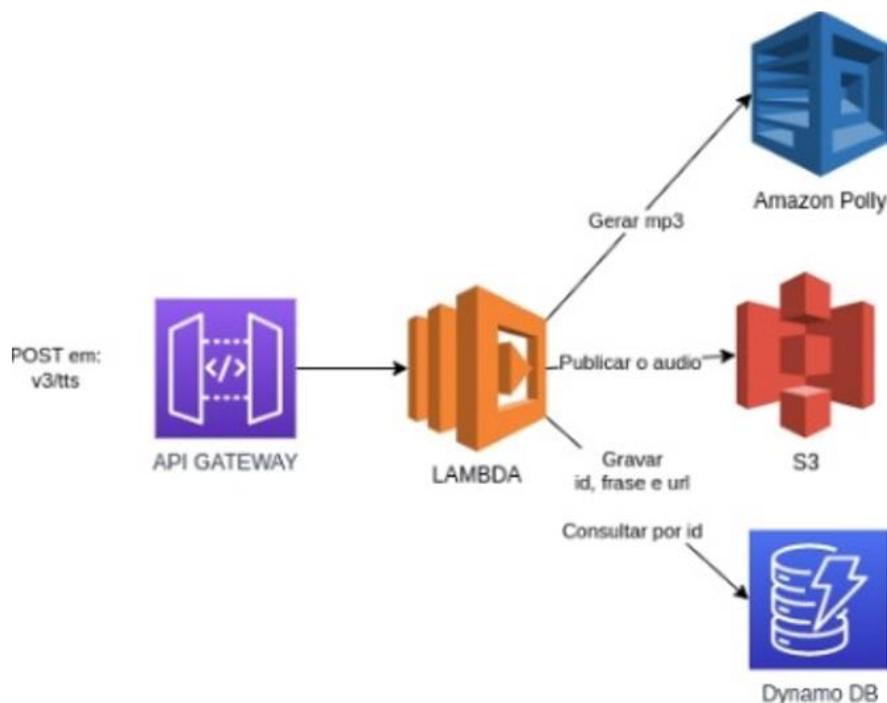


Figura 7 - Arquitetura da rota v3

O *deploy* desta aplicação foi realizado utilizando a arquitetura Serverless, ou seja, seguindo os padrões de infraestrutura como código, não sendo expostas credenciais ou variáveis de ambiente. Michael Hausenblas (2016) acrescenta que numa aplicação serverless é necessária a presença de um gatilho para a execução das funções, uma interface para gerência, configuração e registro das funções e os pontos de integração que conversarão com os sistemas externos. No entanto, não foi utilizada a política de permissões mínimas neste projeto, devido a antes deste projeto, a equipe não ser familiarizada com o *framework* necessário.

Quanto à estrutura do HTML, foram criados três *cards* simples para inserção de frases e uma breve descrição do que acontece em cada um, pois a configuração das rotas, comunicação dos serviços na nuvem e *deploy* em Serverless eram o foco deste projeto. A figura 8 mostra o retorno do código HTML deste projeto.

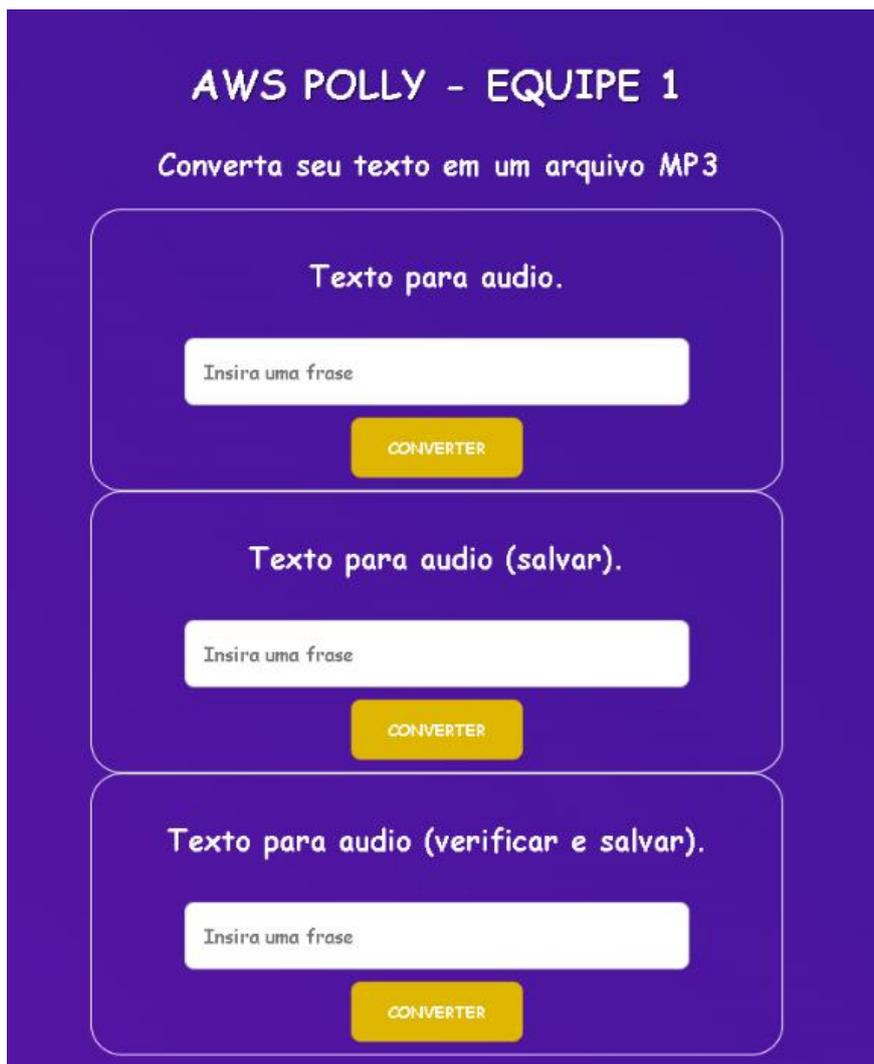


Figura 8 – Página inicial

### 3.7 – Criação de um chatbot com AWS Lex

Para a sétima sprint foi solicitada a criação de um chatbot com o modelo Amazon Lex V2 e a conexão do projeto desenvolvido em uma plataforma de troca de mensagens. Já o tema do chatbot era de livre escolha.

Minha equipe escolheu criar um chatbot, chamado PokeHelper, para auxiliar jogadores iniciantes de *Pokémon* caso encontrem dificuldade na interação dos diferentes tipos de *Pokemons*, ou na utilização dos itens básicos presentes nos jogos. A Figura 9 representa a arquitetura desenhada para esse projeto.

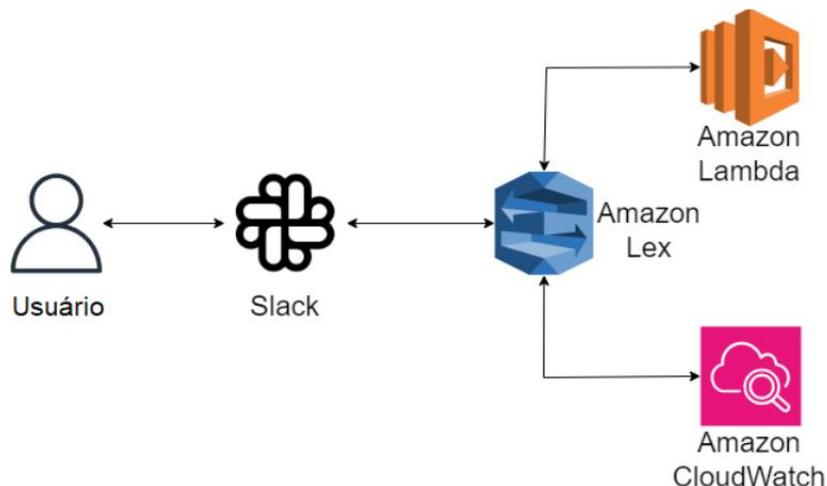


Figura 9 - Arquitetura do projeto

O chatbot trabalhava com quatro *intents*, sendo que uma *intent* é a ação que o usuário pode tomar utilizando o chatbot.

- *Intent* que cumprimenta o usuário, guarda seu nome e garante que durante a execução, o chatbot continuará a chamar o usuário pelo nome;
- *Intent* que mostra o menu de operações realizadas pelo bot;
- *Intent* que informa ao usuário quais tipos de ataques serão super efetivos, inefetivos e quais não terão efeito algum;
- *Intent* que informa o item a ser usado para tratar um status afetando o Pokémon.

Para a realização do cálculo de efetividade de tipos de ataques, o chatbot envia para a função *lambda* o(s) tipo(s) elemental(ais) do Pokémon e nela é realizado o cálculo das diferentes interações com todos os tipos elementais. A *lambda* retorna a lista de ataques com efeitos e os menos efetivos que o normal. Na figura 10 está a representação da lista de vantagens e resistências elementais utilizada nesta *intent*.

		DEFESA																	
		NORMAL	LUTADOR	VOADOR	VENENOSO	TERRESTRE	PEDRA	INSETO	FANTASMA	AÇO	FOGO	ÁGUA	PLANTA	ELETRICO	PSÍQUICO	GELO	DRAGÃO	SOMBRIO	FADA
ATAQUE	NORMAL	1x	1x	1x	1x	1x	1/2x	1x	0x	1/2x	1x	1x	1x	1x	1x	1x	1x	1x	1x
	LUTADOR	2x	1x	1/2x	1/2x	1x	2x	1/2x	0x	2x	1x	1x	1x	1x	1/2x	2x	1x	2x	1/2x
	VOADOR	1x	2x	1x	1x	1x	1/2x	2x	1x	1/2x	1x	1x	2x	1/2x	1x	1x	1x	1x	1x
	VENENOSO	1x	1x	1x	1/2x	1/2x	1/2x	1x	1/2x	0x	1x	1x	2x	1x	1x	1x	1x	1x	2x
	TERRESTRE	1x	1x	0x	2x	1	2x	1/2x	1x	2x	2x	1x	1/2x	2x	1x	1x	1x	1x	1x
	PEDRA	1x	1/2x	2x	1x	1/2x	1x	2x	1x	1/2x	2x	1x	1x	1x	1x	2x	1x	1x	1x
	INSETO	1x	1/2x	1/2x	1/2x	1x	1x	1x	1/2x	1/2x	1/2x	1x	2x	1x	2x	1x	1x	2x	1/2x
	FANTASMA	0x	1x	1x	1x	1x	1x	1x	2x	1x	1x	1x	1x	1x	2x	1x	1x	1/2x	1x
	AÇO	1x	1x	1x	1x	1x	2x	1x	1x	1/2x	1/2x	1/2x	1x	1/2x	1x	2x	1x	1x	2x
	FOGO	1x	1x	1x	1x	1x	1/2x	2x	1x	2x	1/2x	1/2x	2x	1x	1x	2x	1/2x	1x	1x
	ÁGUA	1x	1x	1x	1x	2x	2x	1x	1x	1x	2x	1/2x	1/2x	1x	1x	1x	1x	1/2x	1x
	PLANTA	1x	1x	1/2x	1/2x	2x	2x	1/2x	1x	1/2x	1/2x	2x	1/2x	1x	1x	1x	1/2x	1x	1x
	ELETRICO	1x	1x	2x	1x	0x	1x	1x	1x	1x	1x	2x	1/2x	1/2x	1x	1x	1x	1/2x	1x
	PSÍQUICO	1x	2x	1x	2x	1x	1x	1x	1x	1/2x	1x	1x	1x	1x	1x	1/2x	1x	1x	0x
	GELO	1x	1x	2x	1x	2x	1x	1x	1x	1/2x	1/2x	1/2x	2x	1x	1x	1/2x	2x	1x	1x
	DRAGÃO	1x	1x	1x	1x	1x	1x	1x	1x	1/2x	1x	1x	1x	1x	1x	1x	2x	1x	0x
	SOMBRIO	1x	1/2x	1x	1x	1x	1x	1x	2x	1x	1x	1x	1x	1x	2x	1x	1x	1/2x	1/2x
FADA	1x	2x	1x	1/2x	1x	1x	1x	1x	1/2x	1/2x	1x	1x	1x	1x	1x	2x	2x	1x	

Fonte: www.poke-blast-news.net

Figura 10 - Lista de interações. Fonte: Andretta (2022)

Para tratar os status afetando um Pokémon, os itens usados ficam armazenados dentro do fluxo de conversa. Quando o usuário informar qual status está afetando seu Pokémon, o chatbot informa o item utilizado para tratá-lo. Ao final, o chatbot foi integrado à plataforma *slack*, em um canal de conversa onde foi possível interagir com o PokeHelper.

### 3.8 – Análise e extração de *tags* de imagens com AWS Rekognition

O projeto da oitava sprint consistiu em criar duas funções *lambdas* com suporte às APIs rodando o serviço AWS Rekognition, sendo que as funções *lambdas* pegam imagens salvas em um Bucket *S3* e invocam o Rekognition. Já para a verificação dos resultados foi utilizado o serviço AWS CloudWatch.

O *deploy* da aplicação foi realizado utilizando o *framework serverless*, com duas rotas utilizando o método *POST*. Ambas as rotas comunicam com a função através de um link via *API Gateway*, enviando em formato JSON o nome do *bucket* e o nome da imagem dentro dele para a função *lambda*. Na figura 11, está a representação da arquitetura utilizada neste projeto

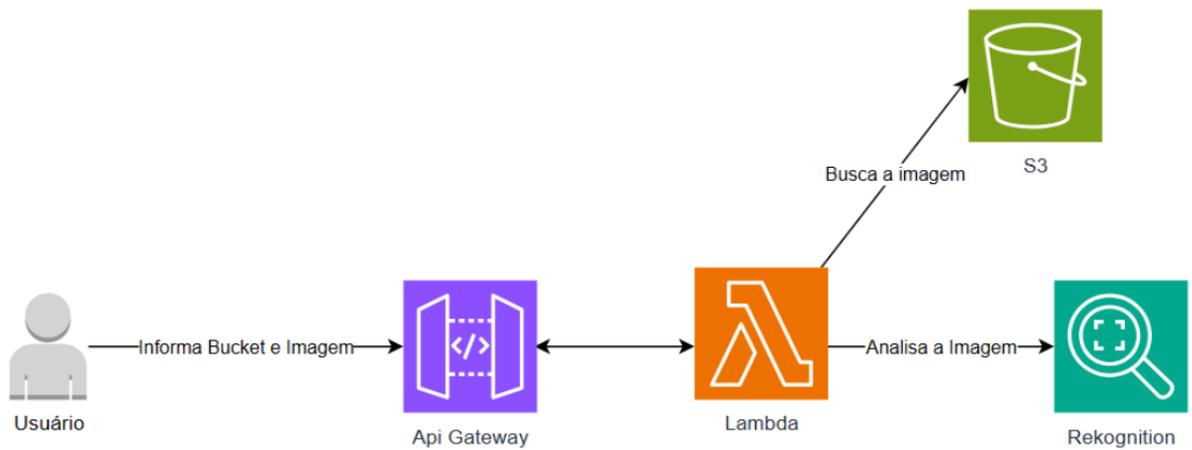


Figura 11 - Arquitetura do projeto

```

{
  "bucket": "mycatphotos",
  "imageName": "cat.jpg"
}
  
```

Figura 12 - Exemplo de *JSON* de requisição

A primeira rota invoca o AWS Rekognition para extrair informações em *tags* passando as informações no formato do JSON representado na figura 12, e o resultado obtido é formatado de acordo com o JSON demonstrado na Figura 13

```

{
  "url_da_imagem": "https://bucket-sp8.s3.amazonaws.com/thefirstcat.jpg/",
  "data_de_criacao_da_imagem": "2024-02-22 13:12:10",
  "r\u00f3tulos": [
    {
      "Confidence": 99.54550170898438,
      "Name": "Abyssinian"
    },
    {
      "Confidence": 99.54550170898438,
      "Name": "Animal"
    },
    {
      "Confidence": 99.54550170898438,
      "Name": "Cat"
    },
    {
      "Confidence": 99.54550170898438,
      "Name": "Pet"
    }
  ]
}
  
```

Figura 13 - Exemplo de resposta da rota 1

A segunda rota invoca o AWS Rekognition para identificar rostos na imagem, classificar qual é a emoção principal apresentada na foto, e a precisão desta afirmação. Quando não são encontrados rostos, o JSON como resposta tem os valores preenchidos com o valor nulo. A resposta obtida é formatada de acordo com o JSON representado na figura 14

```
1 {
2   "url_da_imagem": "https://bucket-sp8.s3.amazonaws.com/preocupado.jpg/",
3   "data_de_criacao_da_imagem": "2024-02-24 20:10:35",
4   "r\u00f3stulos": [
5     {
6       "posi\u00e7\u00e3o": {
7         "Altura": 0.3065621852874756,
8         "Esquerda": 0.20782527327537537,
9         "Topo": 0.14820387959480286,
10        "Largura": 0.15079718828201294
11      },
12      "emo\u00e7\u00e3o_classificada": "FEAR",
13      "confianca_emocao_classificada": 79.19921875
14    }
15  ]
16 }
```

Figura 14: Exemplo de resposta da rota 2

### 3.9 – Projeto final

Durante essa etapa do estágio, desenvolvemos um chatbot chamado IF Navigator, que funciona por texto ou áudio para ajudar estudantes do Instituto Federal Goiano durante o dia a dia. Quando solicitado, são fornecidos, de forma autônoma, o cardápio da cantina, locais com desconto para estudantes em sua região, consulta de editais, rotas e horários de ônibus e/ou caronas disponíveis e o horário e local de aulas do aluno. O usuário também consegue encontrar o nome e curso de outro aluno dentro da instituição por meio de reconhecimento facial a partir de uma foto enviada ao chatbot.

O chatbot foi construído utilizando o Amazon Lex para o fluxo conversacional com o usuário e foi integrado com o aplicativo Telegram, onde se pode adicionar o chatbot aos contatos e conversar com o IF Navigator.

Quanto às suas funcionalidades, o chatbot ao receber um áudio envia o arquivo mp3 para o Amazon Transcribe, que faz a transcrição do áudio e envia para o fluxo de conversa o texto obtido. Para a verificação de identidade do usuário é necessário se insira matrícula, senha e se tire uma foto para validação. Nesse caso, o reconhecimento facial é realizado

utilizando o Amazon Rekognition, que verifica a similaridade do rosto na foto enviada com as fotos existentes dentro de uma tabela no DynamoDB. As informações sobre cardápios, listas de caronas, locais com desconto para estudantes, editais e horários de aulas também são guardadas em tabelas dentro do DyanmoDB com identificadores para cada campus. A figura 15 mostra o exemplo de um fluxo de conversa com o chatbot.

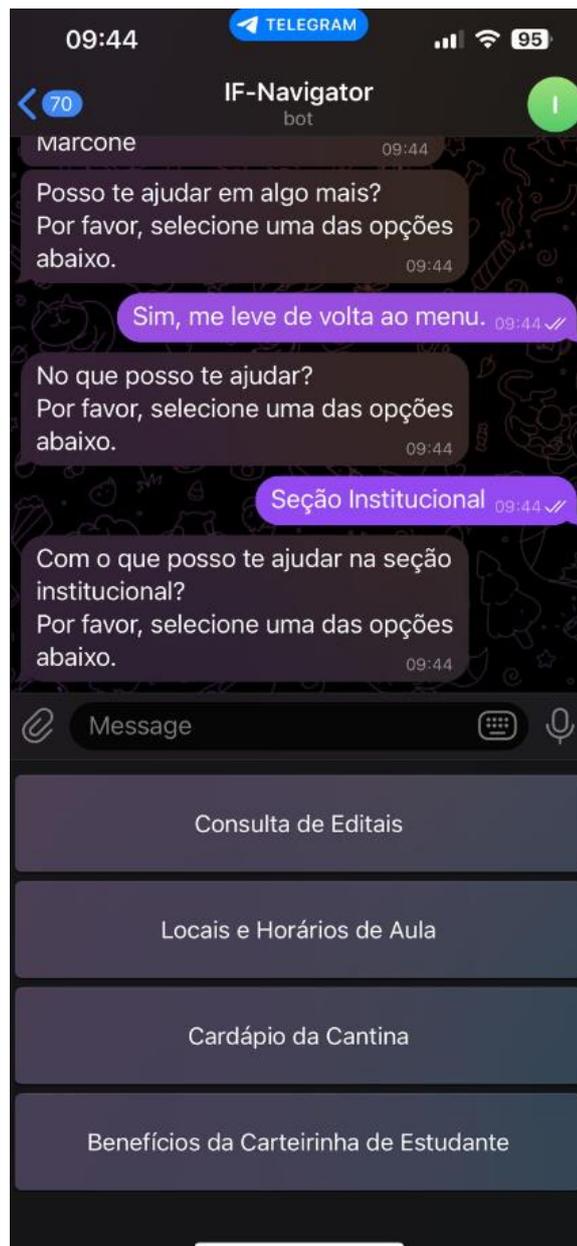


Figura 15 - Exemplo de conversa com o chatbot

#### 4 – CONSIDERAÇÕES FINAIS

O estágio na empresa foi uma experiência transformadora, que me permitiu trabalhar com tecnologias amplamente utilizadas no mercado. Trabalhar em projetos reais proporcionou uma visão dos desafios técnicos e organizacionais, melhorando minha capacidade de adaptação a mudanças e resolução de problemas.

Pude também, utilizar as habilidades e conteúdos que aprendi durante o curso de Tecnologia em Análise e Desenvolvimento de Sistemas com meus professores, aplicando-os em desafios reais de programação e desenvolvimento.

Além do desenvolvimento técnico, o convívio com profissionais experientes me ajudou no aprimoramento de habilidades interpessoais, como trabalho em equipe e comunicação.

As lições aprendidas durante este período, hoje servem de base durante o desenvolvimento de novos projetos, me ajudando desde o planejamento até a disponibilização em produção, assim como também me auxiliou a aprender a lidar com eventuais problemas de forma mais eficaz.

Agradeço ao Instituto Federal Goiano – Campus Iporá e a empresa, pela oportunidade de realizar esse estágio, através dele consegui obter a certificação internacional AWS Certified Cloud Practitioner (CLF C02) e pude dar o pontapé inicial para minha carreira no mercado de trabalho após minha efetivação na empresa.

## REFERÊNCIAS

ANDRETTA, Danilo. **Tipos de Pokémon: Conheça as Vantagens e Desvantagens de Cada Tipo.** [S. l.]: Pokémon Blast News, 20 dez. 2022. Disponível em: <https://www.poke-blast-news.net/2015/10/tipos-de-pokemon.html>. Acesso em: 17 nov. 2024.

DESHPANDE, Tanmay. **DynamoDB Cookbook.** Birmingham, Reino Unido: Packt Publishing, 2015.

FLANAGAM, David. **JavaScript: o guia definitivo.** 6. ed. Porto Alegre: Bookman, 2012.

FLATSCHART, Fábio. **HTML 5: Embarque Imediato.** Rio de Janeiro: Brasport, 2011.

GOMES, Rafael. **Docker para desenvolvedores.** Leanpub: Victoria, British Columbia, Canadá, 2017.

HAUSENBLAS, Michael. **Serverless Ops: a beginner's guide to AWS Lambda and beyond.** 1. ed. Sebastopol, California: O'Reilly, 2017.

SUTHERLAND, Jeff. **Scrum: a arte de fazer o dobro do trabalho na metade do tempo.** 2. ed. São Paulo: Leya, 2016.

TEIXEIRA, Pedro. **Hands-on Node.js.** Victoria, British Columbia, Canadá: Leanpub, 2012.

WITTIG, Andreas; WITTIG, Michael. **Amazon Web Services em ação.** 1. ed. São Paulo: Novatec Editora, 2015.



SERVIÇO PÚBLICO FEDERAL  
MINISTÉRIO DA EDUCAÇÃO  
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA  
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA GOIANO

### TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR PRODUÇÕES TÉCNICO-CIENTÍFICAS NO REPOSITÓRIO INSTITUCIONAL DO IF GOIANO

Com base no disposto na Lei Federal nº 9.610/98, AUTORIZO o Instituto Federal de Educação, Ciência e Tecnologia Goiano, a disponibilizar gratuitamente o documento no Repositório Institucional do IF Goiano (RIIF Goiano), sem ressarcimento de direitos autorais, conforme permissão assinada abaixo, em formato digital para fins de leitura, download e impressão, a título de divulgação da produção técnico-científica no IF Goiano.

#### Identificação da Produção Técnico-Científica (assinale com X)

- Tese
- Dissertação
- Monografia – Especialização
- Artigo - Especialização
- TCC - Graduação
- Artigo Científico
- Capítulo de Livro
- Livro
- Trabalho Apresentado em Evento
- Produção técnica. Qual: \_\_\_\_\_

Nome Completo do Autor: **Douglas Ramos Martins**

Matrícula: 2019105210430199

Título do Trabalho: **RELATÓRIO DE ATIVIDADES DESENVOLVIDAS: ESTÁGIO MACHINE LEARNING COM AWS**

#### Restrições de Acesso ao Documento [Preenchimento obrigatório]

Documento      confidencial:            Não      [      ]      Sim,      justifique:

---

Informe a data que poderá ser disponibilizado no RIIF Goiano: **24/12/2024**.

O documento está sujeito a registro de patente?  Sim  Não

O documento pode vir a ser publicado como livro?  Sim  Não

## DECLARAÇÃO DE DISTRIBUIÇÃO NÃO-EXCLUSIVA

O/A referido/a autor/a declara que:

1. O documento é seu trabalho original, detém os direitos autorais da produção técnico-científica e não infringe os direitos de qualquer outra pessoa ou entidade;
2. Obteve autorização de quaisquer materiais inclusos no documento do qual não detém os direitos de autor/a, para conceder ao Instituto Federal de Educação, Ciência e Tecnologia Goiano os direitos requeridos e que este material cujos direitos autorais são de terceiros, estão claramente identificados e reconhecidos no texto ou conteúdo do documento entregue;
3. Cumpru quaisquer obrigações exigidas por contrato ou acordo, caso o documento entregue seja baseado em trabalho financiado ou apoiado por outra instituição que não o Instituto Federal de Educação, Ciência e Tecnologia Goiano.

Iporá, 24 de dezembro de 2024

Douglas Ramos Martins

*Assinado eletronicamente pelo o Autor e/ou Detentor dos Direitos Autorais*

Ciente e de acordo:

Wesley Flávio de Miranda

*Assinatura eletrônica do(a) orientador(a)*

Documento assinado eletronicamente por:

- Wesley Flavio de Miranda, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 24/12/2024 02:12:50.
- Douglas Ramos Martins, 2019105210430199 - Discente, em 30/12/2024 07:50:36.

Este documento foi emitido pelo SUAP em 24/12/2024. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifgoiano.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código Verificador: 664578  
Código de Autenticação: 9ee69e1fd6



INSTITUTO FEDERAL GOIANO

Campus Iporá

Av. Oeste, Parque União, 350, Parque União, IPORA / GO, CEP 76.200-000

(64) 3674-0400



SERVIÇO PÚBLICO FEDERAL  
MINISTÉRIO DA EDUCAÇÃO  
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA  
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA GOIANO

Ata nº 127/2024 - GE-IP/CMPIPR/IFGOIANO

## ATA DA SESSÃO DE JULGAMENTO DO TRABALHO DE CURSO DE DOUGLAS RAMOS MARTINS

Aos quatro dias, do mês de dezembro de dois mil e vinte e quatro, às dezenove horas e dezenove minutos, em sessão pública, a banca examinadora designada na forma regimental pela Coordenação do Curso para julgar o trabalho de curso intitulado “**RELATO DE ATIVIDADES DESENVOLVIDAS: ESTÁGIO MACHINE LEARNING COM AWS**”, apresentada pelo acadêmico Douglas Ramos Martins, como parte dos requisitos necessários à obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas. A banca examinadora foi presidida pelo orientador do trabalho de curso, professor **Me. Wesley Flávio de Miranda**, tendo como membros avaliadores: o professor **Dr. Thamer Horbylon Nascimento** e a professora **Ma. Luciana Recart Cardoso**. Aberta a sessão, o acadêmico expôs seu trabalho. Em seguida, foi arguido pelos membros da banca e:

( X ) tendo demonstrado suficiência de conhecimento e capacidade de sistematização do tema de seu trabalho de curso, a banca conclui pela **aprovação** do acadêmico, sem restrições.

( ) tendo demonstrado suficiência de conhecimento e capacidade de sistematização do tema de seu trabalho de curso, a banca conclui pela **aprovação** do acadêmico, **condicionada a satisfazer as exigências** listadas na Folha de Modificação de Trabalho de Curso anexa à presente ata, no prazo máximo de 80 (oitenta) dias, a contar da presente data, ficando o professor orientador responsável por atestar o cumprimento dessas exigências.

( ) não tendo demonstrado suficiência de conhecimento e capacidade de sistematização do tema de seu trabalho de curso, a banca conclui pela **reprovação** do acadêmico.

Conforme avaliação individual de cada membro da banca, será atribuída a nota **nove (9,0)** para fins de registro em histórico acadêmico.

Os trabalhos foram encerrados às 19 horas e 56 minutos. Nos termos do Regulamento do Trabalho de Curso do Curso de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal Goiano – Campus Iporá, lavrou-se a presente ata que, lida e julgada conforme, segue assinada pelos membros da banca examinadora.

Assinado eletronicamente

Wesley Flávio de Miranda, Me. Orientador

Thamer Horbylon Nascimento, Dr.

Luciana Recart Cardoso, Ma.

Documento assinado eletronicamente por:

- **Wesley Flavio de Miranda**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 04/12/2024 20:55:44.
- **Thamer Horbylon Nascimento**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 04/12/2024 21:01:01.
- **Luciana Recart Cardoso**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 04/12/2024 21:01:07.

Este documento foi emitido pelo SUAP em 04/12/2024. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifgoiano.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código Verificador: 658376

Código de Autenticação: 1801dd8e54



INSTITUTO FEDERAL GOIANO

Campus Iporá

Av. Oeste, Parque União, 350, Parque União, IPORA / GO, CEP 76.200-000

(64) 3674-0400