

FACULDADE DE TECNOLOGIA DE AMERICANA

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

GIOVANNI MATEUS BARBIERI

[TÍTULO]

TRABALHO DE CONCLUSÃO DE CURSO

RELATÓRIO DE DESENVOLVIMENTO

AMERICANA

2021

SUMÁRIO

[1. INTRODUÇÃO 3](#_Toc117521416)

[1.1 IDENTIFICAÇÃO DO PROBLEMA 3](#_Toc117521417)

[1.2 LEVANTAMENTO DE REQUISITOS 3](#_Toc117521418)

[1.3 DESCRIÇÃO DA SOLUÇÃO PROPOSTA 3](#_Toc117521419)

[1.4 IDENTIFICAÇÃO DAS TECNOLOGIAS 3](#_Toc117521420)

[1.5 IDENTIFICAÇÃO DOS RECURSOS NECESSÁRIOS 4](#_Toc117521421)

[1.6 REVISÃO BIBLIOGRÁFICA 4](#_Toc117521422)

[2. METODOLOGIA 4](#_Toc117521423)

[2.1 ETAPAS DE DESENVOLVIMENTO 4](#_Toc117521424)

[2.2 CRONOGRAMA 5](#_Toc117521425)

[3. PROJETO 5](#_Toc117521426)

[3.1 DIAGRAMA DE CASO DE USO 5](#_Toc117521427)

[3.2 DIAGRAMA DE CLASSES 5](#_Toc117521428)

[3.3 DIAGRAMA DE ATIVIDADES 6](#_Toc117521429)

[3.4 DICIONÁRIO DE DADOS 7](#_Toc117521430)

[4. DESENVOLVIMENTO DO PROTÓTIPO 8](#_Toc117521431)

[4.1 DESCRIÇÃO DAS TELAS 8](#_Toc117521432)

[4.2 LAYOUT 8](#_Toc117521433)

# INTRODUÇÃO

## Identificação do problema

Com o progressivo crescimento da popularidade do jogo de cartas “*Magic: The Gathering”*, jogadores se veem na posição onde ao comprar diversas cartas de outros jogadores, o processo de soma dos preços se vê cada vez mais trabalhoso, devido a falta de um aplicativo especializado no Brasil.

## Levantamento De Requisitos

Levando em conta a opinião de jogadores, foi possível concluir que existem problemas com o método atual de consulta de preços. Partindo disso, podemos gerar uma lista de requisitos mínimos:

|  |
| --- |
| Consulta de Preços em Real (R$) |
| Velocidade Alta de Consulta |
| Consulta por edição |
| Consulta por ‘acabamento’ |

Tabela 1 - Tabela de Requisitos

## Descrição Da Solução Proposta

É proposta a criação de um aplicativo para dispositivos móveis capazes de auxiliar jogadores em um ambiente lúdico a consultar preços de cartas individuais de forma rápida e com a menor quantidade de esforço possível, utilizando de tecnologias de análise de imagem e “*APIs”*.

## Identificação Das Tecnologias

Após uma análise das linguagens de programação do mercado, três se destacaram para o desenvolvimento de aplicativos móveis: Java, Kotlin e Python.

Cada linguagem conta com seu conjunto de vantagens e desvantagens, porém a simplicidade e flexibilidade da linguagem Python a tornou a mais promissora.

Para a análise de imagem em Python, diversas bibliotecas já estabelecidas tornam o processo mais fácil, como *“easyocr”* e *“pytesseract.”*

Já para o processo de requisição de preços, devido a falta de uma *“API”* do mercado brasileiro, se vê necessário o desenvolvimento de uma, com a linguagem C# sendo escolhida para esse objetivo.

## Identificação Dos Recursos Necessários

Para a produção do aplicativo, poucos recursos se mostram evidentes. Além do tempo de desenvolvimento e teste, a API desenvolvida precisa estar disponível na internet, e para isso, um servidor para mantê-la online. Devido ao tamanho do projeto, uma interface simples como Azure se torna efetiva para a implantação da API.

A escolha de bibliotecas terá como foco bibliotecas de código aberto com licenças para uso livre.

## Revisão Bibliográfica

JAIDED AI. **EasyOCR**: Ready-to-use OCR with 80+ supported languages and all popular writing scripts including Latin, Chinese, Arabic, Devanagari, Cyrillic and etc.. V1.6.2. [*S. l.*], 2020. Disponível em: https://github.com/JaidedAI/EasyOCR. Acesso em: 18 out. 2022.  
  
LEE, Matthias. **pytesseract**: A Python wrapper for Google Tesseract. V0.3.10. [*S. l.*], 2010. Disponível em: https://github.com/madmaze/pytesseract. Acesso em: 18 out. 2022.

# METODOLOGIA

## Etapas De Desenvolvimento

Como forma de facilitar o aprendizado e a sequência de desenvolvimento, o processo será divido em partes, separadas por tecnologia, permitindo assim o estudo focado, para que a concatenação do conhecimento seja mais fluida e organizada.

Dessa forma, temos as seguintes etapas:

* Desenvolvimento da API de preços;
* Desenvolvimento da tecnologia de análise de imagem;
* Integração da API com o reconhecimento;
* Desenvolvimento da interface do aplicativo;
* Integração do aplicativo completo;
* Implantação final.

## Cronograma

Linha do tempo

Descrição gerada automaticamente

Figura 1 - Diagrama de Gantt do Cronograma

Com as etapas decididas anteriormente, foi gerado um Diagrama de Gantt com o cronograma esperado do desenvolvimento do aplicativo.

É importante notar que o diagrama trabalha apenas com estimativas, pois o processo de aprendizado em realidade é extremamente orgânico e imprevisível, mas deve seguir ao menos a ordem proposta.

# PROJETO

## Diagrama De Caso De Uso

## Diagrama De Classes

Com o aplicativo em mente, não são necessárias tantas classes, por conta da simplicidade dos dados que serão coletados.

Uma imagem contendo Texto

Descrição gerada automaticamente

Figura 2 - Diagrama de Classe

## Diagrama De Atividades

O fluxo de atividades segue uma ordem lógica concisa, para facilitar a experiência do usuário.

Diagrama

Descrição gerada automaticamente

Figura 3 - Diagrama de Atividades

## Dicionário De Dados

Como o aplicativo trabalhará somente com um objeto, ele contém todas as informação necessárias.

|  |  |  |
| --- | --- | --- |
| **Carta** | | |
| **Atributo** | **Tipo** | **Descrição** |
| nome | String | Contém o nome da carta em inglês |
| precos | Dictionary<String: Dictionary<String: double>> | Dicionário que contém o nome das edições onde a carta foi impressa nas chaves e um dicionário com os acabamentos e respectivos preços. |

# DESENVOLVIMENTO DO PROTÓTIPO

## Descrição Das Telas

## Layout