

# **FOLHAR - EXPLORANDO FOLHAS DE PLANTAS COM REALIDADE AUMENTADA**

Aluno(a): Bruno Geisler Vigentas

Orientador: Dalton Solano dos Reis

Coorientador: Mauricio Capobianco Lopes

# Roteiro

- Introdução
- Objetivos
- Fundamentação teórica
- Trabalhos correlatos
- Visão Geral
- Implementação
- Resultados
- Conclusões
- Sugestões

# Introdução

- Aulas de campo são uma maneira de diversificar a aula.
- Alunos compreendem o ecossistema, aprendendo sobre flora e fauna.
- Realidade Aumentada trás informações virtuais ao mundo real.
- Na educação pode enriquecer o material didático.
- Auxilia no conhecimento de folhas, permitindo a visualização de informações em 3D a partir da digitalização das folhas.
- Amplia a possibilidade de interação dos alunos em suas saídas a campo.
- Facilita o conhecimento sobre folhas.

# Objetivos

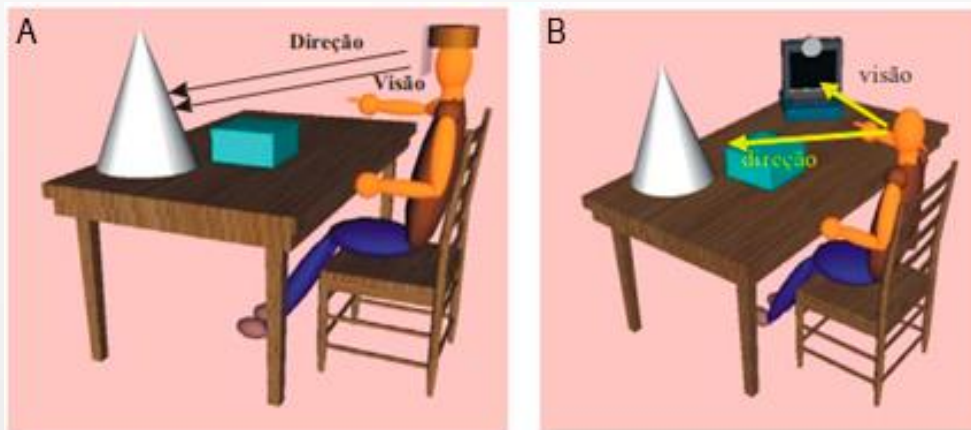
- Auxiliar no conhecimento de folhas de plantas, por intermédio da Realidade Aumentada Imersiva.
- Os objetivos específicos são:
  - utilizar as folhas das plantas como marcadores para apresentação do conteúdo em Realidade Aumentada.
  - analisar a eficácia do aplicativo com usuários da área da Biologia.

# Fundamentação Teórica

# Fundamentação Teórica

## Realidade Aumentada

- Possibilita estender o mundo real, complementando este com informações virtuais para auxiliar usuários.
- Combinação do mundo real e virtual executada de forma interativa em tempo real.
- Apresentada por meio de equipamentos tecnológicos.
- Realidade Aumentada Imersiva x Realidade Aumentada Não Imersiva.
- Marcadores.



# Fundamentação Teórica

## AR Foundation

### Unity's AR Foundation Supported Features

Functionality	ARCore	ARKit	Magic Leap	HoloLens
Device tracking	✓	✓	✓	✓
Plane tracking	✓	✓	✓	
Point clouds	✓	✓		
Anchors	✓	✓	✓	✓
Light estimation	✓	✓		
Environment probes	✓	✓		
Face tracking	✓	✓		
Meshing			✓	✓
2D Image tracking	✓	✓		
Raycast	✓	✓	✓	
Pass-through video	✓	✓		
Session management	✓	✓	✓	✓

- *Package* Unity para Realidade Aumentada.
- Cria aplicações para Android e iOS.
- Interfaces para que as bibliotecas ARCore e ARKit funcionem iguais.
- API multiplataforma.

# Fundamentação Teórica

## Barracuda

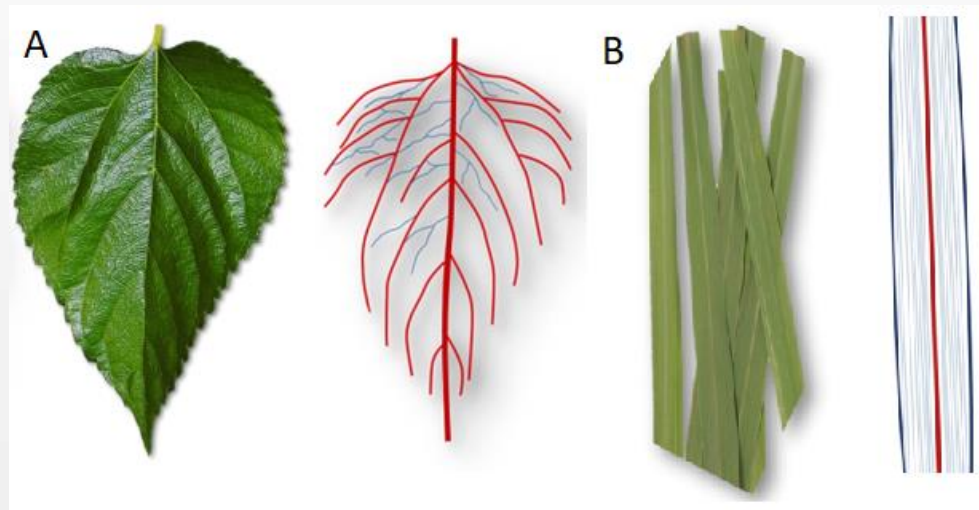
- *Package* para redes neurais no Unity.
- Permite o uso de redes pré treinadas em consoles, desktop e mobiles.
- Suporta modelos de aprendizado de máquina do tipo ONNX.
- Possibilita o intercâmbio de vários frameworks de aprendizado de máquina.
- Permite a execução em GPU e em CPU.
- Utilizada para fazer a detecção dos formatos das folhas, para servirem como marcadores.



# Fundamentação Teórica

## Morfologia das folhas

- Diferença morfológica ocorre devido ao poder da vegetação de se adaptar a diferentes condições e ambientes.
- Morfologia é estuda para reconhecimento de espécies.
- Identificação por observação de componentes como limbo, pecíolo, pulvino, nervura entre outros.
- Variedade morfológica pode ser dar na forma do limbo e nas nervuras.



# Trabalhos Correlatos (1/3)

## Plantarum (BORTOLON, 2014)

- Interação com um webservice para realizar o cadastro e classificação de espécies de plantas utilizando a API Plantarum.
- Permite capturar uma foto e enviar ao servidor para realizar a classificação da espécie ou cadastrar uma nova.
- Construído em Java Android e C#.
- Em testes feitos com 32 fotos de espécies locais, todas foram detectadas apenas uma apresentou falso positivo.

# Trabalhos Correlatos (1/3)

## Plantarum (BORTOLON, 2014)



Bortolon (2014)

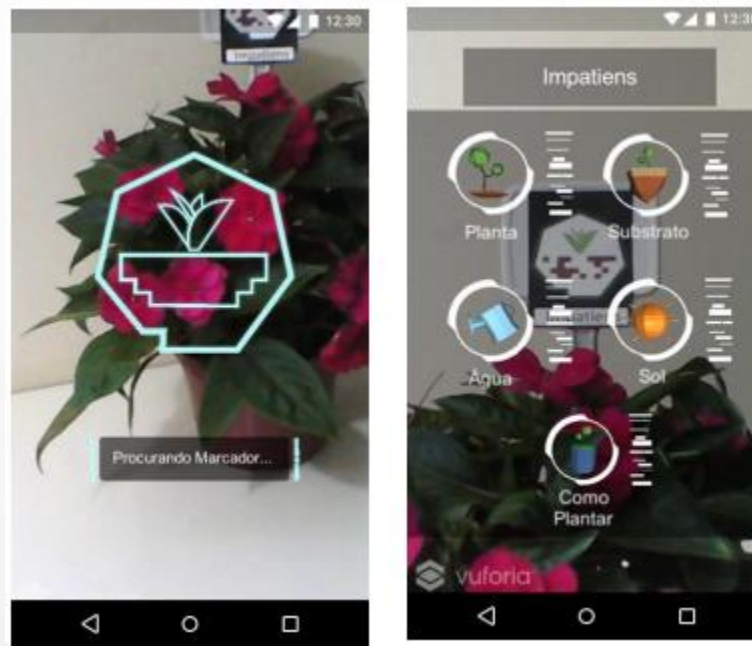
# Trabalhos Correlatos (2/3)

## Gaia (OLIVEIRA; PRADO 2018)

- Oferece informações necessárias e confiáveis para auxiliar pessoas na escolha de flores e plantas.
- Utiliza realidade aumentada para trazer informações para auxiliar o cliente na compra da planta, como a quantidade diária de água e luz necessária.
- Utiliza marcadores com símbolo gráfico para ancorar objeto 3D.
- Desenvolvido em Unity com C# e SDK Vuforia.

# Trabalhos Correlatos (2/3)

## Gaia (OLIVEIRA; PRADO 2018)



Oliveira e Prado (2018)

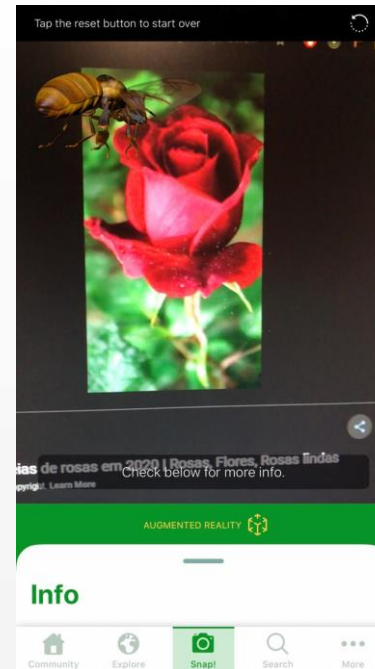
# Trabalhos Correlatos (3/3)

## PlantSnap (2020)

- Aplicativo para identificação de plantas de forma rápida.
- Faz a identificação da planta e conta com interações por meio da Realidade Aumentada.
- Permite a identificação de mais de 620 mil espécies de plantas com seu algoritmo de aprendizado de máquina.

# Trabalhos Correlatos (3/3)

## PlantSnap (2020)



PlantSnap (2020)

# Requisitos Funcionais

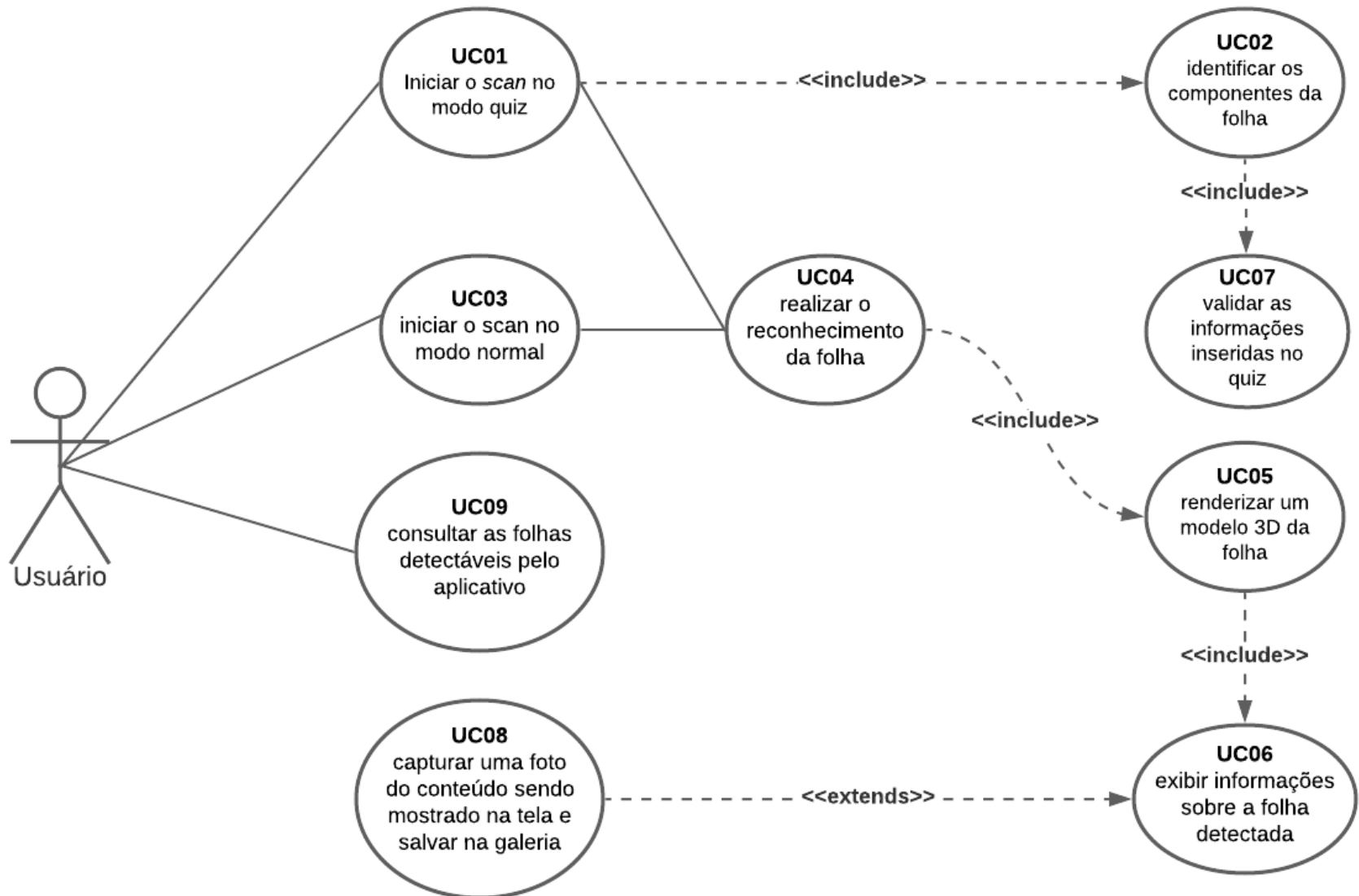
- RF01: permitir ao usuário iniciar o scan no modo quiz
- RF02: permitir ao usuário identificar os componentes da folha
- RF03: permitir ao usuário iniciar o scan no modo normal
- RF04: permitir ao usuário realizar o reconhecimento da folha
- RF05: renderizar um modelo 3D da folha
- RF06: exibir informações sobre a folha detectada
- RF07: permitir ao usuário validar as informações inseridas no modo quiz
- RF08: permitir ao usuário capturar uma foto do conteúdo sendo mostrado em sua tela e salvar em sua galeria
- RF09: permitir ao usuário consultar as folhas detectáveis pelo aplicativo



# Requisitos Não Funcionais

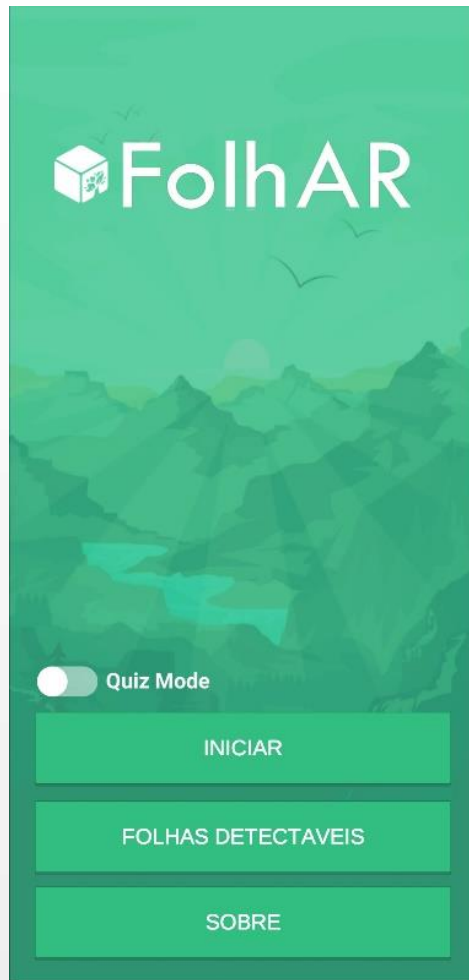
- RNF01: utilizar as folhas de plantas como marcador para ancoragem do conteúdo virtual
- RNF02: utilizar o ambiente de desenvolvimento Unity
- RNF03: ser desenvolvido para plataforma Android
- RNF04: utilizar o pacote Barracuda para a detecção das folhas
- RNF05: utilizar o *framework* AR Foundation para a apresentação da realidade aumentada

# Especificação: Diagrama de Casos de Uso



# Visão Geral

Tela inicial e tela de folhas detectáveis.



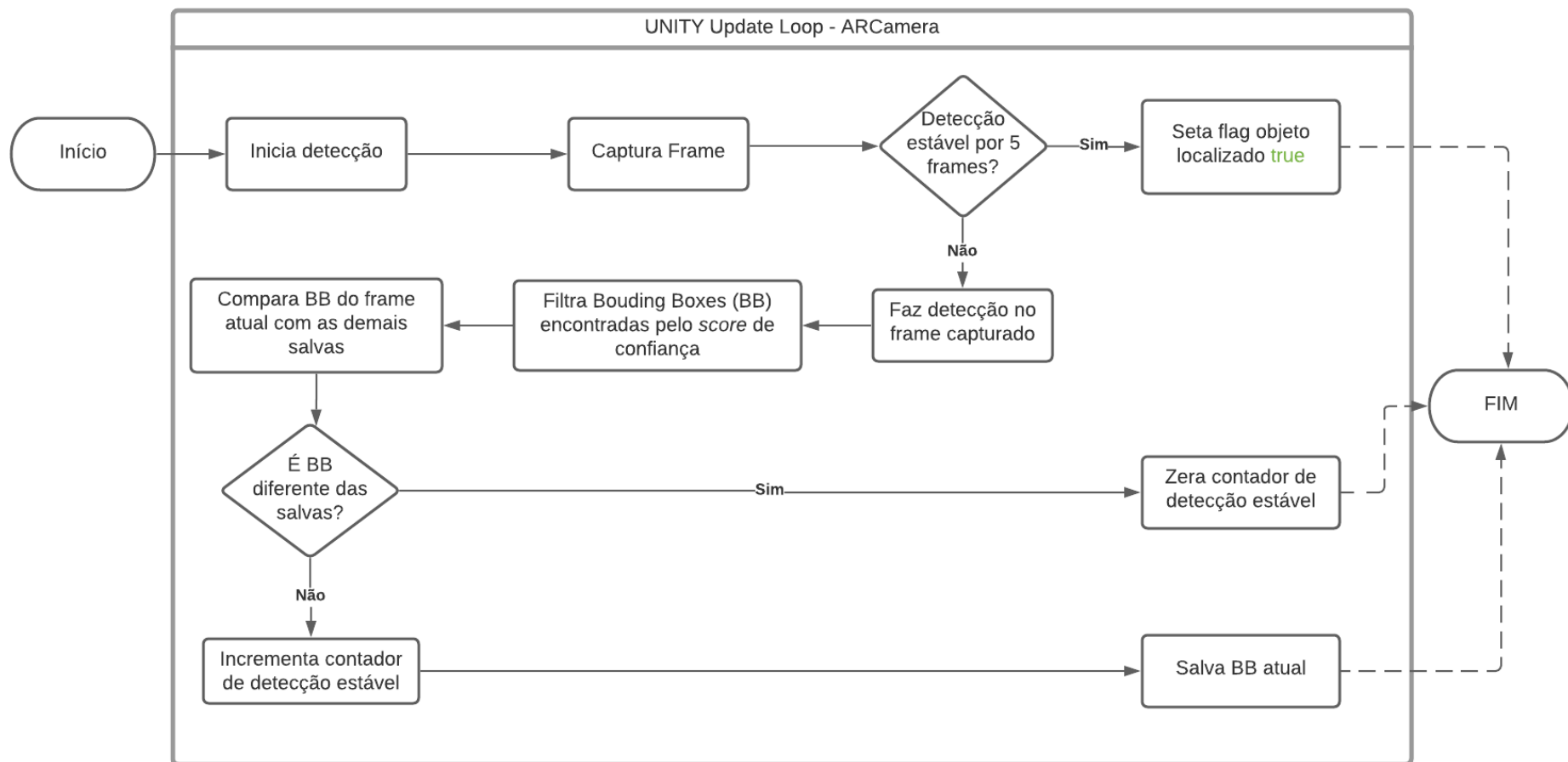
# Visão Geral

Objeto 3D da folha renderizado no modo normal e no modo quiz.



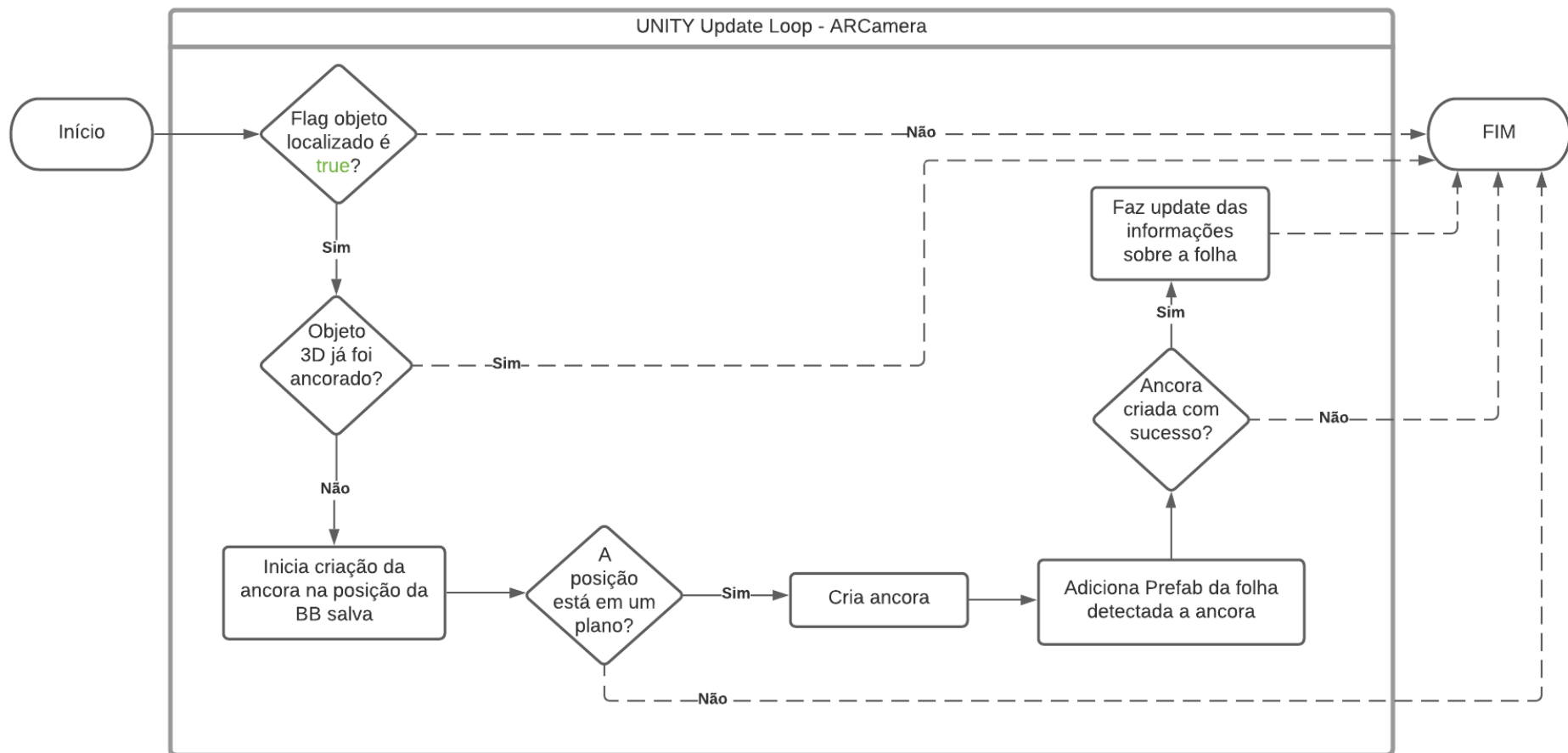
# Visão Geral

## Fluxograma detecção da folha



# Visão Geral

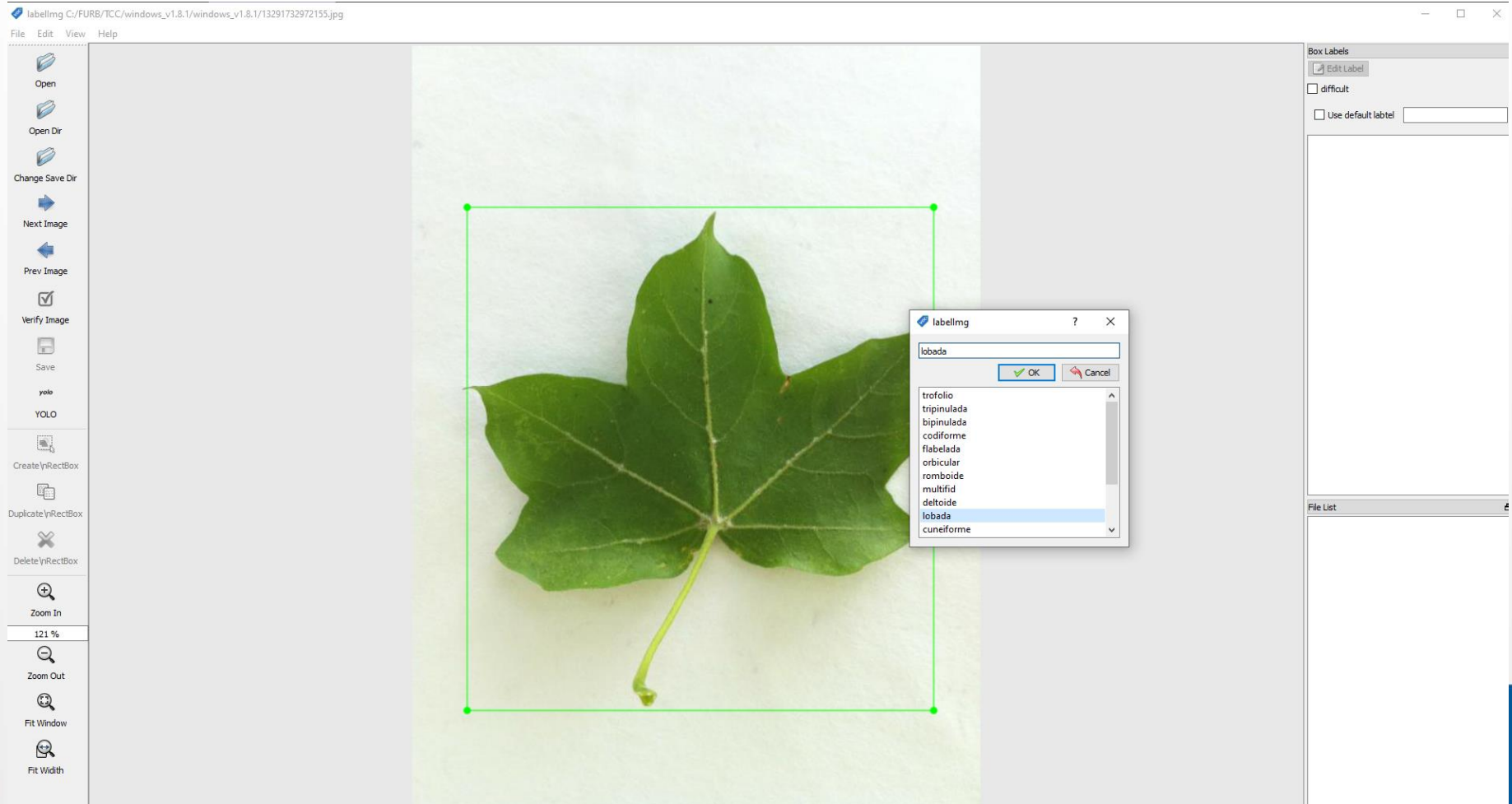
## Fluxograma criação do objeto 3D da folha



# Implementação

## Criação e importação do dataset no Barracuda

- Labellmg



# Implementação

## Rotina de detecção

```
public IEnumerator Detect(Color32[] picture, System.Action<IList<BoundingBox>> callback)
{
    using (var tensor = BarracudaHelper.CreateTensorFromPicuture(picture, IMAGE_SIZE, IMAGE_SIZE))
    {
        var inputs = new Dictionary<string, Tensor>();
        inputs.Add(INPUT_NAME, tensor);
        yield return StartCoroutine(worker.StartManualSchedule(inputs));
        var output_l = worker.PeekOutput(OUTPUT_NAME_L);
        var output_m = worker.PeekOutput(OUTPUT_NAME_M);

        var results_l = ParseOutputs(output_l, MINIMUM_CONFIDENCE, params_l);
        var results_m = ParseOutputs(output_m, MINIMUM_CONFIDENCE, params_m);
        var results = results_l.Concat(results_m).ToList();

        var boxes = FilterBoundingBoxes(results, 1, MINIMUM_CONFIDENCE);
        callback(boxes);
    }
}
```

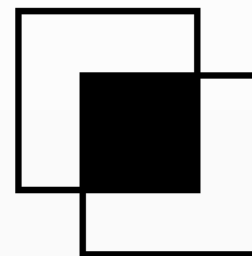


# Implementação

## União de bounding boxes

- Método *intersect over union*.

$$\text{IoU} = \frac{\text{área de sobreposição}}{\text{área de união}}$$



- $\text{IOU} > \text{Trashhold}$  = Combina as bounding boxes

# Implementação

## Criação do objeto 3D

- Inicia após detecção do mesmo objeto por 5 quadros seguidos
- Seleciona-se os pontos X e Y centrais da bounding box com maior grau de confiança.
- Verifica-se se os pontos estão em um plano.

```
private bool VerifyIfPointIntersectsPlanes(float x, float y, TrackableType trackableTypes)
{
    return m_RaycastManager.Raycast(new Vector2(x, y), resultHits, trackableTypes);
}
```

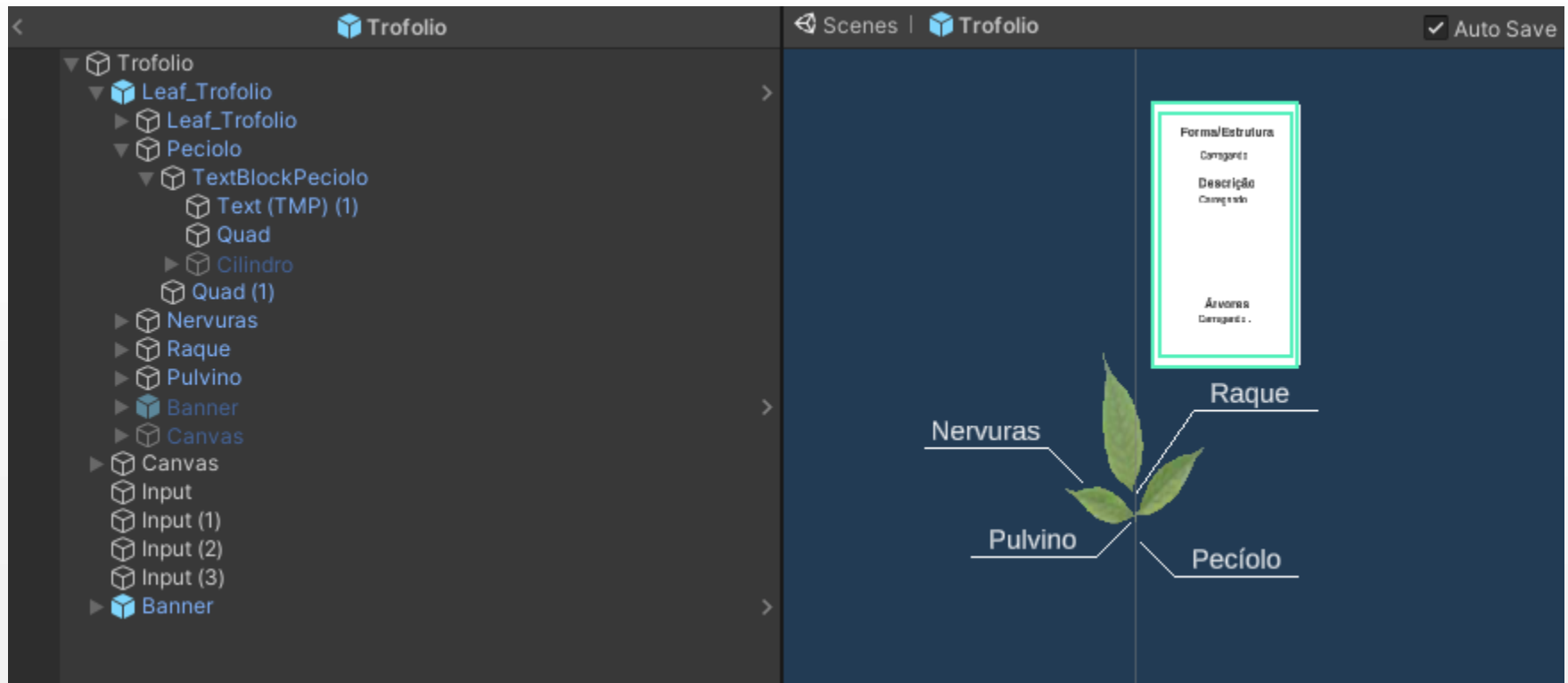
- E se adiciona uma ancora com o *prefab* correspondente a folha detectada.

```
m_AnchorManager.anchorPrefab = prefab[dicPrefab.IndexOf(aRCamera.foundedLeafString)];
anchor = m_AnchorManager.AttachAnchor(plane, hit.pose);
```

# Implementação

## Prefab da folha

- Exemplo de prefab da folha.



# Análise dos Resultados

- Testes de ferramenta
- Testes funcionais
- Teste com acadêmico e especialista
- Submissão do artigo para evento

# Análise dos Resultados

- Testes de ferramenta
  - Realizado testes para garantir que ferramentas fossem capaz de atender ao objetivo.
  - OpenCV for Unity por Barracuda para detecção.
    - Média de 15 fps no dispositivo móvel.
  - OpenCV for Unity por AR Foundation para ancoragem do objeto 3D.
    - Objeto 3D ficava instável, se movendo e girando.

# Análise dos Resultados

- Testes funcionais
  - Dispositivo que suporte tecnologia ARCore.
  - Garantir funcionalidade continua após implementações novas.
  - Testes em diferentes cenários.

Influência da cor do fundo na detecção

Status detecção	Cor do fundo
Detectou	branco
Não detectou	vermelho, verde, amarelo, preto, azul, cinza, marrom, colorido

# Análise dos Resultados

- Testes funcionais

## Influência da iluminação na detecção

Status detecção	Luz natural em ambiente exterior	Sombra de luz natural em ambiente exterior	Luz artificial em ambiente interior	Sombra de luz artificial em ambiente interior
Linear	Detectou	Detectou	Detectou	Detectou
Elíptica	Detectou	Não detectou	Detectou	Não detectou
Trifoliolada	Detectou	Não detectou	Detectou	Não detectou
Palmatífida	Detectou	Detectou	Detectou	Detectou
Pinulada	Detectou	Detectou	Detectou	Não detectou

# Análise dos Resultados

- Teste com acadêmico e especialista
- Formulário com passo a passo da aplicação.
- Ajuda em 3 ou menos etapas de 10.

Usabilidade do aplicativo	5	100%
Cumpriu o objetivo de auxiliar no conhecimento de folhas de plantas com auxílio da Realidade Aumentada	5	100%
Recomendaria o aplicativo para quem deseja aprender mais sobre folhas de plantas	5	100%

- Ótimo forma para o ensino de folhas de uma maneira mais dinâmica
- Ajustes nos nomes e descrições de alguns formatos de folha.



# Análise dos Resultados

- Submissão do artigo para evento
- Envio para o Congresso Brasileiro de Informática na Educação (CBIE) 2021
- Limite 10 páginas
- Envio até 26/07

# Conclusões

- Alcançou o objetivo principal.
- O objetivo específico de utilizar a folha da plantas como marcador foi alcançado, porém necessitando de um fundo controlado para detecção da folha.
- O objetivo específico de analisar a eficácia do aplicativo com usuários da área de Biologia foi alcançado
- Barracuda se mostrou eficiente na detecção de objetos de forma rápida e performática em dispositivos móveis.
- AR Foundation se mostrou eficiente nas rotinas de Realidade Aumentada.
- Contribuições técnicas e sociais.
  - Tutoriais no Apêndice e README

# Sugestões

- Melhorar a detecção das folhas, evitando a necessidade de um fundo controlado atrás das folhas das plantas possibilitando inclusive que a detecção diretamente nas folhas presas às plantas.
- Ampliar o dataset de modo a possibilitar o reconhecimento de um maior número de folhas.
- Realizar não apenas a classificação de formatos das folhas, mas sim da espécie.
- Salvar os resultados do modo quiz e criar um ranking entre os usuários.
- Possibilitar a abertura da galeria de fotos diretamente pelo aplicativo.
- Testar o aplicativo com um maior número de usuários em situação de campo.

# Demonstração