

**Deteção, reconhecimento e alerta de objetos em vídeo**

Licenciatura em Engenharia Informática

Marco António da Silva Ferreira

Leiria, « mês » de 2024



**Deteção, reconhecimento e alerta de objetos em vídeo**

Licenciatura em Engenharia Informática

Válter Santos Pedrosa

Trabalho de Projeto da unidade curricular de Projeto Informático realizado sob a orientação do Professor Doutor Paulo Costa e do Professor Doutor Filipe Neves

Leiria, « mês » de 2024

# Dedicatória

Inserir aqui a dedicatória. Trata-se de um elemento **facultativo**.

Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória.

Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória. Texto da dedicatória.

# Agradecimentos

Inserir aqui os agradecimentos. Trata-se de um elemento **facultativo**.

Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos.

Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos. Texto dos agradecimentos.

# Resumo

Inserir aqui o resumo. Trata-se de um elemento **obrigatório**.

Deve começar sempre numa página ímpar. Se ocupar um número par de páginas (p. ex. 2), deve ajustar-se o texto para que a próxima secção (abstract) se inicie numa página ímpar. O resumo deve acabar com a lista de palavras-chave.

**No resumo deve dar-se nota das principais ideias do trabalho (objetivos e conclusões).**

Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo.

Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo. Texto do resumo.

**Palavras-chave:** deteção, yolo, reconhecimento, IA.

# Abstract

Please insert here the abstract in English. This is a **mandatory** element.

The abstract should always start in an odd page. If the length is a multiple of two, the text should be adjusted in order to the next section start also in an odd page. The abstract should end with a list of keywords.

Please insert here the abstract in English. Please insert here the abstract in English. Please insert here the abstract in English. Please insert here the abstract in English. Please insert here the abstract in English.

Please insert here the abstract in English. Please insert here the abstract in English. Please insert here the abstract in English. Please insert here the abstract in English. Please insert here the abstract in English.

**Keywords:** maximum of 6 words separated by “,”

Índice

Trata-se de um elemento **obrigatório**. Nota: **o índice nunca figura do índice.**

[Dedicatória ii](#_Toc166522130)

[Agradecimentos iii](#_Toc166522131)

[Resumo iv](#_Toc166522132)

[Abstract v](#_Toc166522133)

[Lista de Figuras ix](#_Toc166522134)

[Lista de tabelas x](#_Toc166522135)

[Lista de siglas e acrónimos xi](#_Toc166522136)

[1. Introdução 1](#_Toc166522137)

[2. Título do capítulo 2](#_Toc166522138)

[2.1. Título da secção 4](#_Toc166522139)

[2.2. Título da secção 4](#_Toc166522140)

[2.2.1. Título da subsecção 5](#_Toc166522141)

[2.2.2. Título da subsecção 5](#_Toc166522142)

[2.2.3. Título da subsecção 5](#_Toc166522143)

[2.3. Título da secção 5](#_Toc166522144)

[3. Metodologias e tecnologias 6](#_Toc166522145)

[3.1. Metodologia de trabalho 6](#_Toc166522146)

[3.1.1. Método tradicional 6](#_Toc166522147)

[3.1.2. Metodologias ágeis 6](#_Toc166522148)

[3.1.2.1. Scrum 8](#_Toc166522149)

[3.1.2.2. XP 9](#_Toc166522150)

[3.1.2.3. Kanban 10](#_Toc166522151)

[3.1.2.4. Scrumban 11](#_Toc166522152)

[3.2. Metodologia de trabalho escolhida 11](#_Toc166522153)

[4. Testes 13](#_Toc166522154)

[4.1. Teste de margem ideal para definir posição do objeto 13](#_Toc166522155)

[4.1.1. Introdução 13](#_Toc166522156)

[4.1.2. Condições do teste 13](#_Toc166522157)

[4.1.3. Resultados obtidos: 14](#_Toc166522158)

[4.1.3.1. Class “backpack” 14](#_Toc166522159)

[4.1.3.2. Class “sports ball” 16](#_Toc166522160)

[4.1.3.3. Class “Person” 17](#_Toc166522161)

[4.1.4. Conclusão: 18](#_Toc166522162)

[4.2. Teste da arquitetura para recolher frames de dispositivos 19](#_Toc166522163)

[4.2.1. Introdução 19](#_Toc166522164)

[4.2.1.1. Grab & Retrieve 19](#_Toc166522165)

[4.2.1.2. Parallel 19](#_Toc166522166)

[4.2.1.3. Open & Close 19](#_Toc166522167)

[4.2.1.4. Open & Close threading 19](#_Toc166522168)

[4.2.2. Condições de teste 20](#_Toc166522169)

[4.2.3. Resultados obtidos 20](#_Toc166522170)

[4.2.3.1. Grab & Retrieve 20](#_Toc166522171)

[4.2.3.2. Parallel 20](#_Toc166522172)

[4.2.3.2.1. Parallel 5 sec 21](#_Toc166522173)

[4.2.3.2.2. Parallel 10 sec 22](#_Toc166522174)

[4.2.3.3. Open & Close 22](#_Toc166522175)

[4.2.3.4. Open & Close threading 23](#_Toc166522176)

[4.2.4. Discussão dos Resultados 23](#_Toc166522177)

[4.2.5. Conclusão 24](#_Toc166522178)

[5. Conclusões ou Conclusão 25](#_Toc166522179)

[Bibliografia ou Referências Bibliográficas 26](#_Toc166522180)

[Anexos 27](#_Toc166522181)

[Glossário 28](#_Toc166522182)

# Lista de Figuras

Elemento a figurar, **quando aplicável**.

[Figura 2.1 - Texto ilustrativo da figura 1. 3](#_Toc164784383)

[Figura 2.4 - Texto ilustrativo da figura 2. 3](#_Toc164784384)

[Figura 3.1 - Gráfico cantos backpack a 5M 7](#_Toc164784385)

[Figura 3.2 - Gráfico cantos backpack a 10M 8](#_Toc164784386)

[Figura 3.3 - Gráfico cantos backpack a 15M 8](#_Toc164784387)

[Figura 3.4 - Gráfico cantos bola a 5M 9](#_Toc164784388)

[Figura 3.5 - Gráfico cantos bola a 10M 9](#_Toc164784389)

[Figura 3.6 - Gráfico cantos bola a 12M 10](#_Toc164784390)

[Figura 3.7 - Gráfico cantos person a 5M 10](#_Toc164784391)

[Figura 3.8 - Gráfico cantos person a 35M 11](#_Toc164784392)

[Figura 3.9 - Gráfico cantos person a 70M 11](#_Toc164784393)

[Figura 3.10 - Gráfico performance parallel 13](#_Toc164784394)

[Figura 3.11 - Gráfico performance parallel 5 segundos 14](#_Toc164784395)

[Figura 3.12 - Gráfico performance parallel 10 segundos 15](#_Toc164784396)

[Figura 3.13 - Gráfico performance open & close 15](#_Toc164784397)

[Figura 3.14 - Gráfico performance open & close threading 16](#_Toc164784398)

# Lista de tabelas

Elemento a figurar, **quando aplicável**.

[Tabela 1.1 - Texto ilustrativo da tabela 1. 3](#_Toc92389035)

# Lista de siglas e acrónimos

Elemento a figurar, **quando aplicável**.

|  |  |
| --- | --- |
| ESTG | Escola Superior de Tecnologia e Gestão |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Cuidados na elaboração da lista de siglas e acrónimos:

* Ordenação alfabética;
* Apenas as que sejam relevantes para a leitura do texto.

Adicionar mais entradas à tabela, caso seja necessário (a tabela não tem contornos, mas está no texto).

# Introdução

Introdução deve conter resumidamente os seguintes elementos:

* O objeto do trabalho (o tema);
* A justificação ou a pertinência do tema;
* Os objetivos do trabalho (gerais e específicos, perguntas a responder ou hipóteses a testar);
* Os métodos e as técnicas utilizados;
* Estrutura do trabalho.

# Título do capítulo

Aqui, inicia-se o desenvolvimento dos trabalhos e deve indicar-se o título do capítulo primeiro.

O desenvolvimento do trabalho deve ser adequado à natureza da unidade curricular (dissertação/trabalho de projeto/relatório de estágio) e deve seguir as práticas mais disseminadas na área em causa.

Estrutura: pode ter, por exemplo, capítulos, secções e subsecções.

Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento.

Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento.

De seguida, apresenta-se um exemplo de como as imagens devem ser colocadas no texto:

* Centradas;
* Numeradas;
* Referenciadas no texto;
* Sem contornos;
* Eventualmente, colocar a(s) figura(s) numa tabela para melhorar a formatação;
* Usar sempre o estilo “caption” para o texto das legendas;
* Identificação da figura em “negrito” (o estilo trata desta parte). Texto da legenda não deve estar em negrito.

|  |
| --- |
| Figura 2.1 - Texto ilustrativo da figura 1. |

De seguida apresenta-se a figura 2.

|  |
| --- |
| Figura 2.4 - Texto ilustrativo da figura 2. |

Cuidados com a inserção de tabelas:

* Centradas;
* Numeradas;
* Referenciadas no texto;
* Usar sempre o estilo “caption” para o texto das legendas;
* Identificação da tabela em “negrito” (o estilo trata desta parte). Texto da legenda não deve estar em negrito.

Tabela 1.1 - Texto ilustrativo da tabela 1.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Continuação do texto do desenvolvimento.

Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento.

Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento.

## Título da secção

A figurar no caso de o capítulo ter várias secções. Nota: apenas devemos criar secções e subsecções quando existem mais do que uma.

Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento.

Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento.

## Título da secção

Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento.

Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento.

### Título da subsecção

Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento.

### Título da subsecção

Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento.

### Título da subsecção

Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento. Texto do desenvolvimento.

## Título da secção

# Metodologias e tecnologias

## Metodologia de trabalho

Quando se fala em desenvolvimento de software, existem duas maneiras de abordar a sua gestão e planeamento de modo a tornar a gestão de projeto mais eficiente e eficaz. Por um lado, temos as metodologias tradicionais, como o modelo cascata, que apostam numa abordagem sequencial e planeamento detalhado desde o início do projeto. Enquanto, em contraste, as metodologias ágeis dão prioridade à flexibilidade, colaboração e adaptação contínua ao longo do projeto.

### Método tradicional

A abordagem tradicional de gestão de projetos, também conhecida como método cascata, segue um processo linear e sequencial onde só se passa à fase seguinte quando a anterior é completada. É uma abordagem estruturada e orientada a planos com foco no planeamento completo e documentação antecipada. Todos os projetos, independentemente da sua natureza, seguem o mesmo ciclo de vida que incluem as fases de viabilidade, planeamento, design, construção, teste, produção e suporte. O projeto é planeado todo no início e não há qualquer mudança de requisitos, uma vez que nesta abordagem assume-se que os requisitos são fixos e o tempo e custo variáveis. Como tal, não é adequado a projetos de grande escala em que os requisitos para todo o projeto não se conseguem definir numa fase inicial.

### Metodologias ágeis

Ao contrário do método anterior, o foco das metodologias ágeis é o trabalho em equipa, colaboração com o cliente e flexibilidade. É uma abordagem iterativa que se concentra mais na incorporação do cliente e lançamentos contínuos em cada iteração do desenvolvimento do projeto. O conceito chave desta abordagem é que aprofunda a evolução das mudanças e esforço colaborativo para obter resultados em vez de um processo predefinido. O planeamento adaptativo é a principal característica das metodologias ágeis que convence muitos gestores de projeto.

Na seguinte tabela podemos ver a comparação de alguns aspetos comuns a ambas as abordagens.

|  |  |  |
| --- | --- | --- |
| Parâmetros | Metodologias Ágeis | Gestão de projeto tradicional |
| Flexibilidade e adaptabilidade | Ênfase na flexibilidade e adaptabilidade | Ênfase no planeamento e previsibilidade |
| Requisitos do projeto | Prioriza a satisfação do cliente e software funcional | Prioriza o seguimento de um plano rigoroso e atender aos requisitos do projeto |
| Comunicação | Encoraja comunicação e colaboração cara-a-cara | Confia na documentação escrita. Comunicação formal |
| Fases de projeto e planificação | Sprints curtos para planeamento e entrega. Permite alterações e ajustes ao longo do projeto | Fases de projeto mais longas para planeamento e entrega. Tem um plano rígido que e difícil alterar |
| Funcionalidade de equipa | Equipas multifuncionais | Equipas com funções divididas |
| Organização | Equipas auto-organizadas | Equipas organizadas hierarquicamente |
| Estrutura organizacional | Iterativa | Linear |
| Escala de projeto | Grande escala | Escala pequena e média |
| Modelo de desenvolvimento | Modelo de entrega evolutivo | Modelo de ciclo de vida |
| Requisitos do utilizador | Inputs interativos | Claramente definidos antes de implementação |
| Envolvimento do cliente | Alto | Baixo |

Fonte: <https://www.knowledgehut.com/blog/agile/agile-project-management-vs-traditional-project-management> com ajustes

Optamos por seguir uma metodologia ágil por não saber de antemão os requisitos que vai ter a nossa aplicação, uma vez que esta abordagem permite-nos desenvolver a aplicação de maneira mais flexível e colaborativa. Vamos então analisar as principais metodologias ágeis e a forma como abordam a gestão dos projetos.

### Scrum

O scrum é a framework mais utilizada de todas as metodologias ágeis e caracteriza-se por etapas ou ciclos de desenvolvimento, conhecidos como sprints e pela maximização do tempo de desenvolvimento de um produto de sofware em direção ao Product Goal.

Uma equipa de Scrum é composta por 3 elementos com papeis bem definidos, o Product Owner, o Scrum Master e a equipa de desenvolvimento. O PO é o elemento que liga o cliente à equipa de desenvolvimento, assumindo a função garantir que a visão do cliente é entendida pela equipa de desenvolvimento e definindo os critérios de sucesso. É este que define o que deve ser feito, e em que ordem deve ser feito, atribuindo valor aos requisitos. O SM é referido como o facilitador do projeto uma vez que este certifica que as boas práticas da metodologia Scrum são seguidas, verifica relações saudáveis entre o PO e os programadores e elimina quaisquer obstáculos que impeçam uma colaboração produtiva. Por fim, a equipa de desenvolvimento, como o nome indica, é o conjunto de membros que trabalha em conjunto para criar, estar e lançar versões incrementais do produto final.

Além da equipa Scrum, esta metodologia conta com os seguintes eventos Scrum: Daily Scrum, Sprint, Sprint planning, Sprint review e Sprint retrospective.

O Daily Scrum consiste numa curta reunião diária que ocorre todos os dias no mesmo sítio à mesma hora. Nesta reunião a equipa avalia o trabalho realizado no dia anterior e planeia o que será feito nas próximas 24 horas.

A Sprint representa o período em que o trabalho deve estar concluído, não maior que um mês. Uma nova sprint começa quando a anterior acaba.

A Sprint planning meeting marca o início de uma nova sprint. Nesta reunião, toda a equipa Scrum participa e define objetivos, sendo que no final pelo menos um incremento de software utilizável deve ser produzido.

No fim de cada sprint, acontece a sprint review em que a equipa scrum mostra o trabalho realizado ao PO e stakeholders.

Por fim, a sprint retrospective, dá lugar à equipa para discutir o que correu bem e o que pode ser melhorado em relação à última sprint com o objetivo de atingir uma melhoria contínua.

No Scrum existem ainda 2 backlogs. O Product backlog que contém a lista das tarefas para sprints futuras e s Sprint Backlog que inclui as tarefas a realizar na sprint atual.

### XP

XP é outra metodologia ágil bastante presente no mundo dos projetos de desenvolvimento de software. Tem como objetivo criar sistemas de alta qualidade, assente nos princípios de proximidade com o cliente, testes constantes e ciclos de desenvolvimento curtos. Os 5 componentes básicos do XP são: comunicação, simplicidade, feedback, respeito e coragem. Esta abordagem aposta na velocidade e simplicidade com ciclos curtos de desenvolvimento e menos documentação.

No XP o cliente deve participar ativamente do processo de desenvolvimento. Tudo precisa da comunicação com o cliente. Este deve receber o melhor resultado possível a cada sprint, ver o progresso no sistema e ser informado de mudanças de planos para que possa dar a sua opinião.

Outra prática fundamental no XP é o pair programming. O pair programming consiste numa programação a pares num único computador em que um tem o papel de escrever o código enquanto o outro está sempre ao lado a rever. Desta maneira, erros e detalhes que possam passar despercebidos a um programador, o outro consegue cobri-los diminuído a probabilidade de falhas.

Para perceber mais facilmente o que difere o XP do Scrum podemos consultar a seguinte tabela:

|  |  |
| --- | --- |
| Scrum | Extreme Programming |
| Trabalho em iterações (sprints) podem ir até 1 mês | Trabalho em iterações de apenas 1-2 semanas |
| Modelos Scrum não permitem mudanças no seu cronograma ou diretrizes. | Permite mudanças nos cronogramas definidos |
| Ênfase na auto-organização | Ênfase em fortes práticas de engenharia |
| Equipa determina a sequência na qual o produto é desenvolvido | Equipa segue uma ordem de prioridade pré-determinada |
| A framework Scrum não está completamente descrita. Para adotá-la é necessário completá-la com métodos de trabalho, como XP ou Kanban | Pode ser aplicado diretamente a uma equipa. O XP é também conhecido pelas suas características “Ready-to-apply” |
| Não enfatiza as práticas de engenharia de software que os programadores devem usar | Enfatiza as técnicas de programação que os programadores devem seguir para garantir um melhor resultado |
| Requer que os programadores estejam conscientes da adoção de métodos de engenharia para garantir um melhor progresso ou qualidade | É muito rígido na adoção de métodos de engenharia, como pair programming, design simples, reestruturação, para garantir um melhor progresso ou qualidade. |
| O Product Owner define a prioridade das tarefas de acordo com os requisitos | O cliente define a prioridade das tarefas e analisa os lançamentos. |
| Existe flexibilidade para a equipa ajustar a prioridade das tarefas se necessário | A equipa não pode alterar a prioridade dos requisitos definidos pelo cliente |
| Valores: - Abertura  - Foco  - Compromisso | Valores: - Comunicação  - Simplicidade  - Feedback |
| Menor envolvimento do cliente no projeto | Maior envolvimento do cliente no projeto |

Fonte: https://www.geeksforgeeks.org/difference-between-scrum-and-xp/

### Kanban

A metodologia kanban visa uma melhoria contínua, flexibilidade no gerenciamento de tarefas, e fluxo de trabalho aprimorado. Por ser uma abordagem ilustrativa, o progresso de todo o projeto é facilmente compreendido em poucos segundos.

O Kanban, ao contrário do scrum, não possui um prazo definido para quando as tarefas devem ser concluídas. Em vez disso, esta metodologia é gerida pela prioridade dos requisitos num quadro Kanban. Este quadro é dividido em três estados para uma tarefa: pendente (to do), em desenvolvimento (doing) e concluída (done). As tarefas/requisitos são representadas por cartões visuais que transitam entre os estados até serem finalizadas. Assim, a equipa consegue ter uma noção visual do desenvolvimento do projeto.

### Scrumban

O Scrumban, como o nome sugere, é uma metodologia ágil que combina elementos do Scrum e Kanban. Esta abordagem envolve aplicar os princípios de visualização do fluxo de trabalho e processos flexíveis do Kanban a uma equipa da Framework Scrum. Porém, este princípio, remove alguns dos aspetos mais rígidos do Scrum e deixa as equipas formar uma abordagem personalizada para o desenvolvimento do projeto.

O Scrunban adota um quadro similar ao quadro Kanban na medida em que é a ferramenta principal de fluxo de trabalho com a diferença que permite adicionar as colunas necessárias ao quadro para marcar a fase de progresso para cada requisito.

O Scrum, define limites de tempo e tarefas por cada sprint, o contrário acontece no Kanban que se foca no fluxo de trabalho contínuo. No Scrumban é estabelecido um limite para a quantidade de trabalho que a equipa pode realizar num momento e este limite é o total de cartões no quadro.

Enquanto no Scrum cada tarefa é atribuída a um membro específico da equipa, no Scrumban o foco é estabelecer a ordem de prioridade das tarefas no quadro e cabe à equipa distribuir as tarefas a realizar pelos diversos membros.

Embora esta abordagem não tenha todas as reuniões típicas da estrutura Scrum, no Scrumban pode haver reuniões curtas durante as sprints para a equipa discutir planos e desafios encontrados.

## Metodologia de trabalho escolhida

Das metodologias ágeis apresentadas, excluímos já de partida o XP por não termos o cliente sempre presente no desenvolvimento da aplicação nem condições para aplicar o pair programming que esta abordagem requere. Das restantes, Scrum, Kanban e Scrumban decidimos escolher a última (Scrumbam) por oferecer o melhor de cada uma das metodologias em que esta se baseia enquanto descarta os aspetos menos favoráveis das mesmas. Deste modo conseguimos aproveitar não só a flexibilidade que o Kanban nos permite, mas também os princípios de planeamento Scrum, permitindo-nos adaptar nosso processo de desenvolvimento de acordo com as necessidades do mesmo.

Para aplicar a metodologia ágil Scrumban, realizamos uma reunião semanal com o Product Owner e Scrum Master, interpretados pelos professores orientadores. Nesta reunião discutimos o trabalho desenvolvido na última sprint que acaba nessa mesma reunião e determinamos o que deve ser feito para a próxima. Utilizamos a framework Jira Software da Atlassian que nos permite criar Issues que representam as tarefas/requisitos que a nossa aplicação tem ou vai ter. Aqui também conseguimos criar o quadro kanban que nos proporciona uma noção visual do desenvolvimento do trabalho e onde podemos colocar os Issues previamente criados. Conseguimos ainda criar e terminar as várias sprints ao longo do processo de desenvolvimento.

# Testes

## Teste de margem ideal para definir posição do objeto

### Introdução

O YOLO, após detetar um objeto, desenha uma bounding box que delimita o objeto detetado. Mesmo que o input não se altere, no caso deste projeto em que o input são camaras de vídeo, significa objeto na mesma posição, esta box pode variar ligeiramente devido a vários fatores. Estes fatores compreendem: variabilidade de rede, anchor boxes ou Non-Maximum Supression (NMS). Variabilidade de rede porque, a rede YOLO, como qualquer rede neuronal de deep lerning, tem variabilidade inerente nas suas previsões, devido à variação de valores inicias de peso, processos de otimização e descendente estocástico de gradiente (Stochastic Gradient Descent - SGD). Entende-se por Anchor boxes as formas e tamanhos predefinidos pelo algoritmo usadas para prever as bouding boxes. Se as anchor boxes não corresponderem exatamente à forma e tamanho dos objetos na imagem, pode também levar a variações nas bounding boxes. Por fim, NMS é uma técnica de pós-processamento usada pelo YOLO para filtrar as bounding boxes previstas. Neste processo é escolhida a box com maior grau de confiança e suprime-se as boxes sobrepostas com menor grau de confiança. O limite, isto é, o valor de confiança mínimo necessário para que uma bounding box seja considerada válida, usado nesta técnica pode afetar que bounding boxes são selecionadas, resulta em variações nas previsões finais.

Posto isto, submetemo-nos à criação de um cenário de testes para definir qual a margem ideal a definir para compensar a variação da posição da bounding box em objetos parados. Deste modo garantimos uma deteção de movimento mais rigorosa, e consequentemente alertas de segurança mais precisos e confiáveis.

### Condições do teste

Para avaliar qual a margem que confere a melhor exatidão dos resultados, foram simulados 17 cenários de testes variando os objetos a detetar e a distância a que se encontram do dispositivo de gravação. A posição da camara manteve-se a mesma durante todo o procedimento bem como o cenário onde se encontraram os objetos a detetar. Este cenário consistia num dia de sol, céu limpo com presença de sombra onde os objetos foram detetados.

Os cenários de teste foram os seguintes:

* Mochila (backpack) a 5,10 e 15 metros
* Bola (sports ball) a 5,10 e 12 metros
* Pessoa (person) a 5,10,15,20,25,30,35,40,50,60 e 70 metros

### Resultados obtidos:

Os gráficos seguintes representam, à esquerda, a distância entre os cantos inferior esquerdo (canto 1) do frame anterior e o atual e a distância do canto superior direito (canto 2) entre o frame anterior e atual à direita.

### Class “backpack”

Uma imagem com texto, file, diagrama, Gráfico

Descrição gerada automaticamente

Figura 3.1 - Gráfico cantos backpack a 5M

Uma imagem com texto, file, diagrama, Gráfico

Descrição gerada automaticamente

Figura 3.2 - Gráfico cantos backpack a 10M

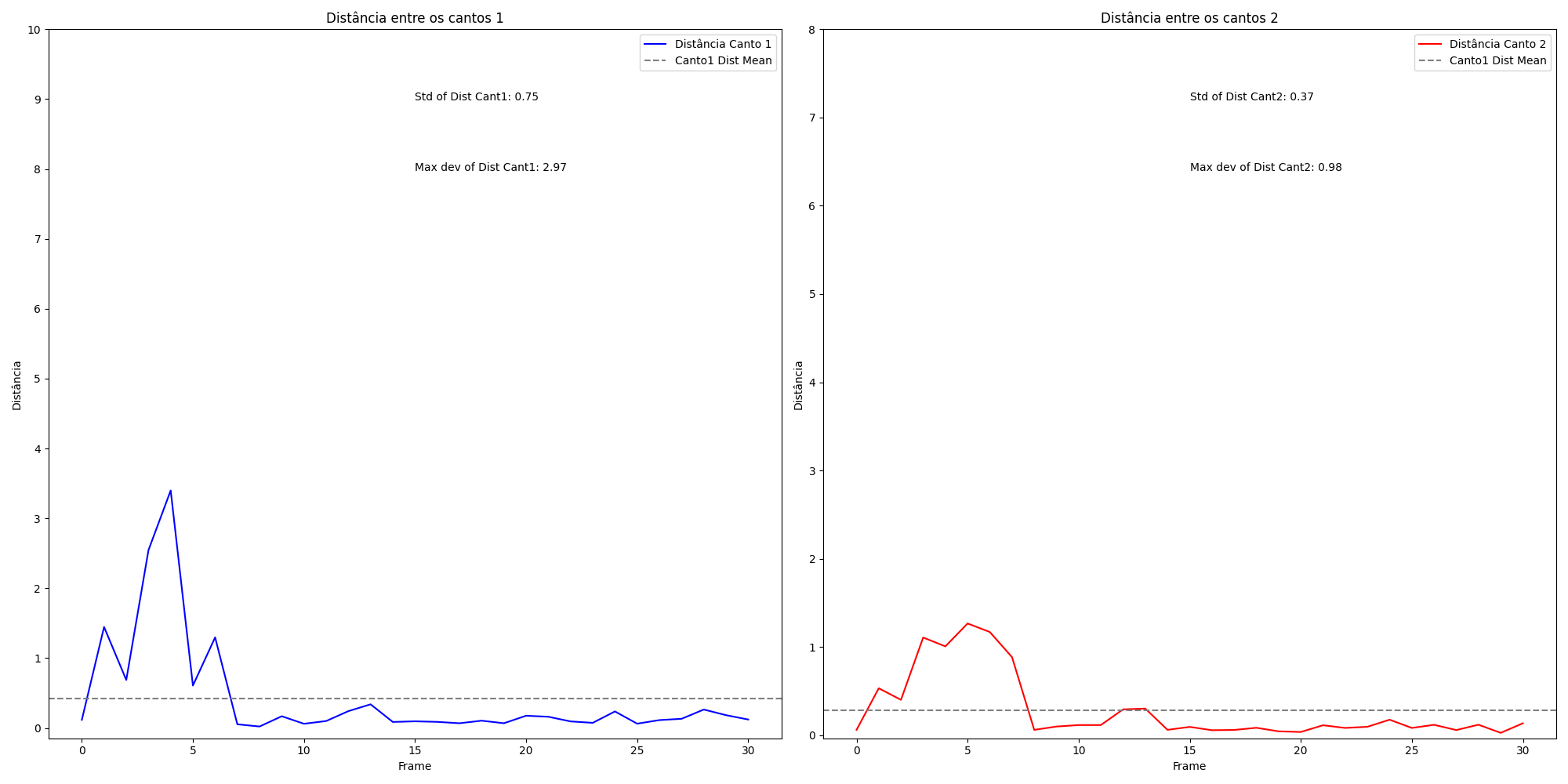


Figura 3.3 - Gráfico cantos backpack a 15M

### Class “sports ball”

Uma imagem com diagrama, file, texto, Gráfico

Descrição gerada automaticamente

Figura 3.4 - Gráfico cantos bola a 5M

Uma imagem com texto, file, Gráfico, diagrama

Descrição gerada automaticamente

Figura 3.5 - Gráfico cantos bola a 10M

Uma imagem com texto, file, diagrama, Gráfico

Descrição gerada automaticamente

Figura 3.6 - Gráfico cantos bola a 12M

### Class “Person”

Uma imagem com file, texto, diagrama, Gráfico

Descrição gerada automaticamente

Figura 3.7 - Gráfico cantos person a 5M

Uma imagem com file, diagrama, texto, Gráfico

Descrição gerada automaticamente

Figura 3.8 - Gráfico cantos person a 35M

Uma imagem com texto, file, diagrama, Gráfico

Descrição gerada automaticamente

Figura 3.9 - Gráfico cantos person a 70M

### Conclusão:

Após teste dos vários cenários, seguidos da análise e discussão dos resultados obtidos, consideramos 4 px como margem ideal a aplicar para desprezar as possíveis variações das bounding boxes geradas pelo modelo de deteção usado em objetos parados. Isto deve-se ao facto dos maiores desvios obtidos serem na ordem dos 3px, sendo o máximo deles 3.15 no caso de teste backpack a 5 metros. Sendo 4 o próximo inteiro, 4px cobre todos os desvios que obtivemos na fase de testes.

## Teste da arquitetura para recolher frames de dispositivos

### Introdução

Serve o presente teste para testar qual a arquitetura que melhor se adequa à nossa aplicação e aos seus requisitos. Para isso temos 4 modelos para teste sendo eles:

### Grab & Retrieve

Neste modelo, após inicializarmos as câmaras com o videCapture de cada dispositivo, criamos uma thread de captura por dispositivo. Esta thread apenas agarra (grab) continuamente frames do dispositivo para posterior descodificação e análise do frame. Na thread principal, de X em X segundos (definido pelo utilizador) chamamos a função retrieveFrames que percorre o vetor das câmaras previamente inicializadas e recolhe (retrieve) o último frame agarrado (grabbed) pela thread de captura do dispositivo associado descodificando-o. De seguida enviamos cada frame para a função YOLO para o processar.

### Parallel

Nesta arquitetura, para cada dispositivo de captura selecionado, é criada uma thread que lê, descodifica e envia o frame ao YOLO para detetar objetos na imagem.

### Open & Close

Neste padrão (ausente de threads) cada dispositivo de captura é inicializado, captura o frame, processa-o para deteção através do YOLO e fecha o dispositivo, seguindo para o próximo continuamente, N vezes, sendo N o número de dispositivos selecionado.

### Open & Close threading

Neste modelo, inicializa-se os dispositivos e cria-se uma thread por dispositivo para ler um frame. Após todas as threads terem lido um frame, cria-se uma thread por frame para o processar no YOLO. Finalmente, depois das threads terminarem o processamento, fecham-se os dispositivos inicializados no início.

### Condições de teste

Para comparar os vários modelos de implementação, utilizamos a biblioteca “psutil” do python que nos permite avaliar a % de CPU usage bem como a % de RAM usage durante todo o processo. Para avaliar os cenários, utilizamos as mesmas 5 câmaras, das quais 2 locais e 3 por IP e definimos 40 iterações para avaliar o tempo de execução.

### Resultados obtidos

### Grab & Retrieve

Neste padrão não foi possível concluir os testes porque ao usar câmaras IP em vez de locais no videoCapture, o retrieve não consegue descodificar (retrieve) o frame agarrado (grabbed) anteriormente, pelo que excluímos das opções.

### Parallel

Uma imagem com texto, file, diagrama, captura de ecrã

Descrição gerada automaticamente

Figura 3.10 - Gráfico performance parallel

Ao analisar a percentagem elevada de CPU usage nesta arquitetura, acrescentamos dois novos casos de teste para tentar diminuir a % de CPU usage sacrificando tempo de execução. Para isso, adicionamos um time.sleep de 5 e 10 segundos.

### Uma imagem com texto, file, captura de ecrã, diagrama Descrição gerada automaticamenteParallel 5 sec

Figura 3.11 - Gráfico performance parallel 5 segundos

### Uma imagem com texto, file, diagrama, Gráfico Descrição gerada automaticamenteParallel 10 sec

Figura 3.12 - Gráfico performance parallel 10 segundos

### Uma imagem com texto, file, Gráfico, diagrama Descrição gerada automaticamenteOpen & Close

Figura 3.13 - Gráfico performance open & close

### Open & Close threading

Uma imagem com texto, file, diagrama, Gráfico

Descrição gerada automaticamente

Figura 3.14 - Gráfico performance open & close threading

### Discussão dos Resultados

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Método | Tempo execução | Máx – Init CPU % | Máx – Init RAM % | Mean – Init CPU % | Mean – Init RAM % |
| Parallel | 1m 44s | 100 | 10,80 | 85,88 | 10,03 |
| Parallel 5 sec | 5m 24s | 100 | 8,50 | 34,55 | 7,67 |
| Parallel 10 sec | 7m 39s | 35,20 | 8,30 | 20,46 | 4,11 |
| Open & Close | 17m 58s | 28,30 | 19,50 | 9,86 | 13,53 |
| Open & Close threading | 10m 25s | 32,40 | 13 | 18,14 | 8,77 |

Através dos dados retirados podemos criar a seguinte tabela:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Menor tempo execução | Maior tempo  Execução | Menor CPU usage | Maior CPU usage | Menor RAM usage | Maior RAM usage |
| Parallel | Open & Close | Open & Close | Parallel | Parallel 10 sec | Open & Close threading |

Neste projeto, priorizamos o menor custo computacional em relação ao tempo de execução. Dito esto, excluímos o modelo Parallel pois utiliza o esforço máximo do CPU e o Open & Close threading por representar o maior esforço sobre a RAM.

Entre o Parallel 5 sec e o Parallel 10 sec, o último representa o melhor comparativamente porque demora 1,42 vezes mais para concluir o teste mas utiliza menos 1,69 vezes o CPU e 1,87 vezes menos a RAM.

O Open & Close é o que apresenta menor custo para o processador (9,86%) mas demora demasiado tempo em relação ao Parallel 10 sec que utiliza 20,46% do processador, o que representa 2,08 vezes mais que o anterior, mas acaba o teste 2,35 vezes mais cedo, o que representa um rácio maior de tempo / CPU usage, pelo que também exclui o Open & Close em comparação ao Parallel 10 sec.

### Conclusão

Após análise e discussão dos resultados obtidos, escolhemos a arquitetura Parallel 10 sec como a mais adequada para a nossa aplicação por ser o mais equilibrado quando aplicamos os nossos critérios.

# Conclusões ou Conclusão

Inserir aqui as conclusões ou conclusão. Trata-se de um elemento **obrigatório**.

A conclusão:

* Deve ser sucinta;
* Não deve conter informações ou ideias novas;
* Deve permitir concluir se se atingiram os objetivos enunciados na introdução.

Texto da conclusão. Texto da conclusão. Texto da conclusão. Texto da conclusão. Texto da conclusão. Texto da conclusão. Texto da conclusão. Texto da conclusão. Texto da conclusão. Texto da conclusão.

Texto da conclusão. Texto da conclusão. Texto da conclusão. Texto da conclusão. Texto da conclusão. Texto da conclusão. Texto da conclusão. Texto da conclusão. Texto da conclusão. Texto da conclusão.

# Bibliografia ou Referências Bibliográficas

Inserir aqui a bibliografia ou referências bibliográficas. Trata-se de um elemento **obrigatório**.

Notas: o sistema a adotar para a apresentação das referências bibliográficas e as suas citações deve:

* Respeitar uma norma estabelecida;
* Seguir as práticas mais disseminadas na área em causa;
* Ser empregue de modo uniforme em todo o documento.

Bibliografia – quando se coloca toda a bibliografia consultada;

Referências bibliográficas – quando se faz referência apenas à bibliografia citada.

# Anexos

Elemento a figurar, **quando aplicável**.

# Glossário

Elemento a figurar, **quando aplicável**.