

[Título do Projeto]

*Imagem ou personalização gráfica do Relatório*

[Identificação dos formandos] [Turma]

[Identificação da(s) UFCD] [Formador(es)]

[Data]

Índice

[**1.** **Introdução** 2](#_Toc148112829)

[**2.** **Planeamento** 2](#_Toc148112830)

[**2.1.** **Metodologia de Desenvolvimento** 2](#_Toc148112831)

[**3.** **Requisitos** 2](#_Toc148112832)

[**3.1.** **Requisitos Funcionais** 2](#_Toc148112833)

[**3.2.** **Requisitos Não Funcionais** 3](#_Toc148112834)

[**3.3.** **Priorização de Requisitos** 3](#_Toc148112835)

[**4.** **Desenho do Sistema e Arquitetura** 3](#_Toc148112836)

[**4.1.** **Desenho do Sistema** 3](#_Toc148112837)

[**4.2.** **Arquitetura e Tecnologia** 3](#_Toc148112838)

[**4.3.** **Diagramas e Modelos** 3](#_Toc148112839)

[**5.** **Implementação** 3](#_Toc148112840)

[**5.1.** **Ambiente de Desenvolvimento** 4](#_Toc148112841)

[**5.2.** **Desenvolvimento** 4](#_Toc148112842)

[**5.3.** **Controle de Versão** 4](#_Toc148112843)

[**6.** **Testes** 4](#_Toc148112844)

[**7.** **Conclusões** 4](#_Toc148112845)

[**8.** **Fontes** 4](#_Toc148112846)

[**Anexos** 5](#_Toc148112847)

1. **Introdução**

Na introdução devem constar os objetivos do trabalho e a sua motivação. A introdução deverá descrever, sucintamente, o modo como o relatório está organizado nas suas diversas partes e a articulação entre elas segundo uma ideia condutora. Descreva, de forma sucinta, o problema em estudo e enuncie os principais métodos que são utilizados no relatório. Seja especialmente cuidadoso na utilização de termos técnicos ou referência a palavras-chave relativas à temática abordada.

1. **Planeamento**

O planeamento é uma das partes mais cruciais de um projeto de software. Estabelece a base e o caminho para todo o projeto, garantindo que haja uma estrutura sólida e um plano bem elaborado para guiar todas as etapas do desenvolvimento.

# **Metodologia de Desenvolvimento**

Neste segmento, a metodologia de desenvolvimento é escolhida e descrita. Pode ser Agile, Waterfall, Scrum, entre outras. A metodologia orientará a abordagem de desenvolvimento e a organização do grupo. No fundo deve ser descrito como o grupo se pretende organizar (ou organizou) durante o projeto.

**2.2 Cronograma**

O cronograma do projeto deve ser criado no seu início e ser detalhado. Inclui todas as fases do desenvolvimento, os marcos (*milestones*), as entregas e os prazos. O cronograma serve como um guia temporal para a execução do projeto, assegurando que as metas sejam atingidas no tempo previsto.

1. **Requisitos**

O capítulo de requisitos é fundamental para definir e documentar as necessidades e expectativas dos *stakeholders* (caso existam, podendo ser os próprios elementos do grupo). Este capítulo serve como uma base para o desenvolvimento, design, teste e implementação do software, garantindo que todas as partes envolvidas tenham um entendimento claro do que o software deve realizar. Neste capítulo, além de se identificarem os trabalhos de levantamento e análise de requisitos (trabalhos efetuados, como por exemplo entrevistas, análise de sistemas de referência, etc.), os requisitos são categorizados, detalhados e priorizados para orientar as fases subsequentes do projeto.

# **Requisitos Funcionais**

Nesta seção, são descritos os requisitos funcionais do software, ou seja, as funções específicas e as operações que o software deve realizar. Pode incluir processos (Diagramas UML), características, funcionalidades e especificações detalhadas que descrevem o que o sistema deve fazer para satisfazer as necessidades dos utilizadores finais.

# **Requisitos não Funcionais**

Os requisitos não funcionais são as características qualitativas do sistema, como desempenho, segurança, usabilidade, compatibilidade, etc. No fundo definem "como" o sistema realiza as suas funções, garantindo que o software atenda a padrões de qualidade e operacionalidade específicos.

# **Priorização de Requisitos**

A priorização é essencial para que os grupos possam dar prioridade aos requisitos mais críticos e importantes. Aqui, os requisitos são classificados com base em sua importância e urgência para o projeto. Numa ótica de evolução futura, requisitos que não venham a ser implementados (por opção ou falta de tempo) possam ser encarados como trabalho futuro (evolução da aplicação).

1. **Desenho do Sistema e Arquitetura**

O capítulo de Desenho e Arquitetura é uma etapa crítica no desenvolvimento de software, onde as especificações e requisitos definidos, anteriormente, são transformados em representações gráficas e técnicas que servirão como guias para a implementação do sistema. Este capítulo assegura que o software está estruturado e organizado de forma eficiente e eficaz.

# **Desenho do Sistema**

Nesta seção, o desenho geral do sistema é abordado, envolvendo a criação de diagramas, layouts e outros elementos visuais que ilustram a estrutura e o funcionamento do software.

# **Arquitetura e Tecnologia**

Os modelos de arquitetura descrevem a estrutura de alto nível do sistema. Eles incluem a tecnologia a utilizar, a organização dos componentes do software, bem como as suas relações (por exemplo no caso de um sistema distribuídos).

# **Diagramas e Modelos**

Neste subcapítulo, são criados e apresentados diversos diagramas, como os de classes, sequência, componentes, entre outros (por exemplo diagramas relacionais das BD), para representar visualmente aspetos específicos do sistema. Estes diagramas ajudam na compreensão, comunicação e documentação da estrutura e dinâmica do Sistema.

1. **Implementação**

O capítulo de implementação é onde a conceção do software começa a tomar forma física. Nesta fase, o design e a arquitetura são transformados em código, seguindo os requisitos estabelecidos. A implementação é uma etapa prática e crítica, envolvendo a codificação, a revisão do código e a preparação para os testes e a implantação.

# **Ambiente de Desenvolvimento**

Esta seção descreve a configuração do ambiente de desenvolvimento. Inclui a seleção e a configuração das ferramentas de desenvolvimento (IDEs, entre outros), sistemas de controle de versão, *frameworks* e bibliotecas necessárias para desenvolver o software.

# **Desenvolvimento**

Neste subcapítulo, os programadores devem descrever em linhas gerais como o código fonte do software é coerente com desenho e arquitetura pretendida, eventualmente explicando opções técnicas tomadas durante o desenvolvimento. Também se deve detalhar as linguagens de programação utilizadas (e.g. C, JAVA, C#, HTML, CSS, JS, PHP, SQL, etc.), os padrões adotados (e.g estruturada, objetos, MVC, APIs, etc.).

# **Controle de Versão**

O controle de versão é essencial para rastrear e gerir as mudanças no código fonte. Esta seção explica o sistema de controle de versão utilizado (e.g. CVS, GIT, etc), os protocolos (estabelecidos dentro do grupo) para *commits*, revisões, *merges* e outros aspetos considerados críticos para garantir a integridade e a segurança do código.

1. **Testes**

Neste capítulo deve ser explicar de que forma a aplicação foi testada, no sentido de melhorar as suas funcionalidades, bem como para verificar e corrigir eventuais erros existentes.

1. **Conclusões**

Nas conclusões deve ser feita uma síntese do corpo do relatório, destacando todos os aspetos julgados como relevantes no trabalho efetuado. Deverá fazer-se uso do espírito crítico com vista a sugerir e a identificar possíveis evoluções no trabalho.

1. **Fontes**

[identificação] autor, "título do livro", editora, número da edição, ano da edição. [identificação] nome do site da Internet, "link para a página".

[identificação] autor, "título do artigo", acta da conferência, volume, páginas, local de apresentação do artigo, ano. Exemplo:

**para um livro**

[1] Parsons,J.D., The Mobile Radio Propagation Channel, Pentech Press, London, UK, 1992.

i.e., nome(s) do(s) autor(es), título do livro, nome da editora, local da edição, país da edição, ano da edição.

**para um artigo em revista**

[2] Fernandes,L., "Developing a System Concept and Technologies for Mobile Broadband Communications", IEEE Personal Communications Magazine, Vol. 2, No. 1, Feb. 1995, pp. 54-59.

i.e., nome(s) do(s) autor(es), título do artigo, nome da revista, volume da edição, número da edição, mês da edição (abreviado com 3 letras, excepção aos meses com 4 letras), ano da edição, páginas.

**para uma comunicação em conferência**

[3] Correia,L.M. and Francês,P.O., "A Propagation Model for the Average Received Power in an Outdoor Environment in the Millimetre Waveband", in Proc. of VTC’94 - 44 th IEEE Vehicular Technology Conference, Stockholm, Sweden, June 1994.

i.e., nome(s) do(s) autor(es), título da comunicação, nome da conferência, local da conferência, país da conferência, mês da conferência (abreviado com 3 letras, excepção aos meses com 4 letras), ano da conferência.

**Anexos**

Devem ser remetidos, para anexo, todos os documentos (datasheets, technical reports, etc.) que sendo referidos no relatório não são considerados essenciais para a compreensão do trabalho desenvolvido.

Cada anexo deverá ser identificado com numeração romana, sem título, e mencionado na parte do texto que complementa.

Um dos anexos que deverá constar é um pequeno tutorial de utilização do sistema criado.

**Formatação e outras recomendações a ter em consideração**

O relatório deve ser escrito num processador de texto, tomando-se em consideração os seguintes aspetos:

Tipo de letra: Deve ser coerente em todo o relatório (Arial ou Calibri)

Tamanho de letra

Título do capítulo: 16 bold Secção: 14 bold Subsecção: 12 bold

Texto: 11

Espaço entre linhas: 1,5

O texto deve ser justificado

Todas as imagens e tabelas devem estar numeradas e legendadas

Exemplo:

1 – TÍTULO DO CAPÍTULO

1.1– TÍTULO DA SECÇÃO

1.1.1 – TÍTULO DA SUB-SECÇÃO

Margens: Superior: 2 cm Inferior: 2 cm Esquerda: 3 cm Direita: 2 cm

**Deve-se sempre evitar:**

* A subjetividade
* As figuras de estilo
* As palavras desnecessárias
* As repetições
* O excesso de adjetivação
* As palavras ambíguas

**Clareza e objetividade:**

O texto deve ser agradável de ler e esclarecedor do leitor, mesmo quando este não é especialista na matéria.

Deve rever o texto e ter especial atenção a:

* A pontuação e ortografia (em caso de dúvida, consultar um prontuário ou um dicionário)
* A divisão do texto em parágrafos que facilitem a leitura
* A utilização de títulos, subtítulos, numerações, alíneas, etc., quando necessário
* Cuidado na apresentação
* Inclusão da bibliografia dos documentos consultados
* Não fazer um trabalho tão longo que se torne maçador
* Não tentar impressionar com a quantidade, mas sim com a qualidade do trabalho
* Não juntar em anexo publicações ou documentos desnecessários
* Não copiar integral ou parcialmente os textos consultados – identificar as citações e fazer resumos