INSTITUTO POLITÉCNICO DE BRAGANÇA
Escola Superior de Tecnologia e Gestão

Título do Projeto

Nome do Aluno - Número Mecanográfico

Nome do Aluno - Número Mecanográfico

Trabalho realizado sob a orientação de

Prof. Nome do Orientador

Prof. Nome do Co-Orientador

Licenciatura em Engenharia Informática

2023-2024

Título do Projeto

Relatório da UC de Projeto
Licenciatura em Engenharia Informática
Escola Superior de Tecnologia e Gestão

Nome do Aluno - Número Mecanográfico
Nome do Aluno - Número Mecanográfico

2023-2024

A Escola Superior de Tecnologia e de Gestão não se responsabiliza pelas opiniões expressas neste relatório.

Declaro que o trabalho descrito neste relatório é da minha autoria e é da minha vontade que o mesmo seja submetido a avaliação.

Nome do Aluno - Número Mecanográfico

Nome do Aluno - Número Mecanográfico

Dedicatória

(Facultativo) Dedico este trabalho a ...

Agradecimentos

(Facultativo) Agradeço a ...

Resumo

O resumo (no máximo com 250 palavras), permite a avaliação do interesse de um documento e facilita a sua identificação na pesquisa bibliográfica em bases de dados onde o documento se encontre referenciado.

É recomendável que o resumo aborde, de forma sumária:

- Objetivos principais e tema ou motivações para o trabalho;
- Metodologia usada (quando necessário para a compreensão do relatório);
- Resultados, analisados de um ponto de vista global;
- Conclusões e consequências dos resultados, e ligação aos objetivos do trabalho.

Como este modelo de relatório se dirige a trabalhos cujo foco incide, maioritariamente, no desenvolvimento de software, algumas destas componentes podem ser menos enfatizadas, e acrescentada informação sobre análise, projeto e implementação do trabalho.

O resumo não deve conter referências bibliográficas.

Palavras-chave: termos (no máximo 4), que descrevem o trabalho.

Abstract

Direct translation (maximum of 250 words) to English of the section “Resumo”.

Keywords: direct translation of “Palavras-chave”

Índice

Lista de Tabelas

Lista de Figuras

Capítulo 1

Introdução

O Capítulo 1 é dedicado a uma introdução ao tema do trabalho, descrevendo as ideias gerais do problema em foco e a sua importância. Devem ainda ser explicitados os objetivos do trabalho, clarificada a estrutura do relatório e indicadas as convenções tipográficas.

1.1 Enquadramento

Deve haver um enquadramento introdutório, que descreva o contexto em que o trabalho se insere, referenciando a proposta original do projeto, que deve constar no primeiro apêndice do documento (ver apêndice A).

1.2 Objetivos

Os objetivos do trabalho devem ser apresentados de forma clara e compatível com a proposta original do projeto. Na eventualidade de os objetivos originais terem sido reformulados, devem ser apresentadas as razões objetivas que conduziram a essa reformulação.

Idealmente, deve-se incluir um cronograma do projeto, indicando explicitamente as tarefas realizadas e o tempo dedicado a cada uma. Existindo um cronograma na proposta original do projeto, deverão justificar-se eventuais discrepâncias com o cronograma real.

1.3 Estrutura do Documento

Este modelo de relatório assume que a maioria dos projetos de fim de curso são centrados no desenvolvimento de uma solução informática para um problema. Sendo esse contexto, é apropriada uma estrutura que descreva a análise, conceção e desenvolvimento da solução implementada. Em particular, em projetos de desenvolvimento de software, espera-se que o relatório documente as principais fases do ciclo de desenvolvimento.

Nos casos em que o trabalho corresponde sobretudo à integração e/ou avaliação de componentes pré-existentes, a estrutura do relatório deverá ser adaptada em conformidade, com ênfase na descrição das tecnologias subjacentes, sua articulação e avaliação.

A estrutura efetivamente adotada para o resto do relatório é, normalmente, clarificada nesta secção, usando texto semelhante a: "O resto do relatório está organizado da seguinte forma: no capítulo 2 descreve-se ...; no capítulo 3 ...; ...; finalmente, o último capítulo apresenta as conclusões e direções de trabalho futuro."

1.4 Normas de Composição

Para além de uma organização que reflita o percurso seguido, o relatório deve estar bem formatado e ter aspeto sóbrio, convidando à leitura e fazendo jus ao mérito do trabalho descrito. Neste sentido, apresentam-se de seguida algumas das normas a levar em conta[1].

Convenções Tipográficas

Por vezes, opta-se por apresentar as convenções tipográficas seguidas no documento, ou seja, em que circunstâncias se usam texto em *itálico*, **negrito**, ou de `espaçamento uniforme` (esta última formatação é normalmente usada para apresentar código fonte), bem como quais as fontes tipográficas usadas, respetivas dimensões, etc.

[1]Estas normas, assim como este modelo de documento, são compatíveis com o preconizado no "Regulamento da Unidade Curricular de Projecto das Licenciaturas", designadamente no que diz respeito ao relatório do projeto.

Tabelas e Figuras

Tabelas e figuras devem ser numeradas automaticamente e ter um tamanho equilibrado (nem muito grande, nem muito pequeno), como a Tabela 1.1 e as Figuras 1.1, 1.2 e 1.3.

Nome da Coluna 1	**Nome da Coluna 2**	**Nome da Coluna 3**
conteúdo A	conteúdo B	conteúdo C
conteúdo D	conteúdo E	conteúdo F

Tabela 1.1: Exemplo de tabela.

Figura 1.1: Exemplo de imagem PNG.

Sempre que possível, devem-se usar formatos escaláveis (e.g., PDF), como na Figura 1.2, e evitar imagens comprimidas (formatos JPG, PNG, GIF, etc.), como na Figura 1.1.

Em gráficos devem-se indicar sempre as grandezas associadas a cada eixo, bem como a respetiva legenda – ver exemplo na Figura 1.3. Adicionalmente, o esquema de cores ou de traços para as linhas, deve ser sóbrio e prevenir ambiguidades na leitura do gráfico.

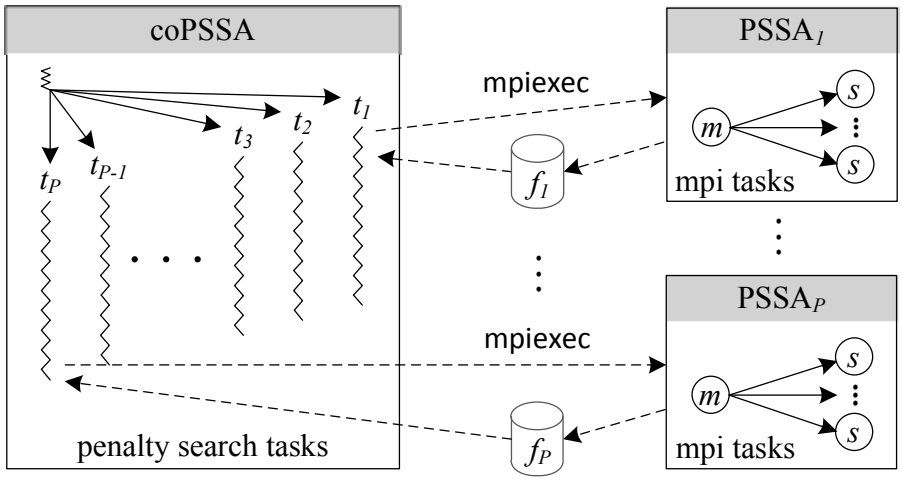

Figura 1.2: Exemplo de imagem PDF.

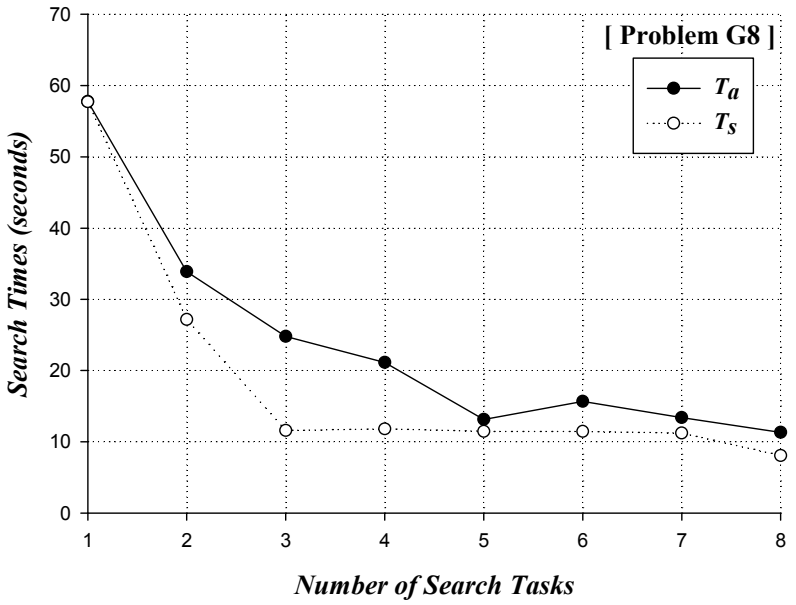

Figura 1.3: Exemplo de gráfico.

Distribuição dos Elementos

A forma como o texto e outros elementos (tabelas, figuras, etc.) se distribui por uma página, deve ser tal que se evitem grandes blocos vazios no final da página. Embora algumas sistemas de composição tendam a garantir isso automaticamente (como o LaTeX), é costume serem necessárias afinações para resolver, manualmente, essas (e outras) desformatações. No entanto, essas afinações devem ser deixadas para a fase pré-impressão, já com o conteúdo do documento estabilizado, de forma a evitar trabalho inconsequente.

Correcção Ortográfica

Para além da qualidade tipográfica do relatório, é imprescindível minimizar (se possível até, erradicar) erros ortográficos. Qualquer sistema de composição de documentos suporta correção ortográfica (e, muitas vezes, sintática), pelo que não é aceitável a submissão para avaliação de relatórios sem revisão ortográfica prévia.

Referências Bibliográficas

Ao longo do documento, deve ficar sempre perfeitamente claro o que é escrita original e o que foi baseado (ou até reproduzido de) noutras fontes.

Todas as fontes devem ser descritas na formatação usada na Bibliografia e referenciadas, no texto, pelos seus identificadores únicos. Por exemplo: "Uma solução para o problema em causa deve respeitar as propriedades x, y e z [1].", ou "Neste trabalho explorou-se a API PThreads [2] com o objetivo de ...". No modelo em LaTeX, as referências bibliográficas são definidas no ficheiro `libs/refs.bib`, onde existem entradas de diferentes tipos: livro [3], artigo de conferência [1], relatório técnico [2] e sítio web [4].

A reprodução fiel de texto de fontes externas deve ser limitada, surgir entre aspas, ligeiramente destacada, e ter apensa a respetiva referência bibliográfica. Por exemplo:

> "Recently, the employment of GPU devices is a key to achieve higher performance for computer systems. On those systems, GPUs are used for general calculation but with extreme parallelism." [1]

Siglas

Na primeira vez as siglas devem surgir por extenso, sendo resumidas nas vezes seguintes. Por exemplo: "o curso atual de Engenharia Informática da Escola Superior de Tecnologia e Gestão (ESTiG) foi reformulado em 2015 e, a par com o curso de Informática de Gestão, representa o leque de licenciaturas da área de Informática que a ESTiG oferece". No modelo em LaTeX, as siglas são definidas no ficheiro `acronym.tex`.

Capítulo 2

Estado da Arte

A transição de infraestruturas baseadas em máquinas virtuais monolíticas para arquiteturas orientadas a *containers* revolucionou a administração de sistemas. Este capítulo explora as tecnologias e os paradigmas atuais que suportam a criação de plataformas de serviços de rede ágeis, seguras e observáveis.

2.1 Containerização e Orquestração Leve

Tradicionalmente, a segregação de serviços exigia a alocação de máquinas virtuais independentes, o que resultava num elevado *overhead* de recursos e numa gestão complexa. Atualmente, motores de *containerização* como o Docker e o Podman tornaram-se o padrão da indústria. Estas ferramentas permitem encapsular uma aplicação e todas as suas dependências num ambiente isolado, que partilha o *kernel* do sistema operativo *host*.

O Podman, em particular, destaca-se pela sua arquitetura *daemonless* e pela capacidade de executar *containers* sem privilégios de *root*, indo ao encontro das melhores práticas de segurança (o princípio do mínimo privilégio). Para a orquestração de infraestruturas de pequena a média dimensão, ferramentas declarativas como o Docker Compose oferecem a "orquestração leve"ideal. Permitem definir infraestruturas complexas como código (IaC - *Infrastructure as Code*) de forma totalmente reprodutível e facilmente versionável.

2.2 Observabilidade Centralizada

Manter um conjunto de serviços distribuídos exige mais do que uma simples monitorização; exige observabilidade. O estado da arte baseia-se em *stacks open source* complementares:

- **Métricas:** O Prometheus atua como o motor principal de recolha de métricas em formato *time-series*, trabalhando em conjunto com o Grafana para a visualização avançada de dados através de *dashboards* dinâmicos.
- **Logs:** Soluções como a *stack* ELK (Elasticsearch, Logstash, Kibana) ou o Loki (otimizado pela Grafana Labs para ambientes de *containers*) permitem agregar *logs* de múltiplos serviços num único ponto central. Isto facilita substancialmente o processo de *troubleshooting*.

2.3 Segurança e CI/CD em Ambientes Locais

A segurança *by design* é um requisito fundamental na gestão de redes modernas. Isto inclui o uso sistemático de *Reverse Proxies* (como o Nginx, Traefik ou Caddy), que gerem a terminação TLS/SSL de forma automática, e o *hardening* de imagens base.

A adoção de *pipelines* de CI/CD (Integração e Entrega Contínuas) deixou de ser um processo exclusivo do desenvolvimento de *software* e passou a integrar a gestão de infraestruturas (GitOps). A utilização de repositórios Git acoplados a *runners* locais permite que qualquer alteração na infraestrutura (como uma atualização de versão do servidor DNS ou a adição de um novo serviço Web) seja testada antes do *deploy* para o ambiente de produção. Esta prática reduz o erro humano e garante um *rollback* fiável em caso de falha dos serviços.

Capítulo 3

Levantamento das Necessidades Funcionais

Com base no guião do projeto proposto, o objetivo central consiste na criação de um "data center"de serviços de rede que possa ser gerido de forma reprodutível, recorrendo exclusivamente a ferramentas *open source*. Para atingir este objetivo, as necessidades do projeto dividem-se em duas categorias principais: os serviços a disponibilizar (requisitos funcionais) e os requisitos técnicos e de arquitetura (requisitos não-funcionais).

3.1 Serviços de Rede a Disponibilizar

A infraestrutura deve contemplar um conjunto de serviços típicos de uma rede de pequena/média dimensão. Os serviços a implementar são os seguintes:

- **Servidor DNS:** Implementação de um servidor com capacidade recursiva e autoritativa para a resolução de nomes na rede interna.
- **Reverse Proxy HTTPS:** Criação de um ponto de entrada único e seguro para o encaminhamento de tráfego para os diversos serviços internos.
- **Serviços Web e de E-mail:** Alojamento de páginas *web* e disponibilização de um sistema de gestão básica de correio eletrónico.

- **Servidor de Ficheiros:** Sistema centralizado para o armazenamento e a partilha de ficheiros na rede.
- **VPN (*Virtual Private Network*):** Implementação de um serviço que permita o acesso remoto e seguro à infraestrutura interna de *containers*.

3.2 Requisitos Técnicos e de Arquitetura

Para além do correto funcionamento dos serviços, a plataforma tem de respeitar rigorosos critérios de implementação e gestão:

- **Containerização e Configuração Declarativa:** Todos os serviços têm de ser executados em *containers* (Docker ou Podman). A configuração deve ser feita de forma declarativa (utilizando ficheiros como o `docker-compose.yml`), o que garante a sua reprodutibilidade e permite o versionamento.
- **Observabilidade e Monitorização:** É obrigatória a integração de uma *stack open source* (por exemplo, Prometheus, Grafana, Loki ou ELK/EFK) para a recolha contínua de métricas, gestão de *logs* centralizados e configuração de alertas básicos (como latência, erros 5xx ou saturação de CPU/RAM).
- **Segurança (*Hardening*):** O projeto exige a aplicação do princípio do mínimo privilégio nos *containers*, gestão segura e isolada de segredos, encriptação obrigatória de tráfego (TLS) e uma política de atualização automatizada das imagens dos *containers*.
- **Pipeline de CI/CD:** Deve ser implementada uma *pipeline* simples (utilizando, por exemplo, o Git em conjunto com um *runner* local) para testar automaticamente as alterações de configuração e assegurar um *deploy* controlado para o ambiente de produção do laboratório.

Capítulo 4

Desenvolvimento/Implementação

Neste capítulo é descrito o trabalho de implementação, salientando os pontos mais relevantes da mesma, dificuldades encontradas ou soluções técnicas inovadoras desenvolvidas ou aplicadas. Em particular, se foi usado código desenvolvido por terceiros (por exemplo, código *open-source*), deve ser facilmente distinguível quais as funcionalidades originais do mesmo e o que foi necessário implementar para obter as funcionalidades desejadas.

Capítulo 5

Testes/Avaliação/Discussão

Este capítulo apresenta os testes realizados para verificar que o projeto desenvolvido cumpre os objetivos assumidos e resolve, de facto, o problema descrito na Análise/Modelação.

Para uma melhor compreensão, os resultados de cada teste devem ser precedidos de uma descrição, mesmo que resumida, do teste realizado e dos resultados esperados.

Os resultados do trabalho são comentados, acrescentando-lhe valor:

- O que é que se pode inferir ou conjeturar dos resultados obtidos?
- O que poderia/deveria ter sido feito de forma diferente?
- Onde se foi além dos objetivos iniciais?
- Quais os objetivos que ficaram por cumprir, e porquê ?

Capítulo 6

Conclusões

As conclusões devem sintetizar e proporcionar uma perspetiva unificadora ao trabalho efetuado. Poderá ser feita uma breve referência a trabalhos de outros com semelhanças ao efetuado e ao conhecimento que resultou do trabalho efetuado, bem como sugestões de trabalho futuro. A coerência do documento implica que as conclusões devem ser coerentes com as ideias expostas na introdução.

Bibliografia

[1] R. Aoki, S. Oikawa, T. Nakamura e S. Miki, "Hybrid OpenCL: Enhancing OpenCL for Distributed Processing," em *Proceedings of the 2011 IEEE Ninth International Symposium on Parallel and Distributed Processing with Applications*, sér. ISPA '11, Washington, DC, USA: IEEE Computer Society, 2011, pp. 149–154, ISBN: 978-0-7695-4428-1. DOI: 10.1109/ISPA.2011.28 URL: http://dx.doi.org/10.1109/ISPA.2011.28

[2] L. L. N. L. Blaise Barney, "POSIX Threads Programming," rel. téc., ago. de 2012.

[3] A. D. Autor Um, *Titulo do livro*, 1st. Cidade, distrito, PAIS: Publicaçoes, lda, 2012, ISBN: 123456789.

[4] R. Ford. "Earthquake: Twitter Users Learned of Tremors Seconds Before Feeling Them. "URL: http://www.hollywoodreporter.com/news/earthquake-twitter-users-learned-tremors-226481

Apêndice A

Proposta Original do Projeto

Curso de Licenciatura em Engenharia Informática

Projeto 3º Ano - Ano letivo de 2016/2017

<Título do projeto>

Orientador: <Nome do orientador>

Coorientador: <Nome do coorientador>

1 Objetivo

<Objetivo do projeto>

2 Detalhes

<Detalhes que julguem ser necessários>

3 Metodologia de trabalho

<Eventual metodologia de trabalho>

Dimensão da equipa:

Recursos necessários:

Apêndice B

Outro(s) Apêndice(s)

Listagens de código fonte, texto/imagens produzidos por testes complementares, etc.