|  |  |
| --- | --- |
| ISEL | Licenciatura em  Engenharia Informática e de Computadores |

**5G Qos Aplicação Android**

|  |  |
| --- | --- |
|  | Afonso Nobre, nº 44777 |
|  | Ricardo Silva, nº 44837 |

|  |  |
| --- | --- |
| Orientadores | José Simão |
|  | Nuno Cota |

Relatório beta realizado no âmbito de Projecto e Seminário,  
do curso de licenciatura em Engenharia Informática e de Computadores  
Semestre de Verão 2020/2021

\*Maio\* de 2021

# Resumo

Os recentes desenvolvimentos na rede 5G oferecem não só uma interface de rádio sofisticada como também uma eficiente arquitetura de sistemas de rede. Estes avanços disponibilizam um leque de oportunidades e novas aplicações, como por exemplo a tecnologia na construção de veículos autónomos. Para determinar a influência das condições da rede em aplicações que dela dependem, irá ser usado um sistema capaz de gerar tráfego em diferentes níveis e protocolos permitindo a recolha de informação para análise da qualidade do serviço.

O sistema usado presentemente é composto essencialmente por três tipos de componentes, nomeadamente:

*On Board Units (OBU)* - Peça de *hardware* e *software* móvel instalada em veículos de forma a gerar tráfego e recolher os parâmetros de medida de performance em diferentes pontos de controlo e observação.

*Fixed Side Units (FSU)* - Agente de *software* fixo instalado em localizações estrategicamente pensadas tanto em Portugal como em Espanha. Estes têm a responsabilidade de gerar tráfego e comunicar com as sondas móveis para recolher parâmetros de rede tais como o *downlink* e o *uplink*.

*Management System* - Plataforma de *softawre* centralizada responsável pela gestão e configuração de planos de testes. Também é responsável por recolher e processar todos os resultados obtidos pelas sondas durante a execução dos planos de testes.

O desenvolvimento do projeto é motivado pela oportunidade da criação de uma aplicação móvel para simular uma sonda móvel mais simplificada num *smartphone* comum, de forma a complementar o sistema já existente oferecendo mais portabilidade e versatilidade.

**Palavras-chave:** 5G; smartphone; aplicação móvel; Management System

# Abstract

The recent developments in 5G technology brings not only a sophisticated radio interface, but also a performant network system architecture. These technical achievements may bring new opportunities and new applications, for example, autonomous vehicles. To determine the influence of radio network conditions on the applications performance, it will be used a system that generates synthetic traffic at different levels and different protocols to collect information, allowing the system to analyze the service quality.

The system architecture is composed by three components:

On Board Unit (OBU) – Hardware and software probe to be installed on vehicles in order to generate traffic and collect performance measurements at different points of control and observation.

Fixed Side Units (FSU) - Software agent to be installed on both Portugal and Spain. The fixed side unit is used to generate traffic and collect performance measurements on the network side, on both downlink and uplink traffic.

Management System - Centralized software platform used to manage test plan configuration. It will also be responsible for collecting and processing all performance assessment results obtained during test trials.

The development of this project is motivated by the opportunity of developing a mobile application to simulate a simplified on board unit in an ordinary mobile phone to complement the Management System by offering more portability and versatility.

**Keywords:** 5G; Mobile; Application; Management System

**Índice**

[Resumo iii](#_Toc73626981)

[Abstract v](#_Toc73626982)

[Lista de Figuras ix](#_Toc73626983)

[Lista de Tabelas xi](#_Toc73626984)

[1. Introdução 13](#_Toc73626985)

[1.1 Iniciativa 13](#_Toc73626986)

[1.2 Motivação 13](#_Toc73626987)

[1.3 Objetivo 13](#_Toc73626988)

[1.4 Especificações do Projeto 13](#_Toc73626989)

[1.5 Estrutura do relatório 14](#_Toc73626990)

[2. Integração com o sistema 15](#_Toc73626991)

[2.1 Sistema já existente 15](#_Toc73626992)

[2.2 Integração no sistema 15](#_Toc73626993)

[2.3 Funcionamento e fluxo de comunicação 16](#_Toc73626994)

[2.4 Linguagens e Frameworks utilizadas (opcional acho que ficava bem e sempre é mais coisas que se mete) 16](#_Toc73626995)

[3. Arquitetura 16](#_Toc73626996)

[3.2 Solução para os objetivos propostos 16](#_Toc73626997)

[3.3 Modelo de Dados 17](#_Toc73626998)

[4. Detalhes de implementação 17](#_Toc73626999)

[4.1 Autenticação 17](#_Toc73627000)

[4.1.1 Refresh Da token 17](#_Toc73627001)

[4.2 Recolha de parâmetros de rede 17](#_Toc73627002)

[4.3 Representação de dados 18](#_Toc73627003)

[4.3.1 Recolha de dados 18](#_Toc73627004)

[4.3.2 Representação de dados 18](#_Toc73627005)

[4.3.3 Comunicação entre fragmentos e criação de novas sessões de teste 18](#_Toc73627006)

[4.4 Implementação de testes com e sem intervenção do utilizador 18](#_Toc73627007)

[5. Avaliação Experimental 19](#_Toc73627008)

[6. Conclusão e trabalho futuro 19](#_Toc73627009)

[6.1 Conclusão 19](#_Toc73627010)

[6.2 Trabalho futuro 19](#_Toc73627011)

[Referências 20](#_Toc73627012)

[A.1 Diagramas da Aplicação 21](#_Toc73627013)

[A.2 Modelos de dados 23](#_Toc73627014)

# 

# Lista de Figuras

[Figura 1 - *Login* 15](file:///C:\Users\afons\Desktop\College\Projeto\5G-QOS\Reports\RelatorioFinalProjeto.docx#_Toc73626455)

[Figura 2 – Diagrama de casos de utilização. 20](#_Toc73626456)

# Lista de Tabelas

[Tabela 1 -Um exemplo de legenda de tabela. Prazos de entrega de Projecto e Seminário, 4](#_Toc416101908)

# Introdução

O capítulo seguinte irá introduzir a iniciativa para este projeto, a motivação, os objetivos e por último uma breve descrição da estrutura do relatório.

## 1.1 Iniciativa

A iniciativa deste projeto foi despoletada pelo recente avanço na tecnologia 5G. Estes avanços disponibilizam um leque de oportunidades e novas aplicações. Uma vez que o sistema já desenvolvido tem uma componente portátil bastante limitada surgiu a necessidade de desenvolver uma aplicação móvel de forma a tentar minimizar este problema.

## 1.2 Motivação

O surgimento de avanços no desenvolvimento da tecnologia 5G implica uma necessidade de reiterar os passos a dar na implementação das aplicações da tecnologia. Surgiu então uma motivação de criar uma aplicação que seja capaz de se servir desta nova tecnologia.

O âmbito é criar uma aplicação que permita oferecer uma nova frente ao projeto *QoS*, para que este possa criar um plano exaustivo de análise à cobertura à tecnologia recém-formada.

O *smartphone* é um dispositivo cada vez mais comum no quotidiano da sociedade, pelo que se torna no candidato perfeito para a execução da nova frente. Com um equilíbrio entre a portabilidade e as necessidades de *hardware*, é criada uma janela de oportunidade para desenvolver uma aplicação que tire partido da praticidade do *smartphone* para executar os planos de testes criados pelo sistema de gestão.

### 1.3 Objetivo

O objetivo deste projeto é desenvolver uma aplicação móvel para complementar o sistema já existente simulando de forma simplificada uma *On Board Unit* oferecendo mais portabilidade e versatilidade. Esta tem que ser capaz de recolher parâmetros de rede de forma passiva, sem criar tráfego ou influência na rede, e realizar planos de testes propostos pelo sistema de gestão.

## 1.4 Especificações do Projeto

Para garantir que a aplicação seja realmente capaz de cumprir estas tarefas deverá de cumprir os seguintes requisitos:

* Recolher e mostrar parâmetros rádio 4/5G e a localização GPS;
* Executar e reportar os planos de testes fornecidos pelo sistema de gestão;
* Executar *pings* para domínios especificados nos planos de testes;
* Guardar e representar todos os resultados obtidos durante a recolha de parâmetros rádio e testes realizados.

## 1.5 Estrutura do relatório

//TODO

# Integração com o sistema

## 2.1 Sistema já existente

O sistema QoS é uma complexa rede de sondas que, ao comunicar entre si, conseguem criar condições de tráfego virtual no âmbito de executar testes e análises à cobertura da rede na posição em que se encontram.

As sondas *on-board* encontram-se instaladas em veículos, e contém vários modems para fazer os testes. Contêm uma interface *web* local, que permite que a sua manutenção seja feita diretamente no dispositivo. Estas conseguem também ligar-se ao CAN BUS do veículo, de forma a aumentar a eficiência e precisão das análises feitas. A maior parte da análise é feita *offlline*, e só apenas depois de todas as medidas estarem completas é que o dispositivo faz uma conexão ao sistema central para transmitir os dados que gravou.

As sondas fixas encontram-se posicionadas em certas localizações ao longo da Península Ibérica e correm numa máquina virtual dedicada, que vai ser instala na *MEC* da tecnologia *5G* e/ou no centro *ITS*. A nível de *hardware*, estas sondas são semelhantes às sondas *on-board*, com a exceção que estas não têm *modems* de comunicação móvel, nem a interface *web* local, porque visto que estas se encontram ligadas à rede central, podem ser geridas pelo sistema de gestão.

O sistema de gestão central é a ponte entre todos estes dispositivos. Este sistema está ligado a todas as sondas e faz a gestão da execução de testes, para evitar interferências no tráfego da rede *5G*. É este sistema que constrói e delega os diferentes planos de testes que as sondas vão executar, bem como recebe os resultados e os processa, para depois os entregar a ferramentas especializadas na sua análise.

## 2.2 Integração no sistema

A aplicação móvel vai ser desenhada na imagem de uma sonda *on-board*. Mas, visto que o *smartphone* não é tão preciso e tão capaz como uma dessas sondas, pequenos desvios vão ter de ser feitos na modelação da aplicação. Apesar dos testes passivos à cobertura serem semelhantes (excetuando a interface de ligação ao veículo), os planos vão ter de ser simplificados para que o telemóvel se comporte com o desempenho desejado. Na face desta condição, tiveram de ser criadas novas rotas na *API*, desenvolvidas exclusivamente para a nova sonda móvel (*Mobile Unit* – *MU*)

(colocar esquema que fizemos no proposal com o esquema do servidor)

## 2.3 Funcionamento e fluxo de comunicação



Figura 1 - *Login*

A aplicação começa por apresentar um ecrã de *login*, para o utilizador se autenticar no sistema. Este tem de estar registado no mesmo para se poder autenticar. Durante o processo de autenticação é também feito o registo do dispositivo na rede, visto que cada *smartphone* atua como uma sonda única, e então tem de ter o seu registo de testes e parâmetros único. Esta individualização é útil para separar as *FSUs*, *OBUs* e *MUs*.

## 2.4 Linguagens e Frameworks utilizadas (opcional acho que ficava bem e sempre é mais coisas que se mete)

À luz do facto da *Google* e do *Android* andarem de mão dada desde o nascimento do segundo, é de notar que o uso do *IDE Android Studio* disponibilizado pela empresa seria o melhor competidor para o desenvolvimento do projeto. Visto que o editor apenas suporta linguagens que corram sobre a *Java Virtual Machine*, foi necessário fazer a escolha entre as já reduzidas opções. Numa última instância fez-se a escolha da linguagem *Kotlin*. Esta linguagem é uma versão evoluída de *Java*, com aumentos de concisão e simplicidade a nível de código. É também 100% interoperável com java, pelo que o uso de *jars* implementados em *Java* não está fora de questão. Sendo que esta linguagem foi implementada pelos criadores do *IDE* no qual a *Google* se inspirou para criar o *Android Studio*, *Kotlin* é a linguagem favorita da empresa para desenhar aplicações *Android*.

Foram usadas várias *Frameworks* para implementar a lógica da aplicação, designadamente:

* *Volley* – Utilizado para fazer os pedidos *HTTP* à *API* disponibilizada. É possível redefinir o modo como os pedidos e respostas são processados para personalizar cada pedido à forma da aplicação.
* *MPAndroidChart* – Esta é a *Framework* utilizada para o desenho dos vários gráficos de informação que se encontram ao longo da aplicação. Com uma estrutura clara, é possível definir o número e aspeto de varáveis presentes em cada gráfico e, através da afetação de *flags*, personalizar a navegação dentro de cada gráfico também.
* *Room* – Implementado por base em *SQLite*, esta é a *Framework* que permite fazer a criação de uma base de dados local ao processo da aplicação. Através de um esquema de Entidades e classes *DAOs* (*Data Access Object*), é possível fazer pedidos de base de dados com facilidade.
* *Material UI* – *Framework* de componentes visuais. É a fonte de vários componentes relativos à *UI* da aplicação, tanto estruturais como meramente visuais.

# Arquitetura

Nesta secção é realizada uma descrição detalhada da arquitetura adotada para a construção da aplicação android e a sua integração no sistema previamente desenvolvido.

## 3.1 Desafios Propostos

Desenvolver uma aplicação de testes à cobertura da rede não é tão linear quanto soa.

Em primeiro lugar, é necessário criar uma certeza que enquanto se fazem os testes, que a aplicação não cria qualquer tipo de interferência na rede, de forma a perturbar os resultados destes.

Um dos desafios iniciais foi paralelizar a execução de testes passivos por parte da aplicação. A recolha de parâmetros de rede, como a velocidade de *download* e *upload*, bem como as coordenadas da localização do dispositivo, entre outros, tem de ser feita simultaneamente de forma a atualizar constantemente os dados a apresentar e gravar.

Além desta paralelização, é necessário que esta análise retrospetiva seja feita em segundo plano, bem como garantir que a análise seja feita mesmo com a aplicação em baixo. Quando a aplicação está em primeiro plano, as recolhas feitas em *background* são lidas e apresentadas ao utilizador, sob forma de gráfico, tabela, ou mesmo um conjunto de campos nomeados.

Existe um conceito de sessões de teste no contexto da aplicação. Estas são sessões cujo propósito é, como o nome indica, um momento que a aplicação usa para executar os planos de testes fornecidos pelo sistema de gestão. Durante essa sessão serão também registados os parâmetros de rede, para se poder analisar como é que a execução dos planos influencia a rede. Sendo que esta gestão e execução de sessões é feita num plano à parte dos planos de apresentação de dados, surge o problema de que tem de haver comunicação entre estes planos.

## 3.2 Solução para os objetivos propostos

Neste paragrafo deveríamos falar de forma muito superficial e pouco técnica dos pontos enumerados anteriormente deixando o leitor a perceber que os problemas encontrados de facto eram impostos pelas limitações do sistema android e a dificuldade que é encontrar soluções para resolver certos problemas.

Para o primeiro desafio, para evitar influências indesejadas na rede, foi criado um conceito de sequencialização de tarefas. Dessa forma, a aplicação cria um *pipeline* nas tarefas que tem a executar dependendo da natureza da mesma.

Na tarefa da execução de testes passivos a paralelização é conseguida através do uso de *threads* de *background* para fazer a recolha de cada tipo de parâmetro. Deste modo, atráves da base de dados é possível criar um canal unidirecional de comunicação entre a *thread* e a *UI*, resolvendo o terceiro desafio apontado.

Na paragem da aplicação quando esta passa para um plano de fundo a comunicação com a *UI* deixa de ser necessária, mas a recolha de parâmetros continua a ser executada pelas *threads* de *background* a quem as tarefas de recolha foram delegadas, resolvendo assim o penúltimo desafio.

Através de um serviço de *Publisher Subscriber* é possível fazer a comunicação de planos no mesmo nível de hierarquia estrutural. Assim, o plano de gestão de sessões consegue enviar as mensagens e comunicados necessários aos outros planos para estes saberem quais as informações relevantes a apresentar durante a execução das sessões.

## 3.3 Modelo de Dados

Acho que ficaria bem aqui o modelo de dados para depois entrarmos mais em detalhe no próximo capitulo.

Iremos explicar aqui também para que serve cada tabela da solução.

# Detalhes de implementação

Podemos falar de como funciona e está estruturada a aplicação colocar o nosso boneco e depois falar de cada componente mais detalhadamente.

## 4.1 Autenticação

Falar da forma como é feita a autenticação e porque.

Porque de não utilizarmos o id do android.

O porque de termos escolhido as sharedPreferences

E que é preciso ter conta na api e registado o dispositivo para utilizar a app

## 4.1.1 Refresh Da token

Podemos falar que para acesso á api e para sermos capazes de ir buscar planos de testes e executar sem interação do user necessitamos de ter a token refreshe e porque que escolhemos um worker

## 4.2 Recolha de parâmetros de rede

Falar de como é recolhido os parâmetros de rede.

Falar da forma como temos estruturado o jobscheduler e o porque de estar assim.

Falar das dificuldades e das soluções que foram tentadas implementar e porque falharam

## 4.3 Representação de dados

### 4.3.1 Recolha de dados

Falar de como uma vez que os dados são recolhidos em background que decidimos aplicar observers a livedatas vindas da db para reportar em tempo real ao utilizador.

### 4.3.2 Representação de dados

Falar do que fizemos de para conseguir representar na ui, ou seja desligar observers e ligar observers. Observar live datas e o que usamos para representar em gráficos.

Colocar aqui também os gráficos realizados e temos que tentar explicar o que que estamos a ver.

### 4.3.3 Comunicação entre fragmentos e criação de novas sessões de teste

Falar da forma como é feita a notificação de uma nova sessão entre fragmentos e porque.

(devido a termos que comunicar com o fragmento pai (DashBoard) que o botão de nova sessão foi pressionado)

## 4.4 Implementação de testes com e sem intervenção do utilizador

Para implementação de recolha de mais parâmetros podemos colocar no jobscheduler.

Se for sem intervenção do utilizador achamos que poderemos colocar workers para facilitar a arquitetura e porque os mesmo são os únicos que permitem correr trabalho programado mesmo quando a app está desligada.

# Avaliação Experimental

Falar do facto que nos baseamos numa aplicação já existente chamada netmonitor

Recomendada pelos docentes.

Mostrar os resultados numa e noutra ao mesmo tempo.

Explicar que há o problema do rssnr e porque (podemos dizer que contactamos o bacano de la e ele mesmo confirmou isso) e explicar que os resultados são idênticos e não iguais porque para serem iguais tinham que estar sobre as mesmas condições, ou seja eles usam a api velha nós a nova e tem que ser tirado exatamente ao mesmo tempo para conterem os mesmos valores.

# Conclusão e trabalho futuro

## 6.1 Conclusão

Falar do que ganhamos com este trabalho.

Falar de que existem alguns bugs (á data de hoje)

## 6.2 Trabalho futuro

Falar que terão que ser implementados mais testes e mais formas de reportar dados

De forma a poder complementar cada vez mais o sistema.

# Referências

|  |  |
| --- | --- |
| [1] | Wikipedia, “Big data --- Wikipedia, The Free Encyclopedia,” http://en.wikipedia.org/w/index.php?title=Big\_data&oldid=648786139, 2015. |
| [2] | X. Ding, X. Zhu e G. Wu, “Data mining with big data,” *IEEE Transactions on Knowledge and Data Engineering,* vol. 26, n.º 1, pp. 97-107, 2014. |
| [3] | J. Andrews, S. Buzzi, W. Choi, S. Hanly, A. Lozano, A. Soong e J. Zhang, “What Will 5G Be?,” *IEEE Journal on Selected Areas in Communications,* vol. 32, n.º 6, pp. 1065-1082, 2014. |
| [4] | L. Boytsov, “Indexing Methods for Approximate Dictionary Searching: Comparative Analysis,” *J. Exp. Algorithmics,* vol. 16, n.º may, p. 1.81, 2011. |
| [5] | T. Jurkiewicz e K. Mehlhorn, “On a Model of Virtual Address Translation,” *J. Exp. Algorithmics,* vol. 19, n.º jan, pp. 1-18, 2015. |
| [6] | J. Neumann, The Computer and the Brain, New Haven, CT, USA: Yale University Press, 1958. |
| [7] | B. Kernighan e P. Plauger, The Elements of Programming Style, New York, NY, USA: McGraw-Hill, Inc., 1982. |

# A.1 Diagramas da Aplicação

Estamos no início do anexo 1. Nalguns casos, é conveniente colocar anexos de forma a complementar os resultados. Por vezes, em casos excepcionais devido à sua dimensão, as figuras têm que ser apresentadas de forma a ocupar toda a página, na forma de paisagem (*landscape*). Podemos fazer isso da forma que se apresenta na figura 3.



Figura 2 – Diagrama de casos de utilização.

# A.2 Modelos de dados

Estamos no início do anexo 2.

O *relatório* é um resumo do projecto global. Apenas como referência, é expectável cerca de 30 a 40 páginas A4 não devendo exceder 50 páginas.. A estrutura deve ser discutida e aceite pelo orientador. Os capítulos apresentados devem ter, em geral, a seguinte organização:

**Capítulo 1** — Introdução

Enquadramento do trabalho, metas, objectivos e especificações do projecto e resumo da solução. Concluir com a descrição breve dos restantes capítulos.

**Capítulo 2** — Formulação do problema

Introdução dos conhecimentos necessários para entendimento do trabalho, estabelecimento de terminologia e descrição detalhada do problema e do seu contexto. Síntese de abordagens anteriores do problema, caso existam, indicando as razões porque são insatisfatórias.

**Capítulo 3** — Grande ideia 1

Este capítulo pode ser subdividido em secções, designadamente:

1. Introdução: descrição da abordagem do problema e da metodologia utilizada; identificação das tarefas;
2. Elenco das características / Análise do problema: especificações, constrições, ferramenta de análise, etc*.*
3. Projecto: modelo para resolução do problema;
4. Implementação: a implementação do modelo como sistema computacional; descrição concisa do *hardware* e do *software*; dificuldades e contradições encontradas e sua resolução;
5. Avaliação: testes realizados e resultados experimentais (quando possível, o objectivo, a montagem e o método usado devem ser brevemente descritos); análise crítica dos resultados.

…

**Capítulo k+2** — Grande ideia k

**Capítulo k+3** — Conclusões

Recapitulação do trabalho desenvolvido. Referir claramente as observações e conclusões importantes. Discussão de ideias e recomendações para trabalho futuro.

**Referências**

Elenco dos livros e artigos citados no relatório. As referências são numeradas consecutivamente ao longo do relatório. O número da referência deve estar entre parêntesis rectos: [1].

**Anexos**

Os anexos devem incluir as partes importantes do *dossier do projecto*. O seu conteúdo depende da natureza do projecto, mas, em geral, pode incluir: listagem de programas, resultados de testes de *software*, exemplos de ecrãs de interface com o utilizador, esquemas dos circuitos, listagem de componentes, *data sheets* críticos, resultados de testes de *hardware*, desenhos mecânicos, análise económica, etc. (quando realizável, o *relatório* deve ser acompanhado da cópia do código, bibliotecas, etc. em suporte electrónico).

**Mais algumas dicas…**

O júri para avaliação do projecto final de curso apreciará o projecto, a sua demonstração e o respectivo relatório final (valorizando a escrita enquanto forma de divulgação de conhecimento). O relatório, depois de aceite e discutido, ficará disponível na biblioteca do departamento, para consulta.

O relatório deve ser auto-suficiente, isto é, o professor ou qualquer aluno finalista deve ficar apto a perceber o trabalho que realizou sem ter de ir à biblioteca ler os artigos originais.

Não escreva para o orientador, conhecedor de todo o detalhe, ou para um principiante. Tente escrever para uma audiência constituída por estudantes finalistas. Mantenha em mente o nível de conhecimentos do leitor a que se dirige. O relatório será uma base de trabalho para estudantes em circunstâncias semelhantes. Não dificulte o trabalho do leitor nem o faça estúpido (obviamente, …). Também é impossível ser totalmente claro. Evite afirmações dogmáticas (exemplo: "O *software* é a parte mais importante do computador.").

O relatório técnico não é uma história: usualmente não segue a cronologia das actividades realizadas. Também não é um romance (atenção à adjectivação). O relatório é um documento formal, feito para descrever os aspectos importantes do trabalho realizado.

Não tente descrever a função de cada componente, por exemplo a frase "O circuito IC2 e os componentes a ele associados formam um amplificador inversor …" é adequada. Contudo, descreva detalhadamente a função de cada componente ou circuito invulgar ou crítico.

As ilustrações (figuras, tabelas, gráficos e exemplos) são auxiliares preciosos para a explicação, mas envolvem muito trabalho. As figuras e as tabelas devem ser legíveis, instrutivas, legendadas e ter título. Os exemplos devem ser suficientemente detalhados para ilustrar o conceito.

O texto deve, pelo menos, ser analisado por um corrector ortográfico: os erros de ortografia são inadmissíveis.

Recomenda-se a leitura de alguns artigos e ou livros bem escritos para adquirir sensibilidade para a arte de escrever. Os artigos premiados em conferência são, normalmente, bons exemplos de escrita.

A escrita do relatório demora sempre mais tempo do que o inicialmente previsto.

**Resumo**

No essencial, a ideia é que tem algo para vender e o “Resumo” é a montra: a mensagem deve ser suficientemente clara e encorajar o cliente a entrar — se ele não a perceber passará ao lado. O resumo inclui: a motivação para o trabalho, como o fez e os resultados principais. Devem ser evitados chavões e palavras longas, as referências são proibidas e não deve utilizar acrónimos. Tenha em conta que o leitor será influenciado quer pela informação contida no resumo quer pela maneira como este está escrito. Não há desculpas para frases curtas ou desligadas, erros de gramática ou erros de sintaxe.

Não é fácil escrever um bom resumo.

**Introdução**

Procure dar resposta às seguintes questões: qual é o problema? porque é importante? o que é que outros já fizeram? quais as ideias base da solução apresentada? como está organizado o resto do relatório?

**Formulação do problema**

Defina o problema. Introduza a terminologia. Discuta as propriedades básicas.

**Solução do problema**

Enumere as coisas que fez e que considere importantes. Não seja modesto mas também não exagere.

A correcta avaliação do projecto é um aspecto crítico.

**Conclusões**

Procure dar resposta às seguintes questões: quais, se for caso disso, as lições aprendidas? o que, se algo, foi explicado? em que medida os objectivos foram atingidos? se existe algo que agora faria de forma diferente? quais as vantagens e desvantagens do trabalho realizado face a exemplos da literatura? que ideias para trabalho futuro?

**Referências**

A ideia subjacente à referência é que esta poupa papel e que o leitor poderá obter o documento em qualquer biblioteca científica razoável. Assim, é critério essencial referir revistas disponíveis em bibliotecas de instituições de ensino superior ou outras instituições profissionais. Em geral, não é razoável a referência a actas de conferências, dado que estas raramente estão acessíveis em bibliotecas pelo que, para todos os efeitos, estão indisponíveis. As referências a “Comunicações Privadas” são inaceitáveis. A informação dada deve ser sempre suficientemente detalhada por forma a que o leitor possa adquirir a publicação ou consultá-la numa biblioteca. Referências a dissertações de doutoramento ou outras devem indicar a instituição e o seu endereço. Sendo a referência essencial para o trabalho, no caso desta ser difícil de obter, dever-se-á incluir no documento, ou em apêndice, os seus pontos essenciais.

Cite uma referência sempre que está a incluir algo que não é contribuição sua ou quer indicar um conjunto de referências que o leitor pode consultar, mas cujo conteúdo não pode ser descrito adequadamente no relatório.

Autores Afonso Nobre, nº 44777

Ricardo Silva nº 44837