Relatório de Especificação: Desenho

[Os comentários e algum conteúdo exemplificativo incluídos no documento destinam-se a apoiar na sua preparação e estão indicados a verde. **Remover os comentários na versão final**.

Os *templates* dos relatórios fornecem uma abordagem consistente para organizar as informações de um projeto. Ajudam a lembrar os conteúdos que é necessário preparar. Não é preciso que se preencha o *template* sequencialmente, do início ao fim. Se determinadas seções de um *template* não são realmente necessárias, deve-se removê-las.]

|  |  |
| --- | --- |
| Projeto: | <nome do projeto aqui> |
| Preparado por: | Grupo xxxx  <nome aluno> (nº 999999)  <nome aluno> (nº 999999)  [ lista dos membros que realizaram (efetivamente) o trabalho] |
| Data de preparação: | Aveiro, 31 de fevereiro de 2017 |
| Circulação: | Docentes e Discentes de AMS. |

Controlo de versões

|  |  |  |
| --- | --- | --- |
| Quando? | Responsável | Alterações significativas |
| <data> | <quem alterou>? | <descrição das principais alterações/secções introduzidas> |
|  |  |  |
|  |  |  |

Índice

[1 Introdução 4](#_Toc530158302)

[2 Objetivos e restrições da arquitetura 7](#_Toc530158303)

[2.1 Requisitos-chave e restrições ao sistema 7](#_Toc530158304)

[2.2 Atributos de qualidade com impacto na arquitetura 7](#_Toc530158305)

[3 Modelo estrutural 9](#_Toc530158306)

[3.1 Arquitetura lógica da solução 9](#_Toc530158307)

[3.2 Arquitetura de instalação 9](#_Toc530158308)

[4 Modelo de comportamento 10](#_Toc530158309)

[4.1 Operações de sistema 10](#_Toc530158310)

[4.2 Integrações com sistemas externos 10](#_Toc530158311)

[5 Tecnologias de implementação 11](#_Toc530158312)

[6 Protótipos dos produtos 12](#_Toc530158313)

[7 Anexos 13](#_Toc530158314)

# Introdução

[Introdução ao documento: qual é o assunto do documento e porque o estamos a fazer?

Resumo dos principais resultados que podem ser apreciados no documento. ]

Índice de diagramas

[actualizar o índice quando o documento completo. Todos os diagramas devem ser legendados.]

[Diagrama 3: Modelo do domínio. **Error! Bookmark not defined.**](#_Toc417508552)

Índice de tabelas

[atualizar o índice quando o documento completo. Todas as tabelas devem ser legendadas.]

[Tabela 1: Principais benefícios do CourseRegistrarSystem. **Error! Bookmark not defined.**](#_Toc214204493)

# Objetivos e restrições da arquitetura

## Requisitos-chave e restrições ao sistema

[Identify issues that will drive the choices for the architecture such as: Will the system be driven by complex deployment concerns, adapting to legacy systems, or performance issues? Does it need to be robust for long-term maintenance?

Identify critical issues that must be addressed by the architecture, such as: Are there hardware dependencies that should be isolated from the rest of the system? Does the system need to function efficiently under unusual conditions? Are there integrations with external systems? Is the system to be offered in different user-interfacing platforms (web, mobile devices, big screens,…)?]

E.g.: (the references cited in [XX ] would be hypothetical links to previous specification documents/deliverables )

There are some key requirements and system constraints that have a significant bearing on the architecture. They are:

* The existing legacy Course Catalog System at Wylie College must be accessed to retrieve all course information for the current semester. The C-Registration System must support the data formats and DBMS of the legacy Course Catalog System [E2].
* The existing legacy Billing System at Wylie College must be interfaced with to support billing of students. This interface is defined in the Course Billing Interface Specification [E1].
* All student, professor, and Registrar functionality must be available from both local campus PCs and remote PCs with internet dial up connections.
* The C-Registration System must ensure complete protection of data from unauthorized access. All remote accesses are subject to user identification and password control.
* The C-Registration System will be implemented as a client-server system. The client portion resides on PCs and the server portion must operate on the Wylie College UNIX Server. [E2]
* All performance and loading requirements, as stipulated in the Vision Document [E2] and the Supplementary Specification [15], must be taken into consideration as the architecture is being developed.

## Atributos de qualidade com impacto na arquitetura

[Enumerar atributos de qualidade (naturalmente já formulados como requisitos não funcionais em relatórios anteriores) com impacto nas escolhas relativas à arquitetura. Só incluir os que contribuem para influenciar a arquitetura do sistema.]

E.g.:

* + 1. The desktop user-interface shall be Windows 95/98 compliant.
    2. The user interface of the C-Registration System shall be designed for ease-of-use and shall be appropriate for a computer-literate user community with no additional training on the System.
    3. Each feature of the C-Registration System shall have built-in online help for the user. Online Help shall include step by step instructions on using the System. Online Help shall include definitions for terms and acronymns.
    4. The C-Registration System shall be available 24 hours a day, 7 days a week. There shall be no more than 4% down time.
    5. Mean Time Between Failures shall exceed 300 hours.
    6. Upgrades to the PC client portion of C-Registration shall be downloadable from the UNIX Server over the internet. This feature enables students to have easy access to system upgrades.

# Modelo estrutural

## Arquitetura lógica da solução

[Como é que a solução de software vai estar organizada?

Explicar a organização prevista do software em módulos, camadas ou subsistemas. Para além dos diagrama, é necessário texto de explicação.

Sugestão: apresentar uma arquitetura lógica por camadas (layered architecture), visualizada com um diagrama de pacotes]

Na descrição da arquitetura, explicar as Tecnologias e ambientes e suporte que se prevê utilizar: web, mobile,…

## Arquitetura de instalação

[Explicar a organização prevista da solução em termos configuração de produção (*deployment*). Modelar num diagrama de *deployment*]

# Modelo de comportamento

## Operações de sistema

[utilizar Diagramas de Sequência de Sistema para identificar as operações de sistema. Nota: como explorado nas aulas TP, as operações se sistema identificam os pontos de entrada no sistema, a partir dos fluxos dos casos de utilização.

Caso o nr de casos de utilização seja extenso, pode-se fazer apenas para um subconjunto, justificando que o subconjunto considerado diz respeito aos casos de utilização a implementar na iteração imediata]

## Integrações com sistemas externos

[explicar as colaborações entre o nosso e outros sistemas de software, identificando (a interação com) a **troca de mensagens**., visualizada com diagramas de sequência]

# Tecnologias de implementação

[Explicar as tecnologias selecionadas para a implementação: linguagens de programação, *frameworks* de que se vai tirar partido, sistemas de comunicação, integração de sensores e dispositivos, etc. ]

# Protótipos dos produtos

[maquetas gerais dos produtos de software. Não só protótipos funcionais.

A ideia não é mostrar todos os *screens*, mas ilustrar as opções feitas

Deve ser acompanhado com referências/links para versões experimentáveis dos protótipos]

[maquetas de *outputs*, se relevante; *eg*: mapas estatísticos a produzir, relatórios, etc.]

# Anexos

[listar a documentação anexa, referida ao longo do relatório]

[listar ficheiros externos, nomeadamente versões eletrónicas do relatório, modelos UML a consultar ou sítios com as maquetas]

[anexar cópias de documentos chave relevantes para a documentação de requisitos, e.g., formulários]