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 para entrega**.

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](#_Toc24842696)

[2 Objetivos e restrições da arquitetura 7](#_Toc24842697)

[2.1 Objetivos da arquitetura face aos requisitos do domínio 7](#_Toc24842698)

[2.2 Atributos de qualidade com impacto na arquitetura 7](#_Toc24842699)

[2.3 Decisões tomadas e justificação 8](#_Toc24842700)

[3 Vistas de arquitetura 9](#_Toc24842701)

[3.1 Arquitetura lógica da solução 9](#_Toc24842702)

[3.2 Arquitetura operacional 9](#_Toc24842703)

[4 Modelo de comportamento 10](#_Toc24842704)

[4.1 Interação entre módulos 10](#_Toc24842705)

[4.2 Operações de sistema 10](#_Toc24842706)

[4.3 Integrações com sistemas externos 10](#_Toc24842707)

[5 Tecnologias e implementação 11](#_Toc24842708)

[6 Protótipos dos produtos 12](#_Toc24842709)

[7 Anexos 13](#_Toc24842710)

# 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

Um dos objetivos principais deste documento é o desenho do modelo arquitetural e do modelo de comportamento para a solução CityFoodPro que responda aos requisitos previamente desenhados. O outro objetivo, não menos importante, é a escolha das tecnologias para a implementação da solução CityFoodPro.

## Objetivos da arquitetura face aos requisitos do domínio

Existem alguns requisitos chave que influenciam a arquitetura. Eles são:

* A plataforma deve, caso ainda não esteja registado, pedir ao utilizador que se registe antes de fazer qualquer compra.
* A plataforma deve ser capaz de listar os estabelecimentos e pratos disponíveis, registar o pedido do cliente e confirmar o pagamento.
* A plataforma deve encarregar-se de encaminhar o pedido para que este seja atendido.

## Atributos de qualidade com impacto na arquitetura

Os atributos de qualidade da solução CityFoodPro encontram-se descritos no “E4-Relatório Técnico: Análise de Requisitos” [1].

Na Tabela 2 estão os requisitos não funcionais e que possuem impacto na arquitetura.

|  |  |  |
| --- | --- | --- |
| **#** | **Atributos de qualidade** | **Impacto na arquitetura** |
| RQua.1 | O layout deve ser intuitivo para permitir uma navegação fluida pelo site que facilite a interação do utilizador sem prévio ensino de como usar o site. | O desenvolvimento do site deve ter em conta as regras de usabilidade de forma a ser responsivo. |
| RQua.2 | Utilização de uma base de dados para guardar dados de utilizadores, etc. | O site deve ter uma base de dados para guardar todo o tipo de informações relevantes. |
| RQua.3 | A solução deve apresentar a informação geo-referenciada. | Incorporação de mapas para as zonas onde os clientes querem que o pedido seja entregue. |
| RQua.4 | Os vários níveis de resposta devem ser cumpridos para garantir um bom desempenho e uma boa experiência do cliente. | A arquitetura tem de permitir um nível de agilidade suficiente para cumprir com os horários definidos pelo utilizador. |
| RQua.5 | O sistema deve estar disponível 24h por dia, 7 dias por semana. | Os sistemas críticos devem possuir redundância. |

Tabela 2: Atributos de qualidade com impacto na arquitetura

## Decisões tomadas e justificação

Foi tomada a decisão de construir um website (em detrimento de outras opções, como uma aplicação web) pois é o meio de acesso mais standard e que vai abranger mais pessoas.

O site vai ser construído usando a linguagem PHP pois esta permite uma mais fácil ligação com a base de dados usada para guardar os dados dos clientes, assim como informações sobre eventos.

# Vistas de arquitetura

## Arquitetura lógica da solução

O sistema vai ser desenvolvido tendo como base um modelo Cliente-Plataforma com uma base de dados associada. O utilizador vai interagir com o sistema através de um conjunto de páginas Web e as suas interações vão resultar de interações com a plataforma e uma base de dados.

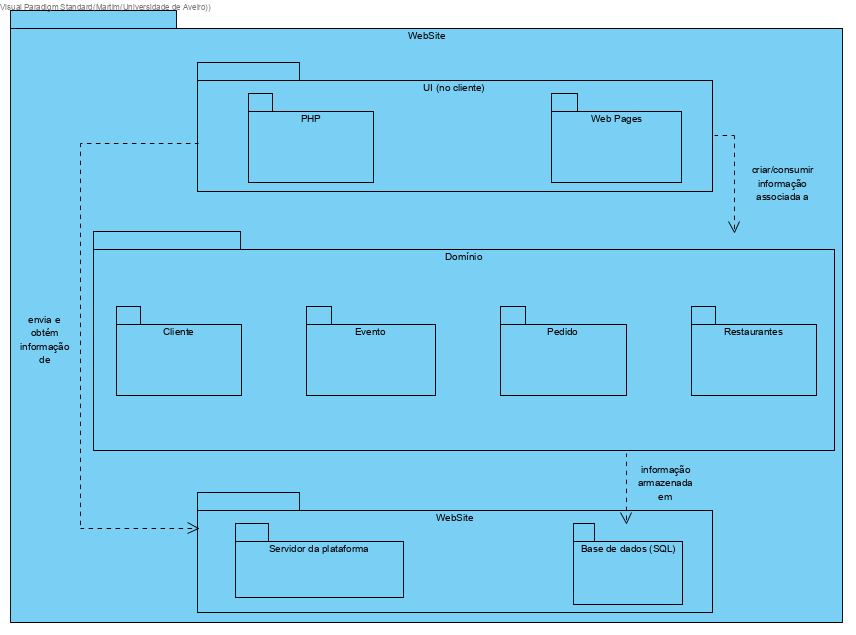
Em termos gerais, é possível identificar 3 grandes conjuntos de blocos de software na solução a desenvolver, como expresso no Diagrama 1: Modelo dos principais Pacotes do Sistema:

Diagrama 1: Modelo dos principais Pacotes do Sistema

Tendo em conta as relações entre pacotes evidenciadas no diagrama, é possível concluir que a arquitetura da solução vai ser constituída por 4 camadas principais.

A primeira camada, designada por *Front-End*, corresponde à camada de interacção com o utilizador e é constituída por um conjunto de páginas Web (PHP) de conteúdo semelhante aos *screens* do protótipo.

A segunda camada permite a ligação entre as camadas de *Front-End* e *Back-End* e permite a obtenção de informação de e para a base de dados associada ao *Back-End*.

A terceira camada, designada por *Back-End*, baseia-se numa base de dados relacional.

O *Back-End* é responsável pelo armazenamento da informação persistente associada aos Eventos e Utilizadores, através de uma base de dados relacional.

A quarta camada da arquitetura corresponde à base de dados relacional, onde são armazenadas as informações dos Pedidos e dos Utilizadores. As informações de autenticação (Utilizador e Palavra-Passe) também são armazenadas na base de dados.

## Arquitetura operacional

Diagrama 2: Diagrama de Deployment

# Modelo de comportamento

## Interação entre módulos

[Tomando como referência os módulos identificados no ponto 3.1, explicar as principais interações que acontecem entre os módulos, a alto nível. Podem ser usados D. de Sequência para mostrar a interação. Não é preciso ser exaustivo, mas deve ficar claro, para um programador que integre a equipa, como é que as partes definidas na arquitetura vão comunicar/colaborar.]

## 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.

Recomendação: pode-se fazer apenas para um subconjunto dos casos de utilização, 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 e implementação

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

[Explicar as atividades de validação da arquitetura que já foram desenvolvidas.

Referir, se aplicável, os recursos relevantes relacionados com os projetos feitos para validar a arquitetura proposta (e.g.: repositório git de código partilhado)]

# Protótipos dos produtos

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

A ideia não é mostrar todos os *screens*, mas incluir ilustrações representativas das opções tomadas

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

[maquetas de outros *outputs*, *eg*: mapas estatísticos a produzir, painéis de apoio à decisão, 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]