16-11-2021

Relatório Intermédio

Sara Silva - 30042

Conteúdo

[Introdução 2](#_Toc89200533)

[Objetivo 3](#_Toc89200534)

[Referências 3](#_Toc89200535)

[Análise do produto 3](#_Toc89200536)

[Planeamento 3](#_Toc89200537)

[Descrição do problema 5](#_Toc89200538)

[Requisitos Funcionais 5](#_Toc89200539)

[Requisitos Não Funcionais 5](#_Toc89200540)

[Arquitetura 6](#_Toc89200541)

[Calendarização 6](#_Toc89200542)

[Diagrama de Sequência 7](#_Toc89200543)

[Diagrama de Classes 8](#_Toc89200544)

[Restrições gerais 8](#_Toc89200545)

[Testes Unitários 8](#_Toc89200546)

[Conclusão 9](#_Toc89200547)

# Introdução

Pretende-se realizar um projeto tendo em conta o conhecimento adquirido nas aulas de Engenharia de Software. Este trabalho consiste no desenvolvimento de todas as etapas de um projeto de software que seja capaz de armazenar e gerir informação, focando no planeamento, especificação, modelização, implementação, testes e documentação do projeto.

A aplicação vai ser apresentada como requisito de avaliação à disciplina de Engenharia de Software.

# Objetivo

Neste projeto pretende-se que desenvolvam um Web Service REST que suporte a gestão de restaurantes que uma empresa executa para seus clientes. Um determinado cliente pode ter uma reserva e cada restaurante é constituído por muitas tarefas. Cada tarefa é executada por um empregado.

O cliente, deverá ser capaz de usar o Web Service para consultar informações sobre as suas reservas, especialmente prato do dia, sobremesa. O Web Service deve ser capaz de gerir as operações necessárias para as funcionalidades descritas.

# Referências

Slides das aulas teóricas de Engenharia Software

# Análise do produto

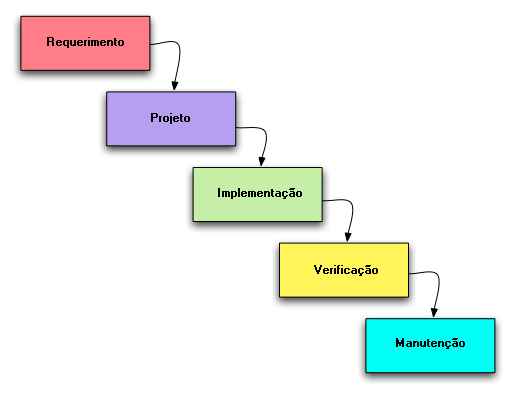
Este tipo aplicação existe para que seja mais fácil a sua gestão, procura gerir reservas e as respetivas tarefas, assim é mais fácil a consulta da informação. Está concentrada e pode ser acedida através do Web Service principal.

# Planeamento

O modelo escolhido para o projeto é o modelo Waterfall (cascata). O modelo Waterfall descreve um método de desenvolvimento que é linear e sequencial.

A vantagem de desenvolvimento em cascata é que ele permite a departamentalização e controlo de gestão. A programação pode ser definida com prazos para cada fase de desenvolvimento e um produto pode prosseguir no processo de desenvolvimento e ser entregue no prazo especificado.

Cada fase do desenvolvimento procede em ordem, sem qualquer sobreposição ou etapas passadas à frente.



# Descrição do problema

## Requisitos Funcionais

|  |  |  |
| --- | --- | --- |
| **Parte do Restaurante** | | |
| UR000 | Permitir registos utilizadores | Obrigatório |
| UR001 | Autenticação utilizadores | Obrigatório |
| UR002 | Editar dados utilizadores | Obrigatório |
| UR003 | Inserção Restaurantes | Obrigatório |
| UR004 | Permitir características de cada restaurante | Obrigatório |
| UR005 |  | Obrigatório |

|  |  |  |
| --- | --- | --- |
| **Parte do Utilizador** | | |
| UR000 | Pesquisa por restaurantes | Obrigatório |
| UR001 | Permitir reservar | Obrigatório |
| UR002 | Cancelamento de reservas | Obrigatório |
| UR003 |  | Obrigatório |

## Requisitos Não Funcionais

|  |  |  |
| --- | --- | --- |
| UR000 | Interface gráfica | Obrigatório |
| UR001 | Garantir a segurança de todos os dados relativos aos utilizadores | Obrigatório |
| UR002 | Garantir o recurso a protocolos de segurança no processo de login | Obrigatório |
| UR003 | Sincronização com as bases de dados dos clientes, empregados, etc. | Obrigatório |
| UR004 |  | Obrigatório |
| UR005 | O sistema deve estar disponível em pelo menos dois idiomas, PT e ENG | Obrigatório |

# Arquitetura

Tendo em conta que vamos ter um sistema interativo onde o núcleo funcional do programa é independente da interface do utilizador, o padrão escolhido para projeto é o Model-view-controller (MVC).

Neste padrão temos uma divisão da aplicação em processamento, input e output. As ideias centrais por detrás do MVC são a reutilização de código e separação de conceitos.



# Calendarização

# Diagrama de Sequência

# 

# Diagrama de Classes

# Restrições gerais

Alguns dos critérios de qualidade poderão depender de outros fatores: periodicidade de utilização de dados do serviço de dados, existência de conectividade, etc...

Supõe-se que o serviço de dados tenha uma disponibilidade próxima de 100%, e que a largura de banda disponível seja suficiente para transferir todos dados necessários.

# Testes Unitários

Podem ser encontrados na pasta Tests/Unit alguns dos testes unitários realizados.

# Conclusão

No final do projeto, concluímos que não optamos pelo modelo mais correto para a implementação do projeto. O modelo escolhido foi o de cascata que, é um risco e pode ser um convite a falhas, como foi verificado no projeto, ou seja, requeria um planeamento perfeito antes de se partir para a implementação do código. Pode-se dizer então, que o modelo em cascata é um modelo inflexível.

Para o projeto, poderia ter optado por outros modelos como o de prototipagem ou ágil, que são mais flexíveis e que seriam melhores para engenheiros que não sejam muito experientes.