



Gorgeous Food

ARQSOFT

2019 / 2020

Grupo 204

Filipe Ferreira | 1160826

Pedro Ferreira | 1140953

Pedro Santos | 1120653

Índice

1	<i>Introdução</i>	4
1.1	<i>Drivers</i>	4
1.2	<i>Casos de uso</i>	5
2	<i>Tecnologias Utilizadas</i>	10
3	<i>ADD</i>	11
3.1	<i>Iteração 1</i>	11
3.2	<i>Iteração 2</i>	12
3.3	<i>Iteração 3</i>	13
4	<i>Análise e avaliação da solução (ATAM)</i>	16
5	<i>Referências</i>	18

1 Introdução

O presente documento descreve o desenvolvimento de uma aplicação capaz de fornecer um serviço de refeições comercializadas pela cantina de uma escola. Este serviço é denominado de *Gorgeous Food* e, o seu desenvolvimento assenta nos moldes da arquitetura *Attribute-Driven Design* (ADD).

Este capítulo apresenta a contextualização da aplicação desenvolvida, os *drivers* fornecidos pelos *stackholders*, casos de uso identificados pela equipa de desenvolvimento, bem como considerações determinantes à continua expansão do serviço. Esta dividiu o processo de implementação da aplicação em três iterações¹.

1.1 Drivers

Os seguintes tópicos são fornecidos pelos interessados à criação do serviço, num documento ilustrativo:

Nº	Descrição
1	The application to be developed must be accessible from a variety of platforms using a web browser
2	This application must be developed by the school's technical staff and should be available in two months. Within five weeks, adding and removing new items to inventory should be demonstrated to many stakeholders. Not only because of these factors, inventory update functionalities are especially relevant to the organization, and very difficult for the team.
3	Multilanguage is desirable as the school accepts foreign students on a regular base, and the service will probably attract tourists, but not for what is used only by school staff. Multilanguage support is considered by the client to be of medium importance, and its difficulty for the developing team is insignificant.
4	The team is free to choose the technological stack to use in the project, but, for cost reasons, only open-source technologies can be adopted.
5	There are several meal descriptors (per 100 g) that are initially considered as calorie, for instance, but others may be considered in the future. Thus, it is intended that the introduction of new one's aspects very little, or nothing, other components of the application. The addition of a new descriptor should be completed and tested in less than 0.5 person-day of effort. This scenario is considered of great importance for the school, and the degree of its technical risk is high.
6	Changes in the meal identification number generation should not aspect more than one component. This is considered very important for the organization as there are

¹ Repositório *Gorgeous Food*: <https://bitbucket.org/mei-isep/arqsoft-19-20-ifp204/src/master/>

	here some uncertainties and, in the future, some special equipment might be used for the label printing and reading, with the introduction of some requirements and changes. A change in the generation of this number identification is to be implemented with no side effects in 1 person-day of effort. This scenario is considered of high importance for the success of the application and the degree of its technical risk is medium.
7	The application should be able to provide management reports, such as the number of meals, grouped by type, sold at certain times of the day (14:30 to 17, for instance) or days of the week. Due to the composition of advisory and management bodies, it should be possible to generate reports in more than one language. This scenario is considered of high importance for the success of the application and the degree of its technical risk is medium.
8	Some architectural specific concerns: authentication, authorization, and activity log. Administrator should be able to query user logs and see their activities, all of them, or performed between dates or only some type of activities.
9	The school has a database server with a backup policy implemented and it is to be used by the application to be developed. In addition, there is an user directory server used by other applications in the school and the new application is also to use it.

Tabela 1 - Drivers fornecidos pelos stakeholders.

1.2 Casos de uso

Ator	Caso de Uso	Descrição
CT	UC1 - Add Meal Item to Inventory	The cooking team/staff can create a item with a specific meal. The ingredients are associated with the meal.
CT	UC2 - Remove Meal Item from Inventory	The cooking team/staff can remove a item of a specific meal.
SU	UC3 - Consult the available meals	Students and Other interested people can consult the available meals.
CT	UC4 - Consult Inventory Meal Number	Cooking team/staff can consult the inventory of the existing meals.

SU	UC5 - Consult Specific Meal Location.	Students and other starving people can consult the location of their preferred meals.
CT	UC6 - Add new Meal Descriptor	The staff can create meal descriptions
CT	UC7 - Remove Meal Descriptor	The staff can remove meal descriptions
CT	UC8 - Generate management report	The staff can request to the system management reports about how many meals and their information.
CT	UC9 - Update identification number generator.	The staff can update the identification generator, so that the number generator meets the specified characteristics.
CT, CA	UC10 - Login	Want to login.
CT, CA	UC11 - Logout	Want to logout.
CA	UC12 – See User Activity Log.	The administrator can query the system to consult logs about the activity on the application.
CA, CT, SU	UC13 - Change App Language.	The users want to use the application on several languages.
CT	UC14 - Add/Update new Meal	The cooking team/staff can create a meal and create/associate the ingredients.
CT	UC15 - Remove Meal	The cooking team/staff can remove a meal.

Tabela 2 - Tabela de casos de uso. Atores SU (ServiceUsers), CT (Cooking Team), CA (Canteen Admin).

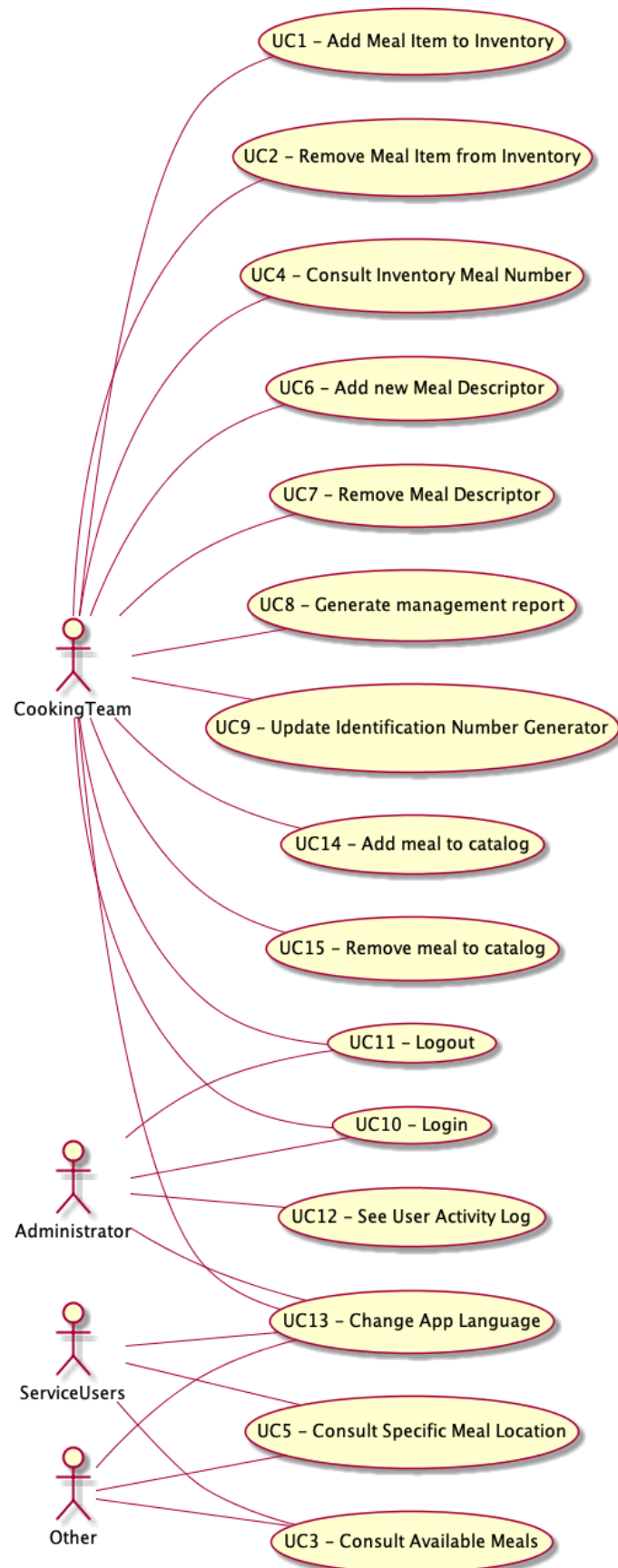


Figura 1 - Diagrama de Casos de Uso.

ID	Quality Attribute	Scenario	Associated UC	Priority	Dev Difficulty
QA-1	Usability	The system displays a variety of L languages.	All	Medium	Low
QA-2	Modifiability	When the generator is edited, the existing components should not be affected.	UC9	High	Medium
QA-3	Modifiability	The addition of a new language should not affect the system (negatively).	All	Medium	Low
QA-4	Compatibility	Run on several browsers, on several devices.	-	High	Medium
QA5	Usability	The admin requests a report with a different language.	UC13	High	Medium
QA6	Modifiability	A new meal descriptor is added or removed and other components of the application are not affected.	UC6 and UC7	Medium	High

Tabela 1 - Quality Attributes.

ID	Description
CT-1	The system is developed and maintained using Open-Source Technologies, without requiring paying licenses.
CT-2	The application should be available in two months.
CT3	Within five weeks, adding and removing new items to inventory should be demonstrated.
CT4	The addition of a new descriptor should be completed and tested in less than 0.5 persons day of effort.
CT-5	A change in the number identification generator should not take more than 1 person-day of effort.
CT6	The application must be accessible from several browser (ex: Chrome, Firefox) in different platforms (ex: Windows, Linux, OSX).

Tabela 2 – Constrains.

ID	Concerns
AC-1	The application should use the school database server.
AC-2	The application should use the school user directory server.
AC-3	Take advantage of the team's technical knowledge of C# programming language.

Tabela 3 - Architectural Concerns

2 Tecnologias Utilizadas

Este capítulo visa apresentar as tecnologias utilizadas, bem como tecnologias consideradas no desenvolvimento da solução.

Todas as tecnologias utilizadas no desenvolvimento da solução são *open-source*, não implicando o pagamento de qualquer licença

Componente	Tecnologia
Gorgeous Food User Interface	React
Gorgeous Food Application Programming Interface	.NET Core + Entity Framework Core
Meal Database	Sql Server 2019
IDE	Visual Studio Code
Other	Windows terminal + Microsoft Powershell

Tabela 4 - Tecnologias utilizadas no desenvolvimento do sistema

3 Attribute-Driven Design (ADD)

3.1 Iteração 1

Esta iteração tem como objetivo criar a arquitetura da estrutura inicial do sistema a ser desenvolvido, considerando que este é do tipo *Greenfield*. Todos os requisitos foram considerados para a estrutura global do sistema. Consequentemente, foi proposta a arquitetura de referência “Basic Web Application in Azure” (Figura 2) bem como produzidas a “Deployment View” (Figura 3) e a “Server Components View” (Figura 4).

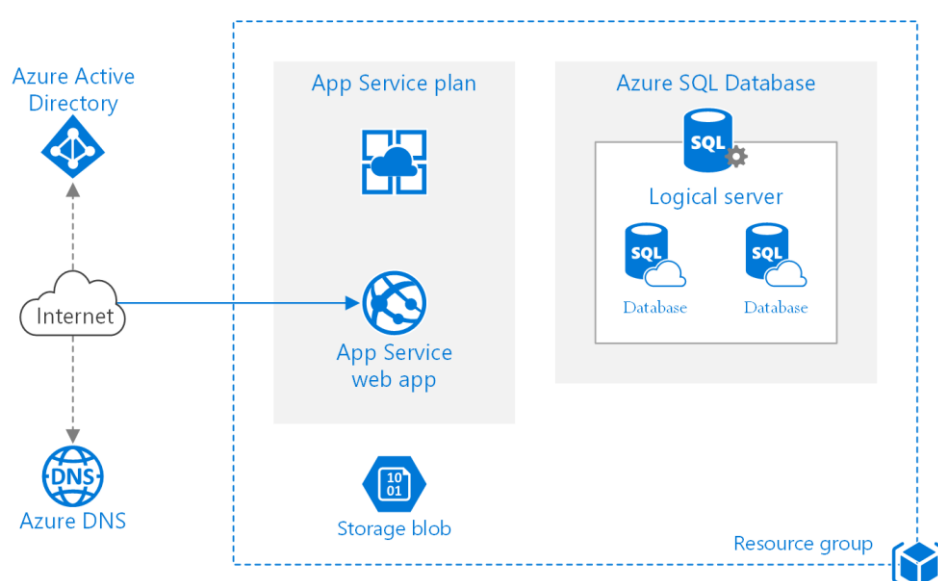


Figura 2 – Arquitetura de referência “basic web application in Azure” [1].

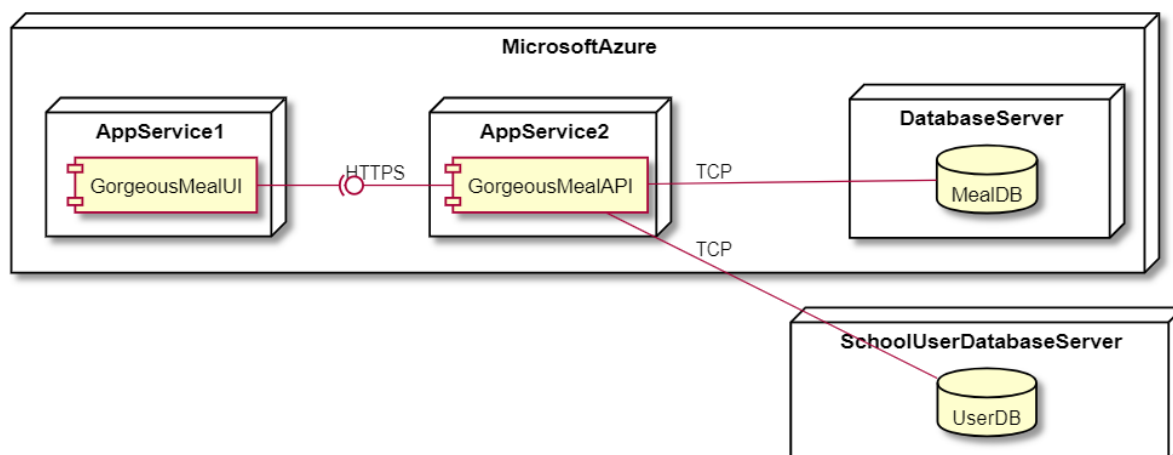


Figura 3 – Deployment View

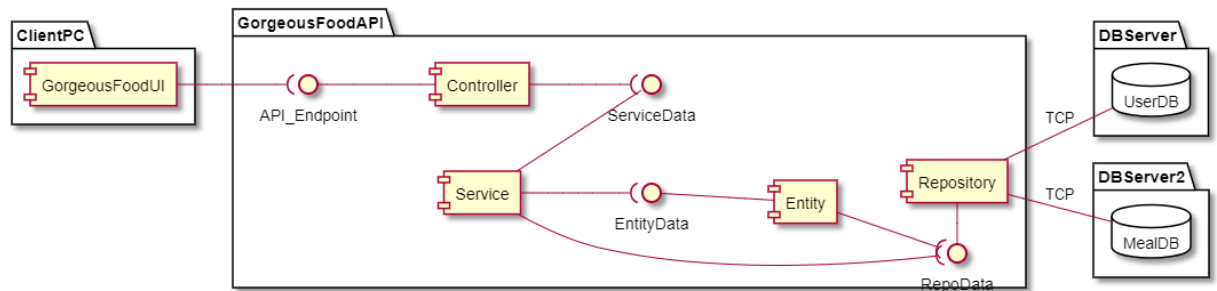


Figura 4 – Server Components View.

3.2 Iteração 2

O foco desta iteração são as funcionalidades base do sistema bem como os principais requerimentos para o protótipo do sistema. O padrão de design escolhido para orientar a arquitetura do software, nesta iteração, foi o “*Domain-Driven Design*”. Os caso de uso e limitações **UC1**, **UC2**, **UC14**, **UC15**, **AC-3**, **CT-6** e **CT-1** foram escolhidos para refinamento. Assim, foi produzido o modelo de domínio Figura 5 e deliberadas as tecnologias a usar no desenvolvimento do sistema Tabela 5.

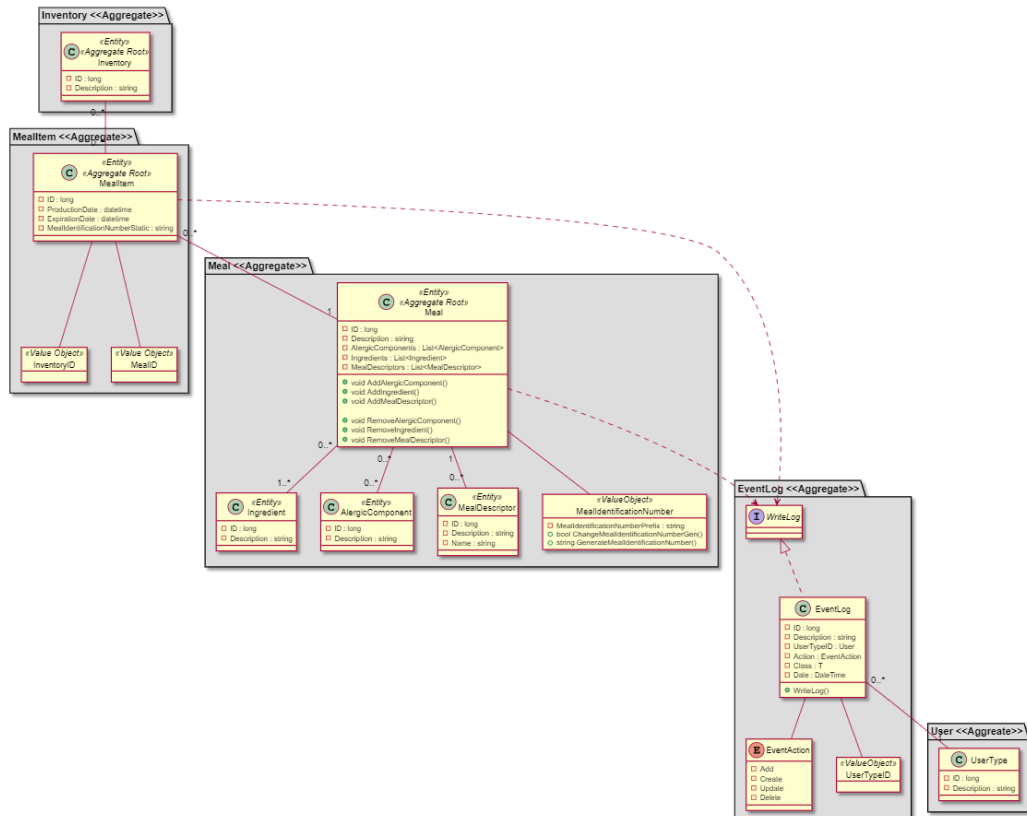


Figura 5- Modelo de domínio

Todas as tecnologias utilizadas no desenvolvimento da solução são *open-source*, não implicando o pagamento de qualquer licença obedecendo assim ao artefacto **CT-1**.

Componente	Opções	Escolha	Razão
Linguagem de Programação	Java, C#, PHP	C#	A equipa está familiarizada com a linguagem AC-3
Framework para desenvolvimento	ASP.NET API, ASP.NET MVC, ASP.NET Core, SignalR, Yii, Zend	ASP.NET Core	As capacidades da ferramenta enquadram-se com os principais objetivos do sistema a desenvolver. Alguns elementos da equipa já trabalharam com esta framework AC-3 .
Persistência	sqlServer, mongoDB, Mysql	sqlServer	Tendo em conta que a base de dados escolhida é do tipo relacional, sqlServer é o sistema de gestão de base de dados que melhor se integra no pacote de ferramentas escolhidas para o desenvolvimento do projeto.

Tabela 5 - Tabela de tecnologias deliberadas

3.3 Iteração 3

O objetivo desta iteração é refinar os casos de uso necessários implementar para o protótipo. Assim foram contemplados os casos de uso **UC-1**, **UC-2**, **UC-3**, **UC-4** bem como o atributo de qualidade **QA-4**. Foram produzidos os diagramas de sequência Figura 6 e Figura 7 para demonstrar as interações do sistema.

Para facilitar a integração de novos membros ao sistema, foram utilizadas um conjunto de práticas para facilitar a integração destes mesmos tais como a nomenclatura de classes como estão presentes na Figura 8.

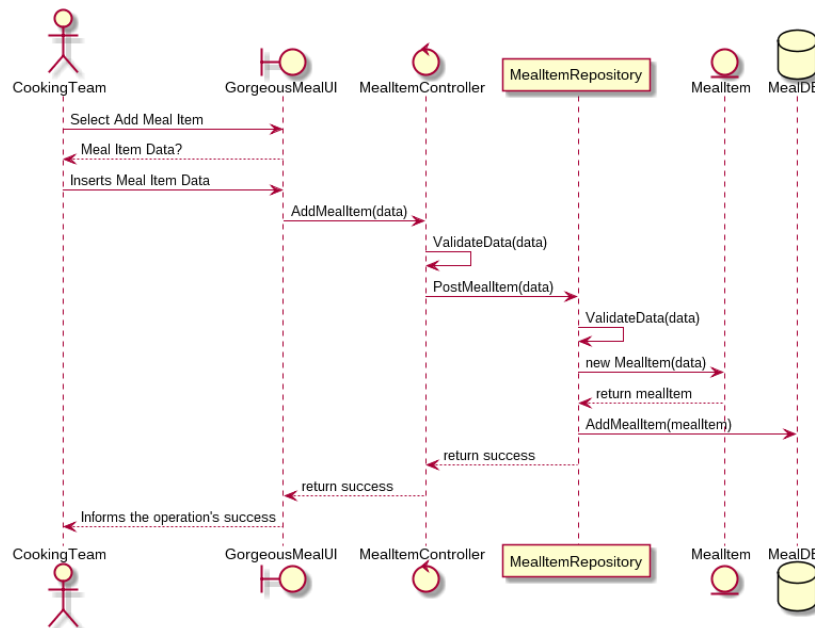


Figura 6 – Diagrama de sequência UC-1.

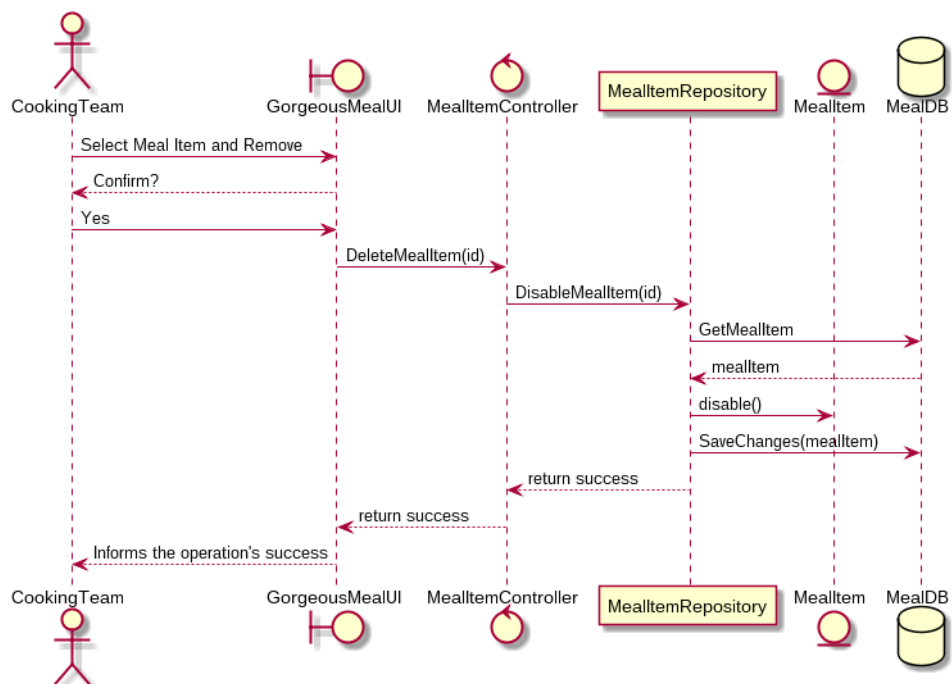


Figura 7 - Diagrama de sequência UC-2.



Figura 8 - Estrutura de Classes utilizada.

4 Análise e avaliação da solução (ATAM)

Cenário: QA-1	O Sistema disponibiliza múltiplas línguas.
Atributo	Usabilidade
Ambiente	Normal funcionamento do sistema
Estímulo	O utilizador muda a linguagem da aplicação.
Resposta	100% da informação do sistema muda para a linguagem selecionada.
Resposta em caso de falha	O sistema mostra as informações na linguagem <i>default</i> (inglês)
Raciocínio	Assegurar que toda a informação disponibilizada pelo sistema fica disponível na linguagem selecionada sem que nenhum erro ocorra.
Risco	Baixo

Cenário: QA-2	Quando o gerador de identificadores muda não afeta outros componentes.
Atributo	Modificabilidade
Ambiente	Normal funcionamento do sistema
Estímulo	O administrador modifica o gerador de identificadores.
Resposta	O sistema passa a usar o novo gerador de identificadores.
Resposta em caso de falha	O sistema usa o último gerador de identificadores guardado.
Raciocínio	Quando o gerador de identificadores é alterado, os restantes componentes do sistema não são afetados. As novas requisições ao “gerador de identificadores” passam a usar a mais recente referência.
Risco	Médio

Cenário: QA-3	A adição de uma linguagem nova não afeta negativamente o sistema.
Atributo	Modificabilidade
Ambiente	Atualização do sistema
Estímulo	O desenvolvedor adiciona uma nova linguagem ao sistema.
Resposta	O sistema passa a disponibilizar a linguagem adicionada.
Resposta em caso de falha	O sistema não disponibiliza a linguagem que despoletou a falha.
Raciocínio	Quando ao sistema é adicionado uma nova linguagem, este passa a disponibilizar essa linguagem. Todas as informações disponibilizadas pelo sistema devem ser suportadas na linguagem.
Risco	Médio

Cenário: QA-4	A aplicação deve ser executável em diversos <i>browsers</i> em diversos dispositivos.
Atributo	Compatibilidade
Ambiente	Normal funcionamento do sistema
Estímulo	A aplicação encontra-se em funcionamento de uma plataforma (servidor) e acessível aos utilizadores.
Resposta	A execução da aplicação é feita através de <i>browsers</i> .

Raciocínio	A aplicação possui um domínio (link), que permite o acesso por qualquer interessado com um dispositivo executador de <i>browsers</i> .
Risco	Baixo

Cenário: QA-5	O administrador requisita um relatório do sistema com uma linguagem diferente.
Atributo	Usabilidade
Ambiente	Normal funcionamento do sistema
Estímulo	O administrador do serviço pretende obter um relatório do sistema numa diferente língua.
Resposta	O sistema deve gerar o relatório na linguagem requerida pelo utilizador.
Resposta em caso de falha	O sistema gera um relatório na língua <i>default</i> (inglês neste caso).
Raciocínio	A linguagem deve estar previamente configurada no sistema. O utilizador requisita o relatório na determinada linguagem. O sistema apresenta o relatório.
Risco	Alto

Cenário: QA-6	Um novo descritor de refeição (<i>meal</i>) é adicionado ou removido e outros componentes do sistema não são afetados.
Atributo	Modificabilidade
Ambiente	Normal funcionamento do sistema
Estímulo	Os gestores do sistema pretendem adicionar/remover descritores às refeições (<i>meals</i>).
Resposta	O sistema altera o descritor e informa o sucesso da operação.
Raciocínio	O novo descritor é associado á refeição (<i>meal</i>), logo todos os itens de refeição (<i>meal itens</i>) criados após a alteração, serão instanciados com o novo descritor.
Risco	Baixo

5 Referências

- [1] “Basic Web App.” [Online]. Available: <https://docs.microsoft.com/en-us/azure/architecture/reference-architectures/app-service-web-app/basic-web-app>. [Accessed: 05-Nov-2019].