

# Planeamento e Gestão de Projeto

Enunciado do Projeto — 2022/2023

## Introdução

Pretende-se fazer a análise, desenho inicial, e planeamento de um projeto de desenvolvimento de um sistema de informação sobre o tema de **suportar a comunidade local**, abordado em Conceção de Produto.

São valorizados trabalhos cuja análise demonstre uma boa compreensão do problema, cujo desenho inicial tenha as bases para uma solução adequada e fazível com tecnologias existentes e com integração com sistemas externos, e cujo planeamento tenha em conta as pessoas da equipa, a análise e desenho inicial do produto, o processo de desenvolvimento, e a calendarização das tarefas do projeto.

O planeamento vai ser executado no próximo semestre, pelo mesmo grupo, em Projeto de Tecnologias de Informação (PTI) e Projeto de Tecnologias de Redes (PTR), pelo que se incluem aqui elementos essenciais ao sucesso nessas duas disciplinas.

## Descrição inicial do sistema

O sistema deve informar os utilizadores de um mercado *online* sobre o **impacto das suas encomendas na comunidade onde reside**, devendo ajudar a escolher as opções que **favoreçam fornecedores locais**. Para o mesmo **produto**, podem existir **vários fornecedores**, uns na mesma freguesia, outros no mesmo município, ou distrito, ou país, ou continente, até fornecedores noutro continente, logo com um impacto muito reduzido ou nulo na comunidade local.

## Atores

Os atores que interagem com o sistema são, pelo menos, os seguintes:

- **Consumidor:** encomenda produtos a fornecedores.
- **Fornecedor:** informa sobre a gama de produtos e satisfaz as encomendas dos consumidores.

Para além destes atores, são habitualmente considerados **administradores de sistema** bem como **sistemas externos** que oferecem serviços de **autenticação de utilizadores**, **pagamentos online**, **mapas digitais**, entre outros.

## Requisitos de informação

Alguns dados a registar são comuns aos **consumidores e fornecedores**, nomeadamente: **nome**, **morada** (e respetivas **coordenadas geográficas**), **identificador fiscal**, e **telemóvel de contacto**. Mas a realidade pode ser mais complexa se um mesmo fornecedor tiver unidades de produção em locais distintos.

Todos os **produtos** de um fornecedor têm um **nome**, **data de produção**, **preço**, e uma ou mais **categorias**, devendo também ser suportados **vários níveis de subcategorias**, para facilitar a navegação nos dados. *Dependendo da categoria, podem existir mais ou menos campos de dados para preencher.* Por exemplo, os produtos alimentares têm um **prazo de validade**, enquanto os **brinquedos a pilhas** têm o **número e tipo de pilhas** necessários ao seu funcionamento. O mesmo produto pode ter **vários fornecedores**, como nos casos da fruta e de hortícolas.

Quando o consumidor está a escolher produtos para fazer uma **encomenda**, deve ter acesso à **distância a que os fornecedores estão**, bem como **se estão situados na mesma freguesia, município, distrito, país, ou continente**. Uma encomenda inclui também a **quantidade de cada produto**, o **preço a pagar**, e a **data**.

Naturalmente, podem ser considerados mais dados, especialmente **se forem suportados por resultados de pesquisas sobre este tema**.

## Requisitos funcionais

O sistema deve suportar as funcionalidades indicadas a seguir, sendo que, em geral, as mais avançadas estão disponíveis apenas para utilizadores autenticados.

### Utilizador não autenticado<sup>1</sup>

- RF-01: Navegação pelos produtos através da hierarquia de categorias.
- RF-02: Visualização de produtos e seus fornecedores.
- RF-03: Pesquisa de produtos através dos *campos comuns a todos os produtos*.
- RF-04: Colocação, consulta, e remoção de produtos num cesto de compras.
- RF-05: Criação de uma conta no sistema.
- RF-06: Fazer *login* no sistema.

### Utilizador autenticado

- RF-07: Edição dos dados e remoção da sua conta no sistema.
- RF-08: Pesquisa de produtos através dos *campos específicos das categorias*.
- RF-09: Visualização do histórico de encomendas<sup>2</sup> e seus detalhes.
- RF-10: Comparação de dois produtos, com as diferenças em destaque. \*

### Consumidor autenticado

- RF-11: Encomenda dos produtos no cesto de compras.
- RF-12: Pagamento de encomenda recorrendo a um sistema externo.
- RF-13: Cancelamento de encomenda, desde que dentro de um dado prazo.

<sup>1</sup> Os requisitos RF-01 a RF-04 também se aplicam aos restantes tipos de utilizadores.

<sup>2</sup> No caso do fornecedor, trata-se das encomendas que recebeu, criadas pelos consumidores.

- RF-14: Notificação sobre a saída de produtos encomendados de um fornecedor.\*
- RF-15: Notificação sobre a chegada iminente de encomenda.\*
- RF-16: Visualização de relatório do impacto local<sup>3</sup> das suas encomendas.
- RF-17: Exportação dos dados das encomendas para ficheiros JSON.\*

#### Fornecedor autenticado

- RF-18: Criação, gestão, e remoção de unidade de produção.
- RF-19: Criação, gestão, e remoção de produto, e ligação a unidade de produção.
- RF-20: Visualização de unidade de produção e dos seus produtos.
- RF-21: Criação, edição, e remoção de veículo de transporte de produtos.
- RF-22: Notificação sobre encomenda de consumidor.
- RF-23: Visualização de encomenda de consumidor.
- RF-24: Colocação de produto encomendado em veículo de transporte disponível.
- RF-25: Registo de saída de veículo de transporte com produtos encomendados.\*
- RF-26: Registo de chegada iminente de encomenda a casa do consumidor.\*
- RF-27: Visualização de relatório do impacto local<sup>3</sup> das vendas dos *seus* produtos.

#### Administrador autenticado

- RF-28: Desativação e reativação de qualquer conta de utilizador no sistema.\*
- RF-29: Visualização de relatório do impacto local<sup>3</sup> das vendas de produtos.

Os requisitos assinalados com o símbolo ‘\*’ têm menor prioridade que os restantes. A lista de requisitos funcionais pode sofrer mudanças. Aliás, os grupos são incentivados a acrescentar requisitos para distinguirem o seu sistema dos restantes.

### Requisitos não funcionais

O desenho inicial do sistema e o planeamento do projeto devem ter em conta os requisitos não funcionais mencionados a seguir.

#### Obrigatórios

- **RNF-01: Arquitetura cliente-servidor Web.** Para aceder ao sistema, os utilizadores devem usar um *browser Web* que comunica com um servidor aplicacional. Por sua vez, o servidor deve ter um *front end* e um *back end*, que são aplicações independentes que interagem na prestação de serviços ao exterior. O *front end*

<sup>3</sup> O relatório deve mostrar a distribuição dos produtos encomendados pela proximidade entre consumidores e fornecedores usando gamas de distâncias (ex. até 10 km, ..., mais de 1000 km) ou a hierarquia freguesia, município, distrito, país, continente, e mundo. Deve ser permitida a análise dos dados em função das categorias de produto e do tempo (ex. só os produtos da categoria fruta, e/ou apenas os encomendados no ano passado).

implementa uma API, fornece conteúdos HTML, CSS e JavaScript, e trata da autenticação/autorização, enquanto o *back end* implementa a lógica de negócio e o modelo, que abstrai as operações disponibilizadas pela base de dados. Ambos podem ser decompostos em aplicações mais pequenas, dependendo do projeto.

- **RNF-02: Coerência dos dados.** O sistema deve garantir a coerência e integridade dos dados, de acordo com as regras do negócio.
- **RNF-03: Usabilidade.** As funcionalidades do sistema devem ser oferecidas de forma eficaz e eficiente, proporcionando uma experiência de utilização satisfatória. O sistema deve *adaptar as funcionalidades consoante o tipo de dispositivo* do utilizador, nomeadamente ecrãs de diferentes tamanhos e formas de interação.
- **RNF-04: Fonte aberta.** O sistema deve ser de código fonte aberto.

### Importantes

- **RNF-05: Ambiente de desenvolvimento.** Deve ter ferramentas de controlo de versões que suportem uma equipa de programadores. Adicionalmente devem ser usados mecanismos de automação que facilitem o desenvolvimento da aplicação.
- **RNF-06: Lançamento em produção.** O sistema deve ter um automatismo que permita lançá-lo em produção a partir de um repositório de código.
- **RNF-07: Tempo de resposta e desempenho.** O sistema deve ter um tempo de resposta e um desempenho adequados à expectativa dos utilizadores.
- **RNF-08: Interoperabilidade.** O sistema deve ter uma API REST que lhe permita a interação com outras aplicações.
- **RNF-09: Escalabilidade.** O sistema deve ter uma capacidade ajustável para atender pedidos. Pode ser horizontal, acrescentando nós, e/ou vertical, adicionando recursos aos nós existentes. Um aspeto complementar é a elasticidade com que o sistema adapta automaticamente a sua capacidade em função da carga, aumentando ou diminuindo o número de nós (dentro de limites pré-definidos).
- **RNF-10: Tolerância a falhas.** O sistema deve tolerar a falha de qualquer dos seus componentes, através de redundância.
- **RNF-11: Segurança.** O sistema e as redes devem estar cobertas por políticas de segurança que fazem uso de mecanismos de proteção de modo a impedir o acesso não autorizado a recursos e dados e também impedir a disrupção do serviço.
- **RNF-12: Testes.** O sistema deve ser desenvolvido com uma política integrada de testes que permita ir testando à medida que se adicionam novos componentes ou que se alteram componentes existentes.

### Outros

- **RNF-13: Salvaguarda de dados.** Os dados do sistema devem ser salvaguardados regularmente conforme indicado numa política de *backup*. Adicionalmente, deve estar definido um procedimento de restauro de qualquer parte dos dados.

- **RNF-14: Custo.** Deve ser escolhida uma plataforma de computação que preste o melhor serviço pelo preço mais baixo, idealmente de forma gratuita.
- **RNF-15: Proteção de dados.** O sistema deve disponibilizar informação para os utilizadores sobre a forma como os dados são tratados e a possibilidade de consulta e eliminação dos seus dados pessoais.

Cada grupo deve escolher os requisitos não funcionais a serem concretizados pelo seu sistema, especificando, para cada um, *o que deve ser satisfeito* pela solução. Por exemplo, a usabilidade pode ser avaliada com testes SUS, e a pontuação mínima desejada pode ser cerca de 85, correspondente a excelente.<sup>4</sup>

No final da implementação do sistema, a satisfação dos requisitos não funcionais deve ser comprovada, por exemplo através de testes à API, testes de carga, análise de vulnerabilidades, testes de usabilidade com utilizadores, e outros meios.

## Realização

O trabalho do projeto está organizado em duas etapas, com os objetivos indicados a seguir, sendo que, em geral, os diagramas seguem a notação UML.<sup>5</sup>

### Etapa 1: até 4 de novembro

Tem como propósito produzir artefactos relativos às seguintes atividades:

- **Modelação:** desenhar um diagrama de *classes* com o esquema de dados e restrições de integridade, diagramas de *atividades* para descrever processos de negócio, e um diagrama de *implementação* com a arquitetura do sistema. Também podem ser elaborados artefactos que o grupo achar relevantes, tais como diagramas de casos de uso, de estados, e protótipos da interface com o utilizador.
- **Pesquisa:** escolher, de forma justificada, uma *framework* de desenvolvimento de aplicações *Web*, e serviços externos para fazer autenticação de utilizadores, mostrar mapas e calcular distâncias, efetuar pagamentos *online*, entre outros que possam vir a ser necessários. A justificação de cada decisão deve incluir uma lista de vantagens, de preferência comparando com um produto concorrente.
- **Desenvolvimento:** demonstração de protótipo da funcionalidade de pesquisa de produtos, RF-03, para começar a formar uma opinião sobre como se trabalha com a *framework* escolhida. Se a opinião for negativa, o grupo pode começar a tomar medidas de mitigação, que, no limite, podem levar à adoção de outra *framework*.
- **Planeamento:** identificar as principais tarefas de desenvolvimento do sistema e fazer um esboço da calendarização, por exemplo, ao mês ou à quinzena, sem ser necessário indicar que pessoas fazem o quê. Este planeamento preliminar serve

<sup>4</sup> Bangor, A., Kortum, P., & Miller, J. (2009). Determining what individual SUS scores mean: Adding an adjective rating scale. *Journal of Usability Studies*, 4(3), 114–123.

<sup>5</sup> Fowler, M. (2005). UML distilled: A brief guide to the standard object modeling language (3ª edição). Addison-Wesley.

para identificar lacunas ou outros problemas, tais como a ausência de testes de usabilidade com utilizadores, ou estes acontecerem demasiado tarde para poderem ser feitas melhorias ao sistema.

## **Etapas 2: até 2 de dezembro**

Tem como objetivo planear o trabalho a realizar no próximo semestre nas disciplinas de PTI e PTR, tendo em conta aspetos relacionados com as pessoas, o produto, o processo, e o projeto. Assim, pretende-se:

- Identificação de **recursos** humanos, das ferramentas de desenvolvimento, e do *software* necessário ao funcionamento do sistema.
- **Estimação** do esforço disponível e do esforço necessário à realização do trabalho, usando dados históricos, decomposição, e um modelo empírico.
- Escolha do **modelo de processo** de desenvolvimento de *software*, com justificação da opção tomada.
- Organização da equipa, e atribuição de **papéis**, tais como gestor de projeto, consultor de dados, programador de *front end* e de *back end*, administrador da base de dados, arquiteto de sistema, especialista de redes e de segurança, técnico de manutenção, e responsável de testes com utilizadores.
- Descrição dos **work packages**, suas tarefas, recursos associados, e principais resultados (*milestones*), e respetiva calendarização num mapa de Gantt.
- Gestão de **risco**, incluindo a preparação de uma tabela de riscos, e um plano de mitigação, monitorização, e gestão dos principais riscos.
- Uma versão preliminar da **API REST** que o sistema deve suportar, sendo recomendado o uso da especificação OpenAPI e do editor Swagger.

De notar que as datas das etapas estão sujeitas à aprovação da coordenadora da LTI e podem vir a ser alteradas.

## **Entregas**

Em cada etapa, os elementos pedidos devem ser colocados num arquivo ZIP com nome PGP-2223-GXX-EY.ZIP, onde XX é o número do grupo (ex. 01) e Y é o número da etapa, o qual deve ser entregue através de atividades existentes na página de entrada da disciplina no Moodle.

## **Avaliações**

Os projetos são avaliados na presença de cada grupo, sendo atribuídas notas qualitativas. Tendo em conta as avaliações presenciais e os ficheiros entregues, é posteriormente atribuída uma nota quantitativa.

Bom trabalho no projeto!