



## Corporate Collaboration Relatório Beta

David Fidalgo  
[goncalo9marco@hotmail.com](mailto:goncalo9marco@hotmail.com)  
968 225 962  
45365

Pedro Santos  
[santos.pedroc@gmail.com](mailto:santos.pedroc@gmail.com)  
962 942 930  
45366

Valdemar Antunes  
[ValdemarPCA@hotmail.com](mailto:ValdemarPCA@hotmail.com)  
966 439 259  
44865

### Orientadores:

Paula Graça, ISEL, [paula.graca@isel.pt](mailto:paula.graca@isel.pt)  
Diogo Pacheco, Do iT Lean, [diogo.pacheco@doitlean.com](mailto:diogo.pacheco@doitlean.com)

Relatório beta realizado no âmbito de Projecto e Seminário,  
do curso de licenciatura em Engenharia Informática e de Computadores  
Semestre de Verão 2019/2020

15 de Junho de 2020

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>3</b>
1.1	Enquadramento . . . . .	3
1.2	Objetivos . . . . .	3
<b>2</b>	<b>Formulação do Problema</b>	<b>4</b>
2.1	Estado da Arte . . . . .	4
2.2	Análise de Requisitos . . . . .	4
2.2.1	Registo e Autenticação . . . . .	5
2.2.2	<i>Home Page — Dashboard</i> . . . . .	6
2.2.3	Registo das necessidades internas da empresa, gestão de candidaturas, notificações e integração com o google maps . . . . .	6
2.2.4	Divulgação e calendarização das necessidades . . . . .	8
2.3	Plataforma OutSystems . . . . .	9
<b>3</b>	<b>Descrição do problema</b>	<b>10</b>
3.1	Diagrama de blocos da solução . . . . .	10
3.2	Arquitetura . . . . .	11
3.2.1	Plataforma <i>OutSystems</i> . . . . .	11
3.2.2	Arquitetura 4 Layer Canvas . . . . .	12
3.3	Modelo de Dados . . . . .	13
3.4	Implementação . . . . .	14
3.4.1	Autenticação . . . . .	14
3.4.2	<i>Dashboard</i> . . . . .	14
3.4.3	Criação e edição de categorias . . . . .	15
3.4.4	Visualização de necessidades . . . . .	15
3.4.5	Visualização de necessidades num calendário . . . . .	16
3.4.6	Criação e edição de necessidades . . . . .	17
3.4.7	Notificações na plataforma . . . . .	19
<b>4</b>	<b>Conclusão</b>	<b>21</b>

# Listas de Figuras

2.1	<i>Use Case</i> — Ações gerais.	5
2.2	<i>Use Case</i> — <i>Login</i> .	5
2.3	<i>Use Case</i> — Necessidades.	6
2.4	<i>Use Case</i> — Categorias.	6
2.5	<i>Use Case</i> — Necessidade.	7
2.6	<i>Use case</i> — Calendário.	8
2.7	<i>Use case</i> — Announcements.	9
3.1	Diagrama de blocos da aplicação	10
3.2	Arquitetura da plataforma <i>OutSystems</i> [1]	11
3.3	<i>4 Layer Canvas</i>	12
3.4	Modelo de dados	13
3.5	Ecrã de autenticação	14
3.6	<i>Dashboard</i>	15
3.7	Ecrã das necessidades.	16
3.8	Ecrã do calendário.	16
3.9	Ecrã <i>Necessity Creation</i> — Criação parte 1	17
3.10	Ecrã <i>Necessity Creation</i> — Criação parte 2	17
3.11	Ecrã <i>Necessity Creation</i> — Aba de detalhe da necessidade.	18
3.12	Ecrã <i>Necessity Creation</i> — Aba de recursos parte 1.	18
3.13	Ecrã <i>Necessity Creation</i> — Aba de recursos parte 2.	19
3.14	Ecrã <i>Necessity Creation</i> — Aba dos participantes.	19
3.15	Ecrã <i>Notifications</i> onde cada utilizador pode consultar as suas notificações.	20
4.1	Timeline	21

# **Capítulo 1**

## **Introdução**

### **1.1 Enquadramento**

O mercado de hoje, cada vez mais tecnológico, exigente e desafiador, impõe um ritmo às empresas que, para além de gerirem os seus principais processos de negócio, estas têm também uma dinâmica significativa de atividades internas para as ajudar no seu crescimento e competitividade. Em particular, as empresas do setor das tecnologias de informação, mantêm atividades internas tais como participação em feiras de emprego, partilha de conhecimento através de apresentações informais, desenvolvimento de componentes de software, ofertas de formação, atividades lúdicas, entre muitas outras. Para isso, tem que existir uma coordenação de recursos que nem sempre é fácil, dada a sua alocação aos projetos em curso. Contudo, se existir um planeamento atempado gerido através de uma plataforma de colaboração, o processo pode ser agilizado, permitindo não só o registo e divulgação das necessidades internas, bem como a aceitação de candidaturas por parte dos colaboradores mais interessados na sua realização.

### **1.2 Objetivos**

A aplicação proposta visa a implementação de uma plataforma colaborativa para agilizar a resposta a necessidades internas das empresas, nomeadamente a organização de eventos, partilha de conhecimento, ofertas formativas, entre outras. Os objetivos que a plataforma visa atingir consistem na divulgação de comunicados para a comunidade da empresa; no contexto de um evento, incluem a partilha de localização, gestão de recursos, registo de presenças, registo de candidaturas e gestão das mesmas; a calendarização de eventos e, por último, um sistema de notificações.

## Capítulo 2

# Formulação do Problema

O desenvolvimento da plataforma *Corporate Collaboration* pretende atingir certos requisitos, de forma a poder enquadrá-la no mercado atual.

### 2.1 Estado da Arte

As plataformas de colaboração existentes no mercado atual tais como a *Microsoft SharePoint* ou a *BaseCamp* apresentam um elevado nível de complexidade, incluindo ferramentas como calendários partilhados, partilha de ficheiros, mensagens instantâneas, armazenamento na *cloud*, video-conferência, entre outros. A aplicação desenvolvida neste âmbito não procura igualar todas as funcionalidades das plataformas já existentes, mas sim, agilizar a dinâmica de atividades internas que ocorrem diariamente nas empresas. Visa, essencialmente, facilitar a procura de pessoas para cumprir certas necessidades que existam no âmbito das atividades internas da empresa, necessidades estas que permitem definir uma tarefa que tem de ser realizada ou um evento que será organizado no contexto da dinâmica interna da empresa. Estas necessidades são auxiliadas por recursos que permitem gerir candidaturas e participantes das mesmas, assim como apresentar informações sobre alojamento, refeições, transporte e localização, se for caso disso.

### 2.2 Análise de Requisitos

As funcionalidades principais da plataforma *Corporate Collaboration* estão representadas na figura 2.1

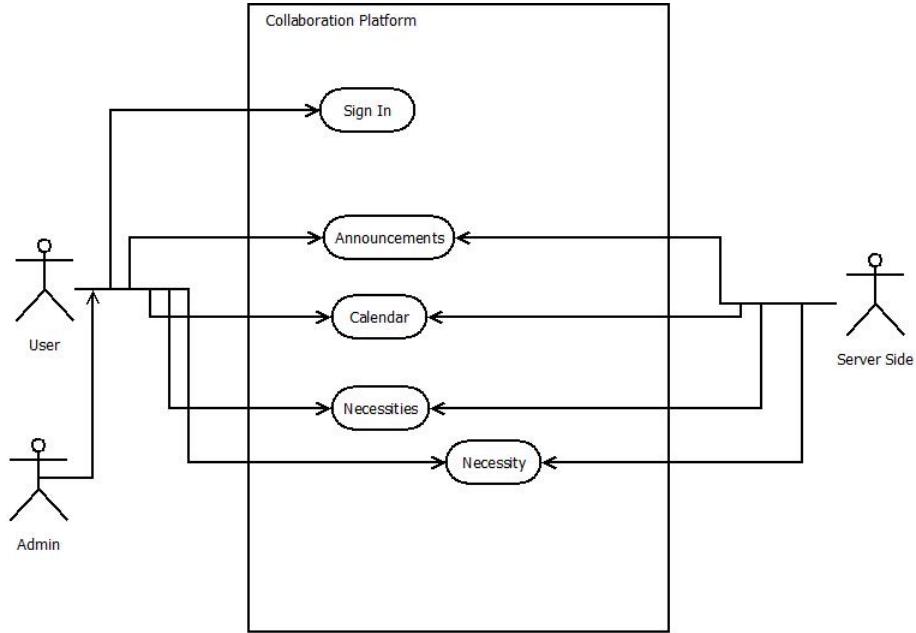


Figura 2.1: *Use Case* — Ações gerais.

### 2.2.1 Registo e Autenticação

Para desenvolver a funcionalidade de registo e autenticação na plataforma de colaboração é necessário existir uma tabela na base de dados que guarde as credenciais e informação básica de cada utilizador, nomeadamente o *email*, *password*, *username*, primeiro e último nome. Esta tabela tanto serve para efeitos de registo de um utilizador, como para verificação posterior da sua autenticação. Os utilizadores não necessitam de fazer registo na aplicação, visto que nesta plataforma é suposto utilizarem as mesmas credenciais atribuídas pela empresa, para e-mail e autenticação nas outras aplicações internas.

Relativamente ao ecrã de *Login* ilustrado na figura 2.2, este suporta:

- Duas caixas de texto onde o utilizador introduz o seu e-mail e password.
- Um botão identificado como *Login* que desencadeia o processo de verificação das credenciais introduzidas.

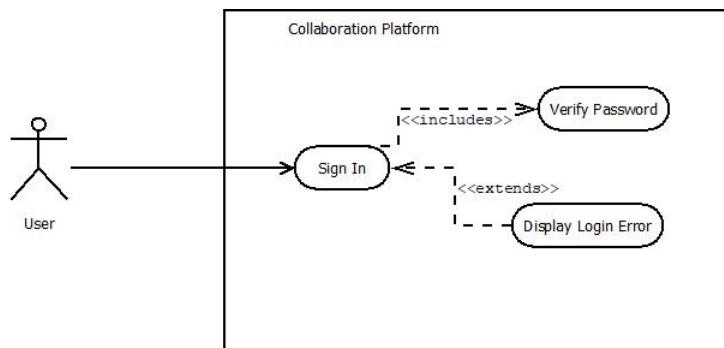


Figura 2.2: *Use Case* — *Login*.

O utilizador com permissões de administrador tem acesso ao *back-office* onde pode criar ou remover categorias.

## 2.2.2 Home Page — Dashboard

Após o *login* na aplicação o utilizador será redirecionado para um novo ecrã que irá conter um *dashboard* com o intuito de organizar e apresentar a informação de uma forma apelativa. As necessidades a que o utilizador se candidatou serão apresentadas neste *dashboard*. O mesmo irá conter também uma secção com estatística relativa ao número de necessidades que existem em cada categoria. A seleção de uma das categorias na estatística redirecionará o utilizador para o ecrã das necessidades, apresentando a lista das mesmas com o filtro correspondente à categoria selecionada.

## 2.2.3 Registo das necessidades internas da empresa, gestão de candidaturas, notificações e integração com o google maps

A funcionalidade de registo de necessidades internas da empresa estão organizadas em categorias de modo a facilitar a navegação do utilizador pelas mesmas.

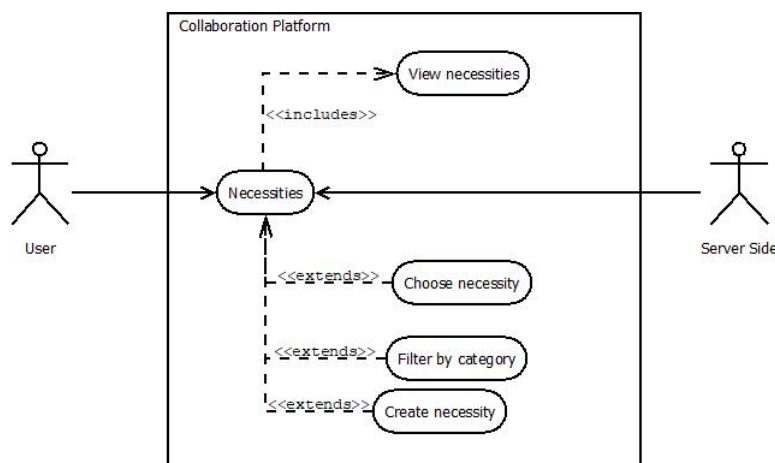


Figura 2.3: Use Case — Necessidades.

Para isso, é fundamental que exista um ecrã que apresente todas as necessidades da empresa. Um utilizador, caso queira registar uma necessidade, irá escolher a categoria que melhor se adequa à mesma. Caso não exista, este, se tiver permissões de admin, poderá criar uma categoria nova no *back-office* da aplicação. Neste ecrã será possível filtrar as necessidades pela sua categoria e pelo seu grau de urgência.

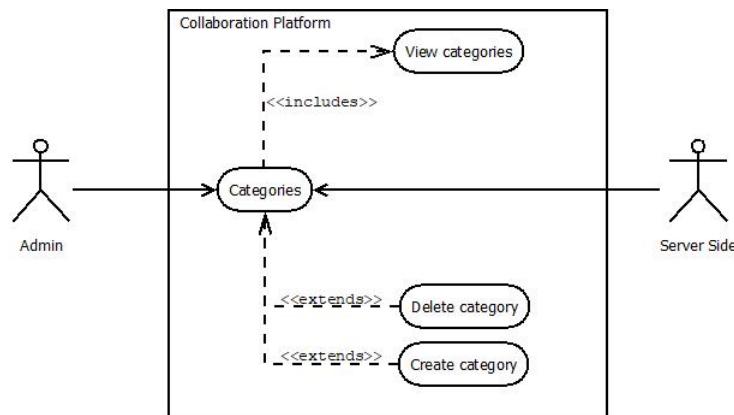


Figura 2.4: Use Case — Categorias.

Posto isto, definimos um conjunto de filtros que podem ser conjugados de modo a permitir um melhor agrupamento e organização das necessidades. O primeiro consiste na filtragem por categorias, como por exemplo *Brown Bags*, *Qualification Offers*, *Software Components Development* ou *Planning of Events*. Foi ainda definido um segundo filtro que poderá tomar os valores *High*, *Medium* ou *Low*, correspondentes ao grau de prioridade com que as necessidades foram criadas. A seleção dos filtros irá levar a uma atualização da lista para conter apenas necessidades que se enquadrem nessa mesma seleção. Este ecrã apresenta ainda um botão que servirá para criar uma nova necessidade, criação esta acessível a todos os utilizadores autenticados, que decorrerá num novo ecrã e que terá como opção (obrigatória) de criação os filtros a qual associar a nova necessidade. Em cada necessidade existe a possibilidade de associar recursos com informações sobre alojamento, refeições, localização do evento ou transporte para o local onde a mesma se irá realizar. Todos os recursos, com exceção do recurso transporte, têm associado um mapa do *Google Maps* onde os utilizadores podem fornecer informações sobre localização. Ao clicar numa necessidade, será apresentado um novo ecrã com os detalhes da mesma, com os recursos associados e também a possibilidade do utilizador se candidatar como colaborador ou participante da necessidade, dependendo da fase em que a mesma se encontra. O autor da necessidade irá receber a notificação de que existe um novo candidato, dando-lhe a opção de apesar carregar na notificação ser redirecionado para o ecrã de detalhe da necessidade onde pode observar os candidatos. Ao criar uma necessidade o autor tem a possibilidade de escolher se todas as candidaturas à necessidade em causa são aceites automaticamente ou se o próprio escolhe os candidatos com base na descrição dada pelos mesmos. O candidato irá receber uma notificação nos casos em que a necessidade for fechada/cancelada, e quando a sua candidatura for aceite/recusada. Terá também a possibilidade de ver quem já se candidatou à necessidade criada pelo próprio no ecrã de detalhe na parte dos participantes, estando candidatos a colaboradores separados de candidatos a participantes. O ecrã que permite mostrar as notificações associadas a cada utilizador é acedido ao carregar no ícone em forma de sino presente na barra da aplicação. Este ícone tem um pequeno contador associado que indica o número de notificações pendentes que cada utilizador tem e, quando pressionado, é apresentado o ecrã das notificações onde as mesmas são apresentadas na forma de uma lista.

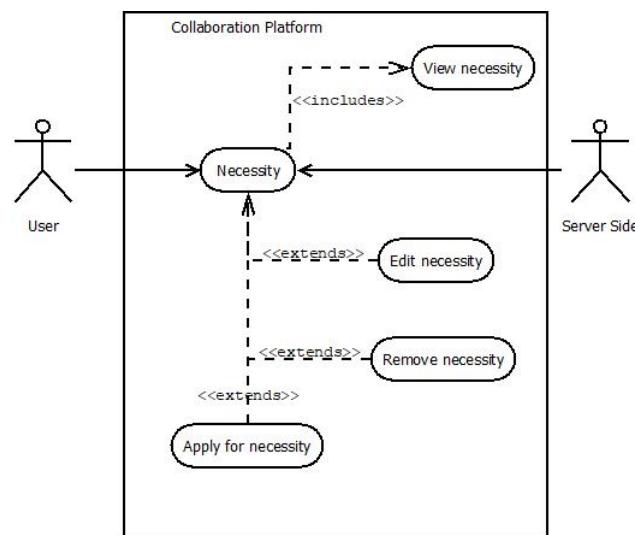


Figura 2.5: Use Case — Necessidade.

Exemplificando, para organizar eventos ou feiras de emprego, o utilizador seleciona a categoria *Planning of events*.

Se por outro lado, tivermos o objetivo de realizar apresentações informais de partilha de conhecimento o utilizador seleciona a categoria *Brown Bags*. Todos os utilizadores devidamente autenticados podem candidatar-se para assistir a uma dada apresentação ou para a orientar. No contexto da necessidade é definida uma linha temporal que consiste numa primeira fase de candidaturas para escolha do orientador (ou orientadores) que irão conduzir a apresentação, seguida de uma segunda fase de candidaturas para utilizadores que queiram assistir à apresentação. O autor de uma apresentação, no ecrã de detalhe da mesma, terá acesso a quem se candidatou para a orientar, podendo escolher um ou mais orientadores com base na descrição apresentada pelos mesmos. Numa segunda fase após a escolha do(s) mesmo(s), o autor pode ver os utilizadores que se candidataram à necessidade no seu ecrã de detalhe e todos os utilizadores autenticados podem ver quem foi aceite/vai ao evento.

#### 2.2.4 Divulgação e calendarização das necessidades

Com o objetivo de divulgar e calendarizar as necessidades internas da empresa, a barra de navegação da aplicação terá um botão que, quando carregado, redireciona o utilizador para um ecrã que apresenta um calendário com o qual ele poderá interagir. Neste calendário são apresentados todos os eventos organizados, na sua respetiva data, e existe a possibilidade de filtrar os eventos em que o mesmo participará. Após a seleção de um dia no calendário, são apresentados os eventos, dando a possibilidade ao utilizador de ver os detalhes individuais após carregar num deles, num novo ecrã.

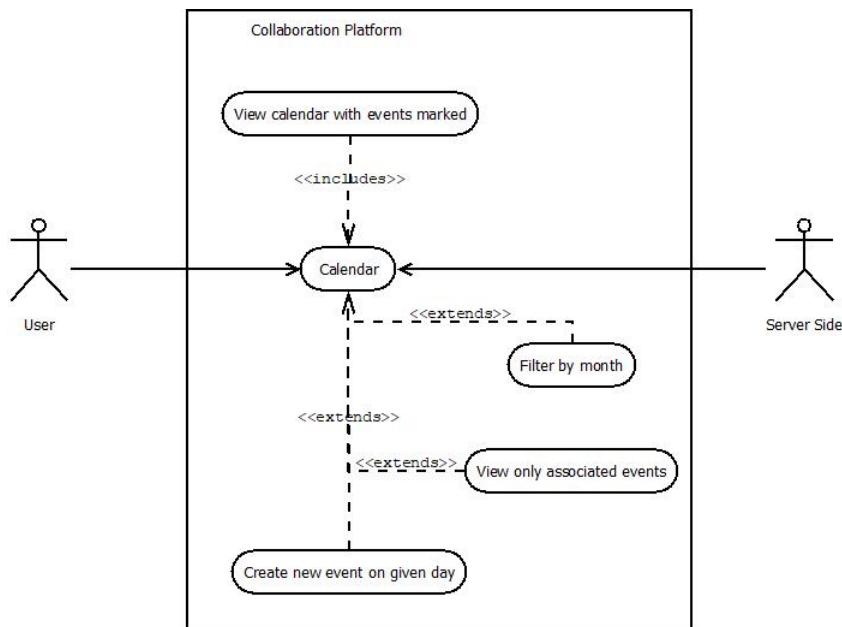


Figura 2.6: *Use case — Calendário.*

Para ser possível divulgar as necessidades da empresa de forma uniforme por todos os seus colaboradores, a barra de navegação da aplicação apresentará um botão denominado *Announcements* que, quando carregado, abre um ecrã (figura 2.7) que contém os comunicados sobre a forma de uma lista. Sempre que for emitido um novo comunicado todos os utilizadores recebem uma notificação com o título do mesmo e, ao carregar nessa notificação, são redirecionados para os detalhes do comunicado. Estes comunicados foram criados por um utilizador com permissões de administrador, sendo visíveis por todos os que estejam autenticados.

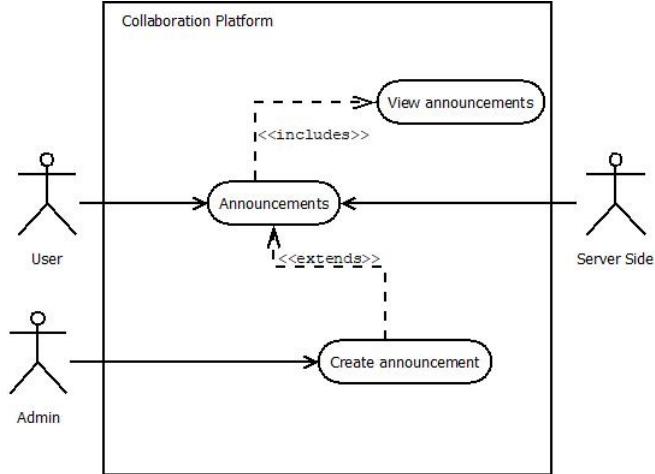


Figura 2.7: *Use case — Announcements.*

## 2.3 Plataforma OutSystems

A escolha da plataforma *OutSystems* [1], para a implementação deste projeto baseia-se essencialmente no facto da mesma ser *low-code*, apresentando vários benefícios essenciais para um projeto de curto prazo como o apresentado. A rapidez com que é possível produzir peças de *software*, as funcionalidades *user-friendly* como *drag and drop* e interfaces de utilizador pré-festas e a possibilidade de produzir aplicações automaticamente através de um simples clique, são elementos essenciais que levaram à escolha desta plataforma. Cada *endpoint* da aplicação está associado a um ecrã com o qual o utilizador interage, havendo uma naveabilidade entre ecrãs que permitindo, assim, que o utilizador visite os diferentes *endpoints* da aplicação. A criação e utilização de *blocks* é uma das funcionalidades altamente vantajosas que a plataforma *OutSystems* [1] apresenta, visto que permite a reutilização deste componente ao longo da *User Interface — UI*. A construção da aplicação torna-se bastante rentável, em termos de tempo e facilidade do desenvolvimento, através da conjugação de conceitos como *Widgets*, *Screens* e *Blocks* no domínio da *UI* e de conceitos como *Server Actions* (ações executadas do lado do servidor), *Client Actions* (ações executadas do lado do cliente) e *Aggregates* (ação com o propósito de aceder à base de dados para retornar informações persistentes na mesma). Também é disponibilizado pela *OutSystems* [1], um *eSpace* por *module* que contém elementos essenciais ao desenvolvimento, como os enunciados anteriormente. A base de dados está alojada numa *cloud*, onde são guardadas todas as entidades dos vários módulos.

# Capítulo 3

## Descrição do problema

O desenvolvimento de uma plataforma colaborativa apresenta diversos desafios ao nível da sua arquitetura, do modelo de dados a construir e das decisões de implementação ao longo do seu desenvolvimento. Neste capítulo irá ser abordado as decisões tomadas no desenvolvimento do projeto, tendo em conta a plataforma *OutSystems* [1]. Irá ser feita uma descrição mais detalhada da solução adotada, acompanhada por um diagrama de blocos, que permite esquematizar a mesma.

### 3.1 Diagrama de blocos da solução

A figura 3.1 apresenta o diagrama de blocos da plataforma *Corporate Collaboration*.

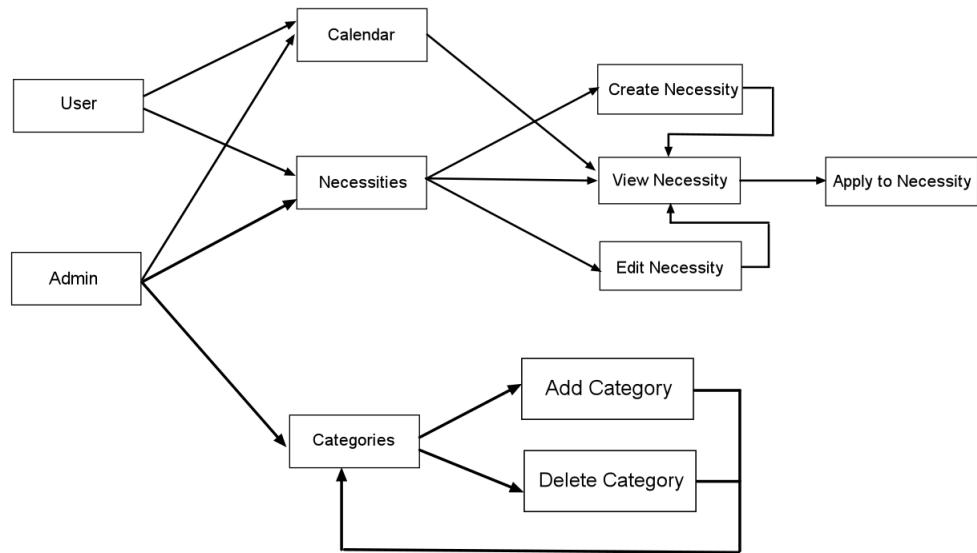


Figura 3.1: Diagrama de blocos da aplicação

## 3.2 Arquitetura

### 3.2.1 Plataforma *OutSystems*

A arquitetura desta plataforma pode ser observada na figura 3.2. O principal componente da plataforma é o Platform Server que permite que as aplicações desenvolvidas sejam geradas, optimizadas, compiladas e publicadas.

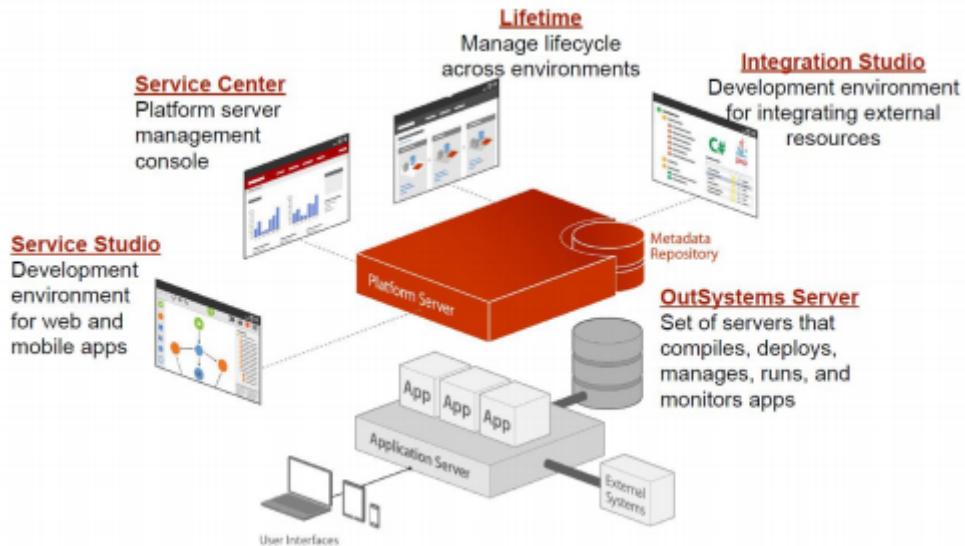


Figura 3.2: Arquitetura da plataforma *OutSystems* [1]

Este componente usa os seguintes serviços:

1. *Code Generator* — Usa a aplicação modelada no *Service Studio* e gera o código necessário usando tecnologias standard (como *.NET*, *SQL Server*, *HTML*, etc.) para a criação de uma aplicação optimizada e segura.
2. *Deployment Services* — Publica o código que foi previamente gerado no servidor, assegurando que a aplicação é instalada consistentemente em cada front-end da infraestrutura.
3. *Application Services* — Gere as aplicações durante o runtime, através da execução de *batches* agendados e serviços de *logging* assíncronos que permitem que sejam armazenados eventos como erros, inspeções e métricas de desempenho.

### 3.2.2 Arquitetura 4 Layer Canvas

Para desenharmos a arquitetura da nossa solução, seguimos a metodologia da plataforma *OutSystems* [1], a *4 Layer Canvas* apresentado na figura 3.3.

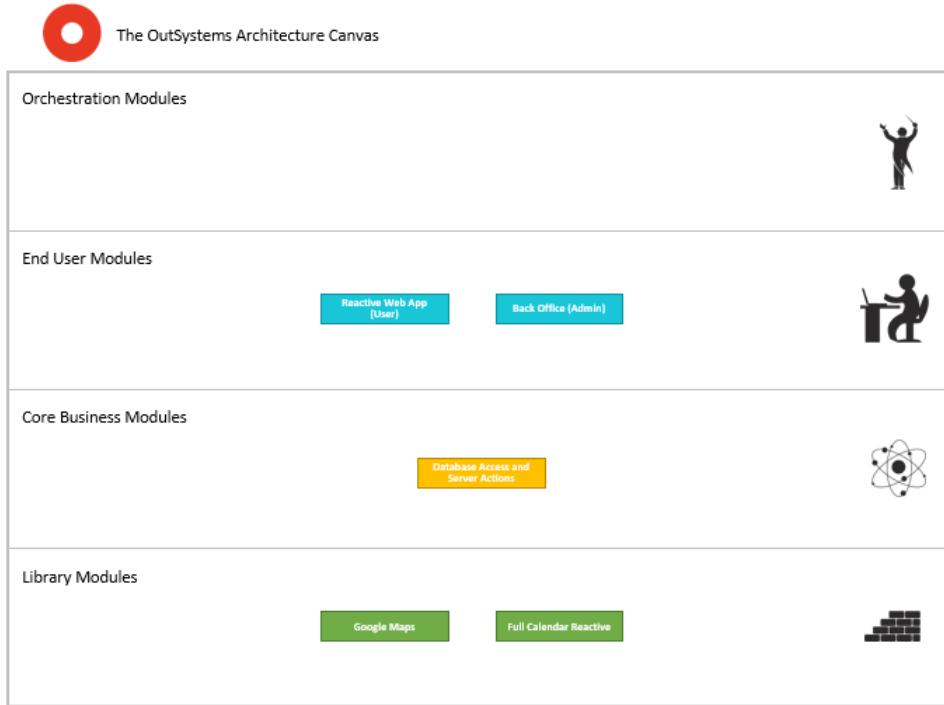


Figura 3.3: 4 Layer Canvas

Esta metodologia propõe que se estruture as várias funcionalidades da aplicação por quatro camadas, sendo estas, começando por baixo:

- *Library Layer* — Aqui devem constar os módulos que são transversais ao domínio do problema, tais como: temas, bibliotecas, etc.
- *Core Layer* — Módulos referentes à lógica de negócio, modelo de dados e *server actions*.
- *End User Layer* — Nesta camada é tratada toda a parte de interface e experiência do utilizador, fazendo uso das camadas anteriores.
- *Orchestration Layer* — Camada que coordena a comunicação entre várias aplicações.

É importante verificar que, apesar da metodologia apresentar quatro camadas, a nossa arquitetura apenas faz uso das primeiras três devido ao nosso projeto consistir em apenas uma aplicação reactive, e não havendo necessidade de coordenar interações com outras aplicações na camada de orquestração. Posto isto, a nossa aplicação assenta sobre cinco módulos, representados pela figura 3.3.

Começando pela *Library Layer* verificamos que são utilizados os módulos relativos à integração da aplicação com o *Google Maps* e com o *Full Calendar Reactive*. De seguida temos a *Core Layer*, onde definimos as entidades de domínio e suas operações. Por fim a *End User Layer* onde são definidos os ecrãs e a lógica de cliente.

### 3.3 Modelo de Dados

O conceito predominante no modelo de dados, (figura 3.4), é o de necessidade, representado pela tabela *Necessity*.

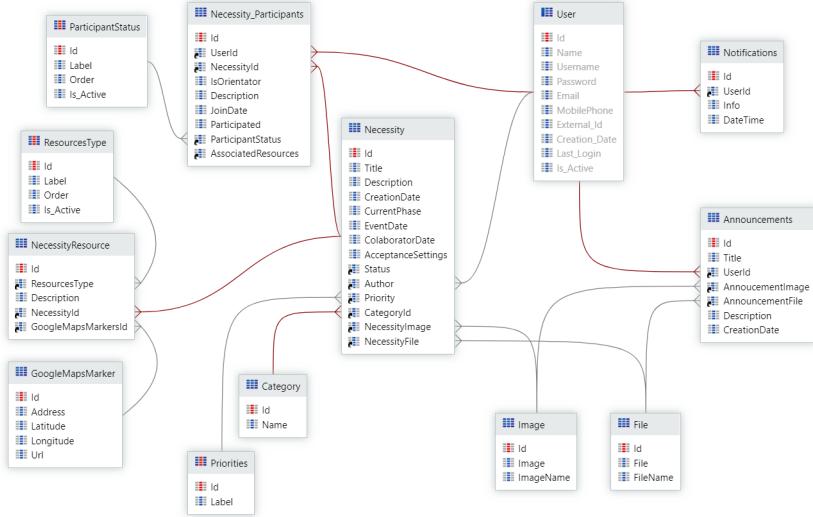


Figura 3.4: Modelo de dados

Uma necessidade é caracterizada por diversos elementos dos quais destacamos o *Status* que referencia uma tabela estática (*NecessityStatus*) que contém os valores possíveis do estado da necessidade, podendo ter os valores Closed; Active ou Archived; A *CurrentPhase* que indica a fase de candidaturas atual, podendo ser para orientadores ou participantes, a *ColaboratorDate* que representa a data limite das candidaturas à posição de orientador e ainda *AcceptanceSettings* que indica se para a necessidade em causa são aceites todos os participantes ou se existe uma filtragem por parte do autor. As tabelas *NecessityImage* e *NecessityFile* existem para o propósito de guardar ficheiros, aliviando a quantidade de dados guardada em cada tuplo *Necessity*, seguindo também as boas práticas da plataforma *OutSystems* [1]. A entidade estática *Priorities* contém os valores possíveis das prioridades que cada necessidade pode ter. Associado ao conceito de necessidade está o de recurso, que tem como objetivo acrescentar informação à necessidade de uma forma flexível. Para suportar este conceito foi definida a entidade estática *ResourcesType* cujos valores representam os tipos de recursos que um utilizador pode, facultativamente, acrescentar à necessidade, consistindo em acomodação, restaurante e menus de refeições, localização do evento ou transporte. A representação de um recurso no modelo de dados traduz-se na entidade *NecessityResource* que guarda informações de cada recurso como o seu tipo (referência à tabela *ResourcesType*), a sua descrição, o id da necessidade a que o mesmo pertence e, como todos os recursos à exceção do relativo ao transporte têm um mapa do *Google Maps* associado, o último atributo da tabela *NecessityResource* é um id do marcador do mapa associado ao recurso. Com o objetivo de guardar informação sobre os participantes de uma necessidade, é definida a entidade *Necessity\_Participants* que contempla o identificador da mesma, o identificador do utilizador, a descrição associada à candidatura, se esta candidatura é para posição de orientador ou de participante, a data da candidatura, se o utilizador participou na necessidade, qual o estado da candidatura e os recursos associados a este participante. Um utilizador pode ser um participante, um autor ou um orientador da necessidade. Os comunicados feitos na plataforma são suportados pela entidade *Announcements*. As notificações apresentadas na plataforma são suportadas pela entidade *Notifications* em que cada notificação está associada a um utilizador através do seu

*id*, o seu conteúdo é guardado no atributo *Info* e a hora da emissão da notificação no atributo *DateTime*. Um utilizador pode ainda ter o cargo de administrador, cujos privilégios incluem editar as categorias representadas pela entidade *Category*.

## 3.4 Implementação

### 3.4.1 Autenticação

Para realizar a autenticação de um utilizador, este introduz as suas credenciais nos respetivos *input fields* e clica no botão *Login*. Se as credenciais introduzidas não corresponderem às de nenhum utilizador na base de dados, será apresentada uma mensagem de erro. Um utilizador só terá acesso a outros ecrãs se tiver autenticado, caso contrário será redirecionado para este ecrã, figura 3.5.

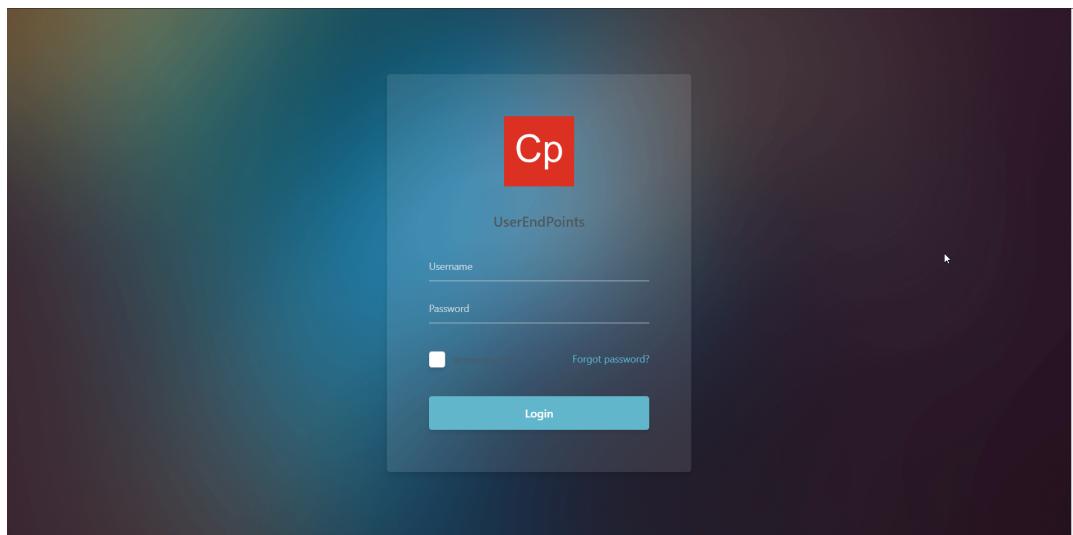


Figura 3.5: Ecrã de autenticação

### 3.4.2 Dashboard

A um utilizador autenticado é apresentado o ecrã da *dashboard* figura 3.6, quando inicia a aplicação. O mesmo apresenta quatro *widgets* que apresentam informação sobre o número de utilizadores que estão *logged in* e o número de necessidades abertas com as suas respetivas prioridades (*high*, *medium*, *low*). Este ecrã apresenta também um gráfico com informação sobre as categorias existentes e a distribuição do número de necessidades criadas em cada uma das mesmas. A seleção de uma categoria do gráfico promove a navegação para o ecrã das necessidades, apresentando a lista das mesmas que pertencem à categoria selecionada. Para cada uma das prioridades que as necessidades podem ter, é apresentado um *ranking* dos utilizadores que mais criaram necessidades, com a respetiva prioridade.

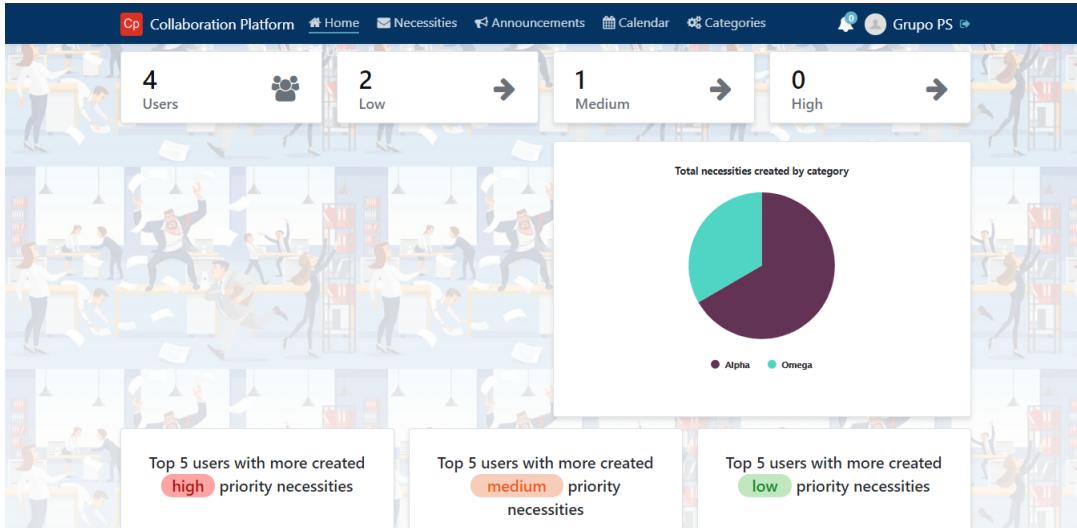


Figura 3.6: *Dashboard*

### 3.4.3 Criação e edição de categorias

Um utilizador com permissões de *admin* tem a possibilidade de aceder ao ecrã *Categories* e de criar ou eliminar categorias. É apresentado um *widget* tabela com todas as categorias existentes e um botão delete que permite eliminar as selecionadas. O processo de criação de uma nova categoria é feito carregando no botão *create* e inserindo o nome da nova categoria no *popup* que aparece no ecrã, seguido do botão *save*.

### 3.4.4 Visualização de necessidades

O utilizador ao carregar no botão *Necessities* presente na barra da aplicação, será redirecionado para o ecrã responsável pela apresentação das necessidades demonstrado na figura 3.7. O intuito do mesmo consiste na apresentação das necessidades criadas pela comunidade empresarial, sendo possível aplicar filtros e/ou pesquisar pelo título de uma necessidade de modo a que sejam apresentadas apenas as necessidades alvo. As necessidades são apresentadas num *widget* tabela, cujas colunas apresentam informação relevante como título, categoria, prioridade, estado, data de criação e data em que a necessidade irá decorrer. Sempre que o utilizador pretender criar uma nova necessidade, apenas tem que carregar no botão *Create Necessity* e será redirecionado para um novo ecrã, onde poderá completar a criação. A seleção do título de uma necessidade promove a navegação para o ecrã dos detalhes da mesma. A barra de pesquisa permite que o utilizador procure uma necessidade pelo seu título. Os filtros aplicáveis são apresentados em dois *dropdowns*, um para permitir a escolha de qual a prioridade e outro para seleção da(s) categoria(s). Sempre que exista uma mudança na seleção de algum dos *widgets* de filtragem de necessidades ou uma introdução de texto na barra de pesquisa, a tabela é atualizada para apresentar apenas aquelas que verifiquem as características alvo. As informações de cada uma das necessidades, das categorias existentes e das prioridades são obtidas comunicando com o servidor através de *Aggregates*. Os *Aggregates* são *queries* optimizadas à base de dados de modo a retornar informação de forma eficiente e apresentá-la nos *widgets* do ecrã.

Figura 3.7: Ecrã das necessidades.

### 3.4.5 Visualização de necessidades num calendário

O utilizador ao carregar no botão *Calendar* presente na barra da aplicação, será redirecionado para o ecrã, apresentado na figura 3.8 responsável por mostrar, num calendário, todas as necessidades existentes organizadas por datas. O objetivo deste ecrã é apresentar de uma forma mais organizada e estruturada todas as necessidades criadas pela empresa, sendo possível filtrá-las. Para preencher o calendário, para cada necessidade, é criado um objeto *Event* com a sua informação. Cada evento estará representado no calendário com uma cor diferente de forma a distinguir entre necessidades criadas pelo utilizador, necessidades associadas, nomeadamente uma inscrição, e necessidades criadas por outros utilizadores. Se o utilizador carregar num evento, este é redirecionado para o seu ecrã de detalhe. É possível filtrar eventos pela sua categoria, através de um popover menu que contém todas as categorias existentes. Caso seja selecionada uma categoria, são removidos todos os eventos presentes no calendário, e de seguida, renderizado com os novos eventos filtrados.

Figura 3.8: Ecrã do calendário.

### 3.4.6 Criação e edição de necessidades

O ecrã *NecessityCreation* (figura 3.9 e 3.10) é utilizado para criar, editar, remover ou ver os detalhes de uma necessidade. No momento da criação de uma necessidade os *widgets* presentes neste ecrã permitem que o utilizador indique o título, a descrição, a categoria a qual associar a necessidade, a sua prioridade, se existe filtragem de participantes ou todas as candidaturas são aceites, uma imagem e um ficheiro para serem associados à necessidade, a data em que a mesma irá decorrer, se é necessário colaborador(es) e a data limite para a(s) sua(s) candidatura(s).

The screenshot shows the 'Necessity' creation form. At the top, there's a header with the platform logo and navigation links: Home, Necessities, Announcements, Calendar, Categories, and a user icon for 'Grupo PS'. The main form area has sections for 'Title \*', 'Description \*', 'Category \*' (with a dropdown menu), 'Priority' (with three options: Low, Medium, High), and 'How do you want to filter applicants to this necessity?'. Below these are two radio button options: 'Accept all applicants' and 'Filter each appliance'. To the right of the form is a large, semi-transparent background image showing people working in an office environment.

Figura 3.9: Ecrã *Necessity Creation* — Criação parte 1

This screenshot shows the continuation of the 'Necessity' creation form. It includes two date pickers: 'Event Date \*' and 'End of Collaborator Phase'. Both pickers show a calendar for June 2020 with specific dates highlighted (e.g., 15th). There is a checked checkbox labeled 'Need Colaborator?'. To the right is a placeholder for an image with the text 'Choose an image...'. At the bottom right of the form are 'Cancel' and 'Save' buttons.

Figura 3.10: Ecrã *Necessity Creation* — Criação parte 2

Ao pressionar o botão presente no final do ecrã com a legenda *Save*, tanto num cenário de edição como de criação, são guardadas as informações presentes nos *widgets* e é desencadeada uma *client action* que posteriormente cria uma ligação ao servidor através da chamada a uma *server action* para modificar a base de dados com uma nova necessidade ou alterando uma necessidade pré-existente.

Se o utilizador desejar editar ou remover uma necessidade, só o poderá fazer se for o autor desta e, no caso de a editar, o ecrã (figura 3.11) apresentará os campos já preenchidos com os dados atuais da necessidade e a possibilidade de os alterar livremente.

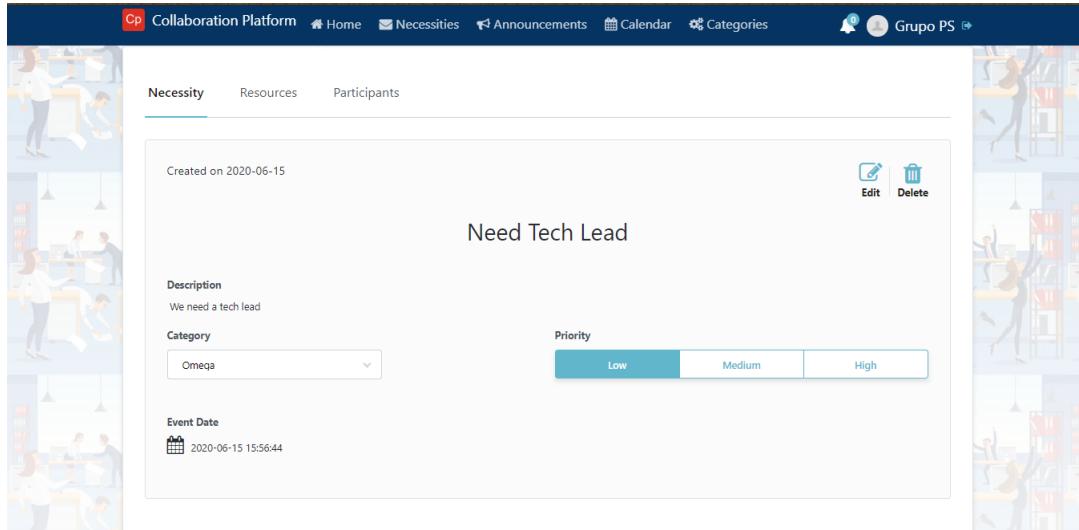


Figura 3.11: Ecrã *Necessity Creation* — Aba de detalhe da necessidade.

É também neste ecrã (figura 3.12) que o utilizador poderá consultar os recursos associados através da navegação até à tab *Resources* e se for o autor da necessidade pode adicionar novos recursos. Todos os recursos com exceção do recurso transporte têm um *block* que contém um mapa do *Google Maps* através do qual o utilizador pode colocar um marker para dar a conhecer a localização associada ao recurso. A integração do *Google Maps* (figura 3.13) na plataforma é realizada através da comunicação com a API do *Google Maps* [2]. Para criar um novo mapa é necessário criar um objeto do tipo *google.maps.Map* que recebe na sua construção o elemento *html* onde o mapa será inserido. Toda a dinâmica de interação com o mapa é gerida pela API através de eventos que são desencadeados com o toque no mesmo.

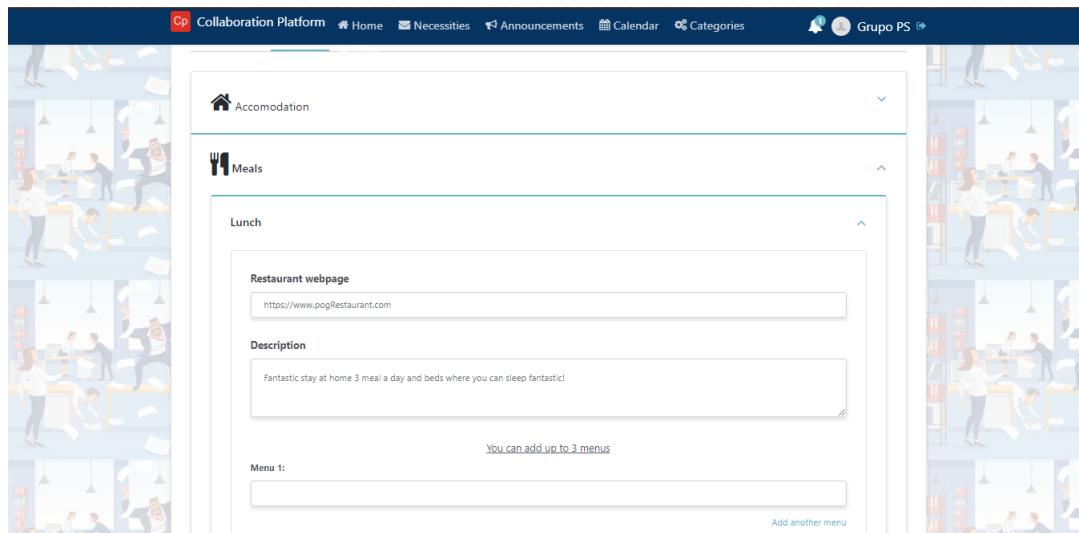


Figura 3.12: Ecrã *Necessity Creation* — Aba de recursos parte 1.

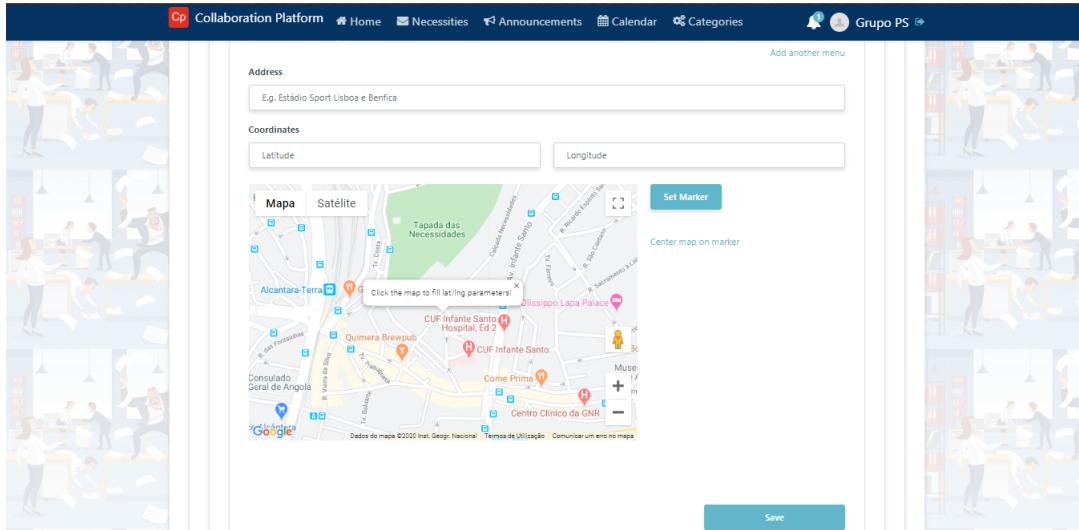


Figura 3.13: Ecrã *Necessity Creation* — Aba de recursos parte 2.

Na última tab do *widget Accordion* presente no ecrã *NecessityCreation* (figura 3.14) encontra-se a secção relacionada com os participantes à necessidade. Nesta secção um utilizador poderá candidatar-se à posição de colaborador ou participante, de acordo com a fase em que a necessidade se encontra, carregando no *widget link* com o texto *Apply as a Collaborator/Participant* e deixando uma descrição se a filtragem dos candidatos estiver activa para aquela necessidade. Um utilizador, ao criar uma necessidade, tem a opção de escolher se todas as candidaturas são aceites ou se o próprio irá proceder à sua filtragem selecionando um dos *widgets radio button*.

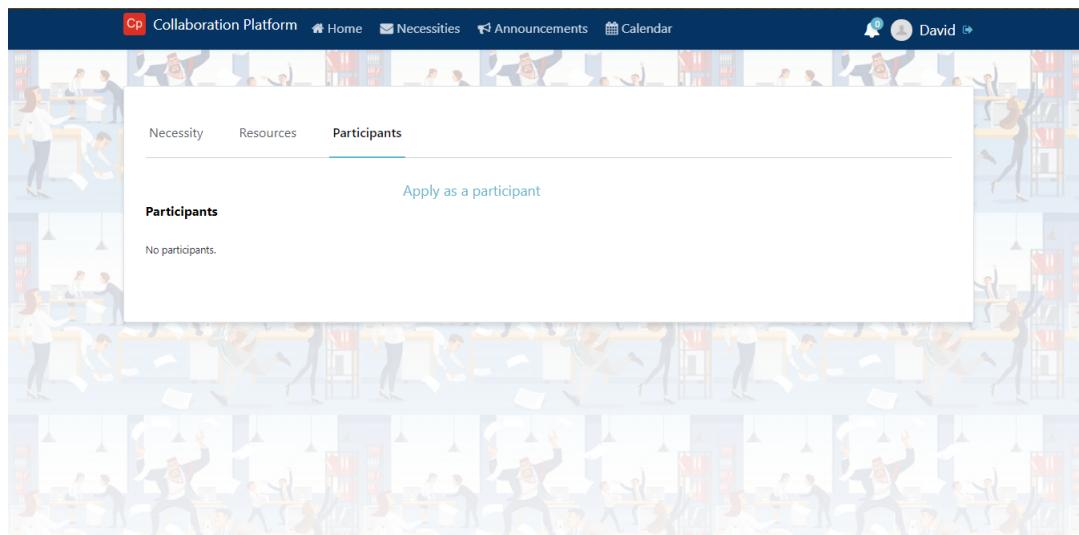


Figura 3.14: Ecrã *Necessity Creation* — Aba dos participantes.

### 3.4.7 Notificações na plataforma

A plataforma tem um sistema de notificações que suporta o seu envio para os utilizadores, num ecrã (figura 3.15) onde é possível visualizá-las em detalhe. Os cenários possíveis de envio de notificações consistem em: sempre que existir uma nova candidatura a uma dada necessidade o autor da mesma é notificado; quando um utilizador vê a sua candidatura aceite

ou não; e, por último, quando uma necessidade é eliminada, os participantes da mesma são notificados. A cada notificação está associado o *id* do utilizador, o que permite que quando o mesmo navegue até ao ecrã das notificações, lhe sejam apresentadas todas as suas notificações que recebeu, provenientes do registo na base de dados.

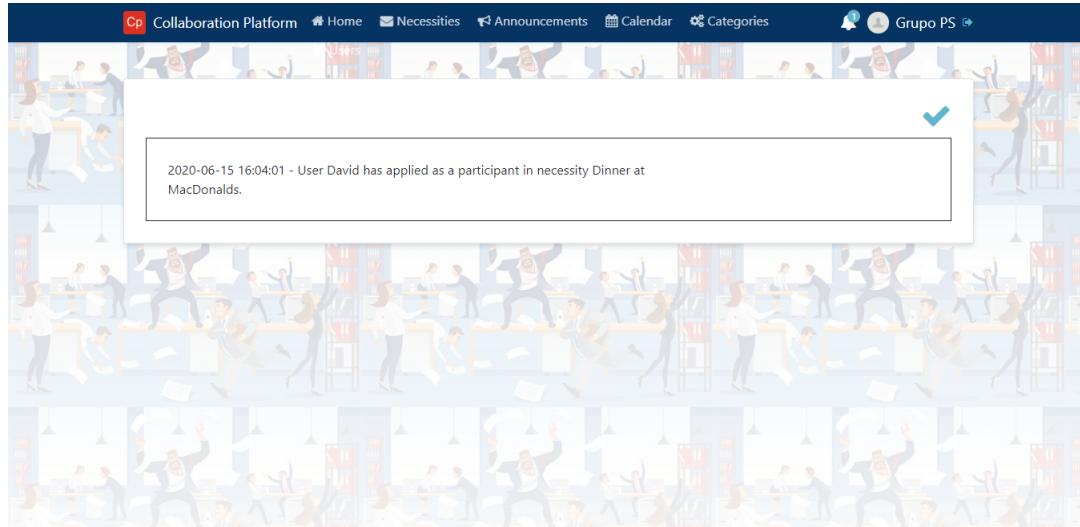


Figura 3.15: Ecrã *Notifications* onde cada utilizador pode consultar as suas notificações.

# Capítulo 4

## Conclusão

De uma forma geral, o progresso do projeto está de acordo com o planeamento apresentado anteriormente, na proposta de projeto, e novamente, na figura 4.1.

Inicialmente, devido à pesquisa e aprendizagem de algumas tecnologias novas, o desenvolvimento ocorreu de forma mais lenta. No entanto, após ultrapassado esse obstáculo, o ritmo de trabalho aumentou de forma considerável e consequentemente os resultados surgiram também mais rapidamente.

De momento, a aplicação já possui um modelo de dados estruturado e algumas funcionalidades, nomeadamente a possibilidade de criação, inscrição e visualização de uma necessidade, e um *back office* que permite adicionar e/ou remover categorias possíveis de uma necessidade. Para além disto, foi desenvolvida uma outra funcionalidade que estava no planeamento ser realizada no próximo sprint, nomeadamente a visualização das necessidades dispostas num calendário.

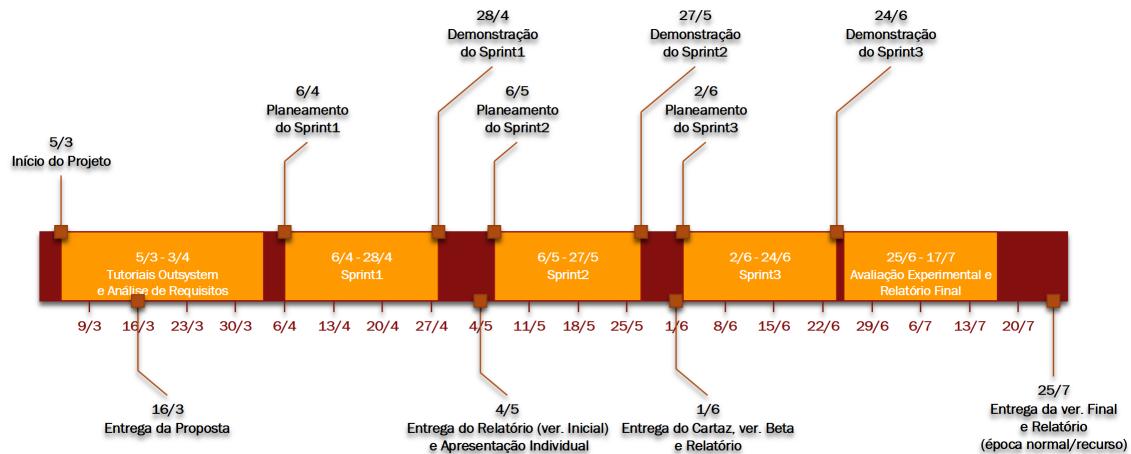


Figura 4.1: Timeline

# Bibliografia

- [1] *OutSystems*
- [2] Documentação da *API Google Maps*
- [3] Documentação da *API Places*
- [4] Documentação do componente da *Forge FullCalendar Reactive*