



Corporate Collaboration Relatório Final

45365 David Fidalgo

45366 Pedro Santos

44865 Valdemar Antunes

Orientadores:

Paula Graça, ISEL, paula.graca@isel.pt

Diogo Pacheco, Do iT Lean, diogo.pacheco@doitlean.com

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

24 de Julho de 2020

Conteúdo

1	Introdução	4
1.1	Enquadramento	4
1.2	Objetivos	4
1.3	Abstract	5
2	Formulação do Problema	6
2.1	Estado da Arte	6
2.2	Análise de Requisitos	7
2.2.1	Módulos constituintes da <i>Collaboration Platform</i> e <i>roles</i> associados . .	8
2.2.2	Registo e Autenticação	9
2.2.3	Manutenção da aplicação e <i>Back-Office</i>	10
2.2.4	<i>Home Page — Dashboard</i>	11
2.2.5	Registo das necessidades internas da empresa, gestão de candidaturas, associação de recursos, notificações e integração com o google maps . .	12
2.2.6	Divulgação e calendarização das necessidades	15
2.2.7	Divulgação de comunicados	16
2.3	Plataforma OutSystems	17
3	Solução proposta	18
3.1	Diagrama de blocos da solução	18
3.2	Arquitetura	20
3.2.1	Plataforma <i>OutSystems</i>	20
3.2.2	Arquitetura 4 Layer Canvas	21
3.3	Modelo de Dados	23
3.4	Implementação	25
3.4.1	Autenticação	25
3.4.2	<i>Back-office</i>	26
3.4.3	<i>Dashboard</i>	29
3.4.4	Visualização de necessidades	30
3.4.5	Visualização de necessidades num calendário	31
3.4.6	Criação e edição de necessidades	32
3.4.7	Geração e leitura de <i>QR Code</i> para marcação de presença	37
3.4.8	Edição de perfil do utilizador	38
3.4.9	Notificações na plataforma	38
3.4.10	Comunicados na plataforma	40
3.4.11	Implementação de <i>timers</i>	41
4	Conclusão	43
A.1	Modelo Entidade-Associação da aplicação	45

Listas de Figuras

2.1	<i>Use Case</i> — Ações gerais.	7
2.2	Funcionalidades a que cada <i>Role</i> tem acesso.	8
2.3	<i>Use Case</i> — <i>Login</i> .	9
2.4	<i>Use Case</i> — Back-Office.	10
2.5	<i>Use Case</i> — Necessidades.	12
2.6	<i>Use Case</i> — Categorias.	12
2.7	<i>Use Case</i> — Necessidade.	13
2.8	<i>Use case</i> — Calendário.	15
2.9	<i>Use case</i> — Announcements.	16
3.1	Diagrama de blocos da aplicação	19
3.2	Arquitetura da plataforma <i>OutSystems</i> [1]	20
3.3	4 Layer Canvas	21
3.4	Modelo de dados	23
3.5	Ecrã de autenticação	25
3.6	Ecrã das necessidades no back-office	26
3.7	Ecrã das categorias	26
3.8	Ecrã <i>ManageUsers</i> com a tab dos utilizadores comuns ativa	27
3.9	Ecrã <i>ManageUsers</i> com a tab dos <i>admins</i> e <i>owner</i> ativa	27
3.10	Ecrã <i>UserMessages</i>	28
3.11	<i>Dashboard</i>	29
3.12	Ecrã das necessidades	30
3.13	Ecrã do calendário.	31
3.14	Ecrã <i>Necessity Creation</i> — Criação parte 1	32
3.15	Ecrã <i>Necessity Creation</i> — Criação parte 2	32
3.16	Ecrã <i>Necessity Creation</i> — Aba de detalhe da necessidade.	33
3.17	Ecrã <i>Necessity Creation</i> — Aba de recursos parte 1.	34
3.18	Ecrã <i>Necessity Creation</i> — Aba de recursos parte 2.	34
3.19	Listagem de código para inicialização do <i>Google Maps</i>	35
3.20	Ecrã <i>Necessity Creation</i> — Aba dos participantes - Candidatura.	35
3.21	Ecrã <i>Necessity Creation</i> — Aba dos participantes - Candidatura efetuada.	36
3.22	Ecrã <i>Necessity Creation</i> — Aba dos participantes - Remoção de um participante da lista.	36
3.23	Listagem de código para acesso à câmera.	37
3.24	Ecrã <i>UserDetail</i>	38
3.25	Ecrã <i>Notifications</i>	39
3.26	Ecrã <i>Announcements</i>	40
3.27	Ecrã <i>AnnouncementCreation</i>	40
3.28	Timers implementados	41
3.29	<i>Timer RemoveUnusedGoogleMapMarker</i>	41

3.30	<i>Timer VerifyNecessityDateToGenerateQRCode</i>	42
3.31	<i>Timer VerifyNecessityStatus</i>	42
4.1	Timeline	44
4.2	Modelo EA	46

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 que se atinja uma boa dinâmica, 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 funcionários mais interessados na sua realização.

1.2 Objetivos

A aplicação proposta visa a implementação de uma plataforma colaborativa, através de uma aplicação web *reactive*, 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 gestão de necessidades criadas pelos utilizadores; divulgação de comunicados para a comunidade empresarial; no contexto de uma necessidade, incluem a partilha de localização, gestão de recursos associados, registo de presenças, registo de candidaturas e gestão das mesmas; a calendarização de necessidades e, por último, um sistema de notificações.

1.3 Abstract

Today's market, increasingly technological, demanding and challenging, imposes a fast pace to the companies that, besides managing their main business processes, also have a significant dynamic of internal activities to help them in their growth and competitiveness. Moreover, companies in the information and technologies sector keep internal activities such as job fairs, sharing knowledge through informal meetings, software component development, professional education offers, playful activities, among others. To achieve a good dynamic, there must be a coordination of resources, which isn't always easy to obtain, because of its allocation to ongoing projects. However, if there is a good planning of time and events through a collaboration platform, this process can be agile, allowing not only the registry and disclosure of the internal necessities of a company but also the acceptance of applications on the behalf of the employees who are most interested in its fulfillment.

The proposed application aims the implementation of a collaboration platform, through a reactive web application, in order to agile the response to a company's internal needs. The goals that this platform aims to achieve consist in the management of necessities created by the users; disclosure of announcements for the company's community; in the context of a necessity, the disclosure of location, management of associated resources, presence registry, application registry and its management; the presentation of necessities and a notification system.

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

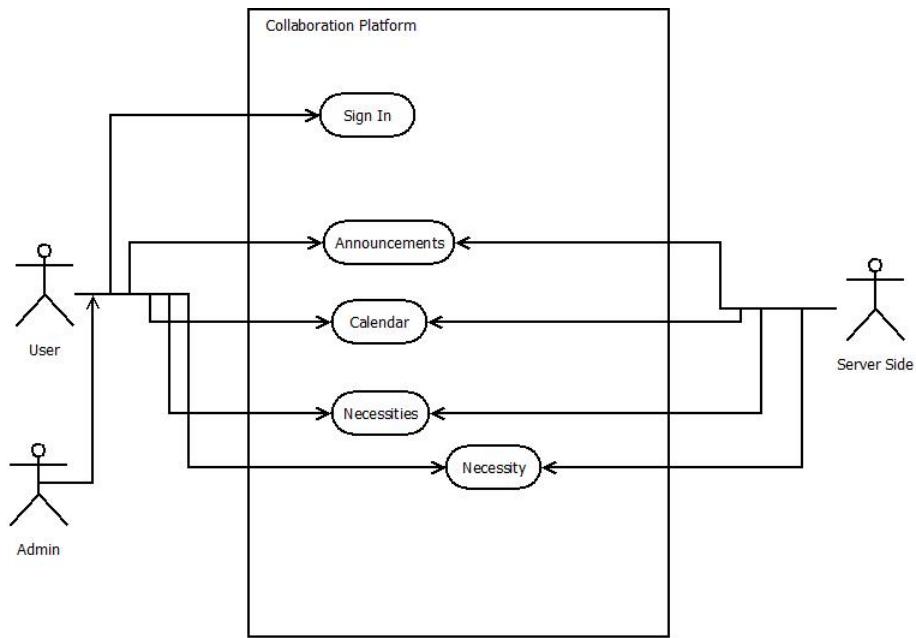


Figura 2.1: *Use Case* — Ações gerais.

2.2.1 Módulos constituintes da *Collaboration Platform* e *roles* associados

A plataforma de colaboração desenvolvida é constituída por dois módulos denominados *back-office* e *UserEndPoints*.

O acesso ao *back-office* é realizado apenas por utilizadores com permissões de administrador ou de *owner*. Um *owner* é descrito como a entidade superior da hierarquia de *roles*, só existe um utilizador com esta permissão. Um administrador é um utilizador com permissões superiores a um utilizador comum, permissões estas descritas na figura 2.2.

No módulo *back-office* são realizadas todas as operações de manutenção e funcionamento da plataforma colaborativa.

O módulo *UserEndPoints* suporta o acesso às principais funcionalidades que a *Collaboration Platform* apresenta e pode ser acedido por todos os utilizadores registados na aplicação.

As funcionalidades a que cada *role* (utilizador comum, administrador e *owner*) tem acesso são descritas na figura 2.2.



Figura 2.2: Funcionalidades a que cada *Role* tem acesso.

2.2.2 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 2.3, 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.

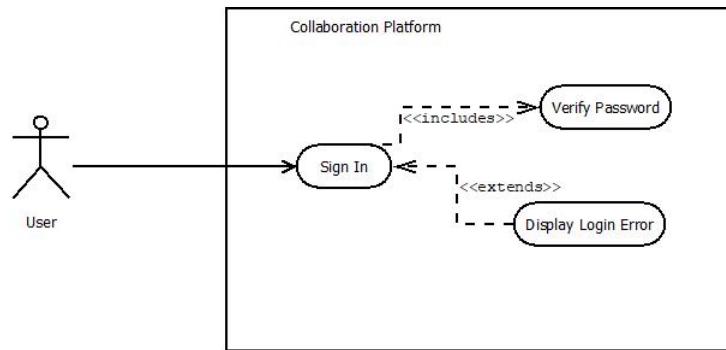


Figura 2.3: *Use Case — Login.*

2.2.3 Manutenção da aplicação e *Back-Office*

Com o objetivo de realizar operações de manutenção e operações que requerem privilégio superior ao de um utilizador comum foi implementado um novo módulo, denominado *back-office*.

Um utilizador com permissões de administrador ou de *owner* tem acesso ao *back-office* onde pode criar ou remover categorias, analisar as necessidades criadas pela comunidade empresarial e escolher se as mesmas são aceites e divulgadas na plataforma ou eliminadas, ver todos os utilizadores da plataforma colaborativa e as suas permissões (sendo que o *owner* pode alterá-las) e analisar mensagens produzidas por um dado utilizador comum com pedidos/sugestões para os utilizadores com permissões superiores (*admin* ou *owner*).

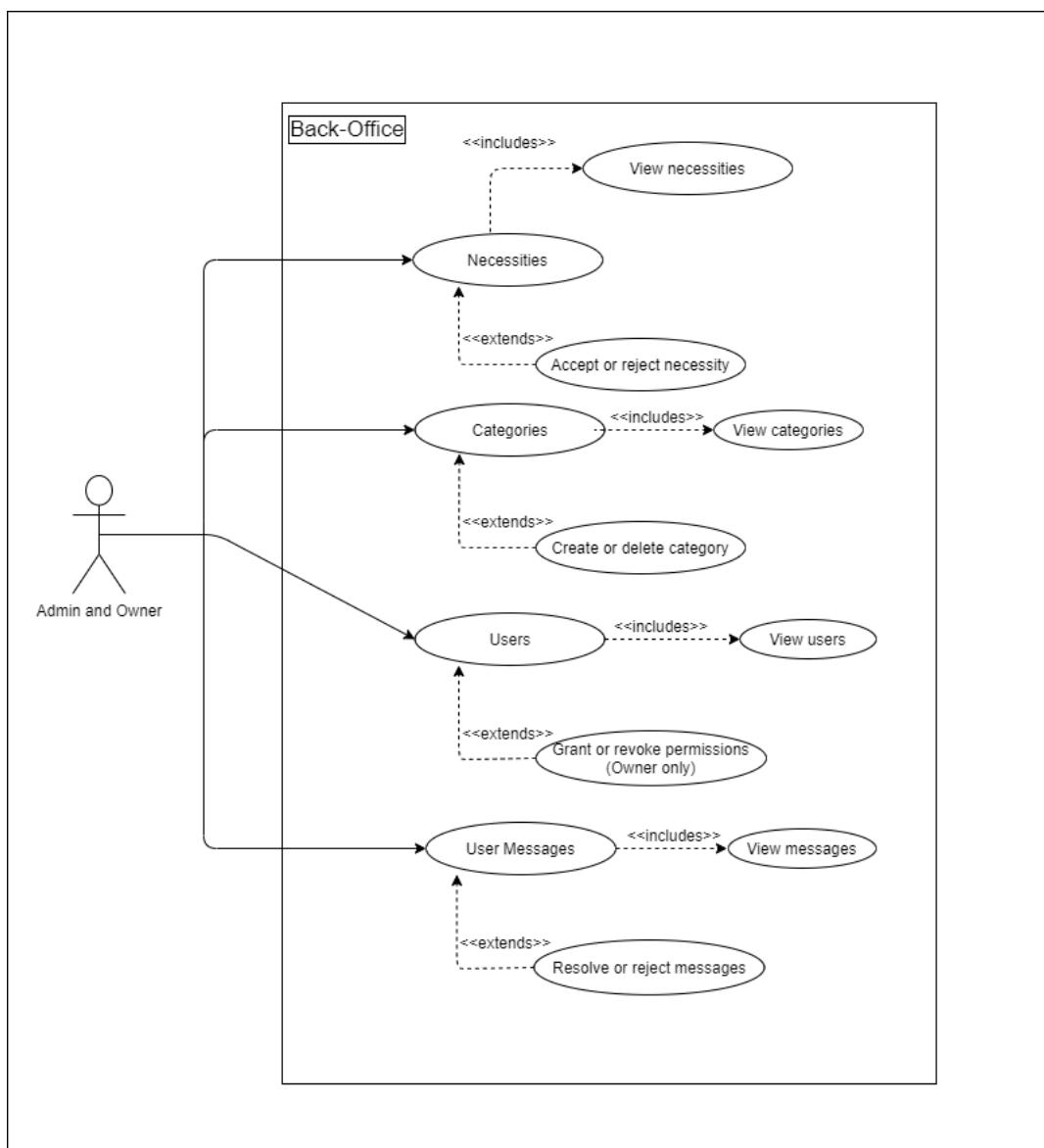


Figura 2.4: *Use Case* — Back-Office.

2.2.4 Home Page — Dashboard

No contexto do módulo *UserEndPoints*, é no domínio do mesmo que se concentram as principais funcionalidades da *Collaboration Platform*, descritas doravante. Estas são acessíveis a todos os utilizadores autenticados e, logo após o *login*, 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.

Este ecrã apresenta o número total de utilizadores registados na aplicação e o total de necessidades criadas até ao momento organizadas por nível de prioridade (Low, Medium e High). Ao carregar num destes níveis de prioridade especificados no ecrã o utilizador é redirecionado para o ecrã das necessidades, apresentando a lista das mesmas com o filtro correspondente à prioridade selecionada.

Este *Dashboard* contém ainda um gráfico circular que demonstra uma estatística relativa ao número de necessidades que existem em cada categoria. A seleção de uma das categorias deste gráfico redirecionará o utilizador para o ecrã das necessidades, apresentando a lista das mesmas com o filtro correspondente à categoria selecionada.

É também apresentado o *top 5* de necessidades criadas mais recentemente e aceites por utilizadores com permissões de administrador (ou pelo *owner*), onde cada uma tem um link que redireciona o utilizador para o detalhe da mesma.

Por fim, é apresentado o top 5 de utilizadores que têm mais necessidades criadas para cada um dos níveis de prioridade que uma necessidade pode ter.

2.2.5 Registo das necessidades internas da empresa, gestão de candidaturas, associação de recursos, notificações e integração com o google maps

A principal funcionalidade que a *Collaboration Platform* apresenta é a de gestão das necessidades internas de uma empresa, descrita na figura 2.5.

O conceito subjacente a esta funcionalidade é o de necessidade (*necessity*), que permite virtualizar uma necessidade interna, assegurando os seus detalhes e características. Um utilizador autenticado tem a possibilidade de ver necessidades criadas pela comunidade e criar, editar, eliminar ou arquivar uma necessidade da sua autoria.

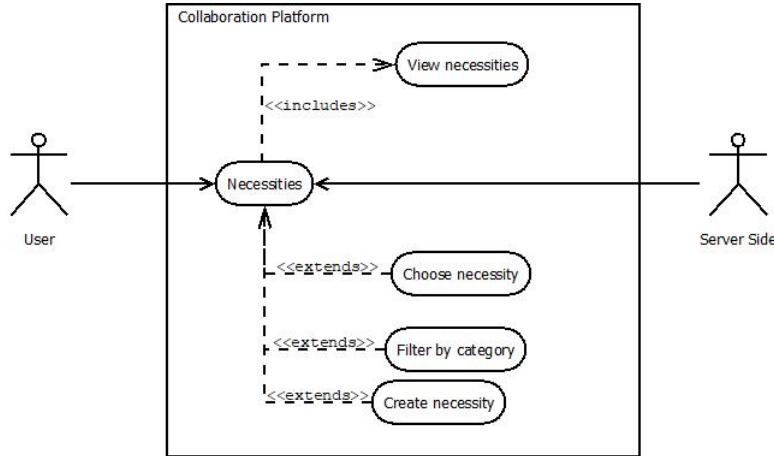


Figura 2.5: *Use Case* — Necessidades.

Posto isto, é fundamental que exista um ecrã que apresente todas as necessidades criadas pela comunidade da empresa. Um utilizador, caso queira registar uma necessidade, irá escolher a categoria que melhor se adequa à mesma.

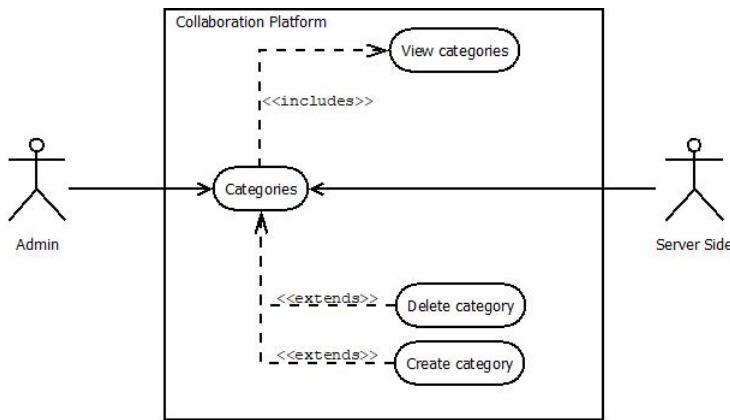


Figura 2.6: *Use Case* — Categorias.

Deste modo, 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*. Apenas um utilizador com permissões de administrador ou o *owner* podem criar ou eliminar novas categorias no *back-office* da aplicação, como descrito na figura 2.6.

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 após 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 á necessidade criada pelo próprio no ecrã de detalhe na *aba* 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.

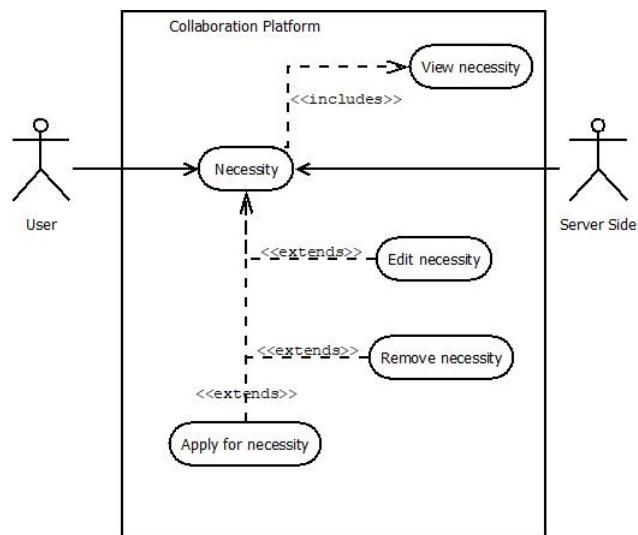


Figura 2.7: Use Case — Necessidade.

Exemplificando, para criar uma necessidade cuja motivação é um evento ou uma feira de emprego, o utilizador seleciona a categoria *Planning of events*.

Se por outro lado, tiver 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 às necessidades existentes, tendo livre acesso para leitura dos detalhes de cada uma.

O autor de uma necessidade é o utilizador que a criou. Tem as permissões de poder editar, eliminar ou arquivar a mesma, associar recursos e gerir participantes/colaboradores.

No contexto da necessidade, é definida uma linha temporal que consiste numa primeira fase de candidaturas para escolha do orientador (ou orientadores), dando origem ao conceito de "*Collaborator*". Um utilizador adquire este cargo, para uma dada necessidade, após ver a sua candidatura aceite pelo autor da mesma. Este utilizador adquire as mesmas permissões de um participante comum e adquire também as permissões de poder editar a necessidade, com exceção da data do evento e de aceitar ou rejeitar participantes.

O autor de uma necessidade, no ecrã de detalhe da mesma, terá acesso a quem se candidatou para a orientar (ser *Collaborator* nesta necessidade), podendo escolher um ou mais orientadores com base na descrição apresentada pelos mesmos. A linha temporal definida no contexto de uma necessidade tem ainda uma segunda fase de candidaturas para permitir o acesso como participante. Um utilizador torna-se um participante após ser aceite pelo autor no caso de haver filtragem dos candidatos ou, no caso de não se verificar a opção de haver filtragem, a candidatura é aceite automaticamente. O cargo de participante permite também a participação nos recursos associados à necessidade.

Todos os utilizadores autenticados podem ver quem foi aceite como participante e como colaborador.

Com o intuito de poder realizar a marcação de presenças de cada um dos participantes a uma necessidade, foi introduzido na plataforma o conceito de *QR Code*. Os mesmos são gerados individualmente para cada participante no dia em que o evento terá lugar e os utilizadores com papéis de autor ou colaborador da necessidade podem fazer scan dos *QR Codes* de cada um dos participantes. Após o scan de um *QR Code* de um dado participante, o mesmo será sinalizado como presente.

2.2.6 Divulgação e calendarização das necessidades

Com o objetivo de divulgar e calendarizar as necessidades internas da empresa, a barra de navegação da aplicação apresenta um botão com o nome *Calendar* que, quando pressionado, 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 utilizador participará.

Após a seleção de um dia no calendário, são apresentadas as necessidades, dando a possibilidade ao utilizador de ver os detalhes individuais após carregar numa delas, num novo ecrã. As ações possíveis de utilização deste calendário estão presentes na figura 2.8. Uma necessidade, no contexto do calendário, é referida como um evento numa determinada data.

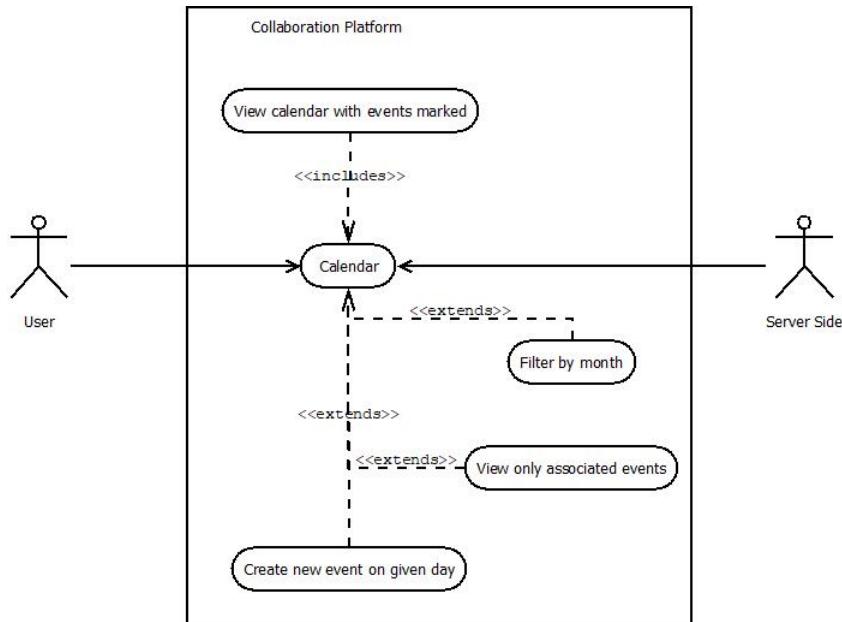


Figura 2.8: *Use case — Calendário.*

2.2.7 Divulgação de comunicados

Para ser possível divulgar comunicados da empresa de forma uniforme por todos os seus colaboradores, a barra de navegação da aplicação apresenta um botão denominado *Announcements* que, quando pressionado, abre um ecrã 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.

Um utilizador comum pode, no mesmo ecrã em que são apresentados os comunicados, enviar uma mensagem para outros utilizadores com permissões mais elevadas de modo a comunicar sugestões ou observações. Um exemplo de uma mensagem a enviar pode ser quando é necessário criar uma nova categoria, algo que apenas administradores e o *owner* podem fazer. Estas mensagens são apresentadas no *back-office* da plataforma onde cada utilizador que tenha acesso ao mesmo pode concretizar os pedidos que constam nas mensagens, marcando-as como "resolvidas" ou rejeitá-las, marcando-as como "rejeitadas". As ações possíveis no contexto do ecrã *Announcements* estão descritas na figura 2.9.

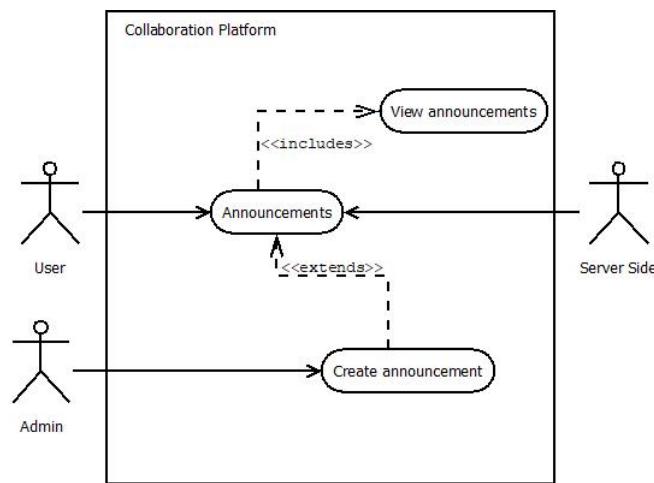


Figura 2.9: Use case — Announcements.

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-feitas 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 permite 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

Solução proposta

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 são abordadas as decisões tomadas no desenvolvimento do projeto, tendo em conta a plataforma *OutSystems* [1].

Na secção 3.1 encontra-se o diagrama de blocos da solução.

Na secção 3.2 encontra-se uma breve análise da arquitetura da plataforma *OutSystems* [1] e a arquitetura *4 Layer Canvas* desenvolvida.

Na secção 3.3 encontra-se o modelo de dados e uma síntese do mesmo.

Na secção 3.4 encontram-se todos os detalhes e abordagens adotadas na implementação da solução.

3.1 Diagrama de blocos da solução

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

No mesmo constam as permissões (*roles*) que um utilizador registado pode apresentar e quais as funcionalidades a que cada *role* pode aceder nos módulos constituintes da aplicação, *UserEndPoints* e *Back-Office*.

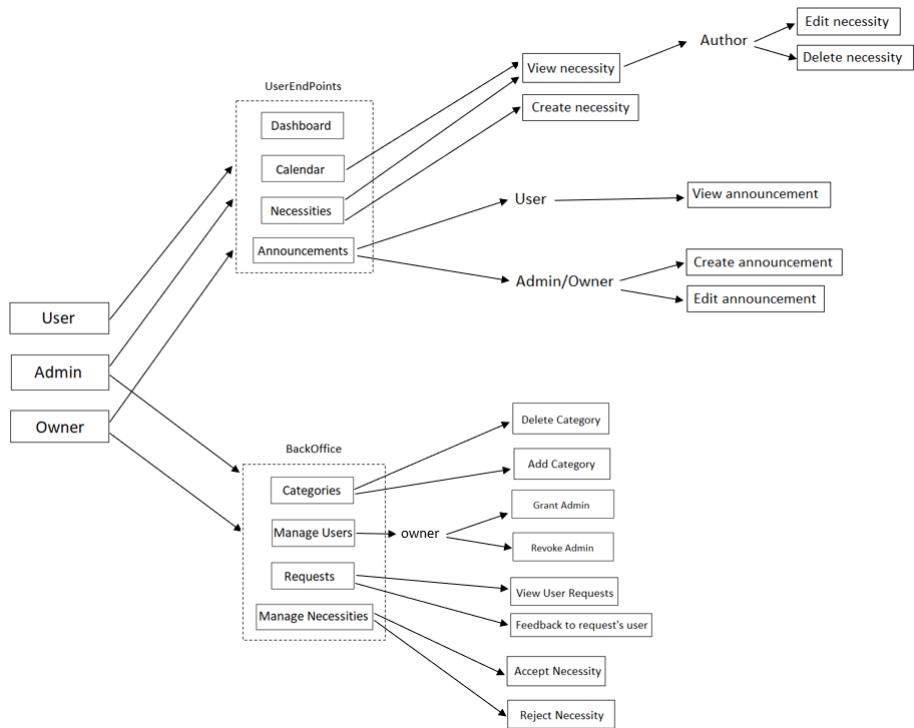


Figura 3.1: Diagrama de blocos da aplicação

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.

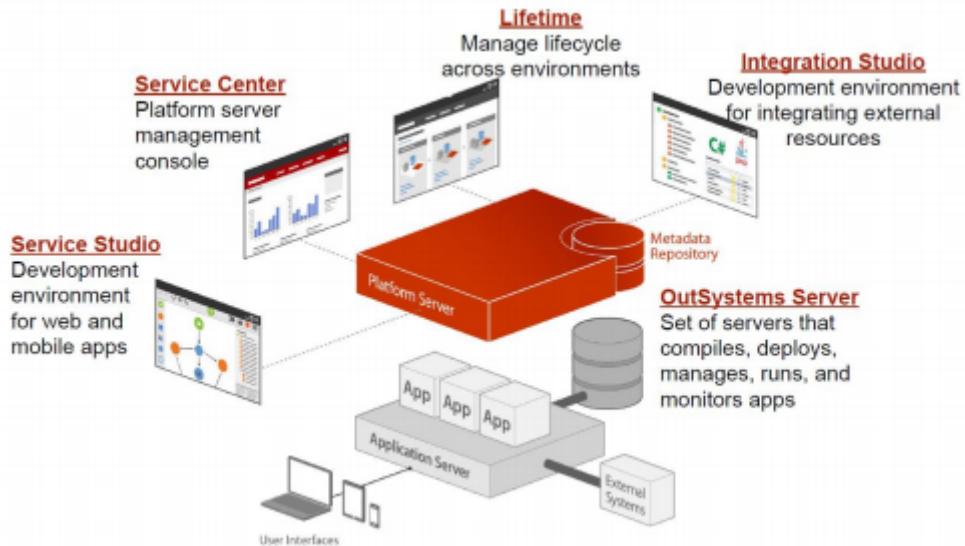


Figura 3.2: Arquitetura da plataforma *OutSystems* [1]

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.

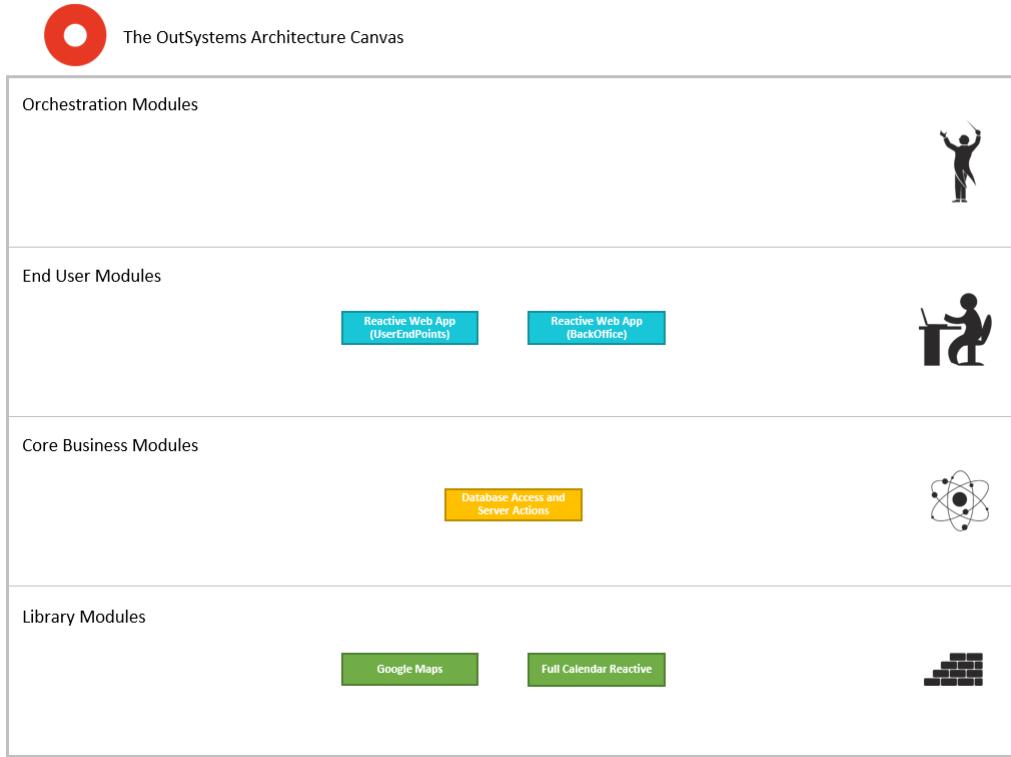


Figura 3.3: 4 Layer Canvas

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 facto do 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*, o que possibilita a apresentação de mapas nos recursos de uma necessidade (descrito na subsecção 3.4.6), e com o *Full Calendar Reactive* que é utilizado para apresentar o calendário no ecrã *Calendar* descrito na subsecção 3.4.5.

De seguida temos a *Core Layer*, onde definimos as entidades de domínio e operações de acesso ao servidor. Toda esta dinâmica está implementada através do módulo *ServerLogic*, onde é coordenada toda a lógica da aplicação.

Por fim a *End User Layer* onde são definidos os módulos *UserEndPoints* e *Back-Office* que suportam os ecrãs e a lógica de cliente.

O tema apresentado em ambos os módulos foi desenvolvido pelo grupo, devido ao facto da solução ser concretizada como uma aplicação *web reactive* e não haver temas já implementados na *forge* da *OutSystems* para este tipo de aplicação. O mesmo foi desenvolvido através da manipulação de *CSS* e de ajustamentos cuidados para apresentar uma *User Interface* apelativa.

3.3 Modelo de Dados

O modelo Entidade-Associação da solução desenvolvida pode ser consultado no anexo 4.2. O conceito predominante no modelo de dados, (figura 3.4), é o de necessidade, representado pela tabela *Necessity*.

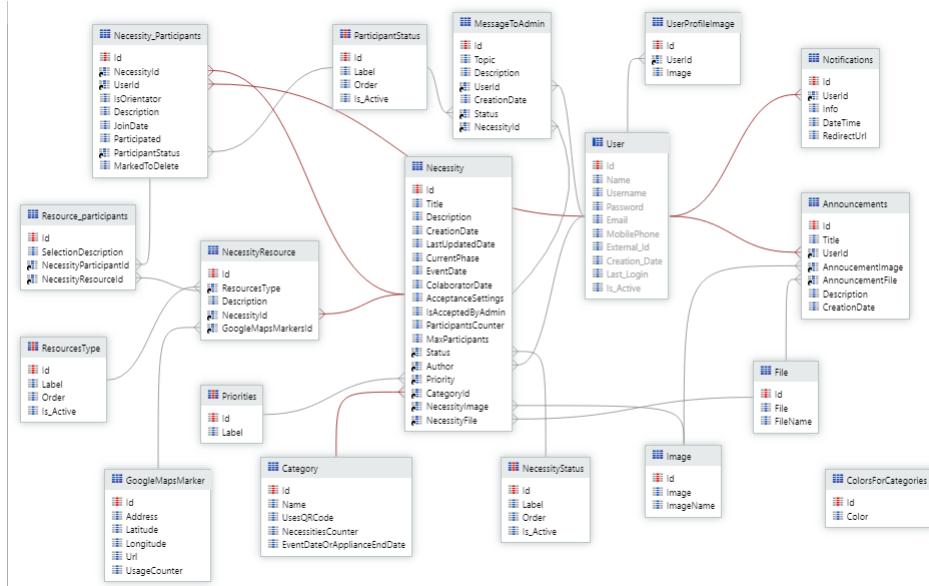


Figura 3.4: Modelo de dados

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 colaboradores ou participantes, a *ColaboratorDate* que representa a data limite das candidaturas à posição de colaborador, *AcceptanceSettings* que indica se para a necessidade em causa são aceites todos os participantes ou se existe uma filtragem por parte do autor, *isAcceptedByAdmin* que indica se a necessidade criada já foi aceite por um administrador ou pelo *owner* no *back-office*, *ParticipantsCounter* que indica o número de participantes desta necessidade, e por último, *MaxParticipants* que serve para guardar o número máximo de participantes que podem existir nesta necessidade.

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 necessidade, o identificador do utilizador, a descrição associada à candidatura, se esta candidatura é para posição de colaborador ou de participante, a data da candidatura, se o utilizador participou na necessidade, qual o estado da candidatura (dado pela entidade estática *ParticipantStatus*, cujos valores representam se a candidatura está pendente - *pending*, aceite - *accepted* ou rejeitada - *rejected*) e se este participante pediu para remover a sua participação na necessidade e está assinalado para remoção da lista de participantes.

De modo a que seja possível guardar informação sobre se um participante vai usufruir dos recursos associados a uma necessidade, é definida a entidade *Resource_Participants*. Esta entidade guarda informações como a descrição da escolha do recurso, o id do participante e o id do recurso.

Com o intuito de suportar a funcionalidade de envio de mensagens entre utilizadores comuns e utilizadores com permissões mais elevadas, foi criada a tabela *MessageToAdmin*. Esta tabela tem os atributos necessários para armazenar o tópico de uma mensagem, o seu conteúdo, o utilizador que a criou, a sua data de criação e, por último, o estado da mesma, que pode tomar os valores presentes na entidade estática *ParticipantStatus*. Estes valores são *accepted* para mensagens cujo conteúdo foi cumprido ou aceitado, *rejected* para mensagens cujo conteúdo não o foi e *pending* para aquelas que ainda não obtiveram resposta.

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 que tenha a permissão de administrador, cujos privilégios incluem editar as categorias representadas pela entidade *Category*. Esta entidade tem como atributos o nome da categoria criada, se é uma categoria cujas necessidades vão necessitar de confirmação de presença através de *QR Code*, quantidade de necessidades criadas associadas a esta categoria e se o dia do evento presente em cada necessidade associada a esta categoria corresponde ao dia em que o evento tem lugar ou simplesmente ao final da data de candidaturas.

Um pequeno exemplo para melhor entendimento deste último atributo é, numa categoria que tenha como conceito a procura de utilizadores para desenvolver peças de software a data associada às necessidades (atributo *eventDate* da entidade *Necessity*) é de fim da fase de candidaturas para produzir a peça de software em questão, enquanto que uma categoria que tenha como conceito atividades lúdicas terá o atributo *eventDate* das suas necessidades como a data em que o evento irá ocorrer.

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.

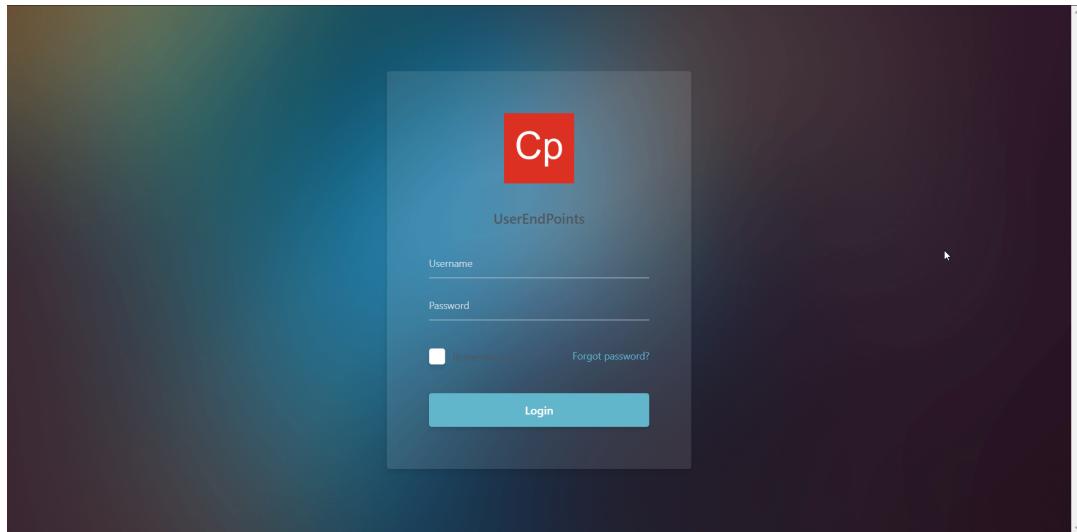


Figura 3.5: Ecrã de autenticação

3.4.2 Back-office

O *back-office* da *Collaboration Platform* tem como principal motivação a manutenção da mesma e o acesso a funcionalidades que requerem permissões superiores, nomeadamente de *admin* ou de *owner*. Posto isto, o *back-office* é implementado num novo módulo e apresenta quatro ecrãs distintos.

No ecrã *Necessities* os utilizadores com acesso ao *back-office* têm a possibilidade de verificar as necessidades criadas pela comunidade da empresa, podendo aceitá-las ou eliminá-las. Estas necessidades são apresentadas sobre a forma de uma lista, tal como demonstrado na figura 3.6. Ao pressionar uma delas, serão apresentados os detalhes da mesma e dois ícones que permitirão aceitar ou rejeitar.

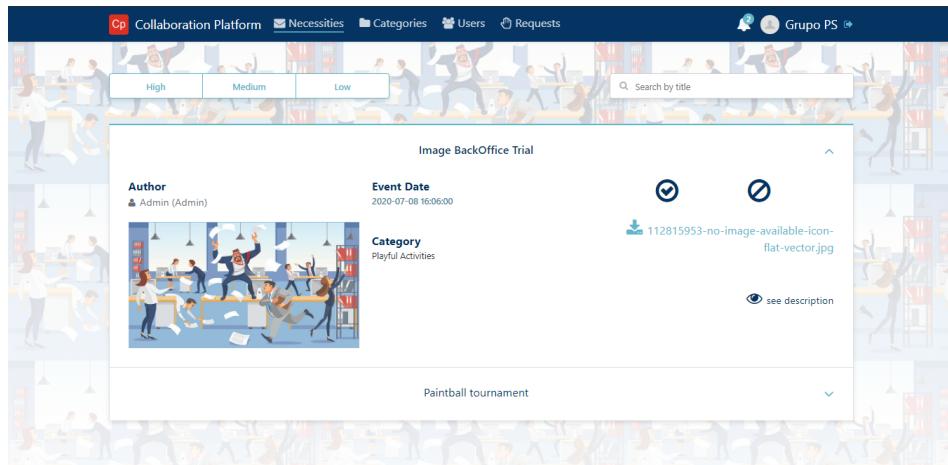


Figura 3.6: Ecrã das necessidades no back-office

O ecrã *Categories* tem como objetivo a criação e remoção de categorias. As mesmas são apresentadas numa tabela, como é possível verificar na figura figura 3.7, e cada uma tem uma checkbox associada que, quando selecionada, irá deixar a categoria apta para ser eliminada, ao carregar no botão delete. Para criar uma nova, basta carregar no botão create e preencher o nome, se é necessária verificação de presenças por *QR Code* e como é interpretada a data de evento de uma necessidade associada a esta nova categoria.

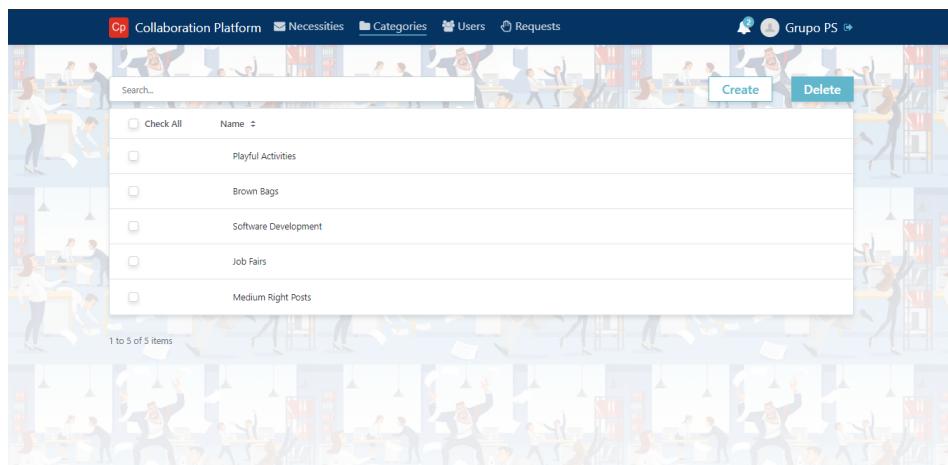


Figura 3.7: Ecrã das categorias

No ecrã *ManageUsers* é possível ver todos os utilizadores da plataforma separados entre utilizadores comuns e utilizadores com permissões mais elevadas, através de duas *tabs*. Deste modo, na tab dos utilizadores comuns (representada na figura figura 3.8) existe a possibilidade de alterar as permissões associadas a um utilizador, concedendo-lhe permissões superiores através do botão *Grant admin*. Na tab de *Admins*, apresentada na figura figura 3.9, existe a possibilidade de despromover um utilizador de administrador a utilizador comum, através do botão *revoke admin*.

Estas mudanças de *roles* apenas podem ser feitas pelo *owner*, a entidade superior da hierarquia de *roles*. Um utilizador com permissões de administrador apenas pode ver os utilizadores e o seu *role*.

Name	Email	Phone	Creation Date	Actions
David	goncalo9marco@hotmail.com	968225962	2020-04-20 15:32:07	<button>Grant Admin</button>
dummy			2020-07-03 00:38:24	<button>Grant Admin</button>

Figura 3.8: Ecrã *ManageUsers* com a tab dos utilizadores comuns ativa

Name	Email	Phone	Creation Date	Actions
Admin			2020-06-15 19:28:30	<button>Revoke Admin</button>
Grupo PS	grupomaravilhaps@outlook.com		2020-04-07 22:38:50	<button>Revoke Admin</button>

Figura 3.9: Ecrã *ManageUsers* com a tab dos *admins* e *owner* ativa

Com o intuito de ver as mensagens criadas pela comunidade da empresa, o ecrã *UserMessages* (figura 3.10) apresenta-as sobre a forma de uma lista e, ao carregar numa mensagem, é apresentado o detalhe da mesma.

Cada utilizador com permissões para aceder ao *back-office* pode sinalizar uma mensagem que esteja por resolver ("unresolved") como resolvida ("resolved") se aceitar/cumprir o conteúdo da mensagem ou como "rejected" se não for o caso.

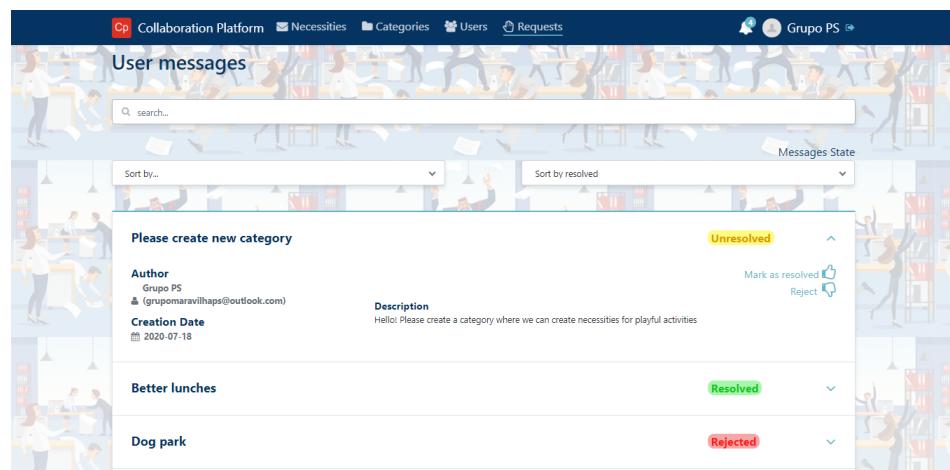


Figura 3.10: Ecrã *UserMessages*

3.4.3 Dashboard

A um utilizador autenticado é apresentado o ecrã *dashboard*, apresentado na figura 3.11, assim que inicia sessão na 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 *widget carousel*, com a descrição "*Recently created necessities*", a partir do qual é possível observar as necessidades que foram criadas mais recentemente.

Posto isto, do lado direito do *widget carousel*, é demonstrado ao utilizador um gráfico com informação sobre as cinco categorias existentes que apresentam um maior número de necessidades criadas. 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.

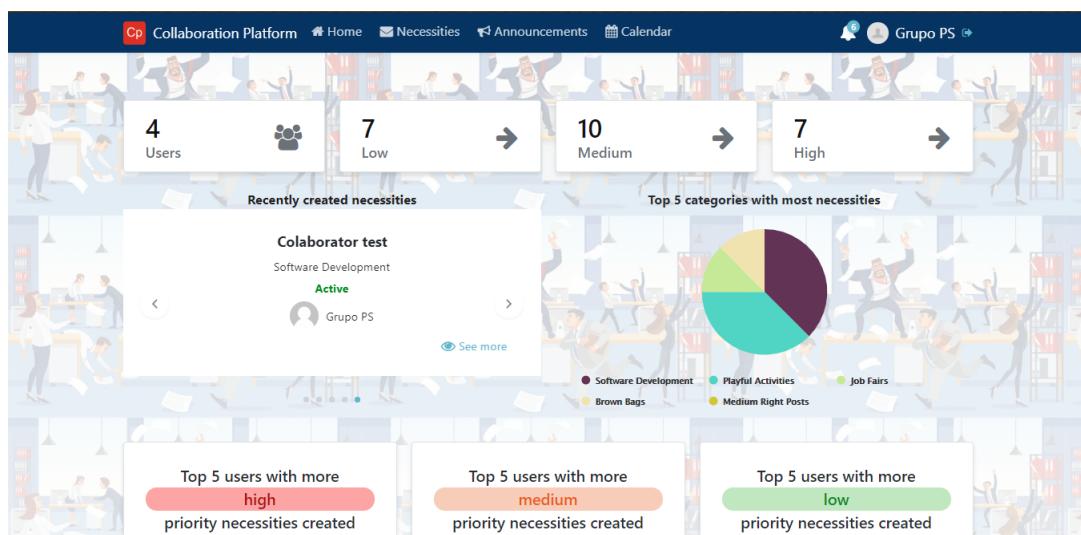


Figura 3.11: *Dashboard*

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 3.12.

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, data em que a necessidade irá decorrer e o número de participantes.

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).

O *widget switch* presente neste ecrã com a descrição *"show only my necessities"*, quando ativo, reorganiza a tabela das necessidades de modo a só mostrar as que o utilizador é autor.

É importante referir que todas as necessidades apresentadas neste ecrã foram previamente aceites por um utilizador com *role* de *admin* ou *owner* no *back-office*, com exceção de quando o *switch* de mostrar apenas as necessidades do utilizador está ativo. Neste caso, as necessidades aparecem mesmo que ainda não tenham sido aceites, mas apenas ao utilizador que neste caso é sempre o autor das mesmas.

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.

Necessities:						
Title	Category	Priority	State	Creation Date	Event Date	Participants
Dormir	Playful Activities	High	Closed	2020-07-07	2020-07-07 00:00:00	0
Colaborator test	Software Development	Low	Closed	2020-07-07	2020-07-11 00:00:00	0
Put Sunscreen On the Kids	Playful Activities	High	Closed	2020-07-03	2020-07-04 01:54:00	0
CentOS debug	Software Development	Medium	Closed	2020-07-02	2020-07-03 13:50:00	0 / 8
iPhone debug	Software Development	Medium	Closed	2020-07-01	2020-07-09 14:02:00	0 / 17

Figura 3.12: 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 (figura 3.13) 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.

Foi utilizado o componente *FullCalendarReactive*, presente na *forge* da plataforma *OutSystems*, de modo a apresentar ao utilizador um calendário interativo. Para preencher o calendário, para cada necessidade, é criado um objeto *Event* com a sua informação.

Cada necessidade estará representada no calendário com uma cor diferente de forma a distinguir entre necessidades criadas pelo utilizador, necessidades associadas, nomeadamente em que o utilizador participa, e necessidades criadas por outros utilizadores. Se o utilizador carregar numa necessidade, este é redirecionado para o seu ecrã de detalhe.

É possível filtrar necessidades pela sua categoria, através de um *popover menu* que contém todas as categorias existentes. Caso seja selecionada uma categoria, são removidas todas as necessidades presentes no calendário, e de seguida, o mesmo é renderizado com as novas necessidades filtradas.

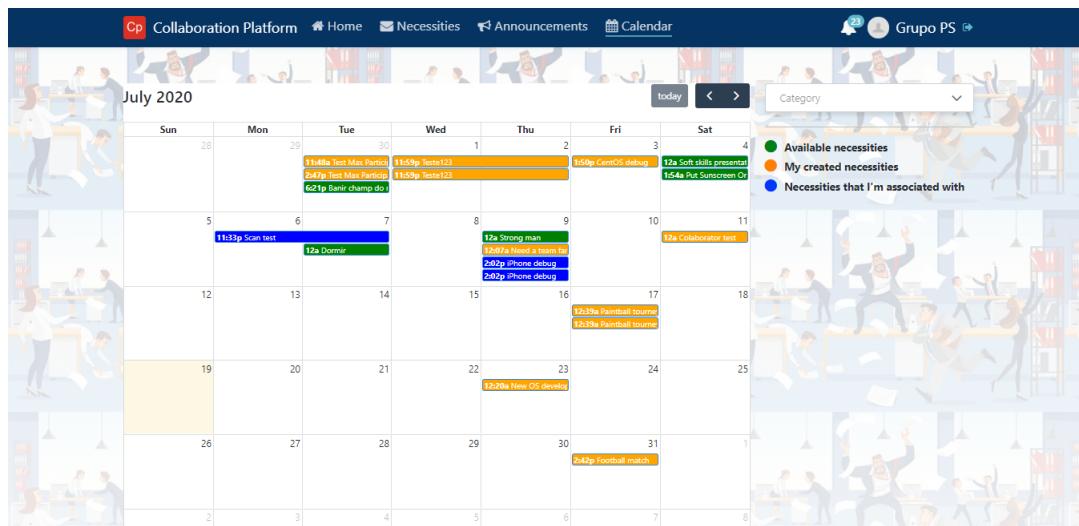


Figura 3.13: Ecrã do calendário.

3.4.6 Criação e edição de necessidades

O ecrã *NecessityCreation* (figura 3.14 e 3.15) é utilizado para criar, editar, remover, arquivar 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, o número máximo de participantes, uma imagem e um ficheiro para serem associados à necessidade, a data em que a mesma irá decorrer e 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 title field labeled 'Title *' with a placeholder 'Leave empty for unlimited'. Below it is a 'Description *' field. A 'Category *' dropdown menu is open, showing 'Choose a category'. To the right, there's a 'Priority' section with three options: 'Low' (blue), 'Medium' (light blue), and 'High' (green). Underneath, there's a section for filtering applicants: 'Accept all applicants' (radio button) and 'Filter each appliance' (radio button, selected). A file upload field 'Choose any file...' is also present. Further down, there's a field 'How many people can go?' with a placeholder 'Leave empty for unlimited'. A 'Need Collaborator?' checkbox is checked. On the right side, there's a large image input field with a placeholder 'Choose an image...' and a 'Save' button at the bottom right.

Figura 3.14: Ecrã *Necessity Creation* — Criação parte 1

The screenshot shows the continuation of the 'Necessity' creation form. It includes a date picker for 'End date for appliances*' with a calendar view for July 2020. A file input field 'Choose an image...' is shown next to a placeholder image of a mountain landscape. At the bottom, there are 'Cancel' and 'Save' buttons.

Figura 3.15: 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, após pressionar o ícone de editar o ecrã (figura 3.16) apresentará os campos já preenchidos com os dados atuais da necessidade e a possibilidade de os alterar

livremente. Um utilizador que seja um colaborador de uma dada necessidade pode também editá-la, mas não tem permissões para a eliminar.

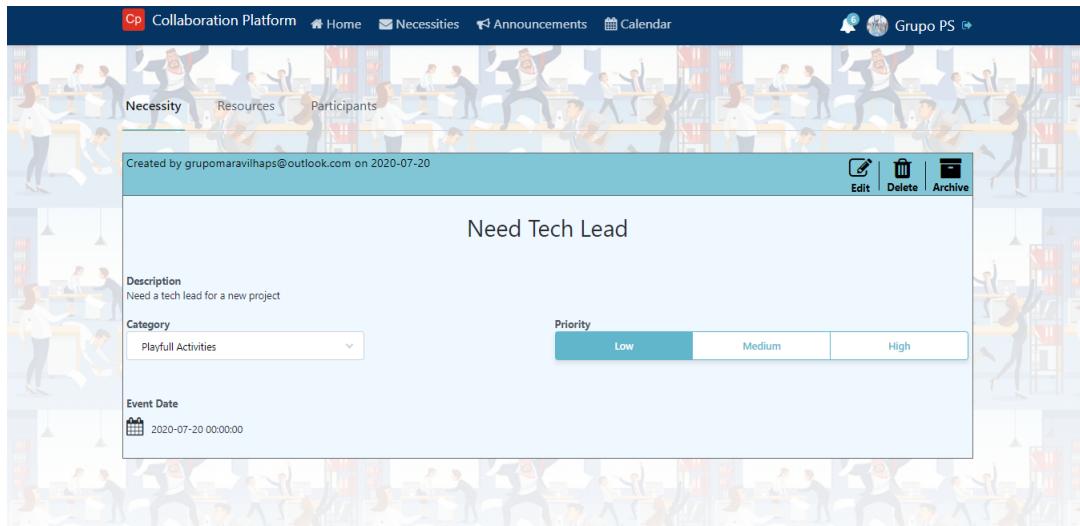


Figura 3.16: Ecrã *Necessity Creation* — Aba de detalhe da necessidade.

É também neste ecrã (figura 3.17) 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 ou editar os existentes.

Um utilizador que seja participante numa dada necessidade pode usufruir dos recursos associados à mesma. Para um dado recurso, o participante pode escolher usufruir deste pressionando o botão *Confirm Attendance*.

O autor da necessidade, assim como os colaboradores associados, podem fazer *download* de um ficheiro *excel* cujo conteúdo consiste nos utilizadores que vão a determinado recurso. Esta funcionalidade permite, de uma forma flexível, saber quem são os participantes que vão a determinado recurso. 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 3.18) na plataforma é realizada através da comunicação com a API do *Google Maps* [7]. 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. Para efeitos de reduzir a quantidade de pedidos feitos à API *Google Maps JavaScript API* com o incentivo de obter um mapa, cada mapa é guardado num objeto *maps* que funciona como um contentor de objetos de mapas e os seus marcadores. Cada mapa é inicializado apenas quando for necessário mostrá-lo em conjunto com um recurso da necessidade, como visto na figura 3.19.

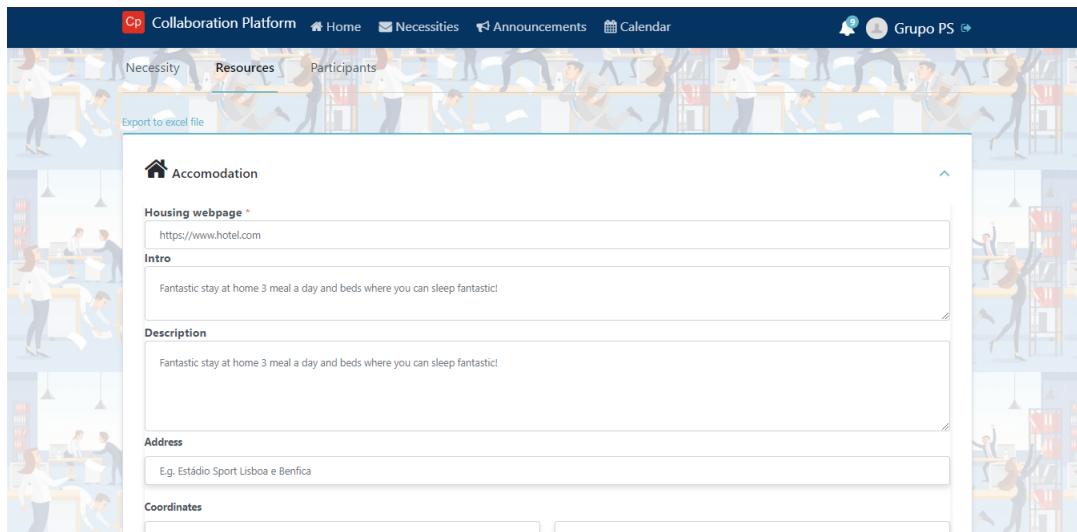


Figura 3.17: Ecrã *Necessity Creation* — Aba de recursos parte 1.

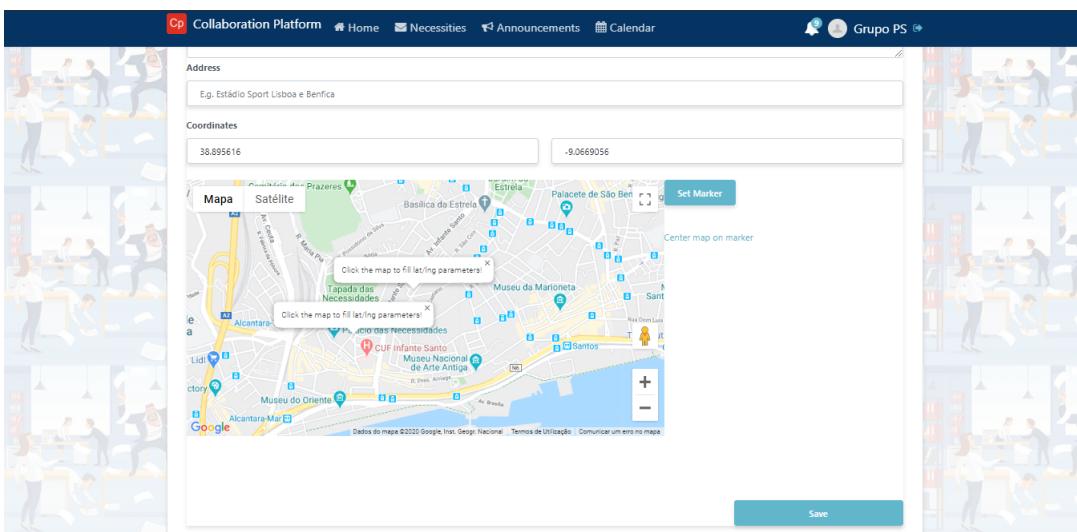


Figura 3.18: Ecrã *Necessity Creation* — Aba de recursos parte 2.

```

1 //init the map if it doesn't exist yet
2 var map = window.maps[$parameters.MapContainer].map || new google.maps.Map(document.getElementById($parameters.MapContainer), {
3   center: {
4     lat: 38.706844,
5     lng: -9.169633
6   },
7   zoom: 16
8 });
9
10 //update current map to pool of maps
11 window.maps[$parameters.MapContainer].map = map;
12 /*
13 * Get current user location, if allowed
14 */
15 if (navigator.geolocation) {
16   navigator.geolocation.getCurrentPosition(success);
17 }
18
19 /*
20 * On success of getting user geolocation recenter map to current location
21 */
22 function success(position) {
23   var crd = position.coords;
24   var lat = crd.latitude;
25   var lng = crd.longitude;
26   map.setCenter({lat: lat, lng: lng});
27   $actions.SetLatLng(lat, lng);
28 }
29
30 if($parameters.lsAuthor) {
31   //small text display over the map
32   var infowindow = new google.maps.InfoWindow(
33     {content: 'Click the map to fill lat/lng parameters!', position: map.getCenter()});
34   //show the small text display
35   infowindow.open(map);
36 }
37
38 /*
39 * Configure the click listener.
40 */
41 map.addListener('click', function(mapMouseEvent) {
42   var position = {lat: mapMouseEvent.latLng.lat(), lng: mapMouseEvent.latLng.lng()};
43   //automatically sets the coords
44   $actions.SetLatLng(position.lat, position.lng)
45   $actions.SetIsAddressAutoCompleted(false);
46 });
47 ...

```

Figura 3.19: Listagem de código para inicialização do *Google Maps*

Na última tab do *widget Tabs* presente no ecrã *NecessityCreation* (figura 3.20) encontra-se a secção relacionada com os participantes à necessidade. Toda a dinâmica relativa a esta tab *"Participants"* do ecrã *NecessityCreation* é suportada por um *Web Block* denominado *ParticipantsBlock*. 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 Colaborator/Participant* e deixando uma descrição, se a filtragem dos candidatos estiver ativa 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*.

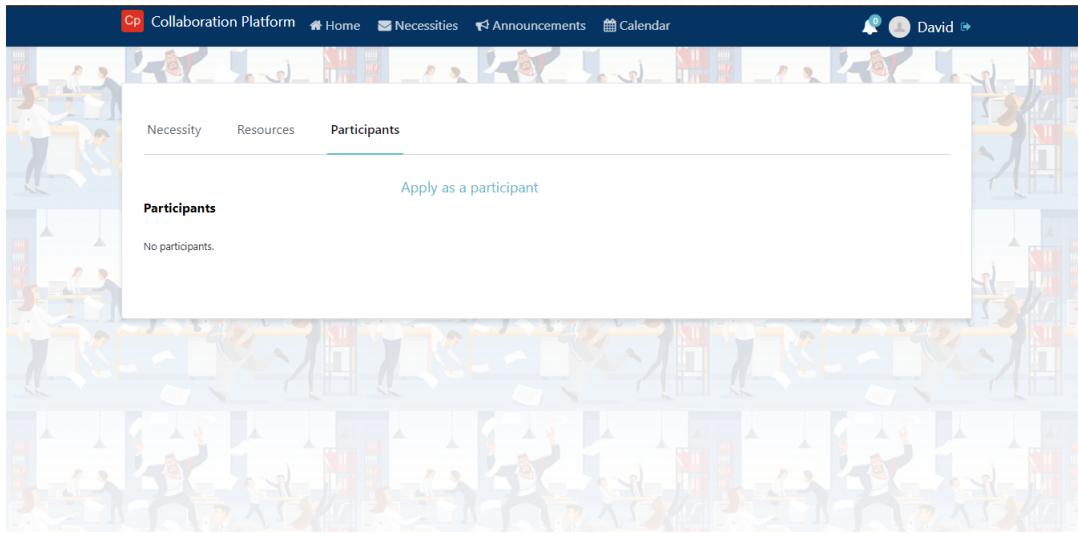


Figura 3.20: Ecrã *Necessity Creation* — Aba dos participantes - Candidatura.

Considere-se que um participante quer remover a sua participação a uma dada necessidade; no caso de ainda não ter sido aceite, isto é, a sua candidatura à necessidade está pendente, o cancelamento é instantâneo ao pressionar o botão *Cancel participation* (apresentado no ecrã da (figura 3.21)). Se já tiver sido aceite, ao carregar no botão *Cancel participation*

irá ser enviada uma notificação para o autor e para os colaboradores da necessidade para que removam este participante da lista. O utilizador, ao pressionar este botão, é desencadeada uma *client action* que posteriormente comunica com o servidor através de uma *server action* para que este participante seja marcado para remover na base de dados.

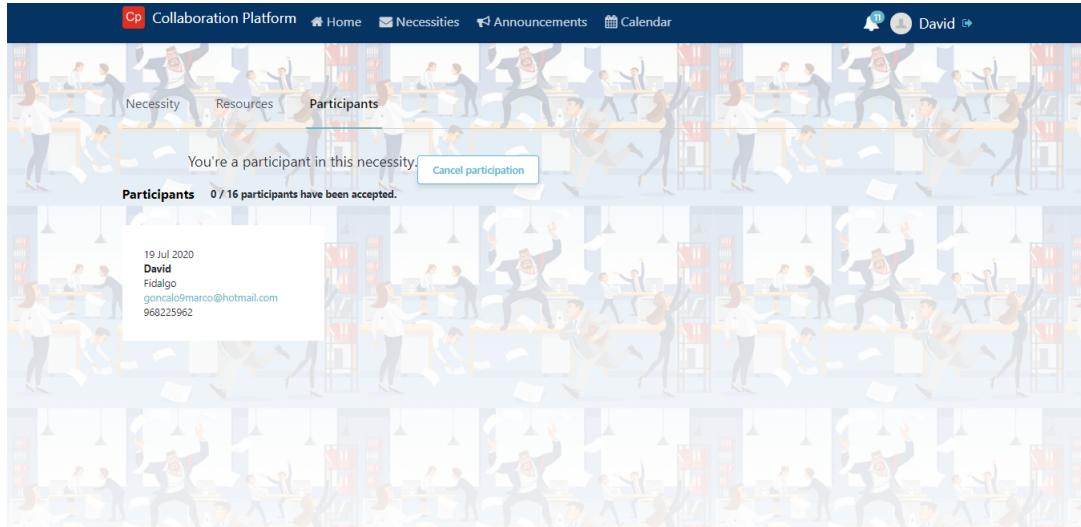


Figura 3.21: Ecrã *Necessity Creation* — Aba dos participantes - Candidatura efetuada.

Na lista dos participantes de uma necessidade (apresentada no ecrã da figura 3.22), um utilizador que esteja marcado para remover apresenta um ícone a vermelho onde um utilizador que seja o autor ou colaborador pode carregar para o remover. Apenas utilizadores que tenham pedido para cancelar a sua participação podem ser removidos.

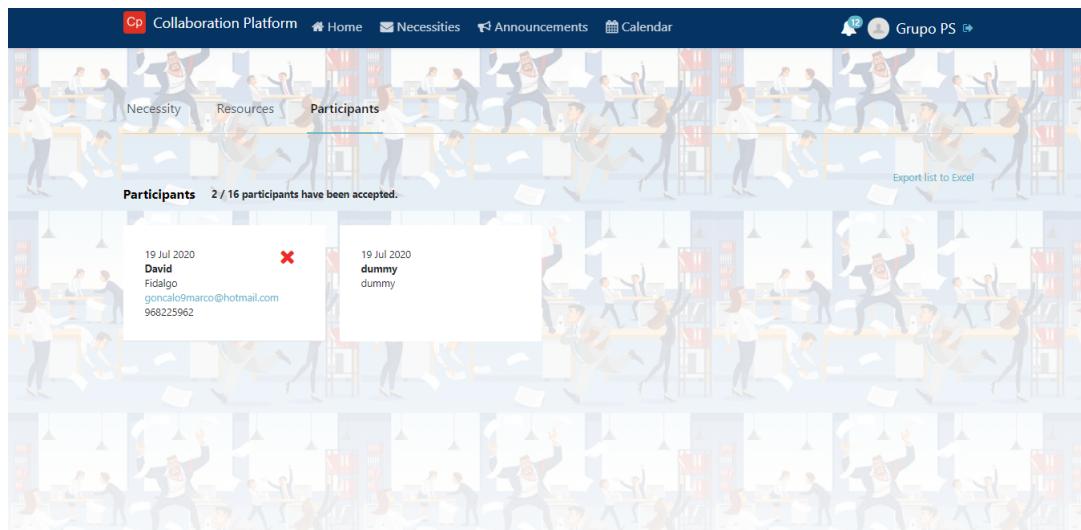


Figura 3.22: Ecrã *Necessity Creation* — Aba dos participantes - Remoção de um participante da lista.

3.4.7 Geração e leitura de *QR Code* para marcação de presença

A geração de *QR code*, para a confirmação de presença numa necessidade caso a categoria da mesma suporte QR code, é feita pelo timer *VerifyNecessityDateToGenerateQRCode* (ver subsecção 3.4.11) no dia de evento definido pelo autor.

Para cada participante será gerado um QR code único para ser lido por um colaborador ou o próprio autor da necessidade em questão.

Semelhante à metodologia de geração de QR codes, é disponibilizado um ecrã de leitura de QR codes apenas visível a colaboradores e autores das respectivas necessidades, estes são notificados da disponibilização do ecrã. Neste ecrã irá ser pedido ao utilizador permissões de utilização da câmara, como visto na figura 3.23, para realizar uma leitura contínua de QR codes pressionando no botão *Scan QR Codes* e de seguida no *Capture* para capturar o conteúdo visto na câmara, se existir um QR code irá ser apresentado uma mensagem de sucesso caso contrário uma mensagem de erro a avisar o utilizador que não foi possível encontrar nenhum QR code. Para realizar o acesso à câmara foi necessário entender os conceitos de *constraints* que é um objeto de *MediaStreamConstraints* que contém informações sobre o tipo de *media* no pedido e os requisitos de cada pedido [10].

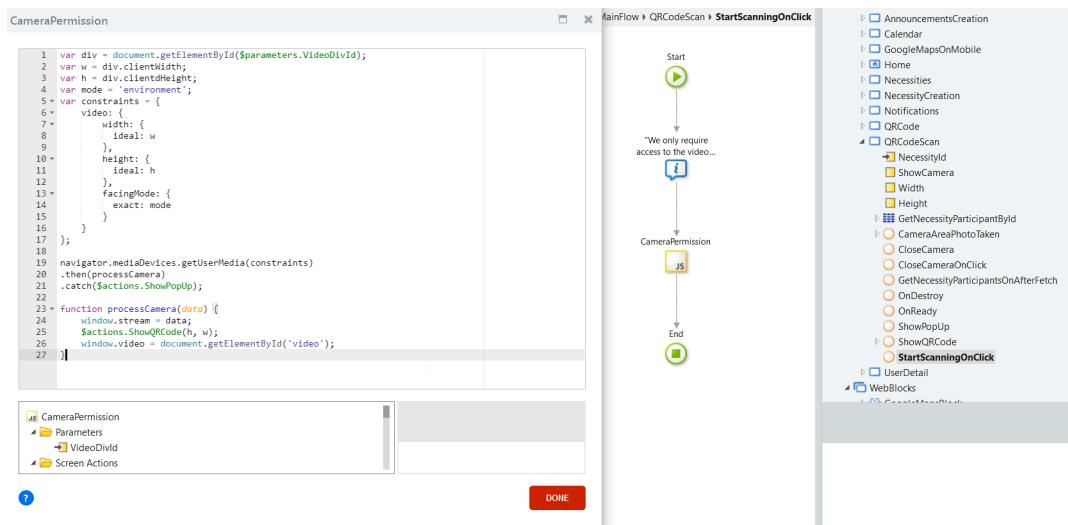


Figura 3.23: Listagem de código para acesso à câmera.

3.4.8 Edição de perfil do utilizador

Com o objetivo de editar o seu perfil, um utilizador ao carregar no seu nome navega até ao ecrã *UserDetail*, representado na figura 3.24.

No contexto deste ecrã, um utilizador tem oportunidade de alterar os campos associados ao seu perfil, nomeadamente o seu nome, nome de utilizador, email, número de telefone e a imagem de perfil.

É importante referir que o *upload* de imagens ou ficheiros de tamanho considerável faz com que a comunicação com o servidor se torne lenta e, em casos extremos, a mesma pode resultar numa exceção do tipo *connection timed out*:

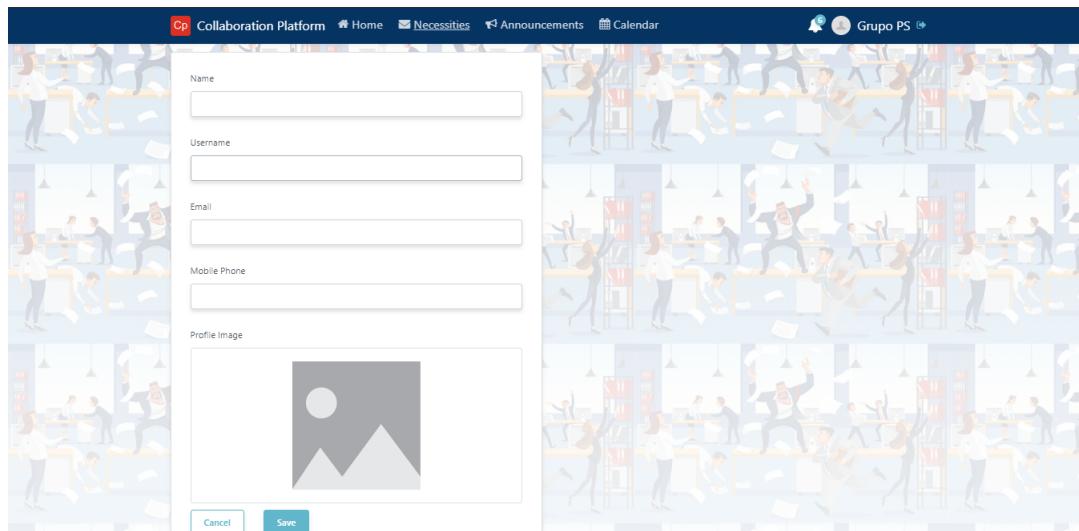


Figura 3.24: Ecrã *UserDetail*

3.4.9 Notificações na plataforma

A plataforma tem um sistema de notificações que suporta o seu envio para os utilizadores, num ecrã (figura 3.25) 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; quando uma necessidade é eliminada ou editada, os participantes da mesma são notificados; quando um dado utilizador quer desistir da sua participação a uma necessidade, o autor da mesma recebe uma notificação para remover esse participante; quando uma mensagem enviada para os administradores da plataforma foi resolvida ou rejeitada; e, por último, quando é o dia em que uma necessidade se vai realizar e a mesma tem *QR code* associado, os participantes recebem uma notificação com um link para o *QR code* gerado.

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.

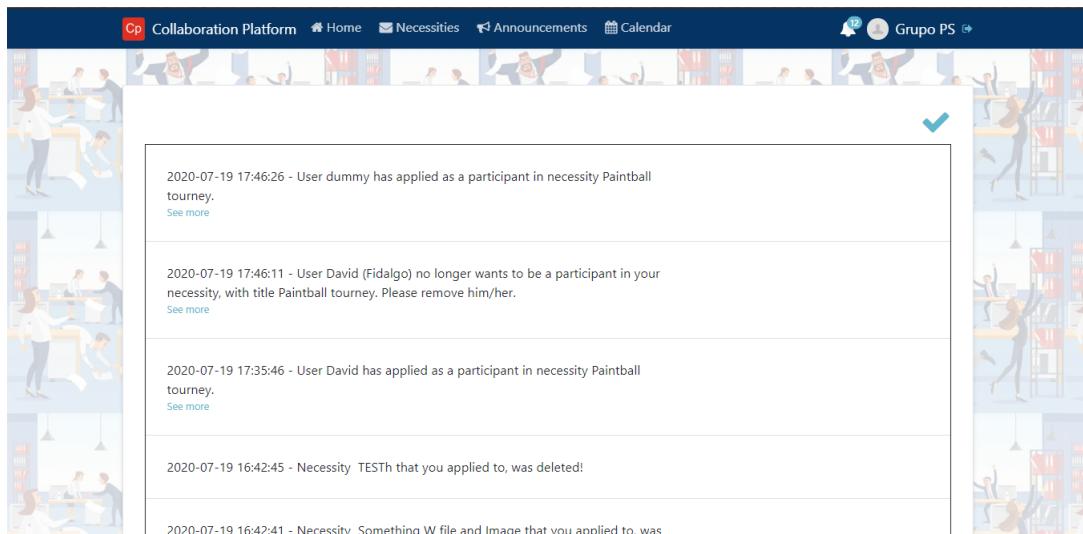


Figura 3.25: Ecrã Notifications

3.4.10 Comunicados na plataforma

A divulgação de comunicados na *Collaboration Platform* tem como objetivo a transmissão de informação de forma uniforme por todos os utilizadores registados.

Um utilizador autenticado acede ao ecrã *Announcements* (figura 3.26) e verifica os comunicados, apresentados sob a forma de uma lista. A seleção de um destes comunicados apresenta os detalhes do mesmo, visto que cada comunicado é apresentado num *widget Accordion*. Um utilizador com permissões de administrador ou o *owner*, ao navegar para o ecrã *Announcements*, têm a possibilidade de criar um novo comunicado pressionando o botão *Create Announcement*, criação esta que é completada no ecrã *AnnouncementCreation* (figura 3.27).

Cada comunicado tem associado um título, uma descrição e, facultativamente, um ficheiro e uma imagem.

The screenshot shows the 'Announcements' section of the Collaboration Platform. At the top, there is a search bar and a 'Sort by...' dropdown. A large 'Create Announcement' button is prominently displayed. Below it, a card titled 'Welcome new Employee' is shown. The card includes details such as 'Author: Grupo PS (grupomaravilhaps@outlook.com)', 'Event Date: 2020-07-19 23:34:27', and a small thumbnail image labeled 'work_panic_700px.jpg'. To the right of the card are two buttons: 'see description' and 'Edit Announcement'. The background features a collage of various office-related illustrations.

Figura 3.26: Ecrã *Announcements*

The screenshot shows the 'Create Announcement' form. It has fields for 'Title *' and 'Description *'. Below these are two input fields: one for 'Choose any file...' with a maximum size of 30MB and another for 'Choose an image...' with a maximum size of 5MB. There is also a preview area showing a small image thumbnail. At the bottom of the form are 'Cancel' and 'Save' buttons.

Figura 3.27: Ecrã *AnnouncementCreation*

3.4.11 Implementação de *timers*

A plataforma *OutSystems* [1] oferece um serviço, representado na figura 3.28, que permite executar lógica de forma assíncrona, isto é, executa uma dada ação (*task*) de forma periódica.

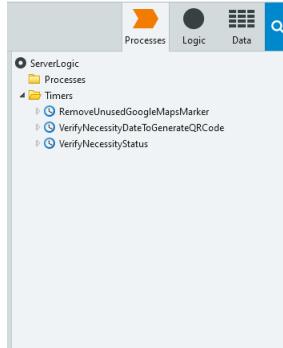


Figura 3.28: Timers implementados

Deste modo, no contexto da *Collaboration Platform* foram desenvolvidos três *timers* com propósitos distintos, apresentados na figura 3.28.

O *timer RemoveUnusedGoogleMapMarker*, representado na figura 3.29, corre semanalmente com o propósito de remover todos os marcadores do *Google Maps* (textit{markers}) que tenham o contador de utilização a 0.

Este *timer* é necessário devido ao facto dos *markers* serem partilhados entre necessidades, ou seja, necessidades com a mesma localização partilham o mesmo marcador. Esta otimização permite poupar espaço na base de dados.

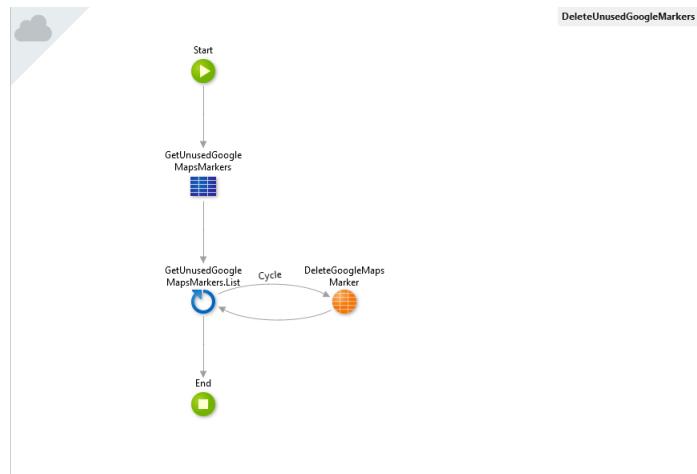


Figura 3.29: Timer *RemoveUnusedGoogleMapMarker*

O *timer VerifyNecessityDateToGenerateQRCode*, representado na figura 3.30, corre diariamente para encontrar as necessidades que irão decorrer no presente dia e que necessitem de QR Code para verificação de presença dos participantes. Quando encontra, procede à geração do QR Code e envia notificações aos participantes da necessidade para que os mesmo o possam apresentar.

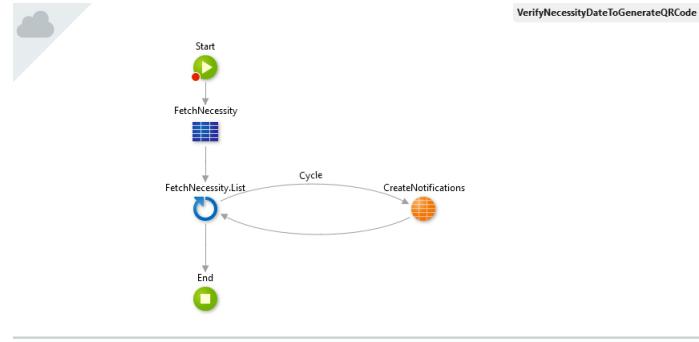


Figura 3.30: *Timer VerifyNecessityDateToGenerateQRCode*

O timer `VerifyNecessityStatus`, representado na figura 3.31, corre diariamente para encontrar as necessidades que tenham ocorrido no dia anterior e é imperativo mudar o seu estado de *active* para *closed*.

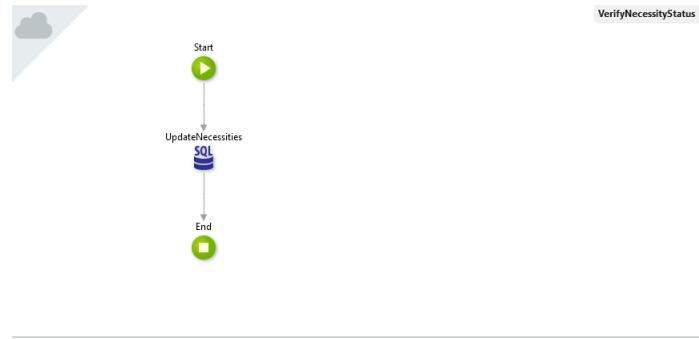


Figura 3.31: *Timer VerifyNecessityStatus*

Capítulo 4

Conclusão

Atualmente as empresas, para além de gerirem os seus principais processos de negócio, têm também uma dinâmica significativa de atividades internas para as ajudar no seu crescimento e competitividade. Nem sempre é fácil manter uma boa dinâmica das atividades internas de uma empresa porém, com uma plataforma que permite a gestão das mesmas, torna-se mais simples. É esse o objetivo do projeto *Collaboration Platform*, no sentido em que promove uma boa gestão destas necessidades e permite que todos os funcionários exponham as suas vontades.

Numa fase inicial do desenvolvimento do projeto, devido à pesquisa e aprendizagem de algumas tecnologias novas, o desenvolvimento ocorreu de forma mais lenta. A maior dificuldade encontrada foi a aprendizagem de como desenvolver aplicações numa plataforma nova e desconhecida, com uma utilização crescente no mercado, sendo também diferente do habitual caracterizando-se por um ambiente gráfico de desenvolvimento *low-code*. A gestão do progresso do desenvolvimento do projeto foi realizada utilizando uma metodologia de trabalho ágil, o *Scrum* [5], sendo um aspeto positivo a ser realçado, pois contribuiu para o processo de aprendizagem, uma vez que se trata de uma metodologia muito utilizada no mercado atual.

Após ultrapassados esses obstáculos relacionados com o primeiro contato, o ritmo de trabalho aumentou de forma considerável e foi possível implementar as funcionalidades de forma mais rápida. Esta boa adaptação ao método de trabalho permitiu que os requisitos propostos fossem cumpridos com sucesso e ainda houvesse tempo para ambicionar atingir outros de caráter opcional, nomeadamente a confirmação de presenças através de *QR Code*.

Ao longo do desenvolvimento da *Collaboration Platform* o grupo deparou-se com diversos obstáculos e decisões de implementação que permitiam boas discussões acerca de como deveríamos proceder. Um dos principais obstáculos foi o facto da aplicação ser de natureza *reactive*, isto é, passível de ser utilizada tanto em dispositivos móveis (*tablet* e *telemóvel*) como em *desktop* e o que faz com que o seu desenvolvimento tenha um cuidado especial com as diferenças que cada dispositivo apresenta.

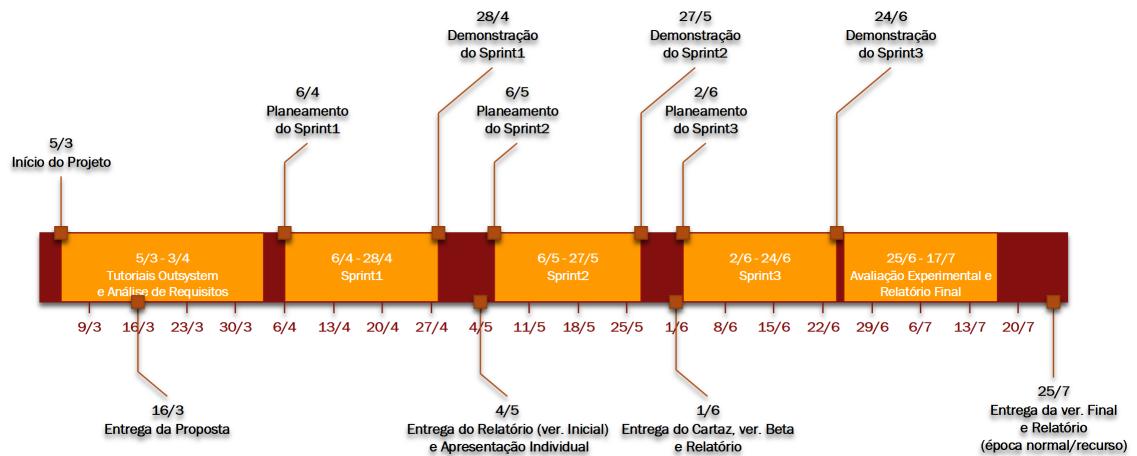


Figura 4.1: Timeline

Bibliografia

- [1] outsystems, "outsystems", [Online]. Available: <http://www.outsystems.com>. [Acedido em: 04 2020]
- [2] outsystems, "outsystems", [Online]. Available: <https://www.outsystems.com/forums/discussion/12775/how-to-bootstrap-users-with-roles-and-groups/>. [Acedido em: 04 2020]
- [3] outsystems, "outsystems", [Online]. Available: <https://www.outsystems.com/forge/component-overview/6891/camera-reactive>. [Acedido em: 04 2020]
- [4] basecamp, "basecamp", [Online]. Available: <https://basecamp.com/>. [Acedido em: 03 2020]
- [5] scrumportugal, "scrumportugal", [Online]. Available: <http://www.scrumportugal.pt/scrum/>. [Acedido em: 03 2020]
- [6] Namrata AroraKeith, "Ordering activities and notifications within a collaboration platform", [Online]. Available: <https://patents.google.com/patent/US9224173B2/>. [Acedido em: 05 2020]
- [7] google maps platform, "google maps api", [Online]. Available: <https://developers.google.com/maps/documentation/javascript/reference>. [Acedido em: 05 2020]
- [8] google maps platform, "placesAPI", [Online]. Available: <https://developers.google.com/places/web-service/intro>. [Acedido em: 05 2020]
- [9] full calendar, "full calendar", [Online]. Available: <https://fullcalendar.io/docs>. [Acedido em: 05 2020]
- [10] The constraints object is a MediaStreamConstraints object that specifies the types of media to request and the requirements of each media type.

A.1 Modelo Entidade-Associação da aplicação

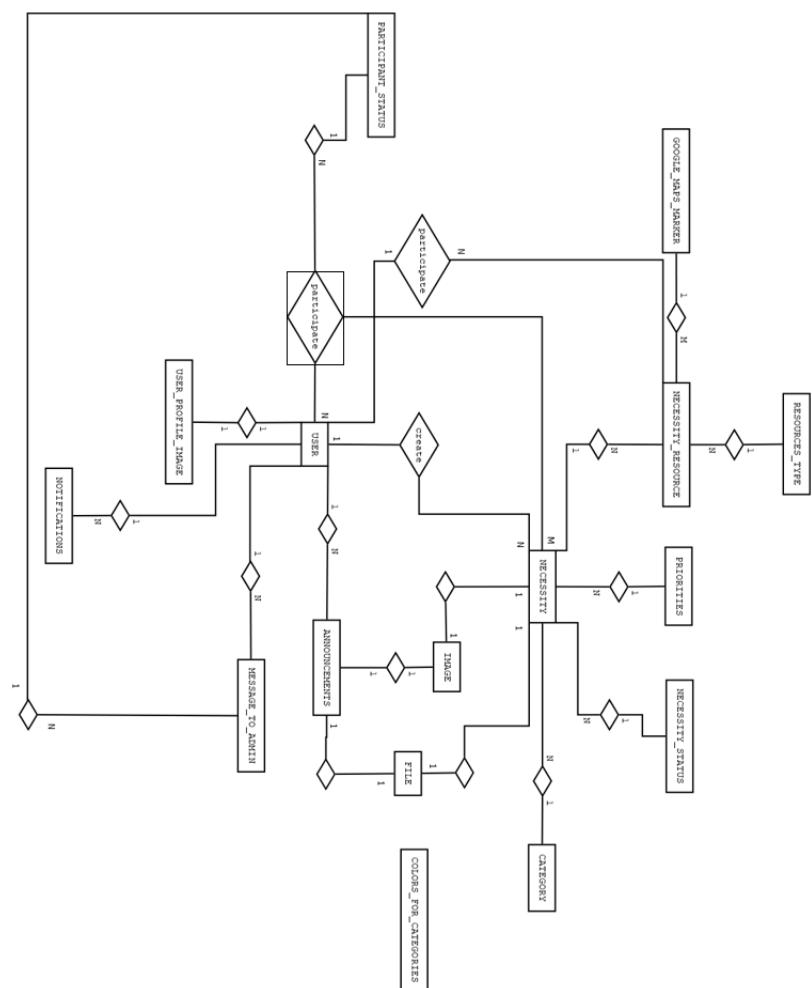


Figura 4.2: Modelo EA