

DI/FCT/UNL

APDC Projeto 2017/2018

**Documento de referência do projeto**  
**Requisitos iniciais e especificações gerais**

**Membros da equipa:**

Nº 46093 Marco Francisco

Nº 47406, Ana Sousa

Nº 47982, Francisco António

Nº 46288, Daniel Machado

Nº 45576, Rui Genovevo

Versão do documento	final	09/07/2018	
Revisto por:			

1. Identificação da Equipa, Nome do produto e Site de Divulgação

<b>Identificação da Equipa</b>	Software Elementalists	<b>LOGO da equipa</b>	 Software Elementalists
<b>Nome do Produto</b>	Agni	<b>LOGO DO PRODUTO</b>	
<b>Site de Divulgação (caso exista)</b>	—————Coming Soon—————		

## 2. Resumo das características gerais do produto a desenvolver

### 2.1 Resumo das características gerais do produto e solução a desenvolver

Tendo como objetivo desenvolver um site web com uma componente móvel, a nossa App “Agni” proporciona aos utilizadores a possibilidade de ajudar na prevenção dos incêndios.

Após o registo e login na app, os utilizadores têm a oportunidade de registar uma zona onde passaram que não esteja limpa, através do GPS ou sinalização no mapa, podendo adicionar ao registo uma foto ou outro conteúdo multimédia para complementar. Podem ainda ver no mapa registos de outros utilizadores, bem como o estado de cada área a intervir.

Serão também visíveis dados estatísticos do desempenho das entidades e de celeridade na resolução dos problemas.

Numa fase mais avançada, irá existir uma secção de *news* na página inicial do nosso site/app com acontecimentos relevantes acerca dos incêndios. Para além disso, de forma a incentivar os utilizadores, estes terão associados ao seu perfil um certo número de pontos, que aumenta quando os mesmos ajudam à prevenção de incêndios (reporte de zonas não limpas, comentários relevantes, ...).

Achamos também importante que, de forma a sensibilizar os *users*, estes recebessem uma notificação (ou simplesmente um pop-up na app) quando uma zona próxima se encontra em risco e quando uma zona não tratada provocou incêndio.

Para finalizar, pensamos ainda em avaliar dados do IPMA, tornando as mais relevantes visíveis aos utilizadores, também como objetivo de os alertar de uma certa zona.

### 2.2 Funcionalidade do produto

#### 2.2.1 Funcionalidade básica do produto (WILL HAVE)

// incluir como lista de funcionalidades previstas que vão mesmo ser oferecidas aos utilizadores (indicando o que vai estar em FrontEnd WEB e em Móvel).

Ter em conta que devem estar os requisitos indicados como obrigatórios no enunciado inicial, para além de outra funcionalidade que vão ser incorporadas.

- Registo de utilizadores com distinção de categorias:
  - Civil ou Empresa
  - Empresas de Limpeza e Entidades Governamentais (Limpeza, Bombeiros, GNR, ...)
- Secção com várias operações a realizar
  - Limpeza de áreas propícias a risco de incêndio (matas, ...)
  - Cortagem de árvores
- Secção com mapas legendados com indicação dos locais a intervir, e respetivas datas de registos, bem como ficheiros multimédia para ajuda da equipa a seleccionar e indicação de urgência de resolução
- Possibilidade de escolha ao utilizador em relação à visibilidade do seu registo, podendo este ser privado
- Reportar TROLLS e SPAM

- Possibilidade de ver o estado das áreas a intervir numa região, assim como as datas previsíveis de realização da limpeza
- Dados estatísticos do desempenho das entidades e de celeridade na resolução dos problemas
- Aplicação móvel – possibilidade de notificação rápida, componente de georreferenciação automática, adição simples de conteúdo multimédia

### **2.2.2 Funcionalidade complementar do produto (COULD HAVE)**

// incluir lista de funcionalidades previstas que podem vir a ser oferecidas aos utilizadores (FrontEnd WEB e Móvel).

Verificar que funcionalidades adicionais vão ser previstas (ver as exemplificativas do enunciado ou outras que sejam entendidas como sendo relevantes de virem a ser consideradas como funcionalidade complementar às indicadas em 2.2.1

- Portal web com possibilidade de delimitar uma região a limpar no mapa, com respetiva interface (referida em cima)
- Obtenção de coordenadas GPS e upload de fotos para auxílio de documentação
- Ver zonas já limpas ou com limpeza agendada e estatísticas fundamentais

### **2.3 Funcionalidades distintivas do produto (HIGHLIGHTS)**

// incluir lista de funcionalidades destacáveis que estão planeadas serem oferecidas e que se argumente como sendo diferenciadoras ou distintivas na solução a desenvolver (FrontEnd WEB e Móvel).

- Secção com news, com acontecimentos relevantes
- Registo de zonas não limpas que provocaram incêndio, e sensibilização às pessoas de zonas próximas para tal acontecimento
- Um temporizador em como o risco de incêndio aumentou (pop-up)
- Ter como referência dados do IPMA como comparação de áreas de risco

### 3. Suporte de gestão do produto (Back Office)

#### 3.1 Funções de suporte operacional de gestão do produto

// incluir lista de funcionalidade para suportar a atividade de *backoffice* para operação e gestão do produto (*product/system operation and management*) quando o produto for operacionalizado em ambiente de produção

Categorizar o que vão considerar MUST HAVE. O que vão considerar como COULD HAVE e o que pensar ser HIGHLIGHTS

##### MUST HAVE

- Registo e gestão de utilizadores com distinção de categorias
- Interação multicanal
- Infraestrutura Cloud-Enable PAAS Solution

##### COULD HAVE:

- Categorização de tipos de operações a realizar e associação com entidades
- Mecanismos para moderação de fóruns
- Controlo de visibilidade do registo do problema
- Dados estatísticos
- Avaliação autónoma do desempenho das equipas de limpeza

#### 3.2 Funções de administração sistema/aplicação

// incluir lista de funcionalidade para suportar a atividade de *backoffice* para operação e gestão do produto (*product/system operation and management*) quando o produto for operacionalizado em ambiente de produção

Categorizar o que vão considerar MUST HAVE. O que vão considerar como COULD HAVE e o que pensar ser HIGHLIGHTS

##### MUST HAVE:

- Gestão da plataforma, incluindo nomeadamente controlo e estatísticas de acesso
- Administração e moderação de fóruns
- Gestão de diversas entidades e associação com zonas
- Controlo completo da plataforma
- Obter panorama de zonas por limpar, mudar seu estado com atualização de datas e juntar comentário. Registo do utilizador que efetuou a alteração.

##### COULD HAVE:

- Definição de urgência e agendamento em canais privados ou públicos
- Delimitação de zonas geográficas de atuação e de tipo de espaço
- Gestão de diversas entidades e associação com zonas

##### HIGHLIGHTS:

- Avisar por mail quando o problema for resolvido

### 4. Funcionalidade de suporte de equipas de intervenção (Backend operacional)

// incluir lista de funcionalidade prevista para suportar a atividade de *backend* de equipas operacionais ou de intervenção que estejam envolvidas em ações de intervenção, ex., limpeza de matas, manutenção de segurança de vias ou caminhos,

adição ou reporte de informação verificada “in loco”, remoção de materiais perigosos ou reporte/notificação/disseminação de intervenções de prevenção junto de pessoa/povoações/habitações, como intervenções de resposta a situações reportadas pelos utilizadores das aplicações e que permitam o suporte operacional de interv(*operational teams / operational persons* ) quando o produto for operacionalizado em ambiente de produção

Categorizar o que vão considerar MUST HAVE. O que vão considerar como COULD HAVE e o que pensar ser HIGHLIGHTS

Identificar funções previstas suportadas em interação WEB (Browser) e/ou Móvel

**MUST HAVE:**

- Votação e avaliação do desempenho das equipas de limpeza e sugestões
- Resposta a pedidos por parte de empresas, privados ou de entidades oficiais

**COULD HAVE:**

- Estatísticas completas a nível de utilizador, entidade, zona geográfica, etc.

**HIGHLIGHTS:**

- Inclusão de previsão de risco de incêndio a partir de sites oficiais
- Sistema de alerta passivo (alerta de locais e viajantes/turistas)
- Introdução de tutoriais de ajuda a limpeza por parte de utilizadores

## 5. Interface e Mockups

// Por agora deixar esta secção e todas as subsecções intencionalmente vazias, para evolução do documento em versões posteriores á versão 1.0)

### 5.1 Especificação da Interface do site (aplicação site/WEB - Layouts)

(até 2 páginas e ANEXOS: a incluir posteriormente)


A aplicação web foi desenvolvida em React. De forma a criar um layout moderno e semelhante de página para página utilizamos como ferramenta complementar o Material-UI (componentes de design de interface de utilizador da Google construídas a partir de React).

As cores predominantes da aplicação são o cinzento, branco e azul – esta paleta foi escolhida pois achamos que representava bem uma app moderna, e o azul, que é a cor do nosso logo da app, representa a limpeza e “um céu limpo” sem o fumo e poeiras predominantes nos incêndios.

**Itens MUST/SHOULD HAVE por implementar** - Quanto aos itens que ficaram por implementar, estes foram relativos à parte das estatísticas, a delimitação no mapa da zona a reportar, e ainda algumas funcionalidades de Back-Office. Há que ter em conta que apenas um membro do nosso grupo que nunca tinha trabalhado com React e até com pouca experiência em Java Script é que foi responsável pela parte Web, pelo que era de algum modo difícil terminar tudo dentro dos prazos.

Na página inicial o utilizador verá uma barra horizontal no cimo da app, onde estão presentes dois botões que correspondem ao:



- **login** realizado com username do utilizador e password
- **registo do utilizador** na aplicação com sugestão de localidades do google conforme é escrita a localidade




Após ser feito login aparece no lugar dos botões login e registo de utilizador (que desaparecem) um ícone com uma imagem que é um menu onde o utilizador poderá **ver o seu perfil** e **fazer logout** na aplicação. 

Se o utilizador com sessão iniciada for um trabalhador aparece ainda um ícone de notificações com a listagem de ocorrências por tratar que ainda não foram atribuídas a nenhum trabalhador.

Na barra lateral da página estão presentes uns ícones com um título que ao serem clicados redirecionam para o link onde se encontra a página pretendida. Os utilizadores poderão clicar no botão mais à esquerda da barra de cima para comprimir ou expandir a barra lateral (fazendo desaparecer o título correspondente a cada ícone, ou mostrá-lo, respetivamente).

A listagem das secções presentes na barra lateral é a seguinte:

- **página inicial** onde é apresentado o logotipo da aplicação e botões semelhantes aos da barra lateral que tal como esta redirecionam para a página pretendida
- **notícias** onde serão apresentadas as notícias mais populares relativamente ao tema incêndios
-  **proteja-se** com informações importantes e dicas para a população do programa “Aldeia Segura e Pessoas Seguras” que tem como principais objetivos “criar nas freguesias a função de Oficial de Segurança, que tem como missão transmitir avisos à população, organizar a evacuação do aglomerado em caso de necessidade e fazer ações de sensibilização junto da população” – tal como está presente no site de divulgação: <http://www.inem.pt/2018/06/04/programa-aldeia-segura-e-pessoas-seguras/>
- **perfil** que contém:
  - a foto do utilizador, ou um avatar default caso este não tenha
  - as informações básicas do user que está com sessão iniciada (nome de utilizador, nome completo, tipo, email, ...)
  - lista de reportes privados do utilizador
  - lista de ocorrências apoiadas pelo utilizador
  - tem ainda a opção de **editar o perfil e mudar a password**
- **registar ocorrência** que tal como o nome indica permite ao utilizador registar um problema que encontra num certo local – função principal da nossa aplicação;
  - se o registo for bem sucedido, então é retornada uma mensagem de agradecimento ao utilizador por “tornar a terra um sitio melhor”;
  - o mapa correspondente à localização do problema começa nas coordenadas do gps do utilizador, podendo este pesquisar outra localização caso o problema não se encontre no local atual do utilizador,  ou ainda pode clicar no ícone que coloca o mapa centrado na localidade do utilizador, caso este a tenha.

- **tabela de operações** que contem a listagem de ocorrências publicas; ao clicar em “informações” poderemos ver mais detalhadamente as informações e estado da ocorrência
  -  **informação de uma ocorrência** – onde estão presentes todas as informações acerca de uma ocorrência, bem como os gostos e comentários da mesma
- **mapa** que contém um mapa com:
  - markers de diferentes cores conforme o grau de urgência e/ou se o problema foi tratado ou não,  e ícones conforme o tipo de problema e quando clicados é possível ver a informação de cada reporte
  - filtros com a possibilidade de ver markers com os graus de urgência escolhidos pelo utilizador
  -  legenda do mapa
- **estatísticas** que irá conter as estatísticas de desempenho de entidades bem como a celeridade na resolução de problemas
- **sobre nós** que contém informações sobre os vários elementos da equipa “Software Elementalists”, equipa fundadora de AGNI

Existem ainda outras secções que apenas são visíveis por gestores do Back-Office, que está dividido em 3 grupos: administradores (têm acesso a todas as secções), moderadores e trabalhadores.



Secções dos administradores:

- **Lista de utilizadores** – contém uma lista de todos os utilizadores registados na aplicação, e o administrador pode editar o perfil de cada um deles
- **Gerir permissões** – onde é possível decidir se um dado utilizador pode ser moderador ou trabalhador, e adicioná-los



Secções dos moderadores:

- **Gerir permissões** – onde é possível decidir se um dado utilizador pode ser trabalhador, e adicioná-los
- **Operações** – nesta secção, para além do que é possível fazer como um utilizador não categorizado, é ainda possível aceitar ou recusar uma dada ocorrência, ou concluí-la, caso já tenha sido tratada e aprovado o trabalho.



Secções dos trabalhadores:

- **Por tratar** – contém duas tabelas, uma com a listagem de ocorrências aceites pelo trabalhador e prontas a tratar; outra com a listagem de ocorrências aceites por um administrador e/ou moderador e ainda não associadas a nenhum trabalhador.

O corpo da página vai portanto variando de secção para secção, e são aí apresentadas as informações que o utilizador poderá ver e/ou os registos ou edições que fará na aplicação.

Note-se que de forma a interagir com o servidor, através de pedidos, são feitos XMLHttpRequests.

**Dificuldades:** Ao longo do trabalho fomos-nos deparando com várias dificuldades na parte da web, no início pelo facto de este funcionar totalmente de forma assíncrona, algo que não estávamos habituados com java, e ainda na parte dos mapas em React visto que foi difícil conseguir encontrar uma ferramenta fácil de utilizar para todas as secções.

Ver anexos 1 a 6, 9 e 10

## 5.2 Diagramas de transição (aplicação site/WEB - Layouts)

(até 2 páginas e ANEXOS: a incluir posteriormente)

Tal como já foi referido no ponto 5.1 no que tem em conta a transição de páginas (ou neste caso do corpo da aplicação) esta é feita através do evento “click” nos ícones presentes na barra lateral da aplicação.

Esta pode ainda ser feita da mesma forma ao clicar nos botões que aparecem no corpo da página inicial.

No que tem em conta o resto das transições (como por exemplo ver as informações detalhadas de cada operação na tabela de operações, o “editar perfil” e “mudar password”, mensagem de sucesso após um registo de um problema, ...) que permitem a transição de uma página para outra com o conteúdo idêntico ao da sua secção inicial, são feitas através de botões (com ícones ou não) ao serem clicados.

Ver anexos 7 e 8

## 5.3 Especificação da Interface do site (App Móvel)

(até 2 páginas e ANEXOS: a incluir posteriormente)

A aplicação móvel foi desenvolvida em Java utilizando o Android Studio.

Todas as funcionalidades propostas foram implementadas com exceção da visualização de dados estatísticos provenientes do uso da aplicação, e ainda a delimitação de uma área no reporte de um problema (tanto web, como android). Esta parte não foi implementada por falta de tempo. Focámo-nos muito no design, apresentação e bom funcionamento de outras partes da aplicação e acabámos por não dedicar o tempo necessário a isto. É de salientar também que durante o último mês alguns elementos do grupo estiveram em época de exames e foi difícil aplicar a mão de trabalho necessária para implementar esta funcionalidade. Face a isto, decidiu-se investir todo o tempo restante em melhorar as funcionalidades a meio gás e/ou melhorar funcionalidade já estabelecidas.

Contudo, foram implementadas várias funcionalidades base da aplicação que consideramos de importância extrema, sendo elas:

**Login/Logout e Registo de utilizador:** É possível o utilizador ter uma conta pessoal que terá funcionalidades diferentes consoante o seu tipo (Civil ou



Trabalhador). É possível também personalizar este perfil com uma foto e uma variedade de informações pessoais. Não diretamente associado à conta de utilizador (pois funciona só localmente no Android), é possível ao utilizador ter uma lista de localizações favoritas, sendo-lhe mais fácil, rapidamente, verificar ocorrências perto destas.

**Mapa com ocorrências:** Um mapa em que é possível ver as ocorrências registadas (públicas). É possível filtrar por tipo de ocorrência e grau de severidade. É aqui também que se encontra a lista de localizações favoritas. Ao clicar numa ocorrência, é possível ver em mais detalhe as suas informações.

**Menu Detalhe de Ocorrência:** Neste menu podemos ver todas as informações sobre uma ocorrência: Título, Descrição, Foto (se tiver), estado, grau de severidade, tipo de ocorrência e localização no mapa. Há também uma checkbox para ser notificado quando a ocorrência for resolvida, mas esta funcionalidade não foi implementada, como dito mais acima. É possível também editar informações sobre a ocorrência (título, descrição, tipo e visibilidade). Acrescentar comentários também é possível assim como por "gosto" na ocorrência. Se o utilizador for do tipo Trabalhador é-lhe permitido aceitar e concluir uma ocorrência.

**Registar Ocorrência:** É possível registar uma ocorrência, com ou sem foto. No caso de querer acrescentar uma foto, é possível usar a câmara do dispositivo ou procurar uma foto guardada localmente. É indicado o título, descrição, grau de severidade e tipo de ocorrência. Também é possível indicar se esta é pública ou privada. A localização da ocorrência é indicada num mapa em que a localização inicial do marcador é na posição do utilizador.

**Help:** Pequeno menu informativo para ajudar o utilizador a melhor se precaver de um incêndio ou, no caso de este já estar em curso, como reagir em segurança.

**Listar ocorrências:** Lista as ocorrências públicas com feedback visual do seu grau de severidade (de verde até vermelho, da menos grave a mais grave). Aqui tivemos alguns problemas em ir buscar o número de likes em cada ocorrência. É apresentado em formato de lista as ocorrências mostrando apenas o seu título e número de likes. Clicando numa, vamos para o menu de detalhes de ocorrência, que foi acima descrito.

**Perfil:** Um resumo do utilizador tal como se encontra no nosso sistema. Para além de consultar as suas informações, é possível editar as mesmas. Nesta secção é possível ao utilizador alterar a sua palavra-passe. Ao clicar nos ícones de favorito e ocorrência o utilizador pode listar as ocorrências em que deu like e as ocorrências que foram registadas por si.

**Ocorrências aceites (apenas disponível para Trabalhador):** Aqui o Trabalhador pode ver todas as ocorrências que foram aceites por si e que ainda não foram concluídas.

A qualquer momento é possível fazer logout clicando no canto superior direito e escolhendo "**Terminar sessão**". Existe também um botão disponível em todo o momento durante a aplicação em que automaticamente abre a câmara e permite rapidamente tirar uma foto e prepara-la para ser anexada ao registo de uma ocorrência.

No menu inicial da aplicação é possível ler alguns headlines de notícias relacionadas com fogos/incêndios florestais. Isto recorre ao uso de uma api chamada NewsAPI. Infelizmente, ao delimitar a pesquisa de notícias ao nosso país,

algumas notícias não são visualizadas, por isso as notícias que aparecem são notícias do mundo inteiro que dão match com palavras "fogo" e/ou "incêndio".

Ver anexos 11,12 e 14 a 24

## **5.4 modelo de navegação e diagramas de transição (App Móvel)**

(até 2 páginas e ANEXOS: a incluir posteriormente)

Na nossa aplicação depois de ser efetuado o login o utilizador é direcionado para a página inicial. Em todas as páginas da nossa aplicação o utilizador pode clicar no botão no canto superior esquerdo que torna visível a barra lateral. Através da barra lateral é possível realizar todas as operações.

Ver anexo 13

## **5.5 Outros aspetos de suporte de interação dos utilizadores**

(até 2 páginas e ANEXOS: a incluir posteriormente)

Na aplicação web é possível na página inicial aceder aos tópicos que também se encontram na barra lateral.

No android existe um botão no canto inferior direito que irá ser usado como atalho para registar uma ocorrência (neste momento tem um símbolo de uma mensagem que irá ser substituído por um “mais” ou algo do género).

## **6. Arquitetura da solução**

// Por agora deixar esta secção intencionalmente vazia, para evolução do documento em versões posteriores á versão 1.0)

O código do servidor/website está constituído em duas partes, uma primeira parte que contém a parte de front-end (web) e outra com o back-end (rest api), esta separação acontece devido à página web estar a ser construída como uma SPA (single page application), ou seja, todo o layout é enviado no primeiro loading da página e a partir daí todas as alterações na página são dinamizadas a partir de javascript e pedidos rest à api do servidor (ou seja o back-end).

O back-end tem vários recursos do qual podemos fazer uso para obter as informações desejadas, sendo estas:

- register (“/rest/register”) : recurso no qual enviando a informação de uma pessoa em json se pode fazer registo de um novo utilizador que fica guardado na base de dados sendo este do tipo “User”, depois, caso o campo “uploadPhoto” venha a true, recebe-se um id do tipo long que pode ser usado para fazer upload de uma imagem de perfil, caso contrario não se recebe nada; tanto o username como o email tem que ser únicos:

Recebe:

```
{
    "username": String,
    "password": String,
    "name": String,
    "email": String,
    "role": String,
```

```

        "district": String,
        "county": String,
        "locality": String,
        "uploadPhoto": boolean,
        "entity": String
    }

```

- login ("/rest/login") : recurso no qual enviando o username e password de um utilizador em json se recebe um token em json para autenticação noutros recursos do servidor no qual este seja necessário, este token fica guardado na base de dados sendo do tipo "UserToken" associado a um utilizador do tipo "User":

Recebe:

```

{
    "username": String,
    "password": String
}

```

Envia:

```

{
    "userID": long,
    "username": String,
    "tokenID": String,
    "creationData": long,
    "expirationData": long
}

```

- logout ("/rest/logout") : recurso no qual enviando as informações de um token em json este se torna inutilizável em futuras autenticações apagando a entidade correspondente de "UserToken" na base de dados:

Recebe:

```

{
    "userID": long,
    "username": String,
    "tokenID": String,
    "creationData": long,
    "expirationData": long
}

```

- profile ("/rest/profile") : recurso no qual enviando as informações de um token e o username ou userID de um utilizador em json se recebe as informações de perfil de utilizador com o mesmo username, para isto o utilizador associado ao token tem de ter permissões para o fazer:

Recebe:

```

{
    "userID": long,
    "username": String,
    "token":
    {
        "userID": long,
        "username": String,

```

```

        "tokenID": String,
        "creationData": long,
        "expirationData": long
    }
}

```

- profile edit ("/rest/profile/edit") : recurso no qual enviando as informações de um token e o userID de um utilizador podemos enviar informações como no registo para substituir as antigas, depois, caso o campo "uploadPhoto" venha a true, recebe-se um id do tipo long que pode ser usado para fazer upload de uma nova foto de perfil, caso contrario não se recebe nada; no caso de tentar mudar a password tem de fornecer a antiga.

Recebe:

```

{
    "name": String,
    "token":
    {
        "userID": long,
        "username": String,
        "tokenID": String,
        "creationData": long,
        "expirationData": long
    },
    "userID": long,
    "newUsername": String,
    "password": String,
    "newPassword": String,
    "email": String,
    "district": String,
    "county": String,
    "locality": String,
    "uploadPhoto": boolean
}

```

- profile username ("/rest/profile/username") : recurso no qual enviando o userID de um utilizador podemos enviar o username atual do utilizador.

Recebe:

```

{
    "userID": long,
}

```

- occurrence register ("/rest/occurrence/register") : recurso no qual enviando as informações de um token e dados de uma ocorrência se cria uma nova ocorrência na base de dados do tipo "UserOccurrence" associado a um utilizador do tipo "User", depois, caso o campo "uploadMedia" venha a true recebe-se uma lista de long de tamanho "nUploads" que pode ser usado para upload de media relacionado com a ocorrência, sendo criado na base de dados novas entidades do tipo "UserOccurrenceMedia", caso contrario não se recebe nada:

Recebe:

```
{
  "token":
    {
      "userID": long,
      "username": String,
      "tokenID": String,
      "creationData": long,
      "expirationData": long
    },
  "title": String,
  "description": String,
  "type": String,
  "level": int,
  "visibility": boolean,
  "lat": float,
  "lon": float,
  "notificationOnResolve": boolean,
  "uploadMedia": boolean,
  "nUploads": int
}
```

- occurrence list ("rest/occurrence/list") : recurso que se fizer um GET envia 100 ocorrências públicas ou se fizer um POST pode-se fazer um filtro para certo utilizador, para ver ocorrências privadas e ou para limitar a uma certa zona sendo isto feito apenas se o utilizador do token enviado tiver esse tipo de acesso; em ambos os casos o servidor envia um cursor para acesso às próximas 100 ocorrências e uma lista de mapas em que cada mapa corresponde a uma ocorrência e contém os seus atributos:

Recebe:

```
{
  "token":
    {
      "userID": String,
      "username": String,
      "tokenID": String,
      "creationData": long,
      "expirationData": long
    },
  "username": String,
  "showPrivate": boolean,
  "cursor": String,
  "lat": float,
  "lon": float,
  "radius": int
}
```

- occurrence edit ("/rest/occurrence/edit") : recurso no qual enviando as informações de um token, userID de um utilizador e id do tipo long de uma ocorrência podemos enviar informações como no registo de uma ocorrência para substituir as antigas.

Recebe:

```
{
  "token":
    {
      "userID": long,
      "username": String,
      "tokenID": String,
      "creationData": long,
      "expirationData": long
    },
  "userID": long,
  "occurrenceID": long,
  "title": String,
  "description": String,
  "visibility": boolean,
  "notificationOnResolve": boolean,
  "uploadMedia": boolean,
  "nUploads": int
}
```

- occurrence delete ("/rest/occurrence/delete") : recurso no qual enviando as informações de um token, userID de um utilizador e id de uma ocorrência podemos apagar essa ocorrência da base de dados:

Recebe:

```
{
  "token":
    {
      "userID": long,
      "username": String,
      "tokenID": String,
      "creationData": long,
      "expirationData": long
    },
  "userID": long,
  "occurrenceID": long
}
```

- occurrence media ("/rest/occurrence/media") : recurso no qual enviando as informações de um token, de um userID, de uma occurrenceID e um id do tipo long de uma entidade "UserOccurrenceMedia" podemos receber o ficheiro media correspondente.

Recebe:

```
{
  "token":
```

```

    {
        "userID": long,
        "username": String,
        "tokenID": String,
        "creationData": long,
        "expirationData": long
    },
    "userID": long,
    "occurrenceID": long,
    "mediaID": long
}

```

- occurrence media resolve ("/rest/occurrence/resolve-media") : recurso no qual enviando as informações de um token, de um userID, de uma occurrenceID e um id do tipo long de uma entidade "UserResolvedOccurrenceMedia" podemos receber o ficheiro media correspondente.

Recebe:

```

{
    "token":
    {
        "userID": long,
        "username": String,
        "tokenID": String,
        "creationData": long,
        "expirationData": long
    },
    "userID": long,
    "occurrenceID": long,
    "mediaID": long
}

```

- upload ("/rest/upload/{id}") : recurso no qual enviando no endereço um certo id do tipo long fornecido noutro recurso (como por exemplo o registo de utilizador) faz upload da media recebida para o google cloud storage baseado nas informações da entidade com um id correspondente do tipo "FileUpload" guardada na base de dados.

- occurrence comment ("/rest/occurrence/comment") : recurso no qual enviando as informações de um token e de uma ocorrência se cria um comentário sobre a ocorrência.

Recebe:

```

{
    "token":
    {
        "userID": long,
        "username": String,
        "tokenID": String,
        "creationData": long,

```

```

        "expirationData": long
    },
    "userID": long,
    "occurrenceID": long,
    "comment": String
}

```

- occurrence comment list ("/rest/occurrence/comment/list") : recurso em que enviando informações de uma certa ocorrência se recebe 20 comentários dessa ocorrência; o servidor envia um cursor para acesso aos próximos 20 comentários:

Recebe:

```

{
    "token":
    {
        "userID": long,
        "username": String,
        "tokenID": String,
        "creationData": long,
        "expirationData": long
    },
    "userID": long,
    "occurrenceID": long,
    "cursor": String
}

```

- occurrence comment edit ("/rest/occurrence/comment/edit") : recurso no qual enviando as informações de um token, de uma ocorrência e o id de um comentário da ocorrência se edita esse mesmo comentário.

Recebe:

```

{
    "token":
    {
        "userID": long,
        "username": String,
        "tokenID": String,
        "creationData": long,
        "expirationData": long
    },
    "userID": long,
    "occurrenceID": long,
    "commentID": long,
    "comment": String
}

```

- occurrence comment delete ("/rest/occurrence/comment/delete") : recurso no qual enviando as informações de um token, de uma ocorrência e o id de um comentário da ocorrência se apaga esse mesmo comentário.

Recebe:

```

{

```



```

    "token":
      {
        "userID": long,
        "username": String,
        "tokenID": String,
        "creationData": long,
        "expirationData": long
      },
    "userID": long,
    "occurrenceID": long,
    "commentID": long
  }

```

- occurrence like ("/rest/occurrence/like") : recurso no qual enviando as informações de um token e de uma ocorrência o utilizador do token faz like ou dislike da ocorrência.

Recebe:

```

{
  "token":
    {
      "userID": long,
      "username": String,
      "tokenID": String,
      "creationData": long,
      "expirationData": long
    },
  "userID": long,
  "occurrenceID": long
}

```

- occurrence like list ("/rest/occurrence/like/list") : recurso em que enviando o userID de um utilizador se recebe 20 ocorrências no qual o utilizador fez like; o servidor envia um cursor para acesso às próximas 20 ocorrências:

Recebe:

```

{
  "token":
    {
      "userID": long,
      "username": String,
      "tokenID": String,
      "creationData": long,
      "expirationData": long
    },
  "userID": long,
  "cursor": String
}

```

- occurrence like count ("/rest/occurrence/like/count") : recurso no qual enviando as informações de uma ocorrência se recebe o numero de likes que essa mesma ocorrência tem.

Recebe:

```
{
    "userID": long,
    "occurrenceID": long
}
```

- occurrence like check ("/rest/occurrence/like/check") : recurso no qual enviando as informações de um token, de uma ocorrência e o userID de um utilizador se vê se o utilizador fez like na ocorrência.

Recebe:

```
{
    "token":
    {
        "userID": long,
        "username": String,
        "tokenID": String,
        "creationData": long,
        "expirationData": long
    },
    "userID": long,
    "userOccurrenceID": long,
    "occurrenceID": long
}
```

## 7. Características e requisitos arquiteturais

incluir quais as vantagens e requisitos arquiteturais que vão ser disponibilizadas pela solução a desenvolver e que sejam destacáveis como fatores que se consideram vir a ser relevantes na qualidade, efetividade, desempenho, disponibilidade e escalabilidade da solução (entre outras que se queiram destacar como previstas).

// incluir lista de requisitos arquiteturais, dedicando um parágrafo a descrever em que consistirá o suporte ou cobertura desse requisito (WHAT), porque o consideram importante (WHY) e como será suportado no sistema a desenvolver (HOW)

### 7.1 Modelo do sistema e arquitetura de software

#### Tier 1: Presentation

UI, Interação de Utilizadores e Suporte de Integração de Aplicação

#### Tier 2: Runtime, Logics

Lógica, componentes e Serviços (Reutilização)

#### Tier 3: Small Data, Large Objects

Gestão e Acesso aos Dados (Persistência)

## 7.2 Tecnologias de implementação e ferramentas utilizadas

- Android Studio
- Google App Engine, Cloud Datastore e Storage
- HTTPS, REST, Jersey
- Java, Json, Maven
- React: HTML, Javascript e CSS

## 7.3 Requisitos arquiteturais suportados

- Tolerância a falhas de componentes do servidor  
Multiple tier system structure e Cloud-Enabled Platform as a Service Solution
- Disponibilidade com Escalabilidade e Elasticidade  
Modelo de Execução (Google App Engine)
- Segurança  
HTTPS e User Authentication + Role Based Access Control Model

## 8. Outros aspetos de implementação, demonstração e avaliação experimental

// deixar por agora intencionalmente vazio

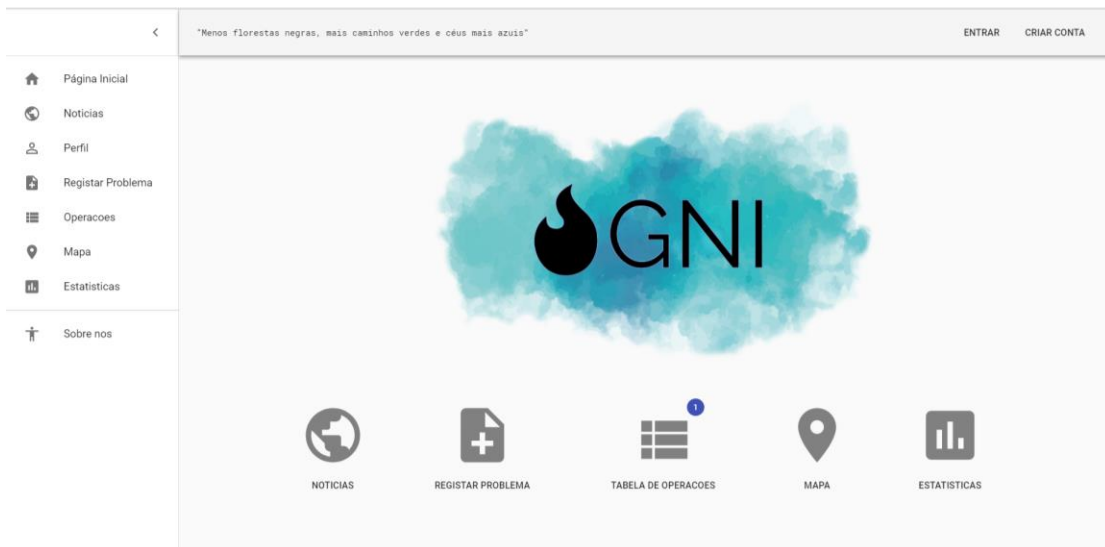
Temos uma secção de notícias onde serão apresentadas as notícias mais populares relativamente ao tema incêndios, ou então o top de ocorrências (ainda não decidimos completamente – isto é um HIGHLIGHT como já referido anteriormente).

Na criação de conta na web (posteriormente existirá também no android) os inputs são todos verificados obedecendo às seguintes condições:

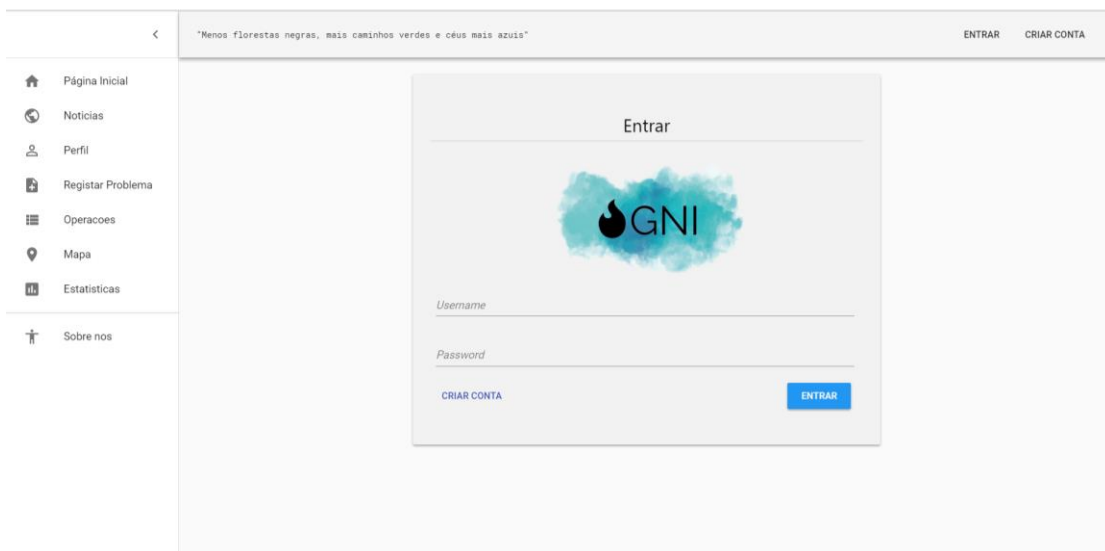
- Nome de utilizador só pode conter minúsculas, ser entre 5 e 25 caracteres e não pode conter certos caracteres especiais
- Nome completo tem que começar com letra maiúscula, ser entre 2 e 50 caracteres e não pode conter certos caracteres especiais
- Email tem que conter o carácter '@' e ser entre 9 a 30 caracteres e não pode conter certos caracteres especiais
- Password de forma a ser segura tem que ser entre 6 a 30 caracteres e conter pelo menos um símbolo e uma letra maiúsculas
- Confirmar Password tem que ser igual à password

## Anexos

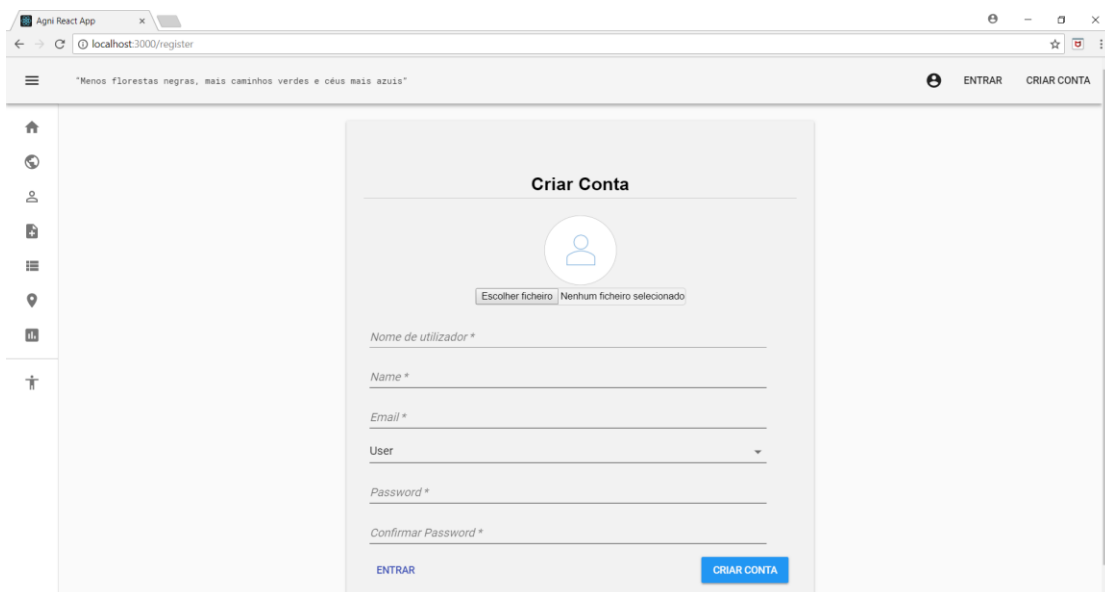
### Interface do site (aplicação web)



Anexo 1 – Página Inicial Web



Anexo 2 – Entrar (Login) Web



?

Escolher ficheiro

Nenhum ficheiro selecionado

Nome de utilizador \*

Name \*

Email \*

Utilizador

Localidade

carregal

Carregado Portugal

Carregal do Sal Portugal

Carregal do Sal Carregal do Sal, Portugal

Carrega Ligure Alexandria, Itália

Carregal Aveiro, Portugal

powered by Google

Password \*

Confirmar Password \*

## Anexo 3 – Criar Conta Web

<

\*Menos florestas negras, mais caminhos verdes e céus mais azuis\*

Página Inicial

Noticias

Perfil

Registar Problema

Operacoes

Mapa

Estatísticas

Sobre nos

EDITAR PERFIL

PERFIL

TERMINAR SESSÃO

margarida

anamsousa@live.com.pt

6

Reportes de Problemas

0

A apoiar

0

Comentarios

Ana Margarida

User

REPORTES

A APOIAR

AMIGOS

Lixo nas matas de Carregal do Sal

Tipo: Limpeza de Mato

Grau: 3

Data do registo: Jun 22, 2018 5:08:34 PM

## Anexo 4 – Perfil Web

## Anexo 5 – Registrar Problema Web

Descrição - descreva o estado genérico do problema

Limpeza necessária em Carregal do Sal

Tipo de espaço \*

☒ Público

☐ Privado

Tipo de Problema

Limpeza de Mato

Grau de urgência

3

Toda a gente pode ver este pro...

Visibilidade

Notificar quando estiver resolvido

+ REGISTRAR PROBLEMA

Tem a certeza que pretende registar o problema?

Poderá posteriormente editar este registo no seu perfil na secção reportes.

CANCELAR SIM

## Anexo 6 – Alerta registar problema Web

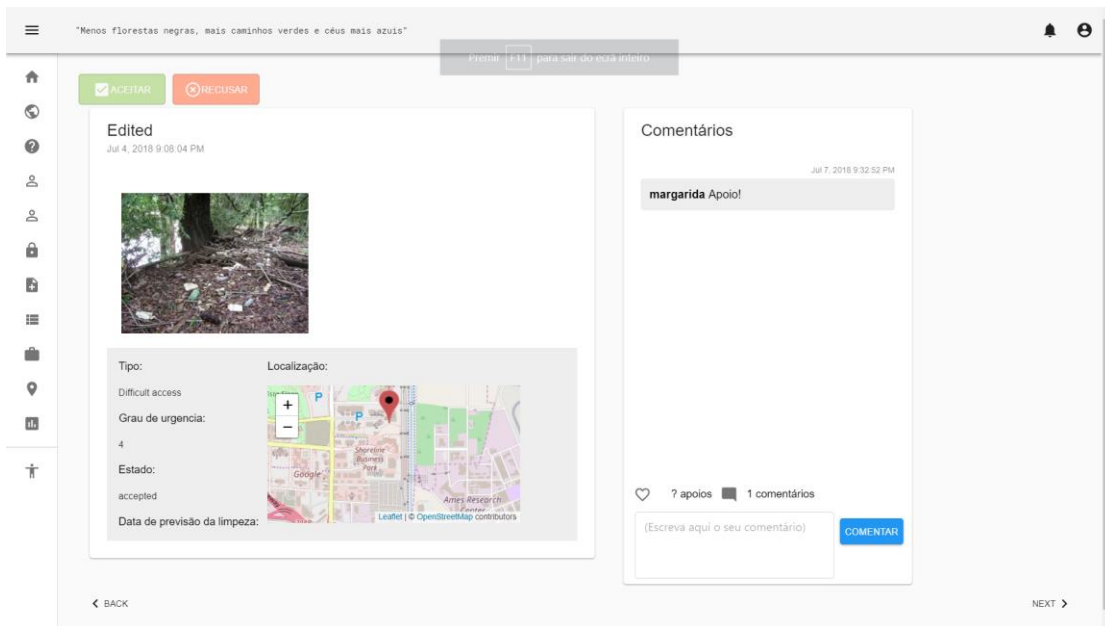
< "Menos florestas negras, mais caminhos verdes e céus mais azuis" ⓘ

**Tabela de Ocorrências** ↻

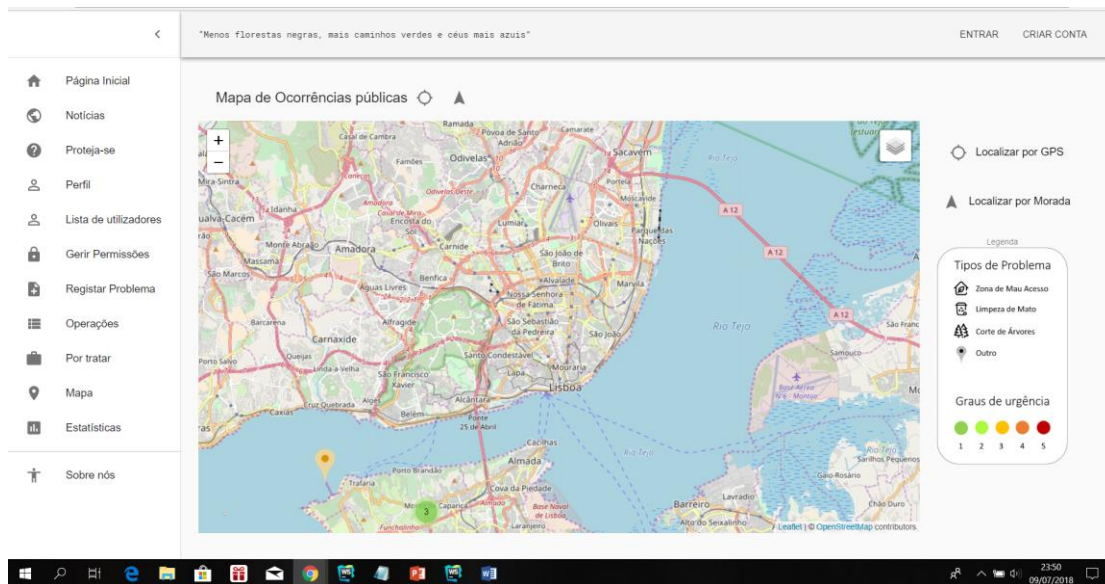
Informações	Nome	Tipo	Data	Estado	Grau de urgência	Visibilidade
📅	Lixo nas matas de Carregal do Sal	Limpeza de Mato		nao tratado	3	público
📅	Limpar florestas	Limpeza de Mato		nao tratado	5	público
📅	Limpeza nas matas	Limpeza de Mato		nao tratado	3	público
📅	Zona de mau acesso	Limpeza de Mato		nao tratado	3	público
📅	Pequena limpeza necessária	Limpeza de Mato		nao tratado	1	público

Rows per page: 5 1-5 of 8 < >

## Anexo 7 – Tabela de Ocorrências Web

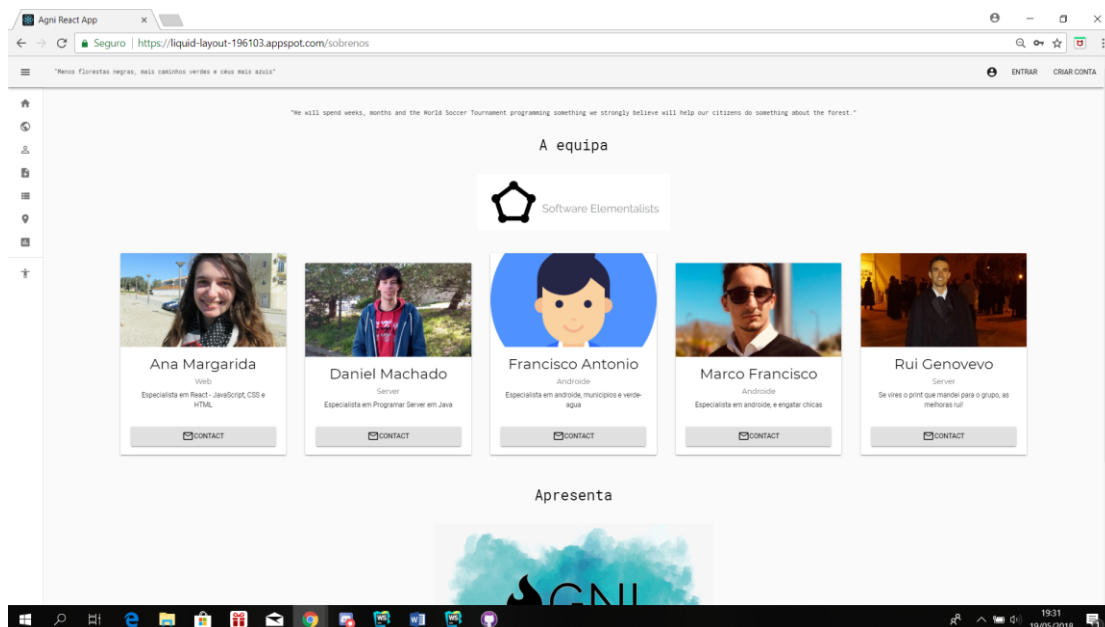
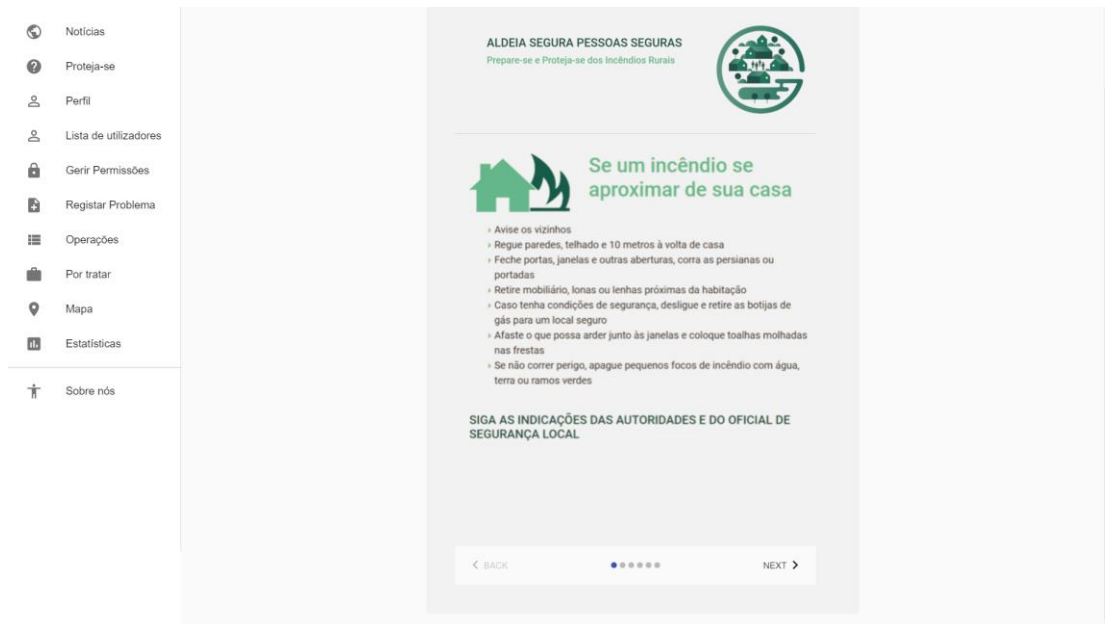


Anexo 8 – Informação de uma ocorrência Web



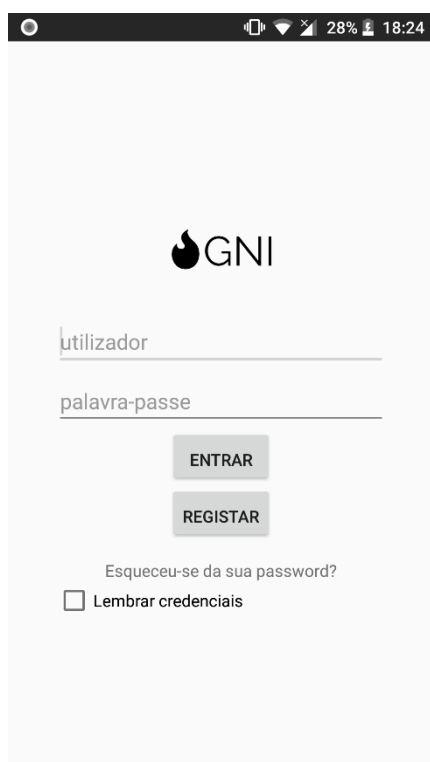
Anexo 9 – Mapa de Ocorrências Web





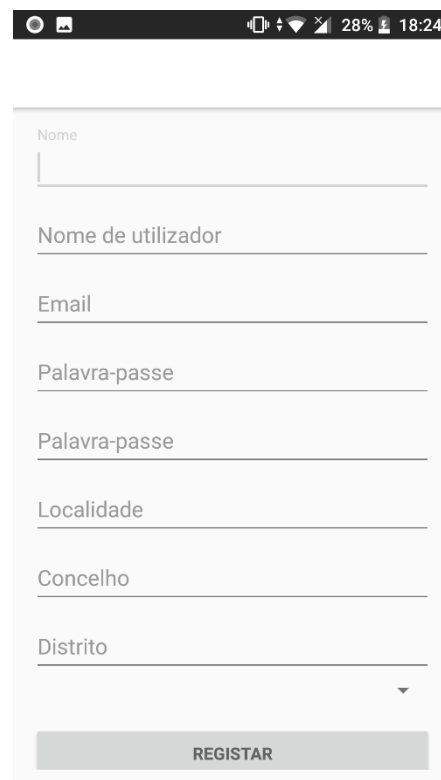
Anexo 10 – Sobre nós Web

## Interface do site (aplicação móvel)



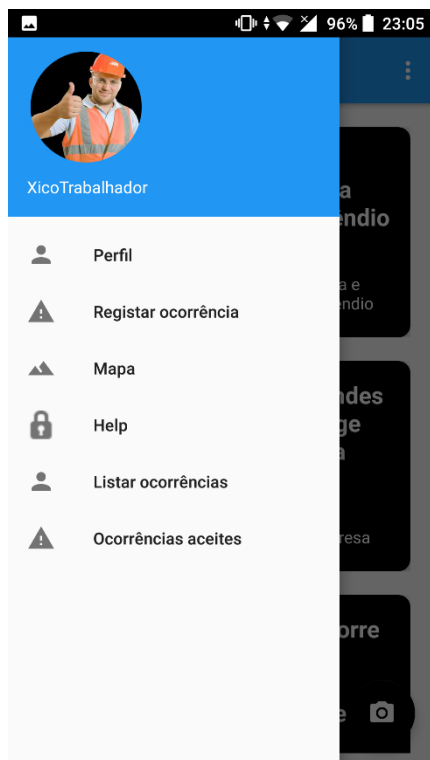
Mobile app login screen. The interface features a dark status bar at the top with icons for signal, Wi-Fi, and battery (28%), and the time 18:24. The main content area is light gray. It displays the GNI logo (a flame icon followed by the text 'GNI'). Below the logo are two input fields: 'utilizador' and 'palavra-passe'. Under these fields are two buttons: 'ENTRAR' and 'REGISTAR'. At the bottom, there is a link 'Esqueceu-se da sua password?' and a checkbox labeled 'Lembrar credenciais'.

Anexo 11

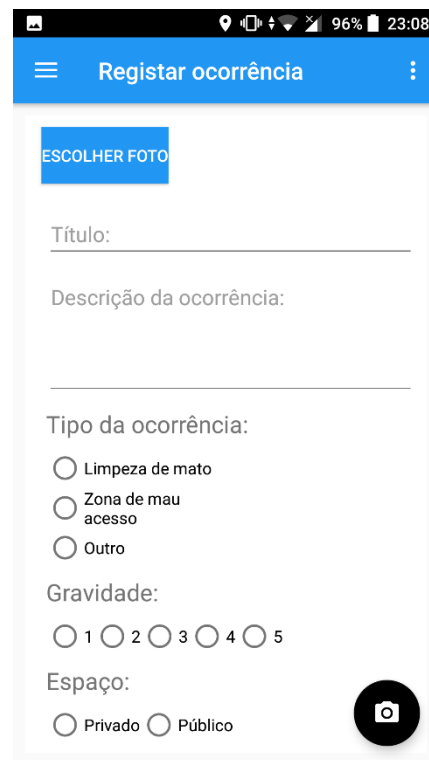


Mobile app registration screen. The interface features a dark status bar at the top with icons for signal, Wi-Fi, and battery (28%), and the time 18:24. The main content area is light gray. It displays a series of input fields for registration: 'Nome', 'Nome de utilizador', 'Email', 'Palavra-passe', 'Palavra-passe', 'Localidade', 'Concelho', and 'Distrito'. Below these fields is a 'REGISTAR' button.

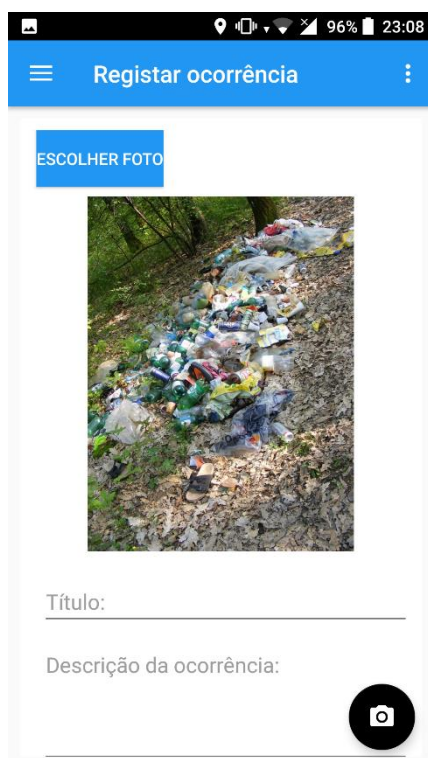
Anexo 12



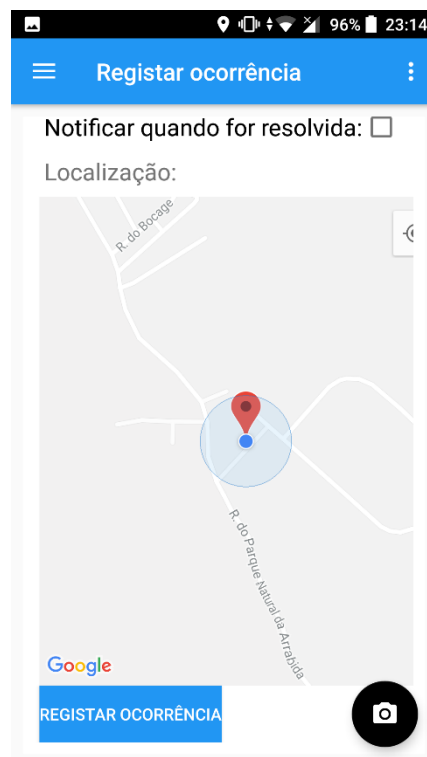
Anexo 13

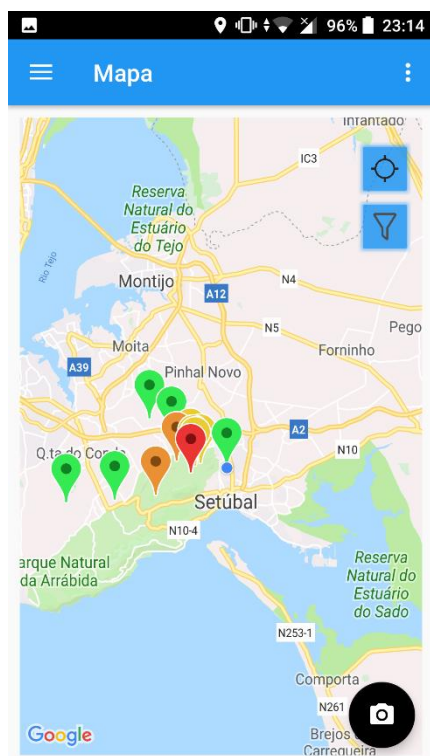


Anexo 14

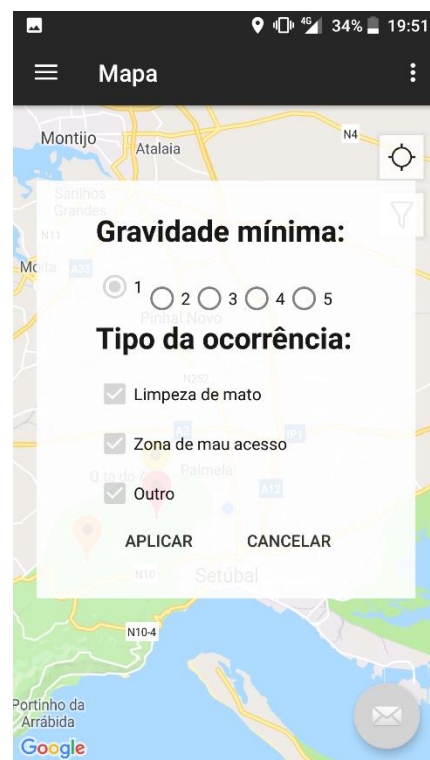


Anexo 15

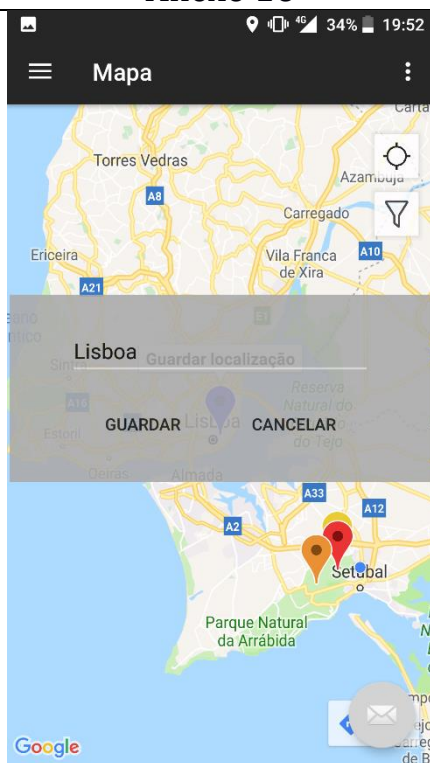




Anexo 16



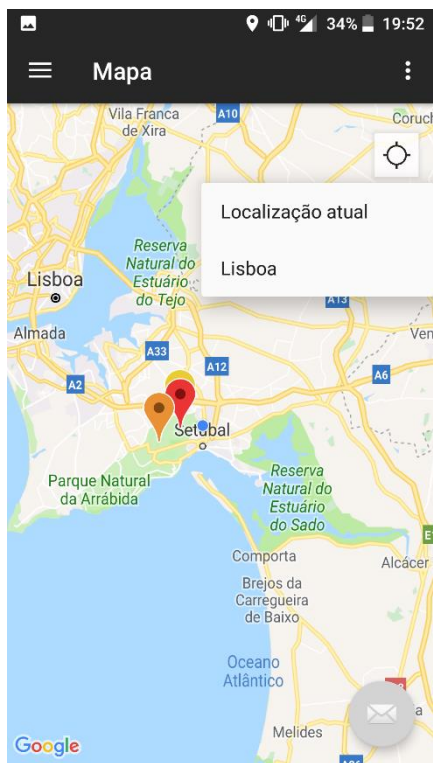
Anexo 17



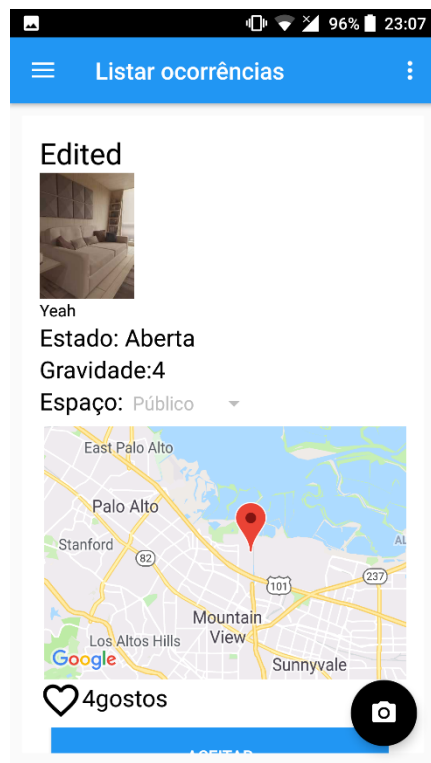
Anexo 18



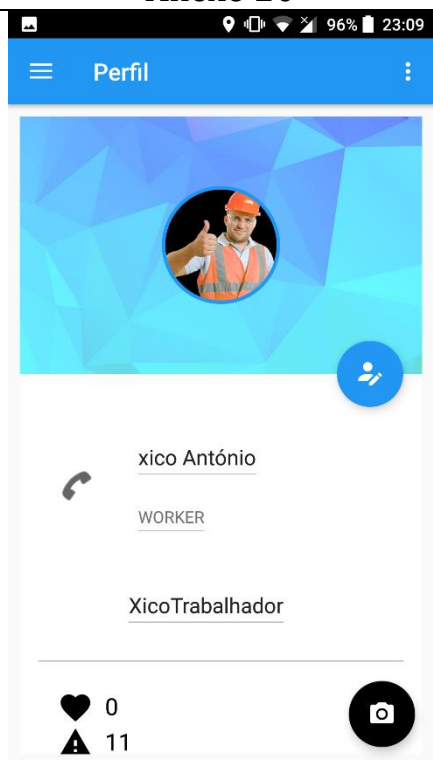
Anexo 19



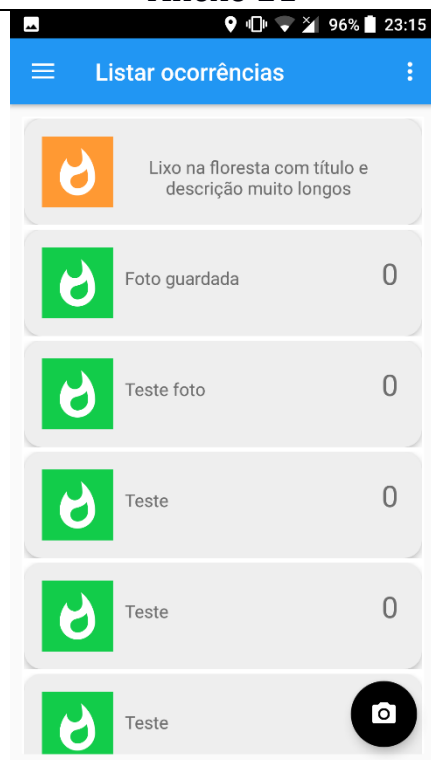
Anexo 20



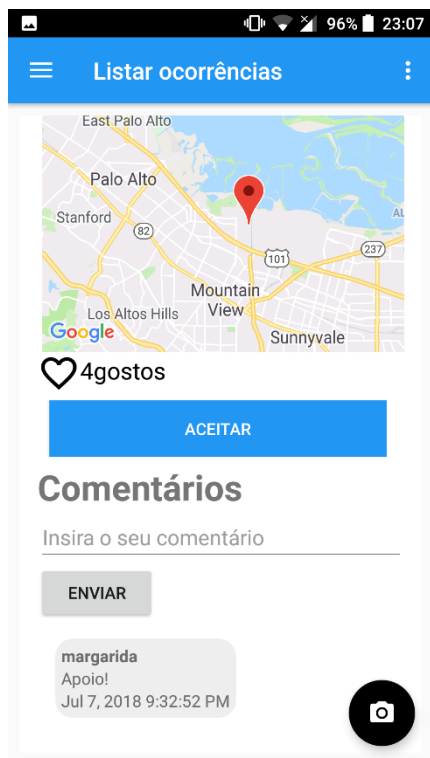
Anexo 21



Anexo 22



Anexo 23



Anexo 24