

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	beta	22/06/2018	
Revisto por:			

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

Identificação da Equipa	Software Elementalists	LOGO da equipa	 Software Elementalists
Nome do Produto	Agni	LOGO DO PRODUTO	
Site de Divulgação (caso exista)	—————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.





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 registro 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.

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, ou então o top de ocorrências (ainda não decidimos completamente – isto é um HIGHLIGHT como já referido anteriormente)
- **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
 - e ainda os amigos/pessoas cujas ocorrências são seguidas pelo utilizador – ainda estamos a analisar melhor este último tópico;
 - 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
- **tabela de operações** que contem a listagem de ocorrências publicas; ao clicar em “informações” poderemos ver mais detalhadamente as informações e estado da ocorrência
- **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 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
- **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

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.



Para além das modificações já referidas em cima, foi também resolvido o problema de atualização de dados quando se clicava numa página (antigamente não atualizava fazendo com que por exemplo um user1 com login que fizesse logout e outro user2 fosse iniciar sessão, as informações no perfil ainda eram as antigas do user1).

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.

As cores predominantes na nossa app são o preto e o branco, e posteriormente o azul, porque são as cores do símbolo da nossa aplicação. Achamos que fica interessante esta combinação de cores mas podemos mudar posteriormente.

Quando o utilizador inicia a aplicação pela primeira vez é direcionado para o ecrã de login que é feito através do username e da password. Se o utilizador não possuir conta pode clicar no botão “Registar” e é direcionado para o ecrã de registo. Na página de login acrescentámos uma checkbox com a legenda “Lembrar credenciais” e o que esta opção faz é guardar as credenciais do utilizador para que da próxima vez que iniciar a aplicação não terá de efetuar o login novamente. Sabemos que a maneira como as credenciais são guardadas é pouco segura e iremos efetuar alterações neste aspeto.

Depois de ser efetuado o login o utilizador é encaminhado para a página inicial da aplicação que apenas possui um botão no canto superior esquerdo que torna visível a barra lateral da aplicação. Nesta barra lateral podem ser escolhidas as seguintes operações:

- **Perfil** – Lista as informações básicas do utilizador, mas também outras informações como a lista de reportes privados do utilizador, lista de ocorrências apoiadas pelo utilizador e ainda os amigos/pessoas cujas ocorrências são seguidas pelo utilizador.  Cada utilizador já pode editar as suas informações. Já pode alterar o seu username, nome, email, localidade e distrito.
- **Registar ocorrência** – Regista uma nova ocorrência. Para registar uma ocorrência é necessário fornecer um título, uma descrição, um tipo, a informação se o espaço é público ou privado, o nível de gravidade da ocorrência, indicar se deseja ser notificado quando a ocorrência for resolvida e a sua localização no mapa. Inicialmente é colocado um marcador na posição do utilizador mas o utilizador deve colocar esse marcador na posição da ocorrência.  Já é possível adicionar uma fotografia a uma ocorrência.
- **Mapa** –  foi adicionado o mapa à aplicação. As ocorrências são transferidas à medida que o mapa muda de posição para que não sejam transferidas ocorrências que não são precisas. O mapa mostra essas ocorrências através de markers em que a cor do marker varia consoante a gravidade da ocorrência. Quando o utilizador clica num marker é apresentado o título dessa ocorrência e quando o utilizador clica nesse título é encaminhado para a página **Detalhes da Ocorrência**. O mapa possui uma funcionalidade de **filtragem** que permite filtrar as ocorrências por tipo e gravidade. Também foi implementada uma funcionalidade para **guardar localizações no mapa**.
- **Estatísticas** – Aqui será possível consultar algumas estatísticas como o desempenho das entidades na resolução de ocorrências e a celeridade dos processos.
- **Listar ocorrências** – Lista todas as ocorrências públicas ou privadas que tenham sido registadas pelo utilizador com o login feito.  Foram efetuadas alterações para que a aplicação apenas transfira 100 ocorrências de cada vez e só transfere mais quando for necessário, isto é, quando o utilizador chega ao fim da lista.
- **Detalhes da Ocorrência** -  É possível listar os detalhes de uma ocorrência clicando no marker no mapa ou clicando na ocorrência na lista de ocorrências. Apresenta seu título, descrição, gravidade, um mapa com a sua localização e a sua foto.
- **Terminar sessão** –  Foi adicionada a opção de terminar sessão no botão situado no canto superior direito da aplicação.

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
}
```

- 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 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:

```
{
    "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 username 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:

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

- 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
}

```

- 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.

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 Plataforma 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

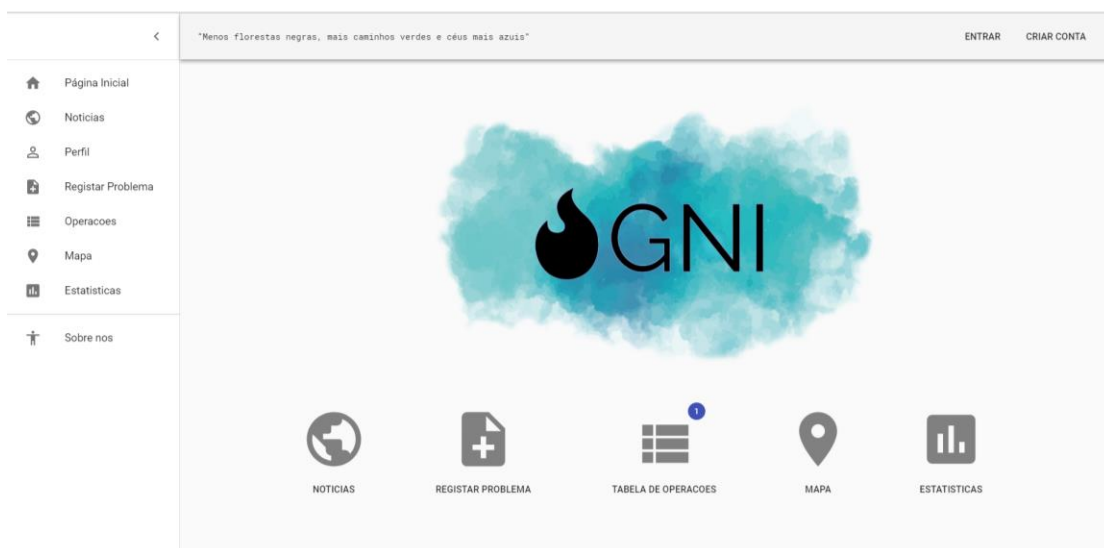
Temos uma secção de notícias onde serão apresentadas as notícias mais populares relativamente ao tema incêndios, ou então o top de ocorrências (ainda não decidimos completamente – isto é um HIGHLIGHT como já referido anteriormente).

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

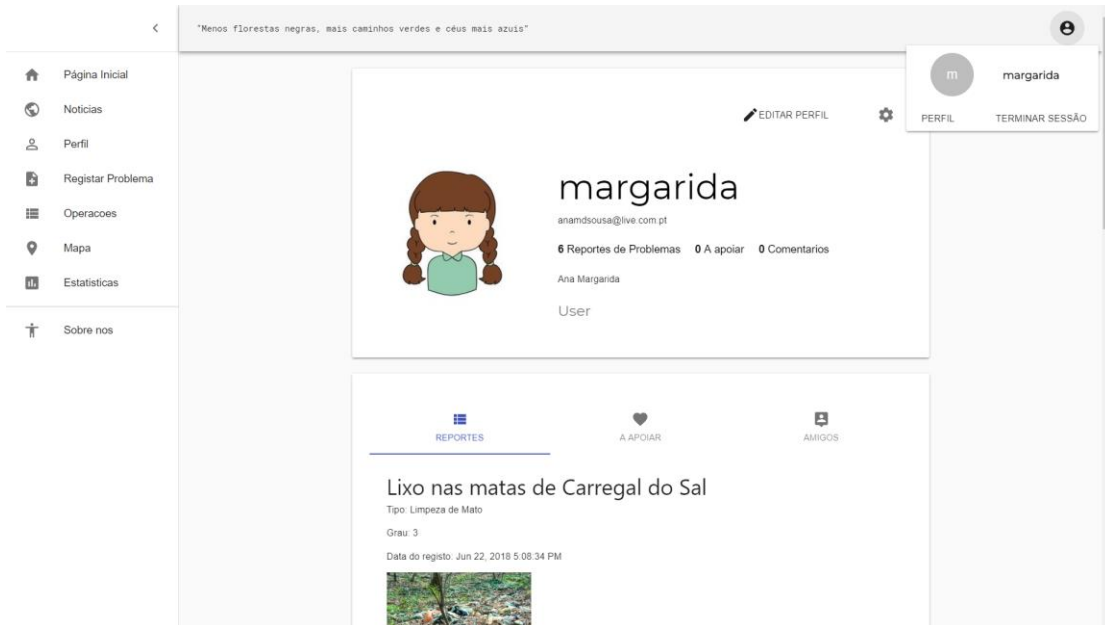
Anexos

Interface do site (aplicação web)

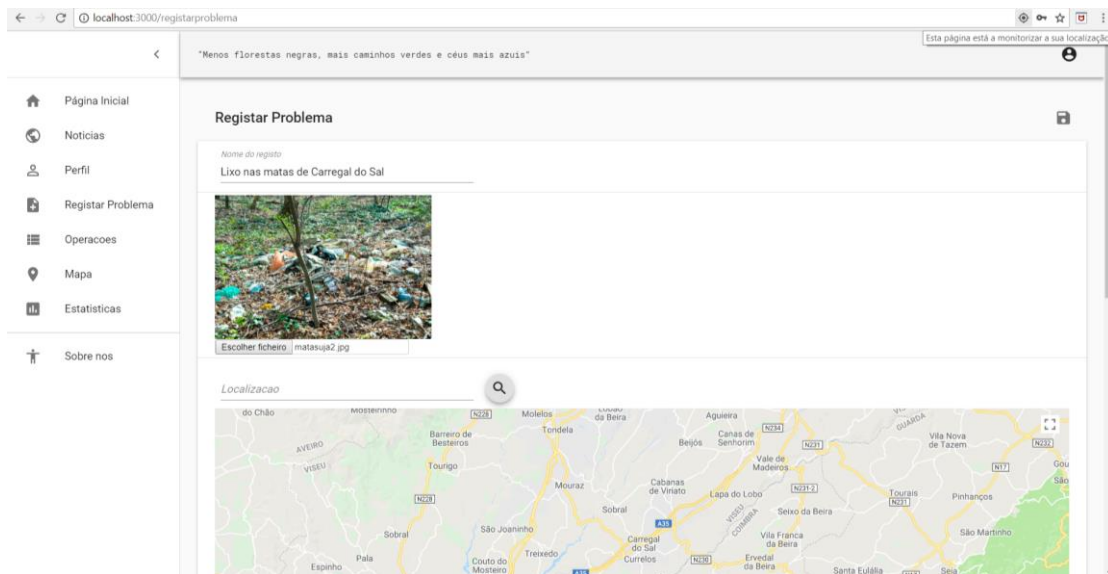


Anexo 1 – Página Inicial Web

Anexo 3 – Criar Conta Web



Anexo 4 – Perfil 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

Anexo 5 – Registrar Problema Web

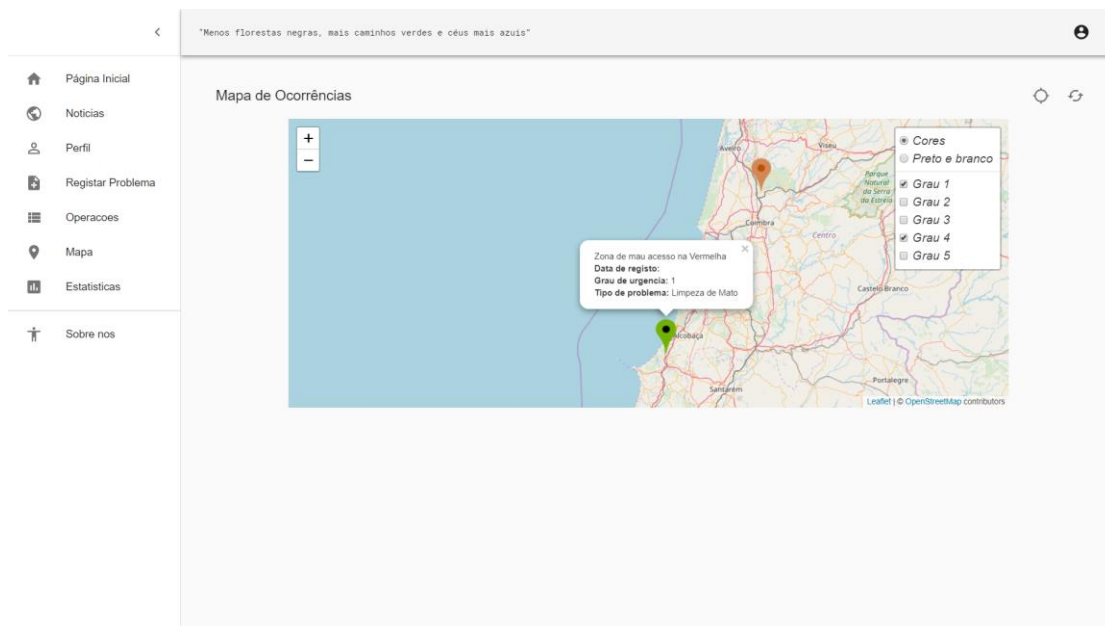
Tem a certeza que pretende registar o problema?

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

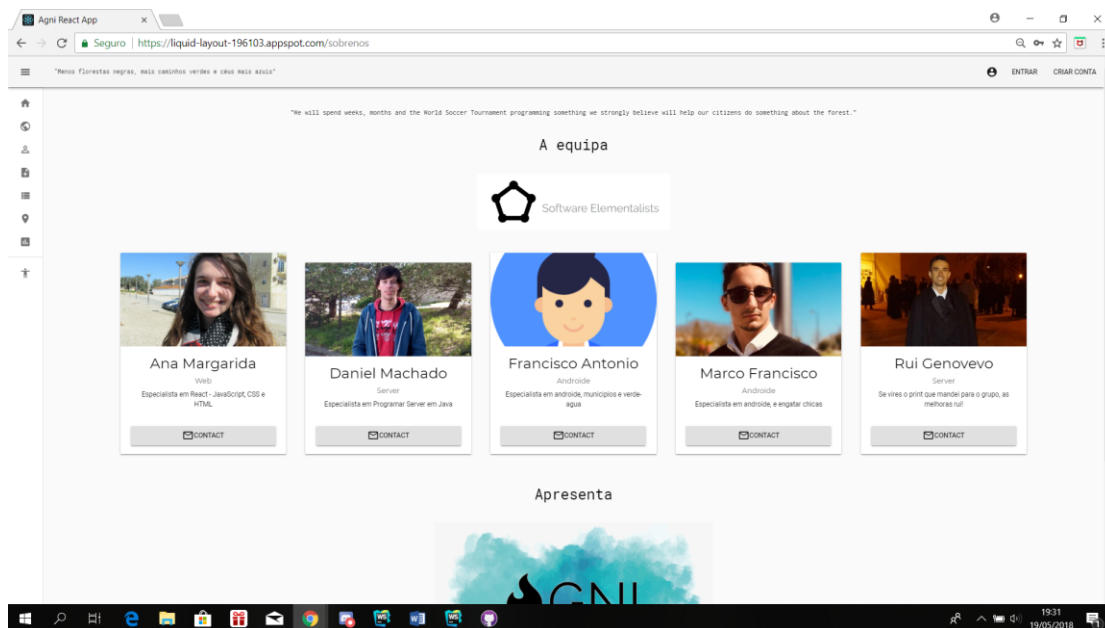
CANCELAR SIM

+ REGISTRAR PROBLEMA

Anexo 6 – Alerta registar problema Web

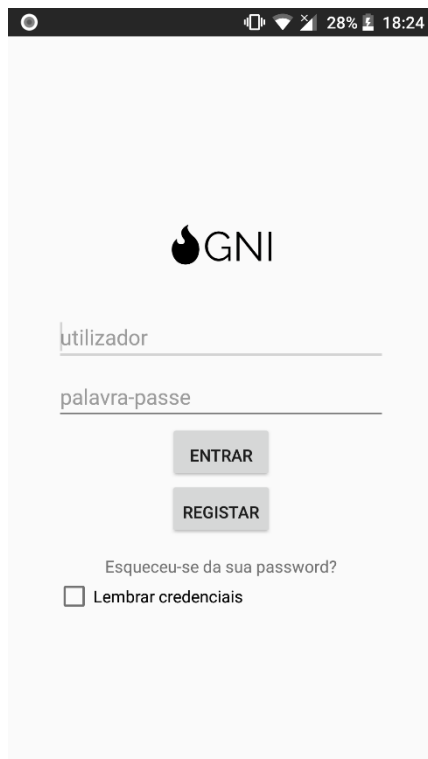


Anexo 9 – Mapa de Ocorrências Web



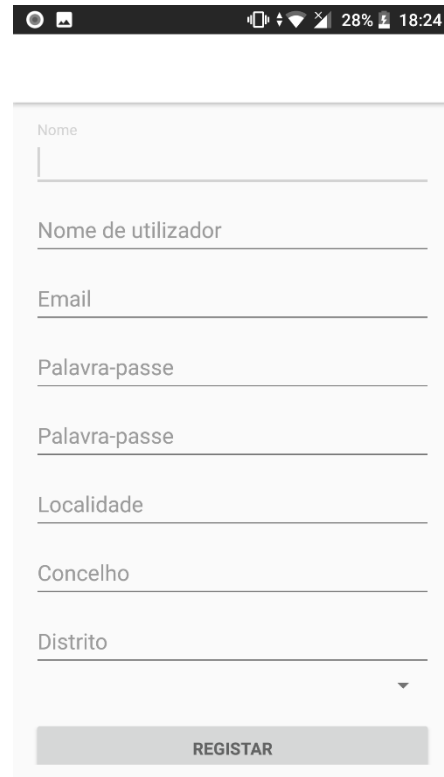
Anexo 10 – Sobre nós Web

Interface do site (aplicação móvel)



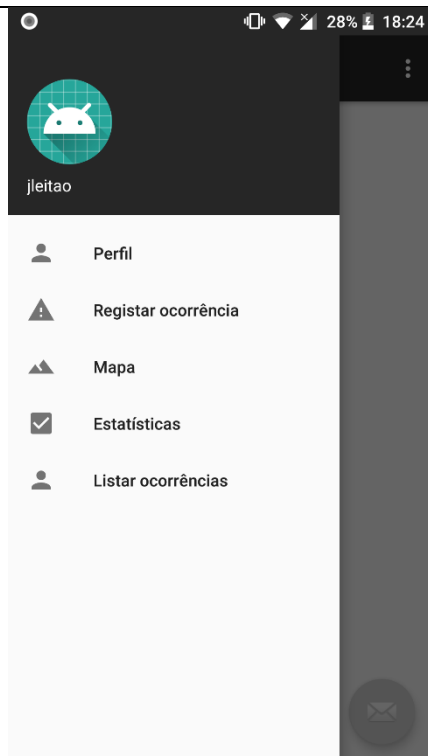
Interface de login da aplicação móvel. O topo exibe o status bar com ícones de bateria, sinal e hora (18:24). O cabeçalho contém o logotipo GNI. Abaixo, há campos de entrada para 'utilizador' e 'palavra-passe'. Botões 'ENTRAR' e 'REGISTAR' estão posicionados abaixo dos campos. Uma opção 'Esqueceu-se da sua password?' e um checkbox 'Lembrar credenciais' estão no rodapé da seção de login.

Anexo 11



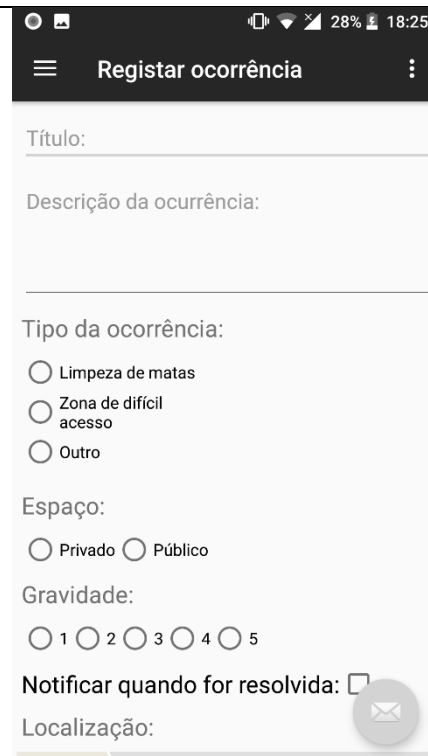
Interface de registo da aplicação móvel. O topo exibe o status bar com ícones de bateria, sinal e hora (18:24). O formulário contém campos para 'Nome', 'Nome de utilizador', 'Email', duas instâncias de 'Palavra-passe', 'Localidade', 'Concelho', 'Distrito' (com uma seta para baixo) e um botão 'REGISTAR' no rodapé.

Anexo 12



Interface de menu principal da aplicação móvel. O topo exibe o status bar com ícones de bateria, sinal e hora (18:24). O cabeçalho mostra um ícone de perfil de usuário e o nome 'jleitao'. O menu principal contém opções: 'Perfil', 'Registar ocorrência', 'Mapa', 'Estatísticas' (com um ícone de marca de seleção) e 'Listar ocorrências'. Um ícone de envelope está no rodapé.

Anexo 13



Interface para registo de ocorrência da aplicação móvel. O topo exibe o status bar com ícones de bateria, sinal e hora (18:25). O formulário contém campos para 'Título:', 'Descrição da ocorrência:', 'Tipo da ocorrência:' (com opções: 'Limpeza de matas', 'Zona de difícil acesso', 'Outro'), 'Espaço:' (com opções: 'Privado', 'Público'), 'Gravidade:' (com opções: '1', '2', '3', '4', '5'), 'Notificar quando for resolvida:' (checkbox) e 'Localização:' (com um ícone de envelope). Um botão de menu está no topo esquerdo.

Anexo 14

28% 18:25

Registar ocorrência

Espaço:

☐ Privado ☐ Público

Gravidade:

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

Notificar quando for resolvida: ☐

Localização:



REGISTAR OCORRÊNCIA

Anexo 15

NOVO!



Anexo 16

34% 19:51

Mapa

Gravidade mínima:

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

Tipo da ocorrência:

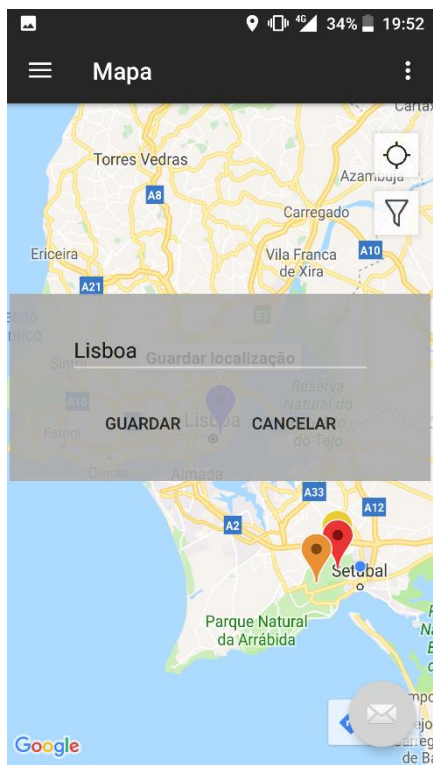
☒ Limpeza de mato

☒ Zona de mau acesso

☒ Outro

APLICAR CANCELAR

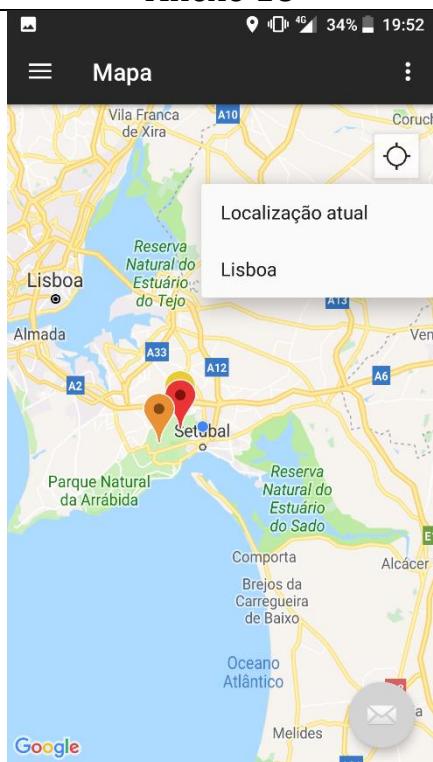
Anexo 17



Anexo 18



Anexo 19



Anexo 20



Anexo 21



Anexo 22



Anexo 23



Anexo 24