



Escola Superior  
de Tecnologia  
e Gestão



Instituto Politécnico  
de Viana do Castelo

## Relatório

Gabriel Sousa, nº 21519

Nuno Oliveira, nº16714

Unidade Curricular: Projeto I

Professor Rui Manuel da Silva Gomes

Professor Ricardo André Pereira Freitas

## RESUMO

Neste relatório, descrevemos detalhada e sucintamente todo o processo de desenvolvimento de um projeto tanto em UML como em base de dados para gerir o funcionamento de uma plataforma de submissão de problemas de uma cidade para a câmara municipal dessa mesma cidade de forma a tornar mais simples e eficiente a forma como os problemas são resolvidos, onde fomos desde a criação dos diversos modelos em UML até à implementação física e esquematização em Base de Dados.

Inicialmente é feita uma apresentação do caso de estudo, contextualização do projeto bem como dos seus principais objetivos.

É feita uma análise de requisitos onde se define os objetivos (aquilo que deve estar presente) da implementação do projeto, e depois é acompanhada a criação do modelo conceptual, bem como a sua análise e validação junto do cliente.

A partir deste modelo é elaborado o modelo lógico, sendo este também analisado e validado junto do cliente antes que se possa proceder à construção do modelo físico.

O modelo físico, que consiste em código SQL, é elaborado a partir do lógico.

Todas as interrogações ou outro tipo de operações sobre a base de dados são efetuadas sobre esse modelo, que é devidamente povoado utilizando código que apresentamos em anexo.

Finalmente, apresentamos conclusões tiradas do trabalho desenvolvido e perspectivas de trabalho futuro no que toca ao aperfeiçoamento nos modelos apresentados neste projeto.

# Conteúdo

<b>Introdução .....</b>	<b>4</b>
<b>Desenvolvimento .....</b>	<b>5</b>
<b>Âmbito do projeto e requisitos.....</b>	<b>5</b>
<b>Diagrama de casos de uso.....</b>	<b>7</b>
<b>Descrição de casos de uso.....</b>	<b>8</b>
<b>Contratos de operações e Diagramas de comunicação .....</b>	<b>23</b>
<b>Diagrama de Classes Conceptual .....</b>	<b>34</b>
<b>Modelo E-R (Alto Nível).....</b>	<b>35</b>
<b>Modelo E-R -Normalizado .....</b>	<b>38</b>
<b>Modelo Relacional.....</b>	<b>39</b>
<b>Modelo Relacional Gerado pela Aplicação.....</b>	<b>44</b>
<b>Views.....</b>	<b>45</b>
<b>Procedures.....</b>	<b>46</b>
<b>Queries Sobre os Caso de uso .....</b>	<b>47</b>
<b>Conclusão.....</b>	<b>52</b>

## Introdução

Neste relatório iremos descrever e aprofundar o trabalho desenvolvido até agora no âmbito da disciplina de Projeto I, mais concretamente, o desenvolvimento de uma plataforma dirigida pela Camara Municipal do local com o objetivo de fazer uma gestão simples e prática de um sistema de manutenção de uma cidade, onde os próprios cidadãos irão ser aqueles que irão comunicar com a plataforma de modo a reportar os problemas que eles acham que deviam de ser corrigidos.

Iremos mais à frente fazer uma descrição mais aprofundada do âmbito do projeto, os requisitos do projeto, o diagrama de caso de uso, onde iremos fazer a descrição, os contratos de operações, o diagrama de comunicações e o diagrama de classes conceptual.

Secundariamente, este documento servirá de suporte à equipa de desenvolvimento, atuando como um elemento orientador e de rastreio, durante a fase de desenvolvimento do sistema, permitindo guiar todos os membros da equipa para um mesmo fim.

# Desenvolvimento

## Âmbito do projeto e requisitos

### Objetivos Específicos

Objetivos específicos podem ser divididos em 3 grupos, o utilizador, o administrador, a entidade resolvente.

### Utilizadores

A primeira fase para a utilização da aplicação, deve ser criar uma conta, caso este ainda não possua uma. Após criada a conta o utilizador efetuar o login na plataforma.

A aplicação deve permitir que os seus utilizadores possam, de uma forma rápida e eficaz, indicar um problema que ache que deve ser resolvido. Para isso deve ser possível indicar num mapa ou inserindo manualmente as coordenadas do local onde existe um problema que determine de forma significativa a informação do problema. Estes problemas deverão ser classificados através de um conjunto de categorias e subcategorias predefinidos pelos administradores da aplicação. Deverá ser possível, ou até obrigatória, a inclusão de uma pequena descrição e também de uma foto da situação. No final o utilizador poderá enviar o problema para ser avaliado e possivelmente resolvido.

Os utilizadores também deveram consultar um histórico dos seus contributos submetidos e quais já se encontram resolvidos. Também será possível fazer a gestão das reclamações que ainda não se encontram resolvidas, como fazer pequenas edições ou anular por completo as mesmas.

O uso inadequado da aplicação por um utilizador deverá ser punido com a restrição ao seu acesso de determinadas funcionalidades da plataforma.

### Administradores

Antes de iniciar a descrição dos requisitos dos administradores, é revelante indicar que estes também são considerados utilizadores da mesma e como tal possuem também as funcionalidades de um utilizador normal.

Os administradores devem ter acesso a um conjunto de ferramentas que lhes permite fazer a gestão das reclamações submetidas e também dos utilizadores da aplicação.

É relevante indicar que para ser possível uma administração deste tipo, os administradores devem pertencer a um órgão executivo colegial do município em questão, mais propriamente, devem pertencer à camara municipal da localidade.

Quando é recebida uma reclamação, o administrador tem acesso a todas as informações submetidas pelo utilizador, que deve permanecer anónimo, e após uma avaliação geral do problema deve classificar o grau de urgência do mesmo através de um sistema classificatório de 0 a 5. Também deverá ser possível resolver uma reclamação durante esta etapa inicial, marcando-a como resolvida ou até anulando a mesma caso o administrador a considere inválida. Quando uma reclamação é considerada inválida esta irá ser classificada como tal.

Após uma avaliação inicial, se uma reclamação não for de imediato resolvida, ficará em estado pendente até que sejam resolvidas. Ao resolver uma reclamação o administrador deve redigir uma mensagem que indica as informações revelantes à reclamação e informa a resolução do problema ao utilizador.

Deverá ser também possível aceder ao histórico de reclamações resolvidas por cada administrador.

Os administradores devem também ser capazes de restringir o acesso a determinadas funcionalidades da plataforma, temporariamente ou, se se justificar, permanentemente, a utilizadores que fazem o uso indevido da mesma. Por exemplo, se um utilizador tem 3 reclamações anuladas, a atividade do mesmo deverá ser avaliada por um administrador que poderá suspender o mesmo se assim se justificar. Esta suspensão poderá ser anulada caso haja razões fortes suficientes para isso.

### **Comunicação Câmara-Instituição**

Após aceite e classificada uma submissão, é encarregado à Câmara Municipal da localidade em questão, através das filtrações das submissões por tópicos, de comunicar com a entidade mais adequada de modo a obter informações de como e quando o processo da resolução será iniciado.

### **Entidade Resolvente**

Após recebido as informações através da plataforma, a entidade que irá tratar do processo pode alterar a forma como irá confrontar a situação ou até mesmo adiar a data prevista da resolução. O utilizador, caso se verifique esta situação, deve ser automaticamente informado da alteração da situação.

Além disso, a entidade deverá indicar ao administrador que a submissão encontra-se em resolução e consequentemente resolvida de forma a ser possível terminar o processo.

### **Requisitos**

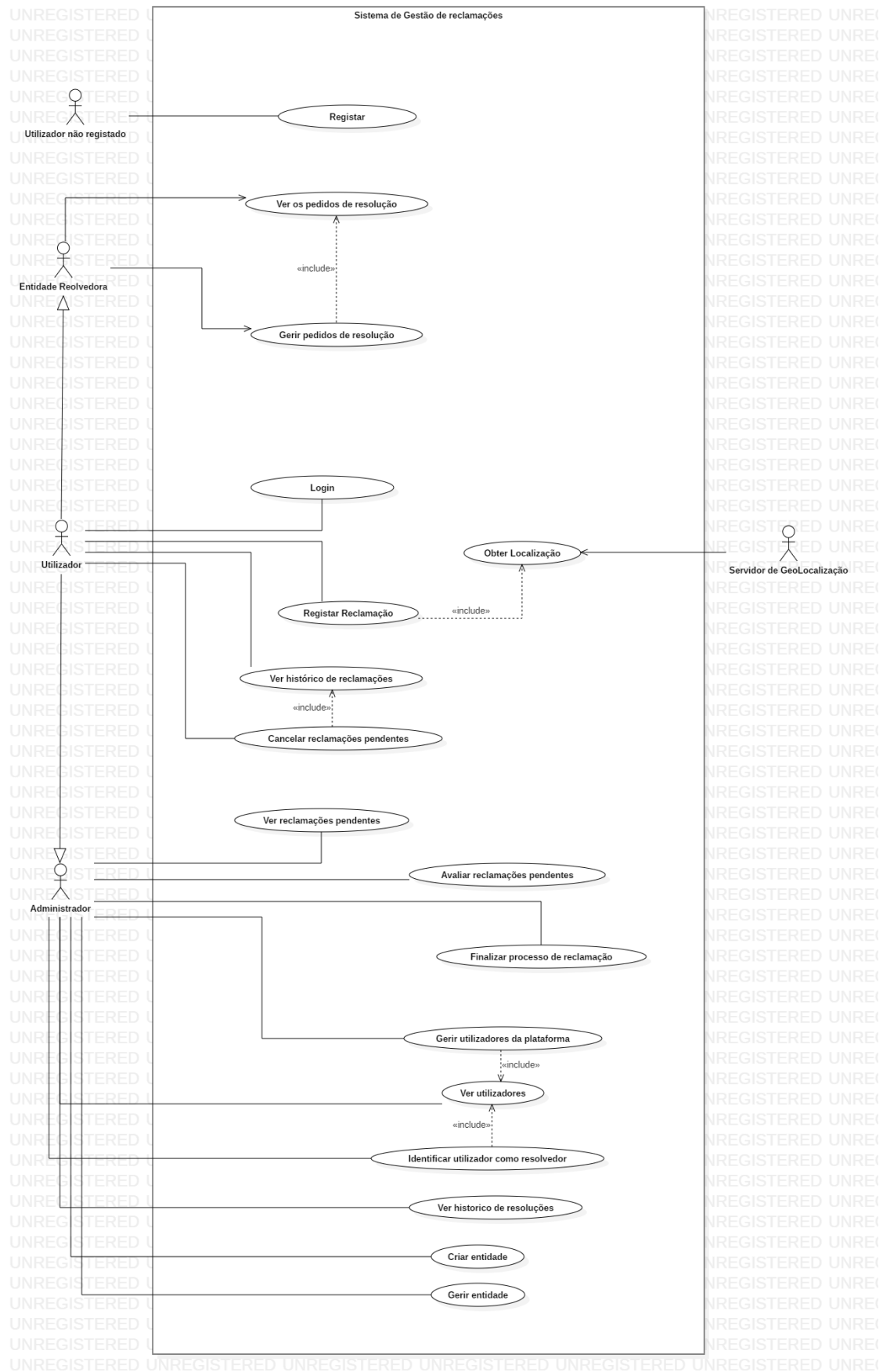
#### ***Requisitos Funcionais***

Os objetivos gerais do projeto são a criação de uma aplicação na qual podemos indicar problemas existentes numa determinada localidade para que estes sejam vistos, resolvidos e geridos por uma entidade superior reguladora. Esta plataforma irá estar localizada um servidor.

#### ***Requisitos Não Funcionais***

Esta aplicação será gerida pela camara municipal da localidade, onde os moradores da localidade podem usufruir da plataforma de modo a comunicar problemas que acham que deviam de ser abordados pela camara municipal de forma a tratar deles. De seguida, caso os administradores da plataforma considerem o problema grave o suficiente para ser abordado, estes irão comunicar com a entidade resolvente mais adequada ao problema, de modo a ser resolvido.

## Diagrama de casos de uso



## Descrição de casos de uso

**Caso de uso:** Ver pedidos de resolução

**Domínio:** Sistema

**Nível:** Objetivo do utilizador

**Actor Primário:** Entidade Resolvedora.

**Partes Interessadas & Interesses:** Habitantes de uma cidade, câmara local, entidades trabalhadoras

**Pré-condição:** O utilizador terá que estar identificado e autenticado como entidade resolvedora

**Garantia de Sucesso:** O utilizador é capaz de ver os pedidos que foram atribuídos ao seu departamento deve resolver

**Iniciador:** Entidade Resolvedora

### Cenário Principal de Sucesso

1. O utilizador acede à página de listagem de pedidos.
2. O utilizador adiciona filtros de pesquisa aos pedidos.
3. O sistema lista todos os pedidos que foram dirigidos ao departamento do utilizador.

### Extensões

#### 2. a) Filtros de pesquisa

##### 1. a) Código de identificação do pedido:

##### 1. O utilizador introdução o código do pedido:

##### 1. a) Código de pedido incompleto ou não existente

1. O sistema apresenta todos os pedidos com códigos semelhantes ao introduzido.

##### 1. b) Código completo existente

1. O sistema apresenta o pedido com o código introduzido ao utilizador.

##### 1. b) Estado do pedido:

##### 1. O utilizador faz a seleção estado do pedido:

1. Pendente
2. Em Resolução
3. Resolvido
4. Cancelado



**Caso de uso:** Gerir pedidos de resolução

**Domínio:** Sistema

**Nível:** Objetivo do utilizador

**Actor Primário:** Entidade Resolvedora

**Partes Interessadas & Interesses:** Habitantes de uma cidade, câmara local, entidades trabalhadoras

**Pré-condição:** O utilizador terá que estar identificado e autenticado como entidade resolvedora e selecionado um pedido para gerir.

**Garantia de Sucesso:** O utilizador consegue iniciar o processo de resolução, concluir ou cancelar um pedido feito à entidade.

**Iniciador:** Entidade Resolvedora

### **Cenário Principal de Sucesso**

1. O sistema lista toda a informação disponível relativa ao problema selecionado.
2. O utilizador analisa a informação e, conforme a sua avaliação, define um estado para o pedido.
3. O sistema guarda os dados atualizados do pedido.

### **Extensões**

2. a) Marcar como “Em Resolução”

2. b) Marcar como “Resolvido”

1. O utilizador formata uma mensagem que indica a resolução do problema.

2. c) Cancelar o pedido

1. O utilizador seleciona o motivo que o levou a cancelar o pedido.

3 a) Dois utilizadores tentam atualizar o estado do pedido em simultâneo:

1. O sistema compara o estado inicial do pedido com o estado atual do pedido no sistema.

1. a) O sistema deteta uma alteração entre as duas versões do pedido

1. O sistema avisa o utilizador que o pedido foi alterado desde o início da sua alteração.
2. O utilizador avalia os novos dados do pedido e seleciona a opção pretendida.
  2. a) O utilizador recarrega os dados atuais do pedido.
  2. b) O utilizador continua a edição do pedido.

3 b) O utilizador pretende cancelar a alteração do pedido:

1. O utilizador sai da página relativa ao pedido em questão sem o alterar ou guardar alterações.

**Caso de uso:** Login

**Domínio:** Sistema

**Nível:** Objetivo do utilizador

**Actor Primário:** Utilizador

**Partes Interessadas & Interesses:** Habitantes de uma cidade, câmara local, entidades trabalhadoras

**Garantia de Sucesso:** O utilizador fica identificado e autenticado na aplicação.

**Iniciador:** Utilizador

**Cenário Principal de Sucesso**

1. O utilizador acede a página de login do sistema.
2. O utilizador introduz os seus dados de login e submete-os.
3. O sistema valida os dados do utilizador e identifica o mesmo como tal.

**Extensões**

3. a) Dados introduzidos inválidos

1. O sistema assinala o erro e não faz a identificação do utilizador.
2. O utilizador introduz novamente os seus dados de utilizador.

3. b) O utilizador foi suspenso

1. O sistema indica ao utilizador que a sua conta esta suspensa e o período de tempo a qual irá permanecer assim e barra o acesso do utilizador à aplicação.

3. c) O utilizador foi banido

2. O sistema indica ao utilizador que a sua conta esta banida e barra o acesso do mesmo à aplicação.

**Caso de uso:** Registar Reclamação

**Domínio:** Sistema

**Nível:** Objetivo-do-Utilizador

**Actor Primário:** Utilizador

**Partes Interessadas & Interesses:** Habitantes de uma cidade, câmara local, entidades trabalhadoras

**Pré-condição:** O utilizador está validado e identificado no sistema.

**Garantia de Sucesso:** O utilizador regista um pedido.

**Iniciador:** Utilizador

#### **Cenário Principal de Sucesso**

1. O utilizador inicia o processo de registo de uma reclamação.
2. O utilizador obtém a sua localização.
3. O utilizador categoriza o seu pedido.
4. O utilizador descreve o problema identificando-o textualmente.
5. O utilizador obtém uma foto descritiva do problema em questão.
6. O utilizador submete a sua reclamação.
7. O sistema valida e guarda o novo pedido.

#### **Extensões**

##### **2. a) Localização obtida por serviço GPS**

1. O utilizador autoriza o sistema a obter a sua localização atual.
2. O sistema contacta um servidor de GeoLocalização e obtém a localização do utilizador
2. a) O sistema não consegue contactar o servidor de GeoLocalização
  1. O sistema informa o erro ao utilizador e pede que este introduza a sua localização manualmente.
3. O sistema regista a localização obtida como a localização do utilizador.

##### **2. b) Localização introduzida manualmente**

1. O utilizador introduz a localização do local onde detetou o problema.
2. O sistema valida as informações obtidas e atualiza a localização da reclamação.

##### **3. a) O utilizador não categorizou o problema**

1. O sistema indica ao utilizador a obrigatoriedade da caracterização do seu pedido.
2. O utilizador define o tipo de problema encontrado e continua com a formatação do pedido.

##### **4. a) O utilizador não descreveu o problema**

1. O sistema indica ao utilizador a obrigatoriedade da inclusão de uma descrição do problema encontrado.
2. O utilizador define o tipo de problema encontrado e continua com a formatação do pedido.

##### **5. a) O utilizador não pretende obter uma foto do problema.**

1. O utilizador continua a formatação ou submissão do pedido sem incluir uma fotografia do problema em questão.

**Caso de uso:** Ver histórico de reclamações

**Domínio:** Sistema

**Nível:** Objetivo-do-Utilizador

**Actor Primário:** Utilizador

**Partes Interessadas & Interesses:** Habitantes de uma cidade, câmara local, entidades trabalhadoras

**Pré-condição:** O utilizador está validado e identificado no sistema e já fez alguma reclamação.

**Garantia de Sucesso:** O utilizador vê as reclamações feitas até à data.

**Iniciador:** Utilizador

#### **Cenário Principal de Sucesso**

1. O utilizador acede a página do histórico de reclamações.
  2. O utilizador aplica filtros de pesquisa à sua lista.
  3. O sistema lista todas as reclamações correspondentes.
- O utilizador repete os passos 2-3 até encontrar o pretendido.

#### **Extensões**

2. a) Procura sem definição de filtros

1. O sistema lista todas as reclamações feitas pelo utilizador.

**Caso de uso:** Cancelar reclamações pendentes

**Domínio:** Sistema

**Nível:** Objetivo-do-Utilizador

**Actor Primário:** Utilizador

**Partes Interessadas & Interesses:** Habitantes de uma cidade, câmara local, entidades trabalhadoras

**Pré-condição:** O utilizador está validado e identificado no sistema e detém de alguma reclamação em estado pendente. O utilizador também já selecionou a atualização que quer cancelar.

**Garantia de Sucesso:** O utilizador regista um pedido.

**Iniciador:** Utilizador

### **Cenário Principal de Sucesso**

1. O utilizador inicia o processo de cancelamento de uma reclamação.
2. O sistema pergunta confirma se o utilizador quer apagar a sua reclamação.
3. O utilizador confirma o facto.
4. O sistema cancela o pedido.

### **Extensões**

3. a) O utilizador não pretende cancelar o seu pedido

1. O utilizador não confirma o pedido de cancelamento do seu pedido.
2. O sistema cancela a operação, alerta o utilizador e redireciona-o para outra pagina do sistema.

4. a) O pedido foi aprovado durante o cancelamento do pedido

1. O sistema confirma se o estado do atual do pedido é igual ao correspondente na base de dados.
2. O sistema deteta uma alteração.
3. O sistema ignora alteração do estado do pedido e cancela o mesmo de qualquer forma.

**Caso de uso:** Obter Localização

**Domínio:** Sistema

**Nível:** Objetivo-do-Utilizador

**Actor Primário:** Servidor de GeoLocalização

**Partes Interessadas & Interesses:** Habitantes de uma cidade, câmara local, entidades trabalhadoras

**Pré-condição:** O utilizador está validado e identificado no sistema, e encontra-se a fazer uma nova reclamação.

**Garantia de Sucesso:** É obtida a localização do utilizador.

**Garantia Mínima:** O sistema permite a introdução manual da localização do problema.

**Iniciador:** Utilizador

#### **Cenário Principal de Sucesso**

1. O sistema pede ao servidor de GeoLocalização que este forneça a informação acerca da localização atual do utilizador.
2. O servidor de GeoLocalização responde com a informação pedida.
3. O servidor utiliza essa informação durante o registo de uma reclamação pelo utilizador.

#### **Extensões**

1. a) O sistema não encontra o servidor de GeoLocalização
  1. O sistema tenta contactar o servidor de GeoLocalização.
  2. O sistema não consegue contactar o servidor de GeoLocalização.
  3. O sistema informa o utilizador do acontecido e pede ao mesmo para inserir a sua localização manualmente.
2. a) O servidor de GeoLocalização não responde
  4. O sistema não obtém resposta do servidor de GeoLocalização.
  5. O sistema informa o utilizador do acontecido e pede ao mesmo para inserir a sua localização manualmente.

**Caso de uso:** Registar utilizador.

**Domínio:** Sistema

**Nível:** Objetivo do utilizador

**Actor Primário:** O não utilizador.

**Partes Interessadas & Interesses:** O utilizador, o administrador e a câmara municipal.

**Garantia de Sucesso:** A informação do registo é armazenada.

**Iniciador:** O não utilizador

**Cenário Principal de Sucesso**

1. O não utilizador chega à plataforma com o intuito de se registar.
2. O não utilizador procede à introdução dos dados pessoais, necessários ao registo.
3. O não utilizador confirma os dados, procedendo à página de conclusão do registo.

**Extensões**

3. a) Já existe um utilizador com o mesmo nome de utilizador.
  6. O sistema informa o utilizador que o nome escolhido já foi utilizado.
  7. O utilizador tenta fazer o registo com outro nome.

**Caso de uso:** Ver reclamações pendentes.

**Domínio:** Sistema

**Nível:** Objetivo do administrador.

**Actor Primário:** O administrador.

**Partes Interessadas & Interesses:** O utilizador, o administrador, a câmara municipal e as entidades trabalhadoras.

**Pré-condição:** O administrador está identificado e validado na plataforma.

**Garantia de Sucesso:** O administrador vê as reclamações pendentes.

**Iniciador:** O administrador

**Cenário Principal de Sucesso**

1. O administrador acede à página das reclamações pendentes.
2. O administrador aplica filtros de pesquisa à sua lista.
3. O sistema lista todas as reclamações pendentes pretendidas.  
O administrador repete os passos 2-3 até encontrar o pretendido.

**Extensões**

2. a) Procura sem definição de filtros

1. O sistema lista todas as reclamações feitas pelo utilizador.



**Caso de uso:** Avaliar reclamações pendentes

**Domínio:** Sistema

**Nível:** Objetivo do administrador.

**Actor Primário:** O administrador.

**Partes Interessadas & Interesses:** O utilizador, o administrador, a câmara municipal e as entidades trabalhadoras.

**Pré-condição:** O administrador está identificado e validado na plataforma e já selecionou uma reclamação pendente.

**Garantia de Sucesso:** O administrador avalia uma reclamação e submete-a para a entidade correta.

**Iniciador:** O administrador

**Cenário Principal de Sucesso**

1. O administrador avalia os dados da reclamação
2. O administrador aprova o processo de resolução da reclamação
3. O administrador avalia a urgência da reclamação numa escala de 1 a 5.
4. O administrador, se necessário, altera a entidade resolvidora a qual se destina o pedido.
5. O administrador confirma a avaliação feita por ele.
6. O sistema recebe a avaliação e guarda a informação

**Extensões**

3. a) O administrador não aprova o processo de resolução da reclamação
  1. O administrador confirma a avaliação.
  2. O sistema recebe a avaliação dada como não aprovada e guarda a informação.
5. a) O administrador não confirma a avaliação
  1. O administrador é redirecionado para a página das reclamações pendentes.

**Caso de uso:** Finalizar processo de reclamação.

**Domínio:** Sistema

**Nível:** Objetivo do administrador.

**Actor Primário:** O administrador.

**Partes Interessadas & Interesses:** O utilizador, o administrador, a câmara municipal e as entidades trabalhadoras.

**Pré-condição:** O administrador está identificado e validado na plataforma e selecionou o pedido a ser terminado.

**Garantia de Sucesso:** O administrador finaliza uma avaliação.

**Iniciador:** O administrador

**Cenário Principal de Sucesso**

1. O administrador acede à página das reclamações pendentes.
2. O administrador seleciona uma reclamação a finalizar
3. O sistema redireciona o administrador para os dados da reclamação.
4. O administrador lê os dados da reclamação e como esta foi resolvida
5. O administrador finaliza a reclamação, dando-a como concluída.
6. O sistema recebe a finalização da reclamação e guarda a informação.

**Extensões**

5. a) O administrador não finaliza a reclamação

1. O sistema recebe a avaliação dada como não aprovada e guarda a informação.

**Caso de uso:** Gerir utilizadores da plataforma

**Domínio:** Sistema

**Nível:** Objetivo do administrador.

**Actor Primário:** O administrador.

**Partes Interessadas & Interesses:** O utilizador, o administrador, a câmara municipal

**Pré-condição:** O administrador está identificado e validado na plataforma.

**Garantia de Sucesso:** O administrador altera o estado da conta de um utilizador.

**Iniciador:** O administrador

**Cenário Principal de Sucesso**

1. O administrador acede à página da lista dos utilizadores.
2. O administrador seleciona um utilizador a inspecionar
3. O sistema lista as informações relativas ao utilizador selecionado.
4. O administrador avalia o histórico de utilização do utilizador
5. O administrador toma medidas sobre a conta do utilizador
6. O sistema guarda as alterações à conta do utilizador e guarda a informação.

**Extensões**

5. a) O administrador suspende o utilizador

1. O administrador decide uma data a dar à suspensão
2. O administrador confirma a suspensão
3. O sistema recebe a confirmação da suspensão e guarda os dados.

5. b) O administrador bane a conta do utilizador

1. O sistema recebe a confirmação da suspensão e guarda os dados.

**Caso de uso:** Identificar utilizador como resolvedor

**Domínio:** Sistema

**Nível:** Objetivo do administrador.

**Actor Primário:** O administrador.

**Partes Interessadas & Interesses:** O utilizador, o administrador, a câmara municipal

**Pré-condição:** O administrador está identificado e validado na plataforma e já efetuou alguma avaliação de pedidos.

**Garantia de Sucesso:** O administrador vê as reclamações pendentes.

**Iniciador:** O administrador

**Cenário Principal de Sucesso**

1. O administrador altera a função do utilizador e atribui-lhe a posição adequada.
2. O administrador finaliza a operação.
3. O sistema guarda os novos dados.

**Extensões**

1. a) O utilizador já tem uma entidade definida

1. O administrador altera a sua entidade.

1. b) O administrador pretende remover a entidade de um utilizador

1. O administrador define a entidade do utilizador como “nenhuma”.

1. c) O utilizador cuja a função foi alterada tem um pedido em resolução

1. O administrador seleciona um utilizador dentro da entidade do mesmo para adquirir os pedidos ativos do utilizador em questão.

**Caso de uso:** Criar entidade

**Domínio:** Sistema

**Nível:** Objetivo do administrador.

**Actor Primário:** O administrador.

**Partes Interessadas & Interesses:** O utilizador, o administrador, a câmara municipal

**Pré-condição:** O administrador está identificado e validado na plataforma.

**Garantia de Sucesso:** O administrador cria uma nova entidade.

**Iniciador:** O administrador

**Cenário Principal de Sucesso**

1. O administrador acede a página de criação de entidades.
2. O administrador fornece as informações necessárias para a criação de uma entidade.
3. O administrador atribui a entidade funções.
4. O administrador finaliza o processo de criação.
5. O sistema guarda as novas informações.

**Extensões**

2. a) A entidade a ser criada já existe
  1. O sistema avisa o administrador e pede que altera a informação relativa a entidade.
  2. O utilizador altera a informação.
3. a) Não atribuir funções a entidade
  1. O administrador não preenche o campo relativo as funções da entidade.
4. a) Cancelar o processo
  1. O administrador sai da página de criação de uma entidade sem finalizar o processo.

**Caso de uso:** Gerir entidade

**Domínio:** Sistema

**Nível:** Objetivo do administrador.

**Actor Primário:** O administrador.

**Partes Interessadas & Interesses:** O utilizador, o administrador, a câmara municipal

**Pré-condição:** O administrador está identificado e validado na plataforma e selecionou uma entidade a gerir.

**Garantia de Sucesso:** O administrador altera as informações relativas a uma entidade.

**Iniciador:** O administrador

**Cenário Principal de Sucesso**

1. O administrador altera as informações pretendidas da entidade.
2. O administrador finaliza o processo.
3. O sistema guarda as alterações efetuadas.

**Extensões**

1. a) As informações alteradas entram em conflito com as informações de outras entidades.
  1. O sistema indica o problema ao administrador.
  2. O administrador altera as informações introduzidas de forma a resolver o problema.
- 1 b) São alteradas as funções atuais da entidade
  1. a) Os utilizadores detêm de pedidos ativos
    1. O administrador atribui os pedidos ativos a novos utilizadores cuja entidade tem a função correta.
    2. O sistema confirma a alteração efetuada.
  1. b) Não existem pedidos ativos
    1. O sistema confirma a alteração efetuada.
2. a) Cancelar processo
  1. O utilizador fecha a página em questão sem finalizar o processo de alteração de informações.

**Caso de uso:** Ver utilizadores

**Domínio:** Sistema

**Nível:** Objetivo do utilizador

**Actor Primário:** Administrador

**Partes Interessadas & Interesses:** Habitantes de uma cidade, câmara local, entidades trabalhadoras

**Pré-condição:** O administrador está identificado e validado na plataforma.

**Garantia de Sucesso:** O administrador tem acesso a uma lista com todos os utilizadores do sistema.

**Iniciador:** Administrador

#### **Cenário Principal de Sucesso**

4. O administrador acede à página de utilizadores
5. O administrador adiciona filtros de pesquisa aos utilizadores.
6. O sistema lista todos os utilizadores dentro dos parâmetros definidos pelo administrador.

#### **Extensões**

##### 2. a) Filtros de pesquisa

###### 1. a) Nome, código ou email do utilizador

1. O administrador introduz o parâmetro do utilizador.
2. O sistema lista todos os utilizadores cuja sua identificação seja igual ou semelhante a introduzida.

###### 1. b) Entidade

1. O administrador seleciona uma entidade.
2. O sistema lista os utilizadores pertencentes a essa entidade.

## Contratos de operações e Diagramas de comunicação

### Contrato criarNovoUtilizador

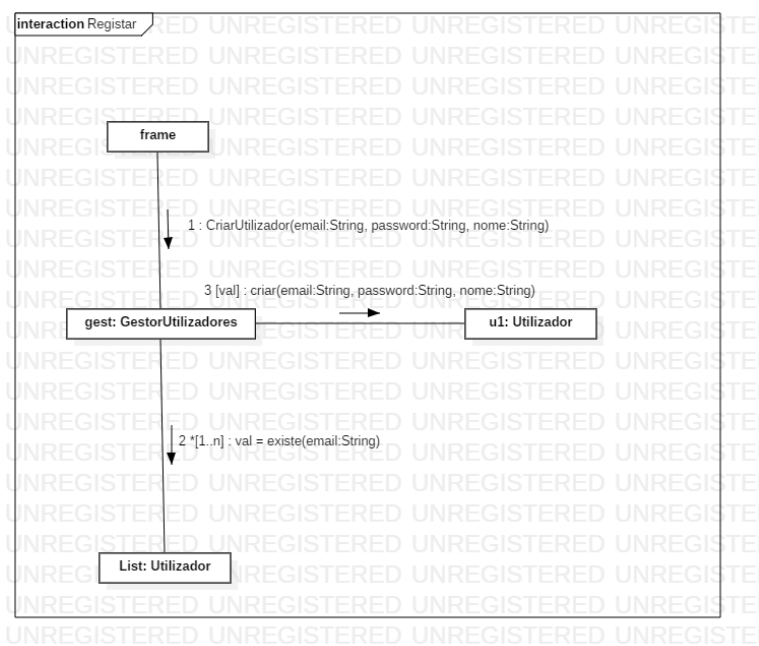
Operação: criarNovoUtilizador (nome: String, password: String, email: String)

Referências Cruzadas: Caso de uso – Registar.

Pré-condições: O email do utilizador (email) não existe no sistema.

Pós-Condições:

1. Foi criado uma nova instância de Utilizador u.
2. u.email tornou-se em email (modificação de atributo).
3. u.password tornou-se em password (modificação de atributo).
4. u.nome tornou-se em nome (modificação de atributo).



### Contrato login

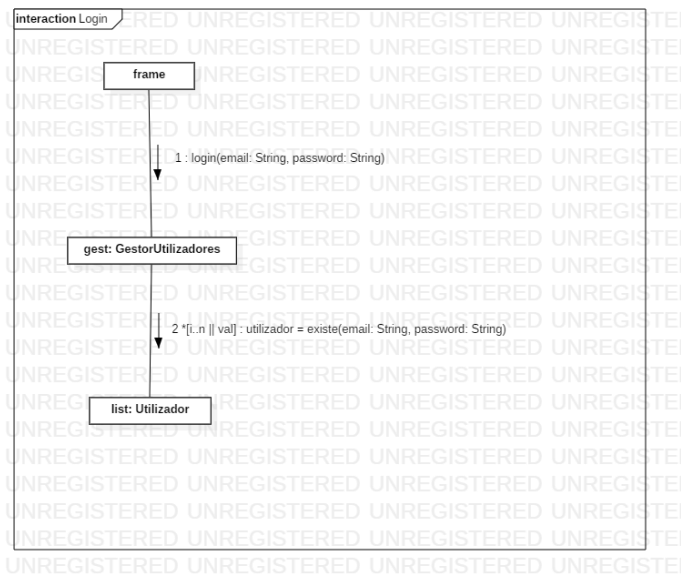
Operação: login (email: String, password: String)

Referências Cruzadas: Caso de uso – Login

Pré-condições:

Pós-Condições:

1. Foi inicializado um Utilizador user, baseado na correspondência com email e password.



## Contrato criarReclamacao

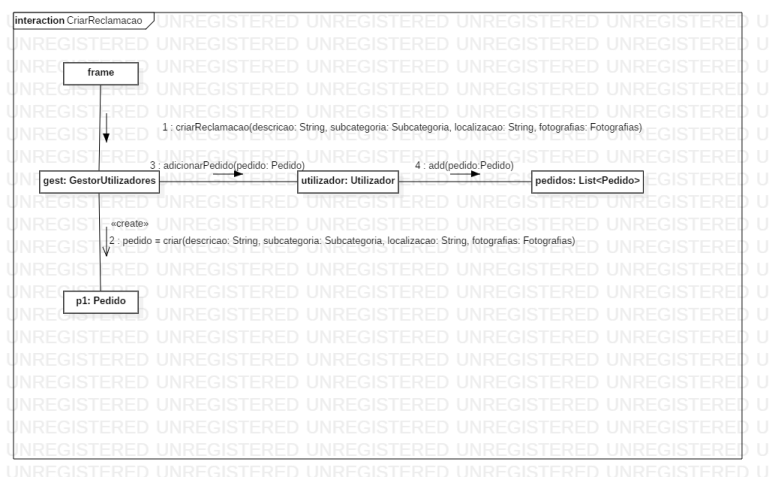
Operação: criarReclamacao (utilizador:Utilizador, descricao: String, subcategoria: Subcategoria, localizacao: String, fotografias: Fotografias)

Referências Cruzadas: Caso de Uso-Registar Reclamação

Pré-condições:

Pós-Condições:

1. Foi criada uma nova instância de Pedido pd.
2. pd foi assossiado com Utilizador através de utilizador.
3. pd.descricao tornou-se em descricao.
4. pd foi associado a Subcategoria através de subcategoria.
5. pd.localizacao tornou-se em localizacao.
6. pd foi associado a Fotografias através de fotografias.



## Contrato cancelarReclamacaoPendente

Operação: cancelarReclamacaoPendente(pedido:Pedido)

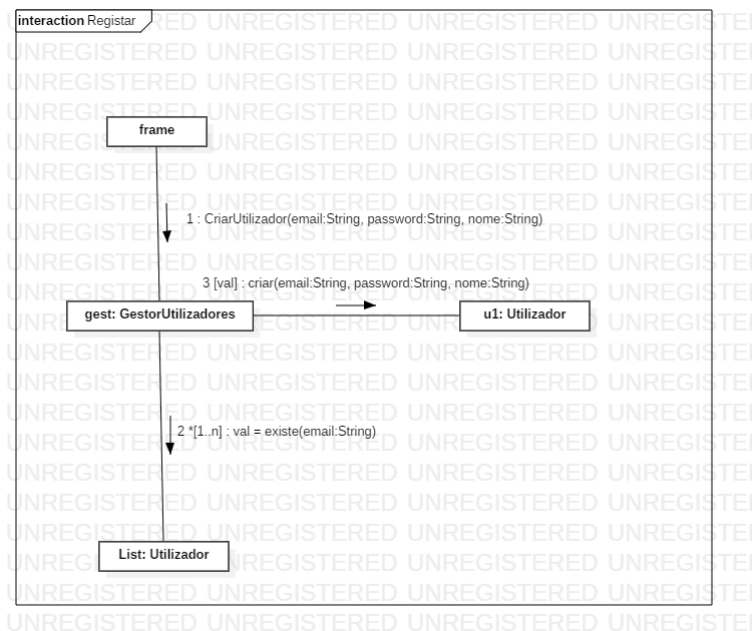
Referências Cruzadas: Caso de uso – Cancelar reclamação pendente



Pré-condições: Pedido ainda não tem associado uma resolução.

Pós-Condições:

1. Foi eliminada a instância de Pedido que corresponde a pedido.



### Contrato aprovarPedido

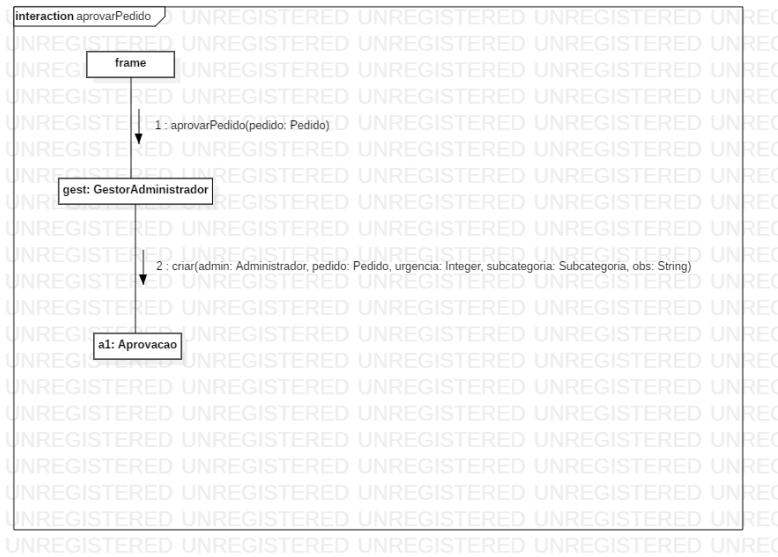
Operação: aprovarPedido (pedido:Pedido, urgencia: Integer, subcategoria: Subcategoria, obs: String)

Referências Cruzadas: Caso de uso – Avaliar reclamação Pendente

Pré-condições: pedido existe no sistema e ainda não esta associado a resolução.

Pós-Condições:

1. Foi instânciado uma nova instância de Aprovacao a.
2. a foi associado com Administrador através de admin.
3. a foi associado com Pedido através de pedido.
4. pedido.subcategoria tornou-se subcategoria.
5. a.urgencia tornou-se em urgência.
6. a.dataAprovacao recebeu o valor da data atual.
7. a.estado tornou-se em 1.
8. a.observacoes tornou-se em obs.



## Contrato invalidarAprovacao

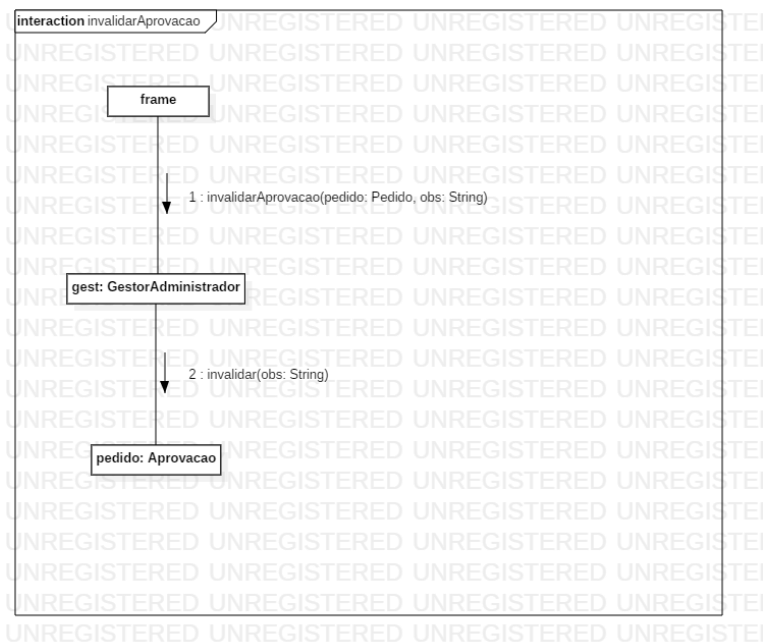
Operação: `invalidarPedido(pedido:Pedido)`

Referências Cruzadas: Caso de uso – Avaliar reclamação Pendente.

Pré-condições: pedido existe no sistema e ainda não esta associado a resolução.

Pós-Condições:

1. Foi instânciado uma nova instância de `Aprovacao` r.
2. a foi associado com `Administrador` através de `admin`.
3. a foi associado com `Pedido` através de `pedido`.
4. `a.dataResolucao` recebeu o valor da data atual.
5. `a.estado` tornou-se em 0.
6. `a.observacoes` tornou-se em `obs`.



### Contrato suspenderUtilizador

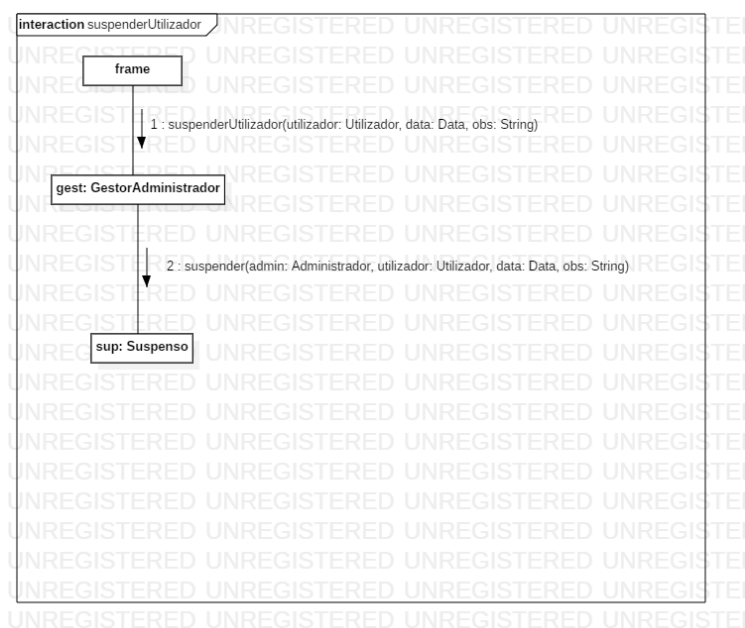
Operação: suspenderUtilizador (utilizador: Utilizador, data: Data, obs: String)

Referências Cruzadas: Caso de uso – Gerir utilizadores da plataforma.

Pré-condições: utilizador e admin existem no dentro do sistema.

Pós-Condições:

1. Uma instância susp de Suspenso foi criada.
2. susp foi associado com um Utilizador utilizador.
3. susp foi associado com Administrador através de admin.
4. susp.data recebeu o valor de data.
5. susp.observacoes tornou-se em obs.



### Contrato banirUtilizador

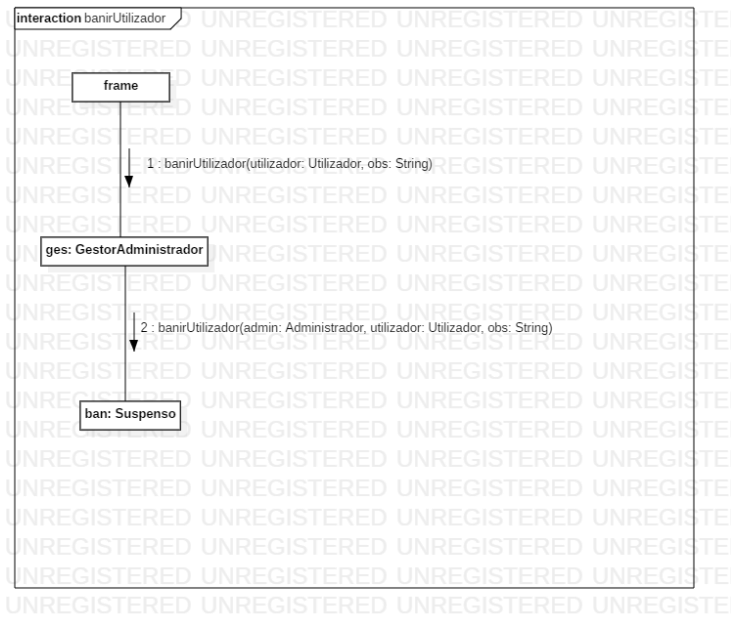
Operação: banirUtilizador (utilizador: Utilizador, obs: String)

Referências Cruzadas: Caso de uso – Gerir utilizadores da plataforma.

Pré-condições:

Pós-Condições:

1. Uma instância susp de Suspenso foi criada.
2. susp foi associado com um Utilizador utilizador.
3. susp foi associado com Administrador através de admin.
4. susp.observacoes tornou-se em obs.



## Contrato reabilitarUtilizador

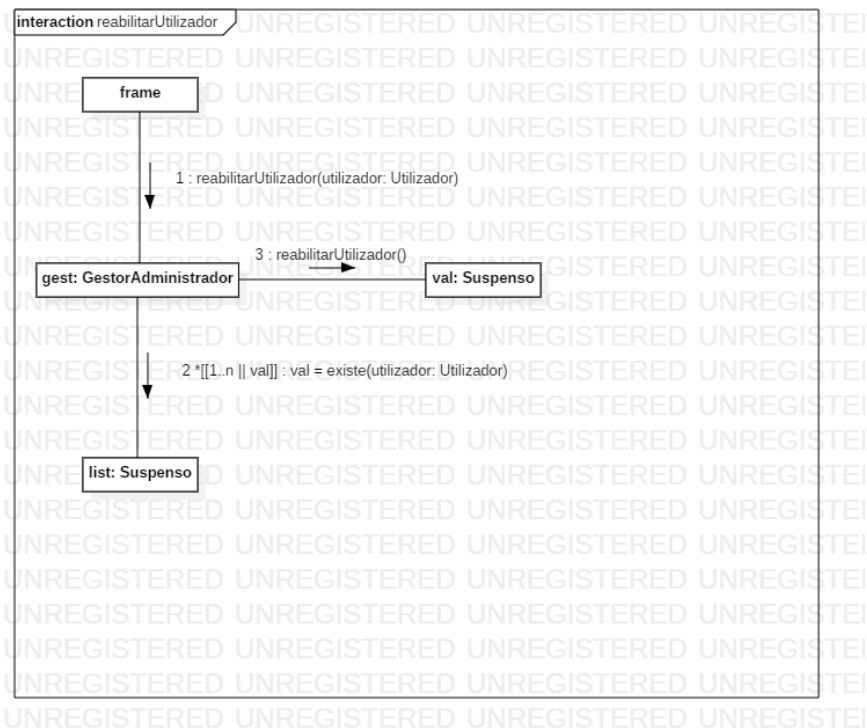
Operação: reabilitarUtilizador (utilizador: Utilizador)

Referências Cruzadas: Caso de uso – Gerir utilizadores da plataforma.

Pré-condições: Existe uma instância de suspenso associado a utilizador.

Pós-Condições:

1. A instância de Suspensao associado com Utilizador utilizador foi eliminada.



### Contrato identificarResolvedor

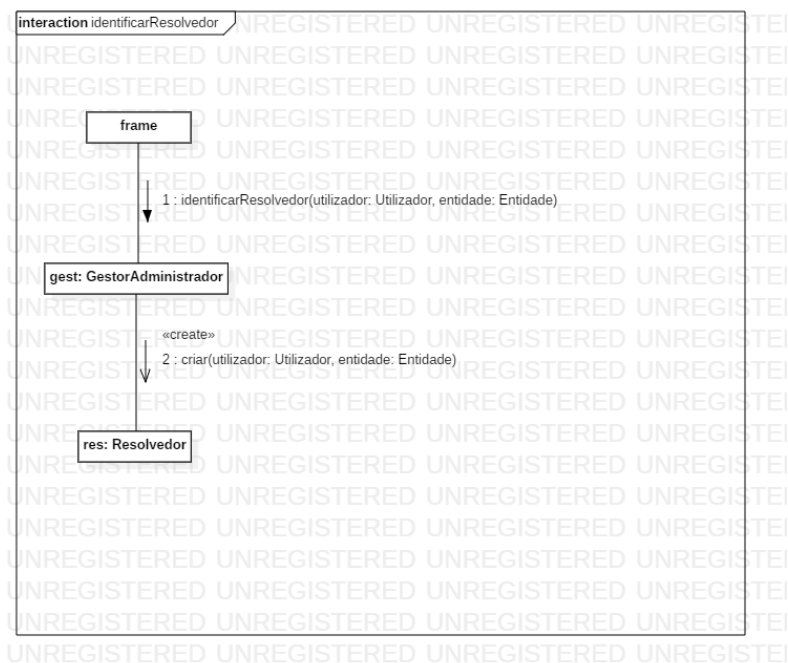
Operação: identificarResolvedor (utilizador: Utilizador, entidade: Entidade)

Referências Cruzadas: Caso de uso – Identificar utilizador como resolvedor

Pré-condições: utilizador e entidade são instâncias existentes das classes Utilizador e Instância respetivamente.

Pós-Condições:

1. utilizador foi transformado num objeto do tipo Resolvedor.
2. utilizador.entidade tornou-se em entidade.



### Contrato desqualificarResolvedor

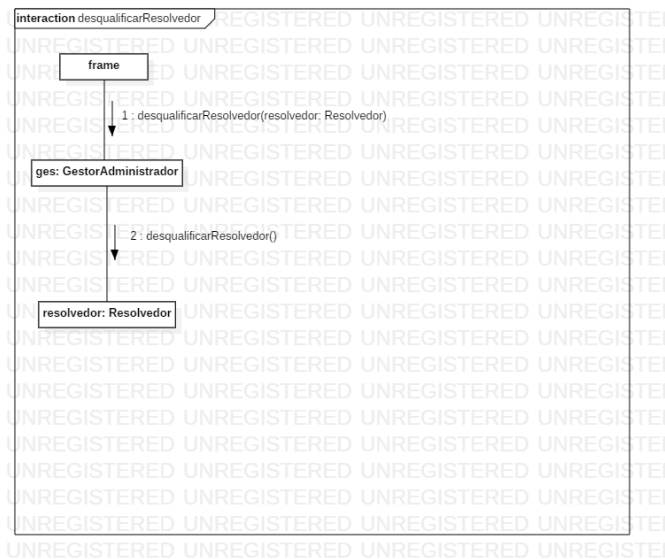
Operação: desqualificarResolvedor (resolvedor: Resolvedor)

Referências Cruzadas: Caso de uso – Identificar utilizador como resolvedor.

Pré-condições: resolvedor é uma instância de Resolvedor.

Pós-Condições:

1. resolvedor foi transformado num objeto do tipo Utilizador.



## Contrato criarEntidade

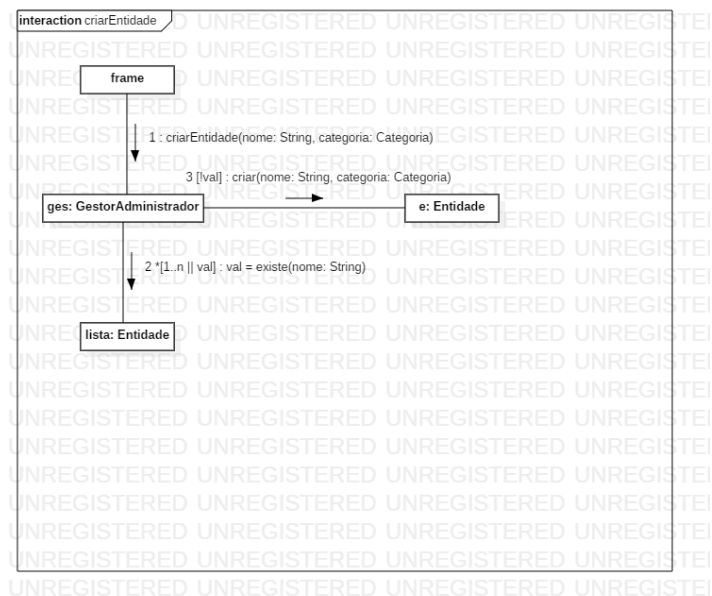
Operação: criarEntidade (nome: String, categoria: Categoria)

Referências Cruzadas: Caso de uso – Criar entidade

Pré-condições: categoria existe como uma instância de Categoria.

Pós-Condições:

1. Uma instância ent de Entidade foi criada.
2. ent foi associado com Categoria através de categoria.
3. ent.nome tornou-se em nome.



## Contrato eliminarEntidade

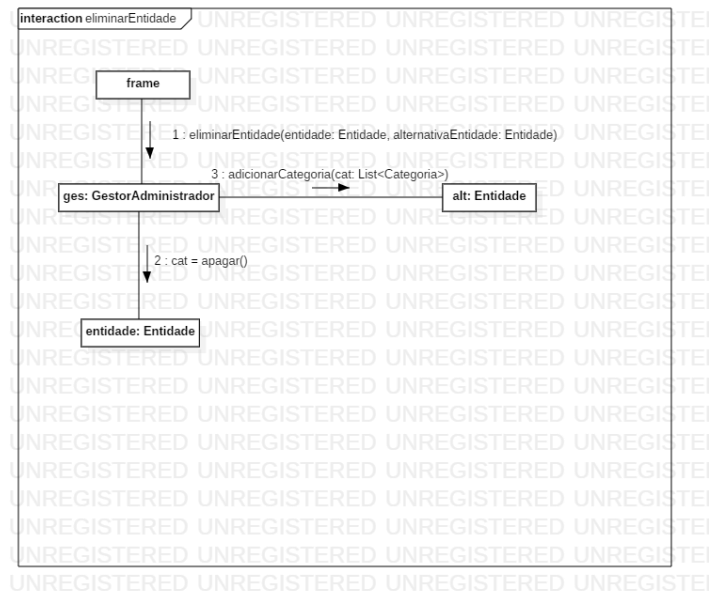
Operação: eliminarEntidade (entidade: Entidade, alternativaEntidade: Entidade)

Referências Cruzadas: Caso de uso – Gerir entidade.

Pré-condições: entidade e alternativaEntidade são instâncias de Entidade.

Pós-Condições:

1. As associações de entidade foram transferidas para alternativaEntidade.
2. A instância entidade foi eliminada.



### Contrato adicionarCategoria

Operação: adicionarCategoria (entidade: Entidade, novaCategoria: Categoria)

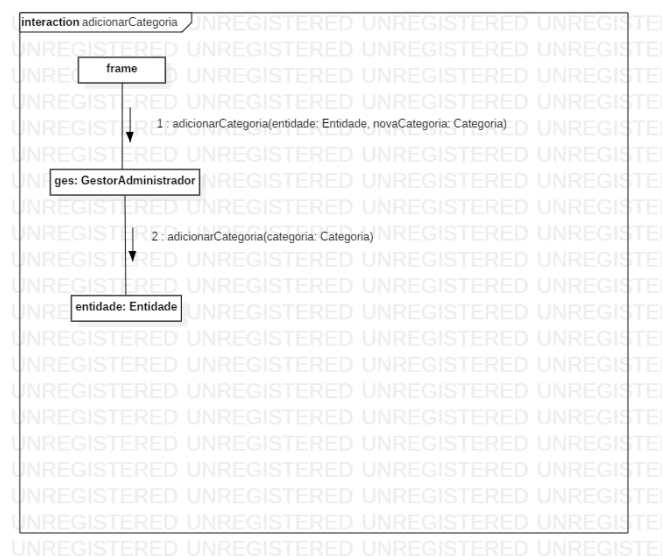
Referências Cruzadas: Caso de uso – Gerir entidade.

Pré-condições:

- entidade e novaCategoria são instâncias de Entidade e Categoria respetivamente.

Pós-Condições:

1. entidadeAlternativa foi associada a Categoria através de entidade.categoria.
2. entidade foi associado a Categoria através de novaCategoria.



### Contrato iniciarResolucao

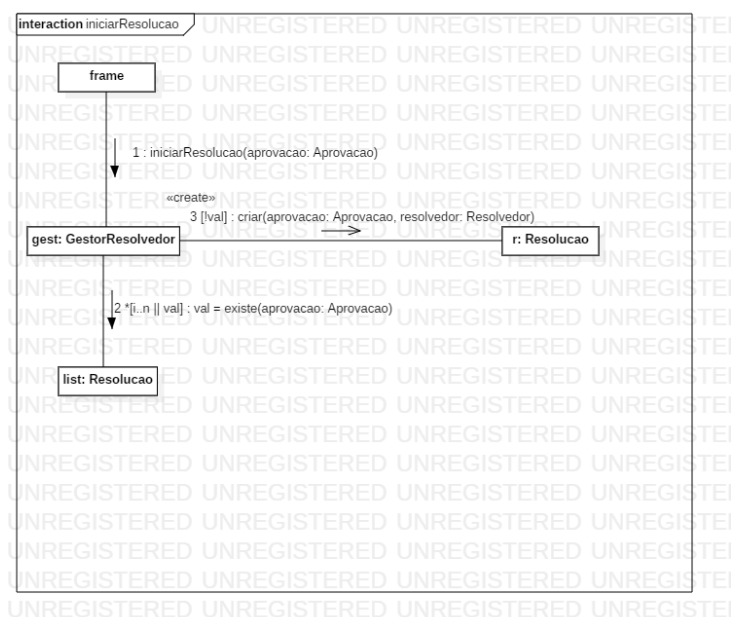
Operação: iniciarResolucao (aprovacao: Aprovacao, resolvedor: Resolvedor)

Referências Cruzadas: Caso de uso – Gerir pedidos de resolução

Pré-condições: aprovacao e resolvedor são instâncias de Aprovacao e Resolvedor respetivamente.

Pós-Condições:

1. Uma instância res de Resolucao foi criada.
2. res foi associado a aprovacao através de aprovacao.
3. res foi associado a Resolvedor através de resolvedor.
4. res.estado assumiu o valor de 1.
5. res.dataInicio assumiu a data atual.



### Contrato cancelarResolucao

Operação: cancelarResolucao (resolucao: Resolucao, obs: String)

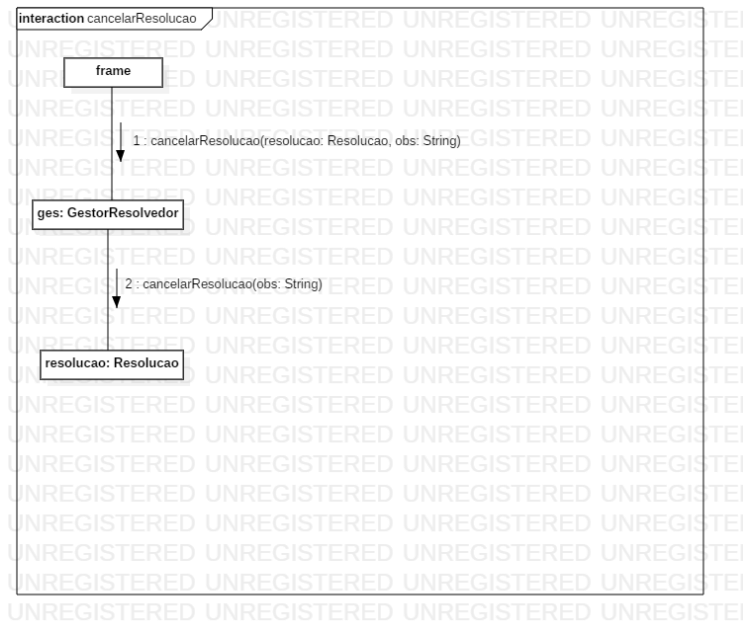
Referências Cruzadas: Caso de uso – Gerir pedidos de resolução

Pré-condições: resolucao é uma instância do tipo Resolucao.

Pós-Condições:

1. resolucao.estado assumiu o valor de 0.
2. resolucao.dataFim assumiu o valor da data atual.
3. resolucao.observacoes tornou-se em obs.





### Contrato alterarResolucao

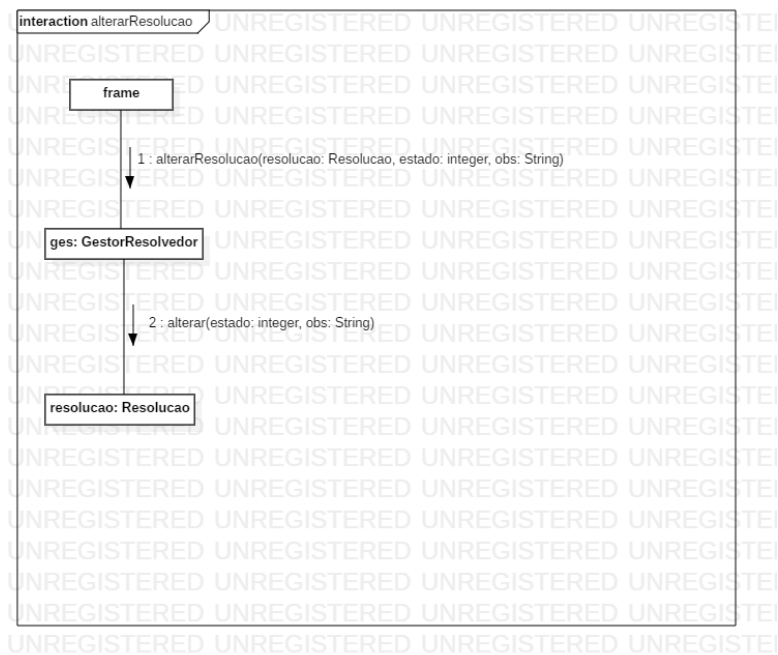
Operação: alterarResolucao (resolucao: Resolucao, estado: integer, obs: String)

Referências Cruzadas: Caso de uso – Gerir pedidos de resolução

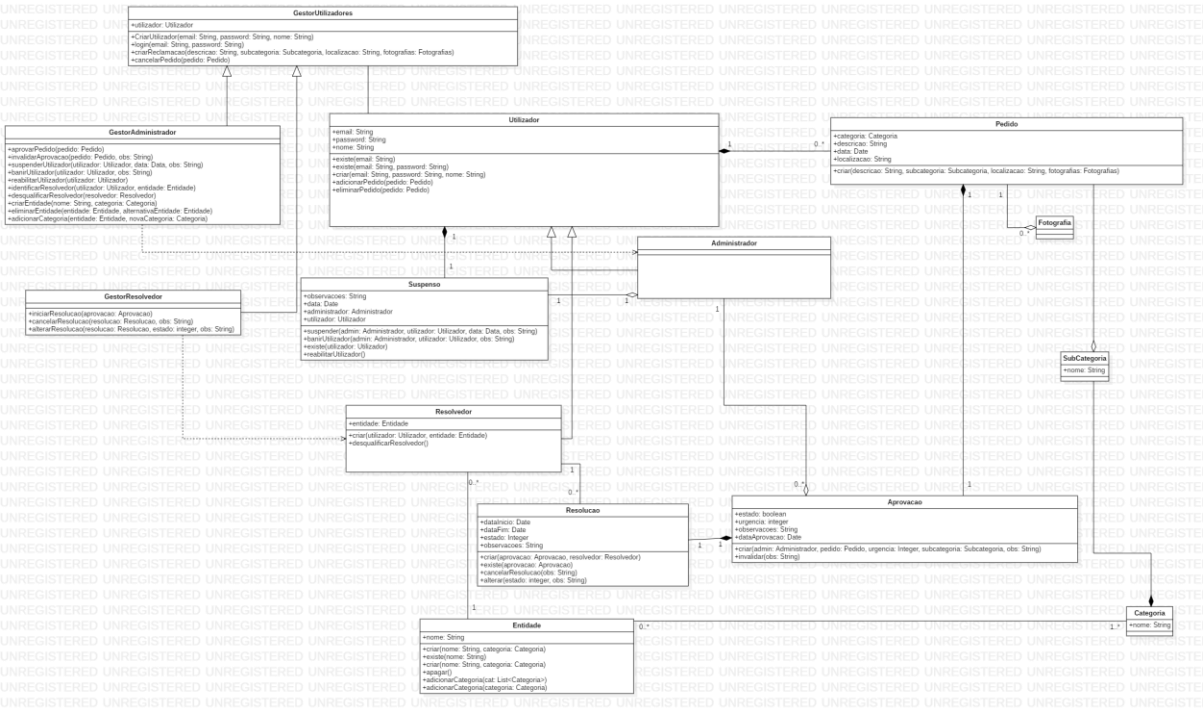
Pré-condições: resolucao é uma instância do tipo Resolucao.

Pós-Condições:

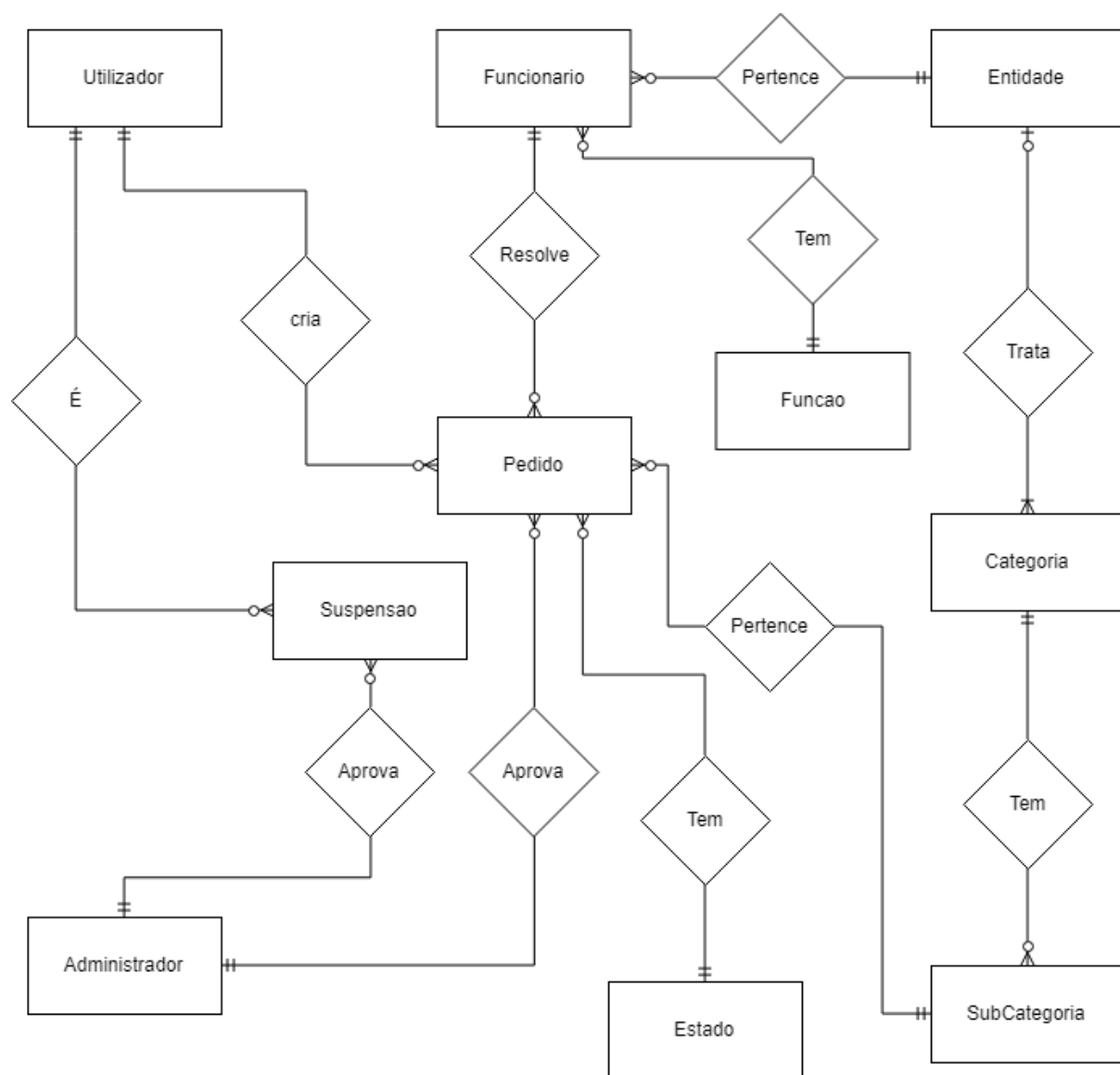
1. resolucao.estado tornou-se em estado.
2. resolucao.observacoes tornou-se em obs.



# Diagrama de Classes Conceptual



## Modelo E-R (Alto Nível)



## Normalização

**Utilizador** (id\_utilizador, email, nome, dt\_nasc, password, username, morada,estado)

**Suspensao** (id\_suspensao, id\_utilizador, id\_administrador, dt\_inicio, dt\_fim, obs)

**Administrador** (id\_administrador, email, nome, password)

**Funcionario** (id\_funcionario, email, nome, password, id\_administrador, id\_entidade,id\_funcao, descricao)

**Pedido** (id\_pedido, id\_utilizador, descricao, id\_estado,descricao\_estado,dt\_criacao, morada, id\_subcategoria, fotos, id\_administrador, dt\_aprovacao, obs\_admin, id\_funcionario, dt\_inicio\_res, obs\_res, dt\_finalizacao)

**Entidade** (id\_entidade, nome, descricao, contactos)

**Categoria** (id\_categoria, nome, id\_entidade)

**Subcategoria** (id\_subcategoria, nome, id\_categoria, descricao)

### 1FN

**Utilizador** (id\_utilizador, email, nome, dt\_nasc, password, username, cod\_postal, rua, n\_porta, estado)

**Suspensao** (id\_suspensao, id\_utilizador, id\_administrador, dt\_inicio, dt\_fim, obs)

**Administrador** (id\_administrador, email, nome, password)

**Funcionario** (id\_funcionario, email, nome, password, id\_administrador, id\_entidade,id\_funcao,descricao)

**Pedido** (id\_pedido, id\_utilizador, descricao, dt\_criacao, cod\_postal, rua, n\_porta, id\_subcategoria, id\_administrador, dt\_aprovacao, obs\_admin, id\_funcionario, dt\_inicio\_res, obs\_res, dt\_finalizacao)  
pedido\_foto(id\_foto ,id\_pedido, foto)

**Entidade** (id\_entidade, nome, descricao)

**Categoria** (id\_categoria, nome, id\_entidade)

**Subcategoria** (id\_subcategoria, nome, id\_categoria, descricao)

**Localidade** (cpostal, localidade)

**Contactos\_Entidade** (id\_contacto, id\_entidade, contacto)

### 3FN

**Utilizador** (id\_utilizador, email, nome, dt\_nasc, password, username, cod\_postal, rua, n\_porta,estado)

**Suspensao** (id\_suspensao, id\_utilizador, id\_administrador, dt\_inicio, dt\_fim, obs)

**Administrador** (id\_administrador, email, nome, password)

**Funcionario** (id\_funcionario, email, nome, password, id\_administrador, id\_entidade, id\_funcao)

Pedido (id\_pedido, id\_utilizador, descricao, id\_estado, dt\_criacao, cod\_postal, rua, n\_porta, id\_subcategoria, id\_administrador, dt\_aprovacao, obs\_admin, id\_resolverdor, dt\_inicio\_res, obs\_res, dt\_finalizacao)

**Funcao** (id\_funcao, descricao)

**Pedido\_foto** (id\_foto, id\_pedido, foto)

**Entidade** (id\_entidade, nome, descricao)

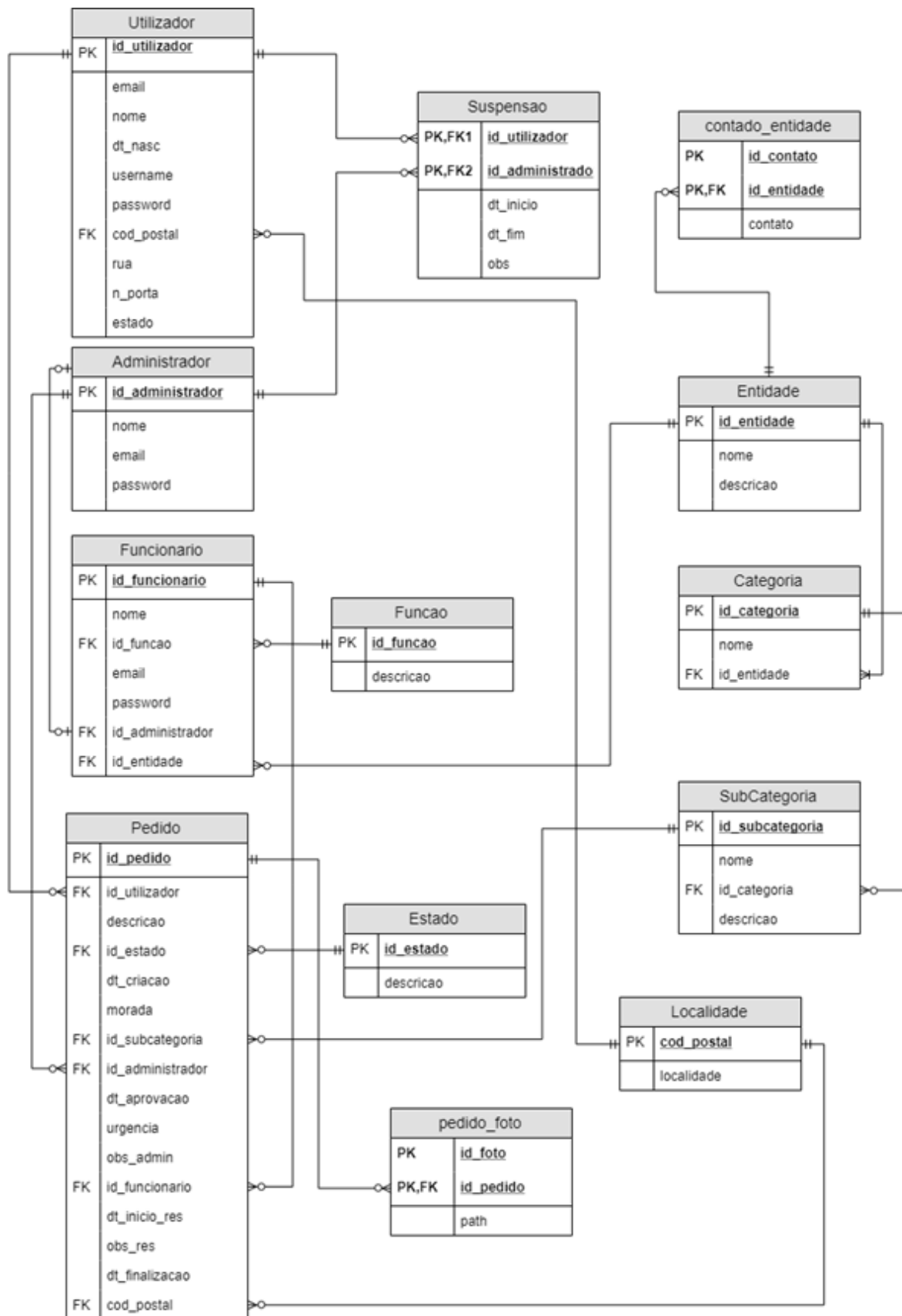
**Categoria** (id\_categoria, nome, id\_entidade)

**Subcategoria** (id\_subcategoria, nome, id\_categoria, descricao)

**Localidade** (cpostal, localidade)

**Contactos\_Entidade** (id\_contacto, id\_entidade, contacto)

## Modelo E-R - Normalizado



## Modelo Relacional – Código SQL - DDL

```
CREATE TABLE localidade (  
    cod_postal NUMBER,  
    localidade VARCHAR2(30) NOT NULL,  
    CONSTRAINT pk_cp PRIMARY KEY (cod_postal)  
);
```

```
CREATE TABLE utilizador (  
    id_utilizador NUMBER,  
    estado VARCHAR2(20),  
    email VARCHAR2(254) NOT NULL,  
    dt_nasc DATE,  
    password VARCHAR (30) NOT NULL,  
    nome_util VARCHAR2(20) NOT NULL,  
    cod_postal NUMBER,  
    rua VARCHAR2(200),  
    n_porta NUMBER,  
    CONSTRAINT constraint_email UNIQUE (email),  
    CONSTRAINT constraint_uti UNIQUE (nome_util),  
    CONSTRAINT pk_id PRIMARY KEY (id_utilizador),  
    CONSTRAINT fk_uti_loc FOREIGN KEY (cod_postal) REFERENCES localidade(cod_postal)  
);
```

```
CREATE TABLE entidade (  
    id_entidade NUMBER,  
    nome VARCHAR2(30) NOT NULL,  
    descricao VARCHAR2(255),  
    CONSTRAINT pk_ent PRIMARY KEY (id_entidade)  
);
```

```
CREATE TABLE categoria (  
    id_categoria NUMBER,  
    nome VARCHAR2(30) NOT NULL,  
    id_entidade NUMBER,  
    CONSTRAINT pk_cat PRIMARY KEY (id_categoria),  
    CONSTRAINT fk_cat_ent FOREIGN KEY (id_entidade) REFERENCES entidade(id_entidade)  
);
```

```
CREATE TABLE subcategoria (  
    id_subcategoria NUMBER,  
    nome VARCHAR2(30) NOT NULL,  
    id_categoria NOT NULL,  
    descricao VARCHAR2(254),  
    CONSTRAINT pk_subcat PRIMARY KEY (id_subcategoria),  
    CONSTRAINT fk_subcat_cat FOREIGN KEY (id_categoria) REFERENCES categoria(id_categoria)  
);
```

```
CREATE TABLE funcao(  
    id_funcao NUMBER,  
    descricao VARCHAR2(25),  
    CONSTRAINT pk_funcao PRIMARY KEY (id_funcao)  
);
```

```
CREATE TABLE funcionario (  
    id_funcionario NUMBER,  
    email VARCHAR2(254) NOT NULL,  
    nome VARCHAR2(30) NOT NULL,  
    password VARCHAR2(30) NOT NULL,  
    id_administrador NUMBER,  
    id_entidade NUMBER,
```



```

id_funcao NUMBER,

CONSTRAINT unique_email_func UNIQUE (email),

CONSTRAINT pk_func PRIMARY KEY (id_funcionario),

CONSTRAINT fk_func_admin FOREIGN KEY (id_administrador) REFERENCES
funcionario(id_funcionario),

CONSTRAINT fk_func_ent FOREIGN KEY (id_entidade) REFERENCES entidade(id_entidade),

CONSTRAINT fk_func_func FOREIGN KEY (id_funcao) REFERENCES funcao(id_funcao)

);

```

```

CREATE TABLE estado (

id_estado NUMBER,

descricao VARCHAR2(25),

CONSTRAINT pk_estado PRIMARY KEY (id_estado)

);

```

```

CREATE TABLE suspensao(

id_suspensao NUMBER,

id_utilizador NUMBER,

id_administrador NUMBER,

dt_inicio DATE NOT NULL,

dt_fim DATE,

obs VARCHAR2(255),

CONSTRAINT pk_susp PRIMARY KEY (id_suspensao),

CONSTRAINT fk_susp_utl FOREIGN KEY (id_utilizador) REFERENCES utilizador(id_utilizador),

CONSTRAINT fk_susp_admin FOREIGN KEY (id_administrador) REFERENCES
funcionario(id_funcionario)

);

```

```

CREATE TABLE contato_entidade(

id_contato NUMBER,

id_entidade NUMBER,

contato NUMBER NOT NULL,

```

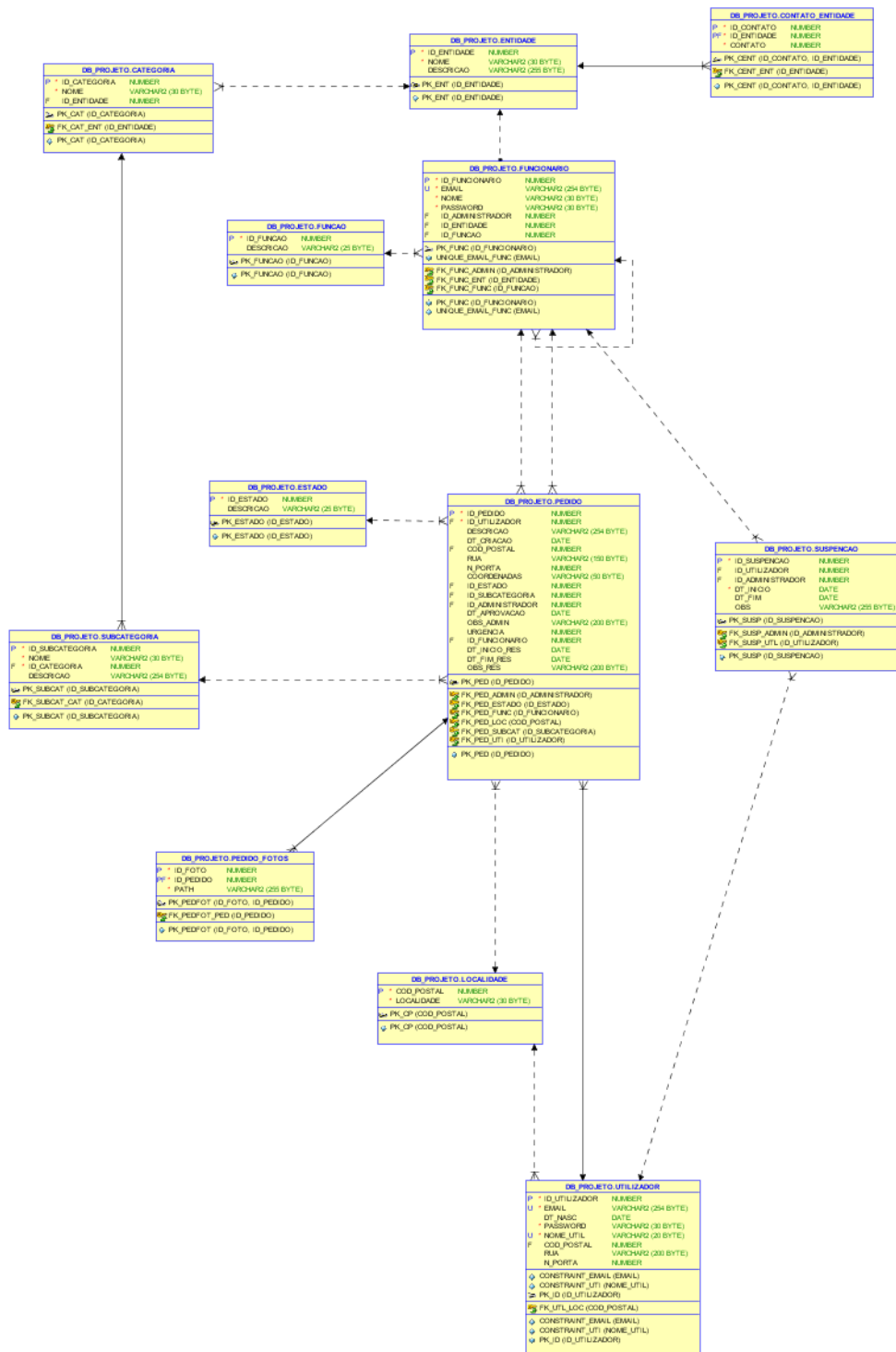
```
CONSTRAINT pk_cent PRIMARY KEY (id_contato, id_entidade),  
CONSTRAINT fk_cent_ent FOREIGN KEY (id_entidade) REFERENCES entidade(id_entidade)  
);
```

```
CREATE TABLE pedido (  
    id_pedido NUMBER,  
    id_utilizador NUMBER NOT NULL,  
    descricao VARCHAR2(254),  
    dt_criacao DATE,  
    cod_postal NUMBER,  
    rua VARCHAR2(150),  
    n_porta NUMBER,  
    coordenadas VARCHAR2(50),  
    id_estado NUMBER,  
    id_subcategoria NUMBER,  
    id_administrador NUMBER,  
    dt_aprovacao DATE,  
    obs_admin VARCHAR2(200),  
    urgencia NUMBER,  
    id_funcionario NUMBER,  
    dt_inicio_res DATE,  
    dt_fim_res DATE,  
    obs_res VARCHAR2(200),  
    CONSTRAINT pk_ped PRIMARY KEY (id_pedido),  
    CONSTRAINT fk_ped_uti FOREIGN KEY (id_utilizador) REFERENCES utilizador(id_utilizador),  
    CONSTRAINT fk_ped_admin FOREIGN KEY (id_administrador) REFERENCES  
funcionario(id_funcionario),  
    CONSTRAINT fk_ped_func FOREIGN KEY (id_funcionario) REFERENCES funcionario(id_funcionario),  
    CONSTRAINT fk_ped_subcat FOREIGN KEY (id_subcategoria) REFERENCES  
subcategoria(id_subcategoria),  
    CONSTRAINT fk_ped_estado FOREIGN KEY (id_estado) REFERENCES estado(id_estado),
```

```
CONSTRAINT fk_ped_loc FOREIGN KEY (cod_postal) REFERENCES localidade(cod_postal)
);
```

```
CREATE TABLE pedido_fotos(
    id_foto NUMBER,
    id_pedido NUMBER,
    path VARCHAR2(255) NOT NULL,
    CONSTRAINT pk_pedfot PRIMARY KEY (id_foto, id_pedido),
    CONSTRAINT fk_pedfot_ped FOREIGN KEY (id_pedido) REFERENCES pedido(id_pedido)
);
```

## Modelo Relacional Gerado pela Aplicação



## Views

### Pedidos atribuídos a entidade

```
CREATE VIEW entidade_pedidos(id_entidade, nome, descricao_entidade, descricao_pedido,
dt_criacao, rua, cod_postal, localizacao, subcategoria, categoria, dt_inicio_res, dt_finalizacao,
estado) AS SELECT e.id_entidade, e.nome, e.descricao, p.descricao, p.dt_criacao, p.rua,
p.cod_postal, l.localidade, sc.nome, c.nome, p.dt_inicio_res, p.dt_fim_res, est.descricao
FROM entidade e
INNER JOIN categoria c ON (c.id_entidade = e.id_entidade)
INNER JOIN subcategoria sc ON (sc.id_categoria = c.id_categoria)
INNER JOIN pedido p ON (p.id_subcategoria = sc.id_subcategoria)
INNER JOIN localidade l ON (p.cod_postal = l.cod_postal)
INNER JOIN estado est ON (est.id_estado = p.id_estado);
```

### Histórico de resoluções dos funcionários

```
CREATE VIEW historico_funcionario(id_funcionario, nome, descricao, descricao_pedido,
dt_inicio_res, dt_finalizacao, estado, localizacao, subcategoria, categoria) AS Select f.id_funcionario,
f.nome, fun.descricao, p.descricao, p.dt_inicio_res, p.dt_fim_res, est.descricao estado, l.localidade,
sc.nome, c.nome funcionario
FROM funcionario f
INNER JOIN pedido p ON (p.id_funcionario = f.id_funcionario)
INNER JOIN estado est ON (est.id_estado = p.id_estado)
INNER JOIN funcao fun ON (fun.id_funcao = f.id_funcao)
INNER JOIN entidade ent ON (ent.id_entidade = f.id_entidade)
INNER JOIN localidade l ON (p.cod_postal = l.cod_postal)
INNER JOIN subcategoria sc ON (sc.id_subcategoria = p.id_subcategoria)
INNER JOIN categoria c ON (sc.id_categoria = c.id_categoria);
```

## TRIGGERS

### Atualizar estado do utilizador

```
CREATE TRIGGER atualiza_estado  
  
AFTER INSERT OR UPDATE on suspensao  
  
FOR EACH ROW  
  
BEGIN  
  
UPDATE utilizador SET  
  
    estado='Suspendido'  
  
WHERE id_utilizador=: NEW.id_utilizador;  
  
end;
```

## Procedures

### Suspender utilizador

```
CREATE OR REPLACE PROCEDURE DesuspenderUtilizador(id_user number)  
  
    IS estado_atual VARCHAR2(20);  
  
    Data_fim date;  
  
BEGIN  
  
    SELECT estado INTO estado_atual FROM utilizador WHERE id_utilizador = id_user;  
  
    SELECT DT_FIM INTO Data_fim FROM suspensao WHERE id_utilizador = id_user;  
  
    IF estado_atual='Suspendido' AND TO_DATE(SYSDATE,'DD-MM-YYYY') > TO_DATE(Data_fim,'DD-MM-YYYY') THEN  
  
        UPDATE utilizador SET estado='Disponivel' WHERE id_utilizador=id_user;  
  
    ELSE  
  
        UPDATE utilizador SET estado=estado WHERE id_utilizador=id_user;  
  
    END IF;  
  
END;
```

## Queries Sobre os Caso de uso

### Ator: Utilizador Não Registrado

#### *Caso de uso: Registrar utilizador*

```
INSERT INTO localidade(cod_postal,localidade) VALUES (4900,'Viana do Castelo');

INSERT INTO utilizador (id_utilizador,
estado,email,dt_nasc,password,nome_util,cod_postal,rua,n_porta) VALUES
(1,'Disponivel','banana@gmail.com',TO_DATE('16-02-1997','DD-MM-
YYYY'),'1234','Gabriel',4900,'Rua das bananeiras',666);
```

### Ator: Utilizador

#### *Caso de uso: Registar Reclamação*

```
INSERT INTO estado VALUES (1,'Por Resolver');

INSERT INTO entidade VALUES (1, 'Ruas é conosco', 'Tudo a haver com ruas');

INSERT INTO Categoria VALUES(1,'Ruas',1).

INSERT INTO subcategoria VALUES(1,'Esgotos',1,'Esgotos');

INSERT INTO funcao VALUES(1,'Teste');

INSERT INTO funcionario VALUES (1, 'blabl@BLUMPACLAM.RU','Gabriel',1234,1,1,1);

INSERT INTO pedido
(id_pedido,id_utilizador,descricao,dt_criacao,cod_postal,rua,n_porta,id_estado,id_subcategoria,id_
administrador)

VALUES (1,1,'Tampa de esgoto está fora do sitio',TO_DATE('04-12-2018','DD-MM-YYYY'),4900,'Rua
das Cabritas',666,1,1,1);
```

#### *Caso de uso: Ver Histórico de reclamações*

```
SELECT id_pedido, descricao, dt_criacao, cod_postal, rua, n_porta, CASE WHEN dt_aprovacao is
not NULL then 'APROVADO' else 'NAO APROVADO' END estado

FROM pedido where id_utilizador=1;
```

#### *Caso de uso: Login*

```
Select count(*) as sucesso from utilizador

where nome_util='Gabriel' AND password=1234;
```

*Caso de uso: Cancelar uma reclamação pendente*

```
DELETE from pedido where id_pedido=1 AND dt_aprovacao IS NULL;
```

Ator: Administrador

*Caso de uso: Ver reclamações pendentes*

```
Select p.id_pedido , p.descricao, p.dt_criacao, p.cod_postal, p.rua,  
p.n_porta,u.nome_util,c.nome,s.nome,l.localidade from pedido p
```

```
inner join localidade l on (p.cod_postal=l.cod_postal)
```

```
inner join subcategoria s on (p.id_subcategoria=s.id_subcategoria)
```

```
inner join categoria c on (s.id_categoria=c.id_categoria)
```

```
inner join utilizador u on (u.id_utilizador=p.id_utilizador)
```

```
where p.dt_aprovacao IS NULL;
```

*Caso de uso: Avaliar reclamações Pendentes*

```
update pedido set dt_aprovacao = to_date('18-12-2018','DD-MM-YYYY'), obs_admin='Nada a  
apontar',urgencia=5
```

```
where id_pedido=1 AND dt_aprovacao is NULL;
```

*Caso de uso: Finalizar Processo de Reclamação*

```
update pedido set id_funcionario=1,dt_inicio_res = to_date('18-12-2018','DD-MM-YYYY'),  
dt_fim_res = to_date('18-12-2018','DD-MM-YYYY'),obs_res='TA TUDO EM ORDEM CHEFE'
```

```
where id_pedido=1;
```

*Caso de uso: Criar entidade*

```
insert into entidade VALUES (2,'Fast VRUM','Limpa Ruas');
```



### *Caso de uso: Gerir utilizadores da plataforma*

#### **Suspender Utilizador**

```
Insert into suspensao(id_suspensao,id_utilizador, id_administrador, dt_inicio, dt_fim, obs)
values(1,1, 1, TO_DATE('19-01-2019', 'dd-mm-yyyy'), TO_DATE('19-01-2020', 'dd-mm-yyyy'), 'Uso
indeviduo da aplicação');
```

#### **Banir Utilizador**

```
Insert into suspensao (id_suspensao,id_utilizador, id_administrador, dt_inicio, obs) values(2,1,
1, TO_DATE('19-01-2019', 'dd-mm-yyyy'), 'Uso indeviduo da aplicação');
```

#### **Reabilitar Utilizador**

```
Delete from suspensao where id_utilizador = 1;
```

#### **Alterar data de fim da suspensão**

```
Update suspensao set dt_fim = TO_DATE ('10-09-2019', 'DD-MM-YYYY') where id_utilizador = 1;
```

### **Caso de uso: Ver utilizadores**

```
Select u.*, l.localidade from utilizador u
inner join localidade l on u.cod_postal = l.cod_postal;
```

### *Caso de uso: Ver histórico de resoluções*

#### **Histórico geral**

```
Select f.id_funcionario, f.nome, fun.descricao, p.descricao, p.dt_inicio_res, p.dt_fim_res,
est.descricao estado, l.localidade, sc.nome, c.nome funcionario
```

```
FROM funcionario f
```

```
inner join pedido p on (p.id_funcionario = f.id_funcionario)
```

```
inner join estado est on (est.id_estado = p.id_estado)
```

```
inner join funcao fun on (fun.id_funcao = f.id_funcao)
```

```
inner join entidade ent on (ent.id_entidade = f.id_entidade)
```

```
inner join localidade l on (p.cod_postal = l.cod_postal)
```

```
inner join subcategoria sc on (sc.id_subcategoria = p.id_subcategoria)
```

```
inner join categoria c on (sc.id_categoria = c.id_categoria);
```

#### Histórico de um funcionário

```
Select f.id_funcionario, f.nome, fun.descricao, p.descricao, p.dt_inicio_res, p.dt_fim_res,  
e.descricao estado, l.localidade, sc.nome, c.nome funcionario
```

```
FROM funcionario f
```

```
inner join pedido p on (p.id_funcionario = f.id_funcionario)
```

```
inner join estado e on (e.id_estado = p.id_estado)
```

```
inner join funcao fun on (fun.id_funcao = f.id_funcao)
```

```
inner join entidade ent on (ent.id_entidade = f.id_entidade)
```

```
inner join localidade l on (p.cod_postal = l.cod_postal)
```

```
inner join subcategoria sc on (sc.id_subcategoria = p.id_subcategoria)
```

```
inner join categoria c on (sc.id_categoria = c.id_categoria)
```

```
where f.id_funcionario = 3;
```

#### *Caso de uso: Gerir entidade*

##### Eliminar Entidade

```
update funcionario set id_entidade = 2 where id_entidade = 1;
```

```
update categoria set id_entidade = 2 where id_entidade = 1;
```

```
delete from entidade where id_entidade = 1;
```

##### Criar nova entidade

```
insert into entidade (id_entidade, nome, descricao) values(3, 'Canil', 'Canil da cidade');
```

##### Atualizar dados da entidade

```
update entidade set nome='Canil', descricao='Entidade relativa ao canil da cidade' where  
id_entidade = 3;
```

##### Atribuir nova categoria a entidade

```
insert into categoria (id_categoria, nome, id_entidade) values(2, 'Animais', 3);
```

##### Atribuir categoria existente a entidade

```
update categoria set id_entidade = 2 where id_categoria = 1;
```

*Adicionar nova subcategoria a categoria*

```
insert into subcategoria (id_subcategoria, nome, id_categoria, descricao) values(2, 'Animal solto', 1, 'Animal solto que represente um problema para o bem esta dos cidadaos');
```

*Alterar categoria de subcategoria*

```
update subcategoria set id_categoria = 2 where id_subcategoria = 1;
```

## Conclusão

Com este trabalho tivemos a oportunidade de aplicar os conceitos adquiridos na unidade curricular de base de dados e Engenharia de Software, de forma a termos uma experiência de como resolver da melhor forma os problemas em contexto de vida real.

Foi através do processo de realização do trabalho que conseguimos tirar todas as nossas dúvidas e aprofundar ainda mais os nossos conhecimentos sobre os temas.

Achamos que oferecemos a melhor solução ao problema proposto, completamente adaptadas as necessidades do consumidor, poderíamos ter complicado mais mas chegamos á conclusão que muitas vezes o mais correcto é o mais simples, desde que consigamos atender ás necessidades do cliente.

Demos bastantes voltas, criamos mais de 4 diagramas diferentes, cada um mais simples que o próximo e acreditamos que ao fazermos desta maneira conseguimos amadurecer bastante quanto á projeção e eficiência tanto de uma base de dados como na realização dos diagramas em UML , tendo também melhorado bastante como profissionais em questões de decisão em situações de trabalho de grupo e estando assim melhor preparados para situações de vida real.

Assim, damos por concluído o nosso trabalho.