FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Gestão Documental Segura de Projectos de Urbanização e Edificação em Municípios

Jesus António Faria Campos

Licenciado em Sistemas de Informação para a Gestão pelo
Instituto Politécnico do Cávado e do Ave

Dissertação submetida para satisfação parcial
dos requisitos do grau de
Mestre em Engenharia Informática

Dissertação realizada sob a supervisão de
Professor Doutor José Manuel Magalhães Cruz
do Departamento de Engenharia Informática
da Faculdade de Engenharia da Universidade do Porto
e de Professor Doutor Paulo Alexandre Teves da Silva
da Faculdade de Arquitectura e Artes
da Universidade Lusíada de Famalicão

Porto, Março de 2009

Resumo

A Administração Pública tem como principal objectivo servir bem o cidadão, rapidamente e ao menor custo possível. Com este propósito, o regime de urbanização e edificação sofreu várias alterações nos seus procedimentos introduzindo, entre outras medidas, novas tecnologias no processo de apreciação de Projectos de Arquitectura pelos Municípios portugueses, que implicam novas formas de relacionamento entre as diversas entidades envolvidas. Isto traduz-se numa oportunidade para a melhoria dos serviços prestados pela Administração Pública Local nesta área.

Esta dissertação assenta numa proposta de implementação de uma solução informática aberta para a gestão documental segura de projectos de urbanização e edificação em Municípios, permitindo assim a desmaterialização "de todo o processo clássico". A prova de conceito concretizou-se num protótipo baseado em serviços web e ferramentas livres.

Numa primeira fase deste documento, apresentam-se os conceitos basilares relativos a projectos de arquitectura em Portugal, especificamente na sua versão electrónica, e à tecnologia indispensável à sua utilização segura. Numa segunda fase, especificam-se os requisitos do sistema pretendido e desenha-se a proposta de solução. Numa terceira fase, analisam-se as opções tecnológicas e a implementação do protótipo. Por fim, apresentam-se os resultados obtidos, que claramente mostram a viabilidade do sistema.

Abstract

The Public Administration's main goal is to well serve the citizens, in the shortest possible time and at the lowest possible cost. Aiming this, the urbanization and edification laws suffered several changes in its procedures introducing, among other measures, new technologies into the appreciation of Architecture Projects by Portuguese town councils, which imply new ways of interaction between the different entities involved in the process. This is an opportunity to improve the services provided by the Local Administration in this area.

This dissertation is based on a proposal to implement a secure documental management open solution for urbanization and edification projects in town councils, thus allowing the dematerialization of the classic process. This was accomplished with a prototype based on web services and developed using open-source tools.

On the first phase of this document, the fundamental concepts regarding the architecture projects in Portugal are presented, specifically in its electronic version, and using tools that assure a secure use of them. On the second phase, the desired system requirements are specified and the solution proposal is drawn. On the third phase, the technological options and prototype implementation are analysed. And last, the obtained results are presented, which clearly demonstrate the system's viability.

Agradecimentos

Aos meus orientadores, aos Professores J. M. Cruz e P. Silva pelo incentivo e motivação, pelas críticas e sugestões e pela contínua disponibilidade.

Ao meu Pai, às minhas Irmãs, à Vânia, ao Rubem, à Ana e aos demais amigos pelo apoio e contribuição.

Índice Geral

Lista de Figuras

Lista de Tabelas

Lista dos Acrónicos Utilizados

CAD – *Computer Aided Design*

CC – Cartão de Cidadão

CCDR - Comissões Coordenadoras de Desenvolvimento Regional

DGAL - Direcção-Geral da Administração Local

DVD-ROM - *Digital Versatile/Video Disc – Read Only Memory*

DWF - Design Web Format

DWG - AutoCAD Drawing Database

DXG - Drawing Exchange Format

FTP - File Transfer Protocol

HTTP - *Hypertext Transfer Protocol*

HTTPS - *Hypertext Transfer Protocol Secure*

IETF - Internet Engineering Task Force

ISO - *International Organization for Standardization*

JPEG – Joint Photographic Experts Group

MVC – *Model View Controller*

OCSP - *Online Certificate Status Protocol*

PDF – Portable Document Format

PKCS - *Public Key Cryptography Standards*

PKI – Public Key Infrastructure

POO – Programação Orientada a Objectos

RJUE – Regime Jurídico de Urbanização e Edificação

SQL – Structured Query Language

SSL - *Secure Sockets Layer*

TCP/IP – Transmission Control Protocol/Internet Protocol

TIF – Tagged Image File

TLS – Transport Layer Security

UML – Unified Modeling Language

1 Introdução

Este capítulo aborda as principais motivações e objectivos para a realização do presente trabalho (Secção 1.1) e apresenta a estrutura e organização da dissertação (Secção 1.2).

1.1 Motivação e objectivos

Os sistemas de gestão documental modernos visam gerir e disponibilizar informação, não só de processos e informação estruturada, mas também de outro tipo de informação importante que, por não estar estruturada, ficam de fora em sistemas mais obsoletos. Exemplo desse tipo de informação é associada a documentos associados.

O elevado número de tipo de processos, de documentos e tipos de documentos, de pessoas intervenientes nos processos e a alteração da própria lei regulamentadora implica que, na gestão de projectos de arquitectura, seja necessário implementar um sistema de gestão documental com as novas especificações ou expandir um já existente.

A introdução de tal sistema permite o apoio e coordenação das pessoas intervenientes na execução das suas tarefas diárias, além de permitir gerir toda a informação que, pelas suas características evolutivas, estão em constante mudança. No entanto, há implicações, não só do ponto de vista tecnológico mas também do ponto de vista organizacional, que obrigam a um estudo do enquadramento legal, do levantamento

do circuito documental e do levantamento de requisitos no que respeita à interacção com outros sistemas já existentes na organização.

À data de entrega da proposta para esta dissertação, as soluções existentes, tanto proprietárias como desenvolvidas internamente pelos municípios, eram carentes relativamente a estas necessidades.

O objectivo principal deste trabalho é o de propor uma implementação de uma gestão documental segura de projectos de urbanização e edificação em Municípios, que vá de encontro às alterações da lei vigente, utilizando ferramentas livres. Assim, deve ser levado em consideração que a ferramenta resultante deve basear-se em serviços Web e deve ser livre, pelo que é essencial a utilização de tecnologia "open-source". Deve ser da confiança dos utilizadores, ou seja, deve preocupar-se com as questões de segurança. Em concreto, deve suportar um sistema de autenticação forte, garantindo a autenticidade dos utilizadores e a integridade e não-repúdio dos documentos submetidos. Deve ser considerado também a utilização dos certificados do Cartão de Cidadão de forma a não acarretar custos de aquisição de certificados qualificados de segurança.

1.2 Organização da Dissertação

O capítulo 2 começa por definir o projecto de arquitectura e a sua composição. Depois é feita uma enunciação do E-Gov na Europa e em Portugal seguido da sua definição, bem como de E-Local Government. De seguida, é feito um balanço das alterações mais significativas do regime jurídico de urbanização e edificação para os sistemas de informação, é exposto o cenário actual do licenciamento electrónico em Portugal e é descrita a problemática da diversidade no formato electrónico da informação.

O capítulo 3 analisa os conceitos basilares de segurança importantes e utilizados ao longo do desenvolvimento do projecto, incluindo a segurança no Cartão de Cidadão.

O capítulo 4 apresenta a abordagem metodológica do sistema e a especificação de requisitos.

O capítulo 5 explica a análise e desenho da solução, decompondo os requisitos em casos de uso, em diagramas de actividade e estados e propõe um conjunto de classes que se considera adequadas ao sistema.

O capítulo 6 justifica as tecnologias utilizadas, apresenta o modelo físico de dados, o diagrama de instalação, a organização do código fonte e conclui com a avaliação experimental do sistema.

O capítulo 7 faz uma avaliação do trabalho desenvolvido, apresenta as principais conclusões do trabalho realizado e dá orientações para trabalhos futuros.

2 Projectos de arquitectura em Portugal

Neste capítulo é feita uma breve definição dos projectos de arquitectura e a sua composição (Secção 2.1). Seguidamente são diferenciados os vários tipos de governo electrónico (Secção 2.2) e é feita uma contextualização ao regime jurídico de urbanização e edificação (Secção 2.3). Por fim, expõe-se o cenário actual do licenciamento electrónico em Portugal (Secção 2.4) e descreve-se a problemática da diversidade no formato electrónico da informação (Secção 2.5).

2.1 Projecto de arquitectura

O projecto de arquitectura é a linguagem utilizada pelo arquitecto para comunicação com os agentes envolvidos na construção incluindo o cliente. Habitualmente recorre ao desenho e a elementos escritos mas é predominantemente constituída por peças desenhadas a partir das quais se consegue comunicar os objectivos e modelos a executar em obra. Um projecto de arquitectura pode ser composto por: desenhos de implantação, desenhos gerais de posição, desenhos parciais de posição, desenhos de construção, desenhos de montagem, desenhos de produção de componentes, quadros de inter-referência de desenhos, mapas descritivos e quadros.

2.2 E-Gov

2.2.1 O E-Gov em Portugal e na Europa

Sob a denominação de Governo Electrónico, grande parte dos Estados Europeus assumiu o compromisso de disponibilizar o maior número possível de serviços públicos através da Internet desenvolvendo para tal diversas iniciativas e programas de acção [1].

A sociedade da informação foi considerada uma prioridade da Comissão Europeia, em 2000, durante a presidência portuguesa.

O Plano de Acção eEurope 2000, aprovado na Cimeira de Líderes Europeus, constitui um passo na conquista da modernização electrónica. Neste Plano, os Estados Membros comprometeram-se num objectivo comum, o de transformar a Europa, colocando-a no primeiro lugar da "Economia da Informação e do Conhecimento."

Em 2002 foi criado em Portugal a Unidade de Missão Inovação e Conhecimento (UMIC), uma estrutura de apoio ao desenvolvimento da política governamental em matéria da Inovação, do Governo Electrónico, da Economia Digital, do Apoio aos cidadãos com necessidades especiais na sociedade da Informação e no Acesso à Internet.

Também no domínio do Governo Electrónico foi criada a Comissão Interministerial para a Inovação e Conhecimento, que tem por competências, entre outras, propor estratégias, promover a articulação dos diversos programas e iniciativas e acompanhar a execução do “Planos de acção e-Europe2005: uma sociedade do conhecimento para todos” e de outros programas da União Europeia no âmbito da inovação, da sociedade da informação e do Governo Electrónico.

Estes programas assumem o papel de transformar a Europa numa sociedade de informação, pretendendo os Governos com estas medidas assegurar aos cidadãos, às empresas e restantes entidades governamentais um acesso mais simples, mais rápido e mais rentável aos serviços públicos.

Estes esforços visam aumentar o grau de satisfação dos cidadãos para com o Estado e reforçar a competitividade destes países.

2.2.2 O conceito de E-Gov

O E-Gov não é mais do que o uso das tecnologias da informação, em particular da Internet, para disponibilizar serviços públicos de uma maneira mais eficaz, não só em termos de operacionalidade, mas também em termos de custos e eficiência.

Tal como no sector privado, também o sector público enfrenta o desafio de integração na sociedade da informação, desempenhando o E-Gov um importante papel no projecto global da modernização administrativa. O E-Gov insere os governos na era da informação.

Tendo a Administração Pública como principal objectivo servir bem o cidadão, rapidamente e ao menor custo possível, o E-Gov apresenta-se como uma oportunidade para o fazer de forma mais eficiente através da utilização de tecnologias como a Internet.

O alvo do E-Gov não deve ser as tecnologias de informação e comunicação, mas sim o seu uso que, combinado com mudanças organizacionais, melhora a prestação de serviços públicos, as políticas públicas e o próprio exercício da democracia.

Contudo, a Internet não deve ser encarada como forma única, devendo o E-Gov disponibilizar todos os serviços possíveis através da rede de telefone fixo ou móvel (M-gov), proporcionando assim aos cidadãos um leque de opções variado, podendo estes optar por escolher a que considerem mais adequada e fácil de utilizar.

Envolvendo o conceito de E-Gov relações electrónicas entre o Governo e os diversos actores envolvidos, afiguram-se-nos uma multiplicidade de abordagens possíveis ao conceito, das quais podemos distinguir [2]:

- Governo para o Cidadão – Vertente de Serviço: o Governo estabelece relações directas com o cidadão de forma a oferecer um serviço e/ou benefício;

- Governo para o Cidadão – Vertente Política: a relação existente entre o Governo e o Cidadão é parte integrante do processo democrático. Ex: votos, referendos “on-line”;

- Governo para o Cidadão – Vertente Financeira: a relação estabelecida implica uma troca de informação de carácter financeiro entre o Governo e o Cidadão. Ex: Envio “on-line” de declarações de impostos;

- Governo para o Cidadão – Vertente Comercial: estabelece-se uma relação comercial entre o Governo e o Cidadão. Ex: Central de compras do Estado ”on-line”;

- Governo para o Cidadão – Vertente Patronal: estabelecem-se relações entre o Governo e os seus empregados;

- Governo para o Governo – As relações estabelecem-se ao nível dos diferentes organismos dentro da máquina Estatal.

Uma estratégia de E-Gov deve ser desenvolvida de forma faseada e podemos identificar 4 etapas de evolução:

- Informacional: O Governo disponibiliza a informação na sua forma mais básica. Por exemplo, o Governo coloca nos seus “websites” informação tal como colocaria num “placard” informativo.

- Comunicacional: A entidade governamental já permite que os seus utilizadores efectuem pedidos simples de informação por “e-mail” ou preencher pequenos formulários “on-line”.

- Transaccional: Nesta fase o Governo já disponibiliza aos seus utilizadores um serviço interactivo que permite transacções rápidas de carácter financeiro e informativo.

- Integracional: Nesta etapa o utilizador já pode aceder a uma série de serviços a partir de um único portal.

2.2.3 O E-local Government

Se o conceito de E-Gov engloba o recurso a práticas de base digital que permitem ganhos fundamentais em termos de eficiência, tempos de resposta, acesso à informação e proximidade ao cidadão, o conceito de E-local Government estende esses mesmos princípios, mas com uma maior proximidade ao cidadão (neste caso munícipe) [3].

Assim, a versão local do E-Gov (E-local Government) herda as características referidas para o E-Gov reduzindo-se, no entanto, a uma dimensão local.

Para que seja possível um desenvolvimento estruturado do E-local Gov torna-se necessária a existência na Autarquia das iniciativas das cidades digitais. "A Autarquia Digital, enquanto conceito, prepara o funcionamento orgânico e quotidiano de uma autarquia para o suporte do digital, alterando práticas de forma mais profunda que a simples inclusão de um canal Web (Internet) ou a agilização e racionalização de processos" [4].

O responsável por assegurar a condução e administração do e-local Gov é o poder local, ou seja, as Câmaras e as Juntas de Freguesia que, em conjunto, repartem uma série de serviços que garantem a gestão do território e as relações com os munícipes que neles vivem [5].

Sendo o conceito de E-local Gov ambicioso e complexo, é visto como uma oportunidade de mudança e dinamização para a democracia local e para a melhoria dos serviços prestados pela Administração Pública Local.

As iniciativas de E-local Gov são tomadas em complemento às de E-Gov pois, enquanto munícipe, o Cidadão pretende ver a mesma qualidade de serviço e capacidade de intervenção que lhe é oferecida pelos serviços centrais.

Podemos entender o E-Gov e o E-local Gov como parte de uma tendência que facilita a relação do Cidadão e Munícipe com o poder político e com a Administração Pública.

2.3 Regime Jurídico de Urbanização e Edificação

A Lei n.º 60/2007, de 4 de Setembro, que procede à 6ª alteração do Decreto-Lei 555/99, veio trazer profundas modificações ao Regime Jurídico da Urbanização e Edificação (RJUE), visando, não só a desmaterialização e simplificação dos processos, mas também o aumento da celeridade processual associada aos diferentes tipos de requerimentos e o reforço da eficácia e eficiência das acções de controlo e fiscalização.

O novo RJUE vem dar cumprimento a uma medida do programa SIMPLEX 2007[1] e pretende fomentar a simplificação do procedimento de licenciamento urbanístico

[1] http://www.simplex.pt/downloads/2007ProgramaSimplex.pdf

introduzindo, entre outras soluções, a utilização de novas tecnologias e novas formas de relacionamento entre as diversas entidades envolvidas.

O novo Regime é suportado por uma plataforma electrónica que interliga os munícipes, as Câmaras, as CCDR (Comissões Coordenadoras de Desenvolvimento Regional) e as outras entidades externas. Com a nova lei, os Municípios são obrigados a criar um sistema informático que permita que todos os procedimentos de licenciamento de obras, como por exemplo o pedido de licenciamento de uma obra de construção, possam decorrer integralmente via Internet (Portaria n.º 216-A/2008). Significa isto que o munícipe A pode estar no Porto, em Londres ou em qualquer parte do mundo e tratar com o Município de Barcelos o licenciamento das obras que pretende fazer. O interlocutor do munícipe A será um "Gestor de Procedimentos", uma figura criada pela nova lei, a quem compete o acompanhamento do requerimento ao longo das diversas fases e o estabelecimento de eventuais contactos com o requerente (art. 8º, n.º 3, Lei n.º 60/2007).

O munícipe, a empresa ou o técnico responsável, pode apresentar a sua pretensão ao Município por via electrónica, todo o processo corre por via desmaterializada, todas as consultas são feitas dessa forma e até a decisão final é comunicada aos interessados por essa via.

Pretende-se, através destas inovações, superar os constrangimentos decorrentes da tramitação em suporte de papel, nomeadamente a deslocação física de todos os elementos que instruem os pedidos e comunicações, dentro e fora dos serviços do município, a impossibilidade de acesso remoto à informação, a inexistência de controlo do cumprimento de prazos e o risco de extravio, entre outros.

Não nos podemos, no entanto, esquecer que, apesar de do art. 9º da Lei n.º 60/2007 resultar que todos os procedimentos previstos no RJUE passem a ser necessariamente operacionalizados por meios electrónicos e através de um sistema informático próprio, neste momento a capacidade técnico-jurídica, bem como a disponibilização e o domínio dos meios informáticos por parte dos agentes que intervêm nos processos urbanísticos nem sempre lhes permite cumprir o estipulado na Lei. Por isso e prevendo as dificuldades de implementação plena do sistema informático, o n.º 1 do art. 8º da Portaria 216-A/2008 estabelece que " Nas situações de inexistência ou indisponibilidade do sistema informático ou plataforma, os procedimentos decorrem com recurso à tramitação em papel…", devendo, no entanto,

segundo o n.2 do mesmo art. "…quando se torne possível, ser integrados no sistema informático ou plataforma."

2.4 O Licenciamento electrónico em Portugal

Nos termos da Lei 60/2007, de 4 de Setembro, a tramitação dos procedimentos de licenciamento e de comunicação prevista no Regime Jurídico da Urbanização e Edificação passa a ser feita informaticamente com recurso a um sistema informático próprio.

A implementação deste sistema informático compreende duas fases distintas. A primeira, que se encontra em pleno funcionamento desde Julho de 2008, é relativa à consulta dos municípios a entidades da administração central para efeitos de parecer em momento anterior à decisão. Nesta fase do projecto, desmaterializou-se o relacionamento entre as autarquias, comissões de coordenação e desenvolvimento regional e outras entidades externas que têm de se pronunciar num processo de licenciamento urbano. Significa isto que as consultas às entidades externas passaram a ser feitas "on-line", eliminando o recurso ao papel.

Na segunda fase é disponibilizada aos Municípios uma solução informática que integra a plataforma centralizadora dos pedidos de licenciamento e possibilita iniciar electronicamente e de forma totalmente desmaterializada qualquer processo de licenciamento abrangido pelo RJUE. Desmaterializa-se, desta forma, o relacionamento entre os particulares e os municípios, permitindo a apresentação de pedidos de licença "on-line" e a notificação sobre a decisão final do processo pela mesma via. Nesta fase para além de se superar os constrangimentos decorrentes da tramitação em papel, eliminam-se as deslocações à Câmara e permite-se um acompanhamento de todo o processo "on-line".

Não obstante existirem já alguns Municípios com plataformas próprias a funcionar, a Direcção-Geral da Administração Local (DGAL) desenvolveu um projecto que permite às Autarquias Locais que não tenham aplicações próprias que possibilitem assegurar um serviço de licenciamento totalmente informático e "on-line", a utilização de um sistema cedido gratuitamente por esta Direcção-Geral.

Este projecto encontra-se em fase de implementação e aplicação em várias Autarquias, como é o caso de Campo Maior, que se tornou o primeiro Município no

país a disponibilizar aos seus cidadãos a possibilidade de pedirem licenças "on-line" e receberem a resposta pela mesma via. Ao arrancar, a 6 de Fevereiro deste ano, com a segunda fase do "Portal do Licenciamento", Campo Maior tornou-se pioneiro na disponibilização aos seus munícipes da ferramenta informática que desmaterializa e simplifica os processos de licenciamento. Segundo informações do Município, para que qualquer cidadão de Campo Maior possa iniciar electronicamente um pedido de licenciamento é necessário que faça o registo no portal autárquico[2] (comum para todas as autarquias) e que possua uma assinatura digital qualificada, tornando-se para este efeito indispensável o Cartão de Cidadão. Ainda segundo informação de um técnico do Município[3], até à data de 3 de Março nenhum munícipe terá acedido à Plataforma e feito qualquer tipo de pedido de licenciamento "on-line."

Também o Município de Águeda permite, desde Fevereiro deste ano, aos seus munícipes a possibilidade de entrega em suporte digital dos processos de licenciamento. Ao invés do Município de Campo Maior, que preferiu utilizar o sistema informático cedido pela DGAL, Águeda optou por desenvolver juntamente com a Medidata um sistema próprio.

A Medidata.net desenvolve e comercializa Sistemas de Informação para Autarquias e possui uma forte presença no mercado autárquico. Tem vindo a desenvolver, desde Outubro de 2008, em articulação com a DGAL, um módulo para permitir a interoperabilidade com o portal do RJUE. Este módulo permite integrar a informação da autarquia com a informação do portal através de "webservices" disponibilizados por este. Assim, os dados são inseridos na aplicação de Urbanismo – SIGMA e, após o seu tratamento, são submetidos automaticamente no portal, juntamente com os documentos anexos. Esta integração permite a articulação com os restantes serviços intervenientes, nomeadamente o planeamento urbanístico, a fiscalização e a receita.

Segundo informações prestadas por uma técnica do Secretariado de Apoio Técnico Administrativo do Município de Águeda[4], a 4 de Março passado, a Autarquia e a Medidata trabalham em conjunto, neste momento, numa alteração ao sistema

---

[2] https://servicos.portalautarquico.pt/portal_citizen/

[3] Informação obtida telefonicamente em 3 de Março de 2009 pelas 17 horas

[4] Informação obtida telefonicamente em 4 de Março de 2009 pelas 17 horas

informático e tentam encontrar uma solução por forma a evitar a necessidade da duplicação de dados que o presente sistema obriga. Também no caso de Águeda, e segundo declarações da técnica, ainda nenhum munícipe terá submetido um pedido de licenciamento "on-line".

Para além do Município de Águeda, a Medidata surge como parceira dos Municípios de Guimarães, Grândola, S. João da Madeira e Vila Nova de Gaia, entre outros.

À semelhança da Medidata, também a AIRC (Associação Informática da Região Centro)[6] desenvolveu e tem disponíveis um conjunto de soluções que pretendem responder às modificações trazidas pelo RJUE, nomeadamente a desmaterialização e tramitação electrónica dos processos de edificação e urbanização. O MyURB[5] permite que, através dos sítios web dos Municípios, os cidadãos e as empresas possam submeter requerimentos "on-line" e consultar o estado dos seus processos. O Toolkit SPO, um módulo de integração, permite a submissão e acompanhamento de pedidos de consulta a entidades externas no portal do RJUE da DGAL. Esta capacidade permitirá o relacionamento da informação da aplicação do município com a disponível no Portal do RJUE. Os processos podem continuar a ser registados e tramitados na aplicação do Município, podendo a partir deste, e de forma automática, submeter os elementos necessários aos pedidos de consulta a entidades externas, bem como durante o processo de apreciação, acompanhar o estado do pedido.

Existem Municípios que ainda não decidiram como vão permitir aos seus cidadãos a submissão de processos de edificação e urbanização "on-line". É necessário ponderar as soluções existentes: utilizar a plataforma da DGAL, instalar uma solução proprietária ou desenvolver uma solução própria. No caso do Município de Barcelos, que desde 2004 disponibiliza informação detalhada do estado e tramitação dos processos "online", tem agendado para este ano o desenvolvimento de uma solução de interactividade com os seus munícipes.

---

5 http://portal.airc.pt/lwp/wcm/connect/AIRC/AIRC/Noticias+e+Destaques/MyNetUrb acedido a 9 de Março pelas 20 horas

2.5 Diversidade no Formato electrónico da Informação

O formato electrónico é a denominação da estrutura usada por determinada aplicação computacional na escrita e leitura de dados gerados. Esse processo permite armazenar dados de forma electrónica[7].

No processo de desmaterialização de processos, a fase da escolha dos formatos a utilizar é de extrema importância visto que os Municípios não vão querer escolher um formato que se torne rapidamente obsoleto ou que, por outro lado, não satisfaça as necessidades de todos os intervenientes dos processos.

O leque de formatos seleccionados deve abranger as peças escritas, imagens/fotografias e ficheiros técnicos de desenho de forma que nenhum elemento do processo seja excluído por não existir representação em formato digital. A selecção deve, na medida do possível, passar por formatos abertos e/ou normalizados. Se todo o "software" utilizado implementar as normas, não haverá dificuldade em abrir, visualizar e utilizar os ficheiros necessários, independentemente da plataforma utilizada, sem custo técnico/financeiro adicional.

A Organização Internacional para padronização[6] (ISO) disponibiliza, desde 1 de Julho de 2008, um padrão aberto (ISSO 32000-1:2008) para o transporte de documentos: PDF (Portable Document Format) originalmente desenhado pela empresa Adobe[7]. Este formato descreve documentos que contenham texto, gráficos e imagens de maneira independente de dispositivos e resolução. A mesma organização define outra norma (ISO/IEC 10981-1) para algoritmos de compressão utilizados nos formatos de imagem JPEG e TIFF. Estes "standards" vêm facilitar a escolha dos formatos a utilizar para texto, imagens e fotografias.

Por outro lado, no que concerne aos ficheiros para o desenho assistido por computador (CAD), não existem normas internacionais para a sua utilização. Para este tipo de ficheiros, para além da visualização, outras funcionalidades devem ser levadas em consideração, tal como a compreensão clara do projecto, a medição rigorosa de

---

[6] http://www.iso.org/iso/home.htm

[7] http://www.adobe.com/

distâncias de acordo com a escala utilizada, o suporte de imagem com linhas escondidas e a conversão para formato vectorial. Tudo isto torna complexo o processo de selecção de um formato electrónico adequado.

A Autodesk[8] tentou, tal como a Adobe no formato PDF, desenvolver um formato que fosse transparente à plataforma e software utilizado: o Design Web Format (DWF). Mas, talvez pelas exigências a nível de requisitos que obrigam a uma constante evolução neste tipo de software, não se impôs numa norma "de facto". De qualquer modo, a DGAL, para o seu portal, optou por este formato, exigindo como requisito técnico um visualizador de DWF, como é o caso da aplicação gratuita da Autodesk: Autodesk DWF Viewer.

A Tabela 1 resume os formatos electrónicos mais utilizados em diversas aplicações de desenho assistido por computador. Os dados foram recolhidas pela informação disponibilizada nos manuais das aplicações.

Os formatos DWF, DWG e DXG (assinalados a cinza) cumprem os requisitos necessários além de que são suportados pela totalidade das aplicações apreciadas.

Formato	Tipo de formato	Descrição do formato	Autocad 2009	Revit Architecture 2009	Microstation	Archicad	Zwcad	Google Sketchup
DAE	Formato 3D	Collada						x
FBX	Formato 3D	ArcView Spatial Index File						x
OBJ	Formato 3D	WaveFront File				x		x
VRML	Formato 3D	VRML World files			x	x		x
XSI	Formato 3D	Softimage						x
3DS	Formato Editável	3D Studio File	x			x		x
DGN	Formato Editável	MicroStation DGN file	x	x		x		
DWF	Formato Editável	Autodesk Design Web Format	x	x	x	x	x	x
DWG	Formato Editável	Autocad Drawing Format	x	x	x	x	x	x
DXF	Formato Editável	Data Exchange Format	x	x	x	x	x	x
DXG	Formato Editável	Autocad Drawing Interchange Format	x					
DXX	Formato Editável	Attribute extract DXF	x					
EPS	Formato Editável	Encapsulated PostScript file	x				x	x
GDL	Formato Editável	Geometric Description Language File				x		
GRD	Formato Editável	MicroStation Field format			x			

[8] http://www.autodesk.com

Formato	Tipo de formato	Descrição do formato	Autocad 2009	Revit Architecture 2009	Microstation	Archicad	Zwcad	Google Sketchup
IFC	Formato Editável	Industry Foundation Classes				x		
IGES	Formato Editável	Initial Graphics Exchange			x			
LP	Formato Editável	Lightscape File				x		
MOD	Formato Editável	Archicade Module File				x		
PLA	Formato Editável	Archicade Project				x		
PSD	Formato Editável	Photoshop document				x		
SAT	Formato Editável	ACIS solid object file	x	x				
SE	Formato Editável	QuickVision-accelerated view						
SGI	Formato Editável	Silicon Graphics Image File				x		
STL	Formato Editável	Solid object stereolithography file	x					
SVG	Formato Editável	Scalable Vector Graphics					x	
TPL	Formato Editável	Archicade Project Template				x		
WMF	Formato Editável	Microsoft Windows Metafile	x			x	x	
BMP	Formato Raster	Device-independent bitmap file	x			x	x	x
CALS	Formato Raster	CALS Raster file format	x					
CGM	Formato Raster	Computer Graphics Metafile			x			
EMF	Formato Raster	Enhanced Metafile				x	x	
EPX	Formato Raster	Piranesi File Format						x
FACT	Formato Raster	ElectricImage File				x		
FLIC	Formato Raster	FLIC file format	x					
GeoSPOT	Formato Raster	GIS-Geospot format	x					
GIF	Formato Raster	Graphics Interchange Format	x					
JFIF	Formato Raster	JPEG File Interchange Format	x					
JPEG	Formato Raster	Joint Photographic Experts Group			x	x		x
PCX	Formato Raster	PC Paintbrush Exchange	x					
PDF	Formato Raster	Portable Document Format			x	x		x
PICT	Formato Raster	Picture file format	x			x		
PNG	Formato Raster	Portable Network Graphics	x			x		x
QTIF	Formato Raster	QuickTime Image				x		
RLC	Formato Raster	ArcView Image File	x					
TGA	Formato Raster	Truevision TGA Image	x			x		
TIF	Formato Raster	Tagged Image File Format	x			x		x

Tabela 1: Formatos em aplicações CAD

3 Conceitos Basilares de segurança

Neste capítulo são exposto conceitos de segurança informática e é feita uma abordagem à segurança do Cartão de Cidadão (Secção 3.10).

3.1 Identidade

A conjuntura actual motiva os organismos públicos a fortalecerem a cooperação e a ligarem os serviços entre eles. Muitos, tendem a expandir as suas actividades através da integração “online” de sistemas e serviços necessitando que entidades externas acedam à sua informação interna. A gestão da identificação digital constituiu uma parte fundamental na integração desses sistemas e serviços. O sistema tem de saber quem acedeu a quê, em alguns casos, e identificar utilizadores, noutros.

A arquitectura da gestão de identidades deve ser baseada em normas abertas para permitir a fácil e segura inter-operação de sistemas e o aumento da escala de serviços e de aplicações.

De uma forma simplificada, pode dizer-se que a nossa identidade é quem somos e quais são os nossos atributos. Quando se pede para provar a identidade, pode-se fazê-lo de várias formas: por impressão digital, assinatura ou simplesmente apresentando um cartão de identificação.

A identidade pode dividir-se em 3 camadas[8]:

- a primeira camada está associada aos atributos intemporais e incondicionais, como a data de aniversário ou a cor dos olhos;
- a segunda camada está associada aos atributos que foram conferidos por uma instituição, como a carta de condução ou um cartão de crédito;
- a terceira camada está associada à identificação do grupo a que se pertence, como por exemplo, “engenheiros com mais de 30 anos”.

A maioria dos problemas de identidade estão relacionados com a segunda camada.

3.2 Confiança

A confiança é algo que é implicitamente compreendido pelos humanos. É algo difícil de medir e de quantificar. A confiança acontece a diferentes níveis: de pessoa para pessoa, de pessoa para grupo e de grupo para grupo. A confiança é normalmente estabelecida ao longo do tempo e o nível de confiança depende das circunstâncias.

A confiança tem as seguintes propriedades:

- só é transitiva em circunstâncias muito específicas;
- não pode ser partilhada;
- não é simétrica;
- a falta de confiança não pode ser declarada pois é baseada na reputação.

A reputação é algo que é constituído gradualmente nas comunidades de confiança. Estas comunidades têm 5 componentes:

- O governo, que estabelece as regras e as responsabilidades;
- As pessoas envolvidas na comunidade de confiança;
- O processo das operações e transacções;
- As ferramentas tecnológicas disponíveis;
- A viabilidade do modelo económico utilizado.

A construção da infra-estrutura da identidade digital deve ir ao encontro de todos os requisitos da comunidade de confiança.

3.3 Política de segurança

A política de segurança é o conjunto de objectivos baseados nos requisitos de negócio, as circunstâncias em que são efectuadas as transacções, o nível de risco que o negócio está disposto a suportar e o custo que o negócio está disposto a pagar para diminuir o risco para valores aceitáveis.

Alguns negócios são mais adversos ao risco que outros pelo que os seus requisitos definem o nível aceitável de risco. O risco associado a uma determinada transacção deve ser quantificado e balanceado com o benefício retornado. Normalmente, para um grande benefício esperado, um grande risco deve ser aceite.

3.4 Privacidade

Genericamente, os indivíduos desejam os seus dados pessoais protegidos e simultaneamente desejam preservar a sua privacidade. Existem empresas que são de tal forma dependentes da privacidade que criam gabinetes próprios de forma a garantir a privacidade deles e dos seus clientes.

Um tema de intenso debate internacional é o nível de privacidade aceitável para cada país. Leis e regulamentos de diferentes países não são unânimes tornando difícil o modo como deve ser utilizada a informação pessoal, e a identificação da violação da privacidade.

Diversas organizações utilizam a Internet para recolher informação pessoal dos seus visitantes, muitas vezes sem o seu conhecimento. Qualquer “internauta” tem o direito a saber que informação está a ser recolhida, a razão de tal e por que entidades, mas nem sempre esse direito é respeitado. Ainda há as organizações que aproveitam esta recolha de informação como uma oportunidade de negócio.

No âmbito da identidade digital, a viabilidade das infra-estruturas de gestão de identidades está directamente relacionada com o sentimento de segurança por parte dos utilizadores. Para que os responsáveis dessas infra-estruturas possam proteger os dados dos utilizadores, estes devem ser capazes de controlar todo o ciclo de vida da

informação: sua criação, o seu tratamento e a sua destruição. O controlo de segurança deve ser tomado nas operações de armazenamento, transporte, criação de cópias de segurança, registos e outras.

3.5 Autenticação

Para que um sistema possa ser considerado seguro terão de existir meios para identificar os seus utilizadores, não apenas para decidir que tipo de acesso à informação deverá ter, mas também para ser possível definir as suas operações dentro do sistema. A operação de identificação bem sucedida de um utilizador num sistema designa-se por autenticação, ou seja, o utilizador afirma ter uma certa identidade e prova-o.

3.6 Criptografia

A criptografia é a ciência de tornar o custo de descobrir informação protegida mais alto do que o seu próprio valor[9].

A criptografia é classificada em dois tipos: criptografia simétrica e criptografia assimétrica (ou criptografia de chave pública).

A criptografia simétrica usa uma única chave secreta para transformar uma mensagem normal numa mensagem ilegível. Para transformar a mensagem ilegível no seu original é necessário estar na posse da referida chave. O secretismo na troca e partilha das chaves secretas entre os comunicadores é uma componente indispensável para garantir a confidencialidade da mensagem.

A criptografia de chave pública usa uma chave pública e uma chave privada. A chave privada é secreta para todos excepto para o seu proprietário, enquanto a chave pública é, como o próprio nome indica, acessível a todos. Uma mensagem cifrada por uma chave pública apenas pode ser decifrada pela correspondente chave privada o que permite garantir a confidencialidade de informação transmitida. Uma mensagem cifrada pela chave privada apenas pode ser decifrada pela correspondente chave pública garantindo a integridade da mensagem e o seu não-repúdio (certificação de que a mensagem foi realmente enviada pelo autor).

Sistemas reversíveis, em que a informação original pode ser recuperada a partir do código gerado pela operação de cifra, garantem a confidencialidade, integridade e o não-repúdio. Os sistemas irreversíveis, em que a informação original nunca mais pode ser obtida a partir do seu código correspondente, apenas garantem a integridade e o não-repúdio.

Sistemas de criptografia baseados em chaves públicas demoram de 100 a 1000 vezes mais tempo a cifrar e decifrar do que os sistemas de chave secreta, do que resulta não ser aconselhada a sua utilização para grandes quantidades de informação. Para além disso, a exposição pública de muita informação cifrada com a mesma chave, facilita a cripto-análise; a solução, a troca frequente de chaves, não é muita prática em sistemas assimétricos. Estes sistemas são por isso, mais utilizados para transaccionar confidencialmente pequenos padrões de dados, como chaves secretas, muitas vezes temporárias. São, também, mais importantes em aplicações cujo objectivo é a garantir integridade e autenticação (Ver secção seguinte).

A combinação do sistema simétrico com o assimétrico é denominada “criptografia híbrida”.

3.7 Assinaturas digitais

O princípio da assinatura digital é similar à convencional assinatura no papel: a uma dada mensagem é adicionada uma assinatura. Tal como na assinatura convencional, só a pessoa que enviou a mensagem deve ser capaz de gerar uma assinatura válida. Num sistema criptográfico, a assinatura é uma função da chave privada, fazendo com que apenas o proprietário da chave privada seja capaz de assinar a mensagem. De forma a fazer com que a assinatura mude em cada mensagem, a função tem também como parâmetro o documento que vai ser assinado.

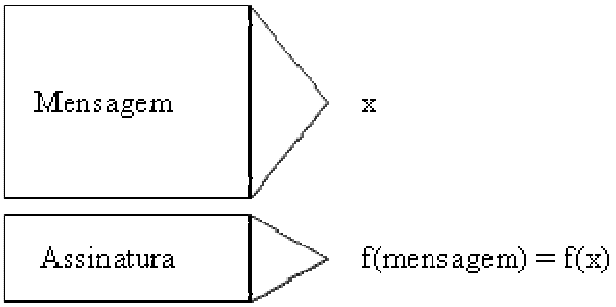


Figura 1: Assinatura digital e o bloco da mensagem

A grande vantagem das assinaturas é que elas permitem que os receptores da informação sejam capazes de comprovar que o remetente realmente gerou a mensagem.

As assinaturas digitais pressupõem que:

- só o remetente é capaz de assinar os seus documentos;
- qualquer pessoa é capaz de verificar a sua autenticidade;
- os destinatários têm a certeza de que a assinatura pertence ao remetente correspondente;
- a mensagem não pode ser alterada sem que tal seja detectado na verificação da assinatura;
- o destinatário tem a certeza de que a mensagem foi enviada pelo remetente.

Com vista a facilitar a compreensão do funcionamento das Assinaturas Digitais, foi criada a documentação RFC 2633 do IETF, onde foram definidas as especificações para a sua correcta utilização, das quais se destacam as seguintes[10]:

- para assegurar a sua compatibilidade com sistemas mais antigos, é aconselhável gerar ficheiros de assinatura com um nome de ficheiro não superior a 8 caracteres, seguido dos 3 caracteres da extensão que lhe corresponde;

- para identificar mais facilmente o conteúdo de uma assinatura digital, as extensões dos ficheiros criados e enviados para o sistema deverão ser:

*.p7m - nos casos em que sejam aglutinados num único ficheiro tanto o ficheiro a assinar como a própria assinatura;

*.p7s - nos casos em que o ficheiro com esta extensão contenha apenas a assinatura e os certificados utilizados no seu processo de criação, tendo o ficheiro de mensagem de ser enviado junto.

3.8 Certificados digitais

Um certificado digital é um ficheiro de computador que contém um conjunto de informações referente à entidade para o qual o certificado foi emitido (seja uma

empresa, pessoa ou computador) juntamente com a chave pública referente à chave privada que se acredita estar na posse unicamente da entidade especificada no certificado. A autenticidade de um certificado é garantida pela entidade certificadora que o assina digitalmente.

Os certificados digitais podem ser usados como parte de uma arquitectura denominada "infra-estrutura de chaves públicas" (Public-Key Infrastructure - PKI). De uma forma muito sucinta, a PKI consiste na especificação, meio de armazenamento e disponibilização de todas as chaves públicas das entidades que nela participarem.

3.9 Secure Sockets Layer

O protocolo *Secure Sockets Layer* (SSL) ou a sua "versão" padronizada, *Transport Layer Security* (TLS), é usado para assegurar as comunicações cliente-servidor em redes informáticas, negociando e fornecendo as funções essenciais de uma transacção segura: autenticação de uma das partes ou mútua (opcional), encriptação de dados e integridade de dados. O protocolo previne, assim, que intermediários entre as extremidades da comunicação tenham acesso indevido ou falsifiquem os dados que são transmitidos[11].

Uma comunicação cliente-servidor segura requer: autenticação do servidor (podendo requerer também a do cliente), a troca de uma chave criptográfica entre ambas as partes e a encriptação dos dados usando a chave criptográfica definida para esse efeito.

Quando um cliente e um servidor concordam em comunicar usando o protocolo SSL, também necessitam de concordar em diversos outros pontos:

- o algoritmo criptográfico que vão usar;
- a necessidade ou não de autenticação mútua;
- os algoritmos de geração de chaves criptográficas;
- que chave de sessão será criada para cifrar a mensagem.

Os certificados digitais permitem ao cliente e ao servidor identificarem-se mutuamente. Em todas as negociações de SSL, o cliente deve verificar a identidade do servidor recorrendo a certificados digitais. O servidor pode também requisitar ao cliente o envio

de um certificado digital. Para que a negociação seja bem-sucedida entre o cliente de SSL, este deve confiar na autoridade certificadora que emitiu os certificados digitais do servidor. Depois do sucesso da negociação, a informação trocada entre o cliente e o servidor será encriptada usando as chaves digitais negociadas.

Um exemplo da implementação do protocolo SSL é o Hypertext Transfer Protocol Secure (HTTPS) que é uma combinação do Hypertext Transfer Protocol (HTTP) com o protocolo criptográfico SSL. Ao contrário do HTTP, que opera na camada mais elevada do TCP/IP, o HTTPS opera numa camada mais baixa, cifrando a mensagem HTTP antes da transmissão e decifrando a mensagem à chegada, de modo a garantir a sua segurança. Por definição, o porto utilizado na interacção com a camada de baixo nível TCP/IP pelo protocolo HTTPS (443) é diferente do utilizado no protocolo HTTP (80) [12].

3.10 Segurança no Cartão de Cidadão

O Cartão de Cidadão é o novo cartão de identificação dos cidadãos nacionais. É um documento que agrega e substitui os actuais bilhetes de identidade, cartões de contribuinte, de utente do serviço nacional de saúde, de beneficiário da segurança social e de eleitor [13].

O Cartão tem o formato "smart card" e exibe, na frente, a fotografia e os elementos de identificação civil, e no verso, os números de identificação dos diferentes organismos cujos cartões agrega e substitui, uma zona de leitura óptica e um circuito electrónico (componente electrónica do Cartão de Cidadão).

No “chip” são armazenados os restantes dados biométricos do cidadão (e.g. impressões digitais), a morada, informação relativa a indicações eventuais, outras informações que podem ser registadas opcionalmente pelo titular numa zona destinada a arquivar notas pessoais de leitura pública, mas de escrita limitada ao titular, data de emissão, certificado para autenticação segura, certificado qualificado para assinatura electrónica qualificada e as aplicações informáticas necessárias ao desempenho das funcionalidades do Cartão de Cidadão e à sua gestão e segurança.

O certificado para autenticação segura permite ao respectivo titular, conforme o disposto na alínea b) do n.º 2 do artigo 6.º do Decreto-Lei n.º 7/2007, provar a sua identidade perante terceiros através de autenticação electrónica. Ainda no mesmo

Decreto-Lei, na alínea c), é regulamentado que o certificado qualificado para assinatura electrónica qualificada permite autenticar de forma unívoca a sua qualidade de autor de um documento electrónico.

A assinatura digital qualificada é o mecanismo que permite ao titular de um Cartão de Cidadão, por vontade própria, assumir de forma inequívoca a autoria de um documento, assinado com a chave criptográfica pessoal residente no seu cartão. Qualquer entidade pode verificar a assinatura digital aposta pelo cidadão num documento, recorrendo ao uso do certificado digital pessoal do cidadão e a mecanismos de verificação da validade deste certificado.

A assinatura digital tem o valor legal conferido pela lei, nomeadamente no Decreto-Lei nº 290-D/99, de 2 de Agosto, republicado pelo Decreto-Lei nº 62/2003, de 3 de Abril e alterado pelos Decretos-Lei nºs 165/2004, de 6 de Julho e 116-A/2006, de 16 de Junho.

De salientar que um documento electrónico é susceptível de representação como declaração escrita quando lhe seja aposta uma assinatura electrónica qualificada; tal documento electrónico tem a força probatória de um documento particular assinado, nos termos do disposto no artigo 376.º do Código Civil. Isto é, esse documento faz prova plena quanto às declarações atribuídas ao seu autor (sem prejuízo de poder ser arguida e provada a sua falsidade).

4 Abordagem Metodológica e Resolução

Todo o software deve ser produzido usando algum tipo de metodologia. Apesar da maioria das metodologias terem sido pensadas para a produção de software de grandes dimensões e produzido por equipas numerosas, pequenas partes de software desenvolvidas por apenas uma pessoa podem ser melhoradas se tiverem em conta uma metodologia [14].

Uma metodologia é a descrição das etapas que devem ser seguidas para fazer coisas. No domínio da Engenharia Informática é um processo cíclico que pode ser seguido desde as primeiras fases do desenvolvimento do software, até à instalação e manutenção do sistema. A metodologia deve contemplar também a gestão de recursos, o planeamento, a calendarização ou gestão de tarefas [14].

Este capítulo explica a metodologia utilizada na elaboração do protótipo (secção 4.1) e da especificação os requisitos (Secção 4.2).

4.1 Metodologia adoptada

Existem várias metodologias de desenvolvimento de projectos que se adaptam a cada estilo de programação e para cada tipo de ambiente. Independentemente da metodologia, existem fases que são comuns a todas elas, mesmo que por vezes sejam abordadas por denominações diferentes.

Como requisito único para o desenvolvimento do protótipo foi definido que a metodologia a adoptar não possuísse uma curva de aprendizagem demasiadamente longa. Nos anos setenta surgiu a programação orientada a objectos (POO) que, pelo facto de ser amplamente estudada no seio universitário, responde a esta necessidade. A POO é um paradigma de análise, projecto e programação de sistemas de software, organizados por unidades designadas de "objectos", que tem como principal objectivo a reutilização do código e a modularidade da escrita.

A modelação do paradigma foi feita utilizando listas, tabelas e artefactos e sempre que possível recorrendo à linguagem de modelação "Unified Modelling Language" (UML). O UML é uma linguagem diagramática utilizável para a especificação, visualização e documentação de sistemas de informação. É uma linguagem de aplicação geral e utilizada nos mais diversos domínios. É caracterizada pela sua independência do domínio da aplicação, do processo ou metodologia de desenvolvimento e da ferramenta de modelação, apresenta mecanismos de extensão e é composta, na versão 2, por treze diferentes diagramas segundo três visões principais: visão estática ou estrutural (diagrama de pacotes, de classes, de objectos, de estrutura composta, de componentes e de instalação), visão funcional (diagrama de casos de utilização e de actividade) e visão dinâmica (diagrama de máquina de estados, de sequência, de comunicação, de visão geral da interacção e temporal). O UML é uma linguagem que suporta todo o ciclo de um projecto de software comportando a modelação do negócio, a modelação de requisitos e a modelação da solução de software[15]. A ferramenta de modelação utilizada para desenhar os diagramas UML foi o Sybase Power Designer[9].

A organização das fases do projecto foi feita segundo a seguinte divisão: especificação dos requisitos, análise, desenho, implementação, e instalação. A Tabela 2 resume as peças criadas, organizados pelas diferentes fases e pelo tipo de anotação utilizada.

[9] http://www.sybase.com/products/modelingmetadata/powerdesigner

Fase		Artefactos	Anotação utilizada
Requisitos	Funcionais	Lista de Actores Lista de casos de uso	Tabela Tabela
	Não Funcionais	Hierarquia de requisitos	
	Sistema	Diagrama de casos de uso Diagrama de actividades Diagrama de sequências	UML UML UML
Análise		Diagrama de classes	UML
Desenho	Sistema	Diagrama de Instalação	UML
	Subsistema	Diagrama de sequência Modelo Entidade-Relacionamento (E-R)	UML Modelo E-R
Implementação		Código fonte	
Instalação		Manuais	

Tabela 2: Peças criadas, organizadas pelas diferentes fases e pelo tipo de anotação utilizada

4.2 Especificação de Requisitos

Os requisitos para um sistema são a descrição dos serviços providenciados e as restrições das operações [16]. Esses requisitos são separados em diferentes níveis, descritos abaixo.

O objectivo do projecto é desenvolver um protótipo que faculte aos municípios a gestão documental de processos de urbanização e edificação de uma forma electrónica, totalmente desmaterializada e segura.

4.2.1 Dicionário de Dados do sistema de informação

O dicionário de dados do sistema é a definição do conjunto de elementos que são pertinentes para o sistema. De seguida são enunciadas algumas definições consideradas importantes:

- Processo: Conjunto de documentos desenhados, escritos, com informação e organização regulamentada, quer pela lei geral, decretos-lei, regulamentos

municipais ou outros documentos normativos, tendo por objectivo o licenciamento de uma obra.

- Requerimento: Documento entregue na autarquia, por empresas ou particulares, destinado a solicitar informações ou a solicitar a aprovação de um processo de licenciamento de obra e que acompanha o processo, devidamente autenticado.

- Informação: Documento onde consta um facto ou acontecimento que é levado ao conhecimento de alguém ou de um público através de palavras, sons ou imagens.

- Ofício: Documento de carácter oficial enviado por uma autoridade.

- Despacho: Nota lançada numa petição ou requerimento.

- Tramitação: Curso de um processo, segundo as respectivas regras (trâmites).

- Alvará/Licença: Diploma emanado de entidade competente em que se fazem concessões e nomeações, se deferem pretensões e mercês ou se aprovam estatutos, e que tem por fundamento disposições legais existentes.

Com intuito de melhor compreender os actores presentes nos casos de uso, é feita uma breve descrição de cada um deles.

- Munícipe: Pessoa singular ou colectiva, que recorre aos serviços da administração pública, neste caso a um Município, e que submete um processo de licenciamento, ou outro qualquer pedido, através de um requerimento devidamente assinado e autenticado;

- Projectista: Técnico com habilitação superior própria para assinar projectos e dirigir obras de construção civil, planeamento urbano, ou outras previstas na legislação vigente, inscrito na respectiva ordem ou associação profissional;

- Gestor de Procedimentos: Tem por missão fazer o acompanhamento do requerimento ao longo das diversas fases e efectuar eventuais contactos com o requerente.

- Administrativo: Tem por missão fazer a triagem inicial dos Requerimentos e carregá-los no sistema.

- Coordenador Autárquico: Tem por missão analisar o processo, propor as Entidades Externas a consultar, enviar o processo para a CCDR e,nesta fase, receber a decisão da CCDR e arquivar o processo.

- Entidades Externas: Organizações que participam nos processos de arquitectura mas que não pertencem à hierarquia dos municípios.

4.2.2 Requisitos Funcionais

Os requisitos funcionais são a lista de serviços que o sistema deve disponibilizar, de como o sistema deve reagir a determinadas entradas e como se deve comportar nessas situações. A especificação dos requisitos funcionais é feita de seguida.

O sistema deve permitir a inserção de utilizadores no sistema de acordo com a informação do departamento municipal. Tal tarefa estará a cargo de administradores, que são utilizadores com poderes alargados de manipulação do sistema. O sistema deve permitir que os administrativos insiram novos munícipes bem como novos projectistas e, se for caso disso, organizá-los por estruturas orgânicas, como por exemplo, gabinetes técnicos. As estruturas orgânicas são grupos virtuais de projectistas. O acesso e permissões dos projectistas aos projectos são definidos pelo responsável técnico da estrutura.

O sistema deve armazenar as habilitações literárias dos projectistas, os respectivos comprovativos e referência de inscrição nas Ordens profissionais, bem como a assinatura dos termos de responsabilidade de acesso ao sistema.

Os munícipes e projectistas devem ter a capacidade de submeter novos projectos e documentos. Os administrativos devem ser capazes de poder supervisionar esses processos e documentos submetidos e, além disso, supervisionar a entrada de novos termos de aceitação e habilitações literárias.

O sistema tem que suportar um mecanismo de mensagens que permita a comunicação entre o munícipe e o coordenador autárquico, entre o projectista e o coordenador autárquico e entre o coordenador autárquico e o projectista ou munícipe.

Deve permitir aos coordenadores autárquicos gerir a disponibilidade para atendimento e permitir aos munícipes ou projectistas a marcação de atendimento nas vagas disponíveis. Deve ser possível aos coordenadores autárquicos registar observações desses atendimentos.

Dentro das estruturas orgânicas os projectistas devem ter dois níveis de acesso. No primeiro nível, o projectista tem acesso à informação do projecto e à troca e partilha de informação, mas de forma não vinculativa; no segundo nível, além do acesso de nível 1, a troca e partilha de informação é efectuada de forma vinculativa, ou seja, por trâmite.

As informações vinculativas são entregues em formato digital ou entregues em papel nos serviços do Município e posteriormente introduzidas no sistema pelo administrativo. A informação, por parte dos munícipes ou projectistas é produzida por requerimento. No caso dos coordenadores autárquicos, a informação é produzida sob o formato de informação, despacho, tramitação ou ofício. O sistema deve estar preparado também para anexar informação por parte de entidades externas, os denominados pareceres, e de visualizar alvarás e licenças emitidas. A informação não vinculativa é efectuada pelo mecanismo de mensagens em formato digital. No caso de o coordenador autárquico ter sido contactado por outro meio que não pelo mecanismo de mensagem, deve ser capaz de registar esse contacto.

O sistema deve facultar a todos os utilizadores um repositório que permita carregar ficheiros antes da sua a anexação.

Deve ser possível visualizar toda a informação do processo em formato digital, inclusive as digitalizações dos documentos em papel entregues nos serviços camarários.

O sistema deve guardar um registo não alterável de toda a actividade do sistema, de forma a poder ser usado em auditorias.

4.2.3 Requisitos Não Funcionais

Os requisitos não funcionais descrevem as restrições dos serviços ou funções oferecidas pelo sistema: restrições de tempo, de desenvolvimento e normas. São os requisitos do sistema como um todo.

Os requisitos não funcionais estão subdivididos nos seguintes pontos:

- Usabilidade: O sistema deve ser baseado numa interface Web, simples e intuitiva, que permita uma navegação e operação rápida por parte de todos os utilizadores do sistema.

- Eficiência: Dado que o sistema será uma ferramenta de uso diário, o tempo de execução das operações deve ser minimizado. O desempenho do sistema não deve ser afectado pelo acesso em simultâneo de um número de utilizadores pouco superior à média diária de utilização. Esta capacidade de atendimento deve ser considerada na escolha da tecnologia a usar.

- Segurança: As funcionalidades devem depender do nível de acesso do utilizador ao sistema. O sistema deve garantir que os dados só são acedidos e alterados por quem de direito. Os utilizadores devem possuir autorização apropriada. Um registo não alterável das operações realizadas permitirá detectar infracções à política de segurança do sistema.

- Fiabilidade: O sistema deve assegurar que a informação apresentada seja fidedigna de forma aos utilizadores poderem nela confiar.

- Manutenção: O sistema deve ser de fácil manutenção no que respeita ao seu funcionamento contínuo. Desta forma, devem existir mecanismos para ajudar na resolução de problemas logísticos ou técnicos, nomeadamente notificações de falha de serviço.

- Portabilidade: Os componentes *Web* do sistema deve ser compatíveis com as normas abertas, emanadas de entidades como o W3C, de forma a conseguir uma cobertura universal de utilizadores.

- Disponibilidade: O sistema deve manter-se operacional 24 sobre 24 horas, 365 dias por ano com um número reduzido de interrupções.

5 Análise e desenho da Solução

Neste capítulo é feita a modelação do sistema de informação. Foram utilizados os diagramas de casos de uso (subsecção 5.1), diagramas de actividades (subsecção 5.2), diagramas de estados (subsecção 5.3) e diagrama de classes (subsecção 5.4).

5.1 Diagramas de Casos de Uso

Os diagramas de caso de uso são a descrição de um conjunto de sequências de acções (incluindo variantes) que um sistema realiza e que produzem um resultado observável por um actor. Estes diagramas são utilizados para representar a interacção do sistema como um todo e o seu relacionamento com o meio externo. Esta interacção ajuda a delimitar o sistema e a definir melhor o que o sistema deve fazer. [15]

Nas Figura 2, Figura 3 e Figura 4 estão representados os casos de uso mais importantes que fazem parte do sistema.

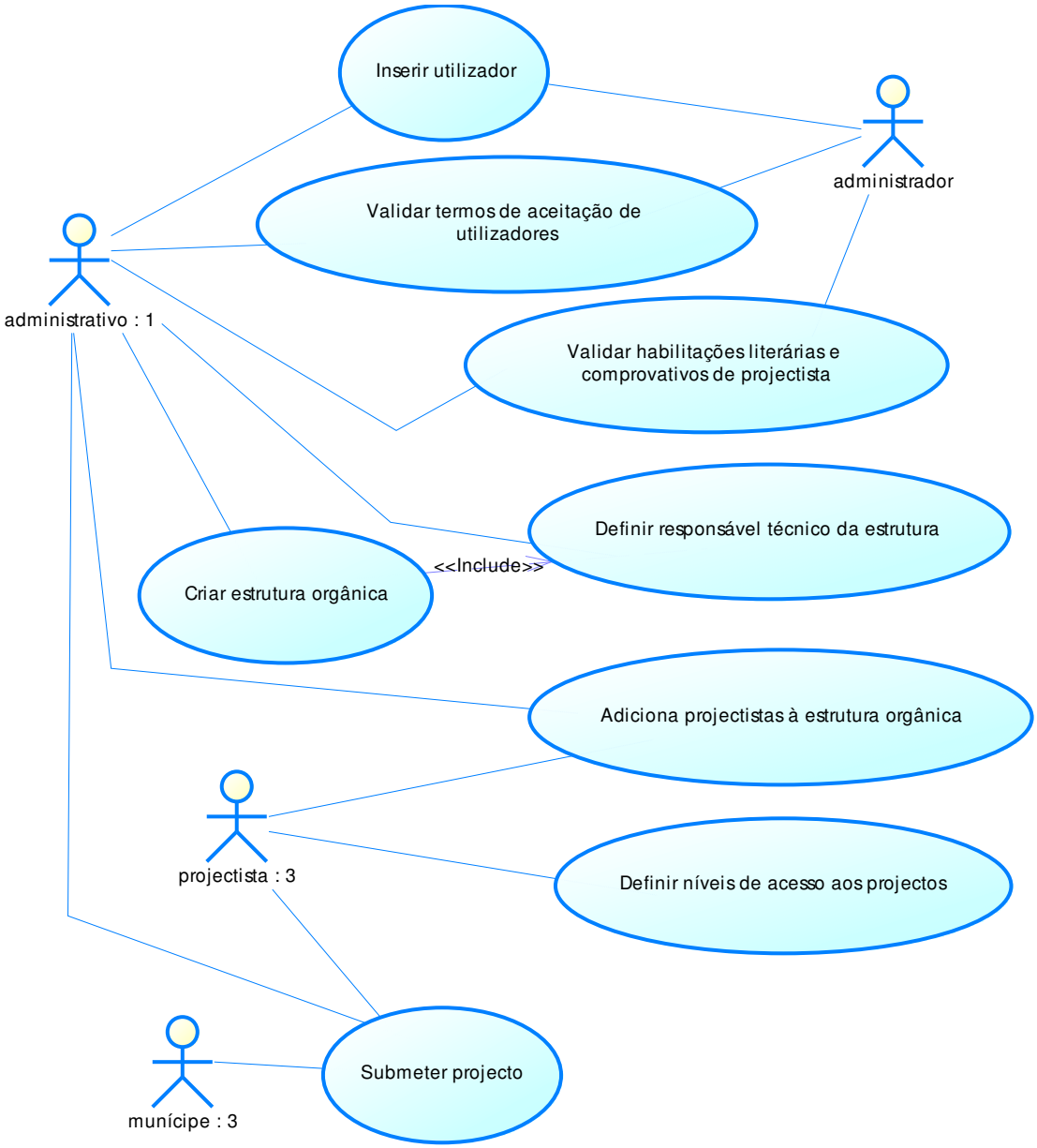


Figura 2: Diagramas de caso de uso - Parte I

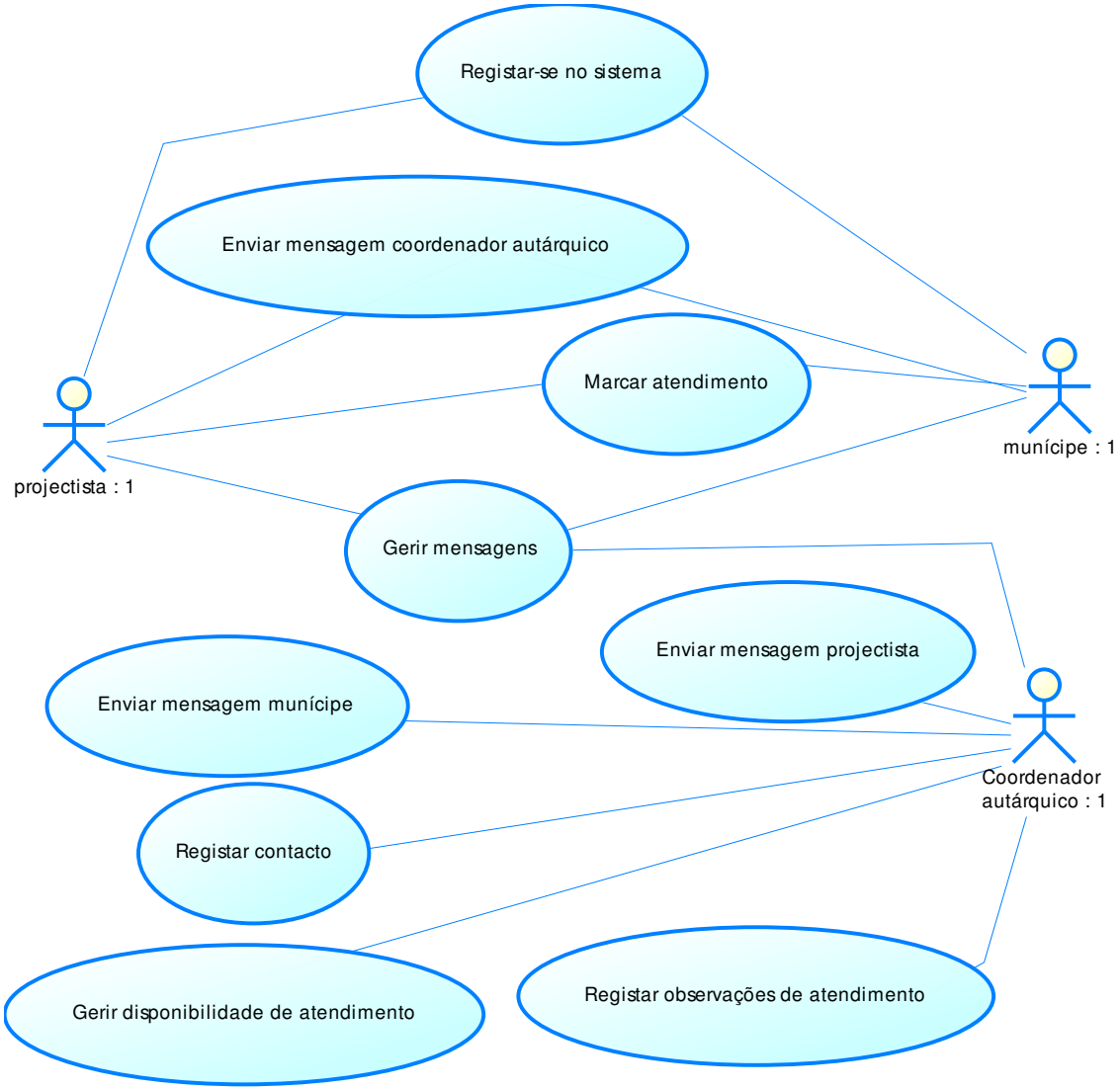


Figura 3: Diagramas de caso de uso – Parte II

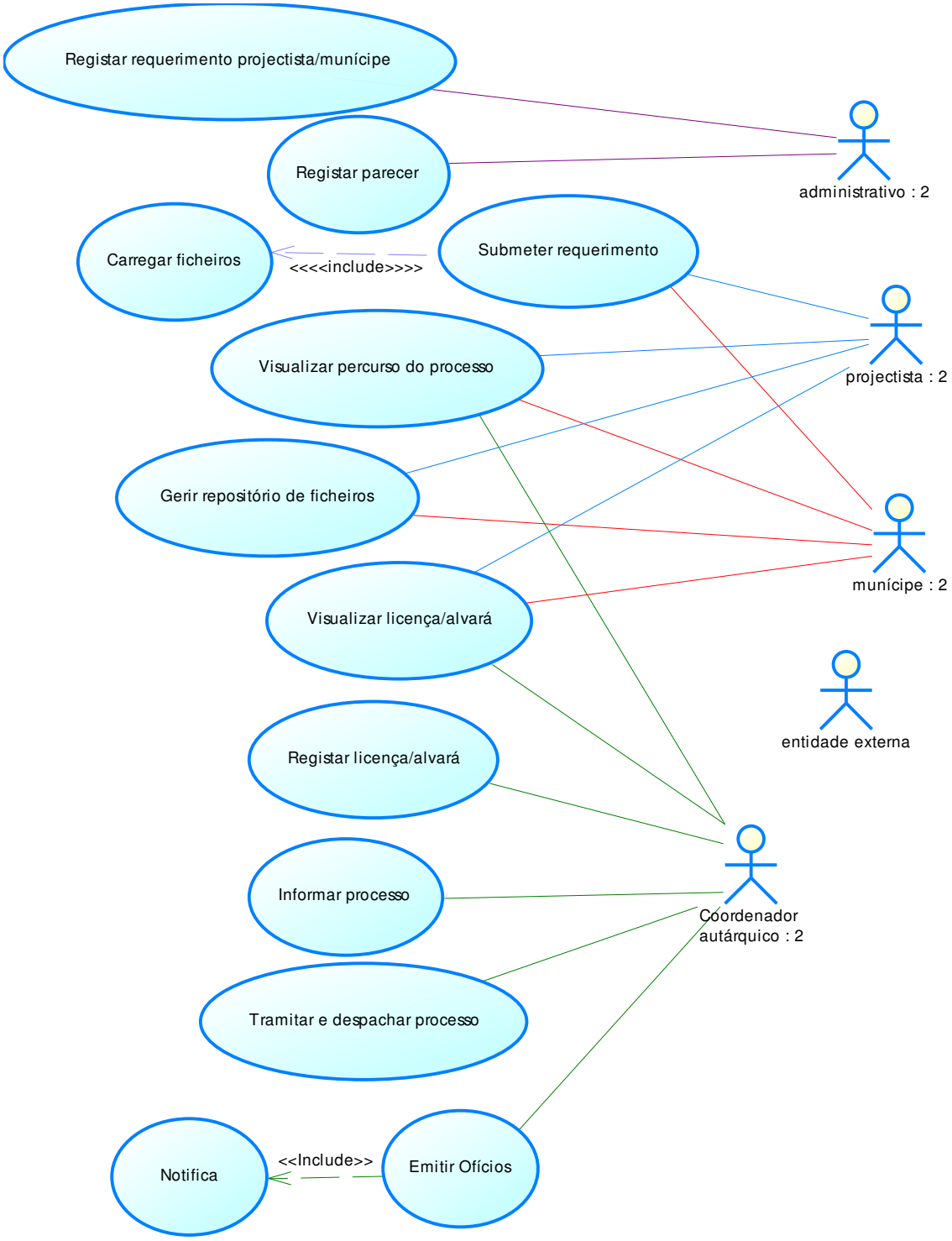


Figura 4: Diagramas de caso de uso – Parte III

5.1.1 Descritor dos casos de uso

A descrição textual seguinte dos casos de utilização é feita segundo o formato de nome, pré-condições, cenário-principal e cenários-alternativos.

A descrição dos casos de uso deve promover a simplicidade dos diagramas focando-se nos aspectos importantes do sistema [17]. Neste sentido, não foi especificado que todos os casos têm como cenário alternativo a possibilidade de cancelar qualquer operação a qualquer momento. De seguida são apresentados as descrições dos casos de uso de submeter projecto, registar-se no sistema e submeter requerimento. Para mais detalhes relativos ao descritores de casos de usos consultar o Anexo C Descritores dos Casos de Usos.

Nome: Submeter projecto

Cenário Principal: É apresentado um ecrã em que o munícipe preenche todos os dados do processo. O projecto é submetido. É associado um novo número de processo e a data actual do sistema.

Nome: Registar-se no sistema

Pré-condição: Certificado digital de autenticação

Cenário Principal: É apresentado um ecrã onde devem ser preenchidos os dados do titular, o tipo de perfil, tipo de notificação, nome de utilizador e palavra-chave. O pedido é submetido.

Nome: Submeter requerimento

Pré-condição: Munícipe

Cenário Principal: É apresentado um ecrã onde é preenchido a descrição do requerimento. É-lhe associada a data do sistema. É seleccionado do repositório de ficheiros os que correspondem ao requerimento. O requerimento é submetido.

5.2 Diagrama de Actividades

O diagrama de actividades permite visualizar o fluxo de controlo sequencial ou concorrencial de uma actividade para outra. As actividades podem ser decompostas em sub-actividades podendo chegar a acções atómicas [15]. O diagrama de actividades pode ser comparado a um fluxograma com concorrência.

Na Figura 5 é possível visualizar o diagrama de actividades referente à entrega de projectos para construção. O fluxo inicia-se pelo pedido de planta de localização do PDM por parte do munícipe ao Município. Depois de preparado e entregue o projecto, a Câmara faz uma avaliação ao projecto. Caso seja indeferido, o munícipe terá que entregar aditamentos até que seja deferido de forma a poder prosseguir com o processo. Depois de avaliado, segue-se a entrega das especialidades, directamente no Município, e dos pareceres, através de consultas externas. No caso do parecer não ser favorável, o requerente e o técnico são notificados e devem proceder com as rectificações necessárias e pedir novo parecer. No caso de ser favorável, são também notificados o requerente e o técnico, e o município verifica se o processo está completo e todos os pareceres são favoráveis. Em caso negativo, faz um pedido de documentos em falta; em caso positivo, segue para o processamento de taxas e emissão de alvará de construção. A obra é iniciada e no final de construção é pedido o alvará de utilização. No caso de ser uma habitação, o alvará é precedido de um pedido de vistoria; no caso da vistoria não ser deferida, o munícipe deve proceder às correcções na obra.

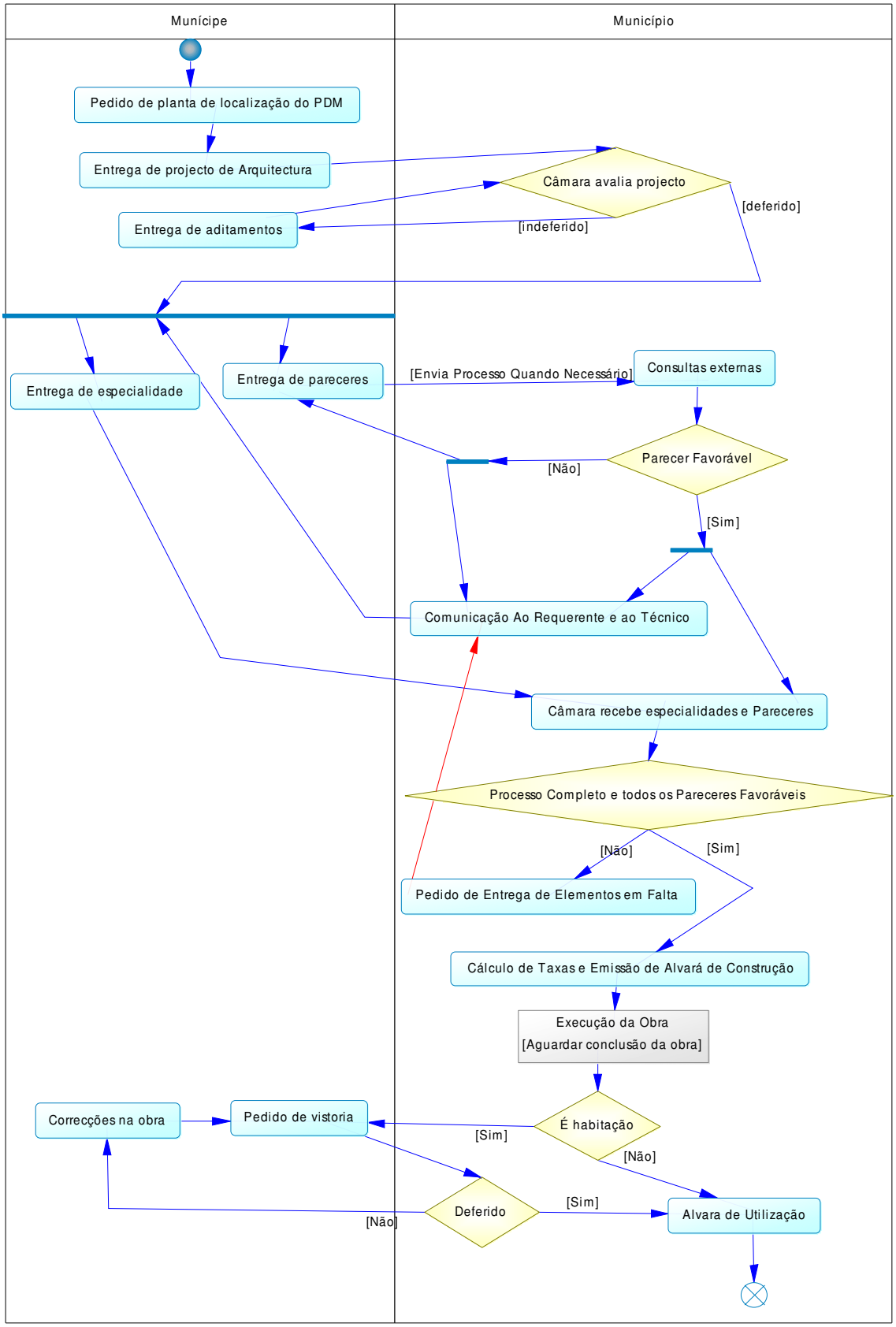


Figura 5: Diagrama de actividades - Entrega de projectos para construção

5.3 Diagrama de estados

Os diagramas de gráfico de estados permitem representar a parte dinâmica de um sistema ou objecto cujo estado evolui por saltos em resposta a eventos, com um número finito de estados; ou seja, um diagrama de estados especifica uma máquina de estados, com estados que os objectos podem assumir (duradoiros e em número finito) e transições entre estados causadas por eventos. O diagrama pode também especificar as acções e actividades realizadas em resposta a eventos ou durante a permanência em estados, respectivamente. Um estado é uma condição ou situação na vida de um objecto, durante a qual o objecto satisfaz alguma condição, realiza alguma actividade ou espera por algum evento. Um evento é uma ocorrência significativa que tem uma localização no tempo (instante de tempo do evento) e no espaço [15].

Na Figura 6 é possível visualizar o diagrama de estados referente aos processos de urbanização e edificação onde é possível avaliar os vários estados por que o processo pode passar.

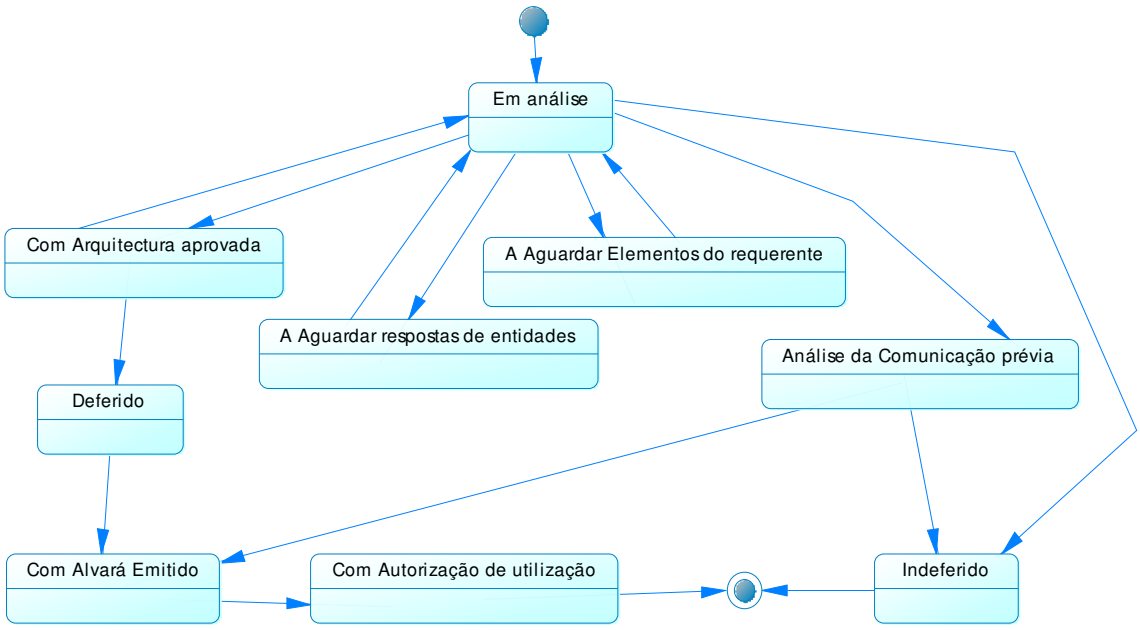


Figura 6: Diagrama de estados dos processo de urbanização e edificação

5.4 Diagrama de Classes

Um diagrama de classe é um diagrama que mostra um conjunto de classes (definição de objectos que partilham os mesmo atributos, operações, relações e semântica), interfaces (colecção de operações que são usadas para especificar o serviço de uma classe ou componente) e colaborações (comportamento cooperativo de um conjunto de classes, interfaces e outros elementos) e seus relacionamentos (dependências, generalizações e associações) [15].

Dado o elevado número de classes e ligações existentes o diagrama foi dividido em duas partes que se encontram em dois anexos: uma, com as classes do protótipo sem as ligações entre elas (Anexo D) e outra, para ajudar a sua compreensão, com a listagem tabular de todas as ligações (Anexo E).

6 Implementação

Neste capítulo são revistas as opções tecnológicas tomadas no projecto (secção 6.1). De seguida é apresentado o modelo físico de dados (secção 6.2) e o diagrama de instalação. Por fim é apresentada uma breve descrição do desenvolvimento do projecto e os resultados da sua avaliação experimental.

6.1 Opções tecnológicas

Após analisar os requisitos é necessário tomar decisões no que respeita à tecnologia a utilizar. Dessa análise resulta as seguintes necessidades tecnológicas: um servidor HTTP (HyperText Transport Protocol), uma linguagem de programação "script" para páginas dinâmicas, uma linguagem de programação de interacção do lado do cliente, uma "framework" de desenvolvimento e, por fim, um sistema de gestão de base de dados. Abaixo são explicadas as opções tomadas em cada uma das situações.

6.1.1 Apache HTTP

Como servidor de HTTP foi escolhido o Servidor Apache, um projecto da Apache Software Foundation[10]. Este projecto é desenvolvido com o propósito de manter um servidor HTTP "open-source", multi-plataforma, seguro, eficiente, extensível e que

[10] http://www.apache.org/

respeite as normas abertas do protocolo HTTP[18]. Em Janeiro de 2009, de acordo com um inquérito realizado pela Netcraft, o Apache era utilizado em 52 % das máquinas dos programadores para a Internet e em 50 % dos sítios activos em todo o mundo[19].

A razão que leva à escolha deste servidor é a natureza do projecto, mais concretamente, a não limitação na plataforma a utilizar, a extensibilidade e o tipo de licença “open-source”.

A versão utilizada no ambiente de desenvolvimento foi a versão 2.2.10.

6.1.2 PHP e Java

Da panóplia de linguagens de programação existentes para a geração de páginas dinâmicas era necessário escolher uma de acordo com a metodologia adoptada. A escolha recaiu sobre o PHP: um acrónimo recursivo para "PHP: Hypertext Preprocessor". Uma vez mais, a opção é fundamentada tendo em conta as características da tecnologia. O PHP é uma linguagem “script” executada do lado do servidor, multi-plataforma, orientada a objectos, extensível, pouco exigente a nível de recursos, com documentação extensa, de aprendizagem rápida e “open-source”[20].

A versão utilizada no ambiente de desenvolvimento foi a versão 5.2.6.

Para além das páginas dinâmicas geradas do lado do servidor era necessário criar interactividade do lado do cliente, mais concretamente, disponibilizar a funcionalidade de assinar digitalmente ficheiros no computador cliente e, para isso, era necessária interoperabilidade com o Cartão de Cidadão. De um leque reduzido de opções, a linguagem escolhida recaiu sobre o Java, utilizando “applets” por definir, desde a versão 1.5, um conjunto de interfaces de programação que permite a realização de operações criptográficas. Além disso é “open-source”[21].

A versão utilizada no ambiente de desenvolvimento foi a versão 1.6.0_11.

6.1.3 Codeigniter

O critério na selecção de uma “framework” a utilizar para implementar o protótipo foi o de garantir que a curva de tempo na aprendizagem para beneficiar das funcionalidades deveria ser inferior ao de construir algo de raiz e que permitisse separar os dados da lógica e da visualização.

A “framework” utilizada no ambiente de desenvolvimento foi o Codeigniter 1.7.0, que é ”open-source”, construída em PHP, leve, rápida, extensível, documentada e baseada na abordagem “Model-View-Controler” (MVC) [22].

O padrão de arquitectura MVC divide as aplicações interactivamente em três componentes: modelos, vistas e controlos. Os componentes de modelos encapsulam o núcleo de acesso aos dados e às funcionalidades, e são independentes da apresentação da informação e dos comportamentos na introdução de dados. Os componentes de vistas são responsáveis por mostrar a informação aos utilizadores e os componentes de controlo, manipulam os seus pedidos e a introdução de dados. A interface com o utilizador fica a cargo das vistas e dos controladores. Desta forma, as alterações feitas na apresentação de informação não afectam o modo como os dados são manipulados, tal como uma alteração na lógica não altera o modo como os dados são apresentados. De uma forma automática é mantido um mecanismo de consistência entre as interfaces e os modelos. [23]

6.1.4 MySQL

O MySQL é um sistema de gestão de bases de dados SQL (Structured Query Language). Entre outras características, as que levaram à sua adopção foram a velocidade de acesso, o suporte de multi-processos e multi-utilizador, a robustez, ser entidade referencial e suportar transacções. A sua velocidade e flexibilidade deriva do facto da informação ser armazenada em várias tabelas separadas em vez de as centralizar numa partição grande e única. Além disso, o MySQL pode ser visto como “open-source” em aplicações não comerciais[24].

6.1.5 Outras bibliotecas e aplicações

A aplicação desenvolvida para a carga de ficheiros para o sistema deriva da aplicação “open-source” JUpload 2 que inclui, entre outras funcionalidades, o “upload” de ficheiros por HTTP e FTP[25].

Para validar os certificados de autenticação do Cartão de Cidadão através do respectivo serviço de OCSP é utilizado o OpenSSL[11], um conjunto de ferramentas que implementa o “Secure Sockets Layer”.

[11] http://www.openssl.org

Para percorrer documentos HTML, tratar de eventos e efectuar interacções Ajax foi utilizado a biblioteca para JavaScript jQuery[12]. Como interface de desenvolvimento em PHP foi utilizado o VIM[13] e, para desenvolvimento em Java, o Eclipse[14].

6.2 Modelo Físico de Dados

O modelo físico de dados foi construído segundo o modelo Entidade-Relacionamento, um modelo de dados abstracto e conceptual de representação de dados. O modelo é constituído por entidades, que interagem umas com as outras, através de relacionamentos e está representado na Figura 7.

[12] http://jquery.com/

[13] http://www.vim.org/

[14] http://www.eclipse.org/

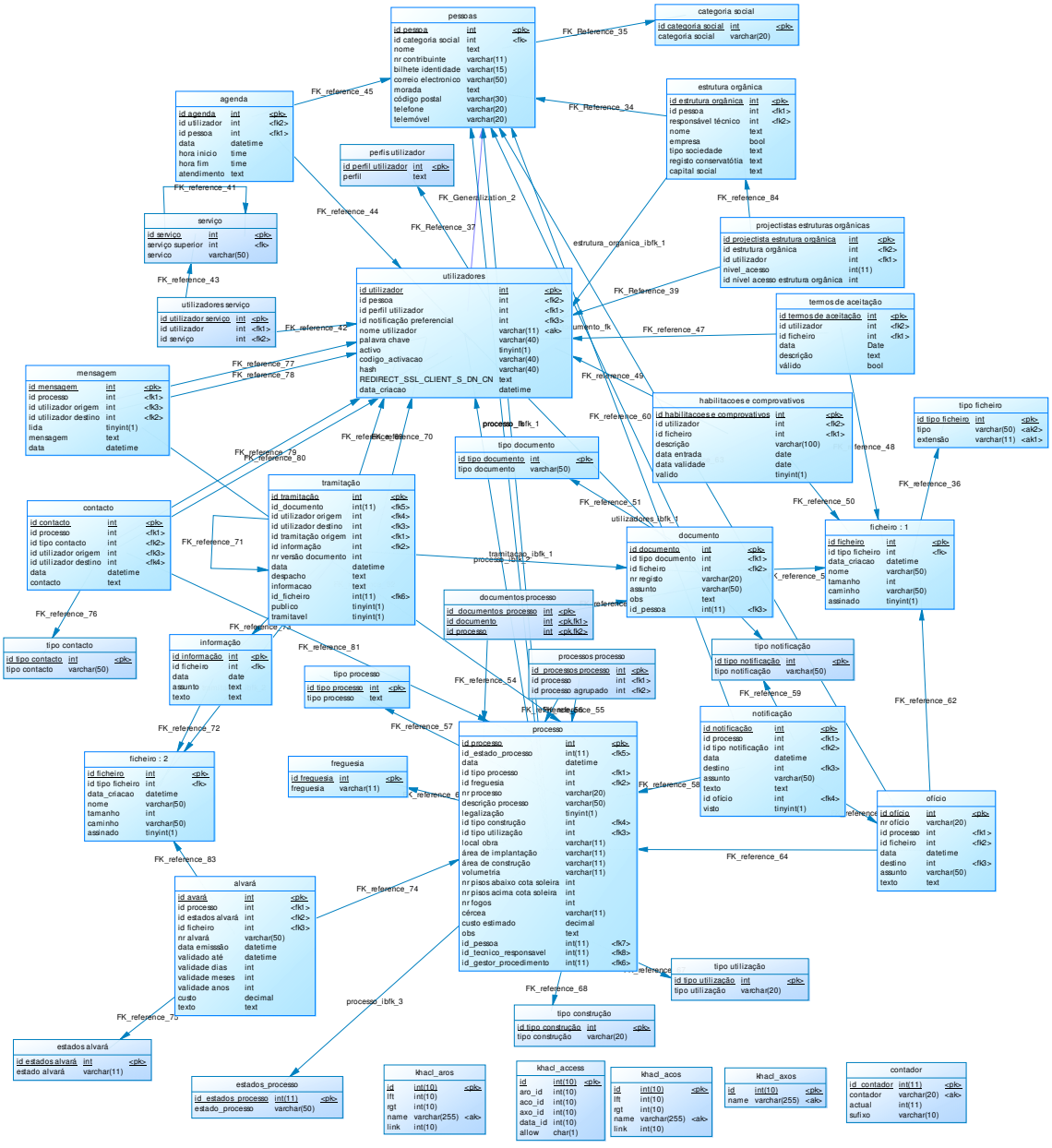


Figura 7: Modelo entidade referêncial da implementação do protótipo

Uma vez mais, o elevado número de tabelas num único diagrama dificulta a sua leitura no formato A4. No DVD em anexo é disponibilizado uma versão JPEG no tamanho real e, no Anexo F, uma descrição sucinta de cada uma das tabelas.

6.3 Diagrama de Instalação

O diagrama de instalação (Figura 8) permite visualizar a configuração de nós de execução e os componentes que coabitam com eles [15].

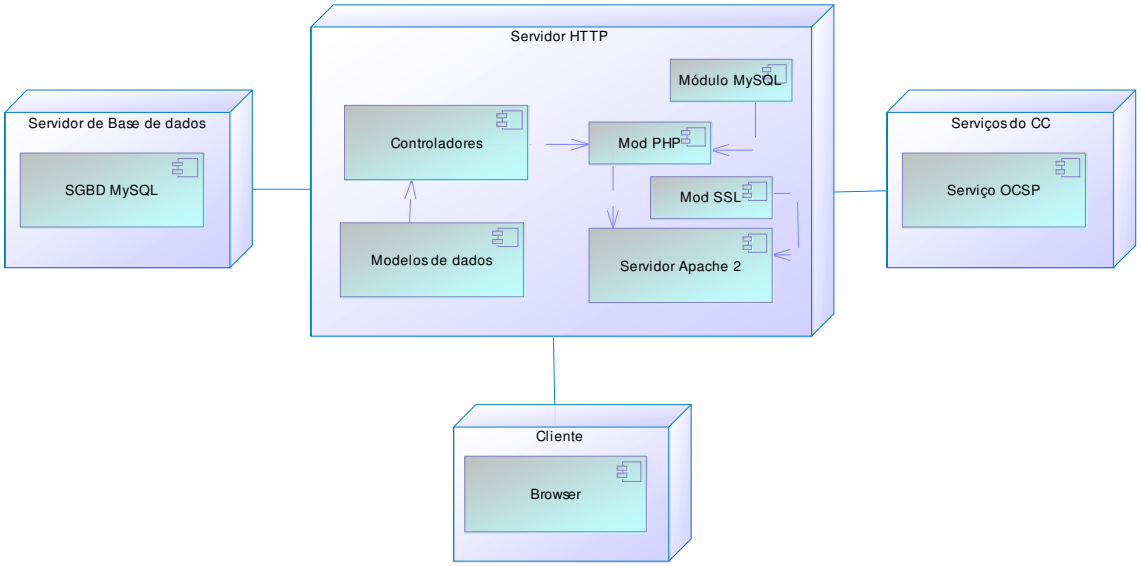


Figura 8: Diagrama de instalação do protótipo

O sistema é composto por quatro nós principais: o nó servidor http, que recebe e processa os pedidos do nó cliente, o nó servidor de base de dados onde são armazenados os dados e processados os pedidos do nó servidor HTTP, e o nó remoto de serviços do Cartão de Cidadão que permite validar, a pedido do nó servidor HTTP, os cartões utilizados nos pedidos de autenticação do nó cliente.

6.4 Desenvolvimento

Nesta secção é descrita a organização do código fonte dos ficheiros PHP e dos ficheiros Java e explicados os desenvolvimentos mais importantes. Todo o código é incluído no DVD-ROM em anexo. A descrição e a explicação das funcionalidades são feitas através do Anexo A, Manual do Utilizador, e as instruções de instalação no Anexo B, Manual Técnico de Instalação.

A denominação dada ao protótipo foi: “PLUD – Protótipo de Licenciamento Urbanístico Digital”.

A estrutura de pastas dos ficheiros PHP, descrita na Tabela 3, está organizada de acordo com as definições da “framework” utilizada: ao organizar cada categoria em diferentes pastas facilita a navegação e localização de ficheiros.

PASTA	CONTEÚDO
config	ficheiros de configuração
controllers	ficheiros de controle
errors	ficheiros personalizados de erros
helpers	ficheiros que agrupam uma determinada categoria colecções de funções
hooks	ficheiros de alteração de funcionalidades da “framework ” sem alterar o seu código-fonte
language	ficheiros de traduções
libraries	ficheiros de bibliotecas personalizadas
models	ficheiros de modelo
my_classes	ficheiros de classes importadas
views	ficheiros de vistas

Tabela 3: Organização de pastas do protótipo

Para responder às necessidades do sistema, para além da parametrização das configurações da “framework” utilizada, foram criadas outras que se enumeram de seguida com as respectivas personalizações ao protótipo.

```
//Configurações gerais e de email
$config['site_name'] = 'PLUD :: Jesus Test'; //nome do site
$config['email_address'] = 'jesuscampos@maisbarcelos.pt'; //endereço de
email
$config['email_name'] = 'PLUD'; //nome para o nome
$config['smtp'] ='localhost'; //servidor smtp
$config['smtp_username'] = plud''; //credencial para o smtp
$config['smtp_password'] = 'plud'; //credencial para o smtp

//Configurações de autenticação
$config['encryption_key'] = "ga4rud7e83swEphUTrat"; // Chave para
encriptação das passwords

//Configurações de ficheiros
$config['files_path_system'] = /var/html/plud/files/system/'; //caminho para
armazenar os ficheiros do sistema
$config['files_path_users'] = /var/html/plud/files/users/'; //caminho para
armazenar os ficheiros dos repositórios dos utilizadores
$config['RootCCCA'] = /var/html/plud/root.pem'; //caminho para o certificado da
autoridade certificadora do CC
$config['CCOCSPUrl'] = 'http://ocsp.auc.cartaodecidadao.pt/publico/ocsp'; //
URL do serviço OCSP do CC
$config['opensslPath'] = "/usr/bin/openssl"; //Caminho para o openssl
```

A configuração da verificação de certificado na autenticação do cliente e o carregamento da informação do Cartão de Cidadão são feitos através de directivas no ficheiro de configuração do Apache como mostra o exemplo a seguir.

```
<Location /plud/profile/ccauth/>
        SSLRequireSSL
        SSLVerifyClient optional_no_ca
        SSLOptions +ExportCertData
        SSLOptions +StdEnvVars
</Location>
```

As alterações efectuadas na aplicação desenvolvida para a carga de ficheiros foram feitas directamente no código fonte do JUpload 2, excepto a pasta PKCS11signVerify criada, que contém os seguintes ficheiros:

- PKCS11signVerify.java – Classe Interface para assinatura e verificação de ficheiros.
- CC.java – Implementação da classe anterior com interoperabilidade com o cartão de cidadão.

A compilação da aplicação resulta num único ficheiro o qual deverá ser usado para a chamada da "applet". No protótipo foi utilizado o seguinte código HTML para efectuar essa chamada.

```
<APPLET codebase="<?php echo base_url(); ?>jars/"
CODE="wjhk.jupload2.JUploadApplet" NAME="JUpload" ARCHIVE="<?php
echo base_url(); ?>jars/wjhk.jupload.jar" WIDTH="576" HEIGHT="230"
MAYSCRIPT></XMP>
   <PARAM NAME = CODE VALUE = "wjhk.jupload2.JUploadApplet" >
   <PARAM NAME = ARCHIVE VALUE = "<?php echo base_url();
?>jars/wjhk.jupload.jar" >
   <PARAM NAME="type" VALUE="application/x-java-applet;version=1.4">
   <PARAM NAME="scriptable" VALUE="false">
   <PARAM NAME = "postURL" VALUE ="<?php echo base_url();
?>profile/upload_file">
   <PARAM NAME = "nbFilesPerRequest" VALUE ="1">
Java 1.5 or higher plugin required.
</APPLET>
```

6.5 Avaliação Experimental

A avaliação experimental limitou-se aos testes de desenvolvimento na programação e a testes de utilização por parte de alunos do ensino de arquitectura. Sempre que era encontrado um erro na implementação de uma nova funcionalidade, era criada uma nova versão e o erro era corrigido no momento. Só desta forma foi possível assegurar que, se numa alteração para correcção de um erro se corrompesse o comportamento de outras funcionalidades, uma nova versão ao sistema minimizasse tais efeitos negativos.

O comportamento da aplicação foi testado em diferentes ambientes. No caso do servidor, em várias máquinas de teste com diferentes configurações de “hardware” e de sistema operativo. No caso do cliente foi testada em “browsers” diferentes.

O cronograma do projecto previa uma avaliação experimental num gabinete de desenho técnico para avaliar o “feedback” dos utilizadores e assim aperfeiçoar o projecto com detalhes técnicos que pudessem ter sido esquecidos ou mal implementados. Porém, por limitações temporais, tal acabou por não ser feito.

O cronograma do projecto previa também a execução de testes de qualidade de software, tais como testes unitários, o que também não aconteceu por manifesta falta de tempo.

7 Conclusão

Por fim, este capítulo faz uma análise aos resultados obtidos (Secção 7.1) e descreve orientações para trabalhos futuros (Secção 7.2).

7.1 Resultados Obtidos

Na presente secção é feita uma avaliação dos resultados obtidos tendo em consideração os objectivos propostos e os objectivos realmente alcançados, salientando-se as contribuições feitas na área em estudo.

Pode concluir-se que o objectivo principal deste trabalho foi alcançado. O protótipo construído é totalmente funcional e operacional e foi construído totalmente com ferramentas livres. Além disso, testes efectuados por alunos no ensino de arquitectura deram resultados favoráveis quanto à ergonomia, interface, e simplicidade.

As opções tomadas nas tecnologias revelaram-se acertadas. As licenças utilizadas permitiram, além da utilização totalmente livre, visualizar e alterar o código fonte. Por outro lado, a selecção das ferramentas revelou-se na etapa do projecto que mais se alongou no tempo previsto para a sua execução, derivado da experimentação de várias “frameworks” de desenvolvimento em várias linguagens de programação.

Na fase de desenvolvimento foram encontradas algumas dificuldades em validar o certificado de autenticação do Cartão de Cidadão no servidor Apache. Optou-se em

delegar essa funcionalidade ao conjunto de ferramentas OpenSSL que, ao contrário do servidor Apache, consegue fazer validações pelo serviço OCSP.

Os diagramas UML utilizados mostraram-se uma valiosa ferramenta na análise e desenho do projecto. Permitiram organizar a informação criando um fio orientador a todo o desenvolvimento do projecto.

7.2 Orientações para Futuros Trabalhos de Investigação

A implementação do protótipo focou-se na concretização de todas as funcionalidades propostas e descritas nos requisitos; no entanto, não foram efectuados todos os testes de que estes tipos de sistemas carecem. Assim, pode considerar-se como trabalho futuro a implementação de testes unitários e a experimentação do sistema num gabinete projectista piloto para detectar eventuais falhas de desenho e especificação.

Em termos de segurança informática, seria muito interessante estudar-se a forma de limitar ao máximo os poderes dos administradores do sistema, de tal maneira que se tornasse difícil subverter o funcionamento correcto e honesto do sistema por acção maliciosa de um utilizador privilegiado ou o seu conluio com um cliente munícipe. Uma linha de acção que atendesse a este tipo de situação passaria sempre pela exigência de extensivos registos das operações realizadas pelo sistema, e o seu armazenamento em meios físicos não susceptíveis de alterações posteriores.

Outro aspecto a desenvolver encontra-se relacionado com o pedido de plantas de localização do plano director municipal. Para que seja, na sua totalidade, possível submeter os projectos de arquitectura sem deslocações aos Municípios, os requerentes, além de apenas visualizar, devem ser capazes de requerer novas plantas de localização.

A 2 de Setembro de 2008 foi solicitado à Direcção Geral das Autarquias Locais os requisitos necessários à criação de interoperabilidade deste sistema com o sistema de informação deles. Até à data de entrega desta dissertação não foi enviada qualquer documentação técnica que possibilite a sua concretização. É de todo importante a implementação desta funcionalidade quando a Interface de Programação de Aplicativos for tornada pública.

Por estar fora do âmbito deste projecto, não foram considerados os cálculos de todas as taxas inerentes ao arranque de um processo de pedido de apreciação de arquitectura urbanística e seus trâmites. De qualquer modo seria enriquecedor a sua integração.

Por fim, para tornar um sistema fortemente automatizado, a inclusão de um repositório de regras de validação poderia, depois de criadas as condições necessárias, pré-validar todos os desenhos de arquitectura submetidos para apreciação. Desta forma, sempre que o sistema encontrasse nesses ficheiros uma violação aos regulamentos municipais, estaria a diminuir o número de trâmites processuais e o tempo decorrido desde a entrega até à emissão da licença.

Referencias Bibliográficas

1. EUROSAI, G.d.T.d. (2005) *A governação electrónica na perspectiva da auditoria*.
2. Ferreira, J.R.d.C., *As Tecnologias de Informação Geográfica na Sociedade da Informação Do e-Gov ao e-Citizen*: Universidade Nova de Lisboa.
3. Gouveia, L.B., *A Administração Pública Local de Base Electrónica : Questões e Desafios*, Porto: Faculdade de Ciência e Tecnologia – Universidade Fernando Pessoa.
4. Gouveia, L.B., *Cidades e Regiões Digitais: impacto nas cidades e nas pessoas*, ed. E.U.F. Pessoa. 2003.
5. Gouveia, L.B., *Local e-government - a governação digital na autarquia*, Porto: Sociedade Portuguesa de Inovação.
6. *AIRC - Associação Informática da Região Centro*. 2009 09/03/2009]; Disponível em: http://portal.airc.pt/lwp/wcm/connect/AIRC.
7. *Formato de arquivo*. 2008 [cited 05/03/2008; Disponível em: http://pt.wikipedia.org/wiki/Formato_de_arquivo.
8. Durand, A. 2007 14/09/2007]; Disponível em: http://www.digitalidworld.com/modules.php?op=modload&name=News&file=article&sid=26.
9. Windley, P.J., *Digital identity*. 1st ed. 2005, Beijing ; Sebastopol, CA: O'Reilly. xviii, 234.
10. Ramsdell, B. *RFC2633 - S/MIME Version 3 Message Specification*. 1999 19/02/2009]; Disponível em: http://www.ietf.org/rfc/rfc2633.txt.
11. Mogollon, M., *Cryptography and security services : mechanisms and applications*. 2007, Hershey, PA: CyberTech Pub. xv, 471 p.
12. Rescorla, E. *HTTP Over TLS*. 2000 15/03/2009]; Disponível em: http://www.ietf.org/rfc/rfc2818.txt.
13. *Perguntas Frequentes* 2009 12/03/2009]; Disponível em: http://www.cartaodecidadao.pt/index.php?option=com_content&task=section&id=5&Itemid=35&lang=pt.
14. O'Docherty, M., *Object-oriented analysis and design : understanding system development with UML 2.0*. 2005, Chichester, England ; Hoboken, NJ: Wiley. xvii, 559 p.
15. Booch, G., J. Rumbaugh, and I. Jacobson, *The unified modeling language user guide*. The Addison-Wesley object technology series. 1999, Reading, MA [etc]: Addison Wesley. XXII, 482.
16. Sommerville, I., *Software engineering*. 7th ed. International computer science series. 2004, Boston: Pearson/Addison-Wesley. xxii, 759 p.
17. Silva, A.M.R.d. and C.A.E. Videira, *UML, metodologias e ferramentas CASE linguagem de modelação UML, metodologias e ferramentas CASE na concepção e desenvolvimento de software*. 2ª ed ed. Tecnologias. 2005, Lisboa: Centro Atlântico. _vol.
18. *Apache HTTP Server Project*. 2009 13/03/2009]; Disponível em: http://httpd.apache.org/.
19. Netcraft. *February 2009 Web Server Survey*. 2009 13/03/2009]; Disponível em: http://news.netcraft.com/archives/2009/02/18/february_2009_web_server_survey.html.
20. Achour, M., et al. *PHP Manual*. 2009 13/03/2009]; Disponível em: http://www.php.net/manual/en/index.php.
21. *JavaTM PKCS#11 Reference Guide*. 2004 13/03/2009]; Disponível em: http://java.sun.com/j2se/1.5.0/docs/guide/security/p11guide.html.

22. *CodeIgniter User Guide Version 1.7.1*. 2009 14/03/2009]; Disponível em: http://codeigniter.com/user_guide/.
23. Buschmann, F., *Pattern-oriented software architecture : a system of patterns*. 1996, Chichester ; New York: Wiley. xvi, 457 p.
24. *MySQL Documentation*. 2009 14/03/2009]; Disponível em: http://dev.mysql.com/doc/.
25. *JUpload - File Upload Applet*. 2009 22/01/2009]; Disponível em: http://sourceforge.net/projects/jupload/.

Anexo A – Manual do Utilizador

Índice

1. Introdução

O PLUD - Protótipo de Licenciamento Urbanístico Digital é uma aplicação que faculta aos municípios a gestão documental de processos de urbanização e edificação de uma forma electrónica e totalmente desmaterializada.

O tipo de autenticação que o sistema exige, que obriga à introdução do nome de utilizador, palavra-chave e apresentação do Cartão de Cidadão (CC) com certificado digital de autenticação, assim como a assinatura digital qualificada dos ficheiros carregados, garantem a segurança do sistema e protegem os documentos introduzidos de alterações indevidas. Além da segurança, a plataforma oferece outras vantagens, entre as quais se destacam:

- a possibilidade de qualquer munícipe poder submeter todo um processo nesta plataforma;
- a disponibilidade a tempo inteiro da ferramenta, dado estar disponível na web;
- a dispensa de deslocações, muitas vezes morosas, desnecessárias ou infrutíferas;
- a diminuição da utilização de papel durante todo o processo;
- a diminuição do tempo decorrido desde o início até ao total término do processo;
- a facilidade de transporte de documentos;
- as notificações automáticas, mais simples e eficazes, através de correio electrónico.

1.1. Requisitos para acesso ao sistema

Tal como mencionado na documentação do CC, o navegador a partir do qual se acede ao sistema PLUD deverá ter suporte à biblioteca PKCS11 e a JavaScript.

Nota: antes de iniciar cada sessão no sistema tem de introduzir o seu CC no leitor de cartões do computador em que pretende aceder.

2. Registo, Autenticação e Edição de Utilizadores

Os utilizadores com acesso ao sistema PLUD podem ser criados de duas formas: através de registo de utilizadores pela interface Web e pela criação directa no sistema pelos administradores do sistema.

2.1. Registo de Utilizadores via interface web

Quando um Utilizador é criado mediante registo pela interface da plataforma, só podem ser escolhidos os perfis "Projectista" e "Munícipe". Para fazer parte dos restantes perfis, o Utilizador tem de ser criado directamente no sistema.

Para registar com sucesso um Utilizador na plataforma, devem cumprir-se TODOS os seguintes requisitos:

- Possuir um CC válido (que tem de ser ligado ao computador antes do início do processo de registo, e não pode ser desligado até ao seu término), do qual o sistema fará a leitura do Certificado Digital de Autenticação e do nome completo da pessoa associada ao utilizador a criar;
- Preencher TODOS os campos do registo;
- Introduzir um endereço de correio electrónico válido. Será enviado automaticamente uma mensagem electrónica para o endereço que introduziu. Para confirmar a conta deverá clicar na hiper-ligação contida na mensagem. De forma a manter íntegros os dados do sistema, os administradores apenas deverão activar os utilizadores que confirmem a recepção de mensagem de correio electrónico;

Após o registo e activação, para uma correcta e total utilização do sistema, o utilizador deve, na sua área de perfil, submeter o ficheiro de termos de aceitação assinado digitalmente, previamente descarregado no formulário de submissão. Os utilizadores com o perfil de projectistas devem ainda submeter, também na área de perfil, os comprovativos das suas habilitações literárias.

2.2. Criação de utilizadores directamente no sistema

Os utilizadores com permissões de administração de utilizadores podem criar directamente no sistema contas de utilizadores de qualquer um dos perfis. Para isso deverão proceder da seguinte forma:

- Certificar-se de que já foi criada a pessoa a associar ao utilizador a criar. No caso de não existir, consulte o Capítulo 5 para saber como proceder;
- A Conta de utilizador poderá ser activada de imediato ou poderá optar por activá-la mais tarde, por exemplo, após confirmação do endereço de correio electrónico introduzido (para isso o Utilizador deve clicar no link da mensagem que recebeu do sistema);

Nota: quando um Administrador cria uma conta de utilizador, não lhe associa nenhum CC. O Utilizador tem 30 dias para proceder a essa associação ou deixará de poder autenticar-se no sistema.

2.3. Autenticação

Antes de proceder à autenticação no sistema, deverá ligar o Cartão do Cidadão associado à sua conta (no caso de já ter sido associado) ou que pretende associar (no caso de ainda não ter sido associado, tal deverá ser feito no prazo de 30 dias após a criação da conta).

Aceda ao sistema, digitando o URL no local apropriado do navegador. Surgirá o ecrã de Boas-Vindas, onde será pedida a introdução do nome de utilizador e da palavra-chave. Deve introduzir os dados indicados no registo, ou que foram dados pelo Administrador do sistema. Para aumentar a segurança pode utilizar o teclado virtual disponível para o preenchimento da palavra-chave.

Nota: após autenticação, em caso de inactividade superior ao período de tempo definido no servidor, a sessão expira, sendo necessário voltar a autenticar-se no sistema.

2.4. Tipos de Perfis de Utilizadores

O contexto do sistema é gerado de acordo com o perfil de utilizador a que cada um tem de estar associado, sendo a própria página inicial específica a cada tipo de utilizador (com excepção dos alertas comuns a todos os perfis, que são o de não ter CC associado e não ter Termos de Aceitação válidos). Os perfis de utilizadores pré-definidos no sistema são os seguintes:

2.4.1. Administrador

Como o próprio nome indica, são responsáveis pela administração e manutenção da plataforma, cabendo-lhes tarefas tais como a validação de utilizadores que tenham efectuado o registo "on-line" e que cumpram com todos os requisitos obrigatórios e criação de regras de permissões a utilizadores.

Na sua página inicial, além dos avisos comuns a todos os perfis de utilizadores, surgem os avisos de termos de aceitação, habilitações literárias e utilizadores não activos.

2.4.2. Administrativo

São responsáveis pela parte burocrática da plataforma, ou seja, introduzir todos os dados no sistema que ainda não foram submetidos, como é o caso de processos e documentos entregues em papel, validar os processos e documentos submetidos pelos munícipes e projectistas, atribui-los aos coordenadores autárquicos e introduzir demais informação gerada pelos processos.

Na sua página inicial, além dos avisos comuns a todos os perfis de utilizadores, surgem os avisos de termos de aceitação, habilitações literárias e utilizadores não activos.

2.4.3. Coordenador autárquico

Funcionários da autarquia responsáveis pela coordenação dos processos que lhes tenham sido atribuídos.

Na sua página inicial, além dos avisos comuns a todos os perfis de utilizadores, surgem os avisos referentes aos processos de que são coordenadores, bem como as marcações na agenda.

2.4.4. Projectista

É o responsável técnico do processo.

Na sua página inicial, além dos avisos comuns a todos os perfis de utilizadores, surgem os avisos referentes aos processos onde são técnicos responsáveis e o aviso de ausência de comprovativo de Habilitações Literárias. Têm a capacidade de adicionar ou remover utilizadores nas estruturas orgânicas em que são responsáveis.

2.4.5. Munícipe

Pessoa singular ou colectiva que recorre ao serviço para submeter um processo de licenciamento e todos os documentos inerentes.

Na sua página inicial, além dos alertas comuns a todos os perfis, são listados todos os processos onde são donos de obra. Têm a possibilidade de contactar, através de mensagem interna, os coordenadores autárquicos dos processos a que têm acesso.

2.5. Perfil pessoal

O Perfil pessoal de Utilizador é a área onde o Utilizador deve aceder sempre que queira consultar, completar ou alterar os seus dados pessoais (tais como morada, contacto, endereço de correio electrónico ou palavra-chave).

É aconselhável que, após o registo, o Utilizador aceda à área do seu perfil e preencha ou corrija todos os campos disponíveis (o tipo de perfil não pode ser alterado).

3. Acções gerais da plataforma

Na generalidade dos menus do sistema PLUD, é possível adicionar, procurar, modificar e eliminar dados (as permissões de cada utilizador variam mediante o contexto). A Imagem 1 representa um menu típico do PLUD, com legendas explicativas.

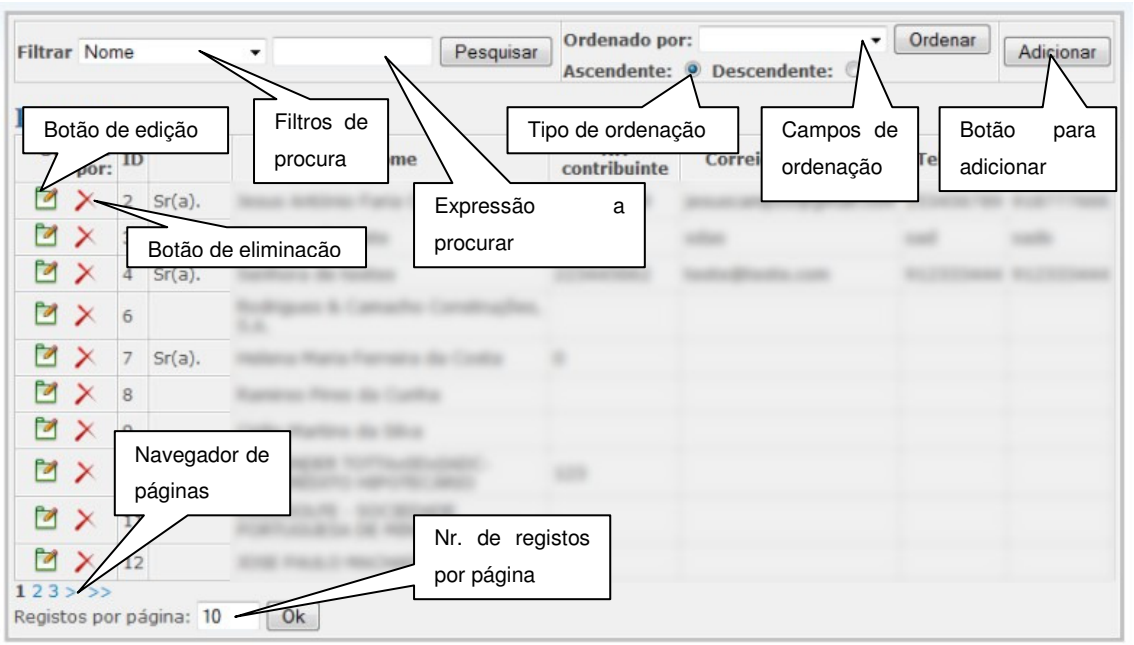


Imagem 1: Acções gerais da plataforma

3.1. Adicionar dados

Ao adicionar dados, deverá introduzi-los nos respectivos campos de preenchimento obrigatório (o sistema indica quais os campos que são por ele validados). Findo este passo, poderá: “Guardar” (na base de dados os dados introduzidos), “Restaurar Formulário” (aos dados pré-definidos) e “Voltar” (à página anterior cancelando a introdução de dados).

3.2. Procurar dados

Na procura de dados, tem de seleccionar o campo onde será aplicado o filtro de procura (a lista de campos será diferente, consoante o contexto), introduzir a expressão a procurar (sendo devolvidos todos os registos em que o campo contenha a expressão indicada) e premir o botão “Procurar”. Por omissão, os resultados são apresentados em grupos de 10 por página, número este que pode ser aumentado até um máximo de 500. É possível ordenar os dados por ordem ascendente ou descendente, podendo esta ordenação ser aplicada em qualquer um dos campos definidos para o contexto. No contexto de procura, poderá também “Adicionar” e “modificar ” os dados como se descreve a seguir;

3.3. Modificar dados

No contexto de Procura, pode modificar os dados existentes. Clicando no botão "Modificar" é apresentado o mesmo formulário da introdução de dados, onde pode proceder às alterações necessárias, e dispõe de todas as funcionalidades enumeradas no ponto 3.1.

3.4. Eliminar dados

No contexto de Procura, pode Eliminar os dados existentes, clicando no botão "Eliminar": é, então, pedida a confirmação de eliminação do registo seleccionado, ou para o cancelamento da acção.

4. Submissão de Ficheiros

Os documentos a usar no sistema têm de ser previamente carregados para o repositório pessoal de dados de cada utilizador. Para isso deve aceder à área de Ficheiros no menu superior direito.

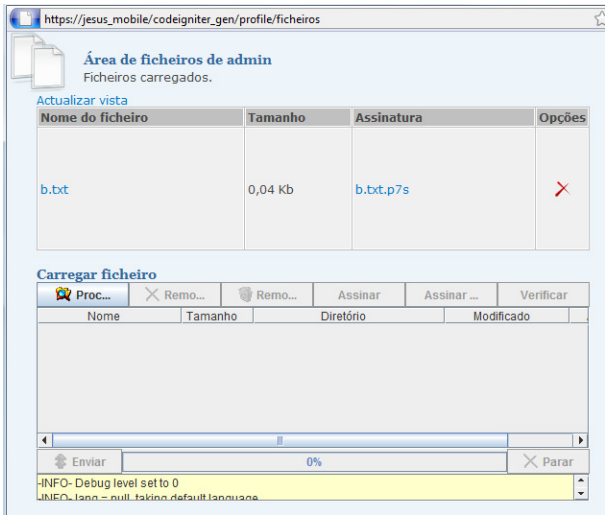


Imagem 2: Carga de ficheiros pessoais

Nota: só ficheiros assinados digitalmente serão carregados para o sistema.

Na área de Ficheiros, estão disponíveis os seguintes botões, na ordem apresentada:

- Procurar Ficheiro;
- Remover Ficheiro seleccionado;
- Remover Todos os Ficheiros;
- Assinar Digitalmente o Ficheiro seleccionado;
- Assinar Digitalmente todos os Ficheiros;
- Verificar Assinatura Digital do Ficheiro seleccionado;
- Enviar Ficheiros.

Para carregar ficheiro(s), deverá clicar no botão “Procurar” (inicialmente é o único botão acessível), que abrirá uma janela onde deve seleccionar o(s) ficheiro(s) a carregar.

Nota: se seleccionar uma pasta, serão carregados todos os ficheiros nela contidos. Sempre que seja necessário carregar mais de um ficheiro para o sistema, aconselha-se a que sejam todos colocados na mesma pasta, para ser mais rápida a sua selecção.

Depois de seleccionado um ou mais ficheiros, clique no botão “Enviar” para que sejam enviados para o sistema. Durante o processo de envio, pode acompanhar o progresso na barra de status existente nesta área. Sempre que um ficheiro tenha sido enviado com sucesso para o sistema, deixa de constar da lista de ficheiros a enviar. No final do processo, pode ver na parte superior da janela a lista de ficheiros actualmente existentes no sistema (clique em “Actualizar Vista” caso queira actualizar a lista).

A existência do campo “Log” permite, caso surja algum problema durante o processo de envio, uma rápida análise da natureza do erro.

5. Pessoas, Processos e Documentos

Os menus Pessoas, Processos e Documentos são os mais utilizados no dia-a-dia sendo através deles que o sistema é alimentado. As possíveis acções nestes menus são as comuns a todo o sistema (adicionar, procurar, modificar e eliminar dados, descritas detalhadamente no Capítulo 3 deste manual), sendo o acesso a essas mesmas acções definido pelo perfil e regras de acesso a que pertence cada utilizador.

5.1. Pessoas

As pessoas podem ser criadas de várias formas. No registo de utilizadores, quando um utilizador efectua um registo, o sistema verifica o nome da pessoa, número de contribuinte e bilhete de identidade e, caso não existam, é criada uma nova pessoa. Na submissão de novos processos por parte dos munícipes ou projectistas, quando o utilizador não encontra o nome da pessoa para associar ao dono da obra ou do técnico projectista, é dada a possibilidade de criar uma pessoa nova no momento. Todos os campos devem ser preenchidos. E no menu pessoas é possível navegar por todas as pessoas carregadas no sistema e só aqui é possível modificar ou eliminar dados.

5.2. Processos

Os processos podem ser criados de duas formas, tanto por parte dos munícipes como dos projectistas. Uma é através do painel inicial. Nele é apresentado um formulário com os dados do processo a preencher. Os processos submetidos são guardados e ficam à espera de confirmação de dados por parte dos administradores ou administrativos. Um processo não é considerado válido enquanto não for numerado. Enquanto o processo não é validado, o dono da obra e o técnico projectista do processo podem efectuar modificações.

Outra forma de criação de processos é através do menu Processos onde também é possível navegar por todos os processos pré-existentes, modifica-los, remove-los e activá-los (atribuir-lhes um número).

Na área “Modificação do processo”, é apresentado um submenu do lado direito onde é possível inserir dados relacionados com o processo:

- Alvarás - Registo dos alvarás do processo;
- Contactos - Registo dos contactos informais efectuados por parte dos munícipes ou projectistas, que faz sentido ligar ao processo;
- Documentos do processo - ligar os documentos carregados que fazem parte do processo;
- Notificações - é possível visualizar todas as modificações que foram criadas para este processo;
- Ofícios - Registo dos ofícios do processo;
- Processos anexos - no caso de um processo estar de alguma forma interligado a um ou mais processos, poder anexa-los.

5.3. Documentos

Os documentos podem também ser criados de duas formas, por parte dos munícipes ou projectistas. No painel inicial, para cada processo a que o utilizador tem acesso, é apresentado um conjunto de botões que permitem operações com o processo, dos quais uma é adicionar documento. É apresentado um formulário com os dados do documento a preencher. Os documentos submetidos são guardados e ficam à espera de confirmação de dados por parte dos administradores ou administrativos. Um documento não é considerado válido enquanto não for numerado. Enquanto o documento não é validado, o dono da obra e o técnico projectista do processo podem efectuar modificações. Na submissão do documento, este fica associado directamente ao processo seleccionado.

A outra forma é através do menu “Documentos” onde é possível navegar por todos os processos, modificar, remover e activá-los, ou seja atribuir-lhes um número.

Na área modificação do documento, é apresentado um submenu do lado direito onde é possível visualizar e, caso seja possível, tramitar o documento. O documento só

pode ser tramitado por qualquer utilizador com acesso caso ainda não tenha qualquer tramitação e, em caso afirmativo, apenas pelo destinatário da tramitação.

6. Tabelas do Sistema

Nas tabelas do sistema é possível parametrizar as tabelas de apoio ao sistema. As possíveis acções nestes menus são as comuns a todo o sistema (adicionar, procurar, modificar e eliminar dados, descritas detalhadamente no Capítulo 3 deste manual), sendo o acesso a essas mesmas acções definido pelo perfil a que pertence cada utilizador e respectivas regras de acesso. Segue uma pequena descrição de cada uma delas:

- Categorias sociais - permite gerir as categorias sociais associadas as pessoas.
- Contador - permite gerir a evolução dos contadores. A sua alteração intervém directamente com a numeração dos processos, documentos, ofícios e alvarás.
- Estados de alvará - permite gerir os vários estados dos alvarás.
- Estados do processo - permite gerir os vários estados dos processos.
- Estruturas orgânicas - permite gerir as estruturas orgânicas do sistema bem como definir o responsável de cada estrutura.
- Ficheiros - permite gerir todos os ficheiros carregados no sistema.
- Freguesias - permite gerir as freguesias afectas aos processos.
- Serviços - permite gerir os serviços do município. Na edição de cada serviço é possível gerir os utilizadores que pertencem a cada serviço.
- Tipos de construção - permite gerir os tipos de construção afectos aos processos.
- Tipos de contacto - permite gerir os tipos de contactos efectuados pelos munícipes ou projectistas e que foram registados num processo.
- Tipos de documento - permite gerir o tipo de documentos afectos aos documentos.
- Tipos de ficheiros - permite gerir os tipos de ficheiros carregados. No carregamento de um novo ficheiro, se o tipo não existir, é criado automaticamente.

- Tipos de notificação - permite gerir os tipos de notificações. Por omissão existem 2 tipos de notificação: “pela plataforma” e “pela plataforma e e-mail”. Neste último é enviado automaticamente uma mensagem de correio electrónico para o endereço de correio electrónico definido no perfil de utilizador.
- Tipos de processo - permite gerir os tipos de processo afectos aos processos.
- Tipos de utilização - permite definir os tipos de utilização afectos aos processos.

7. Administração

Existem 3 menus relativos à administração de utilizadores no sistema. São eles os “Perfis de utilizadores”, os “Utilizadores” e o “Controlo de acessos”.

7.1. Perfis de utilizadores

As possíveis acções neste menu são as comuns a todo o sistema (adicionar, procurar, modificar e eliminar dados, descritas detalhadamente no Capítulo 3), no entanto não é aconselhável efectuar quaisquer alterações visto que o sistema está preparado para operar com os perfis previamente carregados no sistema: Munícipe, Projectista, Coordenador autárquico, Administrativo, Administrador. Na edição de cada utilizador é possível gerir a agenda, termos de aceitação e habilitações literárias de cada utilizador.

7.2. Utilizadores

Neste menu é possível criar os utilizadores directamente no sistema. As possíveis acções neste menu são as comuns a todo o sistema (adicionar, procurar, modificar e eliminar dados, descritas detalhadamente no Capítulo 3). Na criação do utilizador é dado a possibilidade de activar e confirmar o utilizador logo no momento, ou de deixar a operação para mais tarde.

7.3. Controlo de acessos

Através do controlo de acesso é possível definir os acessos aos objectos. O controlo de acessos é composto por:

- ARO (*Access Request Object*), árvore hierárquica de quem requisita os objectos, ou seja os utilizadores. No primeiro nível, a raiz, são todos os utilizadores, no segundo, os perfis, são todos os utilizadores com aquele perfil, e no terceiro os utilizadores individualmente criados no sistema.

- ACO (Access Control Object), árvore hierárquica dos objectos a verificar os acessos. O primeiro nível é o sistema no seu todo e no segundo todos os objectos do sistema.

- AXO (Access eXtension Object), lista de tipo de acessos que o objecto pode receber.

- ID, identificador do objecto

- Tipo de acesso: permitido ou negado.

Uma regra é composta no mínimo por um ARO e um ACO e um tipo de acesso. Neste caso, todos os tipos de acesso e os respectivos identificadores ficarão agregados à regra. Por outro lado, nos casos em que é necessário refinar a regra, tal pode ser feito através da definição de um AXO ou, ainda, através do próprio identificador do objecto.

Nota: O identificador do objecto é dependente de um AXO.

Seguem-se três exemplos para melhor interpretação das regras de acesso:

- O administrador deve ter acesso a todo o site:

ARO: administrador	ACO: PLUD	AXO:	ID:	Tipo de acesso: SIM

- Os administrativos devem ter acesso a adicionar pessoas:

ARO: administrativo	ACO: pessoas	AXO: add	ID:	Tipo de acesso: SIM

- O utilizador jcampos deve ter acesso para modificar o documento com ID 123:

ARO: jcampos	ACO: documento	AXO: modify	ID:123	Tipo de acesso: SIM

8. Painel inicial, Caixa de tramitações e Mensagens

A interactividade entre os utilizadores do sistema reflecte-se, primordialmente, no painel inicial, caixa de tramitações e mensagens. É nestas áreas que utilizador é alertado para os assuntos pendentes.

8.1. Painel inicial

O painel inicial é a página que é apresentada logo após a autenticação do utilizador no sistema. Esta página também pode ser acedida clicando em início no menu superior direito.

Esta página é composta por:

- Alertas, que avisam de alguma incoerência nas definições do utilizador. O utilizador será alertado enquanto não carregar nenhum termo de utilização assinado e válido, não associar nenhum CC ao sistema e, no caso dos projectistas, não carregar nenhum comprovativo de habilitações literárias válido.

- Notificações, que informam os utilizadores das modificações efectuadas nos processos a que têm acesso, respectivos documentos e suas tramitações. No caso de um projectista, se este pertencer a alguma estrutura orgânica, receberá uma cópia das notificações do responsável da estrutura.

- Gestão de estruturas orgânicas, que permite aos projectistas gerir, ou seja, visualizar, adicionar e remover outros utilizadores das estruturas orgânicas nas quais são responsáveis.

- Lista de processos a que o utilizador tem acesso. No caso dos munícipes são os processos em que são donos da obra; no caso dos projectistas, os processos em que são técnicos responsáveis e, no caso de pertencerem a alguma estrutura orgânica, os processos de que são os responsáveis da estrutura. Por fim, no caso

de coordenador autárquico, os processos em que é gestor de procedimentos. Para cada processo está disponível um conjunto de operações a realizar com cada processo: ver ou editar processo (neste caso, se tiver permissões de modificação); últimos movimentos do processo; ver os trâmites dos documentos pertencentes ao processo por ordem descendente da data; ver documentos, com a possibilidade de visualizar ou adicionar novos documentos; contactar o gestor de procedimentos, com a possibilidade de enviar uma mensagem interna ao gestor de procedimentos; agendar reunião com o gestor de procedimentos.

- Utilizadores, Processos, Documentos, Termos de aceitação e Habilitações literárias que não estão verificados, no caso de utilizador ter o perfil de Administrador ou Administrativo.

- Gerir a agenda dos gestores de procedimentos. Os coordenadores autárquicos podem gerir as marcações por parte dos munícipes e projectistas e podem agendar novas reuniões.

8.2. Caixa de tramitações

A Caixa de tramitações está acessível no menu superior direito e permite aos utilizadores aceder às tramitações pendentes e a elas dirigidas e, caso sejam projectistas inseridos numa estrutura orgânica, as tramitações pendentes e dirigidas ao responsável da estrutura.

No caso de se tratar de um munícipe ou projectista, apenas pode tramitar para a pessoa da origem da tramitação. Pelo contrário, os restantes utilizadores podem tramitar para qualquer utilizador do sistema, acrescido da possibilidade de criar cópias dos documentos, o que facilita o envio de versões diferentes para diversos utilizadores.

Dentro da caixa de tramitações existe ainda a possibilidade de visualizar as tramitações enviadas através do submenu caixa de saída que se encontra do lado direito.

8.3. Mensagens

As Mensagens estão acessíveis no menu superior direito, que permite visualizar as mensagens de outros utilizadores, e enviar novas mensagens. Os munícipes e projectistas apenas podem enviar mensagens aos gestores de procedimentos dos processos a que têm acesso, enquanto que os restantes utilizadores podem enviar mensagens a todos os utilizadores do sistema. Cada mensagem fica sempre associada a um processo.

Anexo B – Manual Técnico de Instalação

Este manual descreve a forma de instalar o sistema PLUD – Protótipo de Licenciamento Urbanístico Digital, enumera os requisitos técnicos e descreve as configurações a efectuar.

O PLUD deve ser copiado para uma pasta na qual o servidor HTTP Apache tenha acesso de leitura e escrita. Após a cópia dos ficheiros deve certificar-se que todo o software necessário está instalado correctamente.

Requisitos de software do PLUD:

- Open SSL[15];
- Apache 2[16] com os seguintes módulos carregados: mod_php[17], mod_rewrite[18], mod_ss[19] e mod_xsendfile [20];
- PHP 5[21] com os seguintes módulos carregados: MySQL[22], GD[23];
- MySql 5[24];

---

[15] http://www.openssl.org/

[16] http://httpd.apache.org/

[17] http://pt.php.net/manual/en/book.apache.php

[18] http://httpd.apache.org/docs/2.0/mod/mod_rewrite.html

[19] http://www.modssl.org/

[20] http://tn123.ath.cx/mod_xsendfile/

[21] http://www.php.net/

[22] http://pt.php.net/manual/en/book.mysql.php

[23] http://pt.php.net/gd

[24] http://www.mysql.com/

- Codeigniter 1.7[25]

A instalação destes requisitos varia de acordo com o sistema operativo utilizado. Como não é viável expor o processo de instalação de cada um nos sistemas operativos suportados, a instalação deve ser acompanhada dos manuais técnicos de cada requisito. Estes manuais podem ser obtidos através dos endereços Web disponibilizados em notas de rodapé.

Não é necessário efectuar qualquer alteração às configurações base das aplicações exceptuando o caso do Apache e o do Codeigniter, que são apresentados de seguida.

Configurações do apache:

Ficheiro httpd.conf:

```
#Activar o módulo rewrite
RewriteEngine On
RewriteOptions Inherit
DirectoryIndex index.php

//activar as funcionalidade do módulo xsend_file
XSendFile on
XSendFileAllowAbove on

#Activar SSL para comunicação segura
SSLEngine on

# caminho para o certificado no Formato PEM do Servidor
SSLCertificateFile "/etc/httpd/conf/ssl.crt/my.crt"

#Caminho para a chave privada do servidor
SSLCertificateKeyFile "/etc/httpd/conf/ssl.key/my.key"

#Configurar a verificação de certificado na autenticação do cliente
<Location /plud/profile/ccauth/>
        SSLRequireSSL
        SSLVerifyClient optional_no_ca
        SSLOptions +ExportCertData
        SSLOptions +StdEnvVars
</Location>
```

[25] http://codeigniter.com/

Configurações do Codeigniter:

Ficheiro: system/application/config/config.php

```
//Configurações gerais e de email
$config['site_name'] = 'PLUD :: Jesus Test'; //nome do site
$config['email_address'] = 'jesuscampos@maisbarcelos.pt'; //endereço de
email
$config['email_name'] = 'PLUD'; //nome para o nome
$config['smtp'] ='localhost'; //servidor smtp
$config['smtp_username'] = plud''; //credencial para o smtp
$config['smtp_password'] = 'plud'; //credencial para o smtp

//Configurações de autenticação
$config['encryption_key'] = "ga4rud7e83swEphUTrat"; // Chave para
encriptação das passwords

//Configurações de ficheiros
$config['files_path_system'] = /var/html/plud/files/system/'; //caminho para
armazenar os ficheiros do sistema
$config['files_path_users'] = /var/html/plud/files/users/'; //caminho para
armazenar os ficheiros dos repositórios dos utilizadores
$config['RootCCCA'] = /var/html/plud/root.pem'; //caminho para o certificado da
autoridade certificadora do CC
$config['CCOCSPUrl'] = 'http://ocsp.auc.cartaodecidadao.pt/publico/ocsp'; //
URL do serviço OCSP do CC
$config['opensslPath'] = "/usr/bin/openssl"; //Caminho para o openssl
```

Ficheiro: system/application/config/database.php

```
$active_group = "default";
$active_record = TRUE;

$db['default']['hostname'] = "localhost";
$db['default']['username'] = "plud";
$db['default']['password'] = "plud";
$db['default']['database'] = "plud";
$db['default']['dbdriver'] = "mysql";
$db['default']['dbprefix'] = "";
$db['default']['pconnect'] = TRUE;
$db['default']['db_debug'] = TRUE;
$db['default']['cache_on'] = FALSE;
$db['default']['cachedir'] = "";
$db['default']['char_set'] = "utf8";
$db['default']['dbcollat'] = "utf8_general_ci";
```

Configurações PLUD

Copiar o ficheiro de Termos de aceitação, a assinar pelos utilizadores, na pasta de ficheiros de sistema, sub-pasta 0, como o nome 0.pdf.

Anexo C - Descritor dos casos de uso

Nome: Inserir utilizador

Cenário Principal: É apresentado um ecrã em que o administrador ou o administrativo associa o utilizador a uma pessoa, preenche o tipo de perfil, tipo de notificação, nome de utilizador e palavra-chave e submete o pedido. O sistema testa se o nome de utilizador já existe e, caso não exista, cria o utilizador.

Cenário Alternativo 1 (Nome de utilizador existente): Idêntico ao cenário principal. Quando o nome do utilizador já existe é apresentada uma mensagem de aviso que o nome de utilizador não pode ser utilizado e o caso é reiniciado.

Nome: Validar termos de aceitação

Cenário Principal: O administrador ou administrativo submetem o pedido de validar o termo de aceitação.

Nome: Validar habilitações literárias

Cenário Principal: O administrador ou o administrativo submetem o pedido de validar a habilitação literária.

Nome: Definir responsável técnico da estrutura

Cenário Principal: Obter e verificar a estrutura orgânica. Seleccionar projectista e defini-lo como responsável técnico da estrutura.

Nome: Criar estrutura orgânica

Cenário Principal: Incluir o caso de utilização “Definir responsável técnico da estrutura”. É apresentado um ecrã em que o administrativo preenche o nome da

estrutura e, no caso de ser uma empresa, preenche o tipo de sociedade, o registo de conservatória e o capital social e submete o pedido.

Nome: Adiciona projectistas à estrutura orgânica

Cenário Principal: Obter e verificar a estrutura orgânica. Obter e verificar o projectista. Submeter o pedido.

Nome: Definir níveis de acesso aos projectistas

Cenário Principal: Obter e verificar o projectista dentro da estrutura orgânica. Definir o nível de acesso e submeter o pedido.

Nome: Carregar ficheiros

Cenário Principal: É apresentado um ecrã em que permite seleccionar ficheiros do computador local. É submetido o pedido. Os ficheiros são copiados para o repositório de ficheiros.

Nome: Submeter projecto

Cenário Principal: É apresentado um ecrã em que o munícipe preenche todos os dados do processo. O projecto é submetido. É associado um novo número de projecto e a data actual do sistema.

Nome: Registar-se no sistema

Pré-condição: Certificado digital de autenticação

Cenário Principal: É apresentado um ecrã onde devem ser preenchidos os dados do titular, o tipo de perfil, tipo de notificação, nome de utilizador e palavra-chave. O pedido é submetido.

Nome: Enviar mensagem ao coordenador autárquico

Pré-condição: Coordenador autárquico, processo

Cenário Principal: É apresentado um ecrã para preencher o assunto e o conteúdo da mensagem. A mensagem é enviada.

Nome: Gerir mensagens

Cenário Principal: É apresentado uma lista das mensagens enviadas e recebidas. É visualizada a mensagem seleccionada.

Cenário Alternativo 1 (Eliminar mensagem): Idêntico ao cenário principal. É eliminada a mensagem seleccionada.

Nome: Notifica

Pré-condição: Destinatário

Cenário Principal: Verificar o tipo de notificação do destinatário e enviar uma mensagem de notificação.

Nome: Enviar mensagem a projectista

Pré-condição: Projectista

Cenário Principal: É apresentado um ecrã para preencher o assunto e o conteúdo da mensagem. A mensagem é enviada.

Nome: Enviar mensagem a munícipe

Pré-condição: Munícipe

Cenário Principal: É apresentado um ecrã para preencher o assunto e o conteúdo da mensagem. A mensagem é enviada.

Nome: Registar contacto

Cenário Principal: É apresentado um ecrã onde é preenchido o processo, remetente, tipo de contacto, data e descrição do contacto. O registo é submetido.

Nome: Gerir disponibilidade de atendimento

Pré-condição: Coordenador autárquico

Cenário Principal: É apresentada a agenda do coordenador autárquico onde é registada uma vaga de atendimento com data, hora de início e fim.

Cenário Alternativo 1 (Modificar vaga): Idêntico ao cenário principal. É modificada a data e hora de início e fim da uma vaga de atendimento seleccionada.

Cenário Alternativo 2 (Eliminar vaga): Idêntico ao cenário principal. É eliminada uma vaga de atendimento seleccionada.

Cenário Alternativo 3 (Desmarcar atendimento): Incluir o caso de utilização "Notifica". Idêntico ao cenário Alternativo 2. O munícipe ou projectista é notificado da desmarcação do atendimento.

Nome: Marcar atendimento

Pré-condição: Coordenador autárquico

Cenário Principal: É apresentada a agenda do coordenador autárquico com as vagas disponíveis. A marcação de atendimento é registada com o assunto e número do projecto.

Nome: Registar observações de atendimento

Pré-condição: Coordenador autárquico

Cenário Principal: É apresentada a agenda do coordenador autárquico com as marcações decorridas. É preenchido com as observações da reunião.

Nome: Registar requerimento de projectista/munícipe

Cenário Principal: É apresentado um ecrã onde são preenchidos os dados do requerimento: processo e descrição do requerimento. É associado a data do sistema e um número de registo de entrada. É apresentada uma lista de ficheiros que correspondem a digitalizações de documentos de forma a seleccionar os ficheiros que correspondem ao requerimento. O requerimento é submetido. É apresentada uma mensagem de sucesso com o número do projecto.

Cenário Alternativo 1 (Documento em formato digital): Idêntico ao cenário principal. O requerimento é fornecido em formato digital com assinatura qualificada. O utilizador copia os ficheiros para a área de trabalho.

Nome: Registar parecer

Cenário Principal: É apresentado um ecrã onde são preenchidos os dados do parecer: processo, descrição e entidade externa da origem do parecer. É associado a data do sistema e um número de registo de entrada. É apresentada uma lista de ficheiros que correspondem a digitalizações de documentos de forma a seleccionar os ficheiros que correspondem ao parecer. O parecer é submetido.

Cenário Alternativo 1 (Documento em formato digital): Idêntico ao cenário principal. O parecer é fornecido em formato digital com assinatura qualificada. O utilizador copia os ficheiros para a área de trabalho.

Nome: Submeter requerimento

Pré-condição: Munícipe

Cenário Principal: É apresentado um ecrã onde é preenchida a descrição do requerimento. É associado a data do sistema. É seleccionado do repositório de ficheiros os que correspondem ao requerimento. O requerimento é submetido.

Nome: Visualizar percurso do processo

Pré-condição: Processo

Cenário Principal: É processado um documento com todas as tramitações e respectivos despachos, datas e informações dos ficheiros associados do processo.

Nome: Gerir repositório de ficheiros

Cenário Principal: É apresentada uma lista dos ficheiros carregados para o repositório do utilizador.

Cenário Alternativo 1 (Carregar novo ficheiro): Idêntico ao cenário principal. É carregado um novo ficheiro seleccionado pelo utilizador. A lista é actualizada.

Cenário Alternativo 2 (Eliminar ficheiro): Idêntico ao cenário principal. O utilizador selecciona um ou mais ficheiros do repositório. Submete pedido de eliminação de ficheiro(s). O(s) ficheiro(s) é(são) eliminado(s). A lista é actualizada.

Cenário Alternativo 3 (Visualizar ficheiro): Idêntico ao cenário principal. O utilizador selecciona um ficheiro do repositório. Submete pedido de visualizar ficheiro. O ficheiro é descarregado e aberto na origem do utilizador.

Nome: Visualizar licença/alvará

Cenário Principal: É apresentada uma lista dos processos em que o utilizador tem acesso. É seleccionado um processo. É apresentada uma lista das licenças e alvarás do processo. É seleccionado um alvará ou processo. É processada a visualização do alvará ou licença.

Nome: Registar licença/alvará

Cenário Principal: É apresentado um ecrã onde o coordenador autárquico selecciona o processo correspondente e preenche o tipo de licença/alvará, estado do

alvará/licença, número de alvará/licença, data da autorização, data de emissão, validade, custo e texto do alvará/licença. O pedido é submetido.

Nome: Informar processo

Pré-condição: Tramitação de processo

Cenário Principal: É seleccionado do repositório de ficheiros os que correspondem à informação. A informação é submetida. É registado a data da informação. É apresentada uma mensagem de sucesso.

Cenário Alternativo 1 (Informação em modo de texto): Idêntico ao cenário principal. É apresentado um ecrã onde o coordenador autárquico escreve o texto da informação.

Nome: Tramitar e despachar processo

Pré-condição: Tramitação de processo;

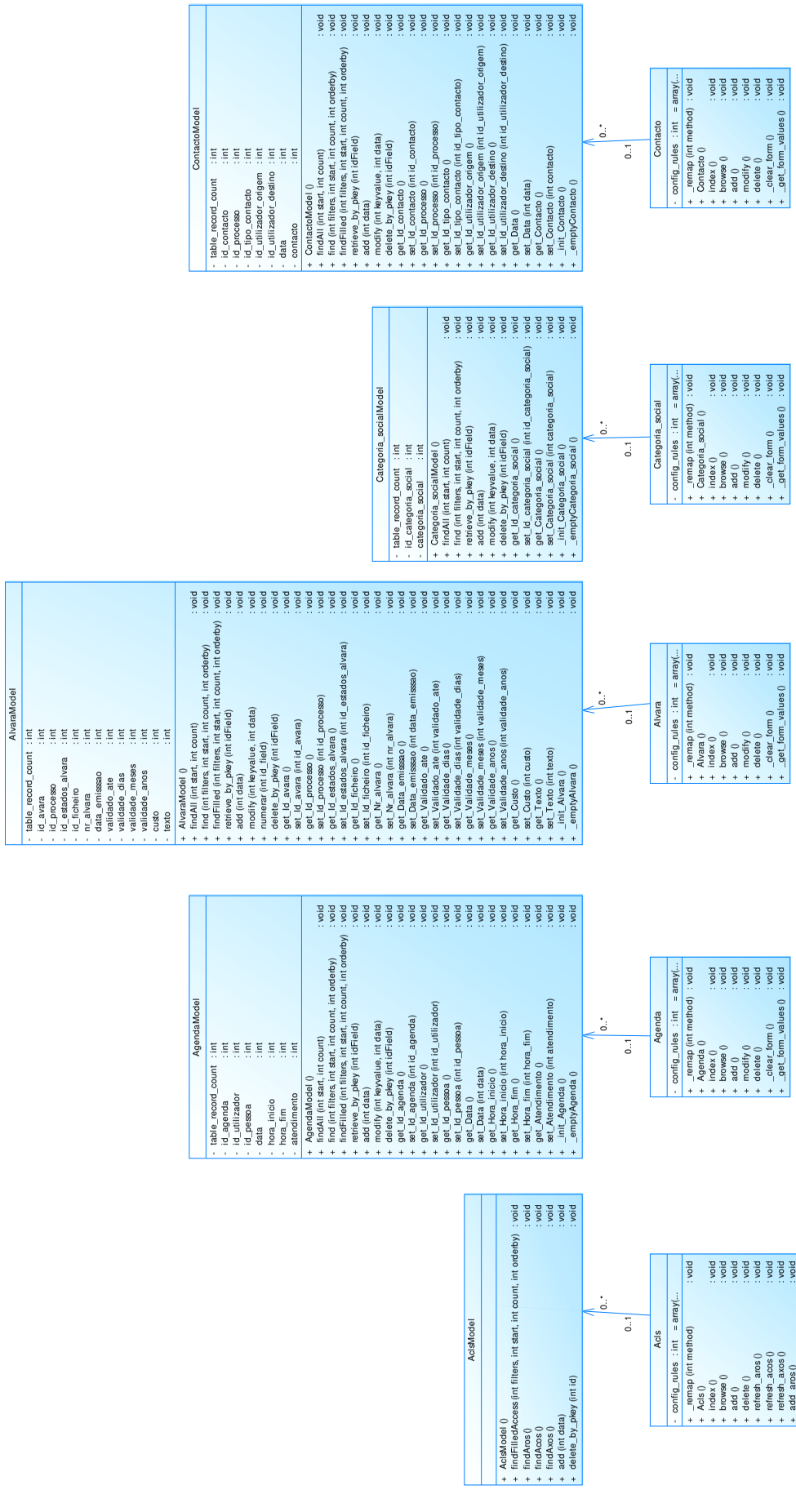
Cenário Principal: É apresentado um ecrã onde o coordenador autárquico escreve o texto do despacho e selecciona o destinatário da tramitação. A tramitação é submetida. É registada a data e hora da tramitação.

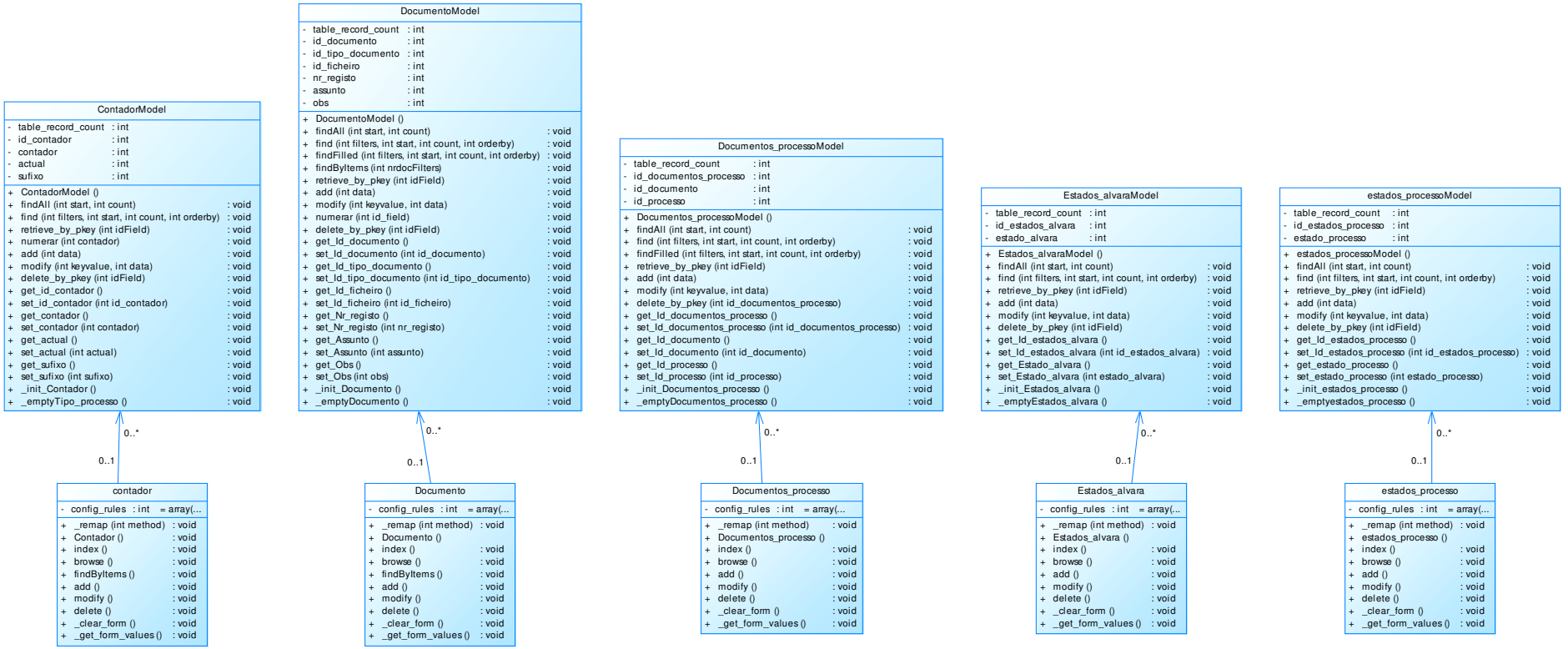
Nome: Emitir Ofícios

Cenário Principal: Incluir o caso de utilização “Notifica”. É apresentado um ecrã onde o coordenador autárquico selecciona o processo correspondente e preenche o destinatário, o assunto, o estado, a data de emissão e a data de envio. É seleccionado do repositório de ficheiros os que correspondem ao ofício. O ofício é submetido. O destinatário é notificado.

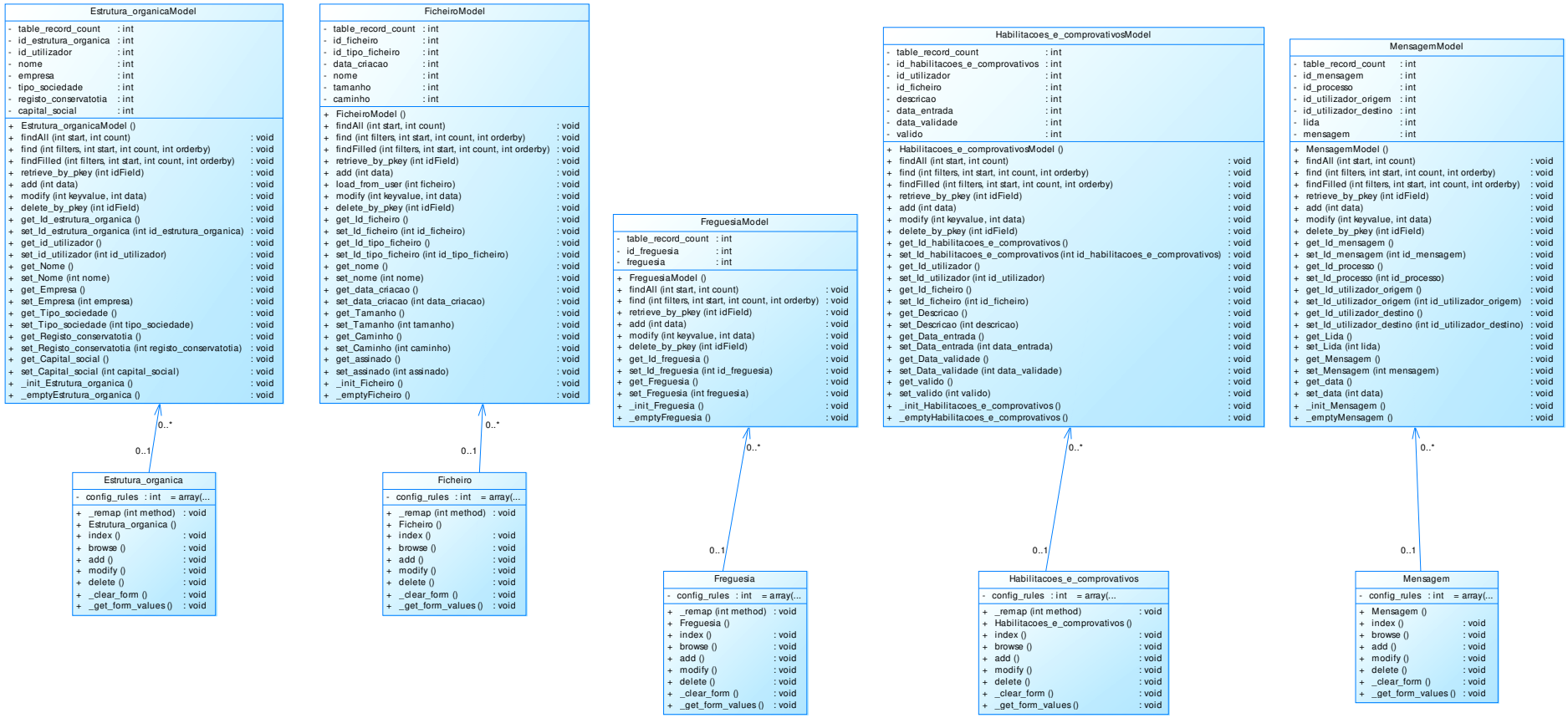
Cenário Alternativo 1 (Aguardar resposta): Idêntico ao cenário principal. É registada a data limite de resposta.

Anexo D - Diagrama de classes

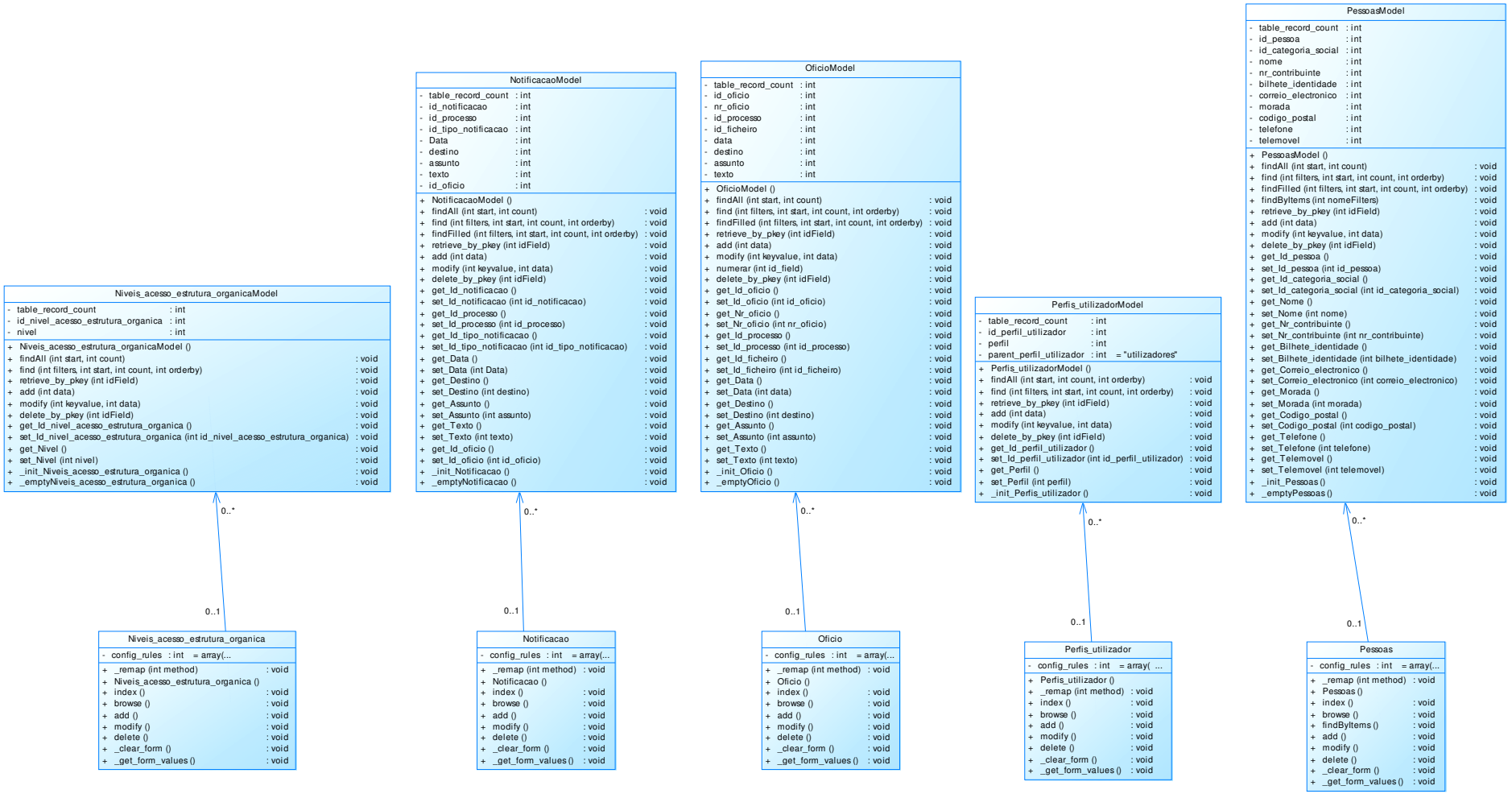



ContadorModel
- table_record_count : int
- id_contador : int
- contador : int
- actual : int
- sufixo : int
+ ContadorModel ()
+ findAll (int start, int count) : void
+ find (int filters, int start, int count, int orderby) : void
+ retrieve_by_pkey (int idField) : void
+ numerar (int contador) : void
+ add (int data) : void
+ modify (int keyvalue, int data) : void
+ delete_by_pkey (int idField) : void
+ get_id_contador () : void
+ set_id_contador (int id_contador) : void
+ get_contador () : void
+ set_contador (int contador) : void
+ get_actual () : void
+ set_actual (int actual) : void
+ get_sufixo () : void
+ set_sufixo (int sufixo) : void
+ _init_Contador () : void
+ _emptyTipo_processo () : void
0..*
0..1
contador
- config_rules : int = array(...
+ _remap (int method) : void
+ Contador () : void
+ index () : void
+ browse () : void
+ findByItems () : void
+ add () : void
+ modify () : void
+ delete () : void
+ _clear_form () : void
+ _get_form_values () : void
DocumentoModel
- table_record_count : int
- id_documento : int
- id_tipo_documento : int
- id_ficheiro : int
- nr_registo : int
- assunto : int
- obs : int
+ DocumentoModel ()
+ findAll (int start, int count) : void
+ find (int filters, int start, int count, int orderby) : void
+ findFilled (int filters, int start, int count, int orderby) : void
+ findByItems (int nrdocFilters) : void
+ retrieve_by_pkey (int idField) : void
+ add (int data) : void
+ modify (int keyvalue, int data) : void
+ numerar (int id_field) : void
+ delete_by_pkey (int idField) : void
+ get_Id_documento () : void
+ set_Id_documento (int id_documento) : void
+ get_Id_tipo_documento () : void
+ set_Id_tipo_documento (int id_tipo_documento) : void
+ get_Id_ficheiro () : void
+ set_Id_ficheiro (int id_ficheiro) : void
+ get_Nr_registo () : void
+ set_Nr_registo (int nr_registo) : void
+ get_Assunto () : void
+ set_Assunto (int assunto) : void
+ get_Obs () : void
+ set_Obs (int obs) : void
+ _init_Documento () : void
+ _emptyDocumento () : void
0..*
0..1
Documento
- config_rules : int = array(...
+ _remap (int method) : void
+ Documento ()
+ index () : void
+ browse () : void
+ findByItems () : void
+ add () : void
+ modify () : void
+ delete () : void
+ _clear_form () : void
+ _get_form_values () : void
Documentos_processoModel
- table_record_count : int
- id_documentos_processo : int
- id_documento : int
- id_processo : int
+ Documentos_processoModel ()
+ findAll (int start, int count) : void
+ find (int filters, int start, int count, int orderby) : void
+ findFilled (int filters, int start, int count, int orderby) : void
+ retrieve_by_pkey (int idField) : void
+ add (int data) : void
+ modify (int keyvalue, int data) : void
+ delete_by_pkey (int id_documentos_processo) : void
+ get_Id_documentos_processo () : void
+ set_Id_documentos_processo (int id_documentos_processo) : void
+ get_Id_documento () : void
+ set_Id_documento (int id_documento) : void
+ get_Id_processo () : void
+ set_Id_processo (int id_processo) : void
+ _init_Documentos_processo () : void
+ _emptyDocumentos_processo () : void
0..*
0..1
Documentos_processo
- config_rules : int = array(...
+ _remap (int method) : void
+ Documentos_processo ()
+ index () : void
+ browse () : void
+ add () : void
+ modify () : void
+ delete () : void
+ _clear_form () : void
+ _get_form_values () : void
Estados_alvaraModel
- table_record_count : int
- id_estados_alvara : int
- estado_alvara : int
+ Estados_alvaraModel ()
+ findAll (int start, int count) : void
+ find (int filters, int start, int count, int orderby) : void
+ retrieve_by_pkey (int idField) : void
+ add (int data) : void
+ modify (int keyvalue, int data) : void
+ delete_by_pkey (int idField) : void
+ get_Id_estados_alvara () : void
+ set_Id_estados_alvara (int id_estados_alvara) : void
+ get_Estado_alvara () : void
+ set_Estado_alvara (int estado_alvara) : void
+ _init_Estados_alvara () : void
+ _emptyEstados_alvara () : void
0..*
0..1
Estados_alvara
- config_rules : int = array(...
+ _remap (int method) : void
+ Estados_alvara ()
+ index () : void
+ browse () : void
+ add () : void
+ modify () : void
+ delete () : void
+ _clear_form () : void
+ _get_form_values () : void
estados_processoModel
- table_record_count : int
- id_estados_processo : int
- estado_processo : int
+ estados_processoModel ()
+ findAll (int start, int count) : void
+ find (int filters, int start, int count, int orderby) : void
+ retrieve_by_pkey (int idField) : void
+ add (int data) : void
+ modify (int keyvalue, int data) : void
+ delete_by_pkey (int idField) : void
+ get_Id_estados_processo () : void
+ set_Id_estados_processo (int id_estados_processo) : void
+ get_estado_processo () : void
+ set_estado_processo (int estado_processo) : void
+ _init_estados_processo () : void
+ _emptyestados_processo () : void
0..*
0..1
estados_processo
- config_rules : int = array(...
+ _remap (int method) : void
+ estados_processo ()
+ index () : void
+ browse () : void
+ add () : void
+ modify () : void
+ delete () : void
+ _clear_form () : void
+ _get_form_values () : void

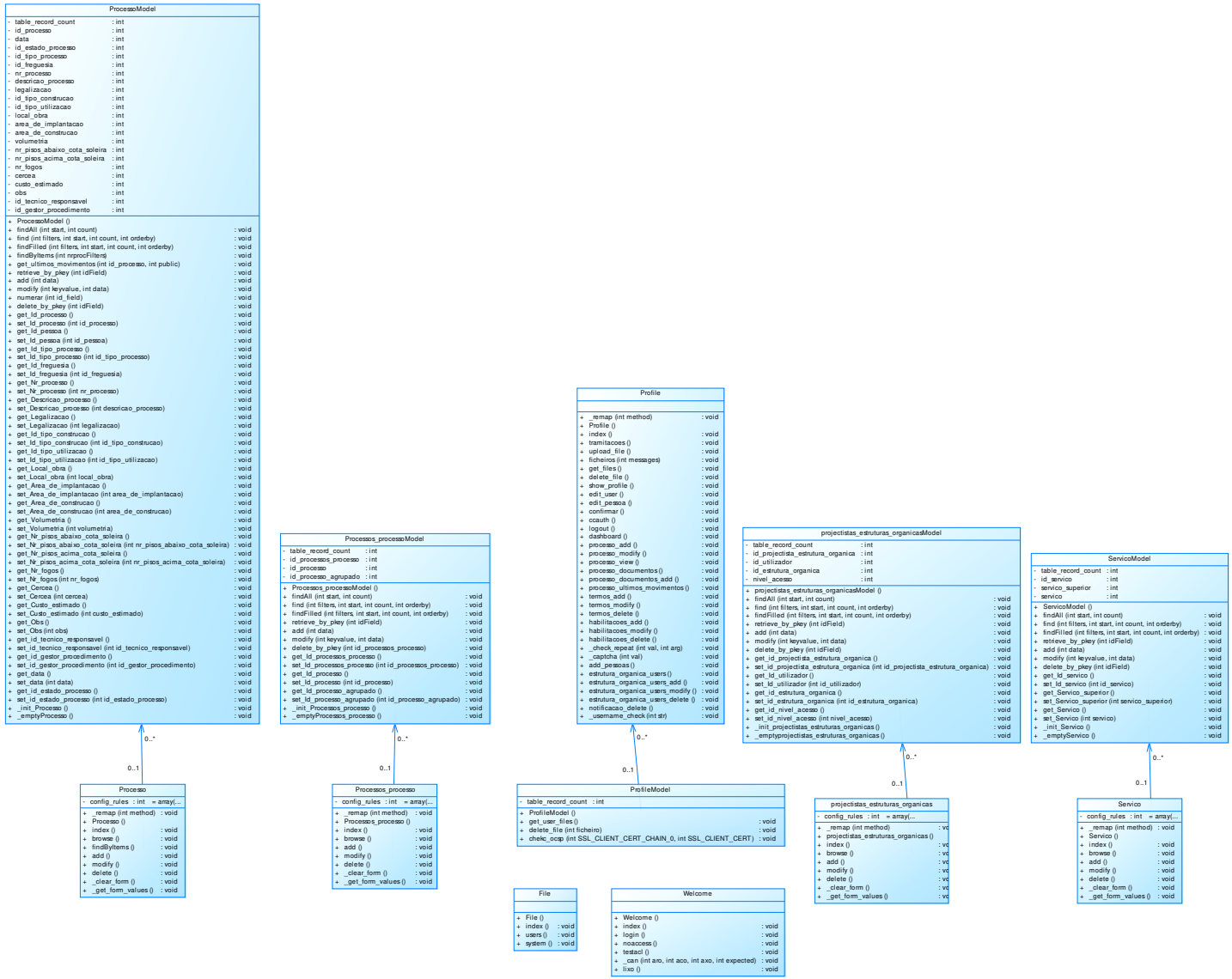
D. 2


Estrutura_organicaModel
- table_record_count : int
- id_estrutura_organica : int
- id_utilizador : int
- nome : int
- empresa : int
- tipo_sociedade : int
- registo_conservatotia : int
- capital_social : int
+ Estrutura_organicaModel ()
+ findAll (int start, int count) : void
+ find (int filters, int start, int count, int orderby) : void
+ findFilled (int filters, int start, int count, int orderby) : void
+ retrieve_by_pkey (int idField) : void
+ add (int data) : void
+ modify (int keyvalue, int data) : void
+ delete_by_pkey (int idField) : void
+ get_Id_estrutura_organica () : void
+ set_Id_estrutura_organica (int id_estrutura_organica) : void
+ get_id_utilizador () : void
+ set_id_utilizador (int id_utilizador) : void
+ get_Nome () : void
+ set_Nome (int nome) : void
+ get_Empresa () : void
+ set_Empresa (int empresa) : void
+ get_Tipo_sociedade () : void
+ set_Tipo_sociedade (int tipo_sociedade) : void
+ get_Registo_conservatotia () : void
+ set_Registo_conservatotia (int registo_conservatotia) : void
+ get_Capital_social () : void
+ set_Capital_social (int capital_social) : void
+ _init_Estrutura_organica () : void
+ _emptyEstrutura_organica () : void
0..*
0..1
Estrutura_organica
- config_rules : int = array(...
+ _remap (int method) : void
+ Estrutura_organica ()
+ index () : void
+ browse () : void
+ add () : void
+ modify () : void
+ delete () : void
+ _clear_form () : void
+ _get_form_values () : void
FicheiroModel
- table_record_count : int
- id_ficheiro : int
- id_tipo_ficheiro : int
- data_criacao : int
- nome : int
- tamanho : int
- caminho : int
+ FicheiroModel ()
+ findAll (int start, int count) : void
+ find (int filters, int start, int count, int orderby) : void
+ findFilled (int filters, int start, int count, int orderby) : void
+ retrieve_by_pkey (int idField) : void
+ add (int data) : void
+ load_from_user (int ficheiro) : void
+ modify (int keyvalue, int data) : void
+ delete_by_pkey (int idField) : void
+ get_Id_ficheiro () : void
+ set_Id_ficheiro (int id_ficheiro) : void
+ get_Id_tipo_ficheiro () : void
+ set_Id_tipo_ficheiro (int id_tipo_ficheiro) : void
+ get_nome () : void
+ set_nome (int nome) : void
+ get_data_criacao () : void
+ set_data_criacao (int data_criacao) : void
+ get_Tamanho () : void
+ set_Tamanho (int tamanho) : void
+ get_Caminho () : void
+ set_Caminho (int caminho) : void
+ get_assinado () : void
+ set_assinado (int assinado) : void
+ _init_Ficheiro () : void
+ _emptyFicheiro () : void
0..*
0..1
Ficheiro
- config_rules : int = array(...
+ _remap (int method) : void
+ Ficheiro ()
+ index () : void
+ browse () : void
+ add () : void
+ modify () : void
+ delete () : void
+ _clear_form () : void
+ _get_form_values () : void
FreguesiaModel
- table_record_count : int
- id_freguesia : int
- freguesia : int
+ FreguesiaModel ()
+ findAll (int start, int count) : void
+ find (int filters, int start, int count, int orderby) : void
+ retrieve_by_pkey (int idField) : void
+ add (int data) : void
+ modify (int keyvalue, int data) : void
+ delete_by_pkey (int idField) : void
+ get_Id_freguesia () : void
+ set_Id_freguesia (int id_freguesia) : void
+ get_Freguesia () : void
+ set_Freguesia (int freguesia) : void
+ _init_Freguesia () : void
+ _emptyFreguesia () : void
0..*
0..1
Freguesia
- config_rules : int = array(...
+ _remap (int method) : void
+ Freguesia ()
+ index () : void
+ browse () : void
+ add () : void
+ modify () : void
+ delete () : void
+ _clear_form () : void
+ _get_form_values () : void
Habilitacoes_e_comprovativosModel
- table_record_count : int
- id_habilitacoes_e_comprovativos : int
- id_utilizador : int
- id_ficheiro : int
- descricao : int
- data_entrada : int
- data_validade : int
- valido : int
+ Habilitacoes_e_comprovativosModel ()
+ findAll (int start, int count) : void
+ find (int filters, int start, int count, int orderby) : void
+ findFilled (int filters, int start, int count, int orderby) : void
+ retrieve_by_pkey (int idField) : void
+ add (int data) : void
+ modify (int keyvalue, int data) : void
+ delete_by_pkey (int idField) : void
+ get_Id_habilitacoes_e_comprovativos () : void
+ set_Id_habilitacoes_e_comprovativos (int id_habilitacoes_e_comprovativos) : void
+ get_Id_utilizador () : void
+ set_Id_utilizador (int id_utilizador) : void
+ get_Id_ficheiro () : void
+ set_Id_ficheiro (int id_ficheiro) : void
+ get_Descricao () : void
+ set_Descricao (int descricao) : void
+ get_Data_entrada () : void
+ set_Data_entrada (int data_entrada) : void
+ get_Data_validade () : void
+ set_Data_validade (int data_validade) : void
+ get_valido () : void
+ set_valido (int valido) : void
+ _init_Habilitacoes_e_comprovativos () : void
+ _emptyHabilitacoes_e_comprovativos () : void
0..*
0..1
Habilitacoes_e_comprovativos
- config_rules : int = array(...
+ _remap (int method) : void
+ Habilitacoes_e_comprovativos ()
+ index () : void
+ browse () : void
+ add () : void
+ modify () : void
+ delete () : void
+ _clear_form () : void
+ _get_form_values () : void
MensagemModel
- table_record_count : int
- id_mensagem : int
- id_processo : int
- id_utilizador_origem : int
- id_utilizador_destino : int
- lida : int
- mensagem : int
+ MensagemModel ()
+ findAll (int start, int count) : void
+ find (int filters, int start, int count, int orderby) : void
+ findFilled (int filters, int start, int count, int orderby) : void
+ retrieve_by_pkey (int idField) : void
+ add (int data) : void
+ modify (int keyvalue, int data) : void
+ delete_by_pkey (int idField) : void
+ get_Id_mensagem () : void
+ set_Id_mensagem (int id_mensagem) : void
+ get_Id_processo () : void
+ set_Id_processo (int id_processo) : void
+ get_Id_utilizador_origem () : void
+ set_Id_utilizador_origem (int id_utilizador_origem) : void
+ get_Id_utilizador_destino () : void
+ set_Id_utilizador_destino (int id_utilizador_destino) : void
+ get_Lida () : void
+ set_Lida (int lida) : void
+ get_Mensagem () : void
+ set_Mensagem (int mensagem) : void
+ get_data () : void
+ set_data (int data) : void
+ _init_Mensagem () : void
+ _emptyMensagem () : void
0..*
0..1
Mensagem
- config_rules : int = array(...
+ Mensagem ()
+ index () : void
+ browse () : void
+ add () : void
+ modify () : void
+ delete () : void
+ _clear_form () : void
+ _get_form_values () : void

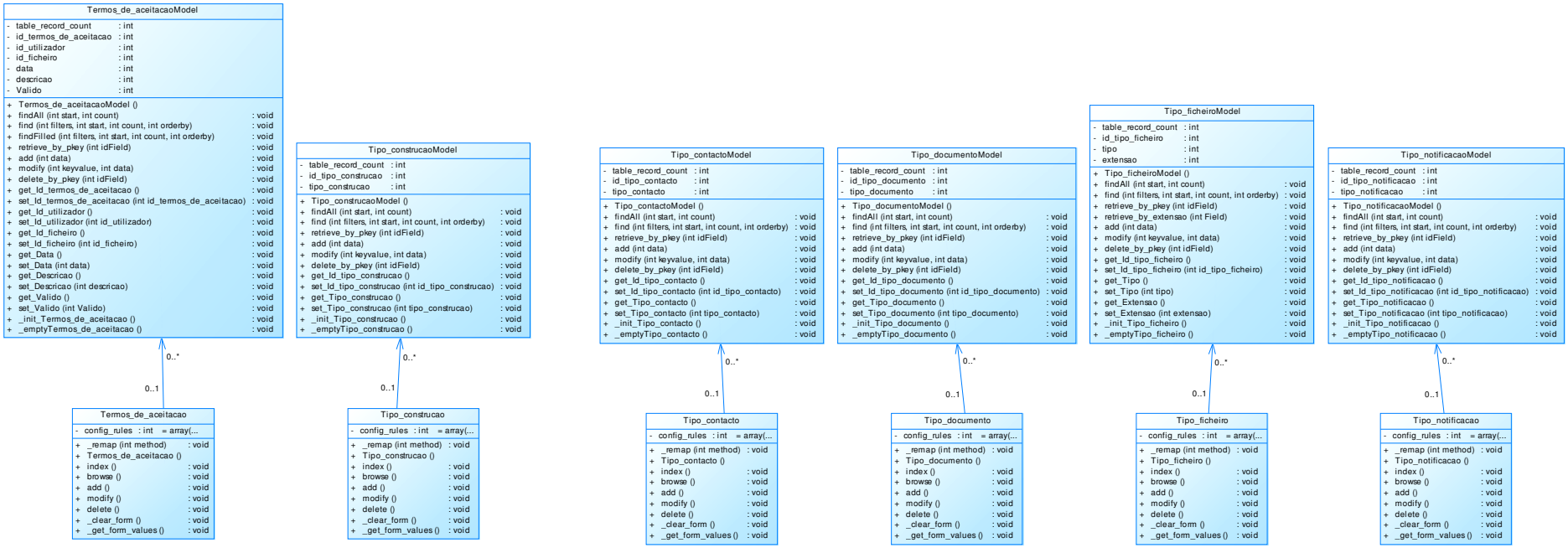
D.3


Niveis_acesso_estrutura_organicaModel
- table_record_count : int
- id_nivel_acesso_estrutura_organica : int
- nivel : int
+ Niveis_acesso_estrutura_organicaModel ()
+ findAll (int start, int count) : void
+ find (int filters, int start, int count, int orderby) : void
+ retrieve_by_pkey (int idField) : void
+ add (int data) : void
+ modify (int keyvalue, int data) : void
+ delete_by_pkey (int idField) : void
+ get_Id_nivel_acesso_estrutura_organica () : void
+ set_Id_nivel_acesso_estrutura_organica (int id_nivel_acesso_estrutura_organica) : void
+ get_Nivel () : void
+ set_Nivel (int nivel) : void
+ _init_Niveis_acesso_estrutura_organica () : void
+ _emptyNiveis_acesso_estrutura_organica () : void
0..*
0..1
Niveis_acesso_estrutura_organica
- config_rules : int = array(...
+ _remap (int method) : void
+ Niveis_acesso_estrutura_organica ()
+ index () : void
+ browse () : void
+ add () : void
+ modify () : void
+ delete () : void
+ _clear_form () : void
+ _get_form_values () : void
NotificacaoModel
- table_record_count : int
- id_notificacao : int
- id_processo : int
- id_tipo_notificacao : int
- Data : int
- destino : int
- assunto : int
- texto : int
- id_oficio : int
+ NotificacaoModel ()
+ findAll (int start, int count) : void
+ find (int filters, int start, int count, int orderby) : void
+ findFilled (int filters, int start, int count, int orderby) : void
+ retrieve_by_pkey (int idField) : void
+ add (int data) : void
+ modify (int keyvalue, int data) : void
+ delete_by_pkey (int idField) : void
+ get_Id_notificacao () : void
+ set_Id_notificacao (int id_notificacao) : void
+ get_Id_processo () : void
+ set_Id_processo (int id_processo) : void
+ get_Id_tipo_notificacao () : void
+ set_Id_tipo_notificacao (int id_tipo_notificacao) : void
+ get_Data () : void
+ set_Data (int Data) : void
+ get_Destino () : void
+ set_Destino (int destino) : void
+ get_Assunto () : void
+ set_Assunto (int assunto) : void
+ get_Texto () : void
+ set_Texto (int texto) : void
+ get_Id_oficio () : void
+ set_Id_oficio (int id_oficio) : void
+ _init_Notificacao () : void
+ _emptyNotificacao () : void
0..*
0..1
Notificacao
- config_rules : int = array(...
+ _remap (int method) : void
+ Notificacao ()
+ index () : void
+ browse () : void
+ add () : void
+ modify () : void
+ delete () : void
+ _clear_form () : void
+ _get_form_values () : void
OficioModel
- table_record_count : int
- id_oficio : int
- nr_oficio : int
- id_processo : int
- id_ficheiro : int
- data : int
- destino : int
- assunto : int
- texto : int
+ OficioModel ()
+ findAll (int start, int count) : void
+ find (int filters, int start, int count, int orderby) : void
+ findFilled (int filters, int start, int count, int orderby) : void
+ retrieve_by_pkey (int idField) : void
+ add (int data) : void
+ modify (int keyvalue, int data) : void
+ numerar (int id_field) : void
+ delete_by_pkey (int idField) : void
+ get_Id_oficio () : void
+ set_Id_oficio (int id_oficio) : void
+ get_Nr_oficio () : void
+ set_Nr_oficio (int nr_oficio) : void
+ get_Id_processo () : void
+ set_Id_processo (int id_processo) : void
+ get_Id_ficheiro () : void
+ set_Id_ficheiro (int id_ficheiro) : void
+ get_Data () : void
+ set_Data (int data) : void
+ get_Destino () : void
+ set_Destino (int destino) : void
+ get_Assunto () : void
+ set_Assunto (int assunto) : void
+ get_Texto () : void
+ set_Texto (int texto) : void
+ _init_Oficio () : void
+ _emptyOficio () : void
0..*
0..1
Oficio
- config_rules : int = array(...
+ _remap (int method) : void
+ Oficio ()
+ index () : void
+ browse () : void
+ add () : void
+ modify () : void
+ delete () : void
+ _clear_form () : void
+ _get_form_values () : void
Perfis_utilizadorModel
- table_record_count : int
- id_perfil_utilizador : int
- perfil : int
- parent_perfil_utilizador : int = "utilizadores"
+ Perfis_utilizadorModel ()
+ findAll (int start, int count, int orderby) : void
+ find (int filters, int start, int count, int orderby) : void
+ retrieve_by_pkey (int idField) : void
+ add (int data) : void
+ modify (int keyvalue, int data) : void
+ delete_by_pkey (int idField) : void
+ get_Id_perfil_utilizador () : void
+ set_Id_perfil_utilizador (int id_perfil_utilizador) : void
+ get_Perfil () : void
+ set_Perfil (int perfil) : void
+ _init_Perfis_utilizador () : void
0..*
0..1
Perfis_utilizador
- config_rules : int = array(...
+ Perfis_utilizador ()
+ _remap (int method) : void
+ index () : void
+ browse () : void
+ add () : void
+ modify () : void
+ delete () : void
+ _clear_form () : void
+ _get_form_values () : void
PessoasModel
- table_record_count : int
- id_pessoa : int
- id_categoria_social : int
- nome : int
- nr_contribuinte : int
- bilhete_identidade : int
- correio_electronico : int
- morada : int
- codigo_postal : int
- telefone : int
- telemovel : int
+ PessoasModel ()
+ findAll (int start, int count) : void
+ find (int filters, int start, int count, int orderby) : void
+ findFilled (int filters, int start, int count, int orderby) : void
+ findByItems (int nomeFilters) : void
+ retrieve_by_pkey (int idField) : void
+ add (int data) : void
+ modify (int keyvalue, int data) : void
+ delete_by_pkey (int idField) : void
+ get_Id_pessoa () : void
+ set_Id_pessoa (int id_pessoa) : void
+ get_Id_categoria_social () : void
+ set_Id_categoria_social (int id_categoria_social) : void
+ get_Nome () : void
+ set_Nome (int nome) : void
+ get_Nr_contribuinte () : void
+ set_Nr_contribuinte (int nr_contribuinte) : void
+ get_Bilhete_identidade () : void
+ set_Bilhete_identidade (int bilhete_identidade) : void
+ get_Correio_electronico () : void
+ set_Correio_electronico (int correio_electronico) : void
+ get_Morada () : void
+ set_Morada (int morada) : void
+ get_Codigo_postal () : void
+ set_Codigo_postal (int codigo_postal) : void
+ get_Telefone () : void
+ set_Telefone (int telefone) : void
+ get_Telemovel () : void
+ set_Telemovel (int telemovel) : void
+ _init_Pessoas () : void
+ _emptyPessoas () : void
0..*
0..1
Pessoas
- config_rules : int = array(...
+ _remap (int method) : void
+ Pessoas ()
+ index () : void
+ browse () : void
+ findByItems () : void
+ add () : void
+ modify () : void
+ delete () : void
+ _clear_form () : void
+ _get_form_values () : void

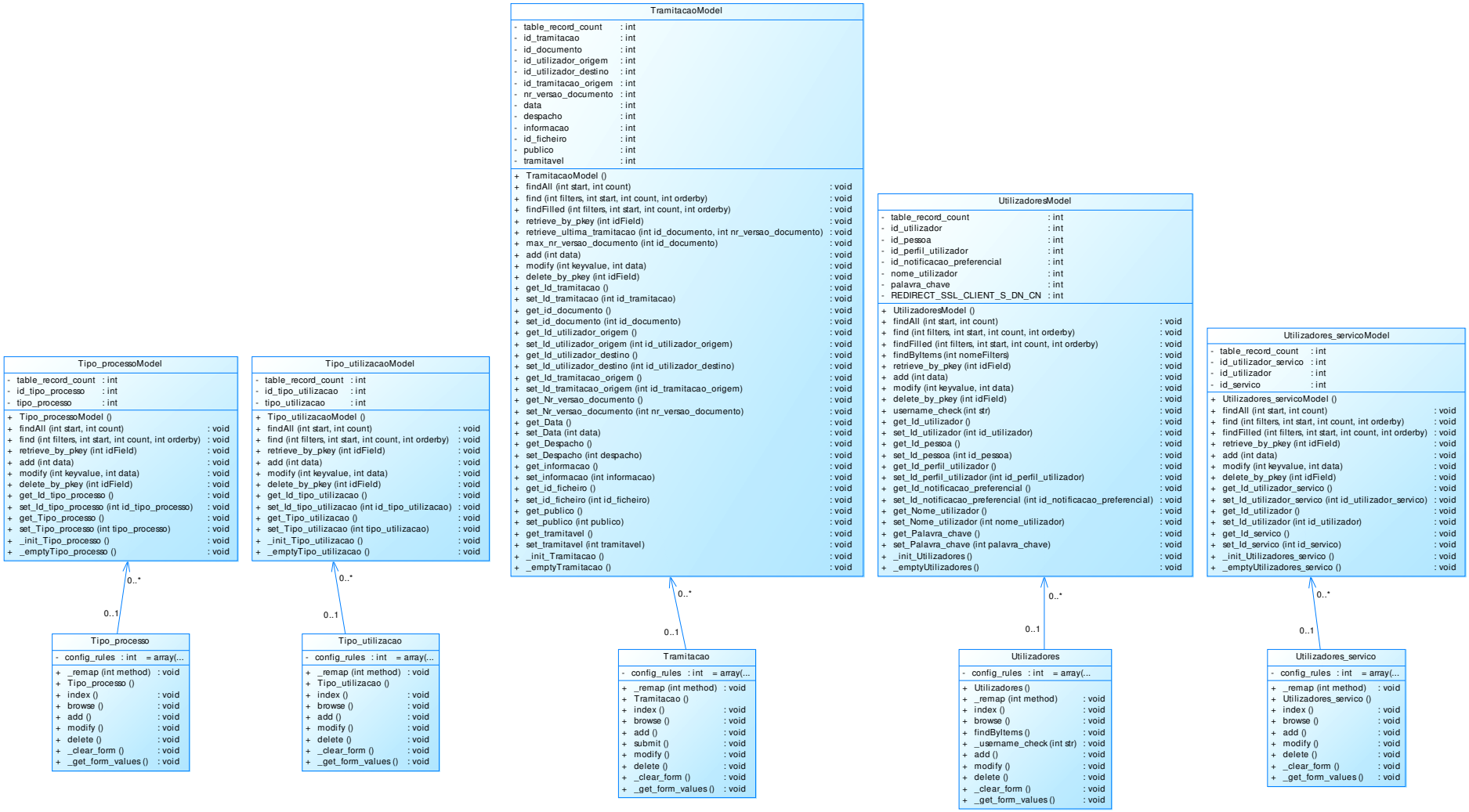
D. 4


ProcessoModel
- table_record_count : int
- id_processo : int
- data : int
- id_estado_processo : int
- id_tipo_processo : int
- id_freguesia : int
- nr_processo : int
- descricao_processo : int
- legalizacao : int
- id_tipo_construcao : int
- id_tipo_utilizacao : int
- local_obra : int
- area_de_implantacao : int
- area_de_construcao : int
- volumetria : int
- nr_pisos_abaixo_cota_soleira : int
- nr_pisos_acima_cota_soleira : int
- nr_fogos : int
- cercea : int
- custo_estimado : int
- obs : int
- id_tecnico_responsavel : int
- id_gestor_procedimento : int
+ ProcessoModel ()
+ findAll (int start, int count) : void
+ find (int filters, int start, int count, int orderby) : void
+ findFilled (int filters, int start, int count, int orderby) : void
+ findByItems (int nrprocFilters) : void
+ get_ultimos_movimentos (int id_processo, int public) : void
+ retrieve_by_pkey (int idField) : void
+ add (int data) : void
+ modify (int keyvalue, int data) : void
+ numerar (int id_field) : void
+ delete_by_pkey (int idField) : void
+ get_Id_processo () : void
+ set_Id_processo (int id_processo) : void
+ get_Id_pessoa () : void
+ set_Id_pessoa (int id_pessoa) : void
+ get_Id_tipo_processo () : void
+ set_Id_tipo_processo (int id_tipo_processo) : void
+ get_Id_freguesia () : void
+ set_Id_freguesia (int id_freguesia) : void
+ get_Nr_processo () : void
+ set_Nr_processo (int nr_processo) : void
+ get_Descricao_processo () : void
+ set_Descricao_processo (int descricao_processo) : void
+ get_Legalizacao () : void
+ set_Legalizacao (int legalizacao) : void
+ get_Id_tipo_construcao () : void
+ set_Id_tipo_construcao (int id_tipo_construcao) : void
+ get_Id_tipo_utilizacao () : void
+ set_Id_tipo_utilizacao (int id_tipo_utilizacao) : void
+ get_Local_obra () : void
+ set_Local_obra (int local_obra) : void
+ get_Area_de_implantacao () : void
+ set_Area_de_implantacao (int area_de_implantacao) : void
+ get_Area_de_construcao () : void
+ set_Area_de_construcao (int area_de_construcao) : void
+ get_Volumetria () : void
+ set_Volumetria (int volumetria) : void
+ get_Nr_pisos_abaixo_cota_soleira () : void
+ set_Nr_pisos_abaixo_cota_soleira (int nr_pisos_abaixo_cota_soleira) : void
+ get_Nr_pisos_acima_cota_soleira () : void
+ set_Nr_pisos_acima_cota_soleira (int nr_pisos_acima_cota_soleira) : void
+ get_Nr_fogos () : void
+ set_Nr_fogos (int nr_fogos) : void
+ get_Cercea () : void
+ set_Cercea (int cercea) : void
+ get_Custo_estimado () : void
+ set_Custo_estimado (int custo_estimado) : void
+ get_Obs () : void
+ set_Obs (int obs) : void
+ get_id_tecnico_responsavel () : void
+ set_id_tecnico_responsavel (int id_tecnico_responsavel) : void
+ get_id_gestor_procedimento () : void
+ set_id_gestor_procedimento (int id_gestor_procedimento) : void
+ get_data () : void
+ set_data (int data) : void
+ get_id_estado_processo () : void
+ set_id_estado_processo (int id_estado_processo) : void
+ _init_Processo () : void
+ _emptyProcesso () : void
0..*
0..1
Processo
- config_rules : int = array(...
+ _remap (int method) : void
+ Processo ()
+ index () : void
+ browse () : void
+ findByItems () : void
+ add () : void
+ modify () : void
+ delete () : void
+ _clear_form () : void
+ _get_form_values () : void
Processos_processoModel
- table_record_count : int
- id_processos_processo : int
- id_processo : int
- id_processo_agrupado : int
+ Processos_processoModel ()
+ findAll (int start, int count) : void
+ find (int filters, int start, int count, int orderby) : void
+ findFilled (int filters, int start, int count, int orderby) : void
+ retrieve_by_pkey (int idField) : void
+ add (int data) : void
+ modify (int keyvalue, int data) : void
+ delete_by_pkey (int id_processos_processo) : void
+ get_Id_processos_processo () : void
+ set_Id_processos_processo (int id_processos_processo) : void
+ get_Id_processo () : void
+ set_Id_processo (int id_processo) : void
+ get_Id_processo_agrupado () : void
+ set_Id_processo_agrupado (int id_processo_agrupado) : void
+ _init_Processos_processo () : void
+ _emptyProcessos_processo () : void
0..*
0..1
Processos_processo
- config_rules : int = array(...
+ _remap (int method) : void
+ Processos_processo ()
+ index () : void
+ browse () : void
+ add () : void
+ modify () : void
+ delete () : void
+ _clear_form () : void
+ _get_form_values () : void
Profile
+ _remap (int method) : void
+ Profile ()
+ index () : void
+ tramitacoes () : void
+ upload_file () : void
+ ficheiros (int messages) : void
+ get_files () : void
+ delete_file () : void
+ show_profile () : void
+ edit_user () : void
+ edit_pessoa () : void
+ confirmar () : void
+ ccauth () : void
+ logout () : void
+ dashboard () : void
+ processo_add () : void
+ processo_modify () : void
+ processo_view () : void
+ processo_documentos () : void
+ processo_documentos_add () : void
+ processo_ultimos_movimentos () : void
+ termos_add () : void
+ termos_modify () : void
+ termos_delete () : void
+ habilitacoes_add () : void
+ habilitacoes_modify () : void
+ habilitacoes_delete () : void
+ _check_repeat (int val, int arg) : void
+ _captcha (int val) : void
+ add_pessoas () : void
+ estrutura_organica_users () : void
+ estrutura_organica_users_add () : void
+ estrutura_organica_users_modify () : void
+ estrutura_organica_users_delete () : void
+ notificacao_delete () : void
+ _username_check (int str) : void
0..*
0..1
ProfileModel
- table_record_count : int
+ ProfileModel ()
+ get_user_files ()
+ delete_file (int ficheiro)
+ chekc_ocsp (int SSL_CLIENT_CERT_CHAIN_0, int SSL_CLIENT_CERT) : void
File
+ File ()
+ index () : void
+ users () : void
+ system () : void
Welcome
+ Welcome ()
+ index ()
+ login () : void
+ noaccess () : void
+ testacl () : void
+ _can (int aro, int aco, int axo, int expected) : void
+ lixo () : void
projectistas_estruturas_organicasModel
- table_record_count : int
- id_projectista_estrutura_organica : int
- id_utilizador : int
- id_estrutura_organica : int
- nivel_acesso : int
+ projectistas_estruturas_organicasModel ()
+ findAll (int start, int count) : void
+ find (int filters, int start, int count, int orderby) : void
+ findFilled (int filters, int start, int count, int orderby) : void
+ retrieve_by_pkey (int idField) : void
+ add (int data) : void
+ modify (int keyvalue, int data) : void
+ delete_by_pkey (int idField) : void
+ get_id_projectista_estrutura_organica () : void
+ set_id_projectista_estrutura_organica (int id_projectista_estrutura_organica) : void
+ get_Id_utilizador () : void
+ set_Id_utilizador (int id_utilizador) : void
+ get_id_estrutura_organica () : void
+ set_id_estrutura_organica (int id_estrutura_organica) : void
+ get_id_nivel_acesso () : void
+ set_id_nivel_acesso (int nivel_acesso) : void
+ _init_projectistas_estruturas_organicas () : void
+ _emptyprojectistas_estruturas_organicas () : void
0..*
0..1
projectistas_estruturas_organicas
- config_rules : int = array(...
+ _remap (int method) : vo
+ projectistas_estruturas_organicas ()
+ index () : vo
+ browse () : vo
+ add () : vo
+ modify () : vo
+ delete () : vo
+ _clear_form () : vo
+ _get_form_values () : vo
ServicoModel
- table_record_count : int
- id_servico : int
- servico_superior : int
- servico : int
+ ServicoModel ()
+ findAll (int start, int count) : void
+ find (int filters, int start, int count, int orderby) : void
+ findFilled (int filters, int start, int count, int orderby) : void
+ retrieve_by_pkey (int idField) : void
+ add (int data) : void
+ modify (int keyvalue, int data) : void
+ delete_by_pkey (int idField) : void
+ get_Id_servico () : void
+ set_Id_servico (int id_servico) : void
+ get_Servico_superior () : void
+ set_Servico_superior (int servico_superior) : void
+ get_Servico () : void
+ set_Servico (int servico) : void
+ _init_Servico () : void
+ _emptyServico () : void
0..*
0..1
Servico
- config_rules : int = array(...
+ _remap (int method) : void
+ Servico ()
+ index () : void
+ browse () : void
+ add () : void
+ modify () : void
+ delete () : void
+ _clear_form () : void
+ _get_form_values () : void

D.5


Termos_de_aceitacaoModel
- table_record_count : int
- id_termos_de_aceitacao : int
- id_utilizador : int
- id_ficheiro : int
- data : int
- descricao : int
- Valido : int
+ Termos_de_aceitacaoModel ()
+ findAll (int start, int count) : void
+ find (int filters, int start, int count, int orderby) : void
+ findFilled (int filters, int start, int count, int orderby) : void
+ retrieve_by_pkey (int idField) : void
+ add (int data) : void
+ modify (int keyvalue, int data) : void
+ delete_by_pkey (int idField) : void
+ get_Id_termos_de_aceitacao () : void
+ set_Id_termos_de_aceitacao (int id_termos_de_aceitacao) : void
+ get_Id_utilizador () : void
+ set_Id_utilizador (int id_utilizador) : void
+ get_Id_ficheiro () : void
+ set_Id_ficheiro (int id_ficheiro) : void
+ get_Data () : void
+ set_Data (int data) : void
+ get_Descricao () : void
+ set_Descricao (int descricao) : void
+ get_Valido () : void
+ set_Valido (int Valido) : void
+ _init_Termos_de_aceitacao () : void
+ _emptyTermos_de_aceitacao () : void
0..*
0..1
Termos_de_aceitacao
- config_rules : int = array(...
+ _remap (int method) : void
+ Termos_de_aceitacao ()
+ index () : void
+ browse () : void
+ add () : void
+ modify () : void
+ delete () : void
+ _clear_form () : void
+ _get_form_values () : void
Tipo_construcaoModel
- table_record_count : int
- id_tipo_construcao : int
- tipo_construcao : int
+ Tipo_construcaoModel ()
+ findAll (int start, int count) : void
+ find (int filters, int start, int count, int orderby) : void
+ retrieve_by_pkey (int idField) : void
+ add (int data) : void
+ modify (int keyvalue, int data) : void
+ delete_by_pkey (int idField) : void
+ get_Id_tipo_construcao () : void
+ set_Id_tipo_construcao (int id_tipo_construcao) : void
+ get_Tipo_construcao () : void
+ set_Tipo_construcao (int tipo_construcao) : void
+ _init_Tipo_construcao () : void
+ _emptyTipo_construcao () : void
0..*
0..1
Tipo_construcao
- config_rules : int = array(...
+ _remap (int method) : void
+ Tipo_construcao ()
+ index () : void
+ browse () : void
+ add () : void
+ modify () : void
+ delete () : void
+ _clear_form () : void
+ _get_form_values () : void
Tipo_contactoModel
- table_record_count : int
- id_tipo_contacto : int
- tipo_contacto : int
+ Tipo_contactoModel ()
+ findAll (int start, int count) : void
+ find (int filters, int start, int count, int orderby) : void
+ retrieve_by_pkey (int idField) : void
+ add (int data) : void
+ modify (int keyvalue, int data) : void
+ delete_by_pkey (int idField) : void
+ get_Id_tipo_contacto () : void
+ set_Id_tipo_contacto (int id_tipo_contacto) : void
+ get_Tipo_contacto () : void
+ set_Tipo_contacto (int tipo_contacto) : void
+ _init_Tipo_contacto () : void
+ _emptyTipo_contacto () : void
0..*
0..1
Tipo_contacto
- config_rules : int = array(...
+ _remap (int method) : void
+ Tipo_contacto ()
+ index () : void
+ browse () : void
+ add () : void
+ modify () : void
+ delete () : void
+ _clear_form () : void
+ _get_form_values () : void
Tipo_documentoModel
- table_record_count : int
- id_tipo_documento : int
- tipo_documento : int
+ Tipo_documentoModel ()
+ findAll (int start, int count) : void
+ find (int filters, int start, int count, int orderby) : void
+ retrieve_by_pkey (int idField) : void
+ add (int data) : void
+ modify (int keyvalue, int data) : void
+ delete_by_pkey (int idField) : void
+ get_Id_tipo_documento () : void
+ set_Id_tipo_documento (int id_tipo_documento) : void
+ get_Tipo_documento () : void
+ set_Tipo_documento (int tipo_documento) : void
+ _init_Tipo_documento () : void
+ _emptyTipo_documento () : void
0..*
0..1
Tipo_documento
- config_rules : int = array(...
+ _remap (int method) : void
+ Tipo_documento ()
+ index () : void
+ browse () : void
+ add () : void
+ modify () : void
+ delete () : void
+ _clear_form () : void
+ _get_form_values () : void
Tipo_ficheiroModel
- table_record_count : int
- id_tipo_ficheiro : int
- tipo : int
- extensao : int
+ Tipo_ficheiroModel ()
+ findAll (int start, int count) : void
+ find (int filters, int start, int count, int orderby) : void
+ retrieve_by_pkey (int idField) : void
+ retrieve_by_extensao (int Field) : void
+ add (int data) : void
+ modify (int keyvalue, int data) : void
+ delete_by_pkey (int idField) : void
+ get_Id_tipo_ficheiro () : void
+ set_Id_tipo_ficheiro (int id_tipo_ficheiro) : void
+ get_Tipo () : void
+ set_Tipo (int tipo) : void
+ get_Extensao () : void
+ set_Extensao (int extensao) : void
+ _init_Tipo_ficheiro () : void
+ _emptyTipo_ficheiro () : void
0..*
0..1
Tipo_ficheiro
- config_rules : int = array(...
+ _remap (int method) : void
+ Tipo_ficheiro ()
+ index () : void
+ browse () : void
+ add () : void
+ modify () : void
+ delete () : void
+ _clear_form () : void
+ _get_form_values () : void
Tipo_notificacaoModel
- table_record_count : int
- id_tipo_notificacao : int
- tipo_notificacao : int
+ Tipo_notificacaoModel ()
+ findAll (int start, int count) : void
+ find (int filters, int start, int count, int orderby) : void
+ retrieve_by_pkey (int idField) : void
+ add (int data) : void
+ modify (int keyvalue, int data) : void
+ delete_by_pkey (int idField) : void
+ get_Id_tipo_notificacao () : void
+ set_Id_tipo_notificacao (int id_tipo_notificacao) : void
+ get_Tipo_notificacao () : void
+ set_Tipo_notificacao (int tipo_notificacao) : void
+ _init_Tipo_notificacao () : void
+ _emptyTipo_notificacao () : void
0..*
0..1
Tipo_notificacao
- config_rules : int = array(...
+ _remap (int method) : void
+ Tipo_notificacao ()
+ index () : void
+ browse () : void
+ add () : void
+ modify () : void
+ delete () : void
+ _clear_form () : void
+ _get_form_values () : void

D.6


TramitacaoModel
- table_record_count : int
- id_tramitacao : int
- id_documento : int
- id_utilizador_origem : int
- id_utilizador_destino : int
- id_tramitacao_origem : int
- nr_versao_documento : int
- data : int
- despacho : int
- informacao : int
- id_ficheiro : int
- publico : int
- tramitavel : int
+ TramitacaoModel ()
+ findAll (int start, int count) : void
+ find (int filters, int start, int count, int orderby) : void
+ findFilled (int filters, int start, int count, int orderby) : void
+ retrieve_by_pkey (int idField) : void
+ retrieve_ultima_tramitacao (int id_documento, int nr_versao_documento) : void
+ max_nr_versao_documento (int id_documento) : void
+ add (int data) : void
+ modify (int keyvalue, int data) : void
+ delete_by_pkey (int idField) : void
+ get_Id_tramitacao () : void
+ set_Id_tramitacao (int id_tramitacao) : void
+ get_id_documento () : void
+ set_id_documento (int id_documento) : void
+ get_Id_utilizador_origem () : void
+ set_Id_utilizador_origem (int id_utilizador_origem) : void
+ get_Id_utilizador_destino () : void
+ set_Id_utilizador_destino (int id_utilizador_destino) : void
+ get_Id_tramitacao_origem () : void
+ set_Id_tramitacao_origem (int id_tramitacao_origem) : void
+ get_Nr_versao_documento () : void
+ set_Nr_versao_documento (int nr_versao_documento) : void
+ get_Data () : void
+ set_Data (int data) : void
+ get_Despacho () : void
+ set_Despacho (int despacho) : void
+ get_informacao () : void
+ set_informacao (int informacao) : void
+ get_id_ficheiro () : void
+ set_id_ficheiro (int id_ficheiro) : void
+ get_publico () : void
+ set_publico (int publico) : void
+ get_tramitavel () : void
+ set_tramitavel (int tramitavel) : void
+ _init_Tramitacao () : void
+ _emptyTramitacao () : void
UtilizadoresModel
- table_record_count : int
- id_utilizador : int
- id_pessoa : int
- id_perfil_utilizador : int
- id_notificacao_preferencial : int
- nome_utilizador : int
- palavra_chave : int
- REDIRECT_SSL_CLIENT_S_DN_CN : int
+ UtilizadoresModel ()
+ findAll (int start, int count) : void
+ find (int filters, int start, int count, int orderby) : void
+ findFilled (int filters, int start, int count, int orderby) : void
+ findByItems (int nomeFilters) : void
+ retrieve_by_pkey (int idField) : void
+ add (int data) : void
+ modify (int keyvalue, int data) : void
+ delete_by_pkey (int idField) : void
+ username_check (int str) : void
+ get_Id_utilizador () : void
+ set_Id_utilizador (int id_utilizador) : void
+ get_Id_pessoa () : void
+ set_Id_pessoa (int id_pessoa) : void
+ get_Id_perfil_utilizador () : void
+ set_Id_perfil_utilizador (int id_perfil_utilizador) : void
+ get_Id_notificacao_preferencial () : void
+ set_Id_notificacao_preferencial (int id_notificacao_preferencial) : void
+ get_Nome_utilizador () : void
+ set_Nome_utilizador (int nome_utilizador) : void
+ get_Palavra_chave () : void
+ set_Palavra_chave (int palavra_chave) : void
+ _init_Utilizadores () : void
+ _emptyUtilizadores () : void
Utilizadores_servicoModel
- table_record_count : int
- id_utilizador_servico : int
- id_utilizador : int
- id_servico : int
+ Utilizadores_servicoModel ()
+ findAll (int start, int count) : void
+ find (int filters, int start, int count, int orderby) : void
+ findFilled (int filters, int start, int count, int orderby) : void
+ retrieve_by_pkey (int idField) : void
+ add (int data) : void
+ modify (int keyvalue, int data) : void
+ delete_by_pkey (int idField) : void
+ get_Id_utilizador_servico () : void
+ set_Id_utilizador_servico (int id_utilizador_servico) : void
+ get_Id_utilizador () : void
+ set_Id_utilizador (int id_utilizador) : void
+ get_Id_servico () : void
+ set_Id_servico (int id_servico) : void
+ _init_Utilizadores_servico () : void
+ _emptyUtilizadores_servico () : void
Tipo_processoModel
- table_record_count : int
- id_tipo_processo : int
- tipo_processo : int
+ Tipo_processoModel ()
+ findAll (int start, int count) : void
+ find (int filters, int start, int count, int orderby) : void
+ retrieve_by_pkey (int idField) : void
+ add (int data) : void
+ modify (int keyvalue, int data) : void
+ delete_by_pkey (int idField) : void
+ get_Id_tipo_processo () : void
+ set_Id_tipo_processo (int id_tipo_processo) : void
+ get_Tipo_processo () : void
+ set_Tipo_processo (int tipo_processo) : void
+ _init_Tipo_processo () : void
+ _emptyTipo_processo () : void
Tipo_utilizacaoModel
- table_record_count : int
- id_tipo_utilizacao : int
- tipo_utilizacao : int
+ Tipo_utilizacaoModel ()
+ findAll (int start, int count) : void
+ find (int filters, int start, int count, int orderby) : void
+ retrieve_by_pkey (int idField) : void
+ add (int data) : void
+ modify (int keyvalue, int data) : void
+ delete_by_pkey (int idField) : void
+ get_Id_tipo_utilizacao () : void
+ set_Id_tipo_utilizacao (int id_tipo_utilizacao) : void
+ get_Tipo_utilizacao () : void
+ set_Tipo_utilizacao (int tipo_utilizacao) : void
+ _init_Tipo_utilizacao () : void
+ _emptyTipo_utilizacao () : void
Tramitacao
- config_rules : int = array(...
+ _remap (int method) : void
+ Tramitacao ()
+ index () : void
+ browse () : void
+ add () : void
+ submit () : void
+ modify () : void
+ delete () : void
+ _clear_form () : void
+ _get_form_values () : void
Utilizadores
- config_rules : int = array(...
+ Utilizadores ()
+ _remap (int method) : void
+ index () : void
+ browse () : void
+ findByItems () : void
+ _username_check (int str) : void
+ add () : void
+ modify () : void
+ delete () : void
+ _clear_form () : void
+ _get_form_values () : void
Utilizadores_servico
- config_rules : int = array(...
+ _remap (int method) : void
+ Utilizadores_servico ()
+ index () : void
+ browse () : void
+ add () : void
+ modify () : void
+ delete () : void
+ _clear_form () : void
+ _get_form_values () : void
Tipo_processo
- config_rules : int = array(...
+ _remap (int method) : void
+ Tipo_processo ()
+ index () : void
+ browse () : void
+ add () : void
+ modify () : void
+ delete () : void
+ _clear_form () : void
+ _get_form_values () : void
Tipo_utilizacao
- config_rules : int = array(...
+ _remap (int method) : void
+ Tipo_utilizacao ()
+ index () : void
+ browse () : void
+ add () : void
+ modify () : void
+ delete () : void
+ _clear_form () : void
+ _get_form_values () : void
0..*
0..1
0..*
0..1
0..*
0..1
0..*
0..1
0..*
0..1

D.7

Anexo E – Listagem de relações entre classes

Classes	Tipo de classe a	Relação
controllers/acls.php	helper	khacl_helper
controllers/acls.php	helper	url
controllers/acls.php	library	form_validation
controllers/acls.php	library	khacl
controllers/acls.php	library	layout
controllers/acls.php	library	pagination
controllers/acls.php	model	AclsModel
controllers/acls.php	model	utilizadoresmodel
controllers/agenda.php	helper	khacl_helper
controllers/agenda.php	helper	url
controllers/agenda.php	library	form_validation
controllers/agenda.php	library	layout
controllers/agenda.php	library	pagination
controllers/agenda.php	model	agendamodel
controllers/agenda.php	model	pessoasmodel
controllers/agenda.php	model	UtilizadoresModel
controllers/agenda.php	plugin	js_calendar
controllers/alvara.php	helper	khacl_helper
controllers/alvara.php	helper	url
controllers/alvara.php	library	form_validation
controllers/alvara.php	library	layout
controllers/alvara.php	library	pagination
controllers/alvara.php	model	AlvaraModel
controllers/alvara.php	model	estados_alvaramodel
controllers/alvara.php	model	ficheiromodel
controllers/alvara.php	model	processomodel
controllers/alvara.php	model	profilemodel
controllers/alvara.php	model	UtilizadoresModel
controllers/alvara.php	plugin	js_calendar
controllers/appmain.php	view	site/main
controllers/categoria_social.php	helper	khacl_helper
controllers/categoria_social.php	helper	url
controllers/categoria_social.php	library	form_validation
controllers/categoria_social.php	library	layout
controllers/categoria_social.php	library	pagination
controllers/categoria_social.php	model	categoria_socialmodel
controllers/categoria_social.php	model	UtilizadoresModel
controllers/contacto.php	helper	khacl_helper
controllers/contacto.php	helper	url
controllers/contacto.php	library	form_validation
controllers/contacto.php	library	layout
controllers/contacto.php	library	pagination
controllers/contacto.php	model	ContactoModel
controllers/contacto.php	model	processomodel
controllers/contacto.php	model	tipo_contactomodel
controllers/contacto.php	model	UtilizadoresModel
controllers/contador.php	helper	khacl_helper
controllers/contador.php	helper	url
controllers/contador.php	library	form_validation
controllers/contador.php	library	layout
controllers/contador.php	library	pagination
controllers/contador.php	model	contadormodel
controllers/contador.php	model	UtilizadoresModel
controllers/documento.php	helper	khacl_helper
controllers/documento.php	helper	url
controllers/documento.php	library	form_validation
controllers/documento.php	library	layout

Classes	Tipo de classe a	Relação
controllers/documento.php	library	pagination
controllers/documento.php	model	documentomodel
controllers/documento.php	model	ficheiromodel
controllers/documento.php	model	PessoasModel
controllers/documento.php	model	profilemodel
controllers/documento.php	model	tipo_documentomodel
controllers/documento.php	model	UtilizadoresModel
controllers/documentos_processo.php	helper	khacl_helper
controllers/documentos_processo.php	helper	url
controllers/documentos_processo.php	library	form_validation
controllers/documentos_processo.php	library	layout
controllers/documentos_processo.php	library	pagination
controllers/documentos_processo.php	model	Documentos_processoModel
controllers/documentos_processo.php	model	processomodel
controllers/documentos_processo.php	model	UtilizadoresModel
controllers/estados_alvara.php	helper	khacl_helper
controllers/estados_alvara.php	helper	url
controllers/estados_alvara.php	library	form_validation
controllers/estados_alvara.php	library	layout
controllers/estados_alvara.php	library	pagination
controllers/estados_alvara.php	model	estados_alvaramodel
controllers/estados_alvara.php	model	UtilizadoresModel
controllers/estrutura_organica.php	helper	khacl_helper
controllers/estrutura_organica.php	helper	url
controllers/estrutura_organica.php	library	form_validation
controllers/estrutura_organica.php	library	layout
controllers/estrutura_organica.php	library	pagination
controllers/estrutura_organica.php	model	estrutura_organicamodel
controllers/estrutura_organica.php	model	PessoasModel
controllers/estrutura_organica.php	model	UtilizadoresModel
controllers/ficheiro.php	helper	khacl_helper
controllers/ficheiro.php	helper	url
controllers/ficheiro.php	library	form_validation
controllers/ficheiro.php	library	layout
controllers/ficheiro.php	library	pagination
controllers/ficheiro.php	model	ficheiromodel
controllers/ficheiro.php	model	tipo_ficheiromodel
controllers/ficheiro.php	model	UtilizadoresModel
controllers/file.php	helper	khacl_helper
controllers/file.php	helper	url
controllers/file.php	model	ficheiromodel
controllers/file.php	model	UtilizadoresModel
controllers/freguesia.php	helper	khacl_helper
controllers/freguesia.php	helper	url
controllers/freguesia.php	library	form_validation
controllers/freguesia.php	library	layout
controllers/freguesia.php	library	pagination
controllers/freguesia.php	model	freguesiamodel
controllers/freguesia.php	model	UtilizadoresModel
controllers/habilitacoes_e_comprovativos.php	helper	khacl_helper
controllers/habilitacoes_e_comprovativos.php	helper	url
controllers/habilitacoes_e_comprovativos.php	library	form_validation
controllers/habilitacoes_e_comprovativos.php	library	layout
controllers/habilitacoes_e_comprovativos.php	library	pagination
controllers/habilitacoes_e_comprovativos.php	model	ficheiromodel
controllers/habilitacoes_e_comprovativos.php	model	habilitacoes_e_comprovativosmodel
controllers/habilitacoes_e_comprovativos.php	model	profilemodel
controllers/habilitacoes_e_comprovativos.php	model	UtilizadoresModel
controllers/habilitacoes_e_comprovativos.php	plugin	js_calendar
controllers/informacao.php	helper	khacl_helper

Classes	Tipo de classe a	Relação
controllers/informacao.php	helper	url
controllers/informacao.php	library	form_validation
controllers/informacao.php	library	layout
controllers/informacao.php	library	pagination
controllers/informacao.php	model	informacaomodel
controllers/informacao.php	model	UtilizadoresModel
controllers/mensagem.php	helper	khacl_helper
controllers/mensagem.php	helper	url
controllers/mensagem.php	library	form_validation
controllers/mensagem.php	library	layout
controllers/mensagem.php	library	pagination
controllers/mensagem.php	model	mensagemmodel
controllers/mensagem.php	model	PessoasModel
controllers/mensagem.php	model	UtilizadoresModel
controllers/niveis_acesso_estrutura_organica.php	helper	khacl_helper
controllers/niveis_acesso_estrutura_organica.php	helper	url
controllers/niveis_acesso_estrutura_organica.php	library	form_validation
controllers/niveis_acesso_estrutura_organica.php	library	layout
controllers/niveis_acesso_estrutura_organica.php	library	pagination
controllers/niveis_acesso_estrutura_organica.php	model	niveis_acesso_estrutura_organicamodel
controllers/niveis_acesso_estrutura_organica.php	model	UtilizadoresModel
controllers/notificacao.php	helper	khacl_helper
controllers/notificacao.php	helper	url
controllers/notificacao.php	library	form_validation
controllers/notificacao.php	library	layout
controllers/notificacao.php	library	pagination
controllers/notificacao.php	model	notificacaomodel
controllers/notificacao.php	model	oficiomodel
controllers/notificacao.php	model	processomodel
controllers/notificacao.php	model	tipo_notificacaomodel
controllers/notificacao.php	model	UtilizadoresModel
controllers/notificacao.php	plugin	js_calendar
controllers/notificacao_preferencial.php	helper	khacl_helper
controllers/notificacao_preferencial.php	helper	url
controllers/notificacao_preferencial.php	library	form_validation
controllers/notificacao_preferencial.php	library	layout
controllers/notificacao_preferencial.php	library	pagination
controllers/notificacao_preferencial.php	model	notificacao_preferencialmodel
controllers/notificacao_preferencial.php	model	UtilizadoresModel
controllers/oficio.php	helper	khacl_helper
controllers/oficio.php	helper	url
controllers/oficio.php	library	form_validation
controllers/oficio.php	library	layout
controllers/oficio.php	library	pagination
controllers/oficio.php	model	ficheiromodel
controllers/oficio.php	model	oficiomodel
controllers/oficio.php	model	processomodel
controllers/oficio.php	model	profilemodel
controllers/oficio.php	model	UtilizadoresModel
controllers/oficio.php	plugin	js_calendar
controllers/perfis_utilizador.php	helper	khacl_helper
controllers/perfis_utilizador.php	helper	url
controllers/perfis_utilizador.php	library	form_validation
controllers/perfis_utilizador.php	library	layout
controllers/perfis_utilizador.php	library	pagination
controllers/perfis_utilizador.php	model	perfis_utilizadormodel
controllers/perfis_utilizador.php	model	UtilizadoresModel
controllers/pessoas.php	helper	khacl_helper
controllers/pessoas.php	helper	url
controllers/pessoas.php	library	form_validation

Classes	Tipo de classe a	Relação
controllers/pessoas.php	library	layout
controllers/pessoas.php	library	pagination
controllers/pessoas.php	model	Categoria_socialModel
controllers/pessoas.php	model	pessoasmodel
controllers/pessoas.php	model	UtilizadoresModel
controllers/processo.php	helper	khacl_helper
controllers/processo.php	helper	url
controllers/processo.php	library	form_validation
controllers/processo.php	library	layout
controllers/processo.php	library	pagination
controllers/processo.php	model	FreguesiaModel
controllers/processo.php	model	PessoasModel
controllers/processo.php	model	processomodel
controllers/processo.php	model	Tipo_construcaoModel
controllers/processo.php	model	Tipo_processoModel
controllers/processo.php	model	Tipo_utilizacaoModel
controllers/processo.php	model	UtilizadoresModel
controllers/processos_processo.php	helper	khacl_helper
controllers/processos_processo.php	helper	url
controllers/processos_processo.php	library	form_validation
controllers/processos_processo.php	library	layout
controllers/processos_processo.php	library	pagination
controllers/processos_processo.php	model	processomodel
controllers/processos_processo.php	model	Processos_processoModel
controllers/processos_processo.php	model	UtilizadoresModel
controllers/profile.php	helper	khacl_helper
controllers/profile.php	helper	url
controllers/profile.php	library	form_validation
controllers/profile.php	library	layout
controllers/profile.php	model	categoria_socialmodel
controllers/profile.php	model	documentomodel
controllers/profile.php	model	documentos_processomodel
controllers/profile.php	model	ficheiromodel
controllers/profile.php	model	freguesiamodel
controllers/profile.php	model	habilitacoes_e_comprovativosmodel
controllers/profile.php	model	perfis_utilizadormodel
controllers/profile.php	model	pessoasmodel
controllers/profile.php	model	processomodel
controllers/profile.php	model	profilemodel
controllers/profile.php	model	termos_de_aceitacaomodel
controllers/profile.php	model	tipo_construcaomodel
controllers/profile.php	model	tipo_documentomodel
controllers/profile.php	model	tipo_notificacaomodel
controllers/profile.php	model	tipo_processomodel
controllers/profile.php	model	tipo_utilizacaomodel
controllers/profile.php	model	tramitacaomodel
controllers/profile.php	model	UtilizadoresModel
controllers/profile.php	plugin	captcha
controllers/profile.php	plugin	js_calendar
controllers/projectistas_estruturas_organicas.php	helper	khacl_helper
controllers/projectistas_estruturas_organicas.php	helper	url
controllers/projectistas_estruturas_organicas.php	library	form_validation
controllers/projectistas_estruturas_organicas.php	library	layout
controllers/projectistas_estruturas_organicas.php	library	pagination
controllers/projectistas_estruturas_organicas.php	model	projectistas_estruturas_organicasmodel
controllers/projectistas_estruturas_organicas.php	model	UtilizadoresModel
controllers/servico.php	helper	khacl_helper
controllers/servico.php	helper	url
controllers/servico.php	library	form_validation
controllers/servico.php	library	layout

Classes	Tipo de classe a	Relação
controllers/servico.php	library	pagination
controllers/servico.php	model	servicomodel
controllers/servico.php	model	UtilizadoresModel
controllers/termos_de_aceitacao.php	helper	khacl_helper
controllers/termos_de_aceitacao.php	helper	url
controllers/termos_de_aceitacao.php	library	form_validation
controllers/termos_de_aceitacao.php	library	layout
controllers/termos_de_aceitacao.php	library	pagination
controllers/termos_de_aceitacao.php	model	ficheiromodel
controllers/termos_de_aceitacao.php	model	profilemodel
controllers/termos_de_aceitacao.php	model	termos_de_aceitacaomodel
controllers/termos_de_aceitacao.php	model	UtilizadoresModel
controllers/termos_de_aceitacao.php	plugin	js_calendar
controllers/tipo_construcao.php	helper	khacl_helper
controllers/tipo_construcao.php	helper	url
controllers/tipo_construcao.php	library	form_validation
controllers/tipo_construcao.php	library	layout
controllers/tipo_construcao.php	library	pagination
controllers/tipo_construcao.php	model	tipo_construcaomodel
controllers/tipo_construcao.php	model	UtilizadoresModel
controllers/tipo_contacto.php	helper	khacl_helper
controllers/tipo_contacto.php	helper	url
controllers/tipo_contacto.php	library	form_validation
controllers/tipo_contacto.php	library	layout
controllers/tipo_contacto.php	library	pagination
controllers/tipo_contacto.php	model	tipo_contactomodel
controllers/tipo_contacto.php	model	UtilizadoresModel
controllers/tipo_documento.php	helper	khacl_helper
controllers/tipo_documento.php	helper	url
controllers/tipo_documento.php	library	form_validation
controllers/tipo_documento.php	library	layout
controllers/tipo_documento.php	library	pagination
controllers/tipo_documento.php	model	tipo_documentomodel
controllers/tipo_documento.php	model	UtilizadoresModel
controllers/tipo_ficheiro.php	helper	khacl_helper
controllers/tipo_ficheiro.php	helper	url
controllers/tipo_ficheiro.php	library	form_validation
controllers/tipo_ficheiro.php	library	layout
controllers/tipo_ficheiro.php	library	pagination
controllers/tipo_ficheiro.php	model	tipo_ficheiromodel
controllers/tipo_ficheiro.php	model	UtilizadoresModel
controllers/tipo_notificacao.php	helper	khacl_helper
controllers/tipo_notificacao.php	helper	url
controllers/tipo_notificacao.php	library	form_validation
controllers/tipo_notificacao.php	library	layout
controllers/tipo_notificacao.php	library	pagination
controllers/tipo_notificacao.php	model	tipo_notificacaomodel
controllers/tipo_notificacao.php	model	UtilizadoresModel
controllers/tipo_processo.php	helper	khacl_helper
controllers/tipo_processo.php	helper	url
controllers/tipo_processo.php	library	form_validation
controllers/tipo_processo.php	library	layout
controllers/tipo_processo.php	library	pagination
controllers/tipo_processo.php	model	tipo_processomodel
controllers/tipo_processo.php	model	UtilizadoresModel
controllers/tipo_utilizacao.php	helper	khacl_helper
controllers/tipo_utilizacao.php	helper	url
controllers/tipo_utilizacao.php	library	form_validation
controllers/tipo_utilizacao.php	library	layout
controllers/tipo_utilizacao.php	library	pagination

Classes	Tipo de classe a	Relação
controllers/tipo_utilizacao.php	model	tipo_utilizacaomodel
controllers/tipo_utilizacao.php	model	UtilizadoresModel
controllers/tramitacao.php	helper	khacl_helper
controllers/tramitacao.php	helper	url
controllers/tramitacao.php	library	form_validation
controllers/tramitacao.php	library	layout
controllers/tramitacao.php	library	pagination
controllers/tramitacao.php	model	documentomodel
controllers/tramitacao.php	model	ficheiromodel
controllers/tramitacao.php	model	profilemodel
controllers/tramitacao.php	model	tramitacaoModel
controllers/tramitacao.php	model	UtilizadoresModel
controllers/tramitacao.php	plugin	js_calendar
controllers/utilizadores.php	helper	khacl_helper
controllers/utilizadores.php	helper	url
controllers/utilizadores.php	library	form_validation
controllers/utilizadores.php	library	layout
controllers/utilizadores.php	library	pagination
controllers/utilizadores.php	model	perfis_utilizadormodel
controllers/utilizadores.php	model	PessoasModel
controllers/utilizadores.php	model	tipo_notificacaomodel
controllers/utilizadores.php	model	UtilizadoresModel
controllers/utilizadores_servico.php	helper	khacl_helper
controllers/utilizadores_servico.php	helper	url
controllers/utilizadores_servico.php	library	form_validation
controllers/utilizadores_servico.php	library	layout
controllers/utilizadores_servico.php	library	pagination
controllers/utilizadores_servico.php	model	utilizadores_servicomodel
controllers/utilizadores_servico.php	model	UtilizadoresModel
controllers/welcome.php	helper	khacl_helper
controllers/welcome.php	library	khacl
controllers/welcome.php	library	layout
models/aclsmodel.php	library	khacl
models/alvaramodel.php	model	ContadorModel
models/documentomodel.php	model	ContadorModel
models/ficheiromodel.php	model	tipo_ficheiromodel
models/ficheiromodel.php	model	UtilizadoresModel
models/oficiomodel.php	model	ContadorModel
models/perfis_utilizadormodel.php	library	khacl
models/processomodel.php	model	ContadorModel
models/profilemodel.php	model	UtilizadoresModel
models/utilizadoresmodel.php	helper	url
models/utilizadoresmodel.php	library	khacl
models/utilizadoresmodel.php	library	redux_auth
models/utilizadoresmodel.php	model	perfis_utilizadormodel
models/utilizadoresmodel.php	model	pessoasmodel

Anexo F - Descrição das tabelas da base de dados

agenda	Tabela onde são armazenados os eventos agendados
alvara	Tabela onde são armazenados os alvarás
categoria_social	Tabela onde são armazenadas as categorias sociais
contacto	Tabela onde são armazenados os contactos
contador	Tabela onde são armazenados os numeradores do sistema
documento	Tabela onde são armazenados os documentos existentes
documentos_processo	Tabela onde se armazena a relação entre os documentos e os processos
estados_alvara	Tabela onde são armazenados os estados possíveis de cada alvará
estados_processo	Tabela onde são armazenados os estados possíveis de cada processo
estrutura_organica	Tabela onde são armazenadas as estruturas orgânicas existentes
ficheiro	Tabela onde são armazenadas as informações de cada ficheiro
freguesia	Tabela onde são armazenadas as freguesias
habilitacoes_e_comprovativos	Tabela onde são armazenadas os comprovativos das habilitações dos projectistas
informacao	Tabela onde são armazenadas as informações inseridas
khacl_access	Tabela onde são armazenadas as regras de acesso dos utilizadores
khacl_acos	Tabela onde são armazenadas os “access control objects“
khacl_aros	Tabela onde são armazenadas os “access request objects”
khacl_axos	Tabela onde são armazenadas os “access eXtension objects”
mensagem	Tabela onde são armazenadas as mensagens
notificacao	Tabela onde são armazenadas as notificações emitidas
oficio	Tabela onde são armazenados os ofícios criados
perfis_utilizador	Tabela onde são armazenados os perfis de utilizadores
pessoas	Tabela onde são armazenadas as pessoas
processo	Tabela onde são armazenados os processos
processos_processo	Tabela onde se armazena a relação entre os diversos processos, quando existente
projectistas_estruturas_organicas	Tabela onde se armazena a estrutura orgânica a que cada projectista pertence
servico	Tabela onde são armazenados os serviços municipais
termos_de_aceitacao	Tabela onde são armazenados os termos de aceitação
tipo_construcao	Tabela onde são armazenados os tipos de construção
tipo_contacto	Tabela onde são armazenados os tipos de contacto
tipo_documento	Tabela onde são armazenados os tipos de documento
tipo_ficheiro	Tabela onde são armazenados os tipos de ficheiro
tipo_notificacao	Tabela onde são armazenados os tipos de notificações
tipo_processo	Tabela onde são armazenados os tipos de processos
tipo_utilizacao	Tabela onde são armazenados os tipos de utilização
tramitacao	Tabela onde são armazenadas as tramitações
utilizadores	Tabela onde são armazenados os utiliadores
utilizadores_servico	Tabela onde se armazena a relação entre os utilizadores e os serviços municipais