



UNIVERSIDADE DE COIMBRA

FACULDADE DE CIÊNCIAS E TECNOLOGIA

Departamento de Engenharia Informática

Programação Avançada em Java

Relatório do Projecto Final

Plataforma para Gestão de Candidaturas

Grupo 10:

2014237447 - Rafaela Lourenço

2014237453 - Ricardo Alves

Conteúdo

| | |
|---|----|
| Índice de figuras | 3 |
| Introdução | 4 |
| Preparação do projecto | 5 |
| Camada de apresentação - Web | 6 |
| Camada de negócio - EJB | 8 |
| Camada de dados -JPA | 9 |
| Empacotamento | 10 |
| Testes e <i>Logging</i> | 12 |
| Conclusão | 13 |
| Anexos | 14 |
| Anexo 1 Mockups | 14 |
| Anexo 1.1 Áreas individuais de utilizadores | 14 |
| Anexo 1.2 Páginas de login individual | 15 |
| Anexo 1.3 Outros protótipos de páginas criados | 16 |
| Anexo 2 Diagramas Conceptual e Físico | 24 |
| Anexo 3 Empacotamento do projecto | 27 |
| Anexo 3.1 Empacotamento da camada JPA | 27 |
| Anexo 3.2 Empacotamento camada EJB | 27 |
| Anexo 3.3 Empacotamento camada Web | 28 |
| Anexo 4 Testes realizados à camada EJB | 30 |
| Anexo 5 Diagrama UML final | 31 |
| Anexo 5.1 Diagrama UML camada JPA | 31 |

Índice de figuras

| | |
|--|----|
| Figura 1 - Aspecto da página pública de descrição da empresa | 5 |
| Figura 2 - Aspecto da página pública de apresentação das posições disponíveis | 5 |
| Figura 3 - Exemplo da página xhtml da nossa aplicação referente à <i>Home Page</i> | 6 |
| Figura 4 - Diagrama ER (obtido a partir do software PowerDesigner)..... | 9 |
| Figura 5 - Diagrama Físico (gerado a partir do software PowerDesigner) | 10 |
| Figura 6 - Visão geral do empacotamento do projecto..... | 10 |
| Figura 7 - Aspecto da área reservada para o utilizador Administrador | 14 |
| Figura 8 - Aspecto da área reservada para o utilizador Gestor (Manager)..... | 14 |
| Figura 9 - Aspecto da área reservada para o utilizador Entrevistador(Interviewer)..... | 14 |
| Figura 10 - Aspecto da área reservada para o Candidato (Candidate) | 15 |
| Figura 11 - Página de login de empregados (employees) | 15 |
| Figura 12 - Página de login para os candidatos | 15 |
| Figura 13 - Aspecto da página de criação de nova posição..... | 16 |
| Figura 14 - Apresentação dos detalhes de uma dada apresentação selecionada | 16 |
| Figura 15 - Pop up de envio de sugestao de posição a amigo | 17 |
| Figura 16 - Aspecto da página de posições para cada um dos utilizadores do tipo empregado (Employee) | 17 |
| Figura 17 - Aspecto da página de edição de posições (não implementado)..... | 18 |
| Figura 18 - Aspecto da página de apresentação de todos os guioes existentes | 18 |
| Figura 19 - Aspecto da página de pesquisa de candidaturas (válido para administrador, gestor e entrevistador) e inserir novas candidaturas na base de dados (administrador e gestor) | 19 |
| Figura 20 - Aspecto da página de criação de relatórios (não implementada) | 19 |
| Figura 21 - Aspecto da página das posições de um dado gestor logado..... | 20 |
| Figura 22 - Aspecto da página das entrevistas de um dado entrevistador..... | 20 |
| Figura 23 - Aspecto da página de preenchimento de um guião após a entrevista..... | 21 |
| Figura 24 - Apresentação das candidaturas de um candidato | 21 |
| Figura 25 - Apresentação das entrevistas de um candidato | 21 |
| Figura 26 - Apresentação dos dados do perfil do candidato, permitindo a sua edição..... | 22 |
| Figura 27 - Registo de um novo candidato..... | 22 |
| Figura 28 - Aspecto da página de candidatura a uma dada posição..... | 23 |
| Figura 29 – Diagrama Entidade-Relacionamento (obtido a partir do software PowerDesigner) | 25 |
| Figura 30 – Diagrama Conceptual (obtido a partir do software PowerDesigner) | 26 |
| Figura 31 - Empacotamento da camada JPA | 27 |
| Figura 32 - Empacotamento da camada EJB | 27 |
| Figura 33 - Empacotamento da camada Web e testes realizados | 28 |
| Figura 34 - Estrutura da parte web | 28 |
| Figura 35 - Visão geral da distribuição da parte web da aplicação | 29 |
| Figura 36 - Testes realizados à camada EJB..... | 30 |
| Figura 37 - Diagrama UML final da camada JPA..... | 31 |

Introdução

O propósito da realização deste projecto final foi o de construir uma plataforma para a gestão de candidaturas. Respondendo deste modo ao desafio lançado pela empresa *CRITICAL SOFTWARE*, dadas as actuais necessidades das organizações face à eficiência nos processos de recrutamento e de controlo e acessos a informações actualizadas.

O ponto de partida deste trabalho consistiu na elaboração exaustiva dos requisitos funcionais do sistema, da prototipagem rápida através da criação de esquemas de ecrã (*mockups*) e da concepção de uma lista de perguntas/dúvidas existentes para expor ao *Cliente*. Este processo levou cerca de duas semanas até ficar devidamente definido.

Posteriormente, concentrámos a nossa atenção no *design* do projecto, nomeadamente na criação de classes e de modelos de dados. Este planeamento foi talvez o mais exigente e delicado de todo o trabalho, mas deveras útil após finalizado. O programa foi todo escrito em *JavaEE* recorrendo ao *Wildfly* para se fazer o *deploy* da página *web*. Havia vários requisitos a cumprir na aplicação, e portanto foi necessário fazer uma correcta abordagem ao problema, pensando na arquitectura antes de começar propriamente a trabalhar. Foi feito o diagrama de classes inicial, o diagrama *ER* e também pensados os tempos necessários à execução de cada tarefa.

Depressa nos apercebemos que existiam três partes fundamentais no projecto, e era sobre essas que devíamos concentrar os nossos esforços: a camada de apresentação, o nosso negócio (código java clássico) e o acesso à base de dados. Dominando bem estas três componentes em princípio teríamos o trabalho facilitado, e a ideia do grupo foi sempre de ambos estarmos por dentro de todas as partes do trabalho, até para nos podermos ajudar quando surgissem dúvidas/problemas.

O trabalho era bem direccionado para aquilo que vamos fazer de futuro, e portanto considerámos ambos ser da maior importância compreender bem a estrutura e adquirir os conhecimentos necessários para estarmos à vontade em futuros projectos com a mesma arquitectura.

Neste relatório focam-se os principais pontos do projecto: a camada de apresentação, a camada de negócio, a camada de dados, a arquitectura do programa e uma breve conclusão.

Preparação do projecto

Antes da implementação em código Java da nossa aplicação foram desenvolvidos vários protótipos da futura aplicação. A ferramenta utilizada foi o *Balsamiq Mockups* 3. Estes protótipos permitiram, e muito, reduzir o tempo do desenvolvimento das páginas web da aplicação.

Na Figura 1 e na Figura 2 encontram-se o *mockups* para a páginas públicas da aplicação para a descrição da empresa e as posições em aberto

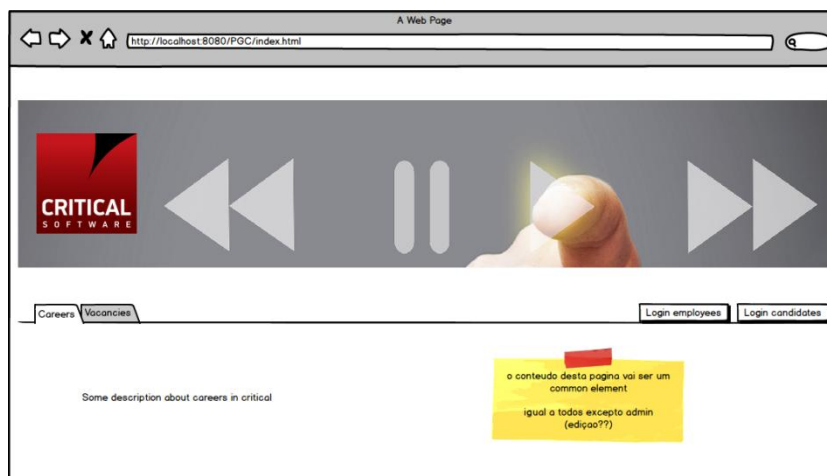


Figura 1 - Aspecto da página pública de descrição da empresa



Figura 2 - Aspecto da página pública de apresentação das posições disponíveis

Foram igualmente criados *mockups* para cada uma das áreas reservadas de cada um dos tipos de utilizadores (Anexo 1.1), páginas de login individual (Anexo 1.2), e algumas páginas específicas para cada um dos perfis: pesquisa de candidatos e de posições, criação de posições, criação de guiões, submissão de feedback das entrevistas, entre outros (Anexo 1.3).

Camada de apresentação - Web

A camada de apresentação foi feita sempre a pensar num *layout* que fosse bastante intuitivo para o utilizador, sem demasiada informação em cada página, mas com o suficiente para não deixar o utilizador sem saber o que fazer.

Esta camada foi desenvolvida segundo um *template* de *CommonLayout*, onde existem três secções que são comuns a todas as páginas: um *CommonHeader*, um *CommonContent* e um *CommonFooter*. Existem também páginas que possuem elementos que são partilhados por vários utilizadores do sistema.

Ficando assim as páginas com o *template* apresentado na Figura 3.



Figura 3 - Exemplo da página xhtml da nossa aplicação referente à Home Page

A utilização deste tipo de *template* na construção de um *website* permite que não seja necessário repetir o código dos elementos que são comuns a todas as páginas, sendo apenas escrito na página *xhtml* correspondente a essa secção. Permite também que a correção e formatação desse código seja efetuada de um modo mais fácil e rápido.

Relativamente à parte da autenticação e segurança, em projectos realizados anteriormente recorremos ao uso de *filtros* para vedar o acesso a páginas específicas sem a devida permissão no acto do *Login*. O trabalho actual dispensa o uso de *filtros*, pois neste caso foi usado um outro mecanismo de segurança. Este mecanismo é do tipo *Application-layer Security*, no qual a segurança é feita pelo *container*. Neste trabalho protegemos o acesso ao nosso *website* através do mecanismo de autenticação denominado de *Form-based authentication*.

Foi também garantida a segurança no armazenamento de *password* dos utilizadores, visto que esta é sempre encriptada aquando a sua inserção, e nunca desencriptada, para que não seja de todo possível aceder às *password* de um utilizador em claro. Por exemplo, se um utilizador tentar fazer *login* com determinada *password*, esta é primeiro encriptada e só depois é comparado o seu valor com o que está na base de dados, para permitir ou não, o acesso à aplicação. Esta é sem dúvida uma boa prática, pois em caso de uma eventual fuga de informação da base de dados, esta não permite o acesso às *password* dos utilizadores.

Para a encriptação propriamente dita, importámos a biblioteca *java.security* que faculta o acesso a várias classes e interfaces para uma *framework* de segurança. O algoritmo de encriptação utilizado para o efeito é o “SHA-256”.

Camada de negócio - EJB

A camada EJB, assume todo o negócio da nossa aplicação, sendo responsável por fazer a ponte entre a camada da *web*, e a camada que detém o acesso à base de dados. Por outras palavras, a camada de negócio funciona como um intermediário que gere todos os acessos e operações que impliquem a consulta à base de dados.

Efectivamente, existiu sempre o cuidado de apenas deixar passar para a camada de apresentação os dados a que o utilizador deve ter acesso, sem permitir o acesso a informação da base de dados que não lhe diga respeito. Nesta camada é feito também o tratamento de grande parte das excepções da aplicação, tendo sempre o cuidado de apresentar ao utilizador mensagens de informação ou de erro, em todos os passos que este dá.

À excepção do *EJB* responsável por popular a base de dados no arranque da aplicação, todos os outros foram devidamente designados para que se saiba qual a sua função. Por exemplo, ao *EJB* responsável pelo acesso à parte das “Posições” da base de dados, chamámos *PositionDAO*, em que *DAO* significa *data access object*, ou objecto responsável por aceder aos dados.

Assim fica bem claro, quais os *EJB*'s responsáveis por aceder a determinados dados. Aprofundando um pouco mais, o *UserDAO* por exemplo, tem métodos para pesquisar todos os utilizadores da aplicação, pesquisar apenas um utilizador com o seu *id*, etc. Neste último caso decidimos que seria sensato, passar todas as suas músicas para um gestor, para que estas não deixassem de estar disponíveis para outros utilizadores. O *UserRegister* é também um *EJB* responsável por aceder a dados dos utilizadores, mas neste caso concreto, serve para verificar se o login está correcto, ou para adicionar um novo utilizador à base de dados.

Na nossa opinião não era necessário fazer demasiadas *NamedQueries*, pois estas são carregadas logo no arranque da aplicação e isso poderia sobrecarregar de forma desnecessária a aplicação, assim tendo a certeza que cada método chama apenas uma *query* temos a garantia de maior rapidez para o utilizador.

De modo a podermos aceder à aplicação ao longo do seu desenvolvimento e ao posterior envio de notificações, foi criada uma classe “Populate.java” (existente na camada de negócio) onde criados objectos de cada uma das entidades que vão permitir preencher a base de dados da aplicação. Na Tabela 1 estão os dados de login dos utilizadores e candidatos criados para testar a aplicação.

Tabela 1 - Dados dos utilizadores e candidatos para acesso à aplicação

| Role | Username | Password |
|---------------|---------------------------|-----------------|
| Administrador | adpfrlra@gmail.com | 123 |
| Gestor | mapfrlra@gmail.com | 123 |
| Entrevistador | inpfrlra@gmail.com | 123 |
| Candidato | lourencorafaela@gmail.com | 456 |
| Candidato | rickardalves@hotmail.com | 456 |

Camada de dados -JPA

De acordo com o nosso diagrama *ER (Entidade - Relacionamento)*, este projecto possui sete entidades principais relacionadas entre si. Assim o diagrama *ER* final é o seguinte:

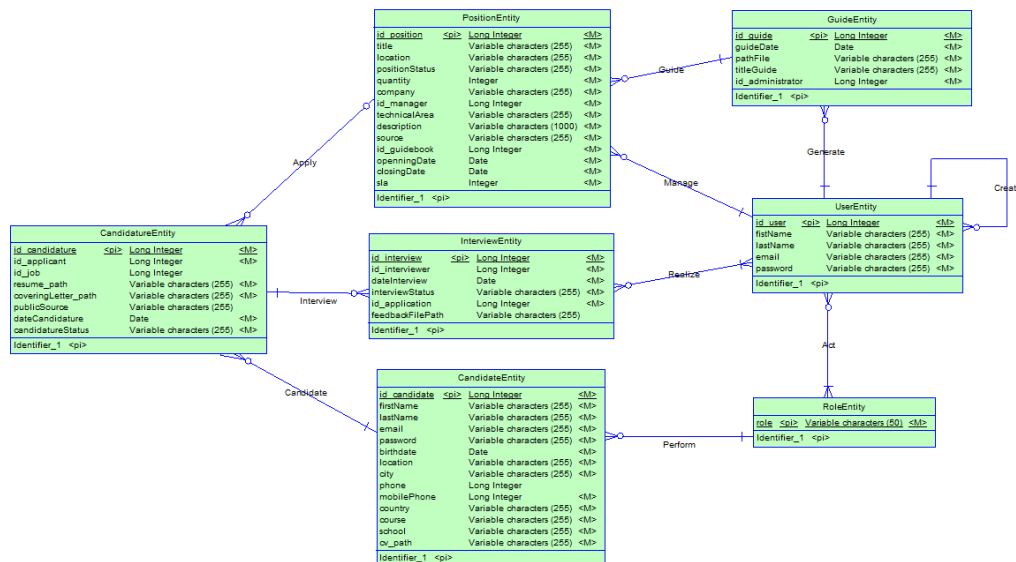


Figura 4 - Diagrama ER (obtido a partir do software PowerDesigner)

É com base no relacionamento entre as ilustradas entidades que funciona toda a ideia deste projecto, tivemos por isso especial cuidado na criação das entidades, com todas as anotações necessárias para termos a garantia de ter uma base de dados fiável e o menos vulnerável possível.

Cada entidade foi também criada com os atributos que pensamos serem essenciais para persistir na base de dados, não iremos fazer uma lista detalhada dos atributos de cada entidade por acharmos redundante. Com base no diagrama *ER* descrito acima foi possível gerar o diagrama físico correspondente que ilustra as tabelas criadas através do relacionamento entre as entidades e seus comportamentos conforme é exibido na Figura 5 .

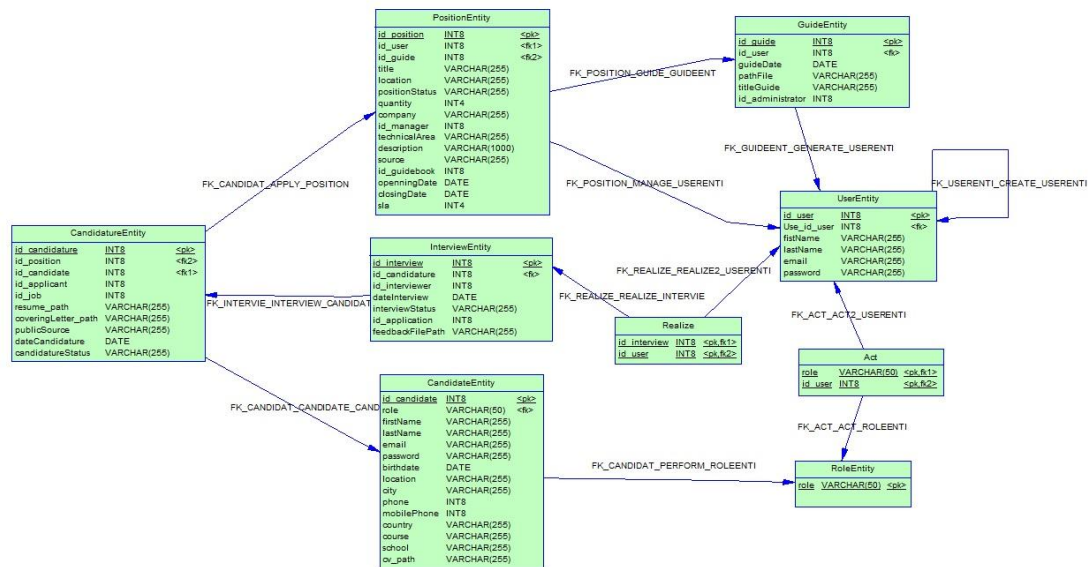


Figura 5 - Diagrama Físico (gerado a partir do software PowerDesigner)

Para uma melhor visualização estes diagramas encontram-se no Anexo 2.

Apos a realização do projecto foi elaborado um diagrama UML (ferramenta code2uml) onde é possível a observação das associações entre cada classe utilizada. Este diagrama encontra-se no Anexo 5.1.

Empacotamento

Tal como já foi mencionado a aplicação foi organizada em camadas, por forma a permitir a separação de artefactos, facilitando a manutenção de código, e futuros desenvolvimentos da aplicação. Na Figura 6 encontra-se uma imagem que mostra o empacotamento do nosso projecto.

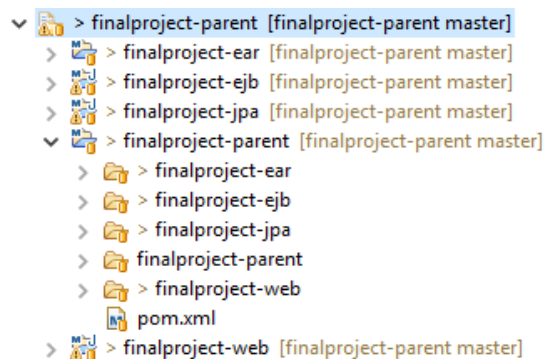


Figura 6 - Visão geral do empacotamento do projecto

Garantimos deste modo que cada camada possa existir de forma independente. Por exemplo, a camada de mais baixo nível (*JPA*), onde se encontram as entidades, não possui conhecimentos das outras camadas, não comunicando assim com os outros artefactos que constituem o projecto. A informação relativa a esta camada, é persistida na base de dados, sem saber das outras camadas (vista do empacotamento no Anexo 3.1).

Os *EJB's* (*Enterprise Java Beans*) estão numa outra camada, denominada camada de serviço, onde é feita a gestão das entidades da aplicação, oferecendo a possibilidade de criar, actualizar, listar ou remover dados das entidades. Mais uma vez, esta camada não tem conhecimento das camadas acima de si, devolvendo dados sem saber para onde (vista do empacotamento no Anexo 3.2).

De seguida temos a camada de apresentação, que contém todas as páginas que o utilizador vai ver, e que é responsável por toda a interacção com o utilizador da aplicação (vista do empacotamento da camada *Web* no Anexo 3.3). Aqui é feito o acesso aos *EJB's* que transmitem a informação necessária a apresentar ao utilizador. Por fim, o módulo *EAR* é apenas responsável por empacotar todos os módulos num só para possibilitar a integração, transporte e instalação no servidor.

Todos estes quatro artefactos têm o seu próprio *pom.xml* mas, estão dentro de um projecto *Maven* a que se deu o nome de *parent* que tem a função de definir e fornecer as dependências e *plugins* que os outros módulos utilizam.

Testes e *Logging*

Em relação aos testes efectuados à nossa aplicação estes incidem predominantemente na comunicação da aplicação com a Base de Dados. Tivemos a preocupação de não efectuar testes que afectassem a base de dados (testando só consultas ou erros em pedidos de manipulação (imagem no Anexo 4), já que não queríamos que a base de dados fosse alterada pelos testes. A autenticação feita no login foi igualmente testada.

Relativamente ao *logging*, usámos uma vez mais a configuração previamente definida através do servidor *Wildfly*. Múltiplas mensagens de *logging* foram introduzidas na maioria das classes implementadas, o que nos permitiu monitorizar com detalhe a execução do programa facilitando a correcção de eventuais falhas ou anomalias.

Conclusão

O projecto em si foi de todo aliciante, exigindo uma arquitectura já muito mais avançada e com um tema que particularmente interessante.

Seria igualmente interessante implementar neste projecto uma ferramenta para criar certificados de segurança (*https*), nomeadamente a “*keytool*” existente na *JDK/bin*, melhorando assim a segurança global de toda a aplicação.

Apesar de toda a aplicação estar a funcionar correctamente e de responder com eficiência ao que era solicitado pelo enunciado do “Cliente”, infelizmente por limitações de tempo não nos foi possível terminar a parte correspondente à criação de relatórios para os administradores do sistema. Lamentamos ainda a não implementação dos requisitos opcionais pelas mesmas razões acima mencionadas.

Devido a este atraso não nos foi possível testar a aplicação tão exhaustivamente como desejávamos, e não pudemos implementar algumas das ideias que tínhamos. Foi mesmo bastante complicado cumprir o prazo, e apenas foi possível com bastante esforço de ambos os elementos do grupo, que diga-se funcionou na perfeição para o pouco tempo disponível.

Apesar dos contratempos sentidos consideramos que a aplicação é bastante apelativa, funcional, intuitiva e segura para o utilizador.

Anexos

Anexo 1 Mockups

Anexo 1.1 Áreas individuais de utilizadores

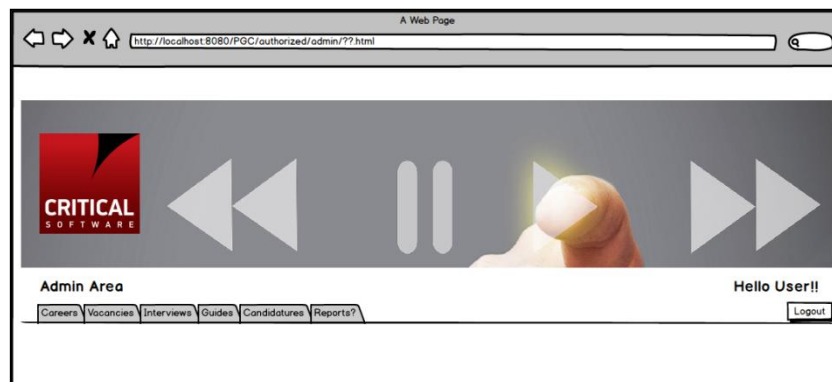


Figura 7 - Aspecto da área reservada para o utilizador Administrador

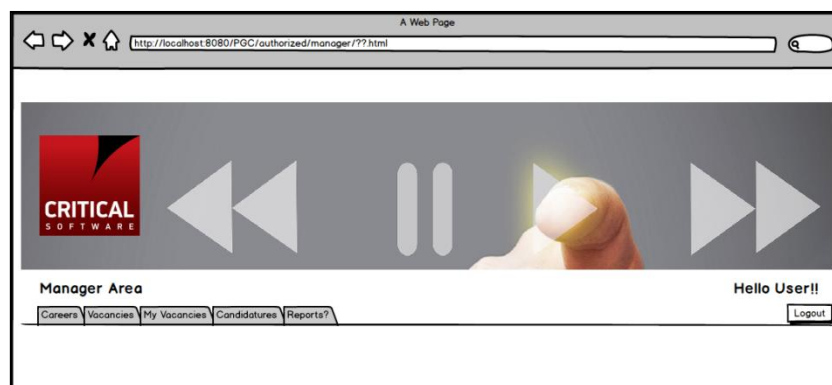


Figura 8 - Aspecto da área reservada para o utilizador Gestor (Manager)

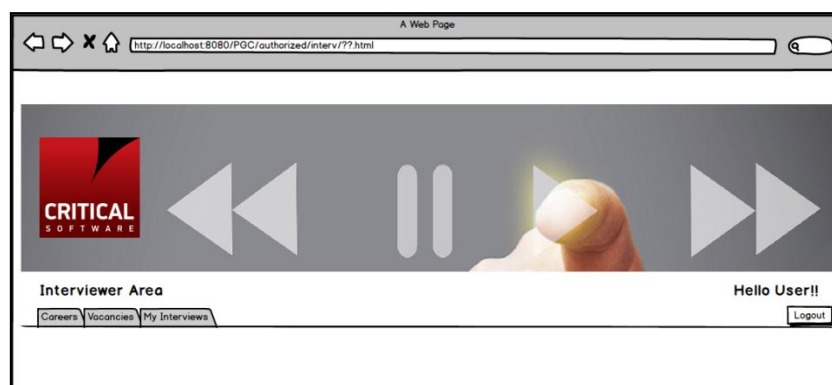


Figura 9 - Aspecto da área reservada para o utilizador Entrevistador(Interviewer)

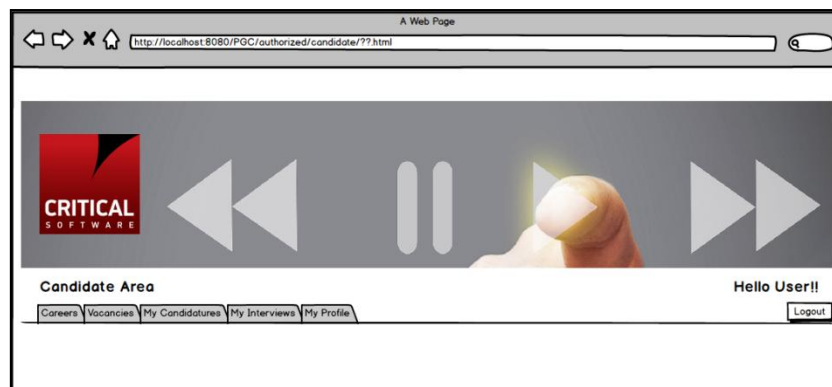


Figura 10 - Aspecto da área reservada para o Candidato (Candidate)

Anexo 1.2

Páginas de login individual

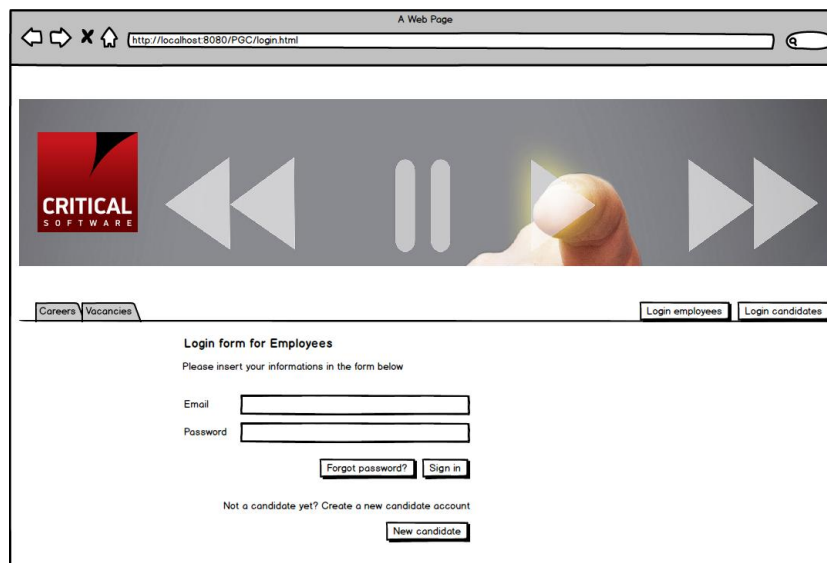


Figura 11 - Página de login de empregados (employees)



Figura 12 - Página de login para os candidatos

Anexo 1.3 Outros protótipos de páginas criados

The screenshot shows a web browser window with the address bar displaying `http://localhost:8080/PGC/authorized/admin/??.html`. The page header features the 'CRITICAL SOFTWARE' logo and navigation arrows. Below the header, the 'Admin Area' is visible, with tabs for 'Careers', 'Vacancies', 'Interviews', 'Guides', 'Candidatures', and 'Reports?'. A 'Hello User!!' message and a 'Logout' button are on the right. The main content area is titled 'Create new Vacancie' and contains the following fields:

- Opening date: / / (with a calendar icon)
- Closing date: / / (with a calendar icon)
- Title:
- Location:
- Status:
- # of openings:
- SLA:
- Technical manager:
- Company:
- Technical area:
- Publication channels: (with a yellow callout bubble labeled 'hselectManyListBox' pointing to it)
- Guide:
- Description:

At the bottom of the form are three buttons: 'Cancel', 'Clear', and 'Create'.

Figura 13 - Aspecto da página de criação de nova posição

The screenshot shows a web browser window with the address bar displaying `http://`. The page title is 'Positions - Critical Software'. The main content area is titled 'Title of the position' and contains the subtitle 'The city of the position'. Below this is a large text area labeled 'Details of the selected position'. At the bottom of the page are three buttons: 'Back', 'Send to a friend', and 'Apply'.

Figura 14 - Apresentação dos detalhes de uma dada apresentação selecionada

Send to a friend

Hi!

I saw this position and I think you will like.
(breve descrição???)

Please see here:
link..

Goodbye,

Your email

Your friend's email

Close
Send



Figura 15 - Pop up de envio de sugestao de posição a amigo

A Web Page
http://localhost:8080/PGC/authorized/admin/??html

Admin Area
Hello User!!
Logout

Careers
Vacancies
Interviews
Guides
Candidatures
Reports?

Search for vacancies
Opening date

Code

Title

Location
☐ city1
☐ city2
☐ city3

Status

Company

Technical area

| Date | Code | Title | Location | Status | Company | Technical area | Technical manager | Details | Edit |
|------|------|-------|----------|--------|---------|----------------|-------------------|---------|------|
| | | | | | | | | | |

Figura 16 - Aspecto da página de posições para cada um dos utilizadores do tipo empregado (Employee)

Admin Area Hello User!! [Logout](#)

[Careers](#) [Vacancies](#) [Interviews](#) [Guides](#) [Candidatures](#) [Reports?](#)

Edit Vacancie

Opening date: Closing date:

Title: Code:

Location:

Status:

of openings:

SLA:

Technical manager:

Company:

Technical area:

Publication channels: (ComboBox)

Guide: (ComboBox)

Description:

Figura 17 - Aspecto da página de edição de posições (não implementado)

Admin Area Hello User!! [Logout](#)

[Careers](#) [Vacancies](#) [Interviews](#) [Guides](#) [Candidatures](#) [Reports?](#)

[Create new guide](#)

| id_guide | Title | Date of creation | Edit | Delete |
|----------|-------|------------------|------|--------|
| | | | | |

Figura 18 - Aspecto da página de apresentação de todos os guioes existentes

Admin Area Hello User!! [Logout](#)

[Careers](#) [Vacancies](#) [Interviews](#) [Guides](#) [Candidatures](#) [Reports?](#)

Search for candidates [Create new candidate](#)

First name: Last name:
 Email: Address:
 City: Country:
 Phone: Mobile phone:
 Main course: School:

☐ Only Spontaneous [Search](#)

search [Search](#)

Result list

| Name (job title) | Age | Nickname | Employee |
|------------------------------------|-----|-----------|-------------------------------------|
| Giacomo Gulizzoni Founder & CEO | 37 | Peldi | <input type="checkbox"/> |
| Marco Botton Tuttofare | 34 | | <input checked="" type="checkbox"/> |
| Mariah Macdonald Better Half | 37 | Patata | <input type="checkbox"/> |
| Valerie Liberty Head Chef | | Val | <input checked="" type="checkbox"/> |
| Guido Jack Gulizzoni | 6 | The Guide | <input type="checkbox"/> |

Figura 19 - Aspecto da página de pesquisa de candidaturas (válido para administrador, gestor e entrevistador) e inserir novas candidaturas na base de dados (administrador e gestor)

Admin Area Hello User!! [Logout](#)

[Careers](#) [Vacancies](#) [Interviews](#) [Guides](#) [Candidatures](#) [Reports?](#)

Tabela?
 Links?
 Criação de novos relatórios?
 O relatório tem os dados dos topicos todos?
 Ou um relatório por topico?

Figura 20 - Aspecto da página de criação de relatórios (não implementada)

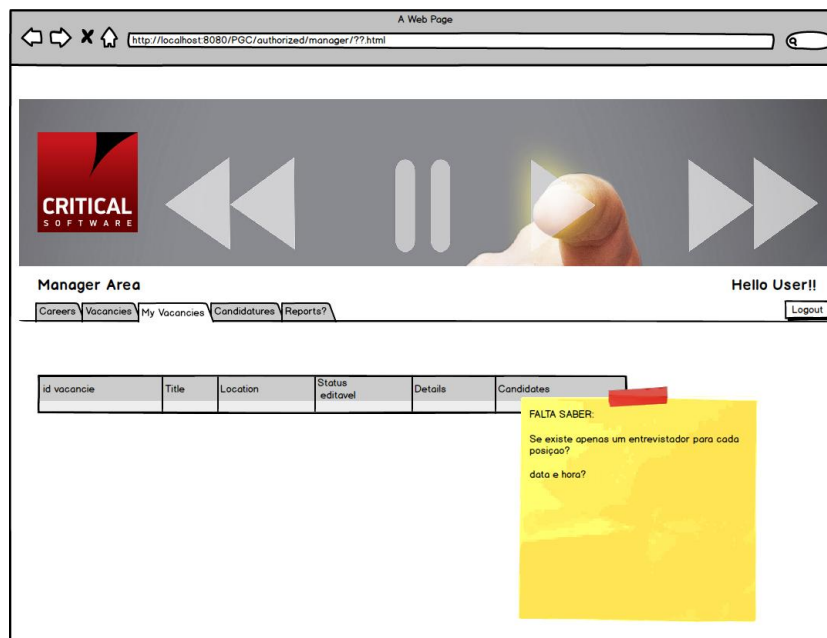


Figura 21 - Aspecto da página das posições de um dado gestor logado

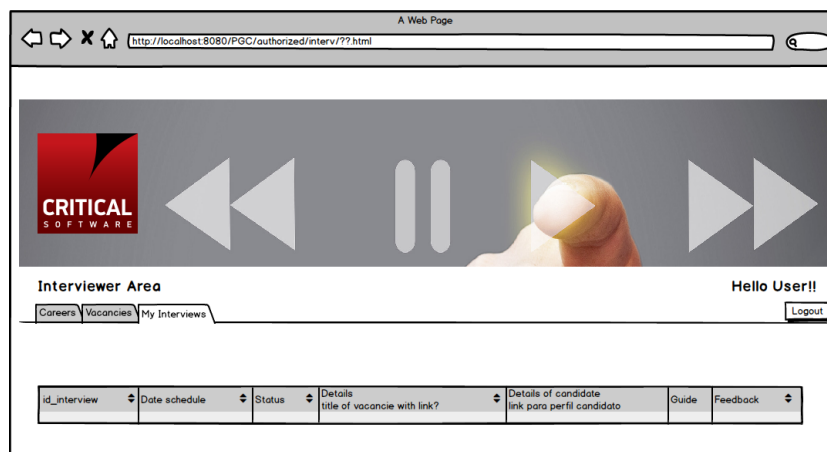


Figura 22 - Aspecto da página das entrevistas de um dado entrevistador

Interviewer

Question 1
The first question

This field permits the insert of the answer of the candidate

Question2
the second question

☐ 1
☐ 2
☐ 3
☒ 4
☐ 5

Question 3
The third question

ComboBox

Status of interview fazer lista de status possiveis para a entrevista

Personal opinion of the interviewer or the justification of status of feedback

Clear Close Print Submit

ao fazer submit do feedback tem de guardar o id da entrevista

status entrevista: agendada, cancelada, aprovada, rejeitado

ficheiro do feedback:

- id entrevista, id entrevistador, id candidato, id posicao,
- data de realizacao da entrevista,
- respostas ao guiao,

talvez por no popup resumo dos dados do candidato

Figura 23 - Aspecto da página de preenchimento de um guião após a entrevista

Candidate Area

Careers Vacancies My Candidatures My Interviews My Profile

| id | Title | Location | Status | Details |
|----|-------|----------|--------|---------|
| | | | | |

Figura 24 - Apresentação das candidaturas de um candidato

Candidate Area

Careers Vacancies My Candidatures My Interviews My Profile

| id | Date | Status | Details title of vacancie with link? |
|----|------|--------|---|
| | | | |

Figura 25 - Apresentação das entrevistas de um candidato

A Web Page
http://localhost:8080/PGC/authorized/candidate??html

CRITICAL SOFTWARE

Candidate Area Hello User!! [Logout](#)

[Careers](#) [Vacancies](#) [My Candidatures](#) [My Interviews](#) [My Profile](#)

First name:
 Last name:
 Email:
 Password:
 Confirm password:
 Birthdate:
 Address:
 City:
 Country:
 Phone:
 Mobile Phone:
 Main Course:
 School:
 Upload CV/Resume: por path do ficheiro previamente submetido

campos com os dados ja preenchidos

Figura 26 - Apresentação dos dados do perfil do candidato, permitindo a sua edição

A Web Page
http://localhost:8080/PGC/newCandidate.html

CRITICAL SOFTWARE

New Candidate Registration

First name:
 Last name:
 Email:
 Password:
 Confirm password:
 Birthdate:
 Address:
 City:
 Country:
 Phone:
 Mobile Phone:
 Main Course:
 School:
 Upload CV/Resume:

Figura 27 - Registo de um novo candidato

A Web Page

http://

CRITICAL
SOFTWARE

Apply of the position
number
title

First name

Last name

Email

Birthdate / /

Address

City

Country

Phone

Mobile Phone

Main Course

School

Upload CV/Resume

Upload Cover Letter

Source

Figura 28 - Aspecto da página de candidatura a uma dada posição

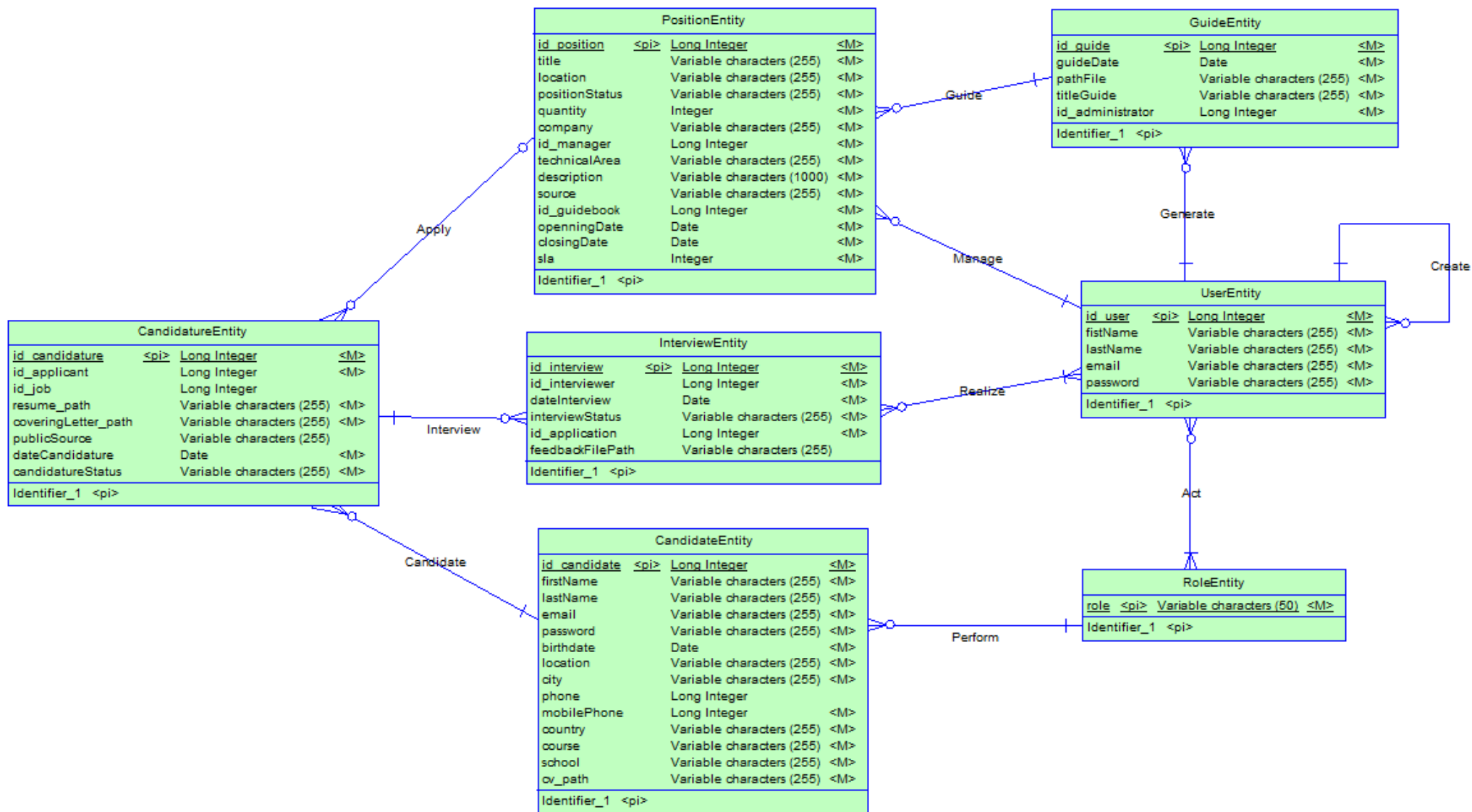


Figura 29 – Diagrama Entidade-Relacionamento (obtido a partir do software PowerDesigner)

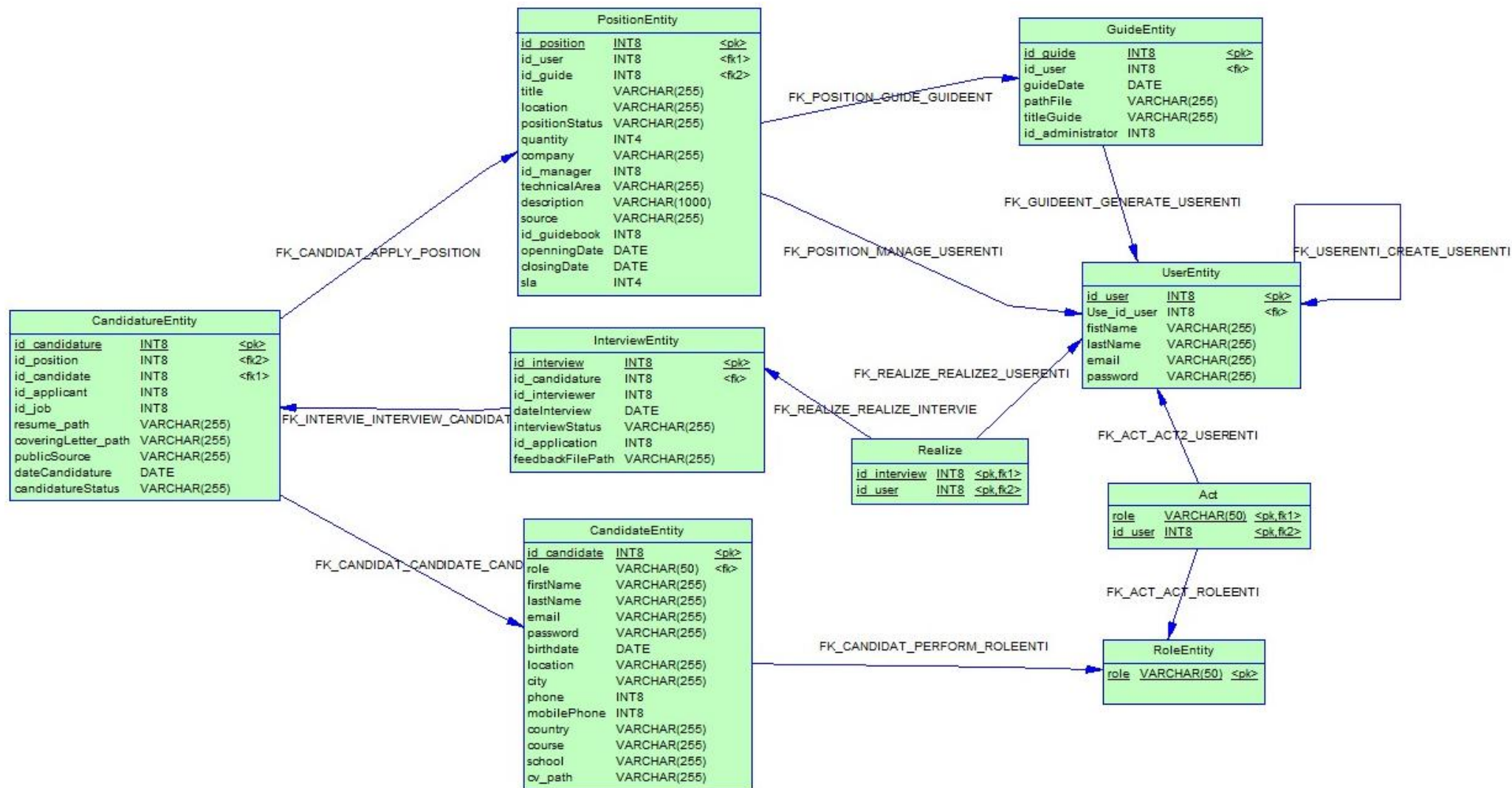


Figura 30 – Diagrama Conceptual (obtido a partir do software PowerDesigner)

Anexo 3 Empacotamento do projecto

Anexo 3.1 Empacotamento da camada JPA

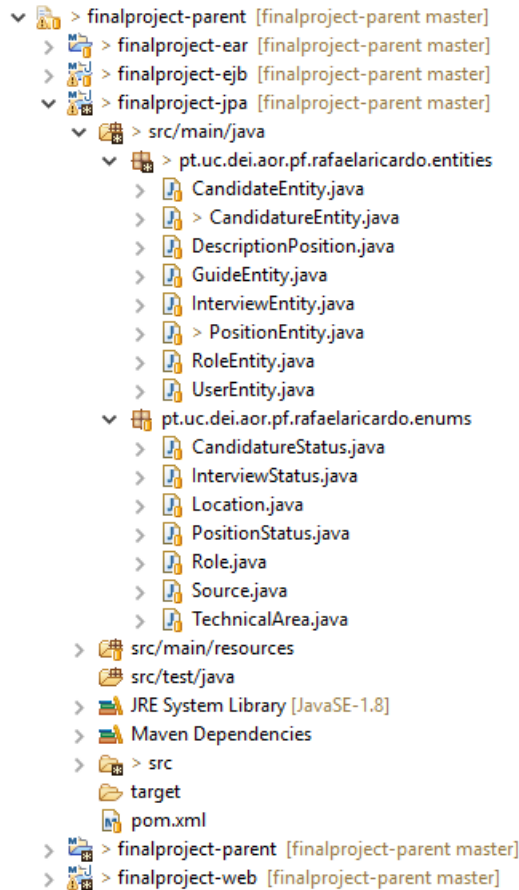


Figura 31 - Empacotamento da camada JPA

Anexo 3.2 Empacotamento camada EJB

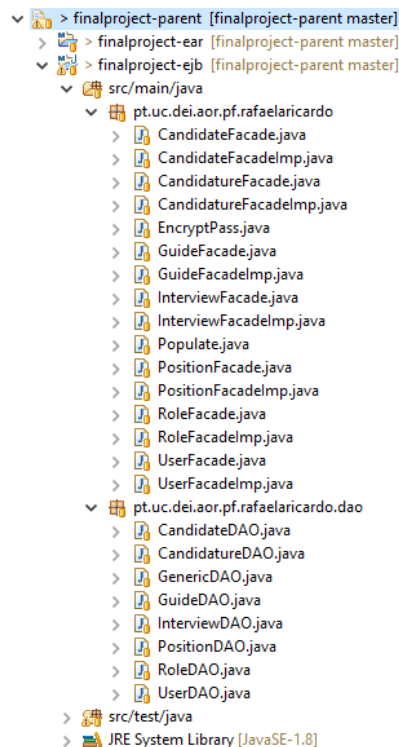


Figura 32 - Empacotamento da camada EJB

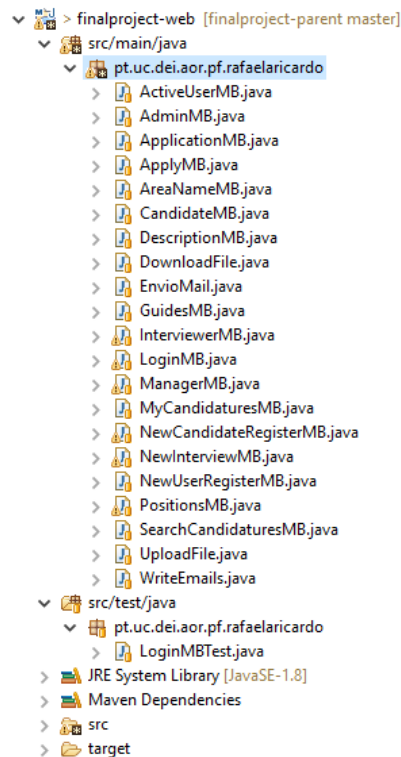


Figura 33 - Empacotamento da camada Web e testes realizados

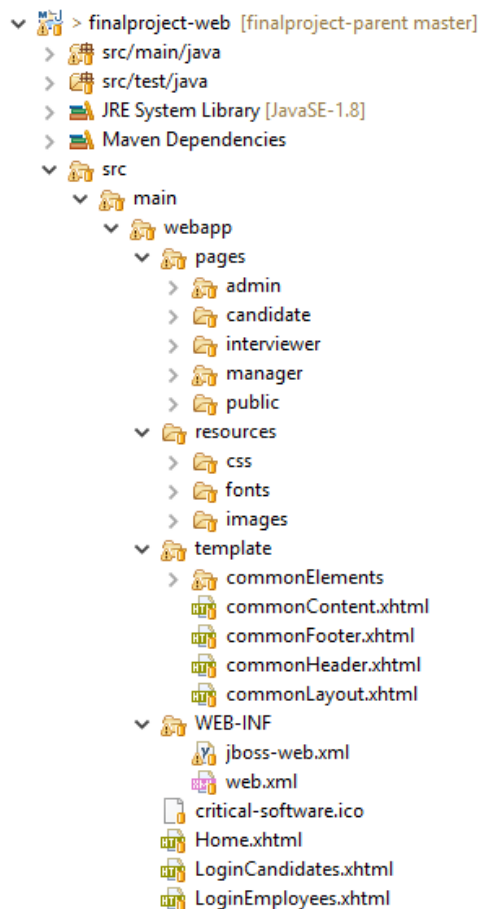


Figura 34 - Estrutura da parte web

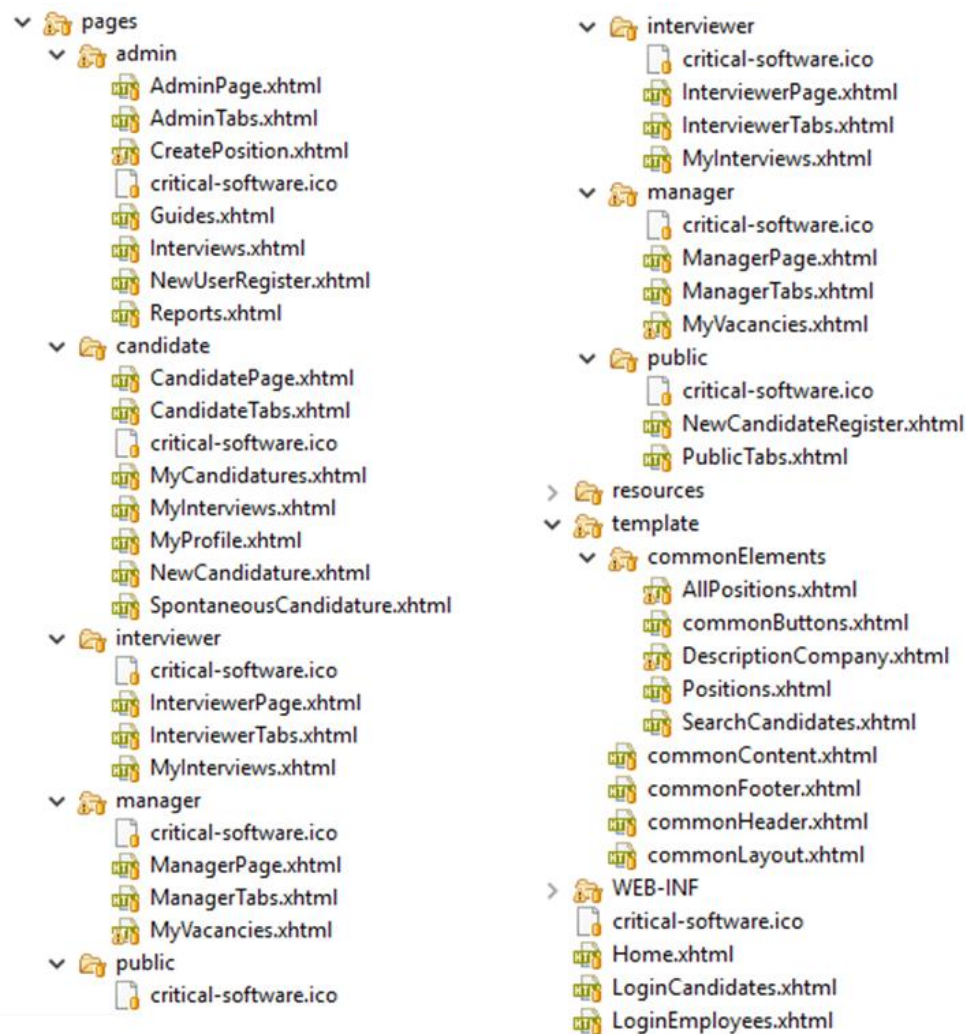


Figura 35 - Visão geral da distribuição da parte web da aplicação

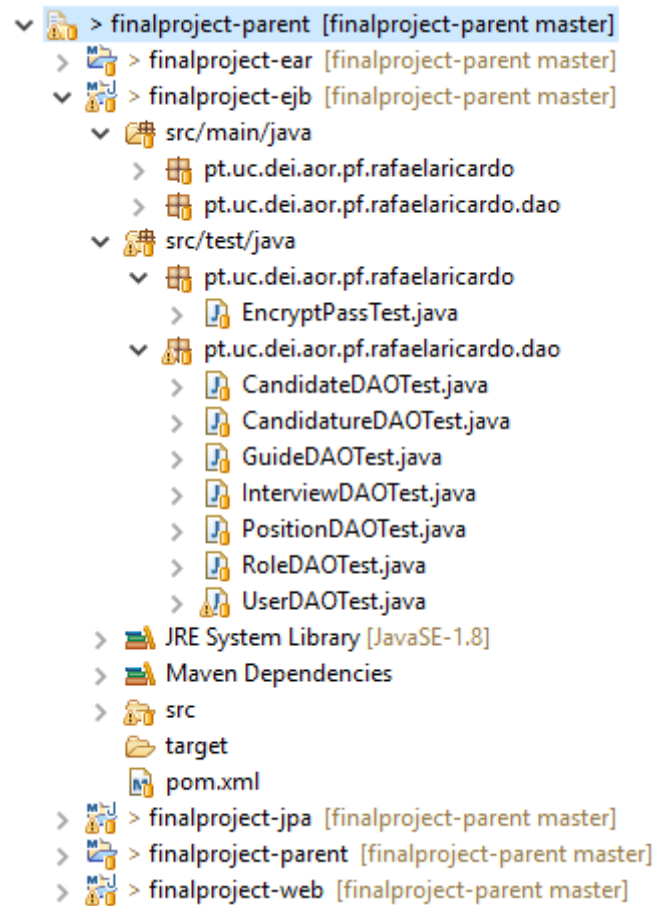


Figura 36 - Testes realizados à camada EJB

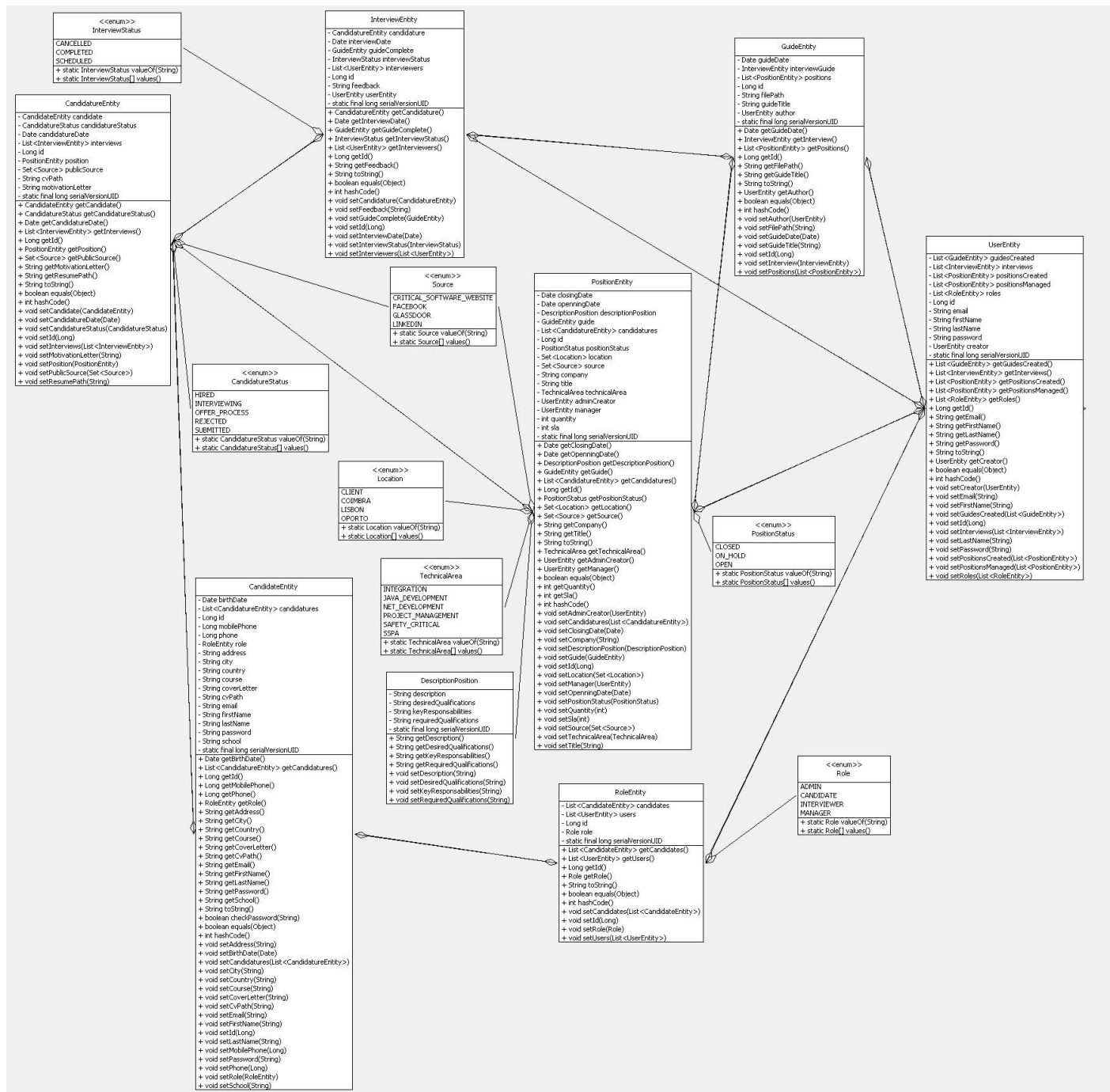


Figura 37 - Diagrama UML final da camada JPA