

UNIVERSIDADE DO MINHO
MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA

Projeto 75 - Gestão do processo de admissão à dissertação

Laboratório em Engenharia Informática

Bruno Teixeira, A84430
Diogo Rio, A84752
Jorge Cerqueira, A85573

21 de junho de 2021

Conteúdo

1	Introdução	3
1.1	Contextualização	3
1.2	Motivação e Objetivos	3
1.3	Tipos de Utilizadores	3
2	Levantamento e Análise de Requisitos	5
2.1	Entidades	5
2.2	Modelos de dados	5
2.2.1	Dissertação	5
2.2.2	Aluno	6
2.2.3	Diretor de Curso/Orientador/Co-orientador	6
2.2.4	Secretaria Direção de Curso	6
2.2.5	Funcionário de Empresa	6
2.2.6	Administrador	6
2.3	Requisitos	7
2.3.1	Aluno	7
2.3.2	Diretor de Curso	7
2.3.3	Secretaria de Direção de Curso	8
2.3.4	Orientador	8
2.3.5	Empresa/Funcionário de Empresa	8
2.3.6	Coorientador	9
2.3.7	Administrador	9
3	Modelação	10
3.1	Modelo de Domínio	10
3.2	Diagrama de Use Cases	11
3.3	Protótipo de Interface	11
4	Considerações Iniciais	14
4.1	Tecnologias Utilizadas	14
5	Implementação	16
5.1	ODOO	16
5.1.1	Models	16
5.1.1.1	Utilizador	16
5.1.1.2	Aluno	16
5.1.1.3	Diretor de Curso/Orientador/Coorientador	16
5.1.1.4	Dissertação	17
5.1.1.5	Plano de Trabalho	18
5.1.2	Views	19
5.1.2.1	Tree Views	19

5.1.3	Form Views	20
5.1.4	Segurança	21
5.1.4.1	Access Control	21
5.1.4.2	Record Rules	21
5.1.5	Wizards	21
5.2	Serviço de Assinaturas Digitais	22
5.2.1	Servidor REST	22
5.2.2	Signature CMD (SCMD)	23
6	Manual de Utilização	25
6.1	Administrador	25
6.2	Aluno	28
6.3	Secretaria Direção de Curso	30
6.4	Orientador/Coorientador/Funcionário de Empresa	31
6.5	Diretor de Curso	32
7	Conclusão	35

Lista de Figuras

3.1	Modelo de Domínio	10
3.2	Diagrama de Use Cases	11
3.3	Página inicial com opção de criação de novo tema de dissertação	12
3.4	Registo de um tema de dissertação	12
3.5	Lista de temas de dissertação disponíveis	12
3.6	Lista de utilizadores da plataforma	12
3.7	Criação de um novo utilizador	13
4.1	Representação da arquitetura <i>Spring</i>	15
5.1	<i>Tree view</i> para gestão de dissertações.	20
5.2	<i>Form view</i> para estudantes de dissertações.	21
5.3	<i>Wizard</i> para criar multiplas dissertações.	22
5.4	Exemplo POST para <code>/requestsign</code>	23
5.5	Exemplo POST para <code>/fetchsign</code>	23
6.1	Menus disponíveis para o Administrador do sistema	25
6.2	Listagem dos alunos na plataforma	25
6.3	Registo de um novo Aluno	25
6.4	Registo de um novo Utilizador na plataforma	26
6.5	Listagem dos Orientadores na plataforma	26
6.6	Registo de um novo Orientador com permissões de Diretor de Curso	26
6.7	Criação de um novo Orientador	26
6.8	Registo de um novo Coorientador	26
6.9	Listagem das Secretarias de Curso na plataforma	27
6.10	Registo de uma nova Secretaria de Curso	27
6.11	Listagem das Empresas registadas na plataforma	27
6.12	Registo de uma nova Empresa	27
6.13	Vista da informação acerca de uma Empresa, incluindo os funcionários da mesma	27
6.14	Listagem dos Funcionários de Empresa	28
6.15	Registo de um novo Funcionário de Empresa	28
6.16	Listagem dos Cursos na plataforma	28
6.17	Registo de um novo Curso	28
6.18	Listagem dos temas disponíveis para o Aluno se candidatar	29
6.19	Informação sobre o tema e possibilidade de Candidatar	29
6.20	Plano de Trabalho associado ao tema que foi atribuído ao Aluno	29
6.21	Opções disponíveis no Plano de Dissertação	29
6.22	Acesso ao Menu Administração Dissertações para entidades relacionadas com o curso	30
6.23	Visualização de todas as dissertações	30
6.24	Listagem de todos os planos de trabalho	30
6.25	Edição de um plano de trabalho para verificação de campos	31

6.26	Listagem dos seus temas de dissertação	31
6.27	Criação de um novo tema de dissertação	31
6.28	Atribuição do tema a um Aluno	32
6.29	Listagem dos plano de trabalho em curso	32
6.30	Visualização das entidades que o Diretor de Curso pode registrar	32
6.31	Visualização do separador Dissertações com permissões iguais a um Orientador	33
6.32	Visualização dos temas de dissertação e o seu estado atual	33
6.33	Visualização das opções para gestão de um tema de dissertação	33
6.34	Adição de uma <i>review</i> a um tema	33
6.35	Publicação de um tema de dissertação	33
6.36	Publicação de vários temas de dissertação	34
6.37	Assinatura de múltiplos documentos em simultâneo - 1	34
6.38	Assinatura de múltiplos documentos em simultâneo - 2	34

Capítulo 1

Introdução

1.1 Contextualização

O desenvolvimento desta plataforma surge no seguimento de diversos problemas e complicações em anos anteriores no que toca à unidade curricular de **dissertação**, nomeadamente em cursos onde o número de alunos é relativamente elevado. Estas complicações surgem devido ao facto de para ser efetuada a admissão à dissertação serem necessárias algumas etapas e cada uma delas com certas burocracias que precisam de ser cumpridas. Por conseguinte, o processo de admissão à dissertação torna-se bastante complexo por parte de quem o tem de gerir, sendo este normalmente o diretor de curso e a secretaria da direção de curso.

1.2 Motivação e Objetivos

Este projeto surge no âmbito da unidade curricular de **Laboratório em Engenharia Informática** onde foram propostos diversos projetos, sendo este o escolhido pelo grupo. Esta plataforma tem como objetivo auxiliar o processo de admissão à dissertação na medida em que irá permitir simplificar e agilizar grande parte do mesmo. Para tal, esta plataforma desenvolverá um certo tipo de interação entre alunos, orientadores, direção de curso e empresas, de modo a automatizar o processo.

Resumidamente, o diretor de curso, orientadores e empresas irão propor e registar os temas de dissertação na plataforma que ficarão visíveis aos alunos de um determinado curso, onde posteriormente estes se "candidatam" a um desses temas e ficam desse modo com um tema de dissertação e com um orientador. Mais à frente neste relatório serão descritos os requisitos de modo a permitir uma melhor interpretação da utilização da plataforma.

1.3 Tipos de Utilizadores

Os utilizadores diferem entre si em função da frequência e da perícia com que usam um produto. Esta distinção permite caracterizar os utilizadores nas seguintes etapas de utilização (Hacks & Redish, 1998):

- **Principiantes:** Todo o utilizador que tem medo de falhar e do desconhecido. Caracterizam-se por serem muito orientados ao objetivo e à tarefa, pelo que estão, normalmente, mais interessados em fazer do que aprender;
- **Principiantes Avançados:** São utilizadores que ultrapassaram o medo de cometer erros e começaram a realizar as tarefas que lhes permitem atingir os seus objetivos. Estes utilizadores são peritos no domínio do problema, mas não na tecnologia. Tentam, assim, realizar as tarefas sem a

ajuda da documentação ou assistentes da interface e, apenas quando não conseguem, recorrem à ajuda de quem já sabe o que fazer;

- **Executantes Competentes:** Utilizadores que aprenderam um número suficiente de tarefas e criaram um modelo mental sólido, quer do domínio do problema, quer do produto, através da experiência. Estes são capazes de prever como é que a interface se irá comportar, e planear como devem realizar uma nova tarefa;
- **Peritos:** Utilizadores altamente motivados, que usam o produto frequentemente como parte integrante do seu trabalho ou das suas atividades pessoais. Têm um conhecimento considerável do domínio do problema e são hábeis na resolução dos seus problemas e dos problemas dos outros. ?

Tendo por base estas classificações, podemos dividir os nossos utilizadores em **principiantes avançados** (Alunos, Coorientadores, Orientadores e Funcionários de Empresa) e em **peritos** (Administrador, Diretor de Curso e Secretaria de Direção de curso).

Capítulo 2

Levantamento e Análise de Requisitos

Com o intuito de identificar os requisitos necessários para a elaboração do projeto, o grupo reuniu-se semanalmente com o docente responsável pela unidade curricular de modo a entender e conhecer todas as etapas do processo de admissão à dissertação para posteriormente passar para a fase de implementação. Desta forma, foi possível identificar os requisitos chave para serem a base de implementação do projeto, permitindo uma redução na probabilidade de insucesso da plataforma.

2.1 Entidades

- Aluno
- Orientador
- Coorientador
- Diretor de Curso
- Secretaria Direção de Curso
- Funcionário de Empresa
- Dissertação
- Administrador

2.2 Modelos de dados

2.2.1 Dissertação

- Título
- Título em inglês
- Descrição
- Autor
- Estado {Reprovado, Aprovação Pendente, Aprovado} e {Não Publicado, Publicado}
- Orientador
- Coorientador (depois de ser atribuído e opcional)

- Aluno (depois de ser atribuído)
- Ano Letivo
- Cursos

2.2.2 Aluno

- Nome
- Email
- Número Mecanográfico
- Curso(s)

2.2.3 Diretor de Curso/Orientador/Co-orientador

- Nome
- Email
- Número Mecanográfico
- Curso(s)
- Departamento(s)
- Centro(s) de investigação

2.2.4 Secretaria Direção de Curso

- Nome
- Email
- Número Mecanográfico
- Departamento
- Curso(s)

2.2.5 Funcionário de Empresa

- Nome
- Email
- Empresa
- Curso(s)

2.2.6 Administrador

- Nome
- Email

2.3 Requisitos

2.3.1 Aluno

1. O Aluno deve conseguir autenticar-se na plataforma.
2. O Aluno deve conseguir ter acesso à lista de temas de dissertação que estão públicos e direcionados aos cursos em que está inscrito.
3. O Aluno deve conseguir visualizar os temas de dissertação que estão públicos e direcionados aos cursos em que está inscrito.
4. O Aluno deve poder candidatar a um tema de dissertação que esteja público e direcionado aos cursos em que está inscrito.
5. O Aluno deve poder visualizar os dados relativos ao tema em que foi aprovado.
6. O Aluno deve conseguir fazer download do seu plano de trabalho semi-preenchido.
8. O Aluno deve conseguir submeter o seu plano de trabalho na plataforma.

2.3.2 Diretor de Curso

1. O DC deve conseguir autenticar-se na plataforma.
2. O DC deve conseguir registrar alunos na plataforma.
3. O DC deve conseguir registrar temas de dissertação na plataforma direcionados aos cursos dos quais é diretor.
4. O DC deve conseguir visualizar os temas de dissertação na plataforma direcionados aos cursos dos quais é diretor.
5. O DC deve conseguir comentar/rever temas de dissertação.
6. O DC deve poder aprovar o estado de temas de dissertação.
7. O DC deve poder tornar públicos temas de dissertação.
8. O DC deve conseguir alterar campos dos temas de dissertação que registou.
9. O DC deve conseguir adicionar um Coorientador dos temas de dissertação que registou.

10. O DC deve conseguir atribuir um tema de dissertação que registou ao um Aluno.

11. O DC deve conseguir assinar digitalmente vários planos de trabalho.

2.3.3 Secretaria de Direção de Curso

1. A Secretaria da DC deve conseguir autenticar-se na plataforma.

2. A Secretaria da DC deve conseguir registar Alunos dos seus cursos na plataforma.

3. A Secretaria da DC deve conseguir verificar todos os planos de trabalho dos cursos e os seus documentos.

4. A Secretaria da DC deve conseguir registar Orientadores e Coorientadores dos seus cursos na plataforma.

5. A Secretaria da DC deve conseguir registar Funcionários de Empresa dos seus cursos na plataforma.

2.3.4 Orientador

1. O Orientador deve conseguir autenticar-se na plataforma.

2. O Orientador deve conseguir registar temas de dissertação direcionados a determinados cursos.

3. O Orientador deve conseguir visualizar e alterar campos dos temas de dissertação que registou.

4. O Orientador deve conseguir adicionar um Orientador e Coorientadores aos temas de dissertação que registou.

5. O Orientador deve poder aprovar um Aluno nos temas de dissertação que registou.

2.3.5 Empresa/Funcionário de Empresa

1. Um funcionário de um Empresa deve conseguir autenticar-se na plataforma.

2. Um funcionário de um Empresa deve conseguir registar temas de dissertação direcionados a determinados cursos.

3. Um funcionário de um Empresa deve conseguir visualizar e alterar campos dos temas de dissertação que registou.

4. Um funcionário de um Empresa deve conseguir adicionar um Orientador e Coorientadores aos temas de dissertação que registou.

5. Um funcionário de um Empresa deve poder aprovar um Aluno nos temas de dissertação que registou.

2.3.6 Coorientador

1. O Coorientador deve conseguir autenticar-se na plataforma.

2. O Coorientador deve conseguir visualizar os temas nos quais é Coorientador.

2.3.7 Administrador

1. O Administrador deve conseguir autenticar-se na plataforma.

2. O Administrador deve conseguir criar e efetuar registos de todas as entidades do sistema.

3.2 Diagrama de Use Cases

Um diagrama de use case representa o levantamento de requisitos de um sistema. Por sua vez, um requisito é uma funcionalidade ou característica considerada relevante na ótica do utilizador e representa o comportamento esperado do sistema. O seu desenvolvimento seguiu um processo iterativo e incremental, na tentativa de implementar as funcionalidades cruciais às necessidades dos seus utilizadores.



Figura 3.2: Diagrama de Use Cases

3.3 Protótipo de Interface

Um protótipo é uma representação parcial do sistema que se pretende desenvolver. A elaboração de um protótipo permite aos utilizadores interagirem com este e explorarem a sua adequação. O seu principal objetivo é reduzir o tempo e o custo necessários para desenvolver algo que possa ser testado pelos utilizadores.

Como será descrito posteriormente, uma das tecnologias utilizadas para o desenvolvimento deste projeto foi a *framework* **ODOO**, desse modo não foram desenvolvidos *mock-ups* para desenvolver um protótipo da plataforma e da sua utilização uma vez que esta *framework* fornece uma interface por defeito que é apropriada para o problema em questão.

Contudo, achou-se por bem efetuar algumas experiências iniciais com a interface fornecida de modo a nos familiarizarmos com a mesma. Posto isto, com os testes iniciais foi desenvolvida uma aproximação ao que poderia ser um protótipo da plataforma, que se pode considerar um protótipo de alta fidelidade, ou seja, uma representação muito próxima da interface final.

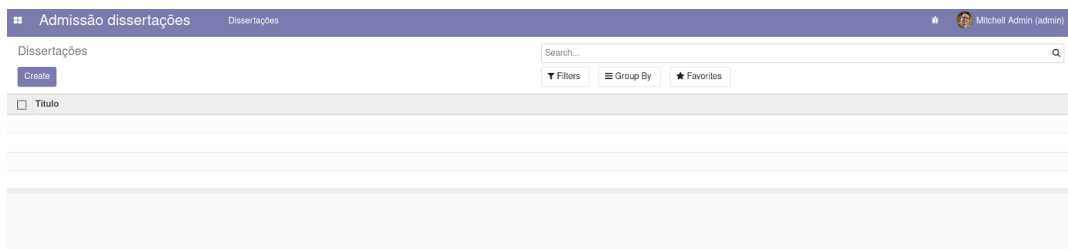


Figura 3.3: Página inicial com opção de criação de novo tema de dissertação

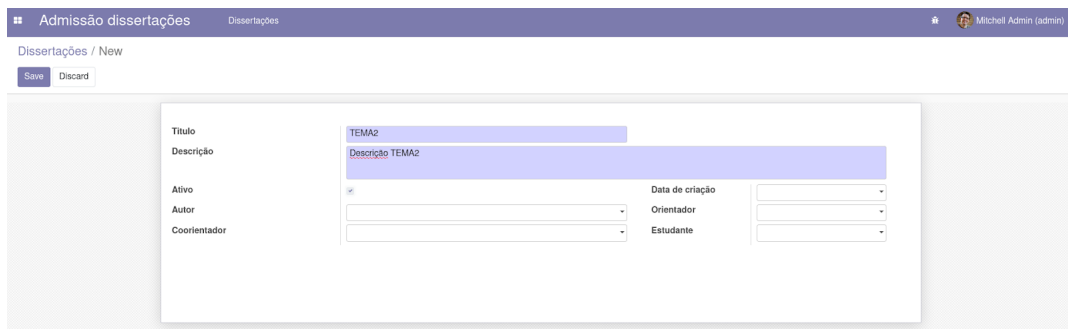


Figura 3.4: Registo de um tema de dissertação

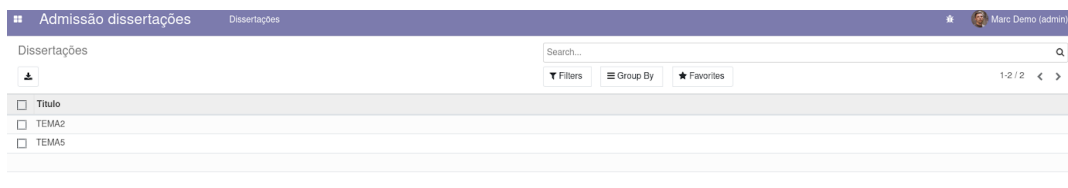


Figura 3.5: Lista de temas de dissertação disponíveis

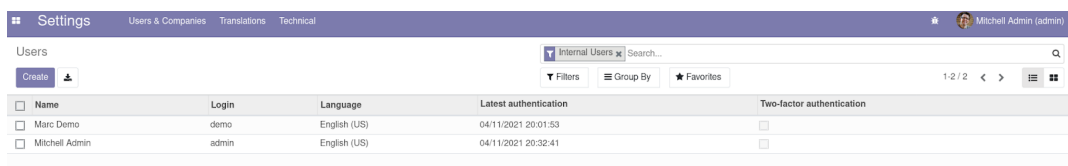


Figura 3.6: Lista de utilizadores da plataforma

Settings Users & Companies Translations Technical Mitchell Admin (admin)

Users / New

Save Discard

Access Rights Preferences

User Type

User types

☒ Internal User
☐ Portal
☐ Public

Administration

Administration

Dissertation Admission

Aluno ☒ Orientador ☐
Secretaria Direção de Curso ☐

Technical

Access to Private Addresses ☐ Access to export feature ☒

Extra Rights

Contact Creation ☒ Multi Companies ☐
Multi Currencies ☐

Figura 3.7: Criação de um novo utilizador

Capítulo 4

Considerações Iniciais

Depois de estudado o domínio em que se encontra o problema e depois de definidos os requisitos do sistema, foi possível iniciar a fase de desenvolvimento da solução. Neste capítulo é apresentada a abordagem e as tecnologias utilizadas.

4.1 Tecnologias Utilizadas

Relativamente a tecnologias utilizadas no desenvolvimento do projeto não foi necessário um estudo acerca das *frameworks* que mais se adaptavam ao tipo de problema uma vez que um dos requisitos por parte do orientador do projeto assentava em usar a *framework* **Odoo**. Este requisito surge na futura possibilidade que fazer uma junção entre vários projetos desenvolvidos no mesmo âmbito e com objetivos paralelos.

O *Odoo* é um software que oferece um conjunto alargado de aplicações ou módulos que permitem a gestão das regras de negócio de uma empresa e está disponível em duas versões: a *Odoo Community*, uma versão *open source*, e a *Odoo Enterprise*, uma versão licenciada que complementa a versão *open source* acrescentando-lhe algumas funcionalidades.

Este sistema foi lançado pela primeira vez em fevereiro de 2005, com o nome de *Tiny ERP* e da qual foram lançadas 4 versões. Em setembro de 2014 foi lançada a oitava versão do sistema, com o novo e atual nome - **Odoo**. À data de escrita deste relatório, a versão mais atual do **Odoo** é a versão **14.0** lançada em outubro de 2020.

O *Odoo* implementa o modelo Cliente-Servidor, com uma base de dados de suporte relacional, em *PostgreSQL*. Os modelos de dados e lógica de negócio são desenvolvidos em *Python*. Quanto à camada de apresentação, estas são elaboradas com recurso a *XML*.

Para este projeto, dado o reconhecimento e toda a comunidade existente e também o uso da linguagem *Python*, decidiu-se então utilizar a versão *open source* do **Odoo 14.0**. A abordagem efetuada e todas as decisões de implementação são apresentadas na próxima secção.



Para a implementação do requisito 11 relativo ao Diretor de Curso, e devido à falta de ferramentas necessárias para criar uma assinatura digital num PDF em *python*, foi necessário desenvolver um serviço **REST** em **Java** de modo a utilizar a biblioteca **Digital Signature Services (DSS)** em conjunto com o serviço **Signature CMD (SCMD)** para devidamente criar as assinaturas digitais. Para criar o serviço **REST**, que será devidamente explorado na próxima secção, recorreu-se à *framework* **Spring Boot**. O

Spring Boot foi uma escolha óbvia, dada a simplicidade do serviço a implementar, assim como a vantagem que esta *framework* acarreta que passa pela simplicidade e diminuição do tempo de desenvolvimento de um serviço/aplicação.

O *Spring* é uma *framework* JAVA criada para facilitar o desenvolvimento de aplicações nesta linguagem. Esta *framework* explora os conceitos de IoC e Injeção de Dependências.

A utilização desta *framework* traz diversas vantagens, tais como:

- Facilita a programação segundo boas práticas, entre as quais, o uso de interfaces em vez de classes;
- É dividida em módulos, permitindo o uso apenas das partes necessárias;
- Fornece um *container* leve que pode ser ativado sem ser necessário recorrer a *web* e *application servers software*;
- Suporta a *framework* JDBC para interação com Bases de Dados;
- O módulo *Spring Test* fornece suporte para testes de código.

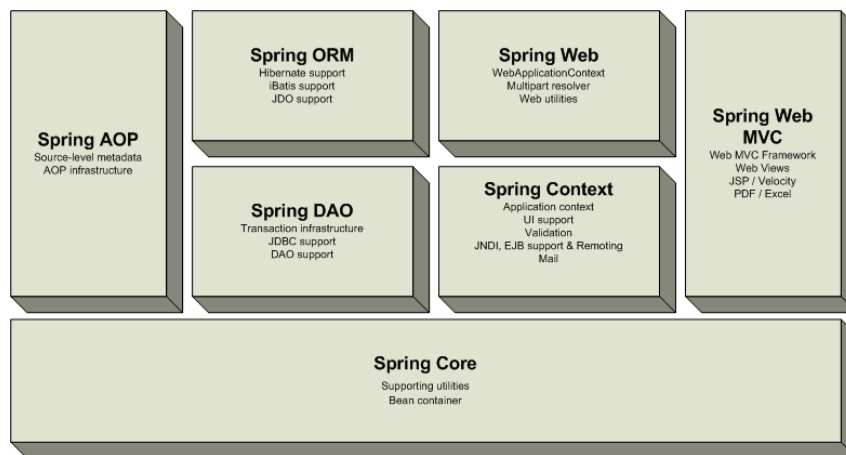


Figura 4.1: Representação da arquitetura *Spring*

Capítulo 5

Implementação

5.1 ODOO

Nesta secção são discutidas todas as decisões de implementação tomadas, nomeadamente alguns dos *Models* (classes) criados, as *Views* definidas e também as regras de segurança.

5.1.1 Models

5.1.1.1 Utilizador

Como já anteriormente referido, existem 6 utilizadores distintos na plataforma, a saber : Aluno, Diretor de Curso, Orientador, Coorientador, Funcionário de Empresa e Secretaria Direção de Curso. O que todos estes utilizadores têm em comum é o facto de serem uma extensão da classe *default* do *Odoo*, ou seja, todos eles possuem campos obrigatórios, nomeadamente os de *login*, e foram sendo acrescentados novos conforme necessário a cada tipo de utilizador.

5.1.1.2 Aluno

Para a representação de um Aluno foi criada a classe *Student* com as seguintes variáveis de instâncias.

```
class Student(models.Model):
    _name = 'dissertation_admission.student'
    _inherits = {'res.users': 'user_id'}
    _description = 'Estudante'
    user_id = fields.Many2one('res.users', ondelete='restrict')
    university_id = fields.Char(required=True)
    course = fields.Many2one('dissertation_admission.course', required=True)
```

Através do excerto de código acima apresentado podemos retirar que cada Aluno, além dos campos da classe *default* do *Odoo* - *res.users*, possui um *user_id*, que será o identificador do aluno na plataforma, um *university_id*, que representa o número mecanográfico do aluno, e ainda uma lista de cursos aos quais o aluno está associado - *course*.

5.1.1.3 Diretor de Curso/Orientador/Coorientador

Para a representação de um Diretor de Curso, um Orientador e um Coorientador foi criada a classe *Adviser*. Foi tomada a decisão de englobar estas 3 entidades do sistema numa só classe uma vez que as variáveis de instância que cada uma possui são relativamente parecidas. Desse modo, foram colocadas no mesmo modelo, contudo irão ter permissões distintas no sistema.

```

class Adviser(models.Model):
    perms = [
        ('coadviser', 'Coorientador'),
        ('adviser', 'Orientador'),
        ('director', 'Diretor de Curso')
    ]
    _name = 'dissertation_admission.adviser'
    _inherits = {'res.users': 'user_id'}
    _description = 'Orientador'
    user_id = fields.Many2one('res.users', ondelete='restrict', required=True)
    university_id = fields.Char(required=True)
    department = fields.Char()
    courses = fields.Many2many('dissertation_admission.course', required=True,
                               relation="dissertation_admission_adviser_course_rel")
    investigation_center = fields.Char()
    perms = fields.Selection(perms, required=True, default='pending')

```

Pelo excerto de código anteriormente apresentado podemos observar que cada uma das 3 entidades incluídas neste modelo, além dos campos da classe *default* do *Odoo* - *res.users*, possui um *user_id*, que será o identificador do aluno na plataforma, um *university_id*, que representa o número mecanográfico do aluno. Além disso, possui também as variáveis *department* e *investigation_center* que identificam respetivamente o departamento e o centro de investigação ao qual o Diretor de Curso/Orientador/Co-orientador pertence. Tal como no modelo anterior, existe também uma lista de cursos - *course* - que representa os cursos aos quais estas entidades podem estar associadas. Por último, temos o campo *perms*, que irá identificar quais as permissões que cada uma das entidades irá possuir.

5.1.1.4 Dissertação

Sendo a Dissertação o tema central desta plataforma era de prever que esta fosse a entidade com mais complexidade e ligações para implementar. Desse modo, nesta secção, serão abordados apenas os campos tidos como mais importantes e que possam causar alguma duvida. Para a representação de uma Dissertação foi criada a classe *Dissertation* com as seguintes variáveis de instâncias.

```

class Dissertation(models.Model):
    states = [
        ('disapproved', 'Reprovado'),
        ('pending', 'Aprovação Pendente'),
        ('approved', 'Aprovado')
    ]
    _name = 'dissertation_admission.dissertation'
    _description = 'Dissertação'

    name = fields.Char('Titulo Português', required=True)
    name_en = fields.Char('Titulo Inglês', required=True)
    description = fields.Text(string='Descrição', required=True)
    state = fields.Selection(states, string='Estado', required=True, default='pending')
    school_year = fields.Selection([(str(num) + '/' + str(num + 1), str(num) + '/'
                                     + str(num + 1))
                                   for num in range(2020, datetime.datetime.now().year)]
                                , required=True)
    is_public = fields.Boolean(string='Público?', required=True, default=False)
    course = fields.Many2many('dissertation_admission.course',
                              relation='dissertation_admission_dissertation_course_rel')

```

```

adviser_id = fields.Many2one('dissertation_admission.adviser', string='Orientador')
coadviser_id_internal = fields.Many2one('dissertation_admission.adviser',
                                         string='Coorientador')
coadviser_id_external = fields.Many2one('dissertation_admission.company_employee',
                                         string='Coorientador')
student_id = fields.Many2one('dissertation_admission.student', string='Estudante')
candidates = fields.Many2many('dissertation_admission.student', readonly=True
                              , relation='dissertation_admission_dissertation_candidates_rel')
reviews = fields.Many2many('dissertation_admission.dissertation_review'
                           , compute='_get_reviews')

work_plan_id = fields.Many2one('dissertation_admission.work_plan',
                               string='Plano de Trabalho', compute="_get_work_plan")

create_uid_internal = fields.Many2one('dissertation_admission.adviser',
                                       compute='_get_create_uid_internal')
create_uid_external = fields.Many2one('dissertation_admission.company_employee',
                                       compute='_get_create_uid_external')

```

Cada entidade Dissertação possui como campos básicos um estado, um título em português, um título em inglês, um ano letivo e uma descrição, que são respetivamente identificados acima por `state`, `name`, `name_en`, `school_year` e `description`. Depois de registada na plataforma, são atribuídos campos à Dissertação que permitem controlar o *workflow* da mesma conforme são executadas tarefas sobre a mesma. Desses campos podemos destacar o `is_public`, que distingue se um certo tema de dissertação está ou não público na plataforma de modo a ser observado pelos candidatos; o `adviser_id` que permite aos candidatos ter conhecimento sobre a quem pertence o tema de dissertação; o `student_id` que só será preenchido após o tema ser atribuído a algum aluno que seja candidato ao tema, que estarão numa lista `candidates`; podemos destacar ainda a variável `reviews` que é uma lista de comentários feitos pelo diretor de curso até que o tema esteja em condições de ser aprovado, e posteriormente público. Por último, temos a referência a um plano de trabalho - `work_plan_id` - que será abordado de seguida.

5.1.1.5 Plano de Trabalho

Outro dos modelos do sistema é o plano de trabalho que consiste num modelo que contém 2 ficheiros PDF que contêm grande parte das informações da dissertação e fazem correspondência a um tema de dissertação. Para este modelo - `WorkPlan` foram criadas as seguintes variáveis de instância.

```

class WorkPlan(models.Model):
    _name = 'dissertation_admission.work_plan'
    _description = 'Plano de Trabalho'

    dissertation = fields.Many2one('dissertation_admission.dissertation',
                                   readonly=True, required=True)
    student = fields.Many2one('dissertation_admission.student', readonly=True,
                              required=True)

    pdf = fields.Binary(readonly=True, default=None)
    pdf_fname = fields.Char(compute="_get_pdf_fname")
    pdf_signed = fields.Binary(default=None)
    pdf_signed_fname = fields.Char(compute="_get_pdf_signed_fname")
    verified = fields.Boolean(default=False)
    signed_director = fields.Boolean(default=False)
    pdf_pre_thesis = fields.Binary(default=None)

```

```
pdf_pre_thesis_fname = fields.Char(compute="_get_pdf_pre_thesis_fname")
```

Como principais campos temos o `dissertation` e `student`, que fazem correspondência ao tema de dissertação escolhido e ao aluno ao qual o tema foi atribuído. Além disso, temos 2 ficheiros pdf - `pdf` e `pdf_pre_thesis` - que são ficheiros necessários para todo o processo de admissão á dissertação e que podem ser obtidos na plataforma. Após o preenchimento desses pdfs é esperado que o aluno faça a submissão dos mesmos na plataforma e posteriormente sejam assinados por os vários intervenientes nessa dissertação. Para tornar o processo de assinatura mais ágil existem campos que permitem filtrar quais os planos de trabalho estão efetivamente bem preenchidos e assinados, entre os quais o `verified` e `signed_director`.

Além dos modelos anteriormente abordados existem ainda um conjunto deles que não foram abordados uma vez que são considerados modelos secundários ou então são relativamente parecidos a um dos anteriores. Posto isto, foi tomada a decisão de não fazer a descrição dos mesmo para uma leitura mais leve e perceptível deste documento.

5.1.2 Views

O *odoo* oferece uma maneira de alterar a maneira padrão que os dados são apresentados, estas alterações são designadas como *views*, permitem escolher que campos devem ser mostrados e a que grupos, bem como marcar campos como *readonly*, *required*, entre outras funcionalidades.

5.1.2.1 Tree Views

Este tipo de *view* é utilizado para quando é necessário apresentar uma lista de *records* de um modelo. Estas listas são normalmente mostradas em forma de tabela com cada linha a representar um *record* e cada coluna a representar um certo *field* (ou em certos casos, coisas como por exemplo botões).

Em baixo pode ver-se a *tree views* definida para dissertações, mais especificamente, a apresentada para o menu de gerir dissertações:

```
<record id="view_dissertation_admission_dissertation_director_tree" model="ir.ui.view">
  <field name="name">dissertation_view_tree</field>
  <field name="model">dissertation_admission.dissertation</field>
  <field name="arch" type="xml">
    <tree create="false">
      <field name="name" string="Titulo"/>
      <field name="create_uid" string="Autor"/>
      <field name="adviser_id" string="Orientador"/>
      <field name="state" string="Estado"/>
      <field name="is_public" string="Publico?"/>
      <field name="school_year" string="Ano Letivo"/>
      <button string="Publicar" name="publish" type="object"
        class="oe_highlight oe_read_only"
        attrs="{ 'invisible': ['|', ('is_public', '=', True),
                                ('state', '!=', 'approved')]}"/>
    </tree>
  </field>
</record>
```

Como podemos ver esta *tree view* deixa ao utilizador ver diversa informação na lista, para além disso está definido um botão para publicar uma dissertação que está invisível quando esta já está publica

ou não está aprovada.

<input type="checkbox"/>	Título	Autor	Orientador	Estado	Publico?	Ano Letivo	
<input type="checkbox"/>	TITULO	Mitchell Admin	Mitchell Admin	Aprovado	<input type="checkbox"/>	2020/2021	Publicar

Figura 5.1: *Tree view* para gestão de dissertações.

5.1.3 Form Views

Este tipo de *view* é especializado para conseguir visualizar e/ou modificar apenas um *record* de um modelo, tal como no anterior é possível definir como os *fields* serão apresentados bem como adicionar coisas como botões para chamarem ações.

Temos por exemplo a *form view* apresentada a alunos que querem candidatar-se a uma dissertação.

```
<record id="view_dissertation_admission_dissertation_simple_form" model="ir.ui.view">
  <field name="name">Dissertation Simple Form</field>
  <field name="model">dissertation_admission.dissertation</field>
  <field name="arch" type="xml">
    <form create="false" edit="false" delete="false" duplicate="false">
      <header>
        <button string="Candidatar" name="register_candidate" type="object"
          class="oe_highlight oe_read_only"
          confirm="Tem a certeza que quer candidatar-se a esta dissertação?"
          groups="dissertation_admission_app.dissertation_admission_group_student"/>
      </header>
      <sheet string="Dissertação">
        <group>
          <field name="name" required="True" id="name"/>
          <field name="name_en" required="True" id="name"/>
          <field name="description" required="True" colspan="4" id="description"/>
          <field name="course" id="course" required="True" widget="many2many_tags"/>
          <field string="Autor" name="create_uid_internal"
            attrs="{ 'invisible': [('create_uid_internal', '=', False)] }"/>
          <field string="Autor" name="create_uid_external"
            attrs="{ 'invisible': [('create_uid_external', '=', False)] }"/>
          <field name="adviser_id" id="adviser_id"/>
          <field name="coadviser_id_internal"
            attrs="{ 'invisible': [('coadviser_id_internal', '=', False)] }"/>
          <field name="coadviser_id_external"
            attrs="{ 'invisible': [('coadviser_id_external', '=', False)] }"/>
          <field name="candidates" id="candidates" widget="many2many_tags" readonly="True"/>
        </group>
      </sheet>
    </form>
  </field>
</record>
```

Podemos ver abaixo o resultado desta *view*.

Figura 5.2: *Form view* para estudantes de dissertações.

5.1.4 Segurança

A *framework odoo* oferece dois mecanismos para tratar da segurança, sendo eles *access control* e *record rules*. Ambos os mecanismos permitem escrever regras que estão associadas a certos grupos e modelos.

5.1.4.1 Access Control

Neste mecanismo é onde estão definidas as regras mais gerais, sendo que se pode definir num nível mais alto se um grupo vai ou não poder ver, editar, criar ou eliminar *records* de um certo modelo.

```
id,name,model_id:id,group_id:id,perm_read,perm_write,perm_create,perm_unlink
access_dissertation_user,User Access,model_dissertation_admission_dissertation,\
dissertation_admission_group_user,1,0,0,0
```

Como podemos ver, a linha acima indica que todos os *users* terão pelo menos algum nível de acesso para ler dissertações (todos os grupos da aplicação de admissão de dissertações herdam o grupo `access_dissertation_user`).

5.1.4.2 Record Rules

Este é o mecanismo que permite refinar as regras definidas em *access control*.

```
<record model="ir.rule" id="student_dissertation_rule">
  <field name="name">Alunos só podem interagir com dissertações publicas no seu curso</field>
  <field name="model_id" ref="model_dissertation_admission_dissertation"/>
  <field name="domain_force">
    ['&','course', 'in', user.student_uid.course.id]
    , ('is_public', '=', True)]
  </field>
  <field name="groups" eval="[(4, ref('dissertation_admission_group_student'))]"/>
</record>
```

Como podemos ver a regra acima aplica-se ao grupo de estudantes, sendo que restringe a que apenas consigam ver dissertações visadas ao seu curso e que para além disso estejam publicas.

5.1.5 Wizards

Wizards são um tipo de modelo efêmero que permite ao utilizador criar temporariamente um *record* de certo tipo, permitindo assim fazer certas computações com ele.

Existe por exemplo um *wizard* que permite ao utilizador escolher diversas dissertações e publica-las todas ao mesmo tempo.


```
class PublishDissertationWizard(models.TransientModel):
    _name = 'dissertation_admission.publish_dissertation_wizard'
    dissertations = fields.Many2many('dissertation_admission.dissertation'
                                     , relation='dissertation_admission_dissertation_publish_wizard_rel')

    def confirm(self):
        for dissertation in self.dissertations:
            dissertation.publish()
```

Titulo	Titulo Inglês	Autor	Orientador	Ano Letivo
TITULO	TITLE	Mitchell Admin	Mitchell Admin	2020/2021

Add a line

Confirmar Cancelar

Figura 5.3: *Wizard* para criar multiplas dissertações.

5.2 Serviço de Assinaturas Digitais

Uma assinatura digital é um esquema criptográfico que utiliza utiliza primitivas de criptografia assimétrica para assegurar a autenticidade e a integridade de um documento digital, além disto é garantido o não-repúdio das entidades que assinam o documento. De uma forma superficial, uma assinatura digital é criada assinando uma *hash* do documento digital com uma chave privada, esta assinatura é concatenada ao documento digital para poder ser posteriormente validada. De modo a validar uma assinatura digital é decifrada a assinatura presente num documento com uma chave pública e é calculada a *hash* do documento, a assinatura é válida se a assinatura decifrada corresponde à *hash* do documento.

Segundo a legislação portuguesa em vigor as assinaturas digitais são consideradas equivalentes a assinaturas tradicionais, desde que sejam devidamente certificadas por entidades competentes.

Existem várias formas de assinar documentos digitais, o nosso serviço oferece uma assinatura que segue a norma **PAdES** (PDF Advanced Electronic Signatures), existem nesta norma quatro níveis de assinatura base, por ser suficiente aos requisitos iremos utilizar o nível PAdES-B-B neste serviço. Além disto a assinatura criada será uma assinatura digital qualificada, que obedece à regulamentação da UE 910/2014 ([eIDAS](#)).

O nosso serviço é então composto pelo servidor REST que fornece uma API onde é possível assinar um ou vários ficheiros em formato PDF.

5.2.1 Servidor REST

Como já abordado anteriormente, foi utilizada a *framework* **Spring Boot** para construir de forma simples e rápida um serviço capaz de receber pedidos de documentos para assinar e entregar os mesmos devidamente assinados.

O serviço possui uma **API** com duas operações de modo a assinar um ou vários PDFs. A assinatura dos documentos é feita em duas partes, numa primeira é recebido no *end-point* `/requestsign` um POST com conteúdo do tipo *form-data* onde um dos campos é um *JSON* com os seguintes campos: `userid` (número de telemóvel), `pin` e coordenadas `x` e `y` onde deverá ser criada a assinatura visual, nos restantes

campos deverão estar presentes um ou mais PDFs que irão ser assinados. Quanto o serviço recebe um *POST* deste tipo é efectuado um pedido ao serviço *SCMD* para assinar os PDFs recebidos, o serviço *SCMD* irá retornar um identificador da operação e irá enviar um código de uso único ao assinante, feito isto o serviço REST irá devolver o mesmo identificador da operação ao cliente. Um exemplo do POST referido pode ser visto na imagem abaixo.

```

1  POST /requestsign HTTP/1.1
2  Host: localhost:8080
3  Content-Length: 515
4  Content-Type: multipart/form-data; boundary=----WebKitFormBoundary7MA4YWxkTrZu0gW
5
6  ----WebKitFormBoundary7MA4YWxkTrZu0gW
7  Content-Disposition: form-data; name="params"
8  Content-Type: application/json
9
10 {
11   "userid" : "+351931444118",
12   "pin" : "12341234",
13   "x" : 250,
14   "y" : 250
15 }
16 ----WebKitFormBoundary7MA4YWxkTrZu0gW
17 Content-Disposition: form-data; name="file"; filename="2.pdf"
18 Content-Type: application/pdf
19
20 (data)
21 ----WebKitFormBoundary7MA4YWxkTrZu0gW
22 Content-Disposition: form-data; name="file"; filename="1.pdf"
23 Content-Type: application/pdf
24
25 (data)
26 ----WebKitFormBoundary7MA4YWxkTrZu0gW
27

```

Figura 5.4: Exemplo POST para /requestsign

Numa segunda fase, o cliente tem de enviar um POST para o *end-point* "/fetchsign" onde o *body* será um *JSON* com o identificador da operação anteriormente recebido e o código de utilização única que recebeu e o serviço irá responder com um *ZIP* dos documentos assinados. Um exemplo do POST referido pode ser visto na imagem seguinte.

```

1  POST /fetchsign HTTP/1.1
2  Host: localhost:8080
3  Content-Type: application/json
4  Content-Length: 80
5
6  {
7    "operationID": "b1315f90-aae3-47a1-a21a-e0931b6ee849",
8    "otp": "811856"
9  }

```

Figura 5.5: Exemplo POST para /fetchsign

De modo a assinar devidamente um documento PDF de acordo com todas as normas actualmente existentes, fizemos uso de alguns módulos da biblioteca de código livre [DSS](#).

5.2.2 Signature CMD (SCMD)

Como explicado anteriormente, são necessárias chaves públicas e privadas para assinar digitalmente um documento. Para a assinatura ser legalmente aceite é necessário que estas chaves sejam geradas e certificadas por uma entidade de certificação (*CA*) legalmente reconhecida, como por exemplo a Agência para a Modernização Administrativa (**AMA**). A chave privada é secreta e como tal, para realizar a

assinatura de um documento, é necessário recorrer ao serviço **SCMD** que faz a assinatura da *hash* de um documento utilizando a chave privada associada a um cidadão que se autentica utilizando um número de telemóvel, um pin e um código de uso único. Além disso, o serviço fornece a chave pública juntamente com os certificados necessários para a certificar. O serviço **SCMD** é um serviço SOAP onde é disponibilizado um ficheiro **WSDL**, utilizando [JAX-WS R](#) é gerado o código de cliente de modo a poder realizar as operações de assinatura.

Capítulo 6

Manual de Utilização

Nesta secção será apresentado um breve manual de utilização da plataforma tendo em conta o papel desempenhado pelas diferentes entidades do sistema.

6.1 Administrador

Como referido anteriormente, o Administrador do sistema tem como funções criar e registar todas as entidades envolvidas na plataforma exceto as Dissertações. De seguida são apresentadas as várias páginas disponíveis para o Administrador, assim como o exemplo do registo das entidades que pode criar.

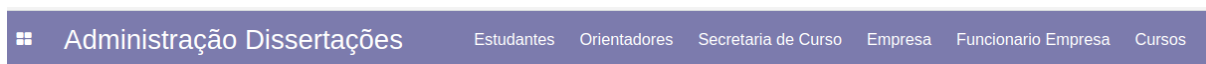


Figura 6.1: Menus disponíveis para o Administrador do sistema

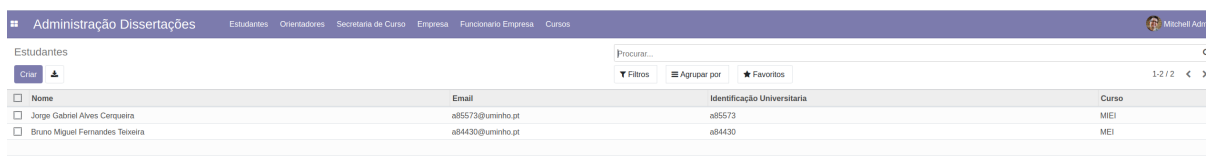


Figura 6.2: Listagem dos alunos na plataforma

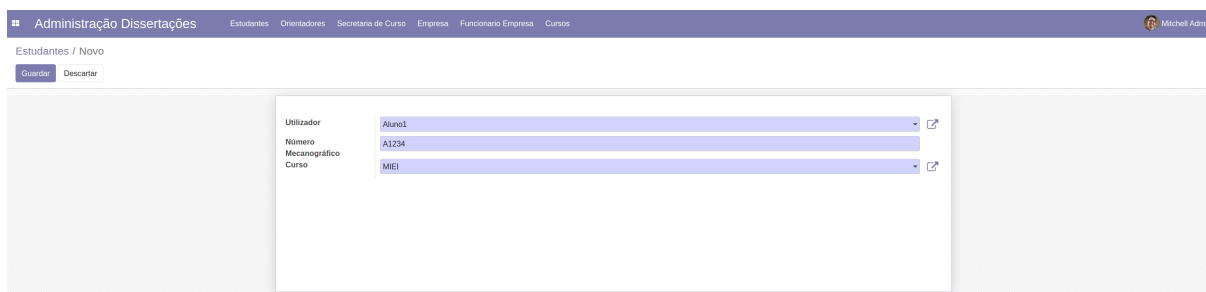


Figura 6.3: Registo de um novo Aluno

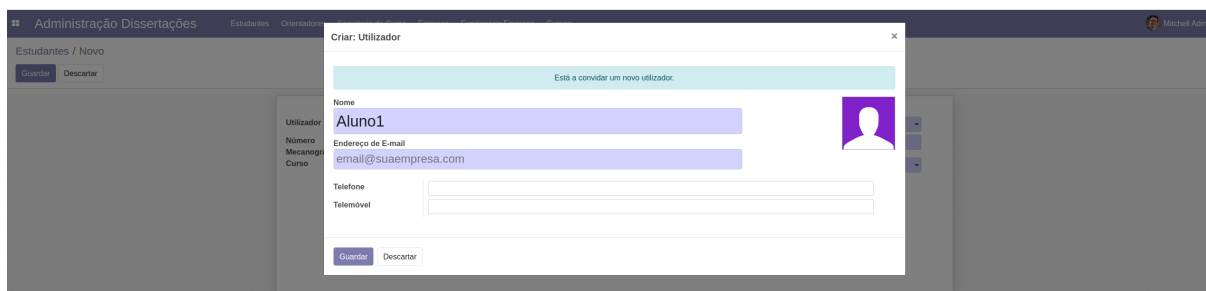


Figura 6.4: Registo de um novo Utilizador na plataforma

Administração Dissertações | Estudantes | Orientadores | Secretaria de Curso | Empresa | Funcionário Empresa | Cursos

Orientadores

Procurar...

Filtros Agrupar por Favoritos

<input type="checkbox"/>	Nome	Email	Identificação Universitária	Permissões	Cursos
<input type="checkbox"/>	Antonio Sousa	als@uminho.pt	D621	Diretor de Curso	2 registos
<input type="checkbox"/>	João Ramalho	jor@uminho.pt	D123	Orientador	1 registo
<input type="checkbox"/>	Coorientador1	coorientador1@mail.com	d456	Coorientador	1 registo

1-3/3 < >

Figura 6.5: Listagem dos Orientadores na plataforma

Administração Dissertações | Estudantes | Orientadores | Secretaria de Curso | Empresa | Funcionário Empresa | Cursos

Orientadores / Novo

Guardar Descartar

Utilizador
Número
Mecanográfico
Cursos
Centro de Investigação
Departamento
Permissões

Antonio Sousa
D621
CIEI
Algoritmi
Departamento de Informática
Diretor de Curso

Figura 6.6: Registo de um novo Orientador com permissões de Diretor de Curso

Administração Dissertações | Estudantes | Orientadores | Secretaria de Curso | Empresa | Funcionário Empresa | Cursos

Orientadores / Novo

Guardar Descartar

Utilizador
Número
Mecanográfico
Cursos
Centro de Investigação
Departamento
Permissões

João Ramalho
D123
CIEI
Algoritmi
Departamento de Informática
Orientador

Figura 6.7: Criação de um novo Orientador

Administração Dissertações | Estudantes | Orientadores | Secretaria de Curso | Empresa | Funcionário Empresa | Cursos

Orientadores / Novo

Guardar Descartar

Utilizador
Número
Mecanográfico
Cursos
Centro de Investigação
Departamento
Permissões

Coorientador1
d456
Gestão
Algoritmi2
Departamento de Gestão
Coorientador

Figura 6.8: Registo de um novo Coorientador

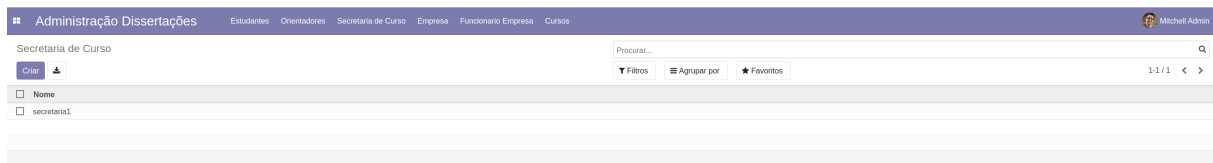


Figura 6.9: Listagem das Secretarias de Curso na plataforma

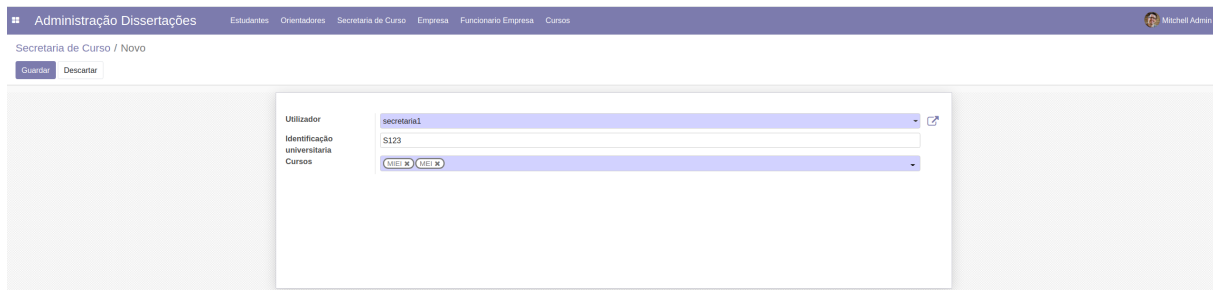


Figura 6.10: Registo de uma nova Secretaria de Curso

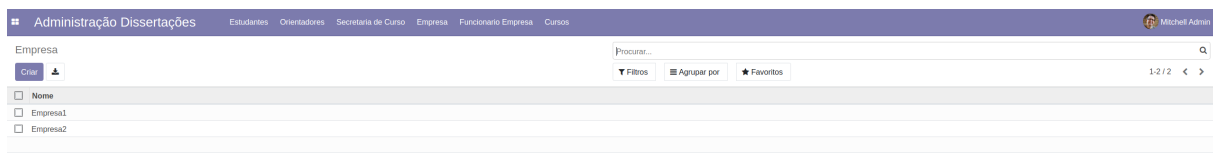


Figura 6.11: Listagem das Empresas registadas na plataforma

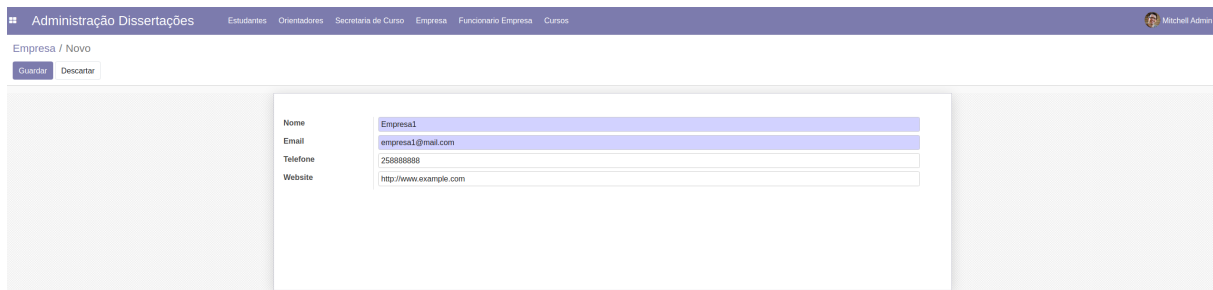


Figura 6.12: Registo de uma nova Empresa

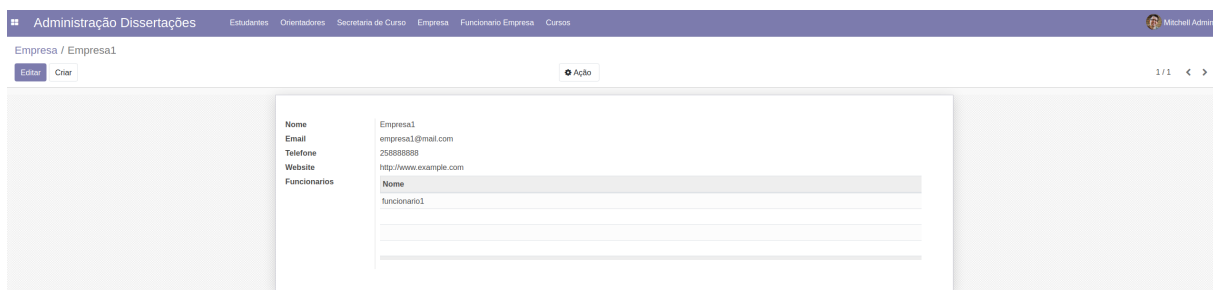


Figura 6.13: Vista da informação acerca de uma Empresa, incluindo os funcionários da mesma

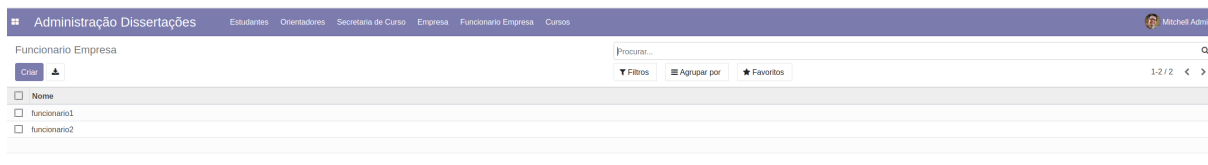


Figura 6.14: Listagem dos Funcionários de Empresa



Figura 6.15: Registo de um novo Funcionário de Empresa

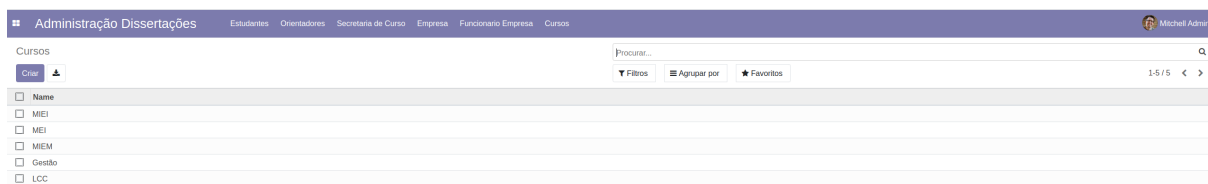


Figura 6.16: Listagem dos Cursos na plataforma

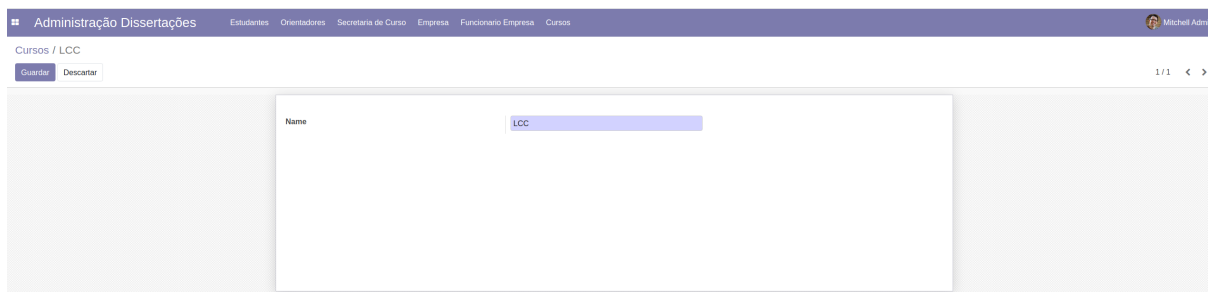


Figura 6.17: Registo de um novo Curso

6.2 Aluno

O Aluno é uma entidade no sistema que apenas tem acesso ao Menu **Admissão Dissertações** e através dele poderá visualizar todos os temas de dissertação alocados ao curso em que está inscrito. Ser-lhe-á possível visualizar informação detalhada acerca de cada tema e candidatar ao número de temas que bem entender. Após o processo de candidatura, será previsível que o Aluno contacte via email o Orientador que propôs o tema pelo qual está interessado de modo a que este lhe atribua o tema.

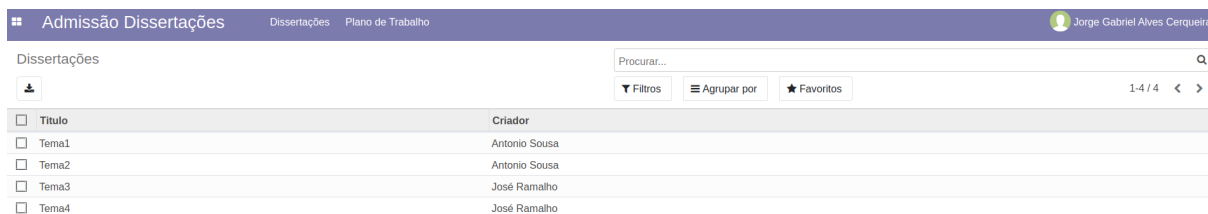


Figura 6.18: Listagem dos temas disponíveis para o Aluno se candidatar

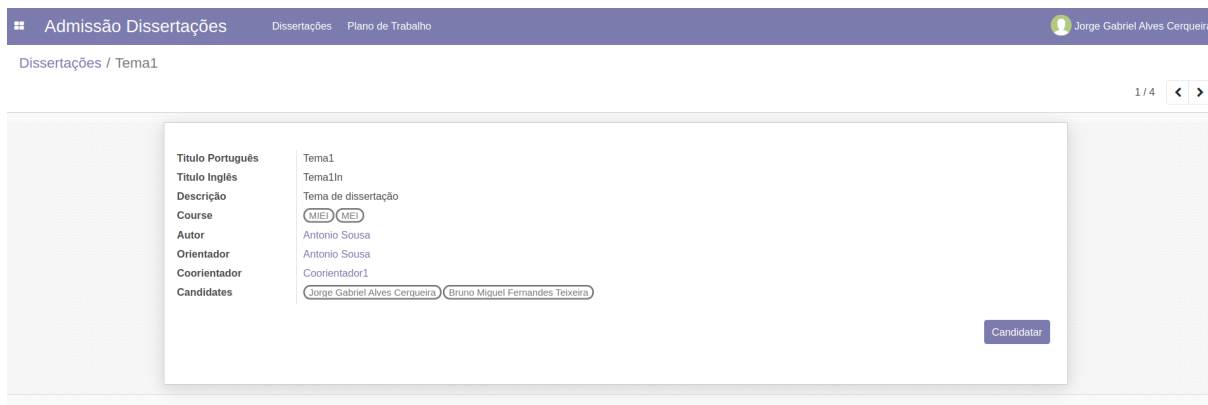


Figura 6.19: Informação sobre o tema e possibilidade de Candidatar

Depois de um tema atribuído, o Aluno terá acesso ao seu Plano de Trabalho onde poderá fazer o *download* do *template*, em *Latex* ou *Word*, e onde depois de preenchido e assinado pelo próprio, pelo Orientador e pelo Coorientador, deverá fazer *upload* do mesmo. Neste Menu conseguirá observar se o seu plano de trabalho já foi validado e assinado pelo Diretor de Curso, assim como numa fase posterior poderá submeter a sua Pré-Tese.

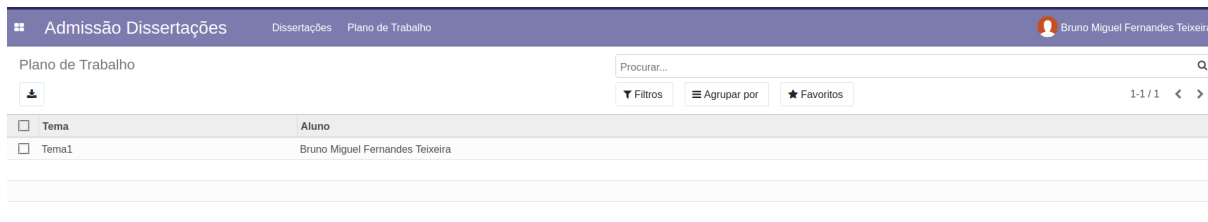


Figura 6.20: Plano de Trabalho associado ao tema que foi atribuído ao Aluno

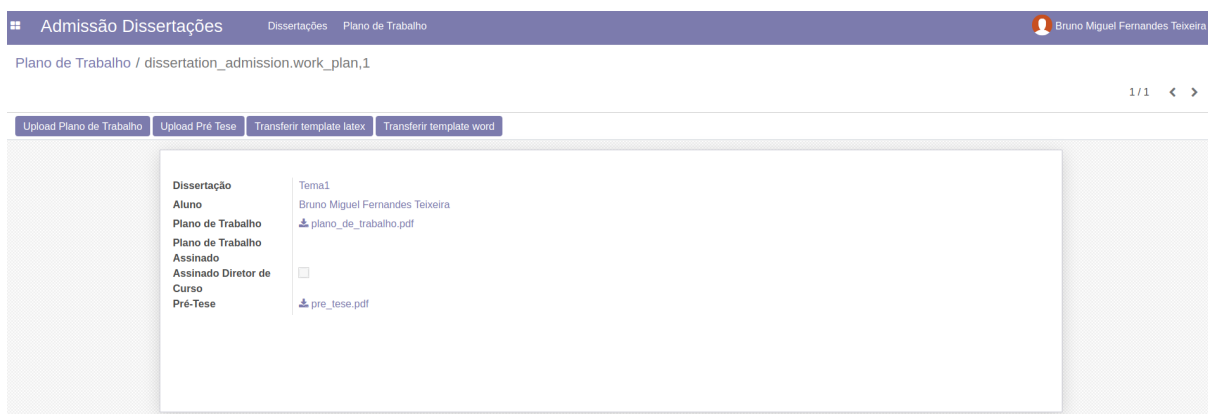
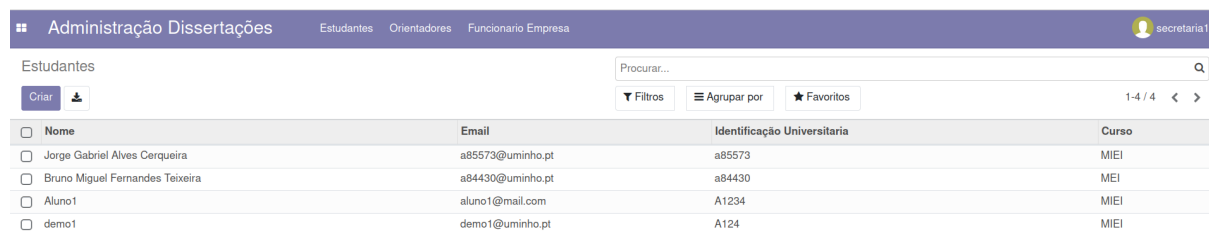


Figura 6.21: Opções disponíveis no Plano de Dissertação

6.3 Secretaria Direção de Curso

A Secretaria da Direção de Curso é um perfil de utilizador que tem várias funções na plataforma, entre as quais conseguir realizar ações semelhantes ao Administrador e Diretor de Curso, contudo apenas para os cursos dos quais desempenha funções de secretaria. Dessa forma, tem acesso ao Menu **Administração de Dissertações** onde pode registar Estudantes, Orientadores e Funcionários de Empresa que estejam associados aos seus cursos.

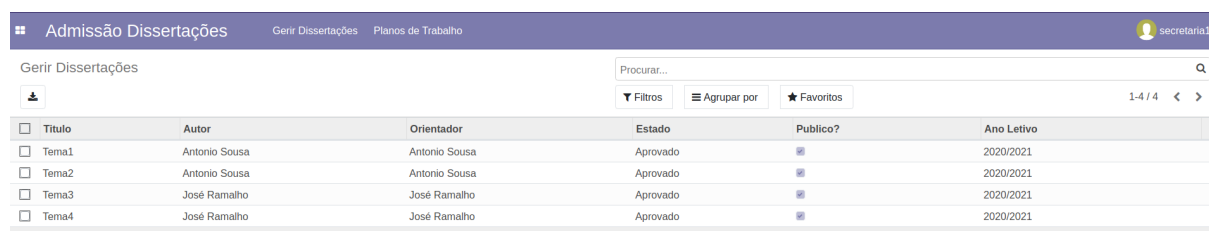


The screenshot shows the 'Administração Dissertações' menu with tabs for 'Estudantes', 'Orientadores', and 'Funcionario Empresa'. The 'Estudantes' tab is active, displaying a table of students with columns for 'Nome', 'Email', 'Identificação Universitaria', and 'Curso'. There are search and filter controls at the top.

<input type="checkbox"/>	Nome	Email	Identificação Universitaria	Curso
<input type="checkbox"/>	Jorge Gabriel Alves Cerqueira	a85573@uminho.pt	a85573	MIEI
<input type="checkbox"/>	Bruno Miguel Fernandes Teixeira	a84430@uminho.pt	a84430	MEI
<input type="checkbox"/>	Aluno1	aluno1@mail.com	A1234	MIEI
<input type="checkbox"/>	demo1	demo1@uminho.pt	A124	MIEI

Figura 6.22: Acesso ao Menu Administração Dissertações para entidades relacionadas com o curso

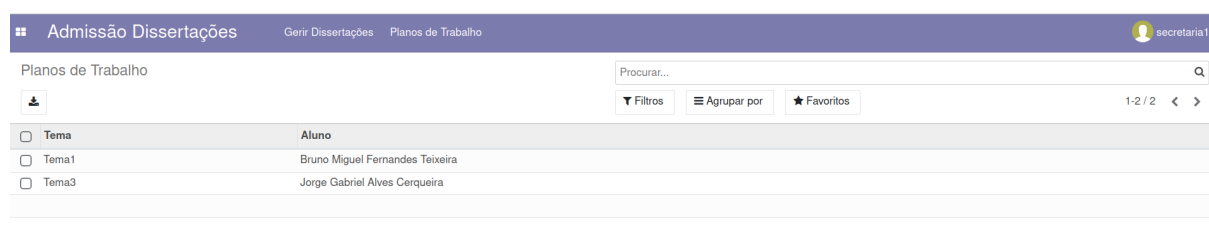
Esta entidade terá também acesso ao Menu **Admissão Dissertações** onde poderá visualizar todos os temas de dissertação e o seu estado, assim como desempenhará funções de verificação de documentação no separador Plano de Trabalho. Nesta segunda função, cabe à Secretaria de Curso verificar se os Planos de Trabalho submetidos pelos alunos cumprem todos os requisitos de modo a que o Diretor de Curso os possa assinar e concluir então esta fase da admissão à dissertação.



The screenshot shows the 'Admissão Dissertações' menu with tabs for 'Gerir Dissertações' and 'Planos de Trabalho'. The 'Gerir Dissertações' tab is active, displaying a table of dissertations with columns for 'Titulo', 'Autor', 'Orientador', 'Estado', 'Publico?', and 'Ano Letivo'. There are search and filter controls at the top.

<input type="checkbox"/>	Titulo	Autor	Orientador	Estado	Publico?	Ano Letivo
<input type="checkbox"/>	Tema1	Antonio Sousa	Antonio Sousa	Aprovado	<input checked="" type="checkbox"/>	2020/2021
<input type="checkbox"/>	Tema2	Antonio Sousa	Antonio Sousa	Aprovado	<input checked="" type="checkbox"/>	2020/2021
<input type="checkbox"/>	Tema3	José Ramalho	José Ramalho	Aprovado	<input checked="" type="checkbox"/>	2020/2021
<input type="checkbox"/>	Tema4	José Ramalho	José Ramalho	Aprovado	<input checked="" type="checkbox"/>	2020/2021

Figura 6.23: Visualização de todas as dissertações



The screenshot shows the 'Admissão Dissertações' menu with tabs for 'Gerir Dissertações' and 'Planos de Trabalho'. The 'Planos de Trabalho' tab is active, displaying a table of work plans with columns for 'Tema' and 'Aluno'. There are search and filter controls at the top.

<input type="checkbox"/>	Tema	Aluno
<input type="checkbox"/>	Tema1	Bruno Miguel Fernandes Teixeira
<input type="checkbox"/>	Tema3	Jorge Gabriel Alves Cerqueira

Figura 6.24: Listagem de todos os planos de trabalho

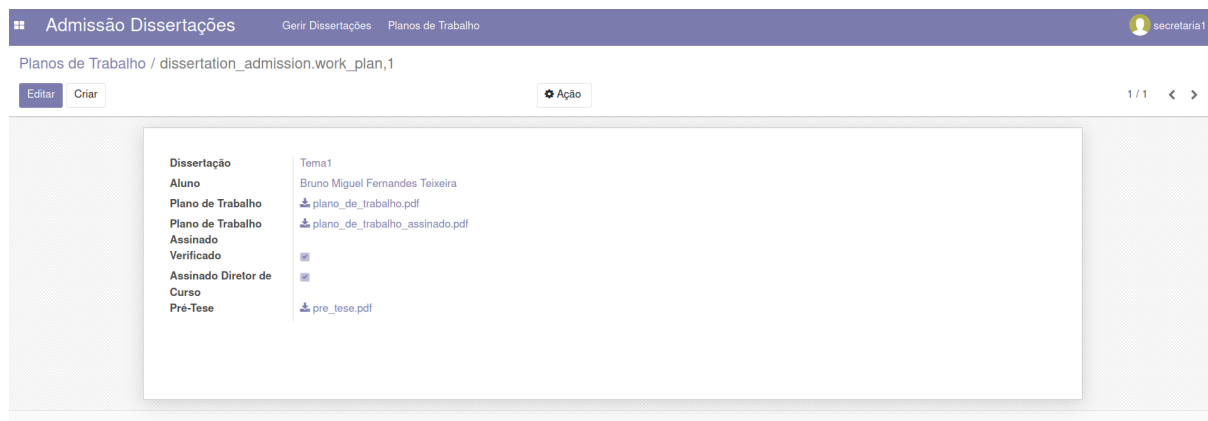


Figura 6.25: Edição de um plano de trabalho para verificação de campos

6.4 Orientador/Coorientador/Funcionário de Empresa

Estas 3 entidades têm funções muito semelhantes no sistema daí a junção das mesmas na mesma secção. Referir previamente que um Coorientador poderá aceder aos mesmos menus que o Orientador, contudo apenas tem permissões de visualização, e um Funcionário de Empresa tem exatamente as mesmas permissões que um Orientador, porém este só está apto para desempenhar o papel de Coorientador visto ser uma entidade externa à Universidade.

Deste modo, um Orientador pode visualizar todos os temas dos quais é autor, assim como fazer o registo de novos temas para os cursos aos quais está associado. Pode ainda fazer a edição de um tema de dissertação (antes deste ser aprovado e publicado) e atribuir o tema a um aluno que se tenha candidatado ao mesmo.

<input type="checkbox"/>	Título	Autor	Orientador	Estado	Público?	Ano Letivo
<input type="checkbox"/>	Tema3	José Ramalho	José Ramalho	Aprovado	<input checked="" type="checkbox"/>	2020/2021
<input type="checkbox"/>	Tema4	José Ramalho	José Ramalho	Aprovado	<input checked="" type="checkbox"/>	2020/2021

Figura 6.26: Listagem dos seus temas de dissertação

Dissertação

Título Português: Tema3

Título Inglês: Tema3En

Descrição: Tema 3 de dissertação

School Year: 2020/2021

Course: MEST

Orientador: José Ramalho

Coorientador Interno: [empty]

Coorientador Externo: funcionario1

Figura 6.27: Criação de um novo tema de dissertação

Dissertações / Tema4

Guardar Descartar 2 / 2

Dissertação
 Título Português
 Título Inglês
 Descrição
 Estado
 School Year
 Publico?
 Course
 Autor
 Orientador
 Coorientador Interno
 Coorientador Externo
 Estudante
 Candidates

Tema4
 Tema4EN
 tema4
 Aprovado
 2020/2021
☒
 MIEI
 José Ramalho
 José Ramalho
 funcionario2
 Jorge Gabriel Alves Cerqueira
 Jorge Gabriel Alves Cerqueira
 a85573@uminho.pt
 a85573
 MIEI

Figura 6.28: Atribuição do tema a um Aluno

Por fim, terá acesso ao separador Plano de Trabalho onde lhe será possível observar quais dos seus temas estão atribuídos, bem como ao Aluno que cada um deles está atribuído.

Admissão Dissertações Dissertações Planos de Trabalho José Ramalho

Planos de Trabalho

Procurar...

Filtros Agrupar por Favoritos 1-2 / 2

Tema	Aluno
<input type="checkbox"/> Tema3	Jorge Gabriel Alves Cerqueira
<input type="checkbox"/> Tema4	demo1

Figura 6.29: Listagem dos plano de trabalho em curso

6.5 Diretor de Curso

O Diretor de Curso tem várias funções na plataforma, entre as quais conseguir realizar ações semelhantes ao Administrador, contudo apenas para os cursos dos quais é diretor. Dessa forma, tem acesso ao Menu **Administração de Dissertações** onde pode registar Estudantes, Orientadores, Funcionários de Empresa e Secretarias de Cursos que estejam associados aos seus cursos. Por motivos de repetição não irão ser exibidas ilustrações relativas ao Menu **Administração de Dissertações** uma vez que o processo é exatamente igual ao ilustrado anteriormente na secção do **Administrador**.

Admissão Dissertações Dissertações Gerir Dissertações Planos de Trabalho Antonio Sousa

Dissertações

Criar

Procurar...

Filtros Agrupar por Favoritos 1-7 / 7

<input type="checkbox"/> Título	Autor	Orientador	Estado	Publico?	Ano Letivo
<input type="checkbox"/> Tema1	Antonio Sousa	Antonio Sousa	Aprovado	<input checked="" type="checkbox"/>	2020/2021
<input type="checkbox"/> Tema2	Antonio Sousa	Antonio Sousa	Aprovado	<input checked="" type="checkbox"/>	2020/2021
<input type="checkbox"/> Tema3	José Ramalho	José Ramalho	Aprovado	<input checked="" type="checkbox"/>	2020/2021
<input type="checkbox"/> Tema4	José Ramalho	José Ramalho	Aprovado	<input checked="" type="checkbox"/>	2020/2021
<input type="checkbox"/> Tema5	Antonio Sousa	Antonio Sousa	Aprovado	<input checked="" type="checkbox"/>	2020/2021
<input type="checkbox"/> Tema6	José Ramalho		Aprovado	<input checked="" type="checkbox"/>	2020/2021
<input type="checkbox"/> Tema7	José Ramalho		Aprovado	<input checked="" type="checkbox"/>	2020/2021

Figura 6.30: Visualização das entidades que o Diretor de Curso pode registar

Relativamente ao Menu **Admissão Dissertações**, o Diretor de Curso poderá efetuar todas as ações que um Orientador executa, nomeadamente criar novos temas e associar um tema a um aluno, contudo, além disso, este tem a responsabilidade de fazer a gestão das dissertações. No processo da gestão das dissertações, o diretor de curso é responsável por Aprovar, Rever, Reprovar, Publicar e Cancelar a publicação de todos os temas antes de estes ficarem visíveis para os alunos se candidatarem.

Título	Autor	Orientador	Estado	Publico?	Ano Letivo
<input type="checkbox"/> Tema1	Antonio Sousa	Antonio Sousa	Aprovado	<input checked="" type="checkbox"/>	2020/2021
<input type="checkbox"/> Tema2	Antonio Sousa	Antonio Sousa	Aprovado	<input checked="" type="checkbox"/>	2020/2021
<input type="checkbox"/> Tema3	José Ramalho	José Ramalho	Aprovado	<input checked="" type="checkbox"/>	2020/2021
<input type="checkbox"/> Tema4	José Ramalho	José Ramalho	Aprovação Pendente	<input type="checkbox"/>	2020/2021

Figura 6.31: Visualização do separador Dissertações com permissões iguais a um Orientador

Título	Autor	Orientador	Estado	Publico?	Ano Letivo
<input type="checkbox"/> Tema1	Antonio Sousa	Antonio Sousa	Aprovado	<input checked="" type="checkbox"/>	2020/2021
<input type="checkbox"/> Tema2	Antonio Sousa	Antonio Sousa	Aprovado	<input checked="" type="checkbox"/>	2020/2021
<input type="checkbox"/> Tema3	José Ramalho	José Ramalho	Aprovado	<input checked="" type="checkbox"/>	2020/2021
<input type="checkbox"/> Tema4	José Ramalho	José Ramalho	Aprovação Pendente	<input type="checkbox"/>	2020/2021

Figura 6.32: Visualização dos temas de dissertação e o seu estado atual

Gerir Dissertações / Tema4

Publicar múltiplas dissertações | Rever dissertação | Limpar Revisões | Aprovar | Reprovar

Título Português: Tema4
 Título Inglês: Tema4EN
 Descrição: tema4
 Estado: Aprovação Pendente
 School Year: 2020/2021
 Publico?: ☒
 Course:
 Autor: José Ramalho
 Orientador: José Ramalho
 Coordenador Interno:
 Coordenador Externo: funcionario2
 Estudante:
 Candidatos:
 Reviews:

Figura 6.33: Visualização das opções para gestão de um tema de dissertação

Rever dissertação

Falta o campo X

Confirmar Cancelar

Figura 6.34: Adição de uma *review* a um tema

Título	Autor	Orientador	Estado	Publico?	Ano Letivo
<input type="checkbox"/> TITULO	Mitchell Admin	Mitchell Admin	Aprovado	<input type="checkbox"/>	2020/2021

Figura 6.35: Publicação de um tema de dissertação

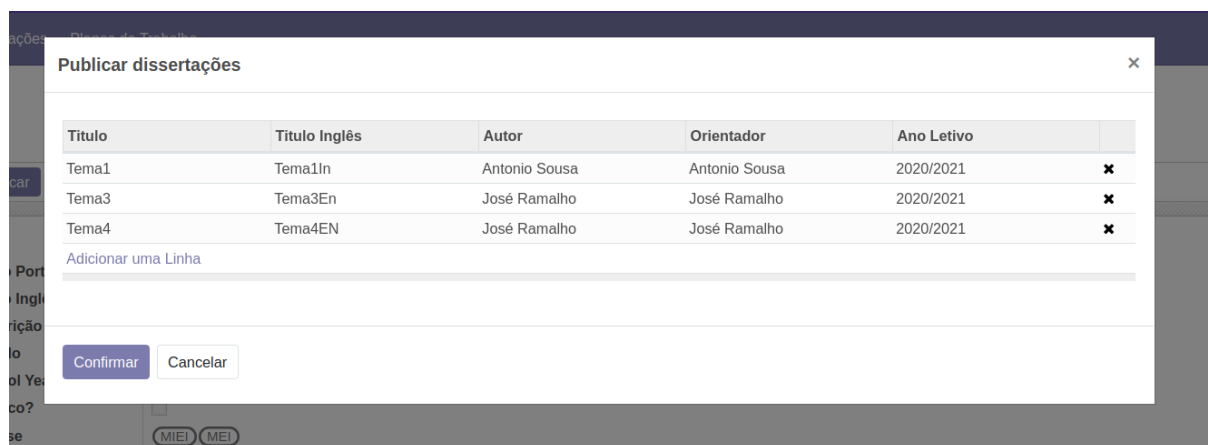


Figura 6.36: Publicação de vários temas de dissertação

Neste mesmo Menu, mas agora no separador de Plano de Trabalho, o Diretor de Curso tem as mesmas permissões que a Secretaria de Curso, ou seja, pode ele mesmo verificar a documentação inserida pelo aluno nos *uploads* dos ficheiros, porém cabe-lhe assinar todos os planos de trabalho relativos aos seus cursos. Nesse sentido, entra aqui o desenvolvimento da assinatura de múltiplos ficheiros PDF com recurso à CMD de modo a tornar este processo muito mais rápido e simples.

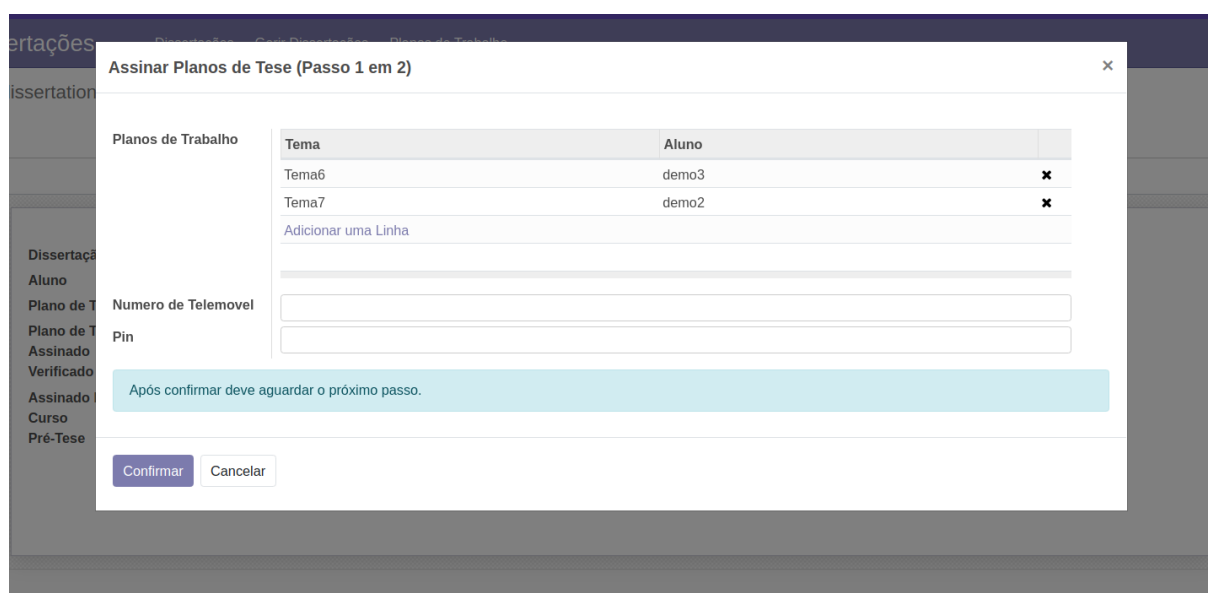


Figura 6.37: Assinatura de múltiplos documentos em simultâneo - 1

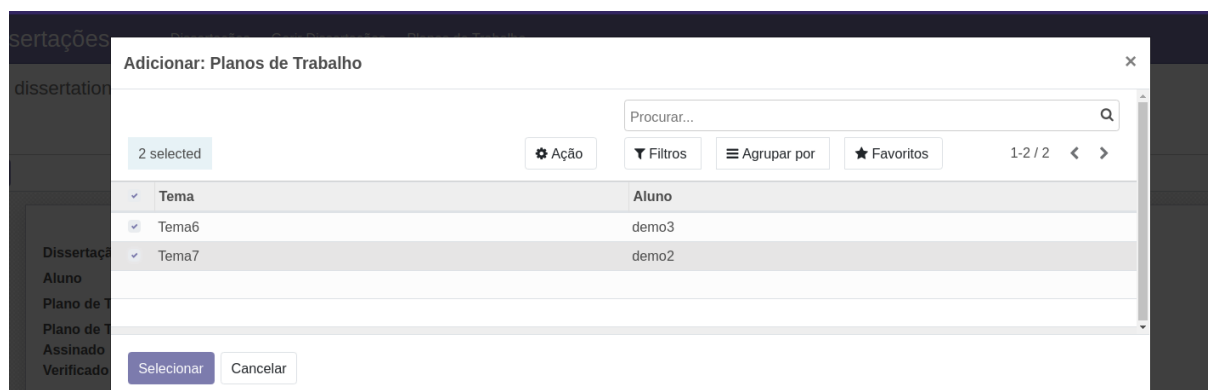


Figura 6.38: Assinatura de múltiplos documentos em simultâneo - 2

Capítulo 7

Conclusão

Uma vez que o tema do projeto escolhido pelo grupo vai de encontro ao desenvolvimento de uma plataforma que tem como objetivo resolver problemas relacionados com o curso e com expectativa de utilização imediata, houve desde logo uma motivação extra por parte dos elementos do grupo para a elaboração do projeto. Nesse sentido, e depois de todo o trabalho desenvolvido, o grupo pensa que os objetivos propostos inicialmente foram totalmente cumpridos visto que foi obtida uma solução válida e sólida para o problema em questão.

Durante o desenvolvimento do projeto, elegem-se como principais desafios o facto da utilização da *framework* **Odo**, uma vez que nenhum dos elementos do grupo tinha qualquer experiência com a mesma, assim como a sua lógica de funcionamento. Outro dos desafios passou pela necessidade de desenvolvimento de um segundo serviço para conseguir cumprir os requisitos relacionados com as assinaturas digitais, visto que foi necessária uma comunicação entre vários servidores, entre os quais o serviço de assinatura digital fornecido pelo governo.

Como trabalho futuro, e tendo em conta a utilização imediata da plataforma desenvolvida, será essencial a alocação do serviço num servidor do Departamento de Informática assim como a realização de testes à plataforma nesse mesmo ambiente de modo a verificar o seu correto funcionamento. Outro dos pontos que será necessário efetuar passa por realizar testes de carga e segurança ao serviço de modo a que no momento da sua utilização haja uma breve noção acerca do ponto de rutura da plataforma e se necessário fazer a replicação de algum dos serviços de modo a melhorar esse aspeto.