

Escola Superior de Tecnologia e Gestão

Licenciatura em Engenharia Informática

Estágio

Ano Letivo 2023/24

Especificação de Views

Elaborado em: 2024/03/13

Vasco Rodrigues 2021154888

Índice

1. Índice de Figuras.....	3
2. Introdução	4
3. Convenções de Nomenclatura Django	4
4. Site Map	4
4.1. Site Map geral	4
4.2. Tarefa	6
4.3. Empresa	10
4.4. Cliente	11
4.5. User_app.....	14
4.6. Accounts.....	17
4.7. Hora de esforço.....	19
5. Referências	23

1. Índice de Figuras

Figura 5-1 - Site Map	5
Figura 5-2 - View tarefa-create.....	6
Figura 5-3 - View tarefa-delete.....	6
Figura 5-4 - View tarefa-update	7
Figura 5-5 - View tarefa-detail.....	7
Figura 5-6 - View tarefa-list	7
Figura 5-7 - View tarefa-generateRel.....	8
Figura 5-8 - View tarefa-export_csv	8
Figura 5-9 - View tarefa-add_doc.....	9
Figura 5-10 - View empresa-deactivate.....	10
Figura 5-11 - View empresa-detail	10
Figura 5-12 - View cliente-create	11
Figura 5-13 - View cliente-delete	11
Figura 5-14 - View cliente-update	12
Figura 5-15 - View cliente-detail.....	12
Figura 5-16 - View cliente-list.....	12
Figura 5-17 - View cliente-generateRel	13
Figura 5-18 - View cliente-export_csv	13
Figura 5-19 - View user_app-detail	14
Figura 5-20 - View login-custom.....	14
Figura 5-21 - View profile	15
Figura 5-22 - View userapp-list.....	15
Figura 5-23 - View userapp-create	15
Figura 5-24 - View userapp-generateRel	16
Figura 5-25 - View userapp-export_csv.....	16
Figura 5-26 - View logout	17
Figura 5-27 - View password_change.....	17
Figura 5-28 - View password_done	17
Figura 5-29 - View password_reset.....	18
Figura 5-30 - View password_reset_done.....	18
Figura 5-31 - View password_reset_confirm.....	18
Figura 5-32 - View password_reset_complete.....	18
Figura 5-33 - View hora_esforco-create	19
Figura 5-34 - View hora_esforco-update.....	19
Figura 5-35 - View hora_esforco-delete	20
Figura 5-36 - View hora_esforco-detail	20
Figura 5-37 - View hora_esforco-list.....	20
Figura 5-38 - View hora_esforco-create-with-tarefa.....	21
Figura 5-39 - View hora_esforco-req-list.....	21
Figura 5-40 - View aceitar-hora_esforco-req.....	21
Figura 5-41 - View recusar-hora_esforco-req.....	22

2. Introdução

Este documento tem como objetivo descrever as *views* que serão implementadas na *Framework Django*, estas definirão o comportamento do site perante dadas situações.

Este documento é de extrema importância para o presente projeto por causa a tecnologia utilizada para o desenvolvimento do mesmo. Como já foi referido este será desenvolvido através da *framework Django* este para definir o funcionamento e apresentação das páginas ao utilizador utiliza algo chamado *views*, e, portanto, estas são capazes de processar os pedidos dos utilizadores.

Ao documentar as *views*, será definido um guia para a definição de cada uma delas facilitando o processo de desenvolvimento mais à frente pois servirá como referência.

3. Convenções de Nomenclatura Django

O *Django* utiliza uma convenção para a nomeação de *url models*, isto para ajudar a garantir a compreensão e manutenção do código, algumas destas convenções passam por:

- *modelname-list*: Esta é utilizada ao definir URL's que apresentem todas as instâncias de um modelo específico, são utilizadas quando a view utilizada é uma *ListView*;
- *modelname-detail*: Esta é utilizada ao definir URL's que apresentem todos os detalhes de uma instância específica de um modelo, são utilizadas quando a view utilizada é uma *DetailView*;
- *modelname-create*: Esta é utilizada ao definir URL's que permitem criar uma instância de um modelo específico;
- *modelname-delete*: Esta é utilizada ao definir URL's que permitem eliminar uma instância específica de um modelo;
- *modelname-update*: Esta é utilizada ao definir URL's que permitem atualizar uma instância específica de um modelo;

Os templates deverão ter exatamente o mesmo nome, mas ao invés de ser um "-" deverá ser um "_" como no exemplo "author-detail" este passará a ser "author_detail". Já para definir formulários como as Create e Update views deverá ser seguida a seguinte convenção: "modelname_form".

4. Site Map

4.1. Site Map geral

No site map serão especificados cada um dos *models* que serão utilizados, e cada view associada a este, no site map estão presentes então os seguintes:

- Tarefa;
- Empresa;
- Cliente;
- User_app;

- Accounts;
- Hora_esforco;
- Documentos.

Estes objetos já foram todos especificados no documento anterior menos o *Accounts*. Este contém todas as views padrão relativas a contas do *Django*, como login, logout, etc. Esta será mais bem especificada nos capítulos à frente.

Todas as views e objetos podem se vistas na **Figura 4-1**.

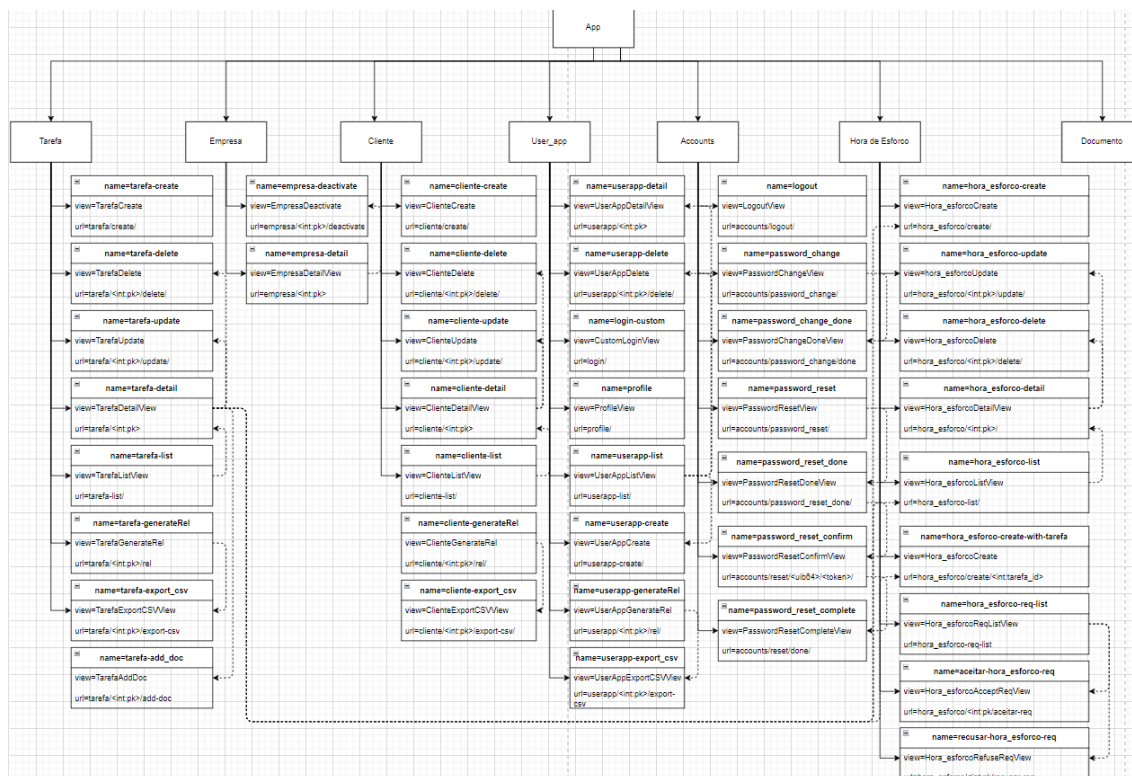


Figura 4-1 - Site Map

4.2. Tarefa

Neste Model, serão definidas todas as views relativas a tarefas, como:

- Criação de tarefas;
- Remoção de tarefas;
- Atualização de tarefas;
- Detalhes de uma tarefa;
- Listagem de tarefas;
- Gerar e exportar relatórios;
- Adicionar documentos

A primeira view será tarefa-create, que como o nome o indica esta permite a criação de tarefas, o seu nome será tarefa-create, a view será TarefaCreate e o seu url será tarefa/create/, tudo isto poderá ser visto na **Figura 4-2**.

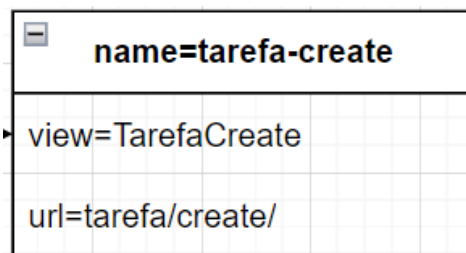


Figura 4-2 - View tarefa-create

Em seguida será a view tarefa-delete, que como o nome o indica esta permite remover uma tarefa específica, o seu nome será tarefa-delete, a view será TarefaDelete e o url será tarefa/<int:pk>/delete/ é passado no url um **int:pk** pois para remover uma tarefa específica será preciso o seu identificador, tudo isto pode ser visto na **Figura 4-3**.

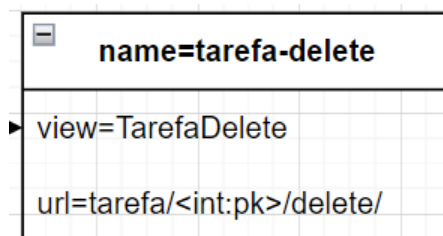


Figura 4-3 - View tarefa-delete

Em seguida será a view tarefa-update, que como o nome o indica esta permitirá a atualização de uma tarefa específica, o seu nome será tarefa-update, a view será TarefaUpdate e o url será

tarefa/<int:pk>/update, novamente, é passado o identificador da view para atualizar uma tarefa específica, tudo isto pode ser visto na **Figura 4-4**.

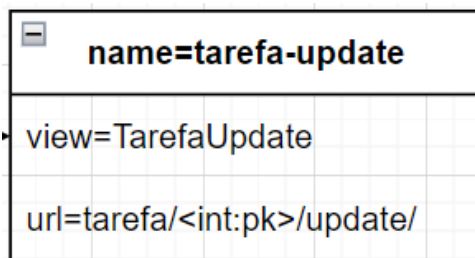


Figura 4-4 - View tarefa-update

Em seguida será a view tarefa-detail, esta permite ver os detalhes de uma tarefa em específico, o seu nome será então tarefa-detail, a view TarefaDetailView e o url tarefa/<int:pk> pois é necessário especificar a view da qual se pretende ver os detalhes, tudo isto pode ser visto na **Figura 4-5**.

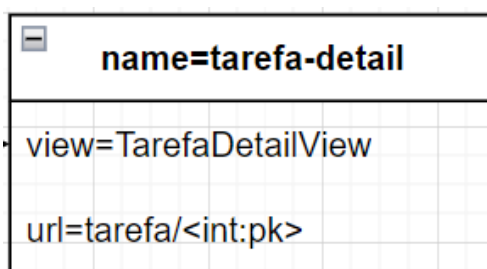


Figura 4-5 - View tarefa-detail

Em seguida será a view tarefa-list, que permite listar todas as tarefas associadas ao sistema, esta tem como nome tarefa-list, a view será TarefaListView e o url tarefa-list/, tudo isto pode ser visto na **Figura 4-6**.

A partir desta página pode ser acedida diretamente a página para adicionar horas de esforço.

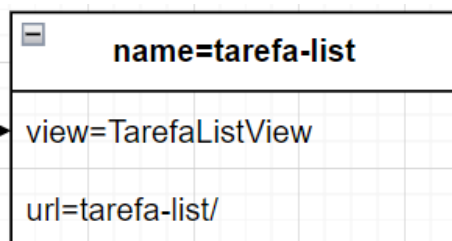


Figura 4-6 - View tarefa-list

Esta permitirá aceder aos detalhes de uma tarefa específica, removê-la ou atualizada, ou seja, poderá ser acedida a página tarefa-detail, tarefa-delete e tarefa-update selecionando uma tarefa desta listagem, será possível ainda aceder à view que permite adicionar novas obras.

Em seguida será a tarefa-generateRel, que irá permitir a um administrador de empresa gerar qualquer tipo de relatórios a uma Obra, então o seu nome será tarefa-generateRel, a view será TarefaGenerateRel e o url será tarefa/<int:pk>/generateRel, tudo isto pode ser visto na **Figura 4-7**.



	name=tarefa-generateRel
	view=TarefaGenerateRel
	url=tarefa/<int:pk>/generateRel

Figura 4-7 - View tarefa-generateRel

Esta página irá dar acesso a uma outra página onde o utilizador transferir o relatório gerado em formato csv, nomeadamente a view tarefa-export_csv.

Nesta nova view tarefa-export_csv, permitirá transferir uma versão em csv do relatório gerado pela view anterior, o seu nome será tarefa-export_csv, a view será a TarefaExportCSVView e o url para ela será tarefa/<int:pk>/export-csv, tudo isto pode ser visto na **Figura 4-8**.

	name=tarefa-export_csv
	view=TarefaExportCSVView
	url=tarefa/<int:pk>/export-csv

Figura 4-8 - View tarefa-export_csv

Por fim vem a view responsável por adicionar documentos à própria tarefa, o nome para aceder a esta view é tarefa-add_doc, a view em si é TarefaAddDoc e o seu url é tarefa/<int:pk>/add-doc, tudo isto pode ser visto na .


	name=tarefa-add_doc
view=TarefaAddDoc	
url=tarefa/<int:pk>/add-doc	

Figura 4-9 - View tarefa-add_doc

4.3. Empresa

Neste Model, serão definidas todas as views relativas à empresa, como:

- Desativação de empresas;
- Detalhes de uma tarefa;

Em seguida será a view empresa-deactivate, que como o nome o indica esta permitirá desativar uma empresa, o seu nome será então empresa-deactivate, a view EmpresaDeactivate e o url empresa/<int:pk>/deactivate é necessário passar o identificador para especificar a empresa que se pretende desativar, tudo isto pode ser visto na **Figura 4-10**.

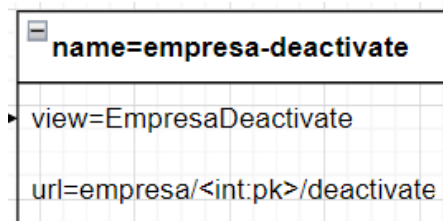


Figura 4-10 - View empresa-deactivate

Em seguida será a view empresa-detail, esta será responsável pela apresentação dos detalhes da empresa do utilizador, é nesta página também que poderão ser editadas as informações da empresa em si, o nome desta será então empresa-detail, a view EmpresaDetailView e o url empresa/<int:pk> para assim especificar a empresa da qual se pretende ver os detalhes, tudo isto pode ser visto na **Figura 4-11**.

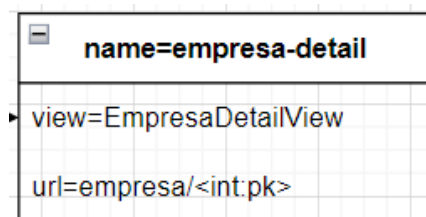


Figura 4-11 - View empresa-detail

A partir desta podem ser acedidas outras duas views nomeadamente empresa-update e empresa-deactivate, isto está ilustrado através das setas na imagem.

4.4. Cliente

Neste Model, serão definidas todas as views relativas aos clientes de uma empresa, como:

- Criação de clientes;
- Remoção de cliente;
- Atualização de cliente;
- Detalhes de um cliente;
- Listagem de clientes;
- Gerar e exportar relatórios.

Então a primeira view será cliente-create, esta como o nome o indica permite criar um cliente, associando-o automaticamente à empresa do utilizador (administrador de empresa) que o criou, então o seu nome será cliente-create, a view ClienteCreate e o url cliente/create, tudo isto pode ser visto na **Figura 4-12**.

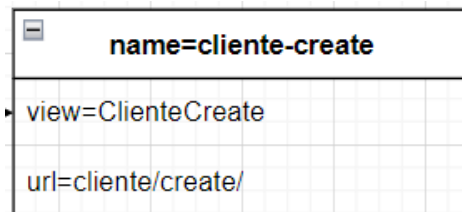


Figura 4-12 - View cliente-create

Em seguida será cliente-delete, que permitirá a um administrador de empresa remover um cliente da sua própria empresa, então o seu nome será cliente-delete, a view ClienteDelete e o url cliente/<int:pk>/delete/, tudo isto pode ser visto na **Figura 4-13**.

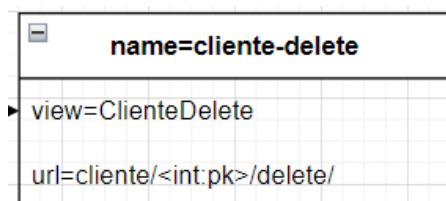


Figura 4-13 - View cliente-delete

Em seguida será a view cliente-update, que permitirá a um administrador de empresa atualizar um cliente específico da sua própria empresa, então o seu nome será cliente-update, a view será ClienteUpdate e o url cliente/<int:pk>/update/, tudo isto pode ser visto na **Figura 4-14**.

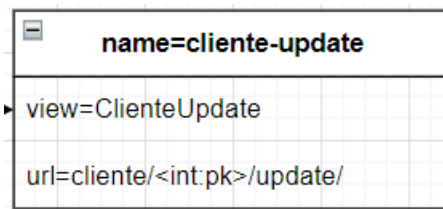


Figura 4-14 - View cliente-update

Em seguida será a view cliente-detail, que permitirá a um administrador de empresa ver os detalhes de um cliente específico associado à sua própria empresa, o nome será então cliente-detail, a view ClienteDetailView e o url cliente/<int:pk>/detail/, tudo isto pode ser visto na **Figura 4-15**.

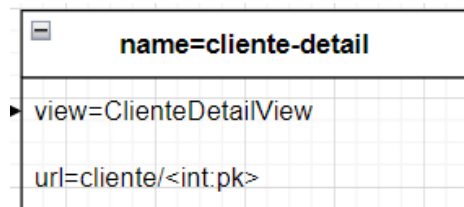


Figura 4-15 - View cliente-detail

Em seguida será a view cliente-list, esta como o nome o indica permitirá a um administrador de empresa listar todos os clientes associados à sua própria empresa, então o seu nome será cliente-list, a view ClienteListView e o url cliente-list/, tudo isto pode ser visto na **Figura 4-16**.

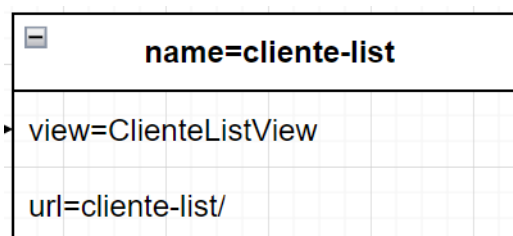


Figura 4-16 - View cliente-list

A partir da listagem de clientes será possível ver os detalhes de um cliente, atualizá-lo e removê-lo ou adicionar clientes.

A seguinte view permite então gerar um relatório de esforço para um cliente através de duas datas definidas pelo utilizador, o nome para a aceder será cliente-generateRel o nome da view em si será ClienteGenerateRel e o seu url cliente/<int:pk>/rel/, tudo isto poderá ser visto na **Figura 4-17**.


	name=cliente-generateRel
	view=ClienteGenerateRel
	url=cliente/<int:pk>/rel/

Figura 4-17 - View cliente-generateRel

A partir desta view pode ser acedida uma outra que permite exportar o relatório gerado.

Por fim existe a view que permite exportar o relatório gerado pela view anterior para um ficheiro csv, esta pode ser acedida através do nome cliente-export_csv, a view terá o nome ClienteExportCSVView e o seu url será "cliente/<int:pk>/export-csv/", tudo isto pode ser visto na **Figura 4-18**.

	name=cliente-export_csv
	view=ClienteExportCSVView
	url=cliente/<int:pk>/export-csv/

Figura 4-18 - View cliente-export_csv

4.5. User_app

Neste Model, serão definidas todas as views relativas aos utilizadores específicos do sistema, como:

- Detalhes de um utilizador específico;
- Atualizar utilizadores;
- Remover utilizadores;
- Listar utilizadores;
- Criar utilizadores;
- Criar e gerar relatórios de esforço;
- Autenticação e perfil de utilizadores

A primeira view será user_app-detail, esta permitirá visualizar os detalhes de um utilizador específico ou os seus próprios, esta terá o nome user_app-detail, a view User_appDetailView e o url user_app/<int:pk>/ para especificar o utilizador desejado, tudo isto pode ser visto na **Figura 4-19**.

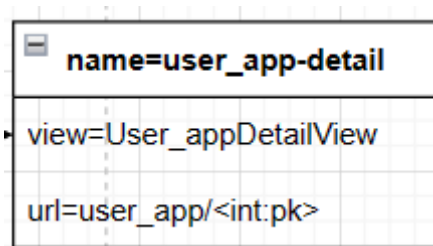


Figura 4-19 - View user_app-detail

Em seguida será a view user_app-update, esta irá permitir que um utilizador atualize os seus próprios detalhes, o nome será então user_app-update, a view User_appUpdate e o url user_app/<int:pk>/update/, tudo isto pode ser visto na **Figura 4-20**. A origem da referência não foi encontrada..

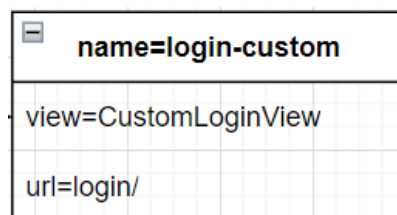


Figura 4-20 - View login-custom

Em seguida a view profile permite visualizar os próprios detalhes do utilizador, é aqui também que ele poderá editar os seus próprios detalhes e alterar a sua palavra-passe, o nome para aceder a esta view é profile, a view em si será ProfileView e o url simplesmente profile/, tudo isto pode ser visto na **Figura 4-21**.


 name=profile
view=ProfileView
url=profile/

Figura 4-21 - View profile

Em seguida vem a view responsável por listar todos os trabalhadores de uma dada empresa, esta pode ser acedida através de userapp-list, tem como nome UserAppListView e como url userapp-list, tudo isto pode ser visto na **Figura 4-22**.


 name=userapp-list
view=UserAppListView
url=userapp-list/

Figura 4-22 - View userapp-list

Esta view permite aceder a três outras, nomeadamente userapp-detail, userapp-delete e userapp-create.

Em seguida vem a view responsável pela criação de trabalhadores para uma empresa esta pode ser acedida através de userapp-create, a view em si tem como nome UserAppCreate e o url é userapp-create/, tudo isto pode ser visto na **Figura 4-23**.


 name=userapp-create
view=UserAppCreate
url=userapp-create/

Figura 4-23 - View userapp-create

A próxima view é a responsável por gerar relatórios de esforço para utilizadores em específico, ela pode ser acedida através de userapp-generateRel, a view tem como nome UserAppGenerateRel e o url é userapp/<int:pk>/rel/, tudo isto pode ser visto na **Figura 4-24**.


	name=userapp-generateRel
	view=UserAppGenerateRel
	url=userapp/<int:pk>/rel/

Figura 4-24 - View userapp-generateRel

A partir desta pode ser acedida uma outra view responsável pela exportação do relatório para um formato csv.

Por fim temos a view responsável pela exportação do relatório gerado pela view anterior para o formato csv, esta pode ser acedida através de userapp-export_csv, a view tem como nome UserAppExportCSVView e o url é userapp/<int:pk>/export-csv, tudo isto pode ser visto na **Figura 4-25**.


	name=userapp-export_csv
	view=UserAppExportCSVView
	url=userapp/<int:pk>/export-csv

Figura 4-25 - View userapp-export_csv

4.6. Accounts

Nesta subcapítulo será utilizado views e funcionalidades já implementadas no Django, aqui encontram-se aspetos como:

- Logout;
- Alteração de password;
- Recuperação de password.

A primeira e é bastante simples, como o nome o indica é responsável pelo logout de um utilizador, tendo mesmo como nome logout, como view LogoutView e url account/logout/, ambos estes casos podem ser visualizados na **Figura 4-26**.



	name=logout
	view=LogoutView
	url=accounts/logout/

Figura 4-26 - View logout

Agora para a alteração de password serão necessárias duas views password-change e password-change-done, a primeira servirá para a alteração da password em si e a segunda será apenas uma página informativa de confirmação de alteração da mesma, estas terão então como nome password-change e password-change-done, como views PasswordChangeView e PasswordChangeDoneView e como url's account/password_change/ e accounts/password_change/done/, tudo isto pode ser visto nas **Figura 4-27** e **Figura 4-28**.



	name=password_change
	view>PasswordChangeView
	url=accounts/password_change/

Figura 4-27 - View password_change



	name=password_change_done
	view>PasswordChangeDoneView
	url=accounts/password_change/done

Figura 4-28 - View password_done

Estas páginas são sequenciais oque significa que uma leva à outra.

Em seguida para a recuperação da password, para isto o Django implementa um total de 4 views `password_reset`, `password_reset_done`, `password_reset_confirm` e `password_reset_complete`, a primeira servirá para iniciar o processo de recuperação, nomeadamente especificação de email para o qual será enviado o o link de recuperação, a seugnda será a página de confirmação de envio de email, a terceira será o formulário para definir uma nova password e por fim será a página informativa de confirmação de alteração da password, então respetivamente estas terão os nomes `password_reset`, `password_reset_done`, `password_reset_confirm` e `password_reset_complete`, terão como views `PasswordResetView`, `PasswordResetDoneView`, `PasswordResetConfirmView` e `PasswordResetCompleteView`, por fim terão como urls `accounts/password-reset/`, `accounts/password_reset_done/`, `accounts/reset/<uib64>/<token>/` e `accounts/reset/done/`, tudo isto pode ser visto nas **Figura 4-29**, **Figura 4-30**, **Figura 4-31** e **Figura 4-32**.

name=password_reset
view>PasswordResetView
url=accounts/password_reset/

Figura 4-29 - View password_reset

name=password_reset_done
view>PasswordResetDoneView
url=accounts/password_reset_done/

Figura 4-30 - View password_reset_done

name=password_reset_confirm
view>PasswordResetConfirmView
url=accounts/reset/<uib64>/<token>/

Figura 4-31 - View password_reset_confirm

name=password_reset_complete
view>PasswordResetCompleteView
url=accounts/reset/done/

Figura 4-32 - View password_reset_complete

Novamente estas páginas são sequenciais oque significa que uma leva a outra e essa levará à seguinte, e assim vai.

4.7. Hora de esforço

Neste Model, serão definidas todas as views relativas às horas de esforço dedicadas a uma Obra/Serviço por parte de um grupo de trabalhadores, como:

- Criação de horas de esforço;
- Atualização de horas de esforço;
- Remoção de horas de esforço;
- Detalhes de horas de esforço;
- Listagem de horas de esforço;
- Gestão de pedido de horas atrasadas.

A primeira view será hora_esforco-create, esta permitirá a um trabalhador associar ou não outros trabalhadores a uma nova hora de esforço, o nome será então hora_esforco-create, a view hora_esforcoCreate e o url tarefa/<int:pk>/hora_esforco/create, tudo isto pode ser visto na **Figura 4-33**.

name=hora_esforco-create	
view=hora_esforcoCreate	
url=hora_esforco/create/	

Figura 4-33 - View hora_esforco-create

Em seguida será a hora_esforco-update, esta permitirá a um trabalhador ou administrador de empresa alterar os detalhes de uma horas de esforço, o nome será então hora_esforco-update, a view hora_esforcoUpdate e o url tarefa/<int:pk>/hora_esforco/<int:pk>/update/, tudo isto pode ser visto na **Figura 4-34**.

name=hora_esforco-update	
view=hora_esforcoUpdate	
url=hora_esforco/<int:pk>/update/	

Figura 4-34 - View hora_esforco-update

Em seguida será a hora_esforco-delete, esta permitirá a um trabalhador ou administrador de empresa remover no caso de um trabalhador um Horas de esforço à qual está associado no segundo caso permitirá a um administrador de empresa remover horas de esforço de todos os trabalhadores que estão associados a esta, esta terá como nome hora_esforco-delete, como view hora_esforcoDelete e como url tarefa/<int:pk>/hora_esforco/<int:pk>/delete/, tudo isto pode ser visto na **Figura 4-35**.


	name=hora_esforco-delete
▶	view=hora_esforcoDelete
	url=hora_esforco/<int:pk>/delete/

Figura 4-35 - View hora_esforco-delete

Em seguida será a hora_esforco-detail, esta permitirá a um administrador de empresa verificar quais os trabalhadores foram associados às horas de esforço, esta tem então como nome hora_esforco-detail, a view será hora_esforcoDetailView e o url será tarefa/<int:pk>/hora_esforco/<int:pk>/, tudo isto pode ser visto na **Figura 4-36**.


	name=hora_esforco-detail
▶	view=hora_esforcoDetailView
	url=hora_esforco/<int:pk>/

Figura 4-36 - View hora_esforco-detail

Em seguida será a hora_esforco-list, esta permitirá a um administrador de empresa listar todas as horas de esforço associadas a uma dada tarefa, esta será chamada hora_esforco-list, a view será Hora_esforcoListView e o url será tarefa/<int:pk>/hora_esforco-list/, tudo isto pode ser visto na **Figura 4-37**.


	name=hora_esforco-list
▶	view=hora_esforcoListView
▶	url=hora_esforco-list/

Figura 4-37 - View hora_esforco-list

Esta página de listagem de horas de esforço poderá levar às views que permitem adicionar, atualizar, eliminar e criar horas de esforço.

Em seguida temos uma view extremamente semelhante à view de criação, porém esta recebe um tarefa, ou seja, esta já tem a tarefa automaticamente selecionada, ela pode ser acedida através de hora_esforco-create-with-tarefa, a view é a mesma hora_esforcoCreate e o url é hora_esforco/create/<int:tarefa_id>, tudo isto pode ser visto na **Figura 4-38**.

name=hora_esforco-create-with-tarefa	
view=hora_esforcoCreate	
url=hora_esforco/create/<int:tarefa_id>	

Figura 4-38 - View hora_esforco-create-with-tarefa

Em seguida temos a view responsável por listar todos pedidos de horas atrasadas, esta pode ser acedida através de hora_esforco-req-list, a view é Hora_esforcoReqListView e o url hora_esforco-req-list, tudo isto pode ser visto na **Figura 4-39**.

name=hora_esforco-req-list	
view=Hora_esforcoReqListView	
url=hora_esforco-req-list	

Figura 4-39 - View hora_esforco-req-list

Esta página permite aceder a duas outras views, nomeadamente às views que permitem aceitar e recusar estes pedidos de horas atrasadas.

Em seguida temos a view responsável por aceitar estes pedidos, ela pode ser acedida através aceita-hora_esforco-req, a view chama-se Hora_esforcoAcceptReqView e o url é hora_esforco/<int:pk>/aceitar-req/, tudo isto pode ser visto na **Figura 4-40**.

name=aceitar-hora_esforco-req	
view=Hora_esforcoAcceptReqView	
url=hora_esforco/<int:pk>/aceitar-req	

Figura 4-40 - View aceitar-hora_esforco-req

Por fim temos a view responsável por recusar estes pedidos, ela pode ser acedida através recusar-hora_esforco-req, a view chama-se Hora_esforcoRefuseReqView e o url é hora_esforco/<int:pk>/recusar-req/, tudo isto pode ser visto na **Figura 4-41**.


	name=recusar-hora_esforco-req
view=Hora_esforcoRefuseReqView	
url=hora_esforco/<int:pk/recusar-req	

Figura 4-41 - View recusar-hora_esforco-req

5. Referências

- Adhikari, Arjun. 'Django URLs Naming Conventions and Best Practices'. Python is Love, 9 March 2023. <https://pythonislove.com/django-urls-naming-conventions-and-best-practices>.
- GitHub. 'Django/Django/Contrib/Auth/Urls.Py at Main · Django/Django'. Accessed 13 March 2024. <https://github.com/django/django/blob/main/django/contrib/auth/urls.py>.