|  |  |
| --- | --- |
| Plano “MyOwnList” | Resultado de imagem para governo de portugal educaÃ§Ã£o  **PAP**  Projeto de Aptidão Profissional    **MyOwnList**  **Gustavo Firmino / José Félix**  a22542 / a22322    **Ano Letivo 2020/2021**  Curso cofinanciado    eu_ce_vert_grdeu_ce_vert_grdeu_ce_vert_grdeu_ce_vert_grd  **[DOC GPSI\_PAP\_206]** |

|  |
| --- |
| **1º Modelo Concetual (Análise de Requisitos e desenho comportamental do sistema)** |
| Visitantes    Utilizadores        Administrador |

|  |
| --- |
| **2º Modelo Lógico (Estrutura de dados do Sistema de Informação)** |
|  |

|  |
| --- |
| **3º Modelo Físico (Implementação da Base de Dados no SGBD)** |
| create table users(  iduser int identity primary key,  username varchar(20) not null check(len(username) >= 3 and len(username) <= 20),  email varchar(100) not null check(email like '%@%.%'),  password varchar(255) not null check(len(password) >= 6 and len(password) <= 255),  profiletype int default(0),  isdarkmode bit default(0),  issuspended bit default(0),  pathimage varchar(max) not null,  created datetime default(getdate()),  lastupdate datetime default(getdate())  )  create table user\_comments(  idcomment int identity primary key,  iduser\_1 int references users(iduser) not null,  iduser\_2 int references users(iduser) not null,  comment text not null,  created datetime default(getdate()),  lastupdate datetime default(getdate())  )  create table typesnotification(  idtypenotification int identity primary key,  name varchar(20) not null check(len(name) > 0 and len(name) <= 20),  created datetime default(getdate()),  lastupdate datetime default(getdate())  )  create table notifications(  idnotification int identity primary key,  idtypenotification int references typesnotification(idtypenotification) not null,  iduser int references users(iduser) not null,  topic varchar(100) not null,  message varchar(255) check(len(message) > 0 and len(message) <= 255),  date datetime default(getdate())  )  create table categoriesfsa(  idcategory int identity primary key,  name varchar(20) not null check(len(name) >= 3 and len(name) <= 20),  description varchar(255) check(len(description) <= 255),  created datetime default(getdate()),  lastupdate datetime default(getdate())  )  create table datafsa(  idfsa int identity,  typefsa int not null,  idcategory int references categoriesfsa(idcategory) not null,  name varchar(100) not null check(len(name) >= 2 and len(name) <= 100),  description varchar(255) check(len(description) > 0 and len(description) <= 255),  duration int default(0),  rating int,  numeps int default(1),  numseasons int default(1),  director varchar(100) check(len(director) > 3 and len(director) <= 100),  pathimage varchar(max) not null,  created datetime default(getdate()),  lastupdate datetime default(getdate()),  constraint pk\_datafsa primary key (idfsa, typefsa)  )  create table fsa\_comments(  idcommentfsa int identity primary key,  iduser int references users(iduser) not null,  idfsa int not null,  typefsa int not null,  message text,  isactive bit default(1),  pathimage varchar(max),  created datetime default(getdate()),  lastupdate datetime default(getdate()),  foreign key (idfsa, typefsa) references datafsa(idfsa, typefsa)  )  create table fsa\_suggestions(  idsuggestion int identity primary key,  iduser int references users(iduser) not null,  typefsa int not null,  name varchar(100) not null check(len(name) > 0 and len(name) <= 100),  description text not null,  created datetime default(getdate())  )  create table userfsa\_regs(  idreg int identity primary key,  iduser int references users(iduser) not null,  idfsa int not null,  typefsa int not null,  personalrating int,  watchedeps int default(0),  statusfsa int,  created datetime default(getdate()),  lastupdate datetime default(getdate()),  foreign key (idfsa, typefsa) references datafsa(idfsa, typefsa)  )  create table friendlists(  idfriendlist int identity primary key,  iduser1 int references users(iduser) not null,  iduser2 int references users(iduser) not null,  friendshipstarted datetime default(getdate()),  friendshipstatus int default(0),  created datetime default(getdate()),  lastupdate datetime default(getdate())  )  create table calendars(  idcalendar int identity primary key,  iduser int references users(iduser) not null,  idfsa int not null,  typefsa int not null,  idtypenotification int references typesnotification(idtypenotification) not null,  date datetime default(getdate()),  description varchar(255) check(len(description) >= 255),  created datetime default(getdate()),  foreign key (idfsa, typefsa) references datafsa(idfsa, typefsa)  )  create table chatmessages(  idchat int identity primary key,  iduser1 int references users(iduser) not null,  iduser2 int references users(iduser) not null,  idtypenotification int references typesnotification(idtypenotification) not null,  message text not null,  pathimage varchar(max),  date datetime default(getdate()),  isread bit default(0)  )  create table typesreport(  idtypereport int identity primary key,  name varchar(20) not null check(len(name) > 0 and len(name) <= 20),  created datetime default(getdate()),  lastupdate datetime default(getdate())  )  create table reports(  idreport int identity primary key,  iduser int references users(iduser) not null,  idtypereport int references typesreport(idtypereport) not null,  topic varchar(255) not null check(len(topic) >= 10 and len(topic) <= 255),  description text not null,  issolved bit default(0),  created datetime default(getdate()),  lastupdate datetime default(getdate())  )  create table shares(  idshare int identity primary key,  iduser1 int references users(iduser) not null,  iduser2 int references users(iduser) not null,  idfsa int not null,  typefsa int not null,  idtypenotification int references typesnotification(idtypenotification) not null,  date datetime default(getdate()),  foreign key (idfsa, typefsa) references datafsa(idfsa, typefsa)  )  create table faq\_categories(  idcategoryfaq int identity primary key,  name varchar(30) not null check(len(name) >= 2 and len(name) <= 30),  description text not null,  isactive bit default(1),  created datetime default(getdate()),  lastupdate datetime default(getdate())  )  create table faq\_topics(  idfaq int identity primary key,  idcategoryfaq int references faq\_categories(idcategoryfaq) not null,  iduser int references users(iduser) not null,  subject varchar(255) not null check(len(subject) >= 3 and len(subject) <= 255),  message text not null,  isactive bit default(1),  pathimage varchar(max),  created datetime default(getdate()),  lastupdate datetime default(getdate())  ) |

|  |
| --- |
| **4º Protótipo (Desenho abstrato da aplicação na ótica do utilizador)** |
| Index    Login    Register    Profile    Another user’s profile    Personal List    Settings    FSA Page (Not Logged)    FSA Page (Logged)    Calendar    Suggestion    Report Problem    FAQ |

|  |
| --- |
| **5º Análise critica (Levantamento de necessidades)** |
| Utilizamos o aplicativo Discord para organizar-nos e para comunicar-nos sobre o nosso projeto.  Uma das dificuldades encontradas foi a ausência de professor acompanhante durante muito tempo, o que nos fez perder mais da metade dos apoios à PAP que teríamos.  Até agora estivemos a fazer a preparação para a criação do site. Estamos a estudar uma nova linguagem para nós, React, o que vai nos levar algum tempo para perceber bem. |

|  |
| --- |
| **6º Fontes – Bibliografia | Webgrafia** |
| [Primeiro site de Inspiração](https://myanimelist.net/)  [Instalação do React](https://www.notion.so/Instala-o-das-ferramentas-9eee9e25550e4477b3d0b432b605aca2)  [Discord](https://discord.com/)  [Segundo site de inspiração](https://www.imdb.com/)  [Draw.io (Diagramas)](https://app.diagrams.net/)  [Aplicação utilizada para mockups](https://balsamiq.com/) |

|  |
| --- |
| **7º Reflexão finais – Conclusões sobre o desenvolvimento do projeto** |
| Para um futuro, gostaríamos de implementar um script o qual irá buscar os Filmes/Séries/Animes à web e os implemente no MyOwnList.  Caso tenhamos tempo após a criação do WebSite partiremos para a criação da aplicação mobile/Desktop. |