



Prova de Aptidão Profissional

Técnico de Gestão e Programação de

Sistemas Informáticos

Relatório de Projeto

KidsHeaven

Elaborado por

Nome: José Pedro Gameiro Duarte

N.º: 2219103 Triénio: 19-22

Índice

1- Introdução.....	4
2 – Tecnologias e recursos	5
3 – Implementação.....	8
3.1 Cronograma de desenvolvimento do projeto	8
3.2 Aspetos técnicos do desenvolvimento do projeto.....	8
4 – Conclusão	11
5 - Bibliografia	12
6 – Anexos	13

1- Introdução

A aplicação KidsHeaven é um website de venda de itens usados direcionados para crianças, nele as pessoas que não têm tanto poder económico podem poupar e as pessoas que querem vender itens usados também podem vender, e assim, não só ajudam os que mais necessitam como ajudam a elas mesmas.

A aplicação é direcionada para pessoas entre os 20 e os 50 anos, já que é nestas idades que as pessoas estão a ter filhos ou estão a querer vender os itens que os filhos já não usam.

Esta aplicação também foi pensada no âmbito de sustentabilidade, atualmente nós usamos muitos recursos naturais para fazer coisas novas e como sabemos isso está a afetar o nosso planeta, então com a aplicação, já que os itens são usados, penso que ajuda nesse ponto.

Com este projeto, como já referi em cima, pretendo ajudar os mais necessitados economicamente e a que as pessoas revendam os seus itens usados. Com isso quero que os dois lados se ajudem economicamente para um crescimento económico mais sustentável. A realizar este projeto também evolui em termos técnicos.

Neste relatório falarei sobre as tecnologias que usei para fazer a aplicação, como as usei e como geri o meu tempo.

2 – Tecnologias e recursos



Para fazer o frontend da aplicação usei o Angular, um dos motivos para usar o angular foi que permite fazer uma SPA (Single Page Application), ao fazer uma SPA a aplicação fica mais performática e otimizada, assim, permitindo ao utilizador ter uma melhor experiência enquanto usa a aplicação. Outro motivo para usar o angular foi que, durante o período de férias, eu experimentei usar angular e pela experiência que tive não achei difícil e gostei de usar.



Para fazer o design da aplicação utilizei o PrimeNG, o PrimeNG é um conjunto de bibliotecas de componentes, com ele consegui fazer todo o design da aplicação, as vantagens de usá-lo foram que me permitiu fazer o design rapidamente e ao mesmo tempo agradável para o utilizador.

O PrimeNG também me ajudou a implementar a API do Google Maps, usando o mapa que eles disponibilizam.



Na aplicação também usei o Firebase, com o Firebase consegui fazer um serviço de login seguro com facilidade e também consegui armazenar as imagens do utilizador e dos anúncios no serviço Storage que o Firebase disponibiliza.



Para fazer a API usei .NET 5, o principal motivo para usar .NET 5 foi que já tinha experiência, desde o segundo ano que aprendi .NET então decidi usar para fazer a aplicação.



Para fazer requisições à base de dados usei o Micro ORM Dapper, usei o Dapper pois é o Micro ORM mais rápido e é fácil de usar, ajudou e facilitou a fazer toda a parte da API.



Para desenvolver a API também usei o Swagger, usei o Swagger para testar os requisitos que a API estava a fazer à base de dados para ver se estava funcional e pronto para ir para o frontend.



Usei SQL Server para a base de dados, pois já venho a usar o SQL Server desde o segundo ano, assim sendo, foi o que me senti mais a vontade de utilizar, também usei porque o SQL é pratico e fácil de usar.



Para gerir a base de dados usei o SQL Server Management Studio, já tinha usado o SQL Server Management e achei pratico e fácil de utilizar.



Usei o WebStorm pois é uma IDE baseada em JavaScript, assim ajudando/facilitando a criar o frontend.



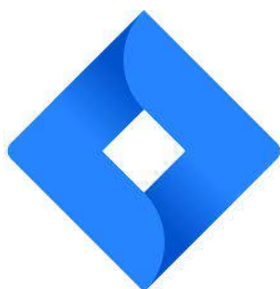
O Rider é um IDE baseado em .NET, e assim como o WebStorm ajudou no frontend, o Rider ajudou a fazer o backend.



Para fazer o esquema da base de dados usei o DbSchema, como já tinha usado e achei que não era difícil de usar, foi a minha escolha para fazer o esquema.



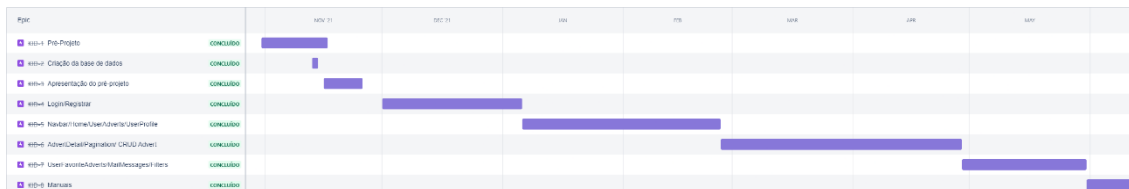
Para fazer o design da aplicação usei o Whimsical, com ele fiz todas as interfaces da aplicação facilmente.



Para fazer o cronograma de desenvolvimento da aplicação usei o Jira Software.

3 – Implementação

3.1 Cronograma de desenvolvimento do projeto



3.2 Aspectos técnicos do desenvolvimento do projeto

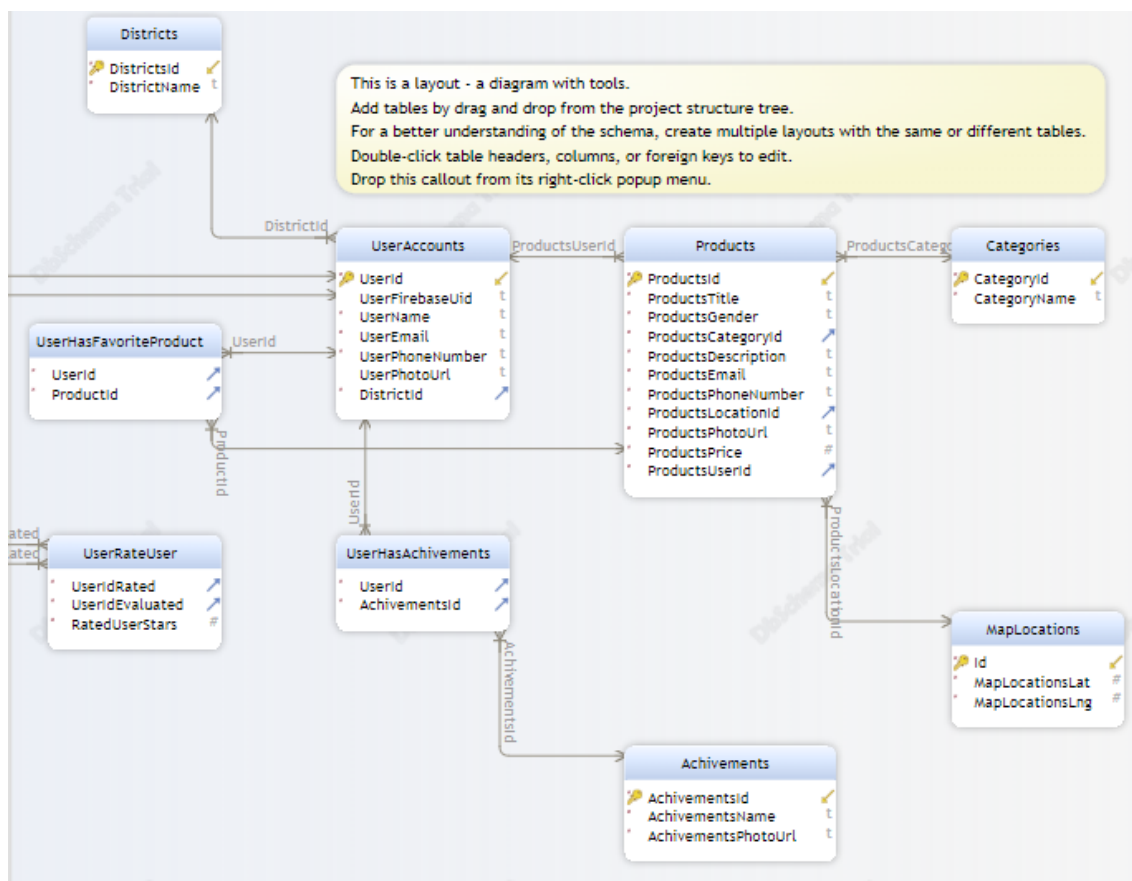


Table Categories

Idx	Field Name	Data Type
*PK	Categoryid	int IDENTITY
*	CategoryName	varchar(100) COLLATE Latin1_General_CI_AS
Indexes		
PK	Pk_Categories_Categoryid	ON Categoryid
Referring Foreign Keys		
FK	fk_products_categories	(Categoryid) --> Products(ProductsCategoryid)

Table Districts

Idx	Field Name	Data Type
*	DistrictsId	int IDENTITY
*	DistrictName	varchar(100) COLLATE Latin1_General_CI_AS
Indexes		
	Pk_Districts_DistrictsId	ON DistrictsId
Referring Foreign Keys		
	fk_useraccounts_districts	(DistrictsId) → UserAccounts (DistrictId)

Table Products

Idx	Field Name	Data Type
*	ProductsId	int IDENTITY
*	ProductsTitle	varchar(max) COLLATE Latin1_General_CI_AS
*	ProductsGender	varchar(50) COLLATE Latin1_General_CI_AS
*	ProductsCategoryId	int
*	ProductsDescription	varchar(max) COLLATE Latin1_General_CI_AS
*	ProductsEmail	varchar(100) COLLATE Latin1_General_CI_AS
*	ProductsPhoneNumber	varchar(30) COLLATE Latin1_General_CI_AS
*	ProductsLocationId	int
*	ProductsPhotoUrl	varchar(max) COLLATE Latin1_General_CI_AS
*	ProductsPrice	int
*	ProductsUserId	int
Indexes		
	Pk_Products_ProductsId	ON ProductsId
Foreign Keys		
	fk_products_categories	(ProductsCategoryId) → Categories (CategoryId)
	fk_products_maplocations	(ProductsLocationId) → MapLocations (Id)
	fk_products_useraccounts	(ProductsUserId) → UserAccounts (UserId)
Referring Foreign Keys		
	fk_userhasfavoriteproduct	(ProductsId) → UserHasFavoriteProduct (ProductId)

Table UserAccounts

Idx	Field Name	Data Type
*	UserId	int IDENTITY
	UserFirebaseUid	varchar(max) COLLATE Latin1_General_CI_AS
*	UserName	varchar(100) COLLATE Latin1_General_CI_AS
*	UserEmail	varchar(100) COLLATE Latin1_General_CI_AS
*	UserPhoneNumber	varchar(20) COLLATE Latin1_General_CI_AS
	UserPhotoUrl	varchar(max) COLLATE Latin1_General_CI_AS
*	DistrictId	int
Indexes		
	Pk_UserAccounts_UserId	ON UserId
Foreign Keys		
	fk_useraccounts_districts	(DistrictId) → Districts (DistrictsId)
Referring Foreign Keys		
	fk_UserId	(UserId) → UserHasAchievements
	fk_products_useraccounts	(UserId) → Products (ProductsUserId)
	fk_userrateduser_useraccounts	(UserId) → UserRateUser (UserIdEvaluated)
	fk_userrateuser_useraccounts	(UserId) → UserRateUser (UserIdRated)
	fk_usershasfavoriteproduct	(UserId) → UserHasFavoriteProduct

Table UserRateUser

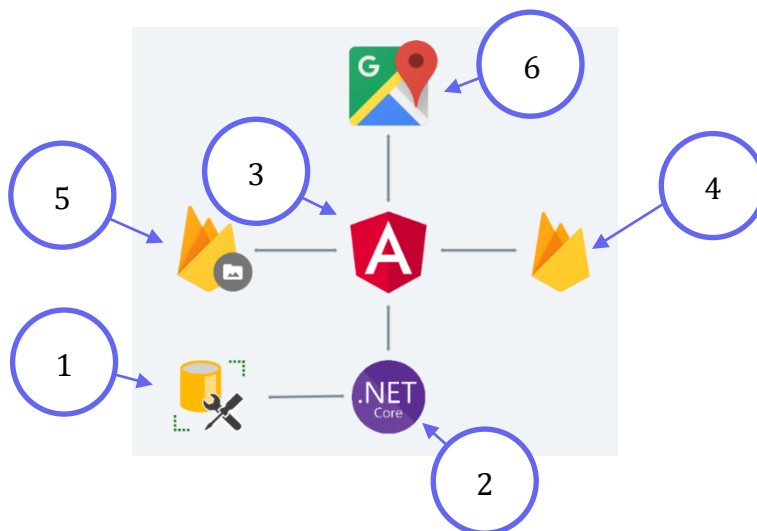
Idx	Field Name	Data Type
*	UserIdRated	int
*	UserIdEvaluated	int
*	RatedUserStars	int
Foreign Keys		
	fk_userrateduser_useraccounts	(UserIdEvaluated) → UserAccounts (UserId)
	fk_userrateuser_useraccounts	(UserIdRated) → UserAccounts (UserId)

Table MapLocations

Idx	Field Name	Data Type
*	Id	int IDENTITY
*	MapLocationsLat	int
*	MapLocationsLng	int
Indexes		
	pk_Tbl	ON Id
Referring Foreign Keys		
	fk_products_maplocations	(Id) → Products (ProductsLocationId)

Table UserHasFavoriteProduct

Idx	Field Name	Data Type
*/	UserId	int
*/	ProductId	int
Foreign Keys		
/	fk_userhasfavoriteproduct	(ProductId) → Products(ProductsId)
/	fk_usershasfavoriteproduct	(UserId) → UserAccounts



Base de dados (1), aqui é onde todos os dados da aplicação são guardados, estes dados serão processados no .NET/Backend (2), no .NET será feito requisitos à base de dados que vão ser enviados para o Angular/Frontend (3), quando os dados forem recebidos por um serviço no frontend poderão ser usados em qualquer lado da aplicação.

No frontend quando se cria conta num formulário, os dados serão mandados primeiro para o Firebase (4) e depois será mandado para o backend que vai mandar os dados para a base de dados.

Quando for para criar um anúncio num formulário, as imagens serão mandadas para o Firebase Storage (5) e a localização será inserida no mapa do google, depois todos os dados também serão mandados para o backend que manda para a base de dados

~~REQ0001 — Sistema de autenticação com Firebase~~

~~REQ0002 — Página de criação de anúncio~~

~~REQ0003 — Página inicial a mostrar os anúncios com paginação~~

~~REQ0004 – Filtros e barra de pesquisa para os anúncios~~

~~REQ0005 – Página do anúncio onde terá as informações do anúncio e onde se poderá dar rating ao utilizador~~

~~REQ0006 – Perfil do utilizador que mostra as suas informações podendo ser editadas~~

REQ0007 – Sistema de conquistas que também é mostrado no perfil do utilizador – Não feito

REQ0008 – Página de mensagens – Não feito

REQ0009 – Sistema de mensagens para os utilizadores conversarem entre si – Não feito

~~REQ0010 – Página de meus anúncios onde os utilizadores podem ver, editar e apagar os seus anúncios~~

~~REQ0011 – Página de anúncios favoritos, onde os utilizadores podem ver os seus anúncios favoritos ou retirá-los dos favoritos~~

Os requisitos 7, 8 e 9 não foram feitos por motivos de tempo, quando fui para realizá-los já não tinha tempo, então, decidi cortar os que achei menos importantes.

Por muito que o sistema de mensagens fosse importante, com a falta de tempo tive de trocar. A troca foi que em vez de haver o sistema todo de mensagens, existe um botão que redireciona para o email, no email terá lá o email do utilizador que será enviada a mensagem e poderá conversar através de email ou por telemóvel pois o utilizador tem sempre um número de telemóvel associado a ele.

4 – Conclusão

Durante a realização deste projeto tive bastantes dificuldades, principalmente, no início do projeto, a compreender o funcionamento do Angular, mas consegui passar essas dificuldades iniciais com algumas pesquisas, assim, conseguindo não só avançar no projeto como aprender bastante. Também tive algumas dificuldades no design do projeto, não consegui fazer bem como tinha planeado, mas quando comecei a compreender melhor o funcionamento do PrimeNG, consegui superar as dificuldades, e assim, não só superei como até melhorei algumas partes do design do projeto.

Com este projeto consegui evoluir e aprender bastante, maioritariamente no Angular, também foi um bom projeto para perceber que tenho de ter bastante atenção ao tempo que tenho e como usar esse tempo eficientemente.

5 - Bibliografia

<https://stackoverflow.com/>

6 – Anexos