Wine Catalog

DataBase

Final Practical Work

P8G6
Bruno Mendes 81850
João Teixeira 80192

Introduction

This work was done in the scope of the database subject. The objective of this practical work is to develop a graphical interface that permits the user to interact with one database in SQL Server to manipulate data.

The work purpose was to do something like of wine catalog with all the information of wine and all information related to this theme. This application was developed using WPF in the language of C# with SQL Server combine.

This application has the idea of being an open catalog to everyone to see, so with that in mind we decided not to do users, don't make sense for this.

Some basic features of the application:

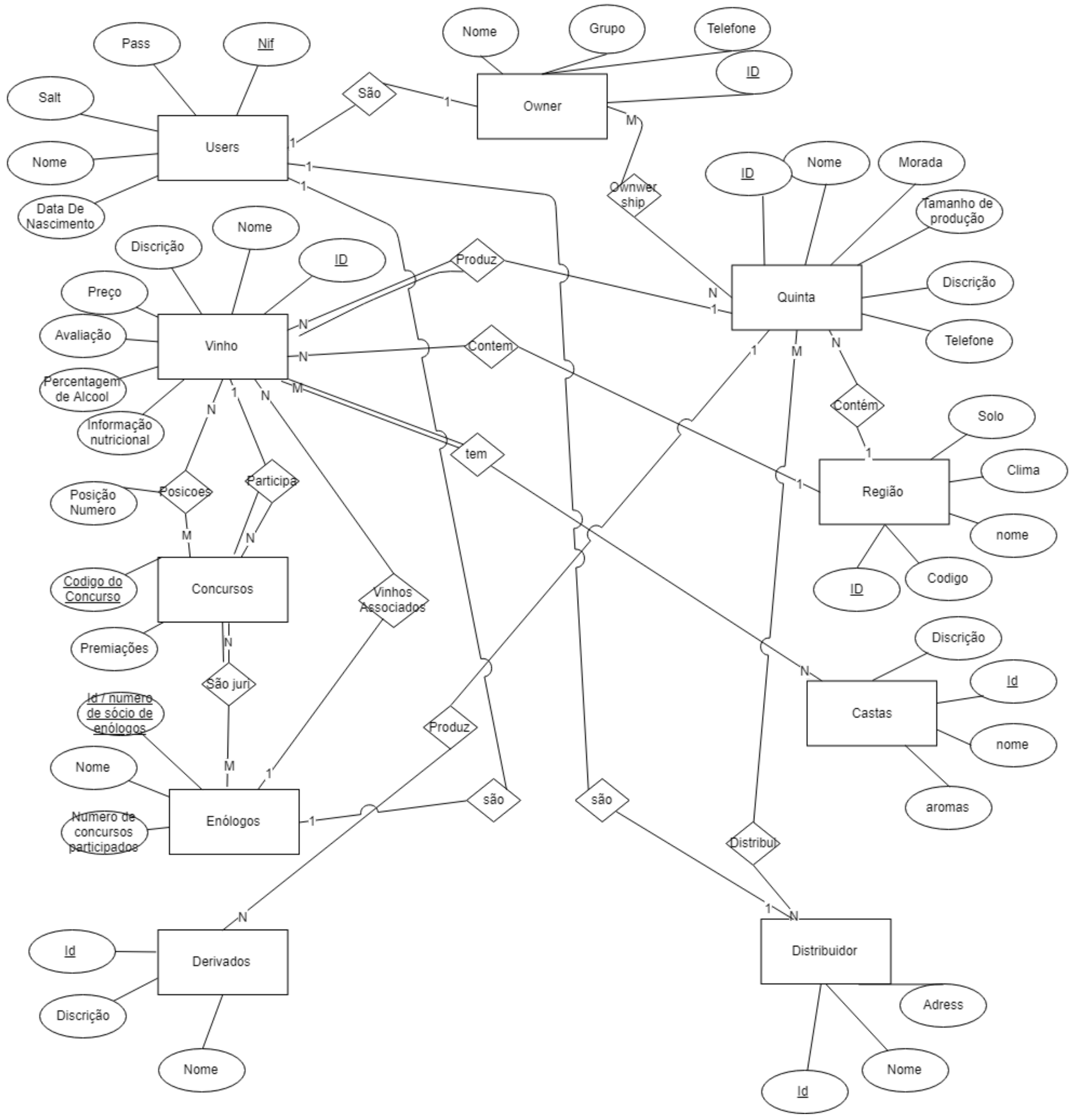
1. Insert/Manage/Remove wines;
2. Insert oenologist and his relation with determined Wine;
3. Manage Regions/Farms
4. Insert/Manage relations between wines and castes;
5. Manage wine contest and prizes

Requirements analysis

1. A system like a catalog of Wines where users can see all the information related to this theme.
2. An Owner can have ownership of more than one farm and is characterized by a Name, group, phone number, and one auto-generated ID.
3. An Oenologist can have more than one wine related with him and can be a jury of one or more contest, he is characterized by a Name, and number of participations in a contest, a unique number of a partner.
4. A Distributor has relations with a farm and is characterized by a Name, Address, and a unique ID.
5. A Region as the relation with farms and wines and is characterized by the type of ground, weather, name, postal code and a unique ID.
6. A Farm has a relation with wine, wine derivatives and with more than one distributor, and is characterized by a name, an address, a production size, description, phone, and his auto-generated ID.
7. A Wine that can participate in contest, and can gain prizes in more than one contest, can have more than one caste, and can have more than one Oenologist related with him, this has the attribute of name, alcohol percentage, price, evaluation, description, nutritional information, service temperature, and his one ID.
8. A derivative is related to farm and is characterized with name, description and his own unique ID.
9. A caste is related to n wines and is characterized by having a name, an aroma, a description, and a unique ID.

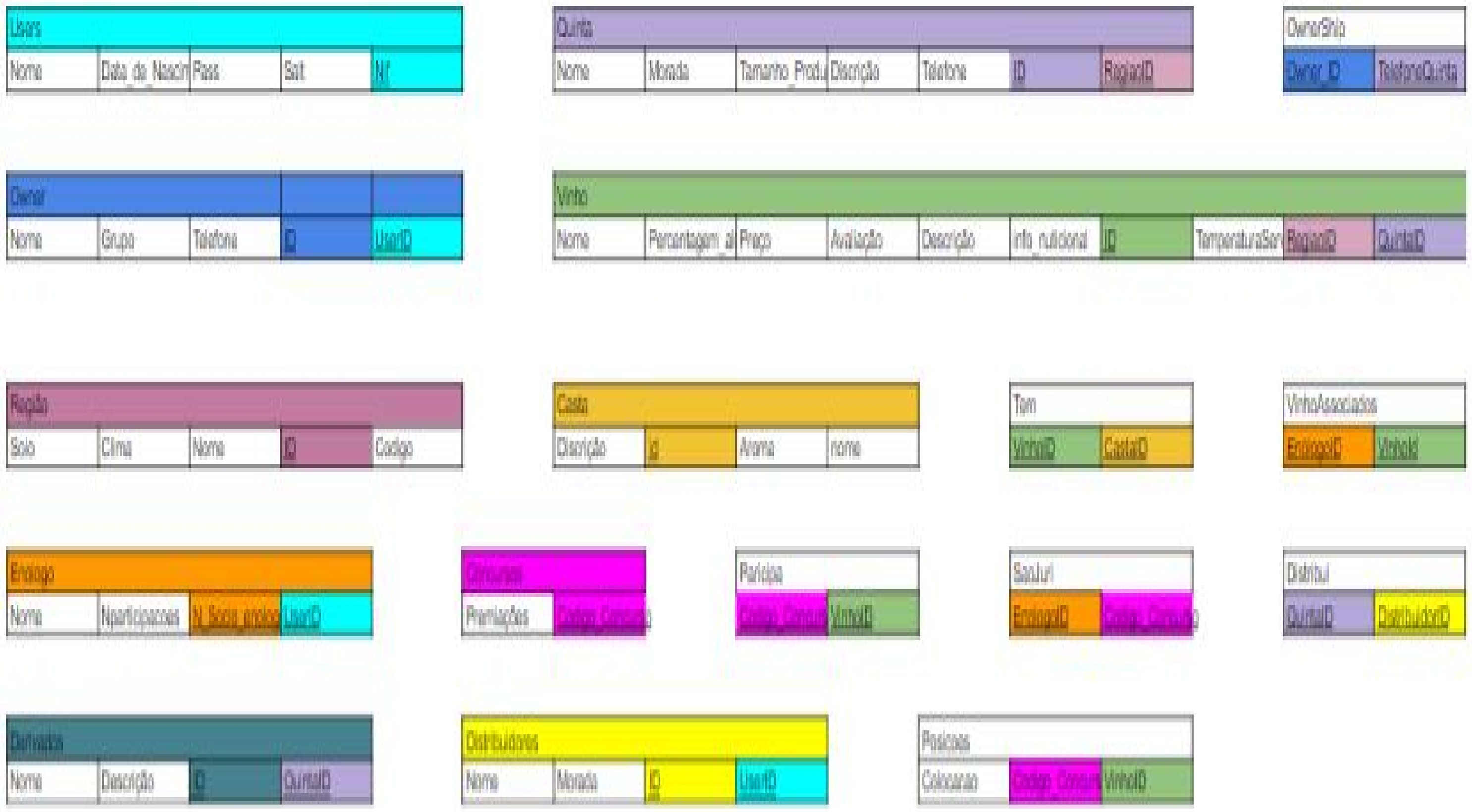
DER

After the realization and analysis of the requirements we did the diagram entity-relation of our system, this will be shown next.



ER

After the realization of the DER, we did the relation model, this was built maintaining the rules of the diagram entity-relation, where each entity and relation will give “birth” to a table. With the normalization, we didn’t think that was necessary because all the data is depending on one primary key, and his attributes are all necessary for the table in question.
We implemented a color system to better seeing of our relation model, where each table has a color associated and his relations have that color.


Users
Nome | Data_de_Nascim | Pass | Salt | Nif
Quinta
Nome | Morada | Tamanho_Produ | Discrição | Telefone | ID | RegiaoID
OwnerShip
Owner_ID | TelefoneQuinta
Owner
Nome | Grupo | Telefone | ID | UserID
Vinho
Nome | Percentagem_al | Preço | Avaliação | Descrição | Info_nutricional | ID | TemperaturaSer | RegiaoID | QuintaID
Região
Solo | Clima | Nome | ID | Codigo
Casta
Discrição | id | Aroma | nome
Tem
VinhoID | CastaID
VinhoAssociados
EnologoID | Vinhoid
Enologo
Nome | Nparticipacoes | N_Socio_enologo | UserID
Concursos
Premiações | Codigo_Concurso
Participa
Codigo_Concurso | VinhoID
SaoJuri
EnologoID | Codigo_Concurso
Distribui
QuintaID | DistribuidorID
Derivados
Nome | Descrição | ID | QuintaID
Distribuidores
Nome | Morada | ID | UserID
Posicoes
Colocacao | Codigo_Concurso | VinhoID

T-SQL DDL

After the relation diagram, we proceed to do with the creation of the tables, the next examples demonstrate the creation of the table of Oenologist, wine associated, and wines.

```
create table Vinhos.Vinho(
   Nome                 validname,
   PercentagemAlcool      validint,
   Preco                decimal(5,2),
   Avaliacao             int,
   Descricao             VARCHAR(500),
   InfoNutricional        VARCHAR(200),
   ID                   validint IDENTITY(1,1),
   TemperaturaServir      varchar(7),
   RegiaoID              validint,
   QuintaID              validint,
   primary key(ID),
   foreign key(RegiaoID) references Vinhos.Regiao(ID) on DELETE CASCADE,
   foreign key(QuintaID) references Vinhos.Quinta(ID) ,
);

create table Vinhos.Enologos(
   Nome                 validname,
   Nparticipacoes         int,
   NSocioEnologo          validint,
   UserId               varchar(12) not null,
   primary key (NSocioEnologo),
   foreign key (UserId) references  Vinhos.Users(NIF)  on delete cascade,
);
```

```
create table Vinhos.VinhosAssociados(
   Enologo_ID        validint,
   Vinho_ID          validint,
    foreign key (Enologo_ID) references Vinhos.Enologos(NSocioEnologo)  on
delete cascade,
   foreign key (Vinho_ID) references Vinhos.Vinho(ID)  on delete cascade,
);
```

The same logic to the other tables.

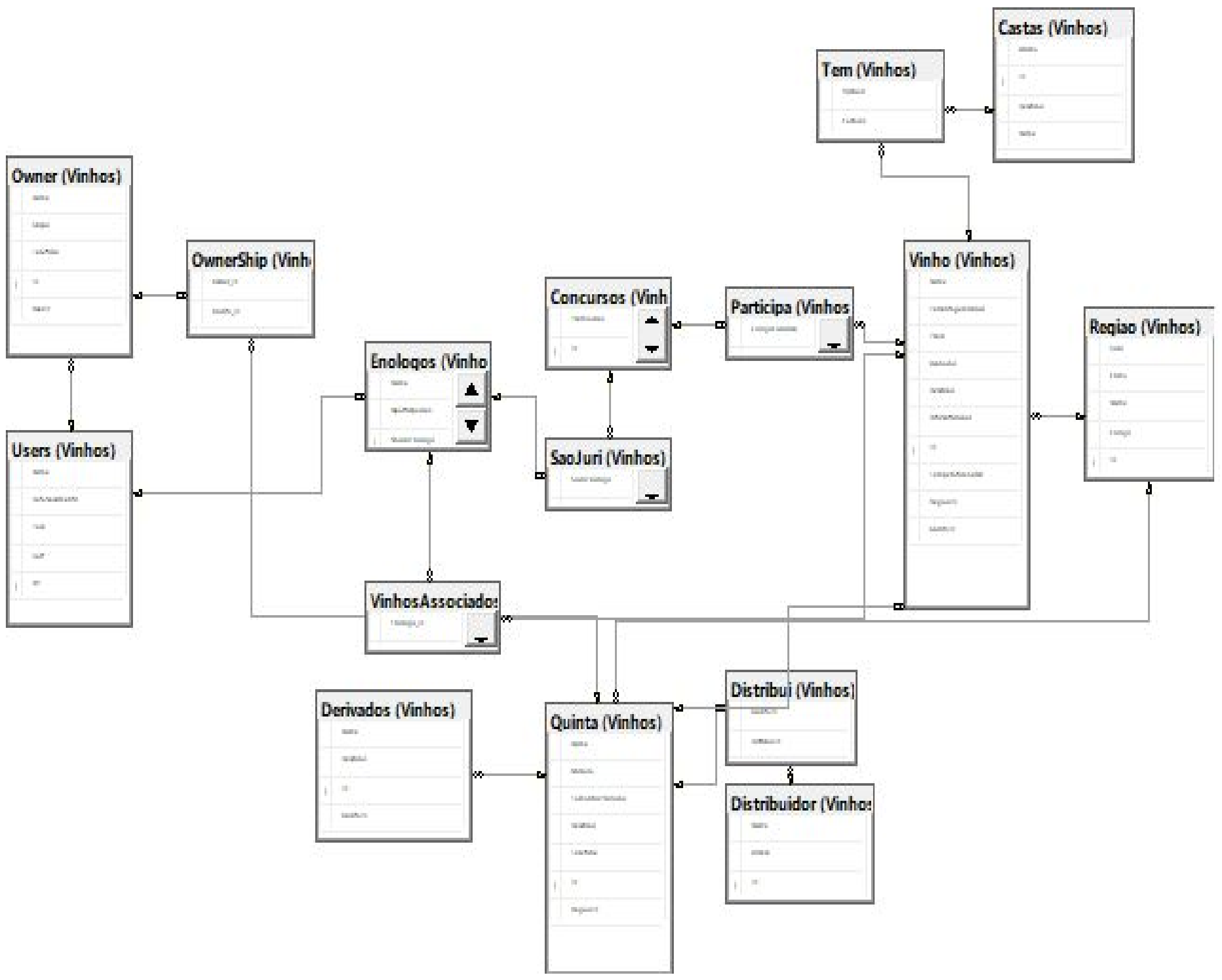
SQL DML

```
insert into Vinhos.Owner values
  ('Pedro Barreiro' ,  'Quinta da Barreira-Vitivinicultura',  '21757401','50240197'),
  ('João Silva e Sousa' ,  'Lua Cheia em Vinhas Velhas',  '21444455','50114523'),
  ('Anselmo Mendes' ,  'Anselmo Mendes Vinhos',  '21444466','10022021'),
  ('Francisco Marques Leandro' ,  'Casa Santa Eulália',  '21444477','10022033'),
  ('Rui Walter Cunha' ,  'Quinta de Paços',  '21444488','10022044'),
  ('Hélder Cunha' ,  'Casca Wines',  '21444499','10022055');


insert into Vinhos.Enologos VALUES
  ('José Gaspar              ',   '5',  '11111',   '12345678'),
  ('João Silva e Sousa       ',   '3',  '11112',   '50114523'),
  ('Francisco Baptista       ',   '6',  '11113',   '11235204'),
  ('Patrícia Peixoto         ',   '2',  '11114',   '13453212'),
  ('Anselmo Mendes           ',   '12',  '11115',   '10022021'),
  ('Francisco Marques Leandro',   '7',  '11116',   '10022033'),
  ('Rui Walter Cunha         ',   '15',  '11117',   '10022044'),
  ('Ana Rola               ',   '12',  '11118',   '12341421'),
  ('Hélder Cunha            ',   '4',  '11119',   '10022055'),
  ('Manuel Vieira           ',   '3',  '11120',   '12465123'),
  ('Luís Cabral de Almeida   ',   '5',  '11121',   '32132132');

insert into Vinhos.Regiao values
     ('normal','quente','Reg. Lisboa','2565-136'),
     ('rochoso','ameno','DOC Douro','5110-214 '),
     ('argiloso','quente','Alentejo','7900-573'),
     ('normal','frio','Monção e Melgaço','4950-483'),
     ('rochoso','ameno','Regional Minho','4850-481');
```

Database Diagram



Store Procedures

- **Vinhos.ConcursoPremios ->** get all prizes for a contest
- **Vinhos.InsertCasta ->** insert caste
- **Vinhos.InsertTem ->** insert relation between caste and wine
- **Vinhos.CastaRelationTem ->** insert casta and relation with that wine
- **Vinhos.DeleteCastaWineAssociation ->** delete association between casteand wine
- **Vinhos.InsertRegiao ->** insert a region
- **Vinhos.InsertQuinta ->** insert a farm
- **Vinhos.InsertEnologos ->** insert a new Oenologist
- **Vinhos.InsertVinho ->** insert a new wine
- **Vinhos.UpdateVinho ->** update that wine
- **Vinhos.UpdateRegiao ->** update that region
- **Vinhos.UpdateQuinta ->** update that farm
- **Vinhos.GetDis** -> get all info from distributors
- **Vinhos.GetDistQuinta** -> get distrubutor from a farm

Example of one store procedure for insert relation between caste and a determined wine.

```
go
create procedure Vinhos.CastaRelationTem(@VinhoID int,@Aroma
varchar(60), @Descricao varchar(800), @Nome varchar(60))
as
   declare @CastaID int;
   BEGIN
   insert into Vinhos.Castas(Aroma, Descricao, Nome) values(@Aroma ,
@Descricao, @Nome)
   Set @CastaID=( SELECT max(Vinhos.Castas.ID) FROM Vinhos.Castas);
   insert into Vinhos.Tem(VinhoId, CastasID) values(@VinhoID , @CastaID)
```

END

go

In this procedure first we create a new Casta, and then we search for the last ID of casta for then create the relation between wine and thar new casta. We search for the last ID searching for the max ID. The other ones are basically the same and we don't think that we need to mention it.

UDF’S

- **Vinhos.EnologoAssociatioWine ->** get all oenologist associated with determinated wine
- **Vinhos.EnologoNames ->** get all oenologist names
- **Vinhos.ConcursoName ->** get all contest names
- **Vinhos.ALLCastas ->** get all castas
- **Vinhos.CastasAssociadaVinho ->** get all castes associated with determinated wine
- **Vinhos.Enologo ->** Get information of a determinated oenologist
- **Vinhos.OwnerInfo ->** Get owner information
- **Vinhos.RegiaoInfo ->** get information of region
- **Vinhos.RegiaoName ->** get region names
- **Vinhos.QuintaName ->** get farm name
- **Vinhos.QuintaInfo ->** get farm information
- **Vinhos.getQuintaByReg ->** get all farms from a region
- **Vinhos.QuintasName ->** get a farm name
- **Vinhos.getRandomQuinta ->** get random farm
- **Vinhos.CountQuintas ->** get number of farms
- **Vinhos.getRandom ->** get random wine
- **Vinhos.CountWines ->** get number of wines
- **Vinhos.WineCaracteristics ->** get all information of the wine

- **Vinhos.WineName ->** get all names of wine

Example of one UDF has the objective to get all information about a farm, we had to get all relations between farm, owner, and region to get all this info.

```
create function Vinhos.QuintaInfo(@ID int) returns table
as
                    return(select Vinhos.Quinta.*,Vinhos.Regiao.Nome as
RegiaoNome,Vinhos.Owner.Nome as OnwerNome,Vinhos.Owner.ID as OwnerID
from ((Vinhos.Quinta join Vinhos.Regiao on
Vinhos.Quinta.RegiaoID=Vinhos.Regiao.ID) join Vinhos.OwnerShip on
Vinhos.Quinta.ID = Vinhos.OwnerShip.Quinta_ID)
join Vinhos.Owner on Vinhos.OwnerShip.Owner_ID = Vinhos.Owner.ID
where Vinhos.Quinta.ID=@ID);
go
```

Example, where we get, all the information about one determined wine an had to search for information in other tables.

```
create function Vinhos.WineCaracteristics( @ID int) returns table
as
          return(Select Vinhos.Vinho.*,Vinhos.Quinta.Nome as QuintaNome,
Vinhos.Regiao.Nome as RegiaoNome from (Vinhos.Vinho join Vinhos.Quinta on
Vinhos.Vinho.QuintaID = Vinhos.Quinta.ID) join Vinhos.Regiao on
Vinhos.Vinho.RegiaoID = Vinhos.Regiao.ID  where Vinhos.Vinho.ID = @ID) ;
Go
```

View

We didn’t use view because we didn’t any need for this type of function in our database.

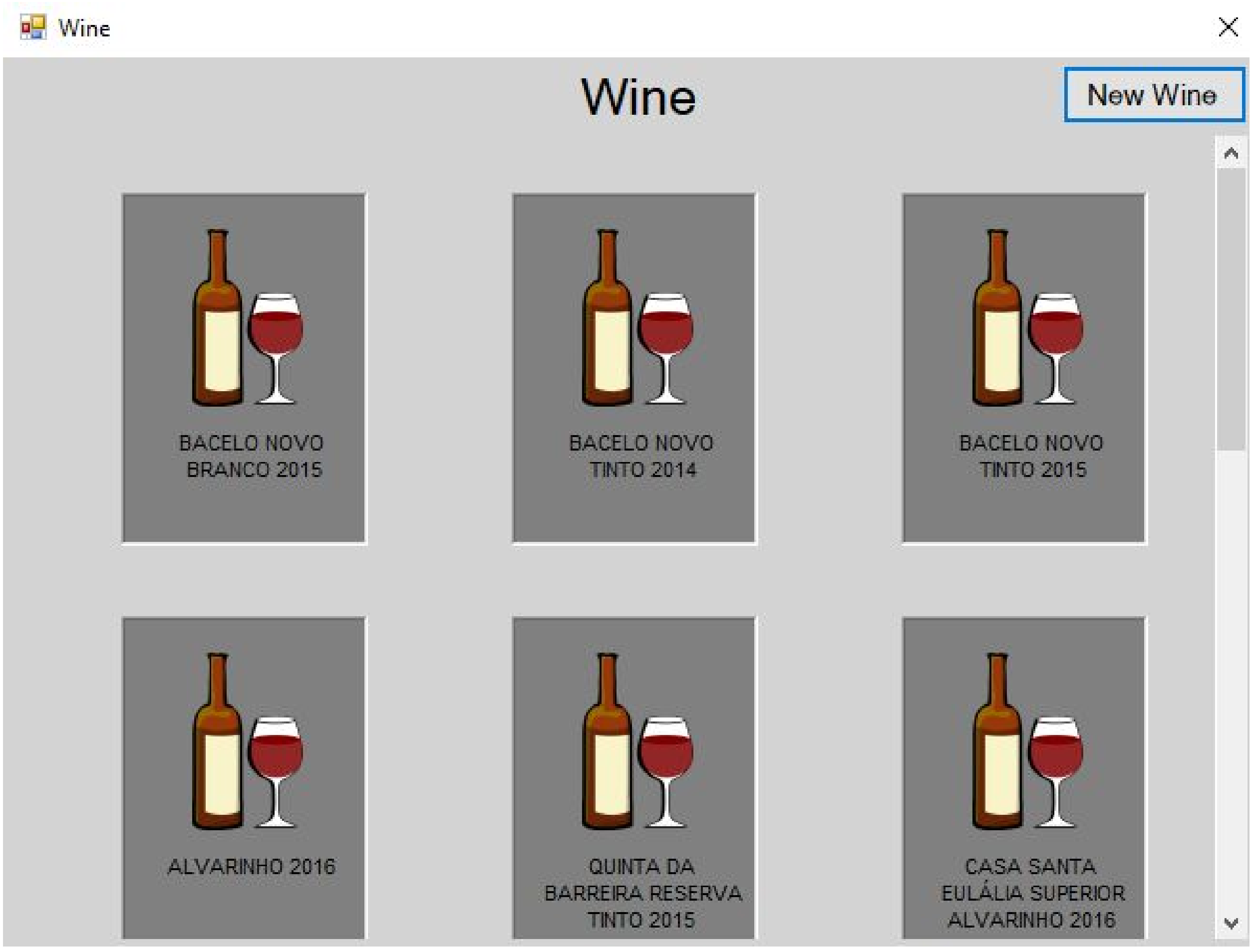
Triggers

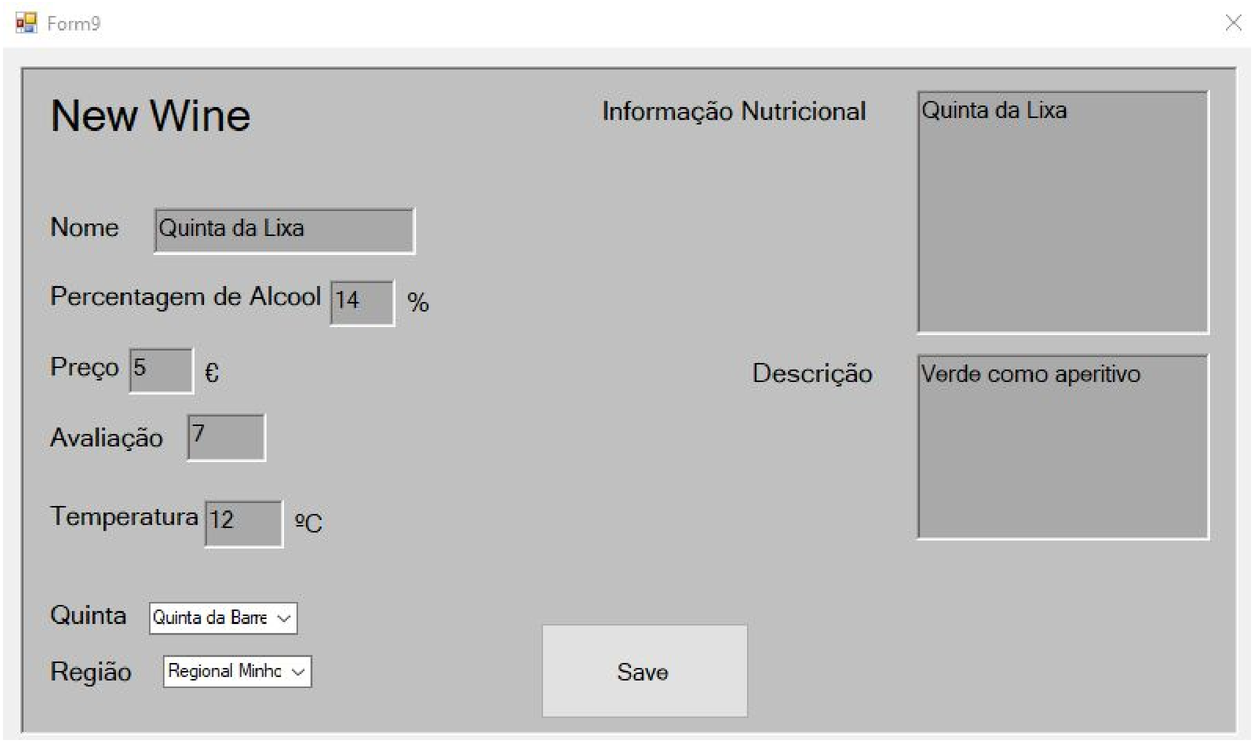
We used triggers for keep control in the updates and inserts in our database
One example is where we had a relation between wine and casta we have to see if that relation already exists we don't have to the table.

```
CREATE TRIGGER Vinhos.ValidateTem ON Vinhos.Tem
INSTEAD OF UPDATE,INSERT
as
begin
   Select *
   Into   #Temp
   From   inserted
   WHILE (SELECT count(*) FROM #Temp) >0
   begin
      DECLARE @CastaID as int;
      DECLARE @VinhoID as int;
      SELECT TOP 1 @CastaID = CastasID FROM #Temp;
      Select top 1 @VinhoID = VinhoID FROM #Temp
                              if (SELECT count(*) FROM Vinhos.Tem WHERE
Vinhos.Tem.CastasID=@CastaID and Vinhos.Tem.VinhoId=@VinhoID) > 0
      begin
         DELETE #TEMP WHERE CastasID=@CastaID and VinhoId=@VinhoID;
         RAISERROR('We already have that realtion', 16, 1);
      end
      ELSE
      begin
          insert into Vinhos.Tem SELECT * FROM #TEMP WHERE CastasID=@CastaID
and VinhoId=@VinhoID;
         DELETE #TEMP WHERE CastasID=@CastaID and VinhoId=@VinhoID;
```

```
    end
end
```

Graphic Interface




Form9

New Wine

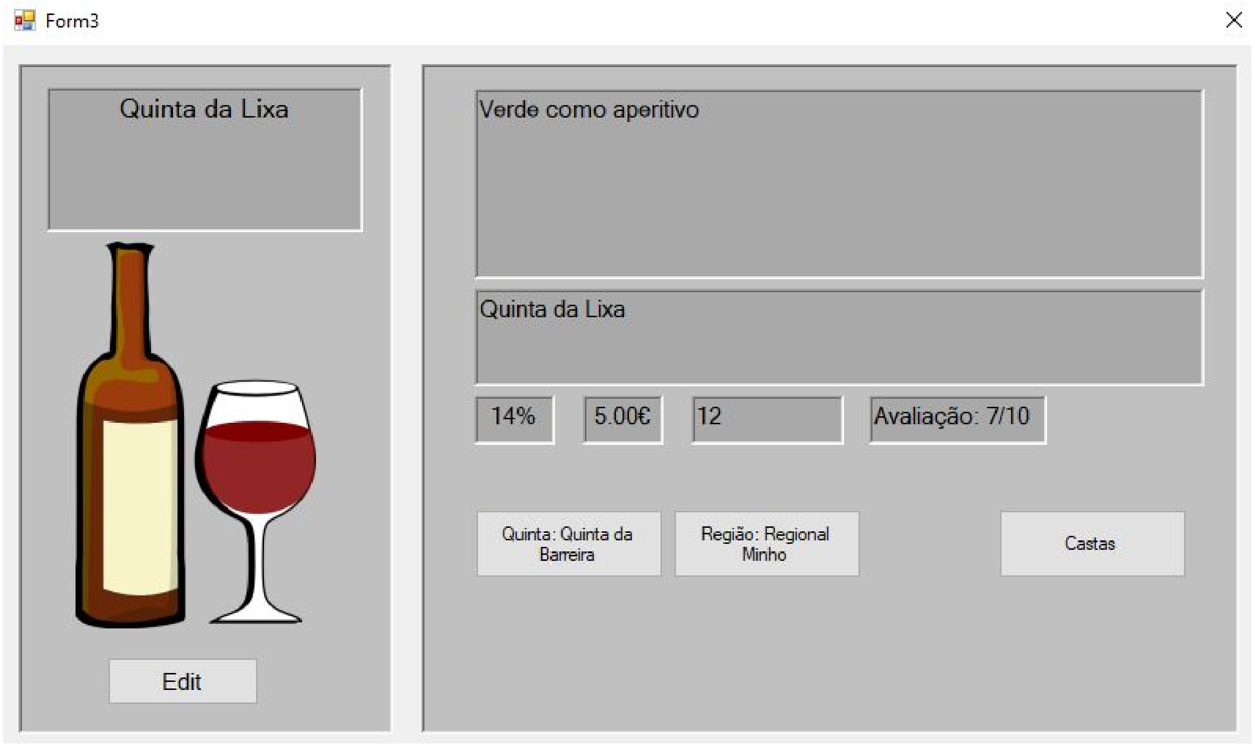
Informação Nutricional

Nome	Quinta da Lixa
Percentagem de Alcool	14 %
Preço	5 €
Avaliação	7
Temperatura	12 ºC
Quinta	Quinta da Barre
Região	Regional Minho

Quinta da Lixa

Descrição

Verde como aperitivo


Form3

Quinta da Lixa

Verde como aperitivo

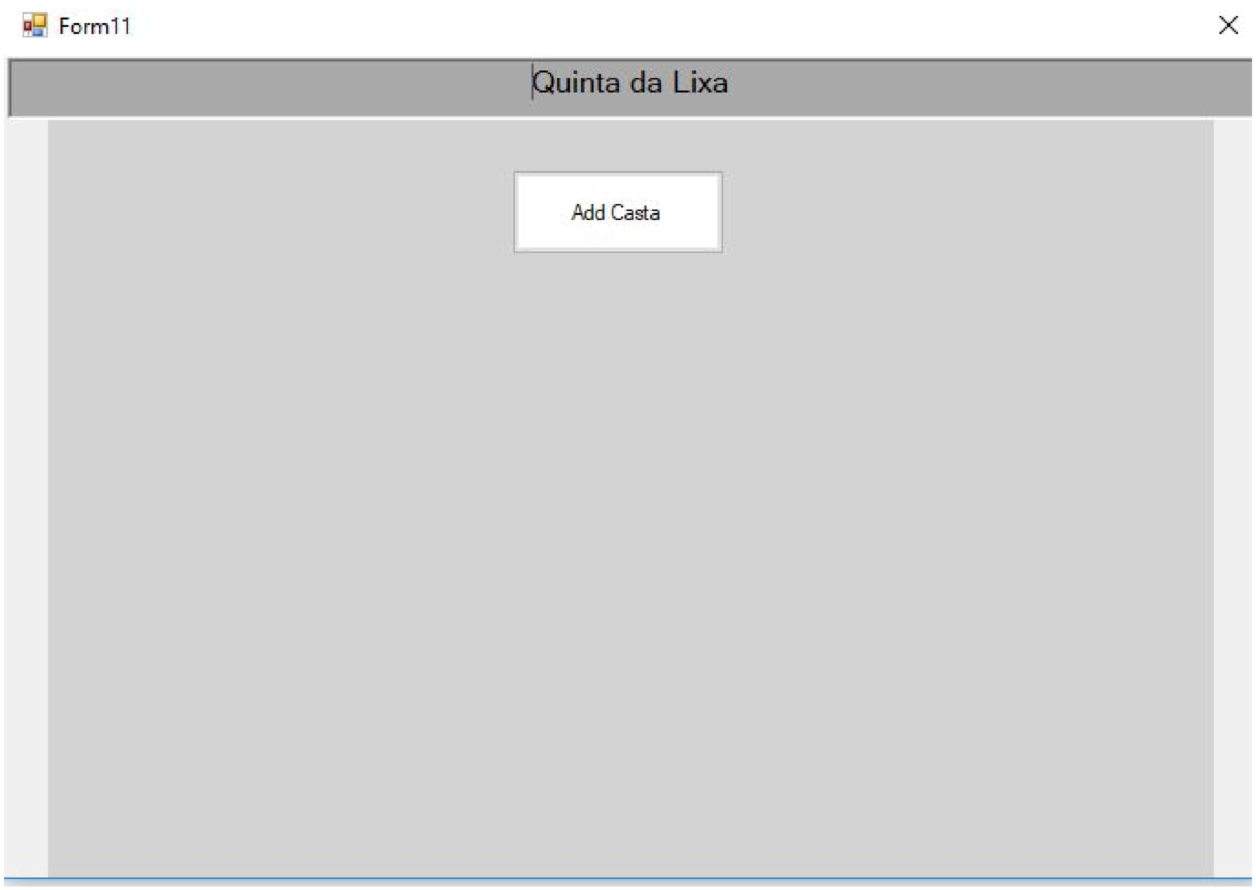
Quinta da Lixa

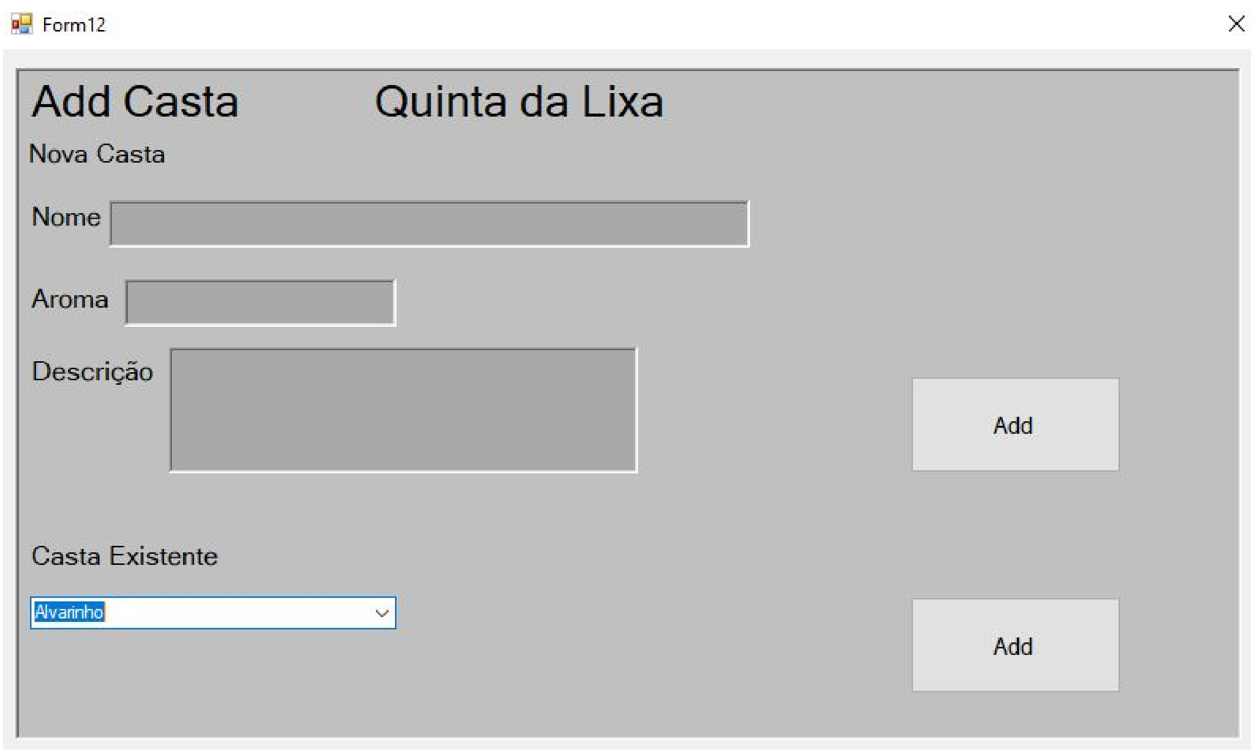
14%

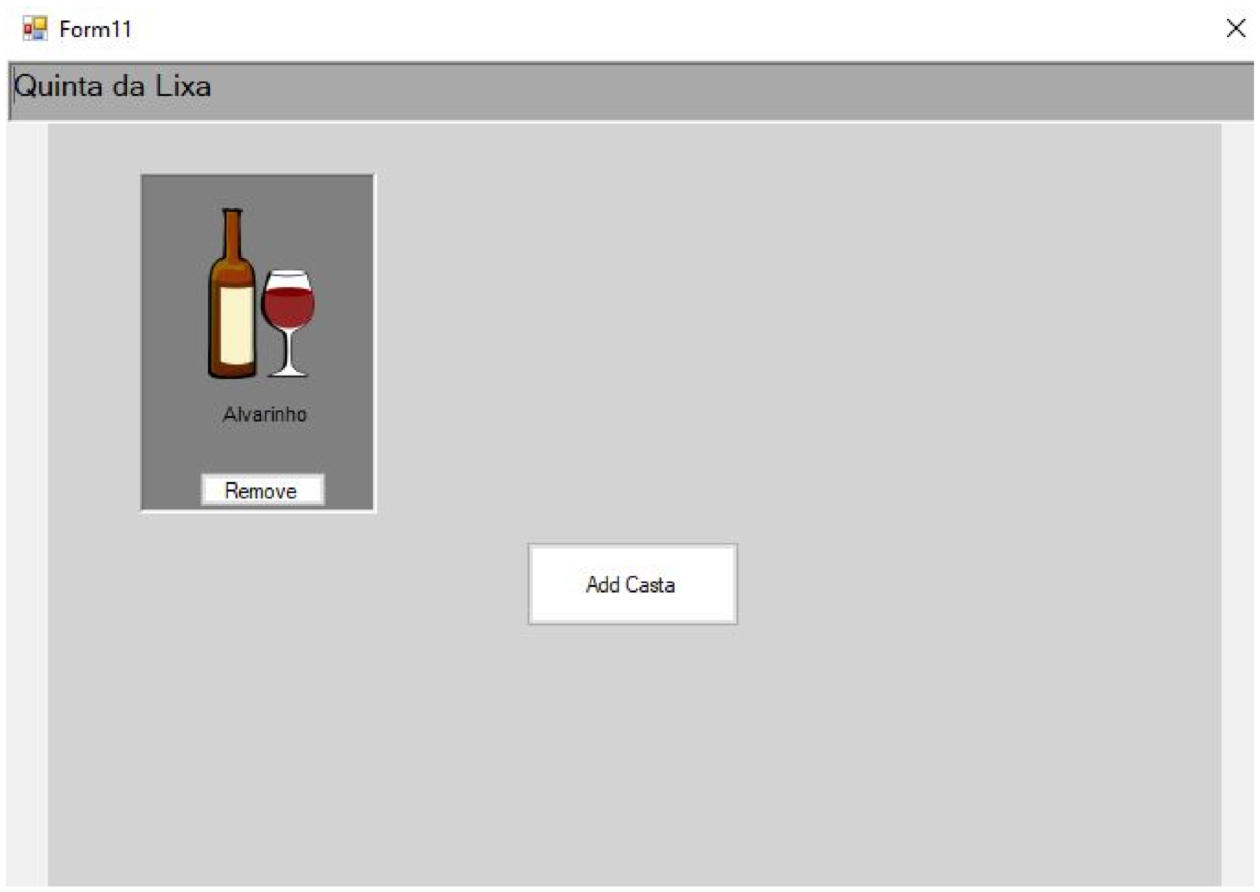
5.00€

12

Avaliação: 7/10


Form11
Quinta da Lixa
Add Casta


Form12
Add Casta
Quinta da Lixa
Nova Casta
Nome
Aroma
Descrição
Add
Casta Existente
Alvarinho
Add



An example where we had a new wine, and his casta, how is done in the interface.

Video

The video is located in our drive, is public and can be accessed by this link

Conclusão

To conclude, we think that we could do some udf's and some store procedures could be more complex. But we think that all the functionalities were all done, the interface is all functional. This work was a very useful work for our development as students, and to put the concepts acquired in the database class.

Attachments

Attached is a folder with all the files related to this project.