Seguradora de Saúde

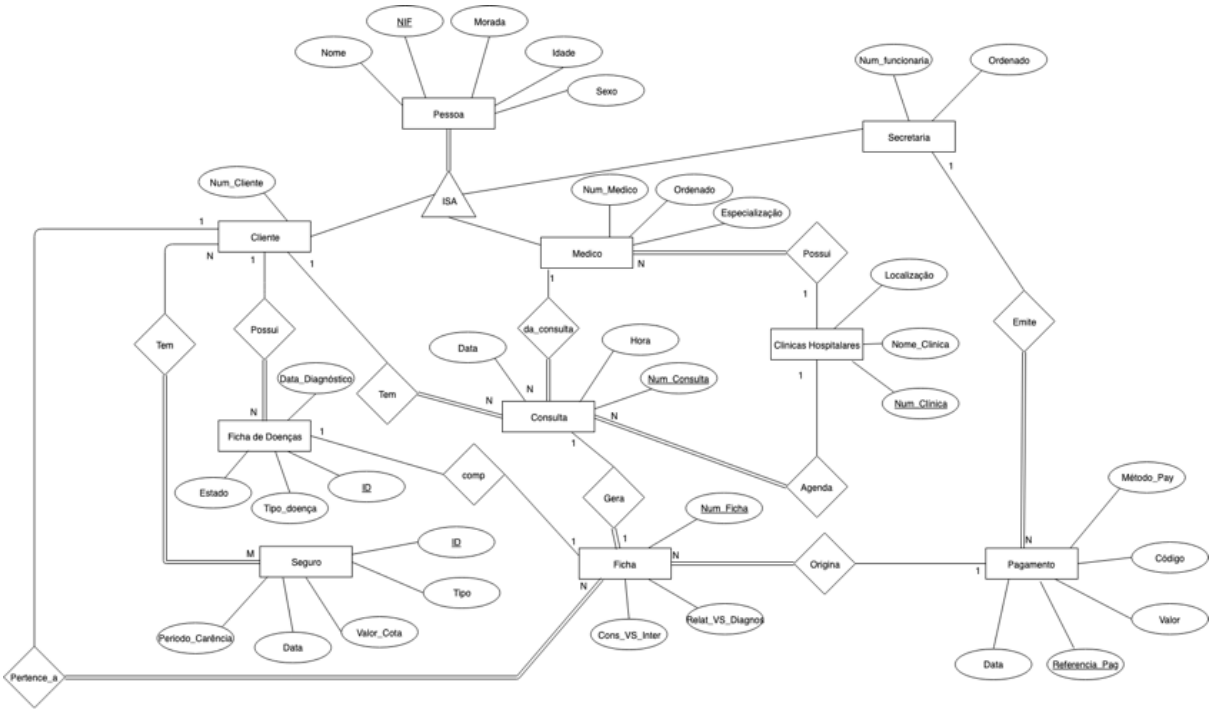
Análise de Requisitos

Considerando uma Seguradora Saúde com vários clientes e protocolos com outras entidades é pretendido fazer um controlo geral do seu funcionamento, tendo como principais algumas destas características:

- A empresa possui clientes (que mensalmente pagam uma cota) assim como tem uma série de protocolos com diversas clínicas hospitalares (que internam ou consultam os seus clientes sempre que necessário).
- Existem três tipos de pessoas: Clientes, Médicos e Secretárias.
- Um cliente é caracterizado pelas suas informações pessoais (Nome, NIF, Morada, Idade, Sexo) e por uma uma ficha de doenças já detetadas (Número de identificação da ficha, tipo da doença e Estado [se já está tratada ou não]).
- Cada médico só pode pertencer a uma clínica hospitalar e um médico é caracterizado pelo respetivo número identificador, especialização e ordenado enquanto que uma clínica hospitalar é caracterizada pelo respetivo nome, número identificador e localização.
- A atribuição de um seguro só será possível se essa doença não constar na ficha de doenças já detetadas.
- Um seguro é caracterizado pelo tipo de seguro e pelo valor da cota mensal.
- Quando um cliente da empresa necessita de uma consulta terá que se deslocar a uma clínica hospitalar com que a empresa tem protocolo.
- Se o médico que o consultar verificar que o cliente não necessita de qualquer internamento ou operação, passa-lhe uma ficha de consulta, onde especifica o número de cliente, nome e diagnóstico.
- O cliente terá então que enviar para a empresa essa ficha que quando lá chegar irá ser arquivada pela secretária a qual também emite uma via de pagamento.
- Se o médico que o atende verifica que o doente necessita de uma intervenção cirúrgica passa-lhe uma ficha de internamento e cirurgia onde consta um relatório.
- A secretária regista a ficha de internamento e cirurgia e posteriormente (a secretária) tendo como base essa ficha e o historial do cliente, preenche uma via de pedido de pagamento.
- O pagamento é caracterizado pela referência de pagamento, método pagamento (Numerário, Multibanco, etc.), Código da transação, Valor e Data.

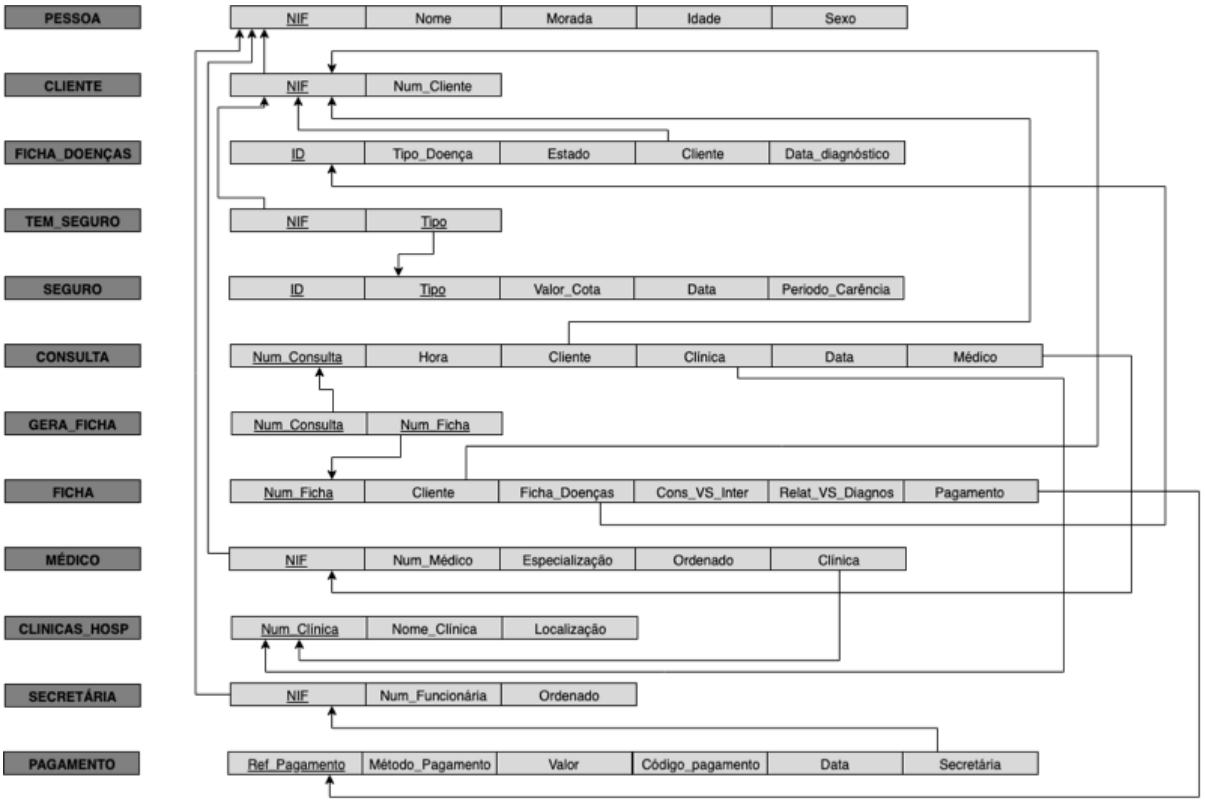
Seguradora de Saúde

Diagrama Entidade-Relação (DER)



Seguradora de Saúde

Esquema Relacional (ER)



Seguradora de Saúde

Estrutura SQL DDL

```
CREATE SCHEMA SeguradoraSaude

GO
CREATE TABLE SeguradoraSaude.Pessoa
(
   Nome    VARCHAR(50) NOT NULL,
   NIF     INT NOT NULL UNIQUE CHECK (NIF >= 000000000 AND NIF <= 999999999),
   Morada  VARCHAR(50),
   Idade   INT CHECK (Idade > 0),
   Sexo    CHAR(1),
   PRIMARY KEY (NIF)
)

GO
CREATE TABLE SeguradoraSaude.Cliente
(
   NIFCliente  INT NOT NULL UNIQUE CHECK (NIFCliente >= 000000000 AND NIFCliente <= 999999999),
   NumCliente  INT IDENTITY(1,1),
   PRIMARY KEY (NIFCliente),
   FOREIGN KEY (NIFCliente) REFERENCES SeguradoraSaude.Pessoa(NIF) ON DELETE CASCADE
)

GO
CREATE TABLE SeguradoraSaude.ClinicaHospitalar
(
   NumClinica      INT IDENTITY(1,1),
   NomeClinica     NVARCHAR(50) NOT NULL,
   Localizacao     VARCHAR(50),
   PRIMARY KEY (NumClinica)
)

GO
CREATE TABLE SeguradoraSaude.Medico
(
   NIFMedico       INT NOT NULL UNIQUE CHECK (NIFMedico >= 000000000 AND NIFMedico <= 999999999),
   NumMedico       INT IDENTITY(1,1),
   Ordenado        INT CHECK (Ordenado > 600),
   Especializacao  VARCHAR(30) NOT NULL,
   NumClinica      INT NOT NULL,
   PRIMARY KEY (NIFMedico),
   FOREIGN KEY (NIFMedico) REFERENCES SeguradoraSaude.Pessoa(NIF) ON DELETE CASCADE,
   FOREIGN KEY(NumClinica) REFERENCES SeguradoraSaude.ClinicaHospitalar(NumClinica) ON DELETE
CASCADE
)
```

Seguradora de Saúde

```
GO
CREATE TABLE SeguradoraSaude.Secretaria
(
   NIFSecretaria   INT NOT NULL UNIQUE CHECK (NIFSecretaria >= 000000000 AND NIFSecretaria <=
999999999),
   NumFuncionaria  INT IDENTITY(1,1),
   Ordenado INT CHECK (Ordenado > 600),
   PRIMARY KEY (NIFSecretaria),
   FOREIGN KEY (NIFSecretaria) REFERENCES SeguradoraSaude.Pessoa(NIF) ON DELETE CASCADE
)

GO
CREATE TABLE SeguradoraSaude.Consulta
(
   NumConsulta     INT IDENTITY(1,1),
   dataConsulta    DATE NOT NULL,
   hora            TIME NOT NULL,
   NIFCliente      INT NOT NULL CHECK (NIFCliente >= 000000000 AND NIFCliente <= 999999999),
   NIFMedico       INT NOT NULL CHECK (NIFMedico >= 000000000 AND NIFMedico <= 999999999),
   NumClinica      INT NOT NULL CHECK (NumClinica > 0),
   PRIMARY KEY (NumConsulta),
   FOREIGN KEY(NIFCliente) REFERENCES SeguradoraSaude.Cliente(NIFCliente),
   FOREIGN KEY(NIFMedico) REFERENCES SeguradoraSaude.Medico(NIFMedico),
   FOREIGN KEY(NumClinica) REFERENCES SeguradoraSaude.ClinicaHospitalar(NumClinica)
)

GO
CREATE TABLE SeguradoraSaude.Pagamento
(
   RefPagamento    INT NOT NULL UNIQUE,
   MetodoPagamento VARCHAR(20),
   Codigo          INT NOT NULL UNIQUE CHECK (Codigo > 0),
   Valor           INT NOT NULL CHECK (Valor > 0),
   DataPagamento   DATE,
   NIFSecretaria   INT NOT NULL CHECK (NIFSecretaria >= 000000000 AND NIFSecretaria <= 999999999),
   PRIMARY KEY (RefPagamento),
   FOREIGN KEY (NIFSecretaria) REFERENCES SeguradoraSaude.Secretaria(NIFSecretaria) ON DELETE
CASCADE
)


GO
CREATE TABLE SeguradoraSaude.Ficha
(
   NumFicha                INT IDENTITY(1,1),
   RelatorioDiagnostico    CHAR(1) NOT NULL,
   ConsultaInternamento    CHAR(1) NOT NULL,
   NumConsulta             INT NOT NULL CHECK (NumConsulta > 0),
   NIFCliente              INT NOT NULL CHECK (NIFCliente >= 000000000 AND NIFCliente <=
999999999),
   RefPagamento            INT NOT NULL,
   PRIMARY KEY (NumFicha),
   FOREIGN KEY(NIFCliente) REFERENCES SeguradoraSaude.Cliente(NIFCliente),
   FOREIGN KEY (NumConsulta) REFERENCES SeguradoraSaude.Consulta(NumConsulta),
   FOREIGN KEY (RefPagamento) REFERENCES SeguradoraSaude.Pagamento(RefPagamento)
)
```

Seguradora de Saúde

```
GO
CREATE TABLE SeguradoraSaude.FichaDoencas
(
   ID              INT NOT NULL UNIQUE CHECK (ID > 0),
   NIFCliente      INT NOT NULL CHECK (NIFCliente >= 000000000 AND NIFCliente <=
999999999),
   TipoDoenca      VARCHAR(30) NOT NULL,
   Estado          BIT NOT NULL,
   DataDiagnostico DATE NOT NULL,
   NumFicha        INT NOT NULL UNIQUE CHECK (NumFicha > 0),
   PRIMARY KEY (ID),
   FOREIGN KEY(NIFCliente) REFERENCES SeguradoraSaude.Cliente(NIFCliente) ON DELETE
CASCADE,
   FOREIGN KEY(NumFicha) REFERENCES SeguradoraSaude.Ficha(NumFicha) ON DELETE CASCADE
)

GO
CREATE TABLE SeguradoraSaude.Seguro
(
   ID          INT NOT NULL UNIQUE CHECK (ID > 0),
   Tipo        VARCHAR(25) NOT NULL,
   Cota        INT NOT NULL CHECK (Cota > 0),
   Carencia    VARCHAR(30) NOT NULL,
   DataSeguro  DATE NOT NULL,
   PRIMARY KEY (ID)
)

GO
CREATE TABLE SeguradoraSaude.ClienteTemSeguro
(
   NIFCliente  INT NOT NULL UNIQUE CHECK (NIFCliente >= 000000000 AND NIFCliente <=
999999999),
   ID          INT NOT NULL UNIQUE CHECK (ID > 0),
   PRIMARY KEY (NIFCliente, ID),
   FOREIGN KEY (NIFCliente) REFERENCES SeguradoraSaude.Cliente(NIFCliente) ON DELETE
CASCADE,
   FOREIGN KEY (ID) REFERENCES SeguradoraSaude.Seguro(ID) ON DELETE CASCADE
)

GO
CREATE TABLE SeguradoraSaude.Login
(
   ID                      INT                NOT NULL,
   PessoaNIF               INT,
   username                VARCHAR(15)        NOT NULL,
   userpass                VARCHAR(32)        NOT NULL,
   type                    VARCHAR(15),
   PRIMARY KEY(ID),
   FOREIGN KEY(PessoaNIF) REFERENCES SeguradoraSaude.Pessoa(NIF)
)
```

Seguradora de Saúde

SQL DML

```
USE [SeguradoraSaude]

GO
INSERT INTO [SeguradoraSaude].[Pessoa] ([Nome],[NIF] ,[Morada] ,[Idade] ,[Sexo])
    VALUES ('Alexandre Lourenco','15174922','Rua Sao Tiago','22','M')
INSERT INTO [SeguradoraSaude].[Pessoa] ([Nome],[NIF] ,[Morada] ,[Idade] ,[Sexo])
    VALUES ('Manuel Augusto','11922355','Rua Tivoli', '61','M')
INSERT INTO [SeguradoraSaude].[Pessoa] ([Nome],[NIF] ,[Morada] ,[Idade] ,[Sexo])
    VALUES ('Ana Lopes','34762323','Avenida da Liberdade','26','F')

GO
INSERT INTO [SeguradoraSaude].[Cliente] ([NIFCliente])
    VALUES ('15174922')

GO
INSERT INTO [SeguradoraSaude].[ClinicaHospitalar] ([NomeClinica],[Localizacao])
    VALUES ('Clinica da Boa Saude','Porto')

GO
INSERT INTO [SeguradoraSaude].[Medico]
([NIFMedico],[Ordenado],[Especializacao],[NumClinica])
    VALUES ('11922355',2500,'Cardiologia',1)

GO
INSERT INTO [SeguradoraSaude].[Secretaria] ([NIFSecretaria],[Ordenado])
    VALUES ('34762323',700)

GO
INSERT INTO [SeguradoraSaude].[Consulta] ([dataConsulta],[hora],[NIFCliente],
[NIFMedico],[NumClinica])
    VALUES ('2019-05-02','19:00:00','15174922','11922355',1)

GO
INSERT INTO [SeguradoraSaude].[Pagamento]
([RefPagamento],[MetodoPagamento],[Codigo],[Valor],[DataPagamento],[NIFSecretaria])
    VALUES ('505243456','Numerario','1234','20','2019-05-3','34762323')

GO
INSERT INTO [SeguradoraSaude].[Ficha]
([RelatorioDiagnostico],[ConsultaInternamento],[NumConsulta],[NIFCliente],[RefPagamento]
)
    VALUES ('R','C','1','15174922','505243456')

GO
INSERT INTO [SeguradoraSaude].[FichaDoencas]
([ID],[NIFCliente],[TipoDoenca],[Estado],[DataDiagnostico],[NumFicha])
    VALUES ('1','15174922','Mononucleose','1','2018-06-6',1)
```

```
GO
INSERT INTO [SeguradoraSaude].[Seguro] ([ID],[Tipo],[Cota],[Carencia],[DataSeguro])
    VALUES ('1','Individual','25','12','2019-03-1')

GO
INSERT INTO [SeguradoraSaude].[ClienteTemSeguro] ([NIFCliente],[ID])
    VALUES ('15174922','1')
```

Normalização

Através do nosso diagrama de Entidade-Relação observamos que seria impossível normalizar ainda mais os nossos dados pois em todas as tabelas, todos os atributos dependem da chave primária e todas elas apenas contêm os atributos necessários para a tabela em questão, reduzindo assim a redundância e também a quantidade de atributos null.

Índices

```
-- Indexar Clientes pelo NIF (chave única)
CREATE UNIQUE INDEX IxCNIF ON SeguradoraSaude.Cliente(NIFCliente)

-- Indexar Médicos pelo NIF (chave única)
CREATE UNIQUE INDEX IxMNIF ON SeguradoraSaude.Medico(NIFMedico)

-- Indexar Secretarias pelo NIF (chave única)
CREATE UNIQUE INDEX IxSNIF ON SeguradoraSaude.Secretaria(NIFSecretaria)

-- Indexar Clínicas pelo número de clínica (chave única)
CREATE UNIQUE INDEX IxClinc ON SeguradoraSaude.ClinicaHospitalar(NumClinica)

-- Indexar Pagamentos pela respetiva referência de pagamento (chave única)
CREATE UNIQUE INDEX IxRefPay ON SeguradoraSaude.Pagamento(RefPagamento)

-- Indexar Seguros pelo número identificador (chave única)
CREATE UNIQUE INDEX IxIDS ON SeguradoraSaude.Seguro(ID)
```

Seguradora de Saúde

Stored Procedures

Stored Procedure	Descrição	Stored Procedure	Descrição
NewClient	Regista um novo cliente na base de dados	DeleteFile	Elimina uma ficha de cliente na base de dados
NewDoctor	Regista um novo médico na base de dados	DeleteDisease	Elimina uma ficha de doenças na base de dados
NewSecretary	Regista uma nova secretaria na base de dados	DeletePayment	Elimina um pagamento na base de dados
NewAppointment	Regista uma nova consulta na base de dados	DeleteInsurence	Elimina um seguro na base de dados
NewInsurance	Regista um novo seguro na base de dados	DeleteClinic	Elimina uma clínica na base de dados
NewDiseasesFile	Regista uma nova ficha de doenças na base de dados	GetUserList	Retorna todos os utilizadores registados na base de dados
NewClinic	Regista uma nova clínica na base de dados	GetClientList	Retorna todos os clientes da base de dados
NewFile	Regista uma nova ficha de cliente na base de dados	GetDoctorList	Retorna todos os médicos da base de dados
NewPayment	Regista um novo pagamento na base de dados	GetSecretaryList	Retorna todas as secretárias da base de dados
UpdateUserInfo	Atualiza os dados de um cliente/médico/secretária na base de dados	GetClinicList	Retorna todas as clínicas da base de dados
UpdateClinic	Atualiza os dados de uma clínica na base de dados	GetAppointmentList	Retorna todas as consultas da base de dados
UpdateAppointment	Atualiza os dados de uma consulta na base de dados	GetInsurenceList	Retorna todos os seguros da base de dados
UpdateFile	Atualiza os dados de uma ficha de cliente na base de dados	GetFileList	Retorna todas as fichas de cliente da base de dados
UpdateInsurence	Atualiza os dados de um seguro na base de dados	GetDiseasesList	Retorna todas as fichas de doenças da base de dados
UpdateDiseaseInfo	Atualiza os dados de uma ficha de doenças na base de dados	GetPaymentList	Retorna todos os pagamentos da base de dados
DeleteUser	Elimina um cliente/médico/secretária na base de dados	GetClientNIF	Retorna a informação do cliente (caso exista na BD) de acordo com o NIF fornecido

Seguradora de Saúde

DeleteAppointment	Elimina uma consulta na base de dados	GetDoctorNIF	Retorna a informação do médico (caso exista na BD) de acordo com o NIF fornecido
GetSecretaryNIF	Retorna a informação da secretária (caso exista na BD) de acordo com o NIF fornecido	GetDiseaseInfoNIF	Retorna a informação da ficha de doenças (caso exista na BD) de acordo com o NIF do cliente
GetClinicNum	Retorna a informação da clínica (caso exista na BD) de acordo com o número de clínica fornecido	GetAppointmentInfoNIF	Retorna a informação da consulta (caso exista na BD) de acordo com o NIF do cliente que a agendou
GetAppointNum	Retorna a informação da consulta (caso exista na BD) de acordo com o número de consulta fornecido	GetInsurenceInfoNIF	Retorna a informação de um seguro (caso exista na BD) de acordo com o NIF do cliente ao qual pertence
GetFileNum	Retorna a informação da ficha de cliente (caso exista na BD) de acordo com o número de ficha fornecido	GetPaymentInfoNIF	Retorna as informações de pagamento (caso existam na BD) de um cliente acordo com o NIF do cliente
GetDiseaseID	Retorna a informação da ficha de doenças (caso exista na BD) de acordo com o número de ficha de doenças fornecido	GetPaymentInfoSec	Retorna as informações de pagamentos processados (caso existam na BD) por uma secretária acordo com o NIF da mesma
GetInsurenceID	Retorna a informação do seguro (caso exista na BD) de acordo com o id de seguro fornecido	GetDoctorClinicNF	Retorna as informações da clínica onde um médico trabalha (caso exista na BD) quando dado o número do médico
GetPayRef	Retorna a informação do pagamento (caso exista na BD) de acordo com a referência de pagamento fornecida	GetClinic	Retorna o nome de uma clínica (caso exista na BD) quando dado o número da mesma
GetPersonType	Retorna o tipo (cliente, médico ou secretária) de um utilizador para ser usado no login	GetLoginType	Retorna o tipo de login
GetPersonID	Retorna o username de um determinado utilizador	GetLogin	Retorna o username e a password de um determinado utilizador

*Muitos dos procedures criados permitem criação de outras entidades (que não a que estamos no momento a criar) caso não existam (p.e. Ao criar um médico novo definimos que este trabalha na clínica 2. Mesmo que essa clínica ainda não esteja criada é possível criá-la aquando da criação do médico inserindo apenas os dados referentes a ela)

Seguradora de Saúde

Triggers

TrgInsurenceCover	Permite verificar se o seguro do cliente cobre os custos do mesmo ou se o seguro ainda se encontra em período de carência, sendo que neste último caso será o cliente a suportá-los

UDF'S

GetClientInfo	Retorna toda a informação de um dado cliente
GetDoctorInfo	Retorna toda a informação de um dado médico
GetSecretaryInfo	Retorna toda a informação de uma dada secretária

Conclusão

Neste projeto tentamos ao máximo proteger a integridade dos dados da base de dados tendo para isso usado diversos store procedures e triggers de forma a não existir contacto direto entre a interface e a base de dados.

Fazendo um balanço da situação concordamos que existem pontos que podem sempre ser melhorados tal como a nossa interface, que em termos visuais pode não estar muito apelativa, contudo julgamos ter atingido tudo aquilo a que nos propusemos neste trabalho.