Solução de controlo de Reparações

Projeto global para a licenciatura de informática

Realizado por Rui Pedro da Cruz Machado
Numero al1729
Licenciatura em Informática
3º Ano, turma B
Lisboa, Ano letivo de 2012

Projeto desenvolvido para cumprimento dos requisitos necessários á obtenção da licenciatura informática.

Coordenador: Prof. Dr. Nuno Correia
Orientador Metodológico: Prof. Dr. Pedro Brandão
Orientador da Especialidade: Prof. Dr. Joaquim Aleixo

Dedicatória

Para os meus melhores amigos, Martim e Leonor…

- FALTA ESTADO DA ARTE
- INTRODUÇÃO INCORRECTA
- FALTA DE REFERÊNCIAS E CITAÇÕES.
- TEM PROBLEMAS DE LÍNGUA.
- SEM CONCLUSÃO.

→ TECNICAMENTE É MUITO BOM

→ NOTA DA EMPREGA E DOS UTILIZADORES SOBRE A USABILIDADE.

Agradecimentos

Ao meu coordenador técnico, Professor Doutor Joaquim Aleixo, que me ajudou a formular da melhor maneira este trabalho final, pela melhor orientação, conselhos, pela paciência e simpatia que sempre me dispensou.

Uma palavra de agradecimento ao meu professor universitário José Neves, foi a pessoa que mais me influenciou nas direções a tomar ao nível da área de desenvolvimento. Foi graças às suas aulas que me apercebi de várias lógicas de programação que hoje em dia utilizo com frequência. O seu método de ensino denominado por "saber-fazer" é, sem dúvida, dos mais andragógicos possíveis, tornando as aulas dinâmicas e participativas.

Á pessoa que me ensinou a ser o que sou hoje, a minha mãe.

Resumo

Em síntese, o projeto enquadra-se no âmbito do desenvolvimento aplicacional e de desenho e conceção de modelos de dados. É uma aplicação que tem como finalidade descrever o ciclo de vida de material de informática. É guardado na base de dados um histórico das operações realizadas sobre o dito material, juntamente com os dados dos respetivos clientes da empresa, seus titulares. Serve também de gestão de carteira de clientes com todos os dados adjacentes ao nível de perfil dos mesmos.

Palavras-chave

Palavras-chave: Base de dados; Desenvolvimento; Plataforma de trabalho.Net; Windows Entity; Windows Forms; Projeto; Relatório; Engenharia de *software*.

Abreviaturas

BD – Base de dados

.Net – Tecnologia da Microsoft

EF – *Entity Framework*

Winforms – *Windows Forms*

ADO – *ActiveX data objects*

EDM – Entity data model

SQL – *Structured query language*

CRM –Customer relationship management

RDBMS – Relational data base management system

Índice

Conteúdo

Introdução

Em todas as empresas que fornecem componentes eletrónicos, existe a necessidade de manter em registo os vários eventos e estados de cada um desses componentes,guardando-os em histórico, a fim de se controlar os seus ciclos de vida. O objetivo da aplicação proposta é centralizar toda a informação relativa aos eventos e estados dos componentes vendidos. Cada componente vendido vai estar associado a um perfil, e sempre que haja necessidade de executar reparações, serão associadas as datas de começo e fim de reparação, descrição de procedimentos para a reparação e dados de envio.

Ao usar a aplicação, um utilizador poderá obter uma lista de todos os clientes da empresa e consultar o material adquirido de cada um, poderá ver também o histórico de reparações associado a cada componente.

Utilizadores autenticados poderão, mediante direitos, criar perfis, associar componentes de uma lista aos perfis e abrir/fechar obras sobre esses componentes. Alguns utilizadores com todos os direitos poderão também adicionar e gerir os direitos de outros utilizadores.

Este documento descreve a informação que foi recolhida sobre os processos de reparação de material eletrónico na empresa. Esta informação é para ser utilizada como guia no desenvolvimento da aplicação que irá manter um registo dos vários estados das reparações.

1- LINGUA PORTUGUESA
2- N INTRODUÇÃO

Requerimentos e Especificação

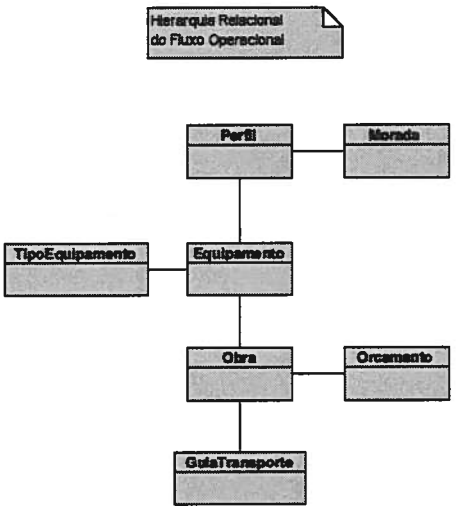
Análise de domínio

Glossário

- **Obra**: O processo de reparação de um dado componente ou conjunto de componentes.
  - **Abertura de Obra**: O momento em que é dada entrada de um componente para reparação.
  - **Fecho de Obra**: O momento em que se dá por concluída a reparação do componente.
  - **Saída de Obra**: O momento em que é emitido a guia de transporte e é enviado o equipamento para o proprietário.
  - **Orçamentos**: Documento a apresentar a um cliente com os valores da reparação.
  - **Guia de Transporte**: Documento legal necessário para envio de equipamentos.
- **Perfil**: Ficha com os dados de um cliente.
  - **Morada:**Possíveis moradas de um perfil. Morada com vários tipos, mas apenas uma é primária.
- **Tipo Equipamento:** A família em que os componentes se enquadram.
  - **Equipamento:**Nome e descrição de um dado componente.
  - ***Spare*:** Equipamento com as mesmas características mas pertencente á entidade reparadora. Tem a finalidade de substituir o equipamento do cliente enquanto este é reparado.

Conhecimento geral do domínio

- As reparações ocorrem durante o horário de expediente.
- Os equipamentos pertencem sempre a uma família de equipamentos.
- Uma obra pode ter um orçamento relacionado.
- Uma obra tem sempre um equipamento relacionado.
- Cada equipamento tem sempre um perfil relacionado.
- Uma obra pode ou não ter um equipamento de substituição relacionado.
- Por cada evento de uma obrasão notificados tantos recipientes de endereços eletrónicos quanto os configurados.
- Obras com o processo total de reparação concluído contam para a estatística do ciclo de vida dos equipamentos.
- Cada operação executada na aplicação ou contra a base de dados terá sempre um registo de utilizador associado.



Clientes e utilizadores

Potenciais clientes são pequenas e médias empresas de venda de material eletrónico ou informático e que mantenham contratos de manutenção com os seus clientes para avarias desse mesmo material.

Outras entidadesenvolvidas no sistema:

- Colaboradores da área de vendas, para consultar estatísticas e outras informações sobre os clientes ou o seu material.
- Colaboradores da área de reparações para trabalharem nas obras mediante necessidade.
- Um administrador de sistemas que, por norma, mantem o ambiente a nível de *hardware*, sistema operativo e rede.
- Um administrador de base de dados para garantir as cópias de segurança, segundo o plano de recuperação de desastres, e manusear os dados conforme necessário.
- Técnicos da área de sistemas que tipicamente instalarão a aplicação para ser usada pelos utilizadores.

Todos os colaboradores indicados terão acesso a um computador disponível para seu uso e também acesso á rede empresarial,a fim de comunicar com o servidor de base de dados.

Ambiente

Os atores terão um computador nas suas secretárias; é comum estes terem o sistema operativo Microsoft Windows e esta será a única plataforma suportada. A aplicação corre sobre a plataforma de trabalho da Microsoft. Net4

A aplicação estará instalada nos computadores e todos os atores terão a mesma configuração de acesso á base de dados a partir do cliente nativo do sistema operativo.

A base de dados estará disponível para consumo num servidor, dentro de uma rede local. Uma vez instalada a aplicação nos computadores e com o acesso remoto á rede em questão devidamente configurado, os dados também poderão ser acedidos a partir de outros locais.

Procedimentos e fluxos de trabalho executados atualmente

A chegada de material dá-se a qualquer hora, ficando pendente o reencaminhamento para o departamento de reparações.

O departamento de reparações recebe o material e dá início á obra. Até esta fase não se criou qualquer registo de informação. Executa a reparação ou cria um orçamento para enviar ao cliente caso seja necessária a substituição de componentes.

Após concordância pelo cliente em caso de orçamento ou de simples reparação, o material é transferido para o departamento de logística que trata de enviá-lo de volta para as instalações do cliente.

Todo o procedimento gera registos mínimos e independentes, e os poucos que gera são sem nexo, o que leva a uma ausência de controlo sobre os procedimentos efetuados na obra e sobre o ciclo de vida do material.

Aplicações competitivas

Existem no mercado várias ferramentas que executam os mesmos procedimentos.No entanto, como normalmente fazem parte de sistemas com mais funcionalidades (ex.: Gestão de relações de clientes ou dePlaneamento de recursos de empresas), tornam-se mais complexos de utilizar e como consequência não são adotados pela generalidade dos colaboradores.Ao

desenvolver a especificação pedida, consegue-se atingir uma aplicação á medida, abdicando assim dos demais subsistemas que todos estes produtos concorrentes oferecem.

Outra vantagem é a de ter a possibilidade de desenvolver a aplicação com as outras tecnologias.

As tecnologias utilizadas foram:

- *Microsoft SQL Server*

  Tecnologia bem estabelecida no mercado do fabricante Microsoft. É um sistema de gestão de base de dados relacionais, que permiteobterrecursos internos demissão críticae permitir tambémideiasrevolucionáriasem toda a organizaçãocomferramentas de análisefamiliarese soluçõesde *Big Dataenterprise-ready*. Arquitetura eferramentascomunspermiteminfraestruturasde TIhíbridasque ajudam aavançar o negócioe desbloquearnovos modelos de negócios.
  Retirado dehttp://www.microsoft.com/en-us/sqlserver/product-info/overview-capabilities.aspx

- *Microsoft. Net Framework*
  - Tecnologia de desenvolvimento do fabricante Microsoft. É uma plataforma de trabalho que oferece inúmeras opções para que o desenvolvimento se concentre na área de negócio, podendo-se assim abstrair da camada mais baixa do desenvolvimento e é totalmente orientada por objetos. Usando esta tecnologia consegue-se tirar partido dos agentes de execução quedisponibilizam serviços como gestão de memória, segurança e outros tipos de código que promovem robustez.
    Retirado de http://msdn.microsoft.com/en-us/library/zw4w595w.aspx

- *TelerikRadControls Framework*
  - Tecnologia de apresentação da *Telerik*. É uma plataforma de trabalho que disponibiliza uma coleção de controlos de apresentação que facilita a disponibilização dos dados para o utilizador, e enriquece a sua experiência, maximizando a produtividade do desenvolvimento.Retirado de http://www.telerik.com/company.aspx

Similaridades com outros domínios

Todas as empresas têm a necessidade de manter um histórico sobre determinadas operações ou transações como por exemplo todos os produtos vendidos num determinado exercício. Em quase todos os casos este histórico é possível por utilização de aplicações que mantêm os registos numa base de dados. Este cenário é muito similar com o Controle de Reparações, mas neste caso, o histórico guardado é referente ao ciclo de vida de componentes eletrónicos.

Definição de problema

Olhando ao cenário atual, todos os procedimentos de reparações são efetuados sem qualquer tipo de controlo.As máquinas são recebidas e encaminhadas diretamente para os departamentos respetivos, reparadas e enviadas de volta aos clientes.Este cenário apenas é funcional com o devido prepósito. Como consequência, não é assim possível demonstrar qualquer tipo de controlo, e também tirar qualquer tipo de estatística das operações. Diariamenteverifica-se a necessidade de se saber como está o estado da obra x ou quantas obras há pendentes, ou outras questões dentro do mesmo âmbito que só é possível responder com prontidão e exatidão com um sistema de controlo onde se registem todas as operações sobre o ciclo de vida dos equipamentos.

Recolha de Requisitos

É necessário por cada entrada de material que haja um registo de data e hora, e seja registado e anexado um número de obra.

Ao criar a obra, procedimento efetuado na receção do material, o departamento de reparações será notificado eletronicamente.

Manter um registo das reparações efetuadas sobre a obra e caso sejam necessários orçamentos, deverãoficar registados em histórico.

Possibilidade de anexar á obra o registo de empréstimos de equipamentos de substituição.

Possibilidade de notificação ao cliente de procedimentos via mensagem eletrónica.

Na conclusão das reparações e fecho de obra, o departamento de logística será notificado para dar início ao procedimento de despacho do material.

Para que seja enviado o material é necessário que seja possível emitir guias de transporte.

É também necessário que sejam guardados todos os registos descritos acima para consulta de histórico.

Análise dos Requisitos

Mediante os requisitos apresentados, será necessário desenvolver uma aplicação com as seguintes funcionalidades:

- Autenticação;
- Gestão de obras;
- Gestão clientes;
- Gestão material;
- Emissão de orçamentos;
- Emissão de guias de transporte;
- Envio de mensagens eletrónicas;
- Relatórios de consulta.

Os departamentos envolvidos são:

- Receção;
- Reparações;
- Financeiro;
- Logísticas.

Especificação dos Requisitos

Autenticação

Sistema de controlo de utilização com capacidade ilimitada de utilizadores. Cada utilizador terá, mediante aprovação, acesso a áreas da aplicação. Será necessário introduzir as credenciais para entrar no sistema. Cada operação significativa terá um registo de execução associado com o utilizador corrente.

Gestão de obras

Esta área enquadra-se no âmbito principal da aplicação. Estará dividida em três secções: Entrada, Lista, Saída. "Entrada" é onde se cria o registo de entrada de equipamento para reparação. Durante o processo de reparação, qualquer tipo de operações são registadas e podem ser consultadas na "Lista". Nesta secção é ainda possível registar pedidos de equipamentos de substituição e orçamentos. Após a reparação ser finalizada dá-se a obra como fechada, e esta passa para a secção de "Saída" onde se dá o processo de envio de volta para o cliente. Nesta secção é possível emitir guias de transporte.

Gestão clientes

Área de gestão dos perfis de cliente. Nesta área pode-se gerir os dados dos clientes da empresa e detentores dos equipamentos. Apesar de não ser na íntegra um CRM, poderá ser encarado como tal, tendo as funcionalidades mínimas. Toda a estrutura de registo assenta sobre os perfis de cliente.

Gestão material

É a área de gestão dos equipamentos. Esta área é dividida em duas subáreas, a dos equipamentos e a de que famílias pertencem, constituindo assim grupos. Todos os equipamentos têm de estar relacionados com o perfil de cliente.

Emissão de orçamentos

Sempre que a reparação de uma obra não esteja ao abrigo do contrato de manutenção ou o equipamento já não se encontre dentro do prazo de garantia, serão emitidos orçamentos a enviar ao cliente para aprovação da reparação. Nesta secção será possível ver os orçamentos pendentes e o histórico.

Emissão de guias de transporte

Área de emissão das guias de transporte. Sempre que se enviam equipamentos, emite-se uma guia de transporte. Podem-se emitir novas guias ou consultar o histórico de emissão.

Envio de mensagens eletrónicas

A funcionalidade de envio de mensagens eletrónicas possibilita o envio de qualquer documento de registo ou de estatística por mensagem eletrónica, com a finalidade de não ser necessária a impressão dos mesmos.

Relatórios de consulta

Área de relatórios. As consultas podem ser efetuadas diretamente na aplicação, ou por emissão de relatórios estatísticos.

Âmbito geral da aplicação

Nenhum registo poderá ser apagado uma vez que tenha operações associadas.

Desenho

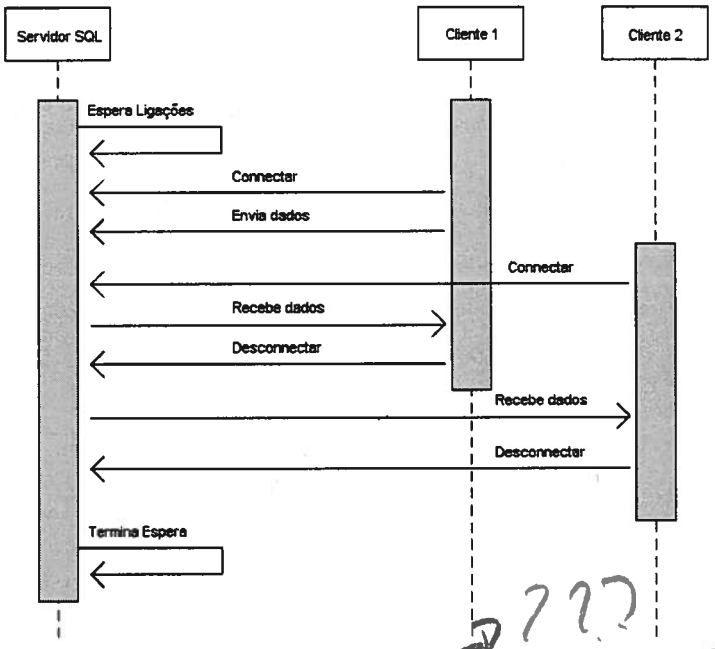
Como requisitos de *hardware*, será necessário como mínimo um computador com um processador central de 2.8GHZ e 1GB de memória para os clientes e servidor.No servidor recomenda-se, nos discos que alojam os ficheiros de dados da base de dados, um sistema de redundância de discos.

As cópias de segurança da base de dados são efetuadas executando um procedimento guardado (*storedprocedure*) do SQL (anexo 1).

Para criação da base de dados é necessário executar o script de criação (anexo 2).

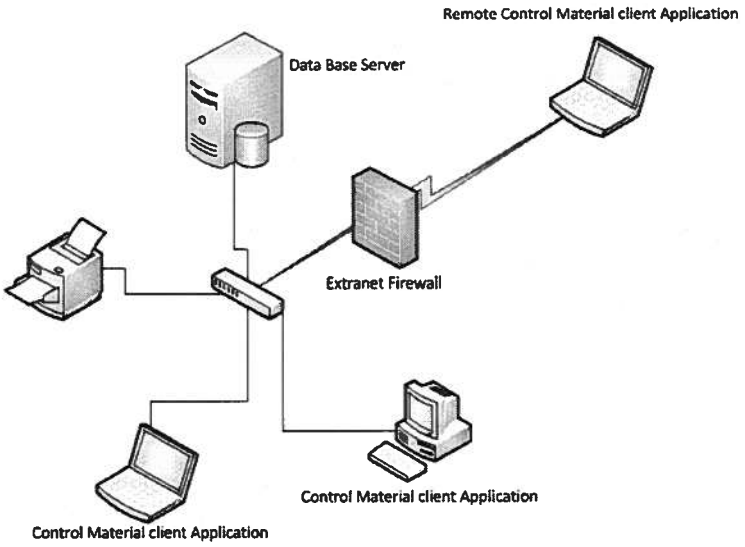
A segurança a nível de base de dados é de acesso total.

A aplicação vai correr num ambiente cliente-servidor, em que no servidor apenas estará a estrutura da base de dados, e nos clientes apenas a aplicação.A tecnologia Entity Framework da plataforma ADO.net usada na aplicação ira tirar partido da funcionalidade piscina de ligações (*ConnectionPooling*) do sistema de base de dados, para melhor aproveitamento e desempenho do sistema.



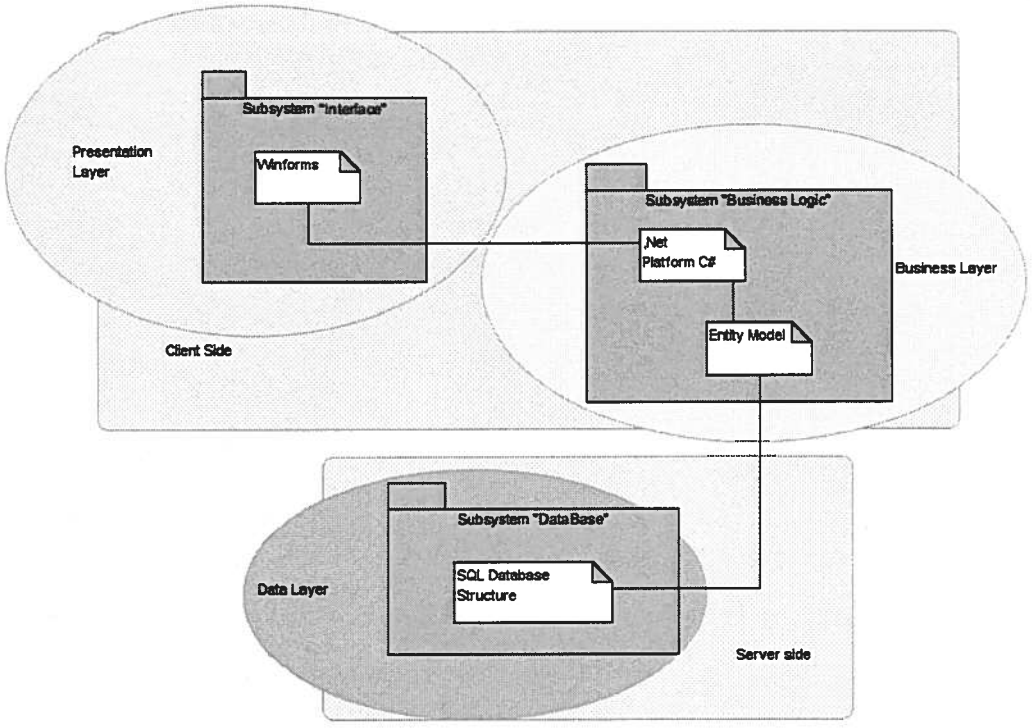
- O serviço da base de dados arranca e fica á escuta de conexões.
- Um qualquer cliente executa a ligação e pode ou não efetuar troca de dados.
- Outro qualquer cliente pode em simultâneo efetuar também uma ligação e trocar dados.
- Este tipo de comunicação continuará disponível até que o serviço que instancia a base de dados seja desligado.

A ligação da aplicação á base de dados é efetuada sobre o protocolo de rede TCP IP dando a possibilidade de ligações remotas através de conexões seguras.



Decisão de metodologia de subsistema

A aplicação é desenhada baseado no método N-tier, respeitando assim as três camadas lógicas de desenvolvimento, método também utilizado na tecnologia MVC (*ModelViewerController)*nas soluções de rede pública:



Camada de Interface

Esta camada apresenta todo o interface gráfico ao utilizador, disponibilizando as opções de escolha e de interação para os fluxos de trabalho disponíveis.Todo o interface de apresentação gráfica é desenvolvido na plataforma Microsoft *WindowsForms*.

Camada de Negócio

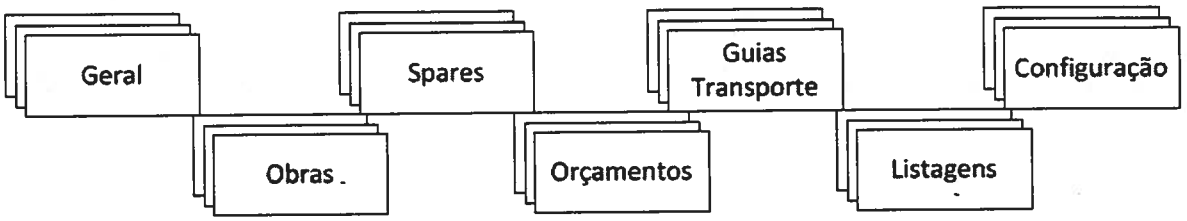
Nesta camada é executada toda a lógica de negócio extraindo, transformando e carregando (*extracttransformload,* método de *Business Inteligence*) os dados mediante as opções do utilizador. A interação com a base de dados dá-se através da tecnologia Microsoft *Entity Framework.*

Camada de Base de dados

Camada onde se encontra o modelo de dados definido dentro do sistema de gestão de base de dados. Para a base de dados é utilizada a solução Microsoft SQL *Server*.

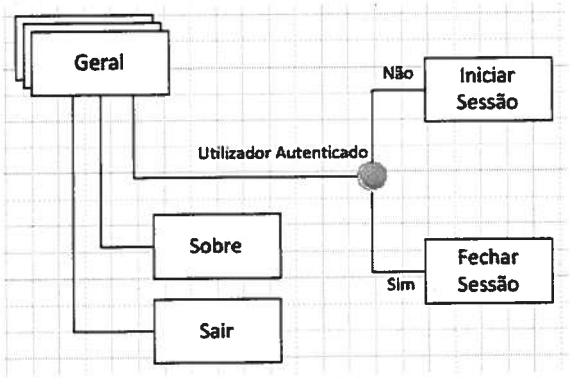
Decisão de Interface de utilização

O interface gráfico de utilização está divida em módulos de trabalho com o intuito depoderem ser distribuídas tarefas pelos respetivos departamentos. As opções do interface são disponibilizadas mediante os direitos atribuídos ao utilizador autenticado.



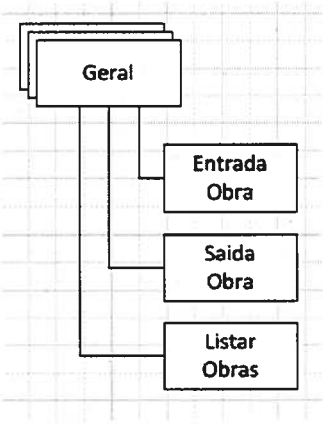
Módulo Geral

Opções de autenticação, sobre a aplicação e opção de saída. Nestas opções o utilizador pode ler sobre a aplicação, sair da aplicação ou autenticar-se para utilização da aplicação.



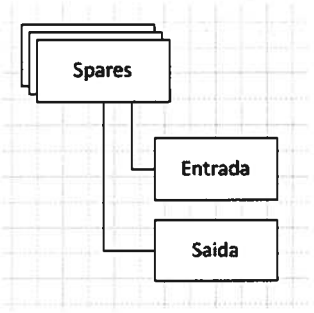
Módulo Obras

Opções de abrir obras, listar obras e saída de obras. Este módulo é o fluxo principal de trabalho da aplicação.Nele iniciam-se as reparações dando entrada domaterial para reparação, executam-se as notas da reparação, com possibilidade de emissão direta de orçamentos e guias de transporte.Dando-se a obra por concluída, finalmente dá-se saída da mesma.



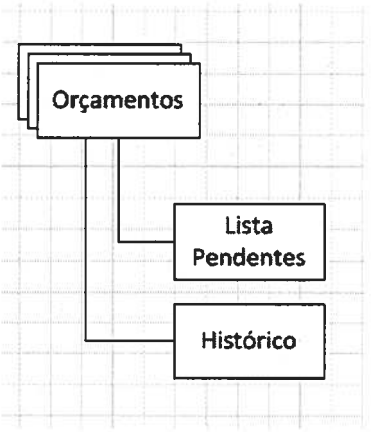
Módulo *Spares*

Opções de saída e entrada de *spares*. Neste módulo encontra-se disponível o resultado do fluxo de *spares* em cada obra, sendo possível a partir dele dar saída de um spare introduzido numa obra ou dar entrada do mesmo que estará disponível após reparação e saída da obra. Uma vez dada a saída de um spare já não é possível removê-lo da obra. Este módulo não é independente, ou seja, apenas existem dados para trabalhar nele caso os mesmos sejam manipulados nas obras.



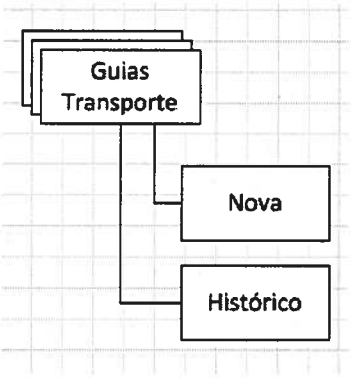
Módulo Orçamentos

Neste módulo podem consultar-seorçamentos pendentes e os seus históricos. Os orçamentos serão criados a partir das obras, ficando automaticamente preenchidos com os dados das mesmas. Orçamentos pendentes são orçamentos que estão é espera de aprovação do cliente, e uma vez aprovados ou rejeitados passam para histórico.



Módulo Guias Transporte

Neste módulo podem criar-se guias de transporte e histórico. Por norma as guias de transporte serão criadas a partir das obras, ficando automaticamente preenchidas com os dados da mesma, no entanto poderão ser criadas novas guias neste módulo.

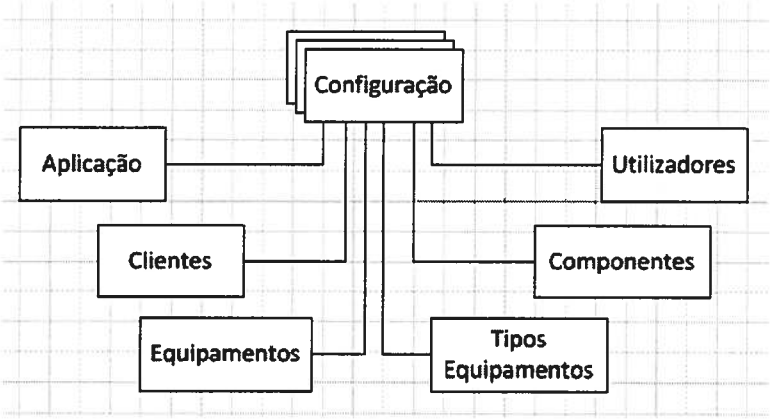


Módulo de Listagens

Neste módulo pode-se consultar várias informações estatísticas sobre os dados existentes. Este módulo apenas tem a opção de listagens.

Módulo de Configuração

O módulo de configuração é o mais significativo, uma vez que é onde se introduzem todos os dados que vão dar origem aos fluxos de trabalho da aplicação e às suas estatísticas. Sem os dados introduzidos não é possível executar operações que derivem desses mesmos módulos.

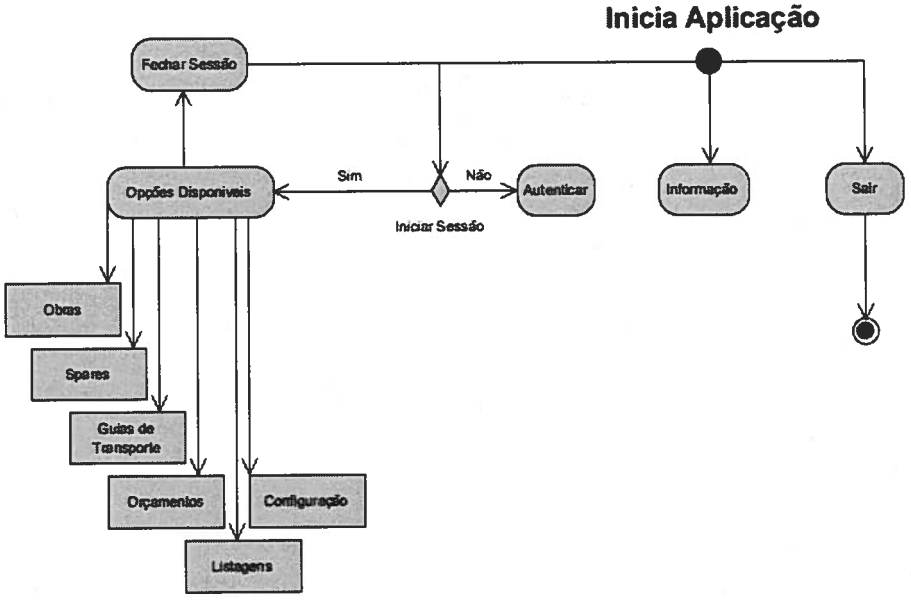


Modelação

Modelo de Utilização

Casos de Uso Geral

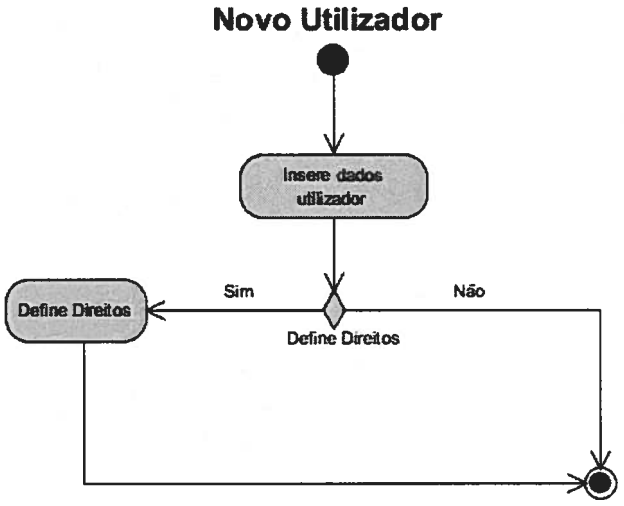
Para entrar na aplicação é necessário um utilizador e da respetiva palavra passe. A partir desse instante o utilizador autenticado está envolvido em todas as operações, sendo sempre registadas as alterações efetuadas.



Casos de Uso Configuração

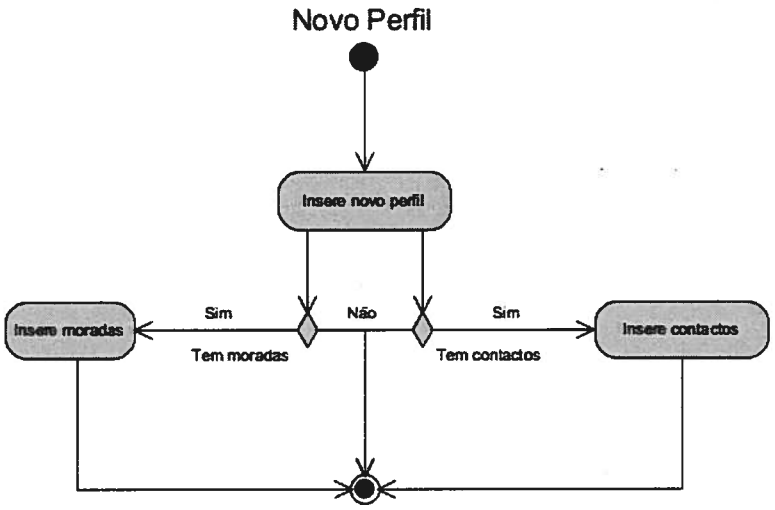
Utilizadores

Para que seja possívelutilizar a aplicação e usufruir das respetivas funcionalidades, é necessário definir um nome de utilizador, palavra passe e os respetivos direitos de utilização. O nome de utilizador tem de ser único.



Perfil

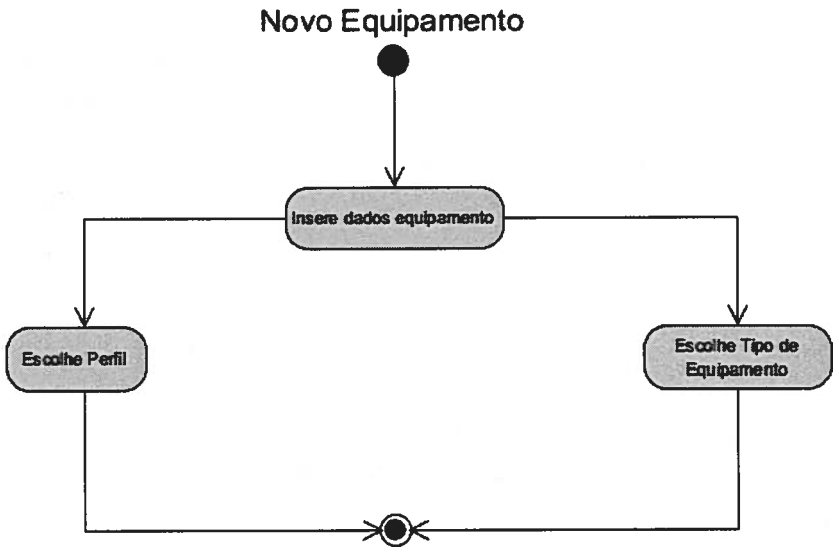
Todos os dados estão relacionados direta ou indiretamente com os perfis. Para inserir um novo perfil não é necessário requisitos, e pode-se ou não inserir moradas e contactos para o perfil a introduzir.



Equipamentos

Os equipamentos são a base da aplicação. Nenhuma obra consegue ser criada sem ter um equipamento associado. As obras são efetuadas sobre os equipamentos. Por sua vez, os equipamentos têm de pertencer a um perfil e a um tipo de equipamentos.

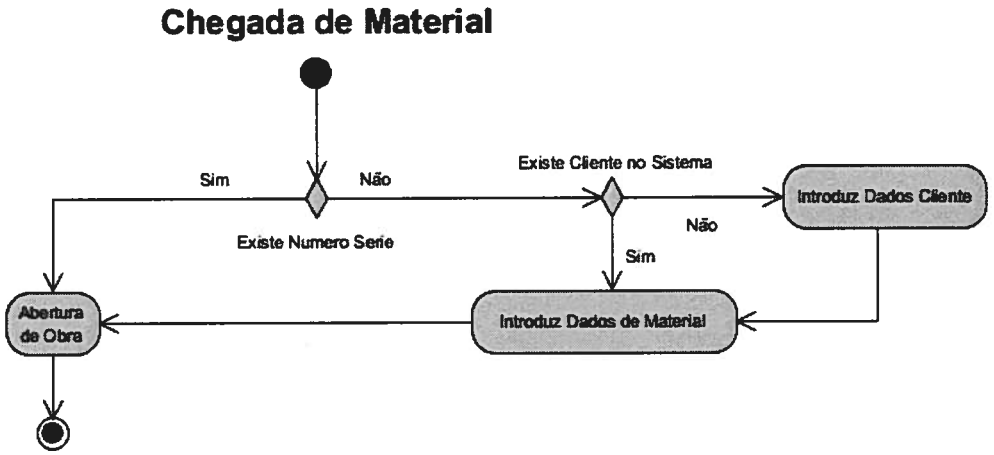
Os *spares* são equipamentos que estão associados ao perfil armazém.



Casos de UsoOperação

Chegada de Material

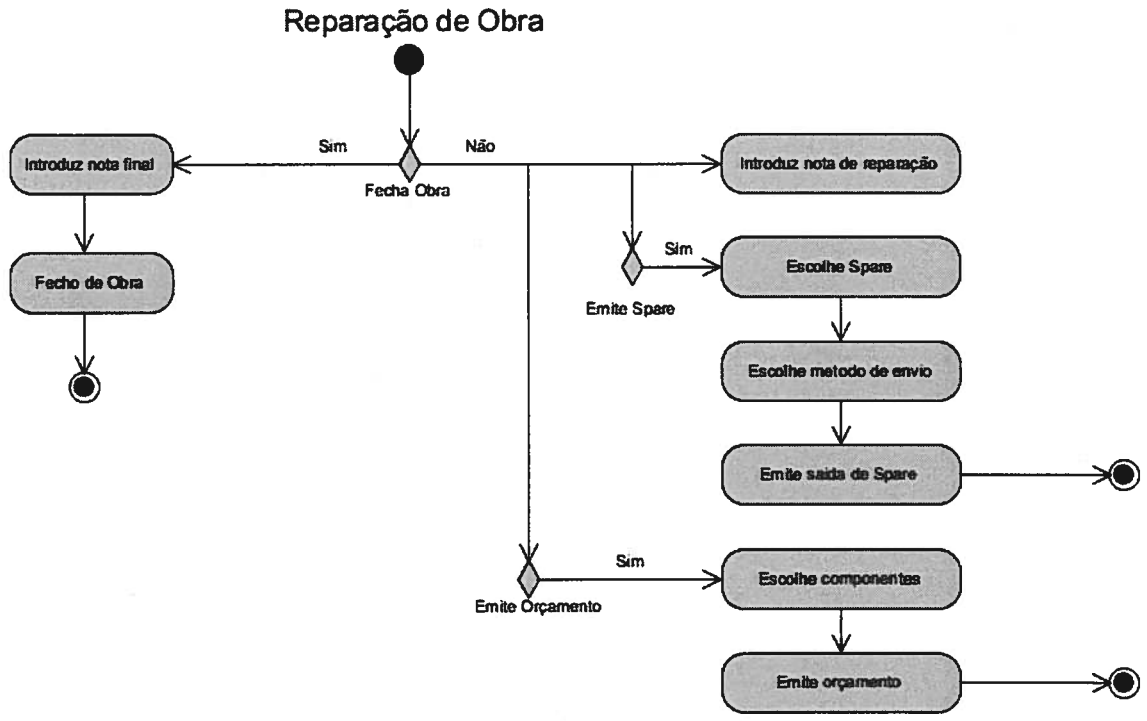
Ao chegar um equipamento para reparação, existem alguns fatores a ter em consideração. Não é possível dar uma nova entrada para reparação sem o equipamento estar registado no sistema. Por sua vez o equipamento só pode estar registo no sistema se pertencer a um perfil que também já deve existir no sistema.



Reparação de Obra

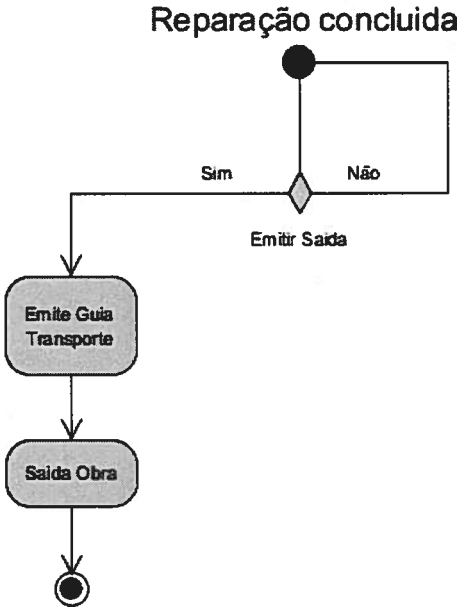
Este é o fluxo de trabalho primário dentro do domínio da aplicação. Numa obra pendente de reparação pode-se:

- Introduzir notas conforme desenvolvimento da reparação;
- Emitir um *Spare* selecionando os equipamentos da lista disponível e especificando o tipo de envio que poderá ser via cliente diretamente, ou via transportadora.Este procedimento dá-se quando a reparação pode levar mais tempo que o previsto e o cliente necessite de um equipamento de substituição. Uma vez gravado o pedido de emissão de *Spare*, o material de substituição fica pendente para saída no menu de *Spares*;
- Emitir um orçamento. Sempre que a reparação não se encontre ao abrigo do contrato de manutenção deve-se emitir um orçamento. A emissão exige a escolha dos componentes necessários para a reparação;
- Fechar a obra alterando o seu estado, e passando-a para a opção de pendentes para saída. Para executar a instrução de fechar é necessário introduzir uma nota de finalização.



Saída de Material

Esta é a última operação dentro do domínio da aplicação. Uma vez que a obra seja fechada passa a estar com o estado pendente para saída, e aparece no respetivo módulo. Uma vez escolhida a opção de saída é emitida uma guia de transporte para o envio do equipamento.



Modelo de Estrutura de Dados

O modelo de dados foi desenvolvido mediante a necessidade apresentada e está desenhado de forma a respeitar a terceira forma de normalizaçãocom Boyce-Codd. Nesta forma garantimos que:

- Todos os atributos da relação estão baseados num domínio simples e não contêm grupos ou valores repetidos (respeita a primeira forma de normalização);
- Cada atributo não chave é dependente da chave primária inteira, ou seja, cada atributo não chave não poderá ser dependente de apenas parte da chave (respeita a segunda forma de normalização);
- Todos os atributos não chave não dependem de outros atributos não chave (respeita a terceira forma de normalização);
- Todos os atributos determinantes podem ser chaves primárias.

Citando uma frase que sumariza as formas normais:

"A table is based on
the key,
the whole key,
and nothing but the key (so help me Codd)"

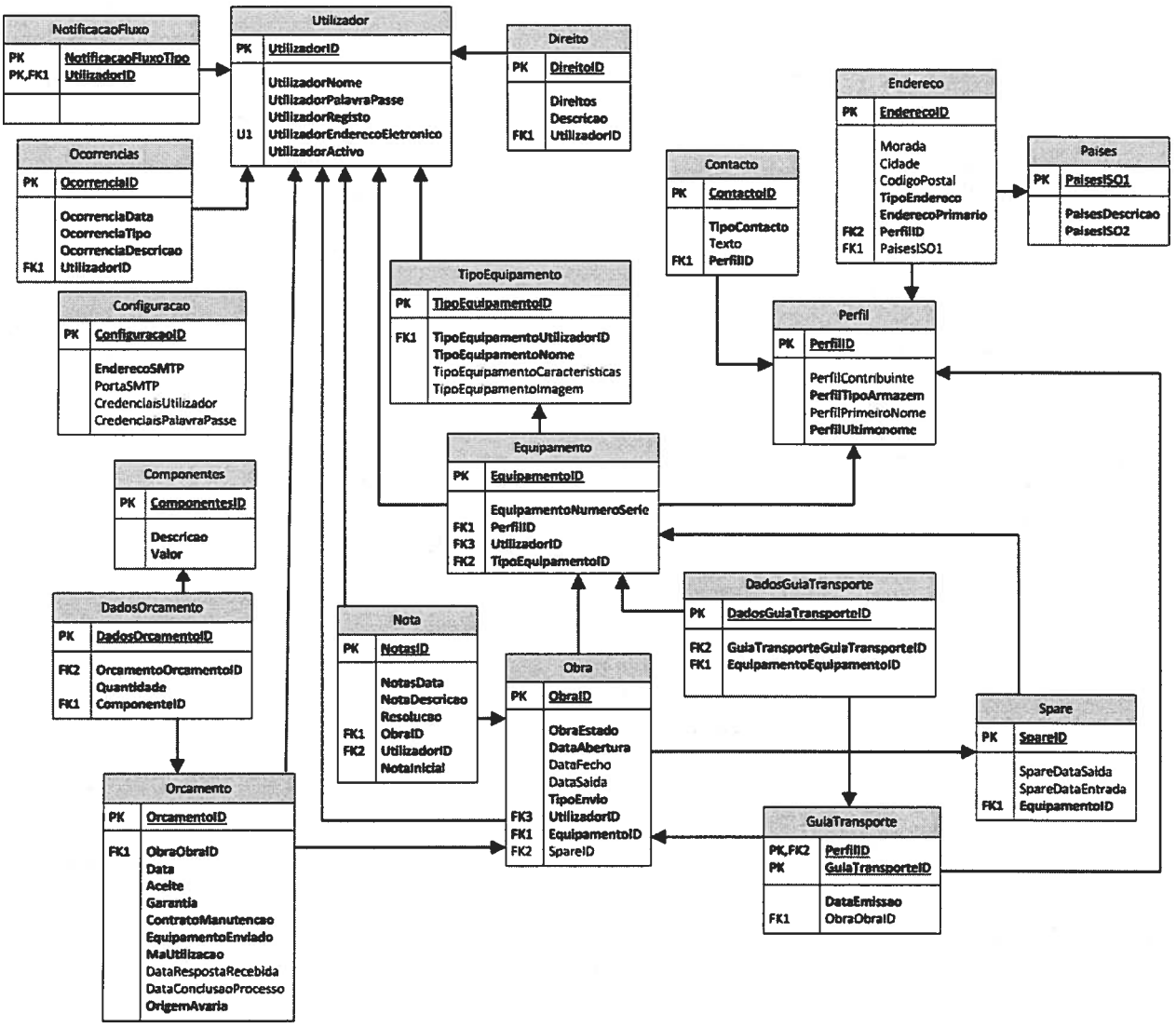
Retirado de "ClareChurcher(2007).BeginningDatabaseDesign.Apress"

Muitos dos atributos têm a finalidade de ter um determinado leque de valores, e para os restringir, foram criados gatilhos mandando uma exceção ao utilizador com a mensagem

correspondente. Existem dois procedimentos guardados (*storedprocedures*), umpara limpar a base de dados caso haja a necessidade de começar tudo de novo, repondo-a no seu estado original, e outro para executar cópias de segurança para o disco local.

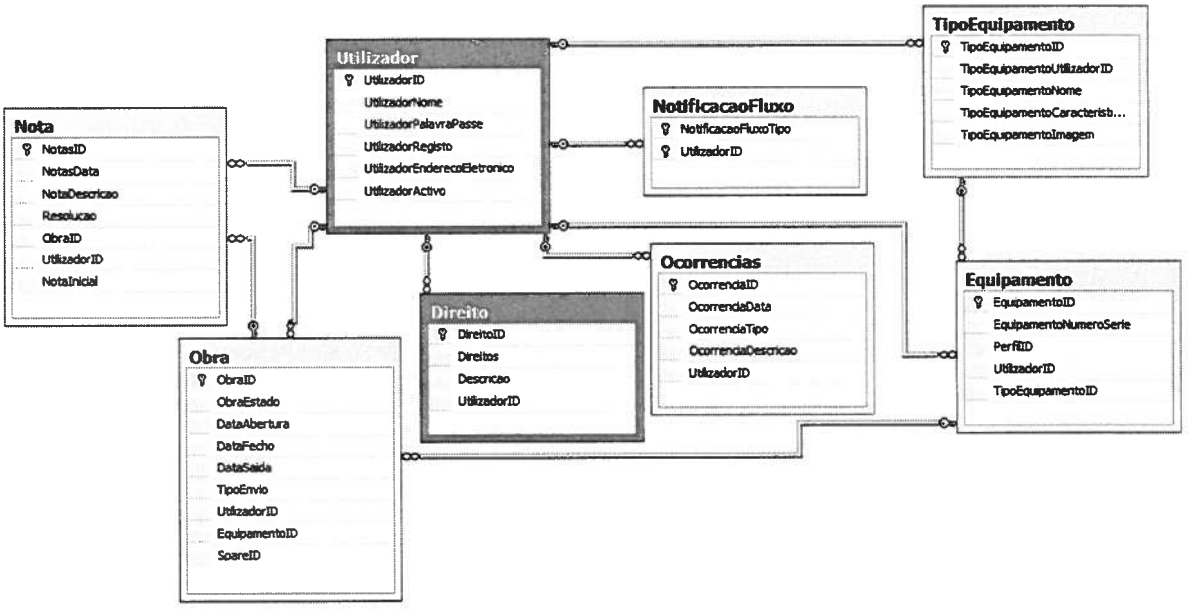
Estes objetos na base de dados dão a possibilidade de a mesma poder ser utilizada por mais aplicações para além da aplicação deste projeto.

Visão geral do Modelo



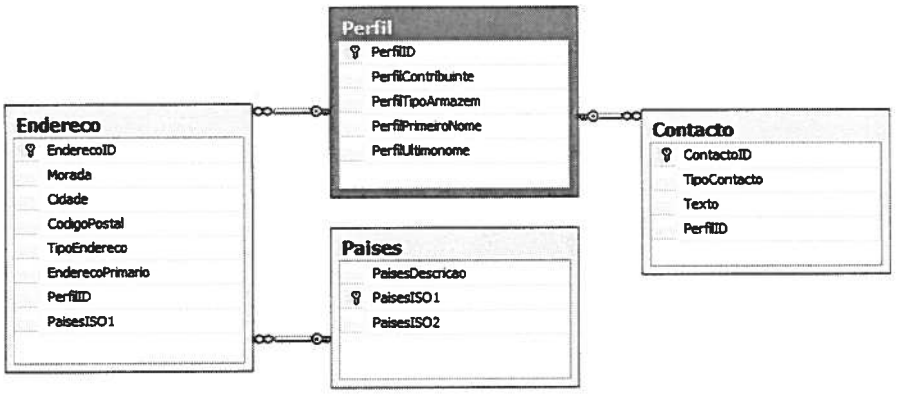
Utilizador

A entidade Utilizador contém informação referente aos operadores da aplicação e está relacionada com as principais operações do domínio. Estas operações registam sempre o utilizador responsável pela alteração.A entidade diretamenterelacionada tem o nome de Direito e contém os direitos referentes a cada utilizador.



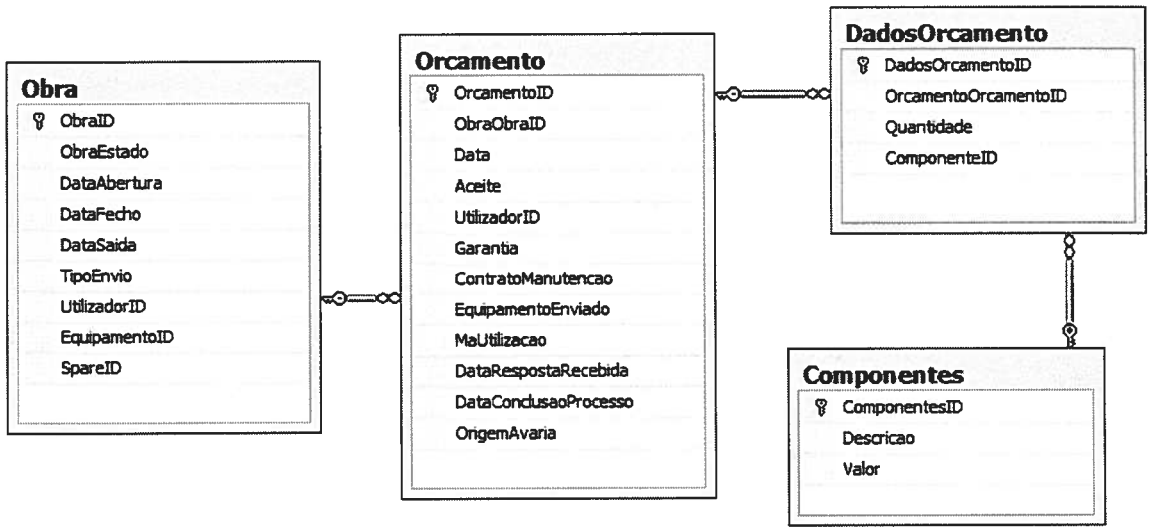
Perfil

A entidade Perfil contém os dados relevantes dos clientes e está localizado no topo da hierarquia da configuração. Para além de conter as suas informações, toda a estrutura de dados referentes a equipamentos, *spares*, obras, orçamentos e guias de transporte está dependente dela. As entidades de relação direta são o Endereço, que por sua vez tem uma relação direta com a entidade Países,e com a entidade Contacto.



Orçamentos

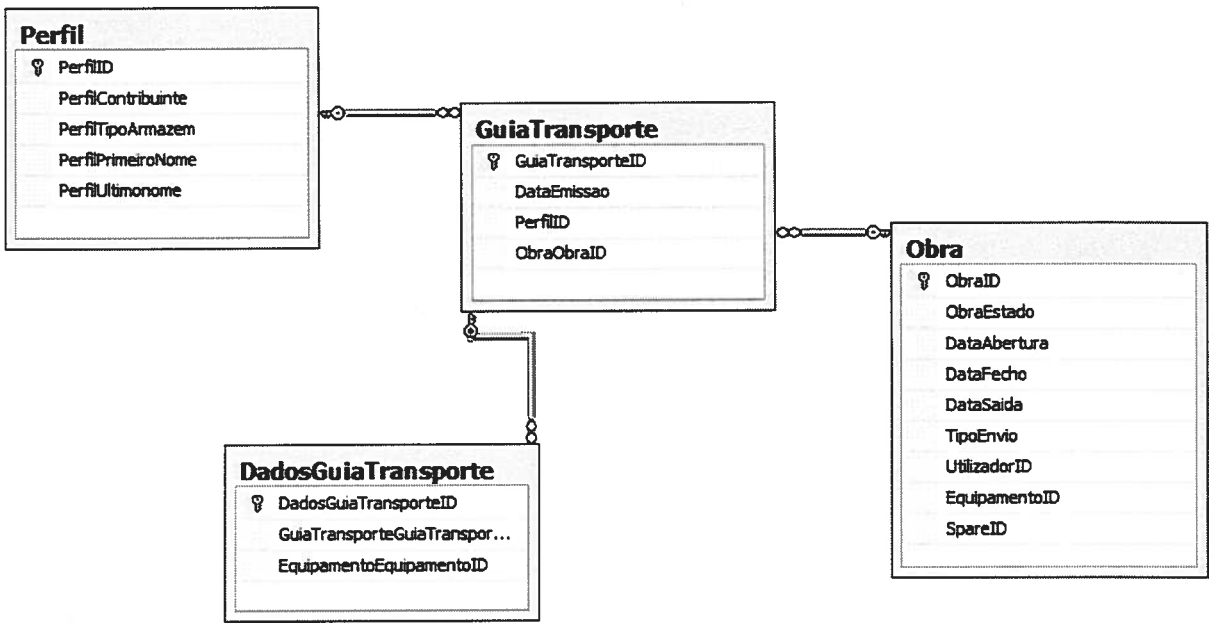
A entidade Orcamento contém os dados referentes ao cabeçalho dos orçamentos. Está diretamente dependente da entidade Obra. A entidade de relação direta é a DadosOrcamento que contêm os dados de detalhe referentes ao respetivo orçamento, que por sua vez tem uma relação direta com a entidade Componentes.



Guias de Transporte

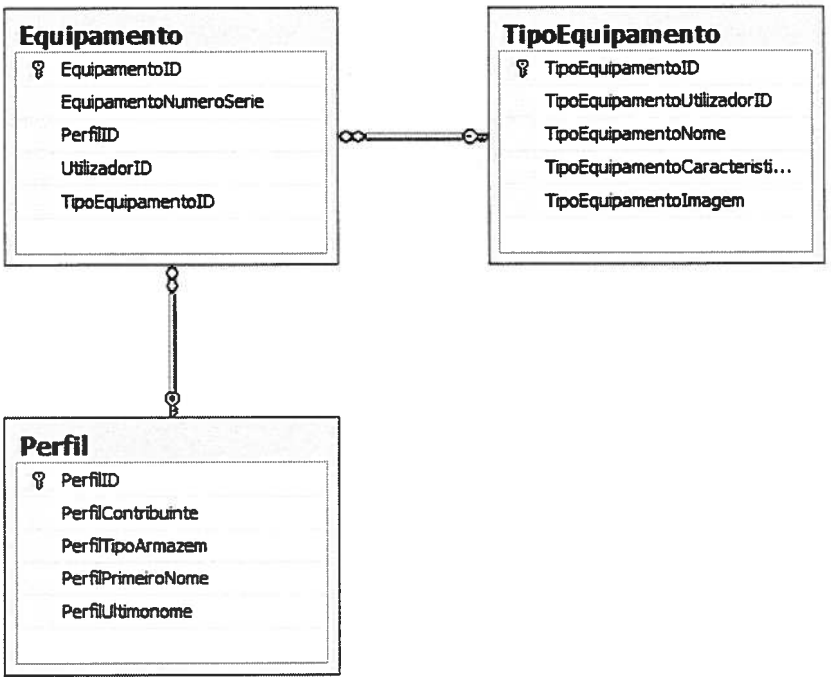
A entidade GuiaTransportescontém os dados referentes ao cabeçalho das guias de transporte. A entidade de relação direta é a DadosGuiaTransporte que contem os dados de detalhe referentes á respetiva guia. A guia de transporte pode ser emitida sem qualquer relação com uma obra em específico, mas tem de estar sempre relacionada direta ou indiretamente com um perfil.

Caso tenha obra relacionada o perfil é automaticamente relacionado com o que por sua vez está relacionado com a obra, caso contrario tem de se relacionar um perfil individualmente.



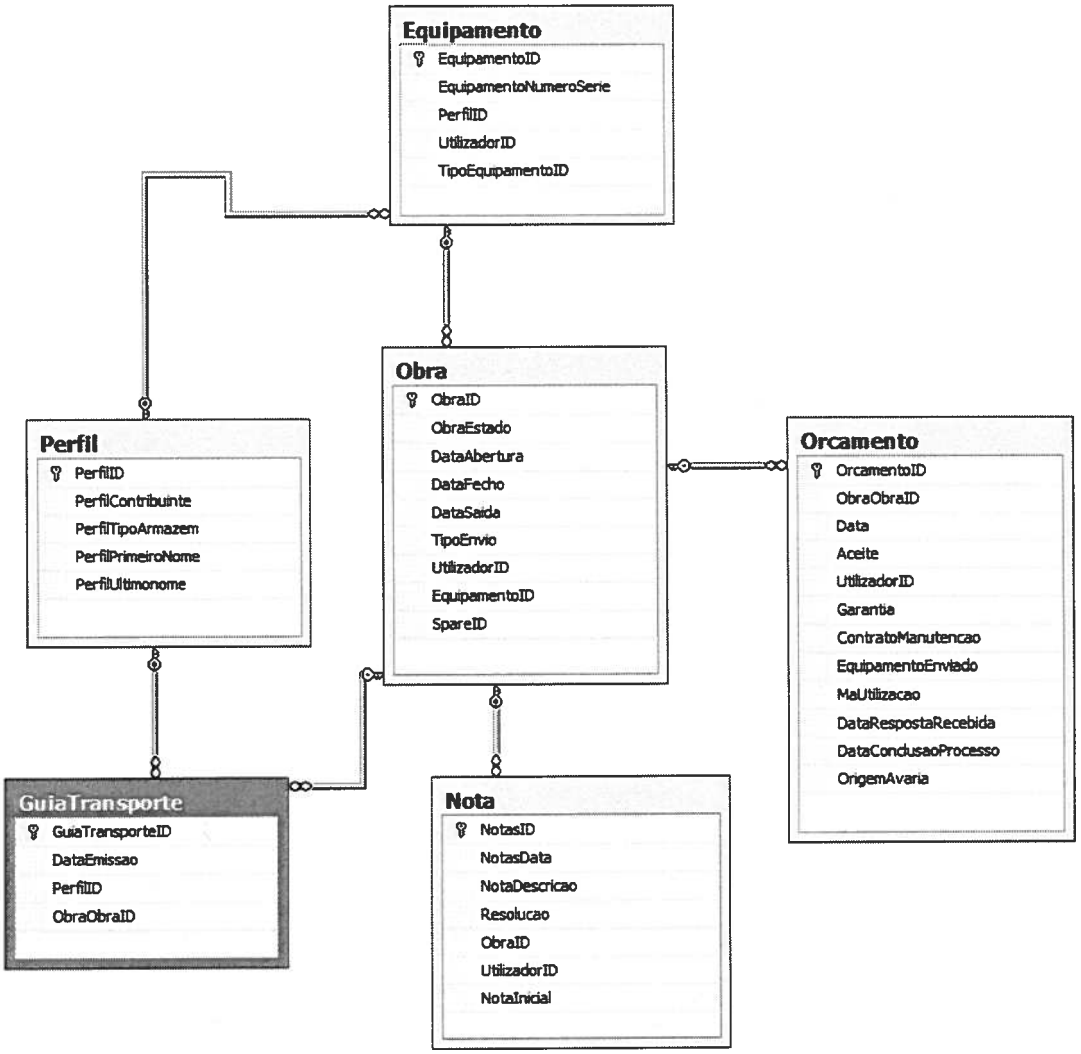
Equipamentos

A entidade Equipamento contém os dados sobre o equipamento em questão para reparação. É com base neste equipamento que se efetuam as obras. Cada equipamento tem uma relação direta com a entidade TipoEquipamento e com a entidade Perfil.



Obras

Sendo a obra o núcleo das operações dentro do domínio, a entidade Obra contém os dados da mesma e uma referência para as principais entidades dentro do modelo relacional.Podem verificar-se as relações de todas as entidades apresentadas no diagramageral.

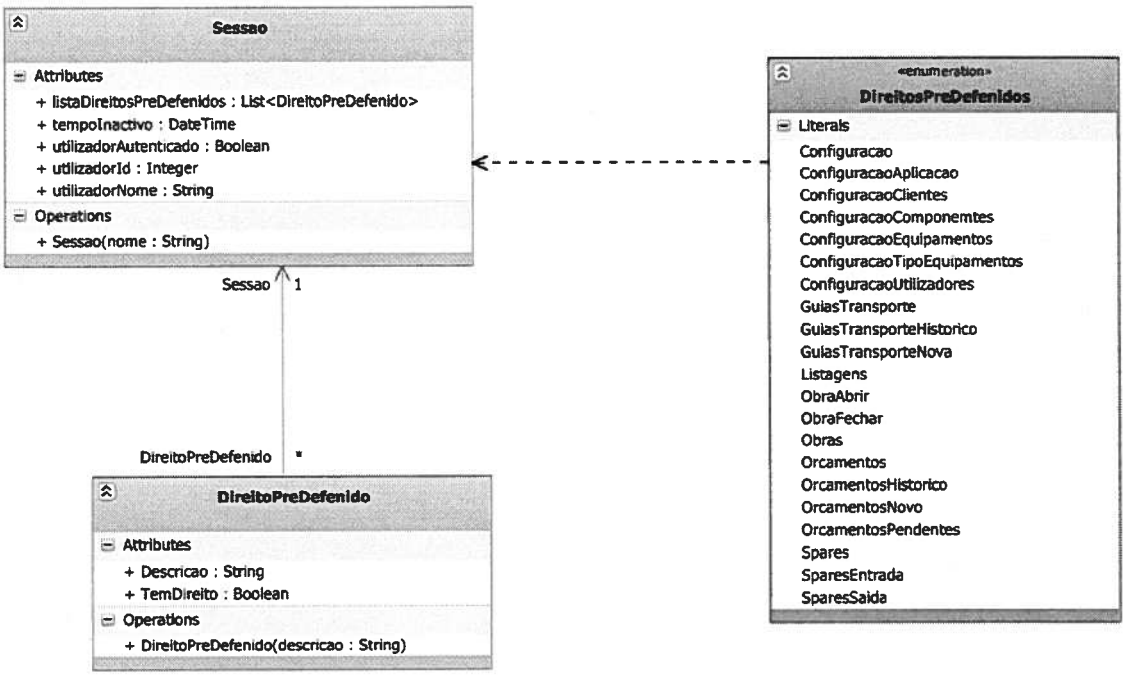


Modelo Dinâmico e de Comportamento

O domínio apresenta uma classe de comportamento dinâmico que é instanciada no início de cada sessão, e contem os dados do utilizador autenticado.O nome da classe é Sessão. Os dados são validados com os existentesna base de dados, e em caso de autenticação sucedida, é guardada uma instância dinâmica da mesma com uma lista agregada de direitos.Esta instância dinâmica navega de forma em forma, conforme o utilizador navega na aplicação. A lista agregada de direitos vai determinando se as funcionalidades da aplicação estão disponíveis ou não, conforme o utilizador vai navegando.A instância desta classe é individual e está relacionada com processo da aplicação.

Foi implementado também um comportamento dinâmico que, com base no valor do atributo tempoInactivo da classe Sessão, pode executar automaticamente a saída do utilizador da aplicação por tempo de inatividade, forçando assim a reautenticação. A implementação deste comportamento exigiu a criação de um delegado de gerenciador de eventos (*eventhandler)*que é aplicado a todas as classes de apresentação, e o evento é disparado sempre que o utilizador execute qualquer ação. Este disparo referido faz com que o atributo

tempoInactivo retorne ao estado inicial, não executando o procedimento de saída do utilizador da aplicação.



Existe uma classe decifra que tira partido das classes da plataforma de trabalho Microsoft.Net *TripleDESCryptoServiceProvidere* *MD5CryptoServiceProvider*, para cifrar e decifrar as palavras secretas dos utilizadores, garantindo assim a confidencialidade das mesmas quando são guardadas na base de dados.

Conclusão

Após um ano de investigação na área de desenvolvimento,é assim possível concluir que a plataforma de trabalho da Microsoft oferece todas as ferramentas necessárias para a análise e desenvolvimento de uma solução completa.

Durante o decorrer do projeto surgiram vários desafios devido á necessidade de responder a todos os requisitos. Só foi possível superar estes desafios devido á documentação que é disponibilizada e que acompanha a plataforma de trabalho.

A solução desenvolvida partiu de uma necessidade específica da Micros Fidelio Portugal com a finalidade de conseguir prestar um melhor serviço aos seus clientes. As principais necessidades exigiam uma solução de baixo custo, de elevada flexibilidadeepersonalização,respeitando sempre os requisitos.

Com base neste problema foi realizado um estudo aprofundado na temática de engenharia de*software*. Este incluiu o estudo do processo de análisee de modelação de requisitos.Através do estudo realizado foi produzida uma ferramenta que não contendo todas as funcionalidadesquando comparada com as ferramentas concorrentes a nível comercial, se distingue peloseguinte:

- Oferece aos seus utilizadores as principais tarefas na reparação dos materiais de umaforma simples e de fácil aprendizagem;
- Disponibiliza um sistema de registode trabalhos executados que permite verificar cadapassoda reparação;
- Apresenta um novo meio estatístico através de relatóriosque permite a análise e discussão das tarefas e dos métodos de trabalho,focando os esforços no melhoramentoda qualidade dos serviços;

Em suma a ferramenta que resultou do estudo e investigação dos requisitos revelou-se ser uma ferramenta adequada para auxiliar a gestão dasreparaçõesdo material informático ao longo do seu tempo de vida útil. Esta ferramenta disponibiliza também algumas tarefas essenciais para a gestão desse material, o que é por si uma mais-valia.

Referências bibliográficas

Timothy C. Lethbridge& Robert Laganière(2005). Object-Oriented Software EngineeringPractical Software Development using UML and Java Second edition. Porto: McGraw Hill

Scott Klein (2010). Pro Entity Framework 4.0. Porto: Apress

Alex Mackey (2010). Introducing .NET 4.0 With Visual Studio 2010. Porto: Apress

Ian Sommervile (2004). Software Engineering 7th edition

Anexos e Apêndices

Anexo 1 – SQL script de criação trabalho de cópia de segurança da base de dados

```
GO
/****** Object:  StoredProcedure [dbo].[ControlMaterialBackup]    Script Date: 06/28/2013 23:40:22 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:                  <Rui Machado>
-- Create date: <28/06/2013>
-- Description:     <backsupcontrolmaterial database to disk>
-- =============================================
CREATE PROCEDURE [dbo].[ControlMaterialBackup]
        -- Add the parameters for the stored procedure here
AS
BEGIN
        BACKUP DATABASE ControleMaterial TO DISK = 'C:\backup\BackupControlMaterial_.bak'
END
GO
```

Anexo 2 – SQL script de criação dos objetos da base de dados

```
IF  EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[Componentes]') AND type in
(N'U'))
   DROP TABLE [dbo].[Componentes]
GO
CREATE TABLE [dbo].[Componentes] (
[ComponentesID] intidentity  NOT NULL
, [Descricao] varchar(255)  NOT NULL
, [Valor] money  NOT NULL
```

```
)
GO

ALTER TABLE [dbo].[Componentes] ADD CONSTRAINT [PK__Componen__2A2E076B07C12930] PRIMARY KEY
CLUSTERED (
[ComponentesID]
)
GO
GO

IF  EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[Perfil]') AND type in (N'U'))
   DROP TABLE [dbo].[Perfil]
GO
CREATE TABLE [dbo].[Perfil] (
[PerfilID] intidentity  NOT NULL
, [PerfilContribuinte] varchar(30)  NULL
, [PerfilTipoArmazem] bit  NOT NULL
, [PerfilPrimeiroNome] varchar(255)  NULL
, [PerfilUltimonome] varchar(255)  NOT NULL
)
GO

ALTER TABLE [dbo].[Perfil] ADD CONSTRAINT [PK__Perfil__0C005B66628FA481] PRIMARY KEY CLUSTERED (
[PerfilID]
)
GO
GO

IF  EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[Configuracao]') AND type in
(N'U'))
   DROP TABLE [dbo].[Configuracao]
GO
CREATE TABLE [dbo].[Configuracao] (
```

```
[ConfiguracaoID] intidentity  NOT NULL

, [EnderecoSMTP] varchar(255)  NOT NULL

, [PortaSMTP] int  NULL

, [CredenciaisUtilizador] varchar(255)  NULL

, [CredenciaisPalavraPasse] varchar(255)  NULL

)

GO


ALTER TABLE [dbo].[Configuracao] ADD CONSTRAINT [PK__Configur__6AAFC0E90D7A0286] PRIMARY KEY
CLUSTERED (

[ConfiguracaoID]

)

GO

GO


IF  EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[Equipamento]') AND type in
(N'U'))

   DROP TABLE [dbo].[Equipamento]

GO

CREATE TABLE [dbo].[Equipamento] (

[EquipamentoID] intidentity  NOT NULL

, [EquipamentoNumeroSerie] varchar(255)  NOT NULL

, [PerfilID] int  NOT NULL

, [UtilizadorID] int  NOT NULL

, [TipoEquipamentoID] int  NOT NULL

)

GO


ALTER TABLE [dbo].[Equipamento] ADD CONSTRAINT [PK__Equipame__7E9115025629CD9C] PRIMARY KEY
CLUSTERED (

[EquipamentoID]

)

GO
```

```
GO

IF  EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[Spare]') AND type in (N'U'))
   DROP TABLE [dbo].[Spare]
GO
CREATE TABLE [dbo].[Spare] (
[SpareID] intidentity  NOT NULL
, [SpareDataSaida] datetime  NULL
, [SpareDataEntrada] datetime  NULL
, [EquipamentoID] int  NOT NULL
)
GO

ALTER TABLE [dbo].[Spare] ADD CONSTRAINT [PK__Spare__8A23B0055CD6CB2B] PRIMARY KEY
CLUSTERED (
[SpareID]
)
GO
GO

IF  EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[Ocorrencias]') AND type in (N'U'))
   DROP TABLE [dbo].[Ocorrencias]
GO
CREATE TABLE [dbo].[Ocorrencias] (
[OcorrenciaID] intidentity  NOT NULL
, [OcorrenciaData] datetime  NOT NULL
, [OcorrenciaTipo] varchar(255)  NOT NULL
, [OcorrenciaDescricao] text  NOT NULL
, [UtilizadorID] int  NOT NULL
)
GO
```

```
ALTER TABLE [dbo].[Ocorrencias] ADD CONSTRAINT [PK__Ocorrenc__30F556084222D4EF] PRIMARY KEY
CLUSTERED (

[OcorrenciaID]

)

GO

GO


IF  EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[Contacto]') AND type in (N'U'))

   DROP TABLE [dbo].[Contacto]

GO

CREATE TABLE [dbo].[Contacto] (

[ContactoID] intidentity  NOT NULL

, [TipoContacto] varchar(30)  NOT NULL

, [Texto] text  NULL

, [PerfilID] int  NOT NULL

)

GO


ALTER TABLE [dbo].[Contacto] ADD CONSTRAINT [PK__Contacto__8E0F85C8693CA210] PRIMARY KEY
CLUSTERED (

[ContactoID]

)

GO

GO


IF  EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[GuiaTransporte]') AND type in
(N'U'))

   DROP TABLE [dbo].[GuiaTransporte]

GO

CREATE TABLE [dbo].[GuiaTransporte] (

[PerfilID] int  NOT NULL

, [GuiaTransporteID] intidentity  NOT NULL

, [DataEmissao] datetime  NOT NULL
```

```
, [ObraObraID] int  NULL

)

GO


ALTER TABLE [dbo].[GuiaTransporte] ADD CONSTRAINT [PK__GuiaTran__99DF5625498EEC8D] PRIMARY KEY
CLUSTERED (

[PerfilID]

, [GuiaTransporteID]

)

GO

GO


IF  EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[Orcamento]') AND type in (N'U'))

   DROP TABLE [dbo].[Orcamento]

GO

CREATE TABLE [dbo].[Orcamento] (

[OrcamentoID] intidentity  NOT NULL

, [ObraObraID] int  NOT NULL

, [Data] datetime  NOT NULL

, [Aceite] varchar(20)  NOT NULL

, [Garantia] bit  NOT NULL

, [ContratoManutencao] bit  NOT NULL

, [EquipamentoEnviado] bit  NOT NULL

, [MaUtilizacao] bit  NOT NULL

, [DataRespostaRecebida] datetime  NULL

, [DataConclusaoProcesso] datetime  NULL

, [OrigemAvaria] text  NOT NULL

)

GO


ALTER TABLE [dbo].[Orcamento] ADD CONSTRAINT [PK__Orcament__4E96F7597D439ABD] PRIMARY KEY
CLUSTERED (

[OrcamentoID]
```

```
)
GO
GO

IF  EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[DadosGuiaTransporte]') AND type
in (N'U'))
DROP TABLE [dbo].[DadosGuiaTransporte]
GO
CREATE TABLE [dbo].[DadosGuiaTransporte] (
[DadosGuiaTransporteID] intidentity  NOT NULL
, [GuiaTransporteGuiaTransporteID] int  NOT NULL
, [EquipamentoEquipamentoID] int  NOT NULL
, [PerfilID] int  NULL
)
GO

ALTER TABLE [dbo].[DadosGuiaTransporte] ADD CONSTRAINT [PK__DadosGui__9CA516A14D5F7D71] PRIMARY
KEY CLUSTERED (
[DadosGuiaTransporteID]
)
GO
GO

IF  EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[Nota]') AND type in (N'U'))
   DROP TABLE [dbo].[Nota]
GO
CREATE TABLE [dbo].[Nota] (
[NotasID] intidentity  NOT NULL
, [NotasData] datetime  NOT NULL
, [NotaDescricao] text  NOT NULL
, [Resolucao] bit  NOT NULL
, [ObraID] int  NOT NULL
, [UtilizadorID] int  NOT NULL
```

```
, [NotaInicial] bit  NOT NULL

)

GO


ALTER TABLE [dbo].[Nota] ADD CONSTRAINT [PK__Nota__494AC75B1920BF5C] PRIMARY KEY CLUSTERED (

[NotasID]

)

GO

GO


IF  EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[TipoEquipamento]') AND type in
(N'U'))

DROP TABLE [dbo].[TipoEquipamento]

GO

CREATE TABLE [dbo].[TipoEquipamento] (

[TipoEquipamentoID] intidentity  NOT NULL

, [TipoEquipamentoUtilizadorID] int  NOT NULL

, [TipoEquipamentoNome] varchar(255)  NOT NULL

, [TipoEquipamentoCaracteristicas] text  NULL

, [TipoEquipamentoImagem] image  NULL

)

GO


ALTER TABLE [dbo].[TipoEquipamento] ADD CONSTRAINT [PK__TipoEqui__DC75FD4F5070F446] PRIMARY KEY
CLUSTERED (

[TipoEquipamentoID]

)

GO

GO


IF  EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[DadosOrcamento]') AND type in
(N'U'))

  DROP TABLE [dbo].[DadosOrcamento]
```

```
GO

CREATE TABLE [dbo].[DadosOrcamento] (

[DadosOrcamentoID] intidentity  NOT NULL

, [OrcamentoOrcamentoID] int  NOT NULL

, [Quantidade] int  NOT NULL

, [ComponenteID] int  NOT NULL

)

GO


ALTER TABLE [dbo].[DadosOrcamento] ADD CONSTRAINT [PK__DadosOrc__E118C16D02084FDA] PRIMARY
KEY CLUSTERED (

[DadosOrcamentoID]

)

GO

GO


IF  EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[Direito]') AND type in (N'U'))

   DROP TABLE [dbo].[Direito]

GO

CREATE TABLE [dbo].[Direito] (

[DireitoID] intidentity  NOT NULL

, [Direitos] bit  NOT NULL

, [Descricao] varchar(255)  NOT NULL

, [UtilizadorID] int  NOT NULL

)

GO


ALTER TABLE [dbo].[Direito] ADD CONSTRAINT [PK__Direito__7A306E6438996AB5] PRIMARY KEY
CLUSTERED (

[DireitoID]

)

GO

GO
```

```
IF  EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[Utilizador]') AND type in (N'U'))
   DROP TABLE [dbo].[Utilizador]
GO
CREATE TABLE [dbo].[Utilizador] (
[UtilizadorID] intidentity  NOT NULL
, [UtilizadorNome] varchar(255)  NOT NULL
, [UtilizadorPalavraPasse] varchar(255)  NOT NULL
, [UtilizadorRegisto] datetime  NOT NULL
, [UtilizadorEnderecoEletronico] varchar(255)  NOT NULL
, [UtilizadorActivo] bit  NOT NULL
)
GO

ALTER TABLE [dbo].[Utilizador] ADD CONSTRAINT [PK__Utilizad__90F8E1C825869641] PRIMARY KEY
CLUSTERED (
[UtilizadorID]
)
GO
CREATE UNIQUE  INDEX [UQ__Utilizad__A12C6486286302EC] ON [dbo].[Utilizador] (
[UtilizadorEnderecoEletronico]
)
GO

IF  EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[NotificacaoFluxo]') AND type in
(N'U'))
   DROP TABLE [dbo].[NotificacaoFluxo]
GO
CREATE TABLE [dbo].[NotificacaoFluxo] (
[NotificacaoFluxoTipo] varchar(255)  NOT NULL
, [UtilizadorID] int  NOT NULL
)
GO
```

```
ALTER TABLE [dbo].[NotificacaoFluxo] ADD CONSTRAINT [PK__Notifica__89F77DF74BAC3F29] PRIMARY KEY
CLUSTERED (

[NotificacaoFluxoTipo]

, [UtilizadorID]

)

GO

GO


IF  EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[Endereco]') AND type in (N'U'))

   DROP TABLE [dbo].[Endereco]

GO

CREATE TABLE [dbo].[Endereco] (

[EnderecoID] intidentity  NOT NULL

, [Morada] ntext  NULL

, [Cidade] varchar(255)  NULL

, [CodigoPostal] varchar(255)  NULL

, [TipoEndereco] varchar(255)  NOT NULL

, [EnderecoPrimario] bit  NOT NULL

, [PerfilID] int  NOT NULL

, [PaisesISO1] varchar(10)  NULL

)

GO


ALTER TABLE [dbo].[Endereco] ADD CONSTRAINT [PK__Endereco__B9D9462F6FE99F9F] PRIMARY KEY
CLUSTERED (

[EnderecoID]

)

GO

GO


IF  EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[Obra]') AND type in (N'U'))

   DROP TABLE [dbo].[Obra]
```

```
GO
CREATE TABLE [dbo].[Obra] (
[ObraID] intidentity  NOT NULL
, [ObraEstado] varchar(10)  NOT NULL
, [DataAbertura] datetime  NOT NULL
, [DataFecho] datetime  NULL
, [DataSaida] datetime  NULL
, [TipoEnvio] nchar(10)  NOT NULL
, [UtilizadorID] int  NOT NULL
, [EquipamentoID] int  NOT NULL
, [SpareID] int  NULL
)
GO

ALTER TABLE [dbo].[Obra] ADD CONSTRAINT [PK__Obra__F3E1F4120DAF0CB0] PRIMARY KEY CLUSTERED (
[ObraID]
)
GO
GO

IF  EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[Paises]') AND type in (N'U'))
   DROP TABLE [dbo].[Paises]
GO
CREATE TABLE [dbo].[Paises] (
[PaisesDescricao] varchar(255)  NOT NULL
, [PaisesISO1] varchar(10)  NOT NULL
, [PaisesISO2] varchar(10)  NOT NULL
)
GO

ALTER TABLE [dbo].[Paises] ADD CONSTRAINT [PK__Paises__DFA38313778AC167] PRIMARY KEY
CLUSTERED (
```

```
[PaisesISO1]

)

GO

GO


GO


GO


GO


ALTER TABLE [dbo].[Equipamento] WITH CHECK ADD CONSTRAINT [FK_Equipamento_Perfil] FOREIGN KEY (

[PerfilID]

)

REFERENCES [dbo].[Perfil] (

[PerfilID]

)

ALTER TABLE [dbo].[Equipamento] WITH CHECK ADD CONSTRAINT [FK_Equipamento_Utilizador] FOREIGN
KEY (

[UtilizadorID]

)

REFERENCES [dbo].[Utilizador] (

[UtilizadorID]

)

ALTER TABLE [dbo].[Equipamento] WITH CHECK ADD CONSTRAINT [FK_Equipamento_TipoEquipamento]
FOREIGN KEY (

[TipoEquipamentoID]

)

REFERENCES [dbo].[TipoEquipamento] (

[TipoEquipamentoID]

)

GO
```

```
ALTER TABLE [dbo].[Spare] WITH CHECK ADD CONSTRAINT [FK_Spare_Equipamento] FOREIGN KEY (
[EquipamentoID]
)
REFERENCES [dbo].[Equipamento] (
[EquipamentoID]
)
GO

ALTER TABLE [dbo].[Ocorrencias] WITH CHECK ADD CONSTRAINT [FK_Ocorrencias_Utilizador] FOREIGN KEY (
[UtilizadorID]
)
REFERENCES [dbo].[Utilizador] (
[UtilizadorID]
)
GO

ALTER TABLE [dbo].[Contacto] WITH CHECK ADD CONSTRAINT [FK_Contacto_Perfil] FOREIGN KEY (
[PerfilID]
)
REFERENCES [dbo].[Perfil] (
[PerfilID]
)
GO

ALTER TABLE [dbo].[GuiaTransporte] WITH CHECK ADD CONSTRAINT [FKGuiaTransp834709] FOREIGN KEY (
[PerfilID]
)
REFERENCES [dbo].[Perfil] (
[PerfilID]
)
ALTER TABLE [dbo].[GuiaTransporte] WITH CHECK ADD CONSTRAINT [FKGuiaTransp812662] FOREIGN KEY (
[ObraObraID]
```

```
)
REFERENCES [dbo].[Obra] (
[ObraID]
)
GO

ALTER TABLE [dbo].[Orcamento] WITH CHECK ADD CONSTRAINT [FK_Orcamento_Obra] FOREIGN KEY (
[ObraObraID]
)
REFERENCES [dbo].[Obra] (
[ObraID]
)
GO

ALTER TABLE [dbo].[DadosGuiaTransporte] WITH CHECK ADD CONSTRAINT [FKDadosGuiaT689485] FOREIGN
KEY (
[PerfilID]
, [GuiaTransporteGuiaTransporteID]
)
REFERENCES [dbo].[GuiaTransporte] (
[PerfilID]
, [GuiaTransporteID]
)
ALTER TABLE [dbo].[DadosGuiaTransporte] WITH CHECK ADD CONSTRAINT [FKDadosGuiaT366358] FOREIGN
KEY (
[EquipamentoEquipamentoID]
)
REFERENCES [dbo].[Equipamento] (
[EquipamentoID]
)
GO

ALTER TABLE [dbo].[Nota] WITH CHECK ADD CONSTRAINT [FK_Nota_Utilizador] FOREIGN KEY (
```

```
[UtilizadorID]

)

REFERENCES [dbo].[Utilizador] (

[UtilizadorID]

)

ALTER TABLE [dbo].[Nota] WITH CHECK ADD CONSTRAINT [FK_Nota_Obra] FOREIGN KEY (

[ObraID]

)

REFERENCES [dbo].[Obra] (

[ObraID]

)

GO


ALTER TABLE [dbo].[TipoEquipamento] WITH CHECK ADD CONSTRAINT [FK_TipoEquipamento_Utilizador]
FOREIGN KEY (

[TipoEquipamentoUtilizadorID]

)

REFERENCES [dbo].[Utilizador] (

[UtilizadorID]

)

GO


ALTER TABLE [dbo].[DadosOrcamento] WITH CHECK ADD CONSTRAINT [FK_DadosOrcamento_Orcamento]
FOREIGN KEY (

[OrcamentoOrcamentoID]

)

REFERENCES [dbo].[Orcamento] (

[OrcamentoID]

)

ALTER TABLE [dbo].[DadosOrcamento] WITH CHECK ADD CONSTRAINT [FK_DadosOrcamento_Componentes]
FOREIGN KEY (

[ComponenteID]

)
```

```
REFERENCES [dbo].[Componentes] (
[ComponentesID]
)
GO

ALTER TABLE [dbo].[Direito] WITH CHECK ADD CONSTRAINT [FK_Direito_Utilizador] FOREIGN KEY (
[UtilizadorID]
)
REFERENCES [dbo].[Utilizador] (
[UtilizadorID]
)
GO

GO

ALTER TABLE [dbo].[NotificacaoFluxo] WITH CHECK ADD CONSTRAINT [FK_NotificacaoFluxo_Utilizador]
FOREIGN KEY (
[UtilizadorID]
)
REFERENCES [dbo].[Utilizador] (
[UtilizadorID]
)
GO

ALTER TABLE [dbo].[Endereco] WITH CHECK ADD CONSTRAINT [FK_Endereco_Perfil] FOREIGN KEY (
[PerfilID]
)
REFERENCES [dbo].[Perfil] (
[PerfilID]
)
ALTER TABLE [dbo].[Endereco] WITH CHECK ADD CONSTRAINT [FK_Endereco_Paises] FOREIGN KEY (
[PaisesISO1]
```

```
)
REFERENCES [dbo].[Paises] (
[PaisesISO1]
)
GO

ALTER TABLE [dbo].[Obra] WITH CHECK ADD CONSTRAINT [FK_Obra_Equipamento] FOREIGN KEY (
[EquipamentoID]
)
REFERENCES [dbo].[Equipamento] (
[EquipamentoID]
)
ALTER TABLE [dbo].[Obra] WITH CHECK ADD CONSTRAINT [FK_Obra_Utilizador] FOREIGN KEY (
[UtilizadorID]
)
REFERENCES [dbo].[Utilizador] (
[UtilizadorID]
)
ALTER TABLE [dbo].[Obra] WITH CHECK ADD CONSTRAINT [FK_Obra_Spare] FOREIGN KEY (
[SpareID]
)
REFERENCES [dbo].[Spare] (
[SpareID]
)
GO

GO

/****** Object:  StoredProcedure [dbo].[ControlMaterialLimpezaDados]    Script Date: 06/28/2013 18:15:14 ******/
SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON
GO

-- ============================================
-- Author:                  <Rui Machado>
-- Create date: <11/04/2013>
-- Description:     <Limpa a base de dados>
-- ============================================
create PROCEDURE [dbo].[ControlMaterialLimpezaDados]
AS
BEGIN
        DELETE FROM Componentes;
        DELETE FROM Configuracao;
        DELETE FROM Contacto;
        DELETE FROM DadosGuiaTransporte;
        DELETE FROM GuiaTransporte;
        DELETE FROM DadosOrcamento;
        DELETE FROM Direito;
        DELETE FROM Endereco;
        DELETE FROM Nota;
        DELETE FROM NotificacaoFluxo;
        DELETE FROM Ocorrencias;
        DELETE FROM Orcamento;
        DELETE FROM Obra;
        DELETE FROM Spare;
        DELETE FROM Equipamento;
        DELETE FROM TipoEquipamento;
        DELETE FROM Perfil;
        DELETE FROM TipoEquipamento;
        DELETE FROM Utilizador;
END
```

```
GO



USE [ControleMaterial]
GO

/****** Object:  StoredProcedure [dbo].[ControlMaterialBackup]    Script Date: 06/28/2013 23:40:22 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO



-- =============================================
-- Author:                <Rui Machado>
-- Create date: <28/06/2013>
-- Description:    <backsupcontrolmaterial database to disk>
-- =============================================
CREATE PROCEDURE [dbo].[ControlMaterialBackup]
        -- Add the parameters for the stored procedure here
AS
BEGIN
        DECLARE @nomevarchar
        DECLARE @destinovarchar
        SELECT @nome= name FROM master.dbo.sysdatabases where name = 'ControleMaterial'
        SELECT @destino = 'C:\\BackupControlMaterial_' + CONVERT(VARCHAR(20),GETDATE(),112) +
REPLACE(CONVERT(VARCHAR(20),GETDATE(),108),':','') + '.bkp'
        BACKUP DATABASE @nome TO DISK = @destino
END
```

```
GO


USE [ControleMaterial]
GO

/****** Object:  Trigger [dbo].[Contacto_TipoContacto_Trigger]    Script Date: 06/28/2013 23:47:51 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

Create Trigger [dbo].[Contacto_TipoContacto_Trigger]
On [dbo].[Contacto]
For Insert, Update
As
Declare @tipoContactovarchar(255)
Select @tipoContacto = Inserted.TipoContacto
From Inserted
If @tipoContacto != 'EnderecoEletronico'
and @tipoContacto != 'Telefone'
and @tipoContacto != 'Telemovel'
Begin
        RollbackTransaction
        RAISERROR('Valor do campo TipoContacto tem de  ter um dos seguintes valores :
EnderecoEletronico,Telefone,Telemovel,',16,1)
End
GO
```

```
USE [ControleMaterial]
GO

/****** Object:  Trigger [dbo].[Endereco_EnderecoPrimario_Trigger]    Script Date: 06/28/2013 23:48:14 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

Create Trigger [dbo].[Endereco_EnderecoPrimario_Trigger]
On [dbo].[Endereco]
For Insert, Update
As
Declare @enderecoPrimarioint
Select @enderecoPrimario = Inserted.EnderecoPrimario
From Inserted
If @enderecoPrimario< 0
and @enderecoPrimario> 1
Begin
        RollbackTransaction
        RAISERROR('Valor do campo EnderecoPrimario tem de ser 0 ou 1',16,1)
End

GO


USE [ControleMaterial]
GO
```

```
/****** Object:  Trigger [dbo].[Endereco_TipoEndereco_Trigger]    Script Date: 06/28/2013 23:48:26 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

Create Trigger [dbo].[Endereco_TipoEndereco_Trigger]
On [dbo].[Endereco]
For Insert, Update
As
Declare @enderecoTipovarchar(255)
Select @enderecoTipo = Inserted.TipoEndereco
From Inserted
If @enderecoTipo != 'Residencia'
and @enderecoTipo != 'Emprego'
Begin
        RollbackTransaction
        RAISERROR('Valor do campo TipoEndereco tem de  ter um dos seguintes valores : Residencia,Emprego',16,1)
End

GO


USE [ControleMaterial]
GO

/****** Object:  Trigger [dbo].[Nota_Resolucao_Trigger]    Script Date: 06/28/2013 23:48:45 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
```

```
GO


Create Trigger [dbo].[Nota_Resolucao_Trigger]

On  [dbo].[Nota]

For Insert, Update

As

Declare @tipoint

Select @tipo = Inserted.Resolucao

From Inserted

If @tipo< 0

and @tipo> 1

Begin

        RollbackTransaction

        RAISERROR('Valor do campo Resolucao tem de ser 0 ou 1',16,1)

End


GO



USE [ControleMaterial]

GO


/****** Object:  Trigger [dbo].[NotificacaoFluxo_NotificacaoFluxoTipo_Trigger]    Script Date: 06/28/2013 23:48:59
******/

SET ANSI_NULLS ON

GO


SET QUOTED_IDENTIFIER ON

GO


CREATE Trigger [dbo].[NotificacaoFluxo_NotificacaoFluxoTipo_Trigger]

On [dbo].[NotificacaoFluxo]
```

```
For Insert, Update

As

Declare @notificacaoFluxoTipovarchar(255)

Select @notificacaoFluxoTipo = Inserted.NotificacaoFluxoTipo

From Inserted

If @notificacaoFluxoTipo != 'AberturaObra'

and @notificacaoFluxoTipo != 'FechoObra'

and @notificacaoFluxoTipo != 'SaidaObra'

and @notificacaoFluxoTipo != 'SparePendenteSaida'

Begin

        RollbackTransaction

        RAISERROR('Valor do campo NotificacaoFluxoTipo tem de  ter um dos seguintes valores :
AberturaObra,FechoObra,SaidaObra,SparePendenteSaida',16,1)

End

GO



USE [ControleMaterial]

GO


/****** Object:  Trigger [dbo].[Obra_ObraEstado_Trigger]    Script Date: 06/28/2013 23:49:16 ******/

SET ANSI_NULLS ON

GO


SET QUOTED_IDENTIFIER ON

GO


CREATE Trigger [dbo].[Obra_ObraEstado_Trigger]

On [dbo].[Obra]

For Insert, Update

As

Declare @obraEstadovarchar(255)
```

```
Select @obraEstado = Inserted.ObraEstado

From Inserted

If @obraEstado != 'Aberto'

and @obraEstado != 'Fechado'

and @obraEstado != 'Enviado'

Begin

        RollbackTransaction

        RAISERROR('Valor do campo TipoEndereco tem de  ter um dos seguintes valores :
Aberto,Enviado,Fechado',16,1)

End


GO



USE [ControleMaterial]

GO


/****** Object:  Trigger [dbo].[Obra_TipoEnvio_Trigger]    Script Date: 06/28/2013 23:49:24 ******/

SET ANSI_NULLS ON

GO


SET QUOTED_IDENTIFIER ON

GO


CREATE Trigger [dbo].[Obra_TipoEnvio_Trigger]

On [dbo].[Obra]

For Insert, Update

As

Declare @tipoEnviovarchar(255)

Select @tipoEnvio = Inserted.TipoEnvio

From Inserted

If @tipoEnvio != 'CTT'
```

```
 AND @tipoEnvio != 'Cliente'
Begin
        RollbackTransaction
        RAISERROR('Valor do campo TipoEnvio tem de  ter um dos seguintes valores : CTT,Cliente',16,1)
End

GO



USE [ControleMaterial]
GO

/****** Object:  Trigger [dbo].[Ocorrencias_OcorrenciaTipo_Trigger]    Script Date: 06/28/2013 23:49:36 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CreateTrigger [dbo].[Ocorrencias_OcorrenciaTipo_Trigger]
On [dbo].[Ocorrencias]
For Insert, Update
As
Declare @tipovarchar(100)
Select @tipo = Inserted.OcorrenciaTipo
From Inserted
If @tipo != 'EntradaSistema'
and @tipo != 'SaidaSistema'
and @tipo != 'RegistoCriado'
and @tipo != 'RegistoAlterado'
and @tipo != 'RegistoApagado'
Begin
```

```
        RollbackTransaction
        RAISERROR('Valor do campo OcorrenciaTipo tem de ter um dos seguintes valores :
EntradaSistema,SaidaSistema,RegistoCriado,RegistoAlterado,RegistoApagado',16,1)
End

GO



USE [ControleMaterial]
GO

/****** Object:  Trigger [dbo].[Orcamento_Aceite_Trigger]    Script Date: 06/28/2013 23:49:48 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

Create Trigger [dbo].[Orcamento_Aceite_Trigger]
On [dbo].[Orcamento]
For Insert, Update
As
Declare @Aceitevarchar(255)
Select @Aceite = Inserted.Aceite
From Inserted
If @Aceite != 'Pendente'
 AND @Aceite != 'Sim'
 AND @Aceite != 'Não'
Begin
        RollbackTransaction
        RAISERROR('Valor do campo  Aceite tem de  ter um dos seguintes valores : Pendente,Sim,Não',16,1)
End
```

```
GO


USE [ControleMaterial]
GO

/****** Object:  Trigger [dbo].[Paises_ReadOnly_Trigger]    Script Date: 06/28/2013 23:50:01 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

Create Trigger [dbo].[Paises_ReadOnly_Trigger]
On [dbo].[Paises]
For Insert, Update, Delete
As
Begin
	RollbackTransaction
	RAISERROR('Tabela apenas de leitura.',16,1)
End
GO


USE [ControleMaterial]
GO

/****** Object:  Trigger [dbo].[Perfil_PerfilTipoArmazem_Trigger]    Script Date: 06/28/2013 23:50:16 ******/
SET ANSI_NULLS ON
GO
```

```
As

Declare @utilizadorActivoint

Select @utilizadorActivo = Inserted.UtilizadorActivo

From Inserted

If @utilizadorActivo< 0

and @utilizadorActivo> 1

Begin

        RollbackTransaction

        RAISERROR('Valor do campo UtilizadorActivo tem de ser 0 ou 1',16,1)

End


GO
```

```
SET QUOTED_IDENTIFIER ON

GO


Create Trigger [dbo].[Perfil_PerfilTipoArmazem_Trigger]

On  [dbo].[Perfil]

For Insert, Update

As

Declare @tipoint

Select @tipo = Inserted.PerfilTipoArmazem

From Inserted

If @tipo< 0

and @tipo> 1

Begin

        Rollback Transaction

        RAISERROR('Valor do campo PerfilTipoArmazem tem de ser 0 ou 1',16,1)

End

GO



USE [ControleMaterial]

GO


/****** Object:  Trigger [dbo].[Utilizador_UtilizadorActivo_Trigger]    Script Date: 06/28/2013 23:50:34 ******/

SET ANSI_NULLS ON

GO


SET QUOTED_IDENTIFIER ON

GO


CreateTrigger [dbo].[Utilizador_UtilizadorActivo_Trigger]

On [dbo].[Utilizador]

For Insert, Update
```