

Relatório do Projeto Final

Gestão Incidentes

Introdução

Objetivo

Este projeto tem como objetivo desenvolver uma aplicação á medida para a resolução de incidentes de IT.

Glossário

O Projeto envolve um conjunto de conceitos e termos que podem não ser do conhecimento geral.

1. Incidente – Um Serviço, Equipamento ou Programa não está a funcionar ou não está a funcionar ao nível esperado.
2. Colaborador – Funcionários da empresa que reportam os incidentes á equipe de IT Support.
3. Técnico – Funcionários da empresa que resolvem incidentes.
4. Ticket – Processo de tratamento do incidente, é criado por um colaborador e resolvido por um técnico, um ticket tem estados de resolução, uns são finais outros não.
5. Requisição de material – Um Ticket pode necessitar de equipamento, software ou serviços adicionais que a equipe de suporte não pode disponibilizar diretamente, a requisição serve para registar essas necessidades adicionais, cada requisição só suporta um item, pelo que será necessária uma requisição por item, Ex: um PC tem duas peças avariadas, serão necessárias duas requisições. Este processo poderá ter uma evolução futura para registar mais que um item, mas só será implementado se for verificada a necessidade por parte do departamento.
6. Prioridade, definida pelo utilizador, serve para organizar os incidentes por prioridade, permitindo uma melhor gestão do tempo dos técnicos e dos Incidentes, visto que uma maior prioridade vai determinar o seu tempo de execução.
7. Processo humano, qualquer programa implica um conjunto de regras de utilização, neste programa o processo humano é importante para que o sucesso da aplicação, determinado pela organização que o usa, o processo humano deve acompanhar as regras de utilização da interface da aplicação. Ex: cada requisição só lida com um item, ora se no campo da descrição forem colocados mais que um item, obviamente a aplicação não se vai queixar, mas deve o processo humano intervir aí e determinar que essa utilização do programa não é aconselhada.
8. Inventário – Todos os equipamentos da empresa devem estar registados no inventário, isso ajuda o programa determinar o tipo de intervenção, no registo dos equipamentos deve ser o mais detalhado possível e todos os campos devem ser preenchidos, os equipamentos têm peças . Ex: um computador pode ser registado com um único numero ou ser dividido em caixa do PC, monitor, programas instalados, inclusive mesmo o teclado e o rato podem ser registados em separado, deixando de ser peças de um PC para passar a ser um equipamento.

Desenvolvimento

Arquitetura do Programa

O Programa é desenvolvido em C# com uma base de dados Local em SQL Server, pode ser usado num computador local ligado a um servidor de Base de Dados local, ou ligado a um servidor de base de dados remoto.

Devido ás exigências do cliente expressas ao analista este programa não permite usar outro tipo de bases de dados, no entanto a sua arquitetura permite a migração para outro tipo de bases de dados com pouca alteração de código, que será explicado nas escolhas feitas para as classes.

O programa tenta observar o melhor possível as regras de Programação orientada por objetos, OOP, a reutilização de código é uma regra constante e tenta-se evitar a redundância ao máximo.

Classes

Este Programa tem um Conjunto de Classes que expõe a base de dados à Interface Gráfica, essas classes estão divididas em classes de Entidades e Classe de Interface, não confundir esta interface com a interface gráfica, esta Classe de Interface é a responsável pela comunicação entre a base de dados e o programa, chamou-se a esta Classe BDTickes, será feita uma descrição mais detalhada no Diagrama de Classes

As Entidades identificadas foram: Tickets, Perfis, MaterialRequests, Equipamento, Material, Status, Habilitacao, Colaborador, Tecnico, SuporteEquipamento, SuporteSoftware, SuporteRede, Area.

Estas classes definem as varias entidades da aplicação e permitem a interação entre elas através da classe BDTicktes.

A Classe BDTicktes está a herdar um Interface aplicacional que permite a expansão futura da aplicação para uma solução com diferentes interfaces conforme a necessidade.

Diagrama de Classes

Base de Dados

A base de dados foi estruturada conforme o modelo relacional, respeitando as três principais regras da normalização de classes, Exemplo da Normalização.

Conclusão