



SISTEMA DE GESTÃO DE OCORRÊNCIAS E RECURSOS

FireCare - Relatório

Síntese

Sistema de gestão de ocorrências de emergência, permitindo o registro e alocação de recursos em tempo real.

A20469@alunos.ipca.pt
ENTREGA PRIMEIRA FASE

Relatório de Projeto - Sistema de Gestão de Ocorrências e Recursos

Este relatório descreve o desenvolvimento do projeto FireCare, um sistema para gestão de ocorrências, profissionais e equipamentos em cenários de emergência. O sistema foi projetado com base em princípios de Programação Orientada a Objetos (POO) e visa facilitar o acompanhamento e gestão de recursos em tempo real, permitindo o registro e alocação de profissionais e equipamentos nas ocorrências registradas.

Conteúdo

Relatório de Projeto - Sistema de Gestão de Ocorrências e Recursos	1
1. Ambiente de Desenvolvimento	2
2. Estrutura e Organização do Projeto	3
Tabela de Classes e Relações.....	3
4. Funcionalidades Futuras e Expansão	7
5. Dificuldades Encontradas.....	8
5.1 Configurações de Ambiente.....	8
5.2 Problemas de UI/UX.....	8
Considerações Finais.....	9

1. Ambiente de Desenvolvimento

O projeto foi desenvolvido em C# utilizando o .NET Framework, SQLite para Base de Dados. O design foi construído com Windows Forms para permitir a interação e visualização dos dados. O ambiente de desenvolvimento utilizado foi o Visual Studio 2022 num sistema operativo Windows 10, garantindo compatibilidade com a tecnologia .NET e a execução eficiente das operações CRUD para gestão dos dados.

As operações CRUD - Create (Criar), Read (Ler), Update (Atualizar) e Delete (Apagar) - são fundamentais para o sistema, permitindo adicionar, visualizar, atualizar e eliminar dados das principais entidades: Ocorrências, Profissionais e Equipamentos. Estas operações são realizadas através dos métodos implementados nos serviços (Pasta Services) específicos para cada entidade, garantindo uma estrutura consistente e organizada.

O GitHub foi utilizado para fazer a gestão e controlo de versões do meu desenvolvimento, conforme o meu avanço e sempre que adicionava uma nova funcionalidade relevante o projeto era enviado para o meu repositório com informação detalhada sobre as últimas alterações.

Foi um recurso importante também para ter um ponto de *backup* sempre que algo no projeto acabava com alguma falha devido a uma implementação errada.

[Link](#) para o projeto.

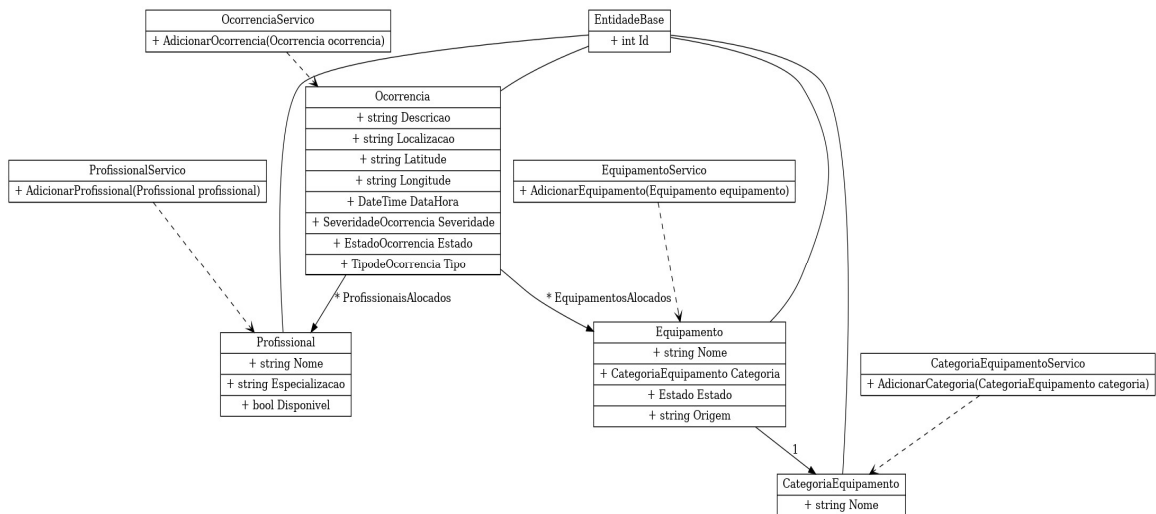
2. Estrutura e Organização do Projeto

O projeto foi estruturado com base em princípios de Programação Orientada a Objetos (POO), utilizando classes para representar as diferentes entidades do sistema, como Ocorrência, Profissional e Equipamento. Abaixo, são detalhadas as principais classes e suas relações.

Tabela de Classes e Relações

Classe	Descrição	Relações
EntidadeBase	Classe base para as principais entidades do sistema, fornecendo um identificador único (ID) para cada instância. É a superclasse de todas as entidades (Ocorrencia, Profissional, Equipamento, CategoriaEquipamento).	Superclasse de Ocorrencia, Profissional, Equipamento, CategoriaEquipamento.
Ocorrencia	Representa uma ocorrência de emergência com detalhes como localização, severidade, estado, tipo e recursos alocados. Relacionada com Profissional e Equipamento através das tabelas de ligação Ocorrencias_Profissionais e Ocorrencias_Equipamentos.	Relacionada com Profissional e Equipamento através das tabelas Ocorrencias_Profissionais e Ocorrencias_Equipamentos.
Profissional	Representa um profissional alocado para atender a uma ocorrência, com atributos como nome, especialização e disponibilidade. Associada a Ocorrencia através da tabela de ligação Ocorrencias_Profissionais.	Associada a Ocorrencia através da tabela Ocorrencias_Profissionais.
Equipamento	Representa equipamentos usados nas ocorrências, como veículos e materiais de suporte. Inclui informações sobre estado e tipo de origem. Associada a Ocorrencia através da tabela de ligação Ocorrencias_Equipamentos.	Associada a Ocorrencia através da tabela Ocorrencias_Equipamentos.

CategoriaEquipamento	Classifica os tipos de equipamentos, facilitando a organização e alocação de recursos específicos. Relacionada com Equipamento como categoria, permitindo atribuir categorias aos equipamentos.	Relacionada com Equipamento como categoria.
OcorrenciaServico	Classe de serviço responsável por operações CRUD nas ocorrências, incluindo registro, atualização e exclusão de ocorrências na Base de dados. Opera diretamente com Ocorrencia e a Base de Dados.	Opera diretamente com Ocorrencia e Base de Dados.
ProfissionalServico	Classe de serviço para manipular dados de profissionais, permitindo adicionar, editar e excluir profissionais. Opera diretamente com Profissional e a Base de dados.	Opera diretamente com Profissional e Base de Dados.
EquipamentoServico	Classe de serviço que manipula dados de equipamentos, incluindo o estado de estado, origem e categoria dos equipamentos. Opera diretamente com Equipamento e a Base de Dados.	Opera diretamente com Equipamento e Base de Dados.
CategoriaEquipamentoServico	Fornece métodos para gerenciar categorias de equipamentos, permitindo listar, adicionar e excluir categorias. Opera com CategoriaEquipamento e o Base de Dados.	Opera com CategoriaEquipamento e Base de Dados.



EntidadeBase

É a classe base para todas as entidades principais do sistema, fornecendo um identificador único Id (int) que será herdado por outras classes. Esta abordagem permite que todas as entidades derivadas compartilhem um identificador comum, simplificando a gestão e manipulação de dados no sistema.

Ocorrencia

A classe Ocorrencia representa ocorrências de emergência e contém atributos como Descricao, Localizacao, Latitude, Longitude, DataHora, Severidade (com valores Baixa, Media, Alta), Estado (com valores Ativo, Resolvido) e Tipo (com valores como Incendio, Resgate, etc.).

Ocorrencia tem uma relação direta com as classes Profissional e Equipamento, por meio de tabelas de ligação (Ocorrencias_Profissionais e Ocorrencias_Equipamentos). Essas relações representam as alocações de profissionais e equipamentos a cada ocorrência, o que é essencial para a gestão de recursos em emergências.

Profissional

A classe Profissional representa os profissionais que podem ser alocados para uma ocorrência. Seus atributos incluem Nome, Especializacao e Disponivel (indicando a disponibilidade do profissional para atendimento).

A relação com a classe Ocorrencia é estabelecida através da tabela de ligação Ocorrencias_Profissionais, permitindo que cada ocorrência tenha vários profissionais atribuídos e que cada profissional possa atender a múltiplas ocorrências.

Equipamento

Equipamento representa os recursos materiais utilizados nas ocorrências, como veículos ou ferramentas. Os atributos incluem Nome, IdCategoria (que se relaciona com CategoriaEquipamento), Estado (Disponivel, Indisponivel, Manutencao) e Origem.

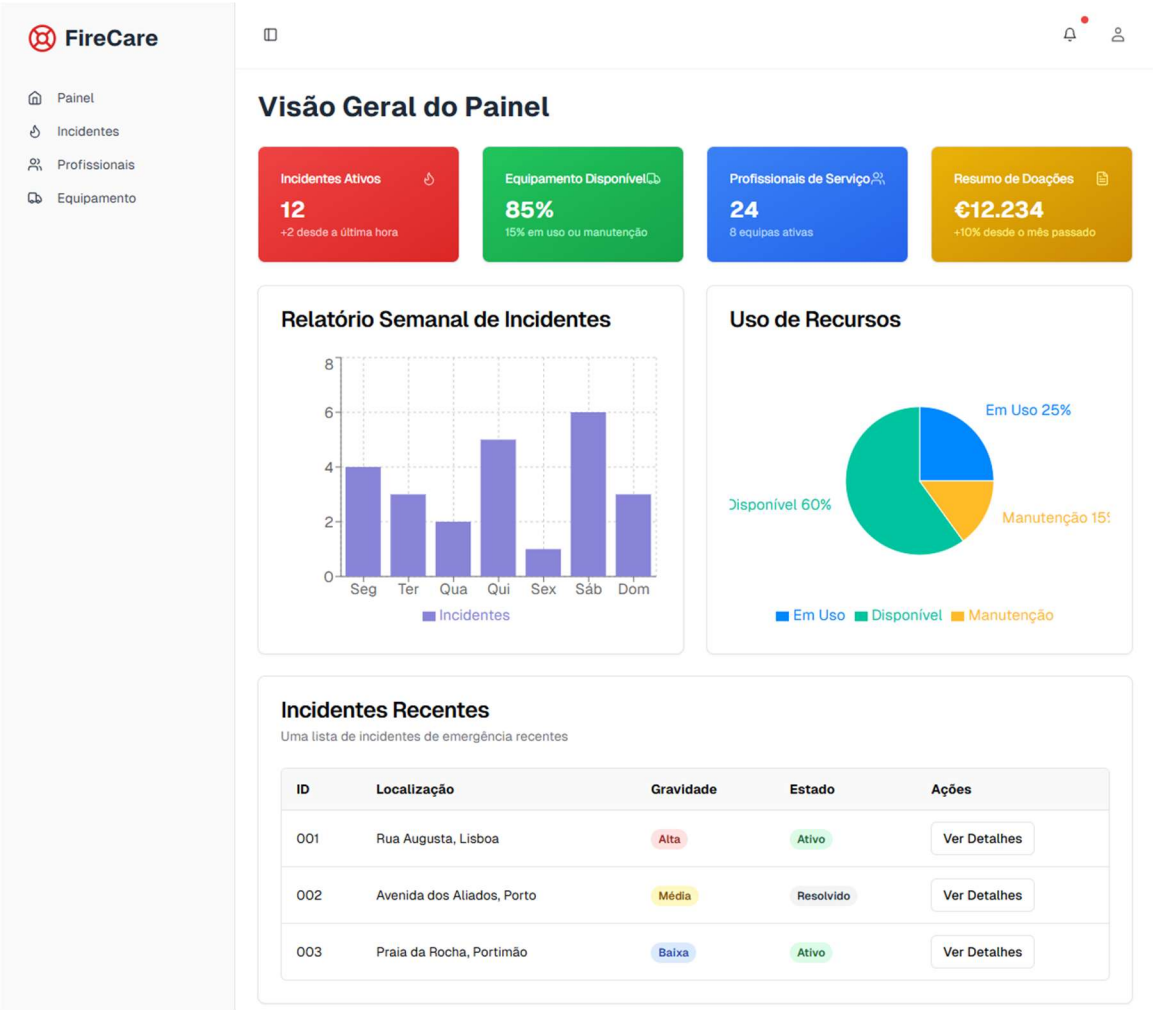
A relação entre Equipamento e Ocorrencia é gerida pela tabela Ocorrencias_Equipamentos, permitindo a alocação de múltiplos equipamentos a uma única ocorrência e vice-versa.

A classe Equipamento também relaciona com CategoriaEquipamento para definir a categoria de cada equipamento, como veículos, ferramentas, etc.

4. Funcionalidades Futuras e Expansão

No futuro, a expansão do projeto incluirá a criação de *website* em React para exibir o dashboard em tempo real das ocorrências e recursos. Hospedado no meu servidor caseiro, vai permitir visualizar ocorrências ativas, os profissionais e equipamentos envolvidos, e acompanhar a localização dos recursos em tempo real.

Esta melhoria visa tornar o FireCare uma ferramenta de gestão de emergências mais acessível e moderna, ao integrar funcionalidades de monitorização e análise em tempo real. Todos os dados do *website* serão integrados a partir da Base de Dados



5. Dificuldades Encontradas

5.1 Configurações de Ambiente

Durante o desenvolvimento do projeto FireCare, foram enfrentadas várias dificuldades relacionadas ao ambiente de programação:

- Limitações de MacOS para Windows Forms: Tendo um portátil MacBook, era inviável utilizá-lo para programar em C# com Windows Forms, pois a Microsoft abandonou o suporte ao Visual Studio 2022 para Mac no que se refere a aplicações de Windows Forms. Esta limitação dificultou o desenvolvimento no ambiente MacOS.

- Servidor Caseiro e Proxmox: Como alternativa, utilizei o meu servidor caseiro, um desktop montado com componentes da China, para funcionar como um labserver pessoal. Este servidor executa Docker, VPN, NAS, servidor de média, e mais recentemente, o Proxmox - uma plataforma de virtualização de código aberto que facilita a criação e gestão de máquinas virtuais (VMs). Utilizei o Proxmox para configurar uma VM com Windows 10 otimizado, permitindo um ambiente leve e eficiente para desenvolvimento.

Para aceder ao meu servidor na escola, configurei uma VPN OpenVPN. Contudo, o serviço de internet da universidade bloqueava conexões VPN, o que me obrigou a adquirir uma VPS Windows para continuar o desenvolvimento através do Windows Remote Desktop, eliminando as restrições impostas pela infraestrutura da universidade.

5.2 Problemas de UI/UX

O desenvolvimento da interface de utilizador (UI) e a experiência de utilizador (UX) apresentaram desafios significativos. Utilizar o Windows Forms e o WPF no C# limitou as possibilidades de design visual e usabilidade. As ferramentas nativas de UI/UX para C# no Windows carecem de modernização e tornam difícil criar interfaces atrativas e responsivas.

Sendo utilizador do Visual Studio desde a sua versão de 2017 não encontrei melhorias significativas nestes anos que se sucederam, as ferramentas nativas podem tornar o desenvolvimento monótono e, por vezes, frustrante. Daí ter considerado a migração para uma dashboard web utilizando React, mais moderno e interativo, proporcionando um design responsivo e acesso simplificado em qualquer dispositivo.

Considerações Finais

O sistema FireCare oferece uma plataforma organizada e eficiente para a gestão de ocorrências, profissionais e equipamentos em cenários de emergência. Utilizando princípios de Programação Orientada a Objetos, o projeto é modular e de fácil manutenção. As melhorias futuras, como o dashboard em React para visualização em tempo real, prometem elevar a funcionalidade do sistema, proporcionando uma solução completa e moderna para a gestão de emergências e recursos em tempo real.