

## Relatório RAD - Sistema de Gerenciamento de Expedições Espaciais

Equipe: Douglas Oliveira Biserra – Derik Lima de Oliveira – Rennan Mariano Oliveira

Matrícula: 201912029626 - 202308424726 - 202309762277

Projeto: Sistema de Gerenciamento de Expedições Espaciais

Ferramenta: Flask (Back-end API)

Data: 02/11/2024

### Introdução

Este projeto tem como objetivo desenvolver um sistema para o gerenciamento de expedições espaciais, facilitando o planejamento, monitoramento e execução de missões. O sistema permite cadastrar missões, visualizar informações detalhadas, atualizar registros e deletar missões com base em critérios específicos, além de realizar buscas por intervalo de datas.

### Escopo do Projeto

- Cadastro de missões espaciais com dados relevantes.
- Listagem de missões em ordem decrescente de data de lançamento.
- Recuperação dos detalhes de uma missão específica.
- Edição e exclusão de missões cadastradas.
- Busca de missões por intervalos de data de lançamento.

### Metodologia

O desenvolvimento seguiu a metodologia RAD (Rapid Application Development), que valoriza a agilidade e a construção iterativa. O projeto foi dividido nas seguintes fases:

## Fase 1: Planejamento dos Requisitos

### Objetivo

Estabelecer as funcionalidades principais do sistema com base nas necessidades do cliente.

### Atividades

1. Levantamento dos requisitos fornecidos pela professora, incluindo:

- Operações CRUD completas para missões espaciais.
- Listagem de missões com ordenação por data de lançamento.
- Pesquisa de missões por intervalo de datas.

2. Definição do modelo de dados para armazenamento das missões.

### Resultados

- Lista de requisitos do sistema, incluindo as operações CRUD e busca por data.
- Definição inicial da estrutura da tabela de missões no banco de dados.

## Fase 2: Prototipação

### Objetivo

Criar uma estrutura básica do sistema para validar os requisitos principais.

## Atividades

1. Configuração do ambiente de desenvolvimento em Flask.
2. Criação de um modelo de dados (`Missao`) para armazenar os dados da missão com os seguintes campos:

- `id`, `nome`, `data\_lancamento`, `data\_retorno`, `destino`, `estado\_missao`, `tripulacao`, `carga\_util`, `duracao`, `custo`, `status\_detalhado`.

3. Implementação inicial das rotas de API para:

- `POST /missao` - criação de uma nova missão.
- `GET /missoes` - listagem de todas as missões.
- `GET /missao/<id>` - recuperação dos detalhes de uma missão específica.
- `PUT /missao/<id>` - atualização dos dados de uma missão.
- `DELETE /missao/<id>` - exclusão de uma missão.

## Resultados

- Estrutura básica da API criada com rotas CRUD.
- Protótipo inicial disponível para testes de funcionalidade.

## Fase 3: Desenvolvimento Iterativo

### Objetivo

Expandir e melhorar o sistema com base no feedback e nas revisões dos requisitos.

## Atividades

1. Ajustes nas Rotas CRUD:

- Validação e tratamento de exceções em todas as rotas.
- Ordenação da listagem de missões por `data\_lancamento` em ordem decrescente.

## 2. Implementação de Pesquisa por Intervalo de Datas

- Criação da rota `GET /missoes/pesquisa`, permitindo filtrar missões por um intervalo de datas (`data\_inicio` e `data\_fim`).

## 3. Testes com Ferramentas de API:

- Teste das rotas usando o Thunder Client, validando as respostas e identificando possíveis falhas.
- Ajustes finais no código com base nos resultados dos testes.

## Resultados

- API completamente funcional com operações CRUD e pesquisa por intervalo de datas.

## Fase 4: Implementação e Conclusão

### Objetivo

Finalizar o desenvolvimento e realizar os últimos ajustes antes da entrega.

### Atividades

1. Revisão final do código e das rotas para garantir consistência e tratamento adequado de exceções.
2. Documentação das rotas e dos parâmetros esperados em cada endpoint.
3. Geração de um arquivo `.zip` contendo:
  - Código-fonte do projeto.

- Relatório RAD.

## Conclusão

A metodologia RAD permitiu desenvolver rapidamente uma solução funcional para o Sistema de Gerenciamento de Expedições Espaciais. A abordagem iterativa facilitou a adaptação às necessidades ao longo do desenvolvimento, garantindo que os requisitos fossem atendidos com eficiência.

## Anexos

### - Detalhamento das Rotas e Parâmetros:

- `POST /missao` - Criação de missões.
- `GET /missoes` - Listagem de missões.
- `GET /missao/<id>` - Detalhes de uma missão específica.
- `PUT /missao/<id>` - Atualização de missão.
- `DELETE /missao/<id>` - Exclusão de missão.
- `GET /missoes/pesquisa?data\_inicio=&data\_fim=` - Busca por intervalo de datas.