

Projeto Interdisciplinar das disciplinas: Banco de Dados Não Relacional / Desenvolvimento Web III
Sistema: Restify – API Client simplificado

Nome aluno (a): Gabriel da Silveira Pessoni

Nome aluno (a): Luis Fernando Mendes

1. Introdução

Propósito (ou Finalidade)

Este documento apresenta a modelagem do sistema Restify, uma ferramenta projetada para facilitar o teste, monitoramento e gerenciamento de requisições a APIs REST. A modelagem aqui descrita segue as diretrizes estabelecidas no livro UML na Prática – Do Problema ao Sistema, oferecendo uma visão clara e estruturada das funcionalidades e componentes do sistema.

O público-alvo deste documento inclui desenvolvedores, analistas de sistemas, profissionais de QA e equipes de DevOps que necessitam de uma solução eficiente para testar e validar APIs REST. Além disso, este documento também pode servir como referência para partes interessadas que buscam compreender a arquitetura e os fluxos de trabalho associados ao Restify.

Escopo

O projeto tem como finalidade o desenvolvimento de um cliente de API REST simplificado, projetado para oferecer uma experiência eficiente e intuitiva no envio, monitoramento e validação de requisições HTTP. A solução incluirá uma aplicação web, um backend robusto e uma versão desktop para testes offline.

O sistema permitirá que os usuários executem requisições HTTP (GET, POST, PUT, DELETE, PATCH), gravem logs detalhados de cada interação e gerem relatórios para análise do comportamento das APIs testadas. Além disso, a ferramenta oferecerá suporte a testes automatizados de endpoints, testes de estresse e gerenciamento de coleções de requisições, proporcionando uma visão estruturada dos testes realizados.

Para os desenvolvedores e equipes de QA, o Restify fornecerá uma interface intuitiva que permitirá a configuração de autenticação (Bearer Token, API Keys, OAuth 2.0, Basic Auth) e a execução de testes automatizados para validação de respostas. Além disso, a ferramenta contará com um histórico de requisições para facilitar a análise de interações passadas e otimizar o fluxo de trabalho. Com essas funcionalidades, o Restify busca oferecer uma solução completa para o gerenciamento de requisições e testes de APIs REST, garantindo maior confiabilidade, eficiência e controle sobre os serviços testados.

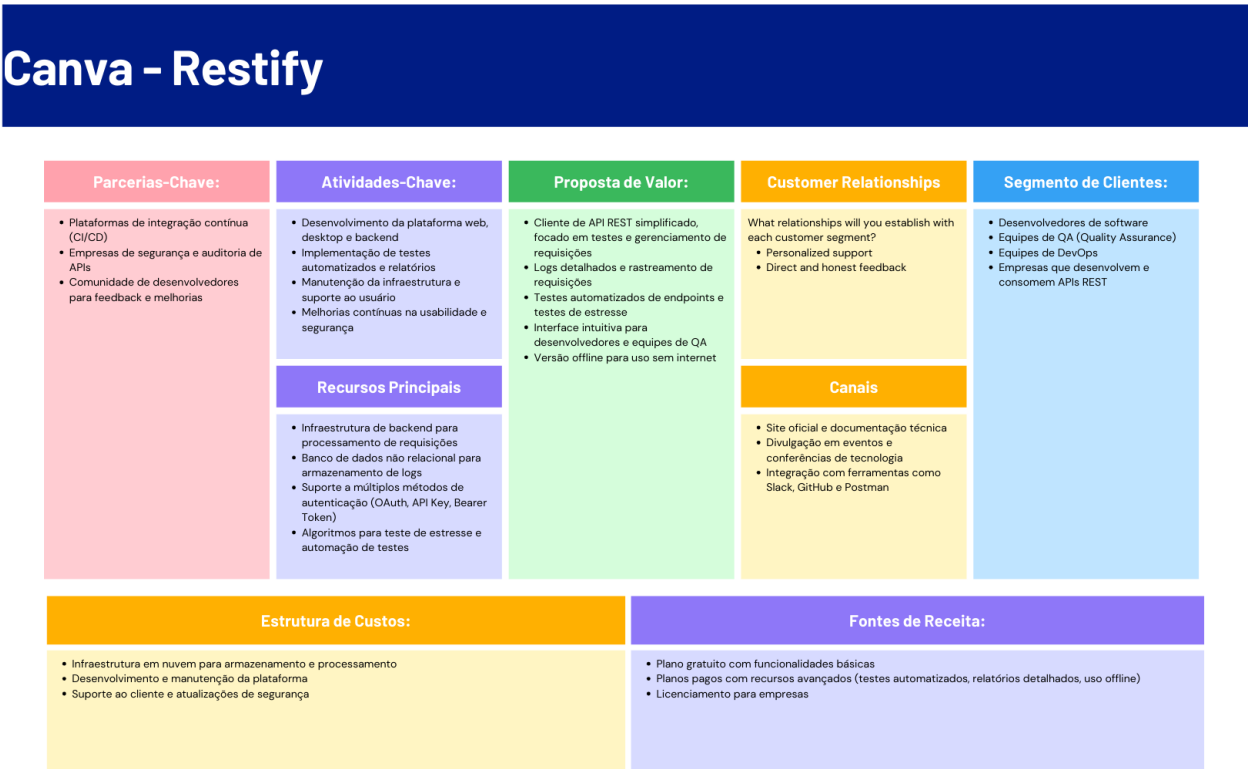
Benefícios Esperados

A implementação do Restify traz inúmeras vantagens para desenvolvedores, equipes de QA e DevOps, abordando desafios comuns enfrentados no teste e validação de APIs REST.

Atualmente, o teste de APIs muitas vezes envolve ferramentas dispersas, documentação fragmentada e dificuldades na rastreabilidade das requisições. Isso pode levar a falhas em produção, baixa confiabilidade e dificuldades na reprodução de erros. O Restify visa solucionar esses problemas por meio de:

- **Automatização de testes:** Permite a criação e execução de testes automatizados para validar o comportamento de APIs, economizando tempo e reduzindo erros humanos.
- **Execução e monitoramento eficientes:** Com logs detalhados e relatórios de requisições, os usuários podem rastrear interações e analisar padrões de comportamento das APIs.
- **Testes de estresse:** A ferramenta simula cenários de carga para avaliar o desempenho de APIs sob alto tráfego, auxiliando na detecção de gargalos e na melhoria da escalabilidade.

Modelo Canva



3. Tecnologias Utilizadas

Frontend:

O frontend web do Restify será construído utilizando o Next.js, um framework robusto baseado em React que oferece renderização híbrida (SSR e SSG), melhoria no desempenho e melhor indexação para SEO. Para estilização e design, será utilizado o Tailwind CSS, um framework utilitário que permite construir interfaces responsivas e elegantes de forma ágil. Essa combinação garante uma experiência fluida e otimizada para usuários que acessam a plataforma via navegador.

Frontend Desktop:

Para a versão desktop do Restify, será utilizado o Electron, um framework que permite criar aplicações desktop multiplataforma (Windows, macOS e Linux) utilizando tecnologias web (HTML, CSS e JavaScript). O Electron possibilita a integração direta com APIs nativas do sistema operacional, permitindo que a ferramenta funcione offline e tenha acesso a funcionalidades como armazenamento local e gerenciamento de processos.

Backend:

O backend será desenvolvido com **Node.js**, uma plataforma altamente eficiente para aplicações assíncronas e em tempo real, garantindo alto desempenho na execução de requisições. Para a camada de acesso ao banco de dados, será utilizado o **Prisma**, um ORM moderno que simplifica a manipulação de dados, oferecendo suporte a tipos estáticos e integração simplificada com o MongoDB. Essa escolha melhora a produtividade do desenvolvimento e reduz a complexidade do gerenciamento de banco de dados.

Banco de Dados:

O sistema utilizará o **MongoDB**, um banco de dados NoSQL altamente escalável, ideal para armazenar logs de requisições e histórico de testes de APIs. Como um banco orientado a documentos, o MongoDB permite flexibilidade na modelagem de dados e oferece alto desempenho para operações de leitura e escrita, tornando-o adequado para os requisitos do Restify. Com essa stack tecnológica, o Restify será uma solução moderna, eficiente e flexível, garantindo uma experiência otimizada para desenvolvedores e equipes de QA no teste e monitoramento de APIs REST.

4. Funcionalidades Principais

1. Execução de Requisições HTTP

O Restify permite o envio de requisições HTTP nos métodos mais comuns: GET, POST, PUT, DELETE e PATCH. Os usuários podem configurar os cabeçalhos, parâmetros, corpo da requisição e autenticação de forma flexível, garantindo compatibilidade com qualquer API REST.

2. Gravação e Gerenciamento de Logs

Cada requisição enviada é armazenada em um banco de dados MongoDB, permitindo que os usuários acessem logs detalhados de todas as interações com APIs. Essa funcionalidade é essencial para rastrear erros, analisar respostas e identificar padrões de comportamento.

3. Testes Automatizados de Endpoints

A ferramenta permite a criação de testes automatizados para validar APIs, verificando códigos de status, estrutura do JSON e tempo de resposta. Os testes podem ser executados manualmente ou programados para rodar automaticamente, garantindo maior confiabilidade nos endpoints testados.

4. Teste de Estresse e Carga

Para avaliar a escalabilidade e o desempenho de APIs, o Restify conta com um módulo de testes de estresse, que simula múltiplas requisições simultâneas. Isso ajuda a identificar gargalos e otimizar a performance dos serviços testados.

5. Gerenciamento de Coleções de Requisições

Os usuários podem organizar requisições em coleções reutilizáveis, facilitando a estruturação de testes e possibilitando a execução em lote. Isso torna o fluxo de trabalho mais eficiente e melhora a colaboração entre equipes.

6. Suporte a Diversos Métodos de Autenticação

O Restify oferece suporte a múltiplos métodos de autenticação, incluindo Bearer Token, API Keys, OAuth 2.0 e Basic Auth, permitindo a integração com APIs protegidas sem complicações.

7. Histórico de Requisições e Reexecução Rápida

O sistema mantém um histórico completo de requisições, possibilitando a reexecução rápida de chamadas anteriores. Isso facilita a depuração e acelera os testes recorrentes.

8. Geração de Relatórios e Exportação de Dados

Os usuários podem gerar relatórios detalhados das requisições e testes realizados, exportando os dados em formatos como CSV e JSON. Isso facilita a análise do consumo das APIs e a documentação de testes realizados.