

Documento de Especificação do Projeto
RPA Boettscher

RPA BOETTSCHER

Feito por: Pierry Boettscher
Curso: Engenharia de Software

Data: 13/12/2024

Documento de Especificação do Projeto

RPA Boettscher

Sumário

1	Introdução	3
1.1.	Contexto	3
1.2.	Justificativa	3
1.3.	Objetivos	3
2	Descrição do Projeto	3
2.1.	Tema do Projeto	3
2.2.	Problemas a Resolver	4
2.3.	Limitações	4
3	Especificação Técnica	4
3.1.	Requisitos de Software	4
3.2.	Considerações de Design	7
3.3.	Stack Tecnológica	10
3.4.	Considerações de Segurança	12
4	Próximos Passos	12
4.1.	Portifólio I:	12
4.2.	Portifólio II:	12
5	Referências	13
5.1.	Frameworks e Bibliotecas.....	13
5.2.	Ferramentas	13
5.3.	Serviços AWS.....	13
6	Considerações dos Professores.....	13
6.1.	Professor(a) 1:.....	13
6.2.	Professor(a) 2:.....	13
6.3.	Professor(a) 3:.....	13

Documento de Especificação do Projeto

RPA Boettscher

1 Introdução

1.1. Contexto

O desenvolvimento tecnológico contínuo tem aumentado a demanda por automação de processos repetitivos em diversas áreas. A Automação de Processos Robóticos (RPA) emerge como uma solução eficaz, permitindo que software ou "robôs" simulem ações humanas em interfaces digitais para executar uma vasta gama de tarefas sem intervenção humana. Este projeto se insere nesse contexto, focando no desenvolvimento e implementação de soluções RPA para otimizar processos em ambientes corporativos.

1.2. Justificativa

A relevância deste projeto para o campo da engenharia de software reside na sua capacidade de demonstrar como a RPA pode ser aplicada para aumentar a eficiência, reduzir erros e liberar recursos humanos para tarefas mais complexas e criativas. Por exemplo, um estudante da Católica SC que precisa entrar no site da faculdade mensalmente para gerar o boleto de pagamento da sua mensalidade, com o uso da tecnologia RPA, podemos gerar o boleto automaticamente para ele. A implementação de soluções RPA não apenas otimiza operações, mas também proporciona insights valiosos sobre a integração de tecnologias emergentes no desenvolvimento de software.

1.3. Objetivos

Desenvolver uma solução RPA customizada que atenda às necessidades específicas do processo selecionado, demonstrando a viabilidade e eficácia da automação.

2 Descrição do Projeto

2.1. Tema do Projeto

Este projeto visa o desenvolvimento de uma solução integrada de Automação de Processos Robóticos (RPA) que permita aos usuários gerarem relatórios atualizados da Tabela FIPE e realizar downloads de boletos em PDF diretamente do site de uma instituição de ensino superior. A solução será acessada por meio de uma página web, onde o usuário poderá escolher entre as funcionalidades disponíveis. Uma API intermediará a comunicação entre a interface web e os executáveis dos robôs RPA na máquina do usuário, garantindo a execução das tarefas selecionadas de forma eficiente e segura.

Documento de Especificação do Projeto

RPA Boettcher

2.2. Problemas a Resolver

Geração de Relatório da Tabela FIPE: Desenvolver um robô RPA capaz de acessar os dados mais recentes da Tabela FIPE, gerar um relatório personalizado e disponibilizá-lo como um arquivo Excel com todos os registros.

Download de Boletos de Mensalidade: Implementar uma solução RPA que automatize o processo de login no portal da Católica SC, localize o boleto de mensalidade mais recente e o baixe em formato PDF, facilitando o acesso do usuário ao documento.

Integração Web-to-RPA: Criar uma API robusta que facilite a comunicação entre a página web e os robôs RPA, permitindo aos usuários acionarem as automatizações desejadas com simples cliques e sem a necessidade de intervenções técnicas complexas.

2.3. Limitações

- 1) A eficácia da solução está condicionada à estabilidade e às mudanças nos sites de origem (Tabela FIPE e portal da Católica SC), podendo requerer ajustes periódicos nos scripts de automação para manter a funcionalidade.
- 2) A implementação pressupõe que o usuário tenha uma conexão ativa com a internet e as permissões necessárias para instalar e executar os robôs RPA em sua máquina.
- 3) As execuções dos RPA's pelo navegador dependem que o usuário os tenha instalado em sua máquina.

3 Especificação Técnica

3.1. Requisitos de Software

Requisitos Funcionais (RF):

RF1: O sistema deve permitir que o usuário selecione entre gerar um relatório da Tabela FIPE ou baixar um boleto da faculdade.

RF2: O sistema deve acessar o site da Tabela FIPE, realizar a pesquisa com e gerar um relatório com as informações coletadas.

RF3: Para o download de boletos, o sistema deve solicitar ao usuário suas credenciais de acesso ao portal da faculdade.

Documento de Especificação do Projeto

RPA Boettscher

RF4: O sistema deve acessar o portal da faculdade, autenticar o usuário com as credenciais fornecidas, localizar o boleto mais recente e realizar o download em formato PDF.

RF5: O sistema deve disponibilizar os relatórios gerados e os boletos baixados para o usuário, por meio da interface web.

Requisitos Não-Funcionais (RNF):

RNF1: A interface web deve ser responsiva e acessível em diferentes dispositivos e navegadores.

RNF2: O sistema deve prover mensagens de erro claras e instrutivas em caso de falhas na execução das tarefas.

RNF3: A API do projeto deve iniciar a execução do RPA selecionado assim que o usuário clicar em “Executar” na página web.

RNF4: O site deve conter uma página para login e autenticação do usuário e uma página para seleção e execução dos RPA's.

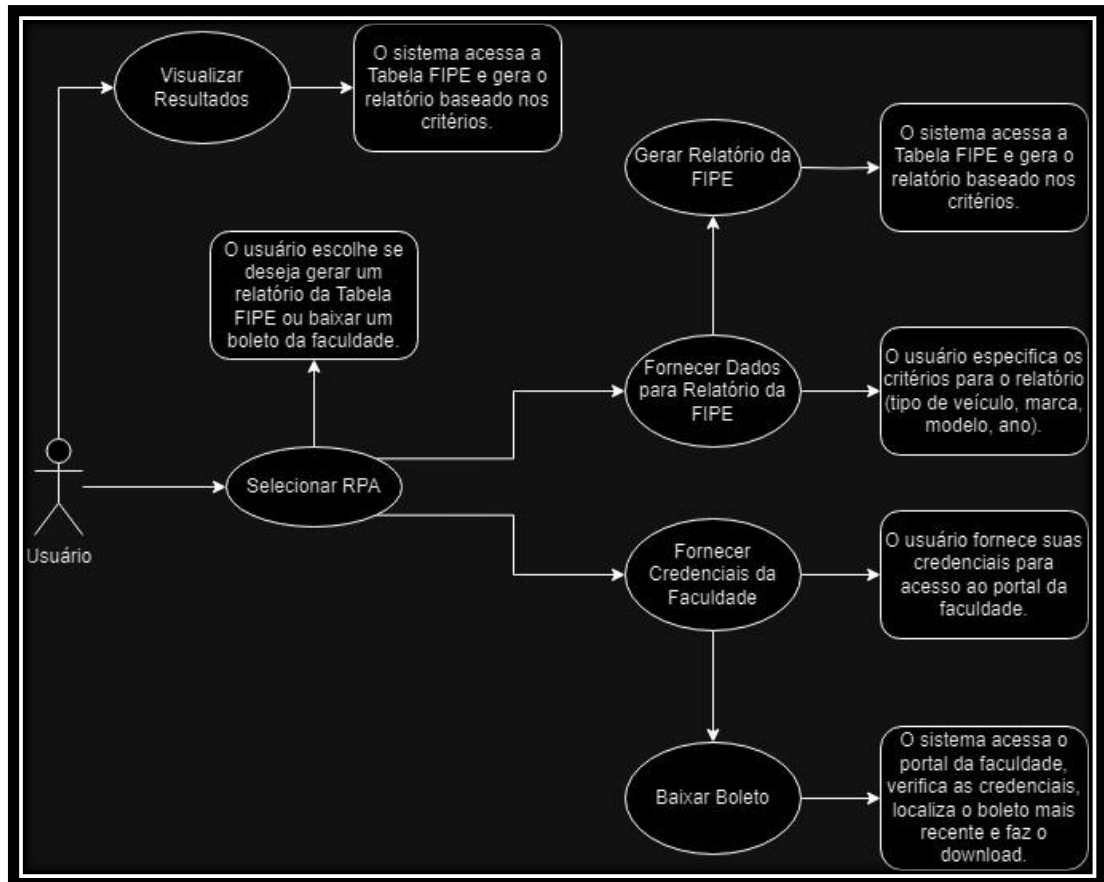
RNF5: A API recebe a solicitação do usuário via interface web e encaminha para o RPA correspondente.

RNF6: A API recebe os resultados do processamento do RPA e os disponibiliza no diretório temporário da máquina do usuário.

Documento de Especificação do Projeto

RPA Boettcher

Diagrama de Casos de Uso:



Documento de Especificação do Projeto

RPA Boettcher

3.2. Considerações de Design

Escolhas de Design:

1. Execução em AWS EC2

- a. Escolha: Utilização de AWS EC2 para executar os RPA's.
- b. Justificativa: AWS EC2 oferece escalabilidade, custo-efetividade e elimina a necessidade de manter infraestrutura própria para execução dos RPA's.
- c. Alternativa Considerada: Execução local dos RPA's na máquina do usuário.
- d. Razão para Rejeição: Maior complexidade de instalação e manutenção, além de dependência da configuração do ambiente do usuário.

2. Integração via API

- a. Escolha: Uso de uma API para conectar a interface web aos RPA's executados em AWS EC2.
- b. Justificativa: Facilita a comunicação entre a interface de usuário e os serviços EC2, promovendo uma arquitetura desacoplada e escalável.
- c. Alternativa Considerada: Comunicação direta entre a interface web e os executáveis dos RPA.
- d. Razão para Rejeição: Menor flexibilidade e maior complexidade na gestão de atualizações e manutenção do sistema.

3. Automação com AWS EC2

- a. Escolha: Uso de AWS EC2 para executar tarefas de automação.
- b. Justificativa: EC2 é altamente escalável, gerenciado e suporta execução dos processos definidos no escopo.
- c. Alternativa Considerada: Execução local.
- d. Razão para Rejeição: Necessidade de gerenciamento de infraestrutura.

Visão Inicial da Arquitetura:

1. Usuário

Documento de Especificação do Projeto

RPA Boettcher

- a. Interação com: Interface Web
- 2. Interface Web
 - a. Interação com: Usuário, API
 - b. Descrição: Plataforma onde o usuário seleciona operações e fornece dados necessários.
- 3. API
 - a. Interação com: Interface Web, AWS EC2
 - b. Descrição: Intermediário que processa solicitações da interface web e aciona os RPAs no AWS EC2.
- 4. AWS EC2
 - a. Interação com: API, Site da Tabela FIPE, Portal da Faculdade
 - b. Descrição: Executa RPAs para automação das tarefas de coleta de dados e download de arquivos.
- 5. Site da Tabela FIPE
 - a. Interação com: AWS EC2
 - b. Descrição: Fonte de dados para geração de relatórios de valores de veículos.
- 6. Portal da Faculdade
 - a. Interação com: AWS EC2
 - b. Descrição: Plataforma de onde os boletos de mensalidade são baixados.

Fluxo de Dados:

1. O Usuário interage com a Interface Web para selecionar uma operação.
2. A Interface Web envia a solicitação à API.
3. A API encaminha a solicitação ao RPA correspondente.
4. O RPA realiza a tarefa no Site da Tabela FIPE ou Portal da Faculdade.
5. Os resultados são retornados à API, que os disponibiliza na Interface Web para o Usuário.

Padrões de Arquitetura:

Documento de Especificação do Projeto

RPA Boettscher

1. MVC (Model-View-Controller):

Aplicação: Estruturação da Interface Web.

Justificativa: Facilita a separação das responsabilidades de apresentação (View), lógica de negócio (Controller) e gerenciamento de dados (Model).

2. Microserviços:

Aplicação: Estruturação da API.

Justificativa: Promove escalabilidade e facilidade de manutenção, permitindo que cada serviço seja desenvolvido, implantado e escalado independentemente.

3. RESTful API:

Aplicação: Design da API.

Justificativa: Padrão amplamente adotado para criar APIs que são simples, escaláveis e utilizam HTTP para comunicação.

4. Automação Robótica (RPA):

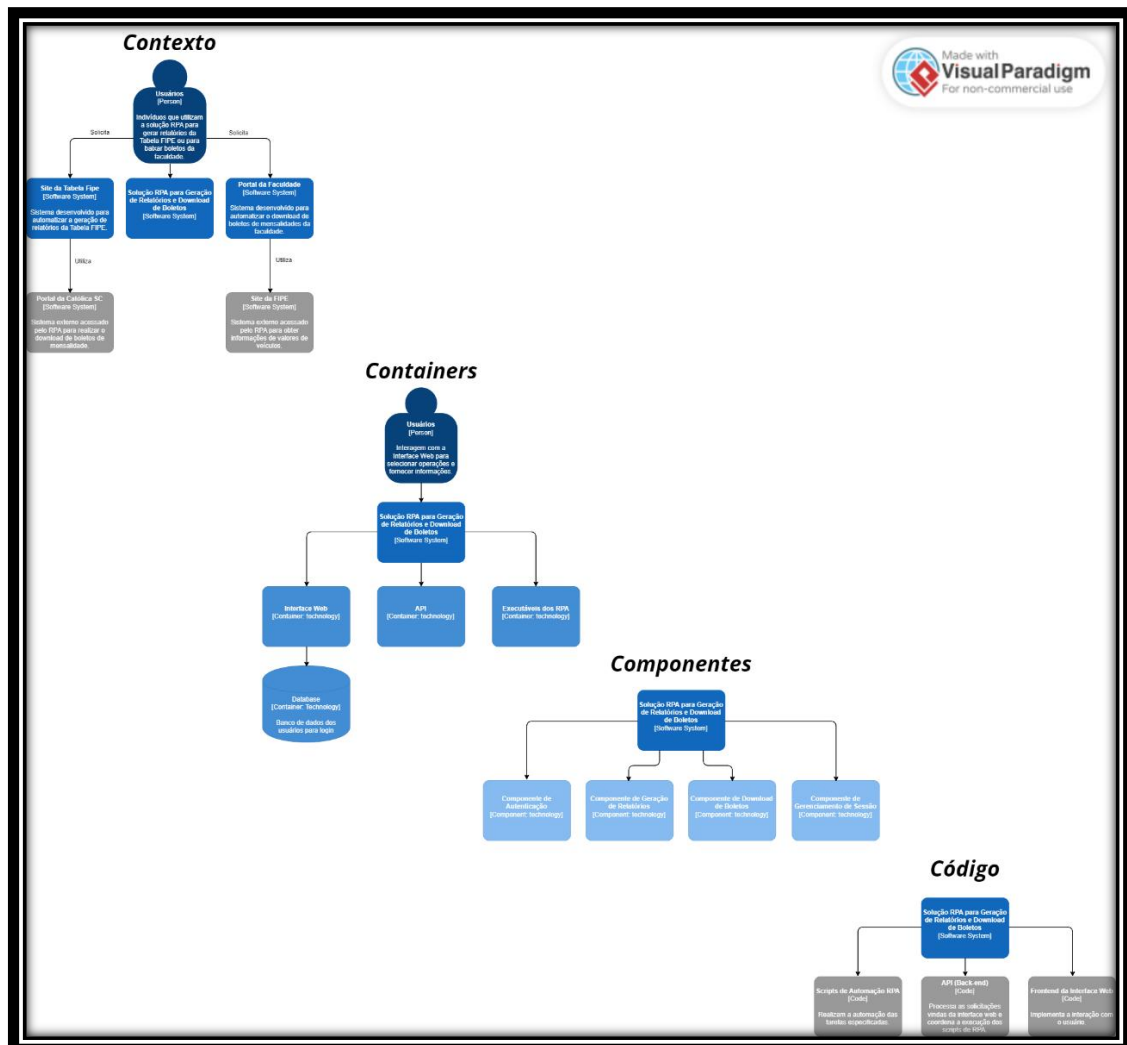
Aplicação: Design dos RPAs para automação.

Justificativa: Especialmente adequado para automatizar interações repetitivas com interfaces de usuário.

Documento de Especificação do Projeto

RPA Boettscher

Diagrama C4:



3.3. Stack Tecnológica

Linguagens de Programação

1. JavaScript

- Justificativa: Amplamente utilizado para desenvolvimento de front-end e back-end (Node.js).

2. C#

- Justificativa: Forte suporte para automação e bibliotecas RPA e linguagem que o desenvolvedor mais tem conhecimento sobre.

Frameworks e Bibliotecas

1. Front-end

Documento de Especificação do Projeto

RPA Boettcher

- a. Next.js - Justificativa: Facilita a criação de interfaces de usuário dinâmicas e responsivas.
- 2. Back-end
 - a. Node.js / Express - Justificativa: Popular para criar APIs RESTful eficientes e escaláveis.

Automação

- 1. AWS EC2
 - a. Justificativa: Permite execução dos RPAs, escalável e fácil de integrar com outros serviços AWS.
- 2. Puppeteer
 - a. Justificativa: Usado para automação de navegadores web, ideal para scraping de dados e interação com sites.

Ferramentas de Desenvolvimento e Gestão de Projeto

- 1. Controle de Versão
 - a. GitHub - Justificativa: Plataforma de hospedagem de código com integração CI/CD.
- 2. Integração Contínua/Entrega Contínua (CI/CD)
 - a. GitHub Actions - Justificativa: Automação de fluxos de trabalho de CI/CD.
- 3. Gerenciamento de Projeto
 - a. Azure DevOps - Justificativa: Ferramentas para rastreamento de tarefas e gerenciamento de projetos ágeis.
- 4. IDE/Editor de Código
 - a. Visual Studio Code - Justificativa: Popular, com suporte extensivo para várias linguagens e integração com diversas ferramentas.

Outras Tecnologias

- 1. AWS Services
 - a. AWS S3 - Justificativa: Armazenamento de arquivos gerados pelos RPA's.
 - b. AWS IAM - Justificativa: Gestão de permissões e segurança dos recursos AWS.

Documento de Especificação do Projeto

RPA Boettcher

3.4. Considerações de Segurança

1. Autenticação e Autorização

Mitigação: Uso de AWS IAM para controle de acesso. Implementação de OAuth2.0 para autenticação de usuários.

2. Proteção de Dados Sensíveis

Mitigação: Criptografia de dados sensíveis em trânsito e em repouso, utilizando HTTPS e AWS KMS.

3. Segurança de API

Mitigação: Implementação de rate limiting e validação de entrada para prevenir ataques de DDoS e injeção de código.

4. Segurança do Front-end

Mitigação: Implementação de medidas contra XSS (Cross-Site Scripting) e CSRF (Cross-Site Request Forgery).

5. Logging e Monitoramento

Mitigação: Implementação de Logs para monitoramento no servidor EC2 e sites de terceiros dos serviços utilizados (Auth0, AWS e Vercel).

4 Próximos Passos

4.1. Portifólio I:

- Desenvolvimento da Interface Web
- Implementação da API
- Desenvolvimento dos Scripts RPA
- Integração Inicial com AWS EC2
- Teste e Validação das Funcionalidades Básicas

4.2. Portifólio II:

- Otimização e Refinamento dos RPA's
- Escalonamento e Testes de Carga
- Implementação Completa de Segurança
- Documentação Completa
- Deploy Final e Treinamento de Usuários

Documento de Especificação do Projeto

RPA Boettscher

5 Referências

5.1. Frameworks e Bibliotecas

- React.js: <https://reactjs.org/>
- Next.js: <https://nextjs.org/>
- Node.js: <https://nodejs.org/>
- Express: <https://expressjs.com/>
- Puppeteer: <https://pptr.dev/>

5.2. Ferramentas

- GitHub: <https://github.com/>
- Visual Studio Code: <https://code.visualstudio.com/>

5.3. Serviços AWS

- AWS S3: <https://aws.amazon.com/s3/>
- AWS IAM: <https://aws.amazon.com/iam/>
- AWS EC2: <https://aws.amazon.com/>

6 Considerações dos Professores

6.1. Professor(a) 1:

...

6.2. Professor(a) 2:

...

6.3. Professor(a) 3:

...