

ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

THALES YUMIYA

ARQUITETURA DE SOFTWARE PARA APLICATIVOS  
SUSTENTÁVEIS: UMA SOLUÇÃO EM NUVEM PARA A  
GESTÃO DE RESÍDUOS

SÃO PAULO

2025

ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

THALES YUMIYA

ARQUITETURA DE SOFTWARE PARA APLICATIVOS  
SUSTENTÁVEIS: UMA SOLUÇÃO EM NUVEM PARA A  
GESTÃO DE RESÍDUOS

Dissertação apresentada à Escola Politécnica  
da Universidade de São Paulo para obtenção do  
Título de Especialização em Engenharia de  
Software.

SÃO PAULO

2025

ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

THALES YUMIYA

ARQUITETURA DE SOFTWARE PARA APLICATIVOS  
SUSTENTÁVEIS: UMA SOLUÇÃO EM NUVEM PARA A  
GESTÃO DE RESÍDUOS

Dissertação apresentada à Escola Politécnica  
da Universidade de São Paulo para obtenção do  
Título de Especialização em Engenharia de  
Software.

Área de Concentração:  
PECE

Orientador:  
Prof. Dr. Alexandre dos Santos Mignon

SÃO PAULO

2025

# SUMÁRIO

<b>1. MOTIVAÇÕES E CONTEXTUALIZAÇÃO.....</b>	<b>8</b>
1.1. INTRODUÇÃO AO PROBLEMA.....	8
1.2. SITUAÇÃO NO BRASIL E OS DESAFIOS LOCAIS.....	8
1.3. OPORTUNIDADE DE SOLUÇÕES TECNOLÓGICAS.....	9
1.4. OBJETIVO DO TRABALHO.....	10
1.5. MOTIVAÇÕES PARA O DESENVOLVIMENTO DO APLICATIVO.....	10
<b>2. CONCEITO DE COMPUTAÇÃO EM NUVEM E O USO DA AWS CLOUD.....</b>	<b>11</b>
2.1. CONCEITO DE COMPUTAÇÃO EM NUVEM.....	12
2.2. AWS CLOUD: UMA VISÃO GERAL.....	13
<b>3. CENÁRIO ATUAL DA EMPRESA DE COLETA DE RESÍDUOS.....</b>	<b>15</b>
3.1. INTRODUÇÃO.....	15
3.2. COMO FUNCIONA O SERVIÇO ATUAL.....	15
3.3. RAZÕES PARA A FALTA DE RECURSOS TECNOLÓGICOS.....	16
3.4. NECESSIDADES DA EMPRESA PARA A IMPLEMENTAÇÃO DO APLICATIVO.....	16
3.5. NECESSIDADES DOS CLIENTES DA EMPRESA.....	17
<b>4. ARQUITETURA E IMPLEMENTAÇÃO DO APLICATIVO.....</b>	<b>18</b>
4.1. VANTAGENS DE USAR COMPUTAÇÃO EM NUVEM PARA O APLICATIVO DE RECICLAGEM.....	18
4.1.1. ESCALABILIDADE E FLEXIBILIDADE.....	18
4.1.2. SEGURANÇA E CONFORMIDADE.....	18
4.1.3. BAIXO CUSTO E REDUÇÃO DE INVESTIMENTO INICIAL.....	19
4.1.4. ALTA DISPONIBILIDADE E CONFIABILIDADE.....	19
4.2. ARQUITETURA DE REFERÊNCIA CLOUD-AGNOSTIC.....	20
4.2.1. DEFINIÇÃO DE ARQUITETURA DE REFERÊNCIA.....	20
4.2.2. A PROPOSTA COMO ARQUITETURA DE REFERÊNCIA.....	21
4.2.3. ADAPTABILIDADE E EXTENSIBILIDADE DA ARQUITETURA.....	23
4.2.4. DIAGRAMA DA ARQUITETURA DE REFERÊNCIA.....	24
4.2.5. FLUXOS PRINCIPAIS NA ARQUITETURA DE REFERÊNCIA.....	26
4.2.6. CONSIDERAÇÕES SOBRE PORTABILIDADE.....	27
4.3. IMPLEMENTAÇÃO NA AWS CLOUD.....	28
4.3.1. MAPEAMENTO DE COMPONENTES PARA SERVIÇOS AWS.....	28
4.3.2. DIAGRAMA DA IMPLEMENTAÇÃO AWS.....	29
4.3.3. COMPONENTES ESPECÍFICOS DA AWS.....	33
4.3.3.1. FUNÇÕES LAMBDA.....	33
4.3.3.2. ELASTIC CONTAINER SERVICE (ECS).....	34
4.3.3.3. AMAZON EC2 E AWS FARGATE.....	35
4.4. CONTRIBUIÇÃO DO APLICATIVO.....	37
4.5. FUNCIONALIDADES DO APLICATIVO.....	38
4.5.1. AGENDAMENTO DE COLETAS.....	38

4.5.2. HISTÓRICO DE COLETAS.....	38
4.5.3. CÁLCULO DO IMPACTO AMBIENTAL.....	38
4.5.4. RASTREAMENTO DE VEÍCULOS DE COLETA.....	39
4.5.5. PAGAMENTO E ASSINATURAS.....	39
4.5.6. NOTIFICAÇÕES E LEMBRETES.....	39
4.5.7. SUPORTE AO CLIENTE.....	39
4.6. FLUXO DE DADOS E INTERAÇÕES.....	40
4.7. NOTIFICAÇÃO DE CHEGADA DO VEÍCULO DE COLETA.....	42
4.8. PIPELINE DE DESENVOLVIMENTO E DEPLOY.....	44
4.9. MOCKUP DO APLICATIVO MOBILE.....	46
4.9.1. TELA INICIAL.....	46
4.9.2. TELA "SUA COLETA".....	48
4.9.3. TELA DO CÁLCULO DO IMPACTO.....	49
4.9.4. TELA DO HISTÓRICO.....	53
<b>5. ANÁLISE DE CUSTOS DA IMPLEMENTAÇÃO NA AWS CLOUD.....</b>	<b>54</b>
5.1. INTRODUÇÃO.....	54
5.2. SERVIÇOS AWS NECESSÁRIOS.....	54
5.3. ESTIMATIVA DE CUSTOS MENSAIS.....	55
5.4. BENEFÍCIOS ECONÔMICOS DA AWS.....	56
<b>6. CONCLUSÃO.....</b>	<b>58</b>
<b>7. REFERÊNCIA BIBLIOGRÁFICA.....</b>	<b>60</b>

## **LISTA DE FIGURAS**

<b>Figura 1 - Diagrama da arquitetura de referência.....</b>	<b>7</b>
<b>Figura 2 - Modelo de arquitetura AWS para o aplicativo.....</b>	<b>29</b>
<b>Figura 3 - Diagrama de Arquitetura para rastreamento.....</b>	<b>43</b>
<b>Figura 4 - Modelo de Arquitetura para o Pipeline.....</b>	<b>44</b>
<b>Figura 5 - Tela inicial do aplicativo.....</b>	<b>46</b>
<b>Figura 6 - Tela do menu lateral.....</b>	<b>47</b>
<b>Figura 7 - Tela de rastreamento.....</b>	<b>48</b>
<b>Figura 8 - Tela do cálculo do impacto ambiental.....</b>	<b>49</b>
<b>Figura 9 - Tela do histórico de coletas.....</b>	<b>50</b>
<b>Figura 10 - Tela de resultados do impacto ambiental.....</b>	<b>53</b>

## **LISTA DE TABELAS**

<b>Tabela 1 - Deficiências e limitações da empresa de coleta.....</b>	<b>15</b>
<b>Tabela 2 - Razões para falta de recursos tecnológicos na empresa de coleta.....</b>	<b>16</b>
<b>Tabela 3 - Necessidades da empresa para a implementação do aplicativo.....</b>	<b>17</b>
<b>Tabela 4 - Necessidades dos clientes da empresa.....</b>	<b>17</b>
<b>Tabela 5 - Componentes para serviços AWS.....</b>	<b>31</b>
<b>Tabela 6 - Fatores de Equivalência para Materiais Recicláveis.....</b>	<b>51</b>
<b>Tabela 7 - Custo estimado mensal dos serviços AWS.....</b>	<b>56</b>

## **1. MOTIVAÇÕES E CONTEXTUALIZAÇÃO**

### **1.1. INTRODUÇÃO AO PROBLEMA**

O problema da gestão de resíduos sólidos é um dos principais desafios ambientais e sociais da atualidade. O crescimento populacional e o aumento do consumo têm gerado volumes expressivos de lixo, cujo descarte inadequado impacta diretamente o meio ambiente. A Organização das Nações Unidas (ONU) estima que mais de 2 bilhões de toneladas de resíduos sólidos urbanos são gerados no mundo a cada ano, e que aproximadamente 33% desse total não recebem tratamento adequado, sendo descartados em áreas abertas, rios e aterros de baixa infraestrutura (United Nations Environment Programme, 2022). Esse descarte inadequado gera consequências como poluição dos solos, contaminação das águas e emissões de gases de efeito estufa, intensificando os problemas relacionados às mudanças climáticas.

A situação do descarte de resíduos não apenas afeta o meio ambiente, mas também representa uma questão de saúde pública. Em áreas urbanas, onde a concentração de pessoas e a produção de resíduos são maiores, o manejo incorreto do lixo pode acarretar riscos de contaminação, proliferação de vetores de doenças e degradação da qualidade de vida (Silva, 2021). Por essas razões, a gestão eficiente dos resíduos, aliada à reciclagem e à compostagem, surge como uma necessidade urgente, com o potencial de mitigar os impactos ambientais e promover práticas sustentáveis na sociedade.

### **1.2. SITUAÇÃO NO BRASIL E OS DESAFIOS LOCAIS**

O Brasil, sendo um dos países mais populosos do mundo, enfrenta um grande desafio na gestão de resíduos. Segundo o Sistema Nacional de Informações sobre Saneamento (SNIS, 2023), o Brasil gerou mais de 82 milhões de toneladas de resíduos sólidos em 2022, o que representa cerca de 1,4 kg de lixo por pessoa, diariamente.

Apesar do volume expressivo, menos de 4% de todo o resíduo coletado é efetivamente reciclado, uma taxa baixa comparada à de países desenvolvidos

(Abrelpe, 2022). Esse dado expõe a carência de infraestrutura e políticas públicas que incentivem a coleta seletiva e o processamento de materiais recicláveis.

A infraestrutura limitada e a falta de incentivo ao setor de reciclagem são aspectos que limitam o avanço de práticas sustentáveis no Brasil. Muitas cidades brasileiras ainda não possuem coleta seletiva, e a informalidade nas cooperativas de reciclagem, que representam grande parte do setor, dificulta a reciclagem efetiva de materiais como papel, vidro, metais e plásticos (Oliveira & Freitas, 2022). Além disso, o tratamento adequado do lixo orgânico ainda é incipiente em muitas regiões, com uma porcentagem significativa sendo destinada a aterros sanitários sem qualquer tipo de aproveitamento. A compostagem, prática que converte resíduos orgânicos em adubo, surge como uma alternativa viável para reduzir o volume de resíduos orgânicos nos aterros e aproveitar o material de maneira ecologicamente responsável (Dias et al., 2023).

O incentivo à compostagem ainda é baixo no Brasil, com poucos municípios oferecendo essa alternativa aos seus habitantes. Assim, surge uma oportunidade para um serviço inovador e eficiente que facilite o descarte adequado e o aproveitamento de resíduos, contribuindo para uma gestão sustentável e alinhada às práticas globais.

### **1.3. OPORTUNIDADE DE SOLUÇÕES TECNOLÓGICAS**

Com o avanço da tecnologia, especialmente em soluções digitais e de computação em nuvem, o setor de sustentabilidade tem encontrado novas formas de conectar cidadãos a práticas responsáveis. O crescimento de plataformas e aplicativos móveis que promovem serviços relacionados à sustentabilidade e consumo consciente reflete uma tendência global (Kumar & Craig, 2023). A tecnologia permite que as empresas otimizem processos, ampliem o alcance de suas soluções e ofereçam uma comunicação direta e eficiente com os usuários.

O uso de aplicativos para coleta e destinação de resíduos facilita a comunicação e a logística, permitindo que cidadãos agendem, paguem e monitorem serviços de forma prática e acessível. Por meio da integração de dados, um aplicativo pode automatizar operações, gerenciar coletas de maneira eficiente e fornecer dados em

tempo real aos usuários, promovendo uma experiência transparente e alinhada às expectativas de um consumidor consciente (Chen et al., 2023).

#### **1.4. OBJETIVO DO TRABALHO**

Este trabalho tem como objetivo propor e detalhar uma arquitetura de software robusta que sirva como base teórica para o desenvolvimento de um aplicativo destinado à gestão de resíduos sólidos. A proposta surge como resposta aos desafios enfrentados pelo setor, como a baixa taxa de reciclagem e a falta de ferramentas tecnológicas que conectem consumidores e empresas de coleta de maneira eficiente e sustentável.

O aplicativo proposto tem como principal objetivo servir de ponte entre usuários e empresas de coleta sustentável. Ele oferece uma solução prática para o descarte de resíduos orgânicos e recicláveis, integrando funcionalidades que tornam o processo de agendamento e acompanhamento da coleta simples e acessível. Com o aplicativo, o usuário pode contratar um serviço de coleta semanal, onde o lixo orgânico é destinado à compostagem e o reciclável, às cooperativas de reciclagem.

#### **1.5. MOTIVAÇÕES PARA O DESENVOLVIMENTO DO APLICATIVO**

Diante dos desafios e da falta de opções de coleta sustentável no Brasil, o desenvolvimento deste aplicativo surge como uma alternativa promissora. A falta de infraestrutura para coleta seletiva e compostagem, especialmente em áreas urbanas, evidencia a necessidade de um serviço acessível e que incentive a sustentabilidade (Oliveira et al., 2022).

Além disso, o aplicativo não é apenas uma ferramenta de serviço, mas também de conscientização ambiental. Ao permitir que o usuário faça parte de um ciclo sustentável de manejo de resíduos, o aplicativo incentiva a conscientização e o engajamento dos cidadãos na busca por soluções para os problemas ambientais. Através de uma plataforma intuitiva e prática, a ideia é que os usuários se sintam

motivados a participar ativamente da reciclagem e da compostagem, incorporando esses hábitos em sua rotina.

A integração entre tecnologia e sustentabilidade é um incentivo ao desenvolvimento de soluções inovadoras que promovam o cuidado com o meio ambiente. Esse aplicativo tem o potencial de servir como um modelo replicável em outras regiões e até mesmo em outros países, promovendo uma visão de sustentabilidade que utiliza a tecnologia como aliada na busca por um futuro mais verde (Rodrigues & Souza, 2023).

## 2. CONCEITO DE COMPUTAÇÃO EM NUVEM E O USO DA AWS CLOUD

### 2.1. CONCEITO DE COMPUTAÇÃO EM NUVEM

A computação em nuvem (Cloud Computing) é uma tecnologia que permite o armazenamento, processamento e gerenciamento de dados e aplicações em servidores remotos, acessíveis via internet. Em vez de hospedar dados e programas em servidores locais, a nuvem permite que esses recursos sejam acessados de qualquer lugar, oferecendo flexibilidade, escalabilidade e redução de custos. Segundo Mell e Grance (2011), a computação em nuvem apresenta-se em três principais modelos de serviço:

1. **Software como Serviço (SaaS):** Esse modelo permite que o usuário acesse aplicações diretamente pela internet, sem a necessidade de instalação local, como serviços de e-mail, armazenamento de arquivos e plataformas de produtividade.
2. **Plataforma como Serviço (PaaS):** No modelo PaaS, a nuvem oferece um ambiente completo para desenvolvimento, teste e implementação de aplicações, facilitando a criação de novos softwares sem a complexidade da infraestrutura subjacente.
3. **Infraestrutura como Serviço (IaaS):** Aqui, a nuvem oferece componentes fundamentais de infraestrutura, como servidores, redes e armazenamento, permitindo que as empresas utilizem esses recursos sem precisar mantê-los fisicamente.
4. **Funções como Serviços (FaaS):** é um modelo de computação em nuvem que permite aos desenvolvedores executar fragmentos específicos de código, conhecidos como "funções", em resposta a eventos, sem a necessidade de gerenciar ou provisionar servidores. Esse modelo é um dos pilares da arquitetura serverless, na qual os recursos computacionais são alocados automaticamente conforme a necessidade, cobrando apenas pelo tempo de execução e recursos utilizados.

A computação em nuvem tem se tornado uma escolha cada vez mais popular para empresas de diversos setores, devido às suas vantagens, que incluem:

- **Escalabilidade:** A capacidade de aumentar ou reduzir recursos conforme a demanda.
- **Redução de custos:** A eliminação de investimentos em infraestrutura física, como servidores, e a redução de gastos com manutenção.
- **Disponibilidade:** Garantia de que os recursos estarão acessíveis a qualquer momento e de qualquer lugar.
- **Flexibilidade e inovação:** Acesso a ferramentas avançadas que facilitam o desenvolvimento de soluções inovadoras.

Essas características fazem da computação em nuvem uma tecnologia essencial para empresas que buscam otimizar a gestão de seus recursos e desenvolver soluções ágeis e eficientes (Buyya, Yeo, & Venugopal, 2008).

## 2.2. AWS CLOUD: UMA VISÃO GERAL

A Amazon Web Services (AWS) é uma das plataformas de computação em nuvem mais completas e amplamente utilizadas no mundo, oferecendo uma ampla gama de serviços que abrangem armazenamento, processamento, rede, inteligência artificial, análise de dados e muitos outros. Lançada em 2006, a AWS foi uma das pioneiras no setor e, desde então, tem evoluído continuamente para atender às necessidades de empresas de todos os tamanhos, em diferentes setores da economia.

Os principais serviços oferecidos pela AWS incluem:

1. **Amazon EC2 (Elastic Compute Cloud):** Um serviço de infraestrutura que oferece capacidade computacional sob demanda, permitindo a criação e gerenciamento de servidores virtuais.
2. **Amazon S3 (Simple Storage Service):** Um serviço de armazenamento escalável que permite o armazenamento de grandes volumes de dados de maneira segura e acessível.

3. **AWS Lambda**: Um serviço de computação sem servidor (serverless) que executa código em resposta a eventos e gerencia automaticamente os recursos necessários, ideal para aplicações que demandam alta disponibilidade e escalabilidade.
4. **Amazon RDS (Relational Database Service)**: Um serviço de banco de dados gerenciado que facilita a configuração, operação e escalabilidade de bancos de dados relacionais na nuvem.

Além desses serviços básicos, a AWS oferece uma variedade de ferramentas para análise de dados, machine learning, IoT (Internet das Coisas), inteligência artificial, segurança e gerenciamento de identidades, entre outros. Esses serviços fazem com que a AWS seja uma escolha popular para empresas que buscam uma solução de nuvem completa e confiável (Amazon Web Services, 2023).

### **3. CENÁRIO ATUAL DA EMPRESA DE COLETA DE RESÍDUOS**

#### **3.1. INTRODUÇÃO**

Para entender as necessidades e motivações para a criação do aplicativo de reciclagem, é essencial examinar o cenário atual da empresa de coleta de resíduos e identificar os pontos de melhoria. Atualmente, a empresa já oferece o serviço de coleta de resíduos orgânicos e recicláveis, mas a falta de um sistema digital e centralizado limita a experiência dos usuários e a eficiência das operações. Este capítulo explora o cenário vigente, analisando as limitações operacionais, a experiência do cliente, e as razões pelas quais a empresa precisa de um aplicativo para modernizar seus processos.

#### **3.2. COMO FUNCIONA O SERVIÇO ATUAL**

A empresa de coleta de resíduos realiza coletas regulares, direcionando o lixo orgânico para compostagem e o reciclável para cooperativas de reciclagem. No entanto, o gerenciamento de dados e a interação com os clientes são feitos de forma manual e descentralizada.

A seguir, temos as principais deficiências e limitações da empresa coletora:

<b>Aspecto</b>	<b>Descrição</b>
Pagamentos	O pagamento pelos serviços é realizado manualmente via WhatsApp, dificultando o controle financeiro e o planejamento de receita.
Acompanhamento de dados	Não há um sistema para que os usuários consultem informações sobre volume de resíduos reciclados ou impacto ambiental.
Agendamento de coletas	Os clientes não têm flexibilidade para agendar coletas para dias específicos, seguindo um cronograma fixo.
Informação em tempo real	Não há um sistema para informar o horário aproximado de chegada do caminhão de coleta.

Tabela 1 - Deficiências e limitações da empresa de coleta

### **3.3. RAZÕES PARA A FALTA DE RECURSOS TECNOLÓGICOS**

Embora essas limitações comprometam a experiência dos clientes, o modelo atual ainda é comum em pequenas empresas do setor de resíduos, especialmente em mercados onde a digitalização ainda está em processo inicial. Algumas das razões para essa realidade estão listadas na tabela 2, a seguir:

<b>Fator</b>	<b>Descrição</b>
Custos de Implementação	A criação de um aplicativo e a adoção de tecnologias representam um custo significativo para a empresa.
Falta de Especialização Técnica	A empresa foca nas operações de coleta e não possui equipe especializada em TI para projetar soluções digitais.
Baixa Demanda Percebida	Os clientes ainda não pressionam pela digitalização, e a empresa opera de maneira tradicional sem grandes dificuldades.

Tabela 2 - Razões para falta de recursos tecnológicos na empresa de coleta

### **3.4. NECESSIDADES DA EMPRESA PARA A IMPLEMENTAÇÃO DO APPLICATIVO**

A implementação de um aplicativo pode proporcionar à empresa melhorias significativas em várias áreas de sua operação. As principais necessidades da empresa, que poderiam ser atendidas com o uso de um aplicativo são descritas na tabela 3, a seguir:

Necessidade	Descrição
Centralização de Pagamentos	Melhor controle das mensalidades e redução da inadimplência com pagamentos automatizados.
Disponibilização de Informações	Acesso a dados sobre o volume de resíduos reciclados, incentivando a participação dos clientes.
Flexibilidade no Agendamento	Permitir que os usuários escolham dias e horários para as coletas, conforme suas necessidades.
Monitoramento em Tempo Real	Implementar um sistema para rastreamento dos caminhões de coleta, informando a chegada aproximada.

Tabela 3 - Necessidades da empresa para a implementação do aplicativo

### 3.5. NECESSIDADES DOS CLIENTES DA EMPRESA

Os clientes também enfrentam dificuldades devido à falta de um sistema digital eficiente. Na tabela 4, temos as principais necessidades dos clientes que podem ser atendidas com o aplicativo:

Necessidades	Descrição
Agendamento Personalizado	Escolher datas e horários conforme a disponibilidade do cliente.
Histórico de Coletas	Acompanhar dados sobre as coletas realizadas e o impacto ambiental.
Notificações e Lembretes	Receber alertas sobre a chegada do caminhão de coleta e vencimento do pagamento.
Pagamento Online	Facilidade para pagamento de assinaturas e serviços, evitando cobranças manuais.
Rastreamento do Veículo	Visualizar a localização do caminhão de coleta e estimar a chegada.

Tabela 4 - Necessidades dos clientes da empresa

## **4. ARQUITETURA E IMPLEMENTAÇÃO DO APLICATIVO**

### **4.1. VANTAGENS DE USAR COMPUTAÇÃO EM NUVEM PARA O APLICATIVO DE RECICLAGEM**

O desenvolvimento de um aplicativo de reciclagem baseado em nuvem exige uma infraestrutura robusta e escalável que possa atender à demanda de usuários e garantir a segurança e disponibilidade dos dados. A computação em nuvem oferece uma série de vantagens para esse tipo de projeto, as quais serão discutidas a seguir.

#### **4.1.1. ESCALABILIDADE E FLEXIBILIDADE**

Uma das principais vantagens da computação em nuvem é a sua escalabilidade. O aplicativo de reciclagem poderá ter variações de uso, especialmente em datas ou períodos específicos do ano, como campanhas de reciclagem ou em datas comemorativas. Com a nuvem, é possível aumentar ou reduzir a capacidade de computação conforme a necessidade, evitando a subutilização de recursos ou a sobrecarga dos sistemas. Esse ajuste de capacidade é feito automaticamente e sem interrupções, garantindo que o aplicativo mantenha um desempenho consistente.

Além disso, a nuvem oferece uma ampla gama de ferramentas e serviços que podem ser facilmente integrados ao aplicativo, permitindo a adição de novas funcionalidades ou o aprimoramento das existentes. Essa flexibilidade é essencial para acompanhar as demandas do mercado e as preferências dos usuários. (Buyya et al., 2008).

#### **4.1.2. SEGURANÇA E CONFORMIDADE**

A segurança é um aspecto crítico para qualquer solução tecnológica, especialmente quando envolve dados dos usuários. Os provedores de nuvem fornecem uma infraestrutura segura, com criptografia de dados em trânsito e em repouso, controle de acesso rigoroso, gerenciamento de identidades e monitoramento em tempo real. Além disso, os principais provedores mantêm conformidade com regulamentações

e padrões internacionais de segurança, como a ISO 27001 e o GDPR, que são essenciais para a proteção dos dados dos usuários e a manutenção da privacidade.

Esses recursos de segurança são fundamentais para garantir a confiança dos usuários no aplicativo de reciclagem, especialmente em um momento em que a privacidade de dados se torna cada vez mais relevante. Com a nuvem, é possível implementar políticas de segurança avançadas sem a necessidade de infraestrutura física, o que facilita a gestão de segurança do aplicativo.

#### **4.1.3. BAIXO CUSTO E REDUÇÃO DE INVESTIMENTO INICIAL**

Outra vantagem de utilizar a nuvem é a redução de custos. Com a nuvem, o aplicativo de reciclagem pode ser desenvolvido e mantido sem a necessidade de investir em servidores físicos, manutenção de hardware ou custos de atualização. Os provedores de nuvem utilizam um modelo de pagamento sob demanda, onde o cliente paga apenas pelos recursos que efetivamente utiliza, o que é altamente vantajoso para startups e projetos que estão em fase inicial.

Os principais provedores também oferecem programas de suporte para novos negócios e startups, como créditos de uso gratuito e planos de baixo custo, que permitem que projetos inovadores como o aplicativo de reciclagem possam ser desenvolvidos sem que haja um comprometimento financeiro inicial elevado (Mell & Grance, 2011).

#### **4.1.4. ALTA DISPONIBILIDADE E CONFIABILIDADE**

Os provedores de nuvem operam em diversas regiões ao redor do mundo, com uma arquitetura de data centers distribuída, que permite que os aplicativos hospedados na plataforma tenham alta disponibilidade. Essa distribuição geográfica oferece redundância, minimizando as chances de interrupções no serviço e garantindo que o aplicativo esteja sempre disponível para os usuários.

Para o aplicativo de reciclagem, a alta disponibilidade é essencial para que os usuários possam acessar informações em tempo real sobre o status das coletas e

realizar pagamentos sem interrupções. Os provedores de nuvem também oferecem serviços de backup e recuperação, garantindo que os dados estejam sempre seguros e possam ser restaurados rapidamente em caso de falhas (Chen et al., 2023).

## **4.2. ARQUITETURA DE REFERÊNCIA CLOUD-AGNOSTIC**

A arquitetura proposta foi concebida como uma arquitetura de referência agnóstica à plataforma de nuvem, podendo ser implementada em qualquer provedor que ofereça serviços equivalentes. Esta seção apresenta o modelo conceitual que pode ser adaptado a diferentes ambientes.

### **4.2.1. DEFINIÇÃO DE ARQUITETURA DE REFERÊNCIA**

Uma arquitetura de referência representa um template ou blueprint abstrato que fornece um framework estruturado e reutilizável para o desenvolvimento de sistemas em um domínio específico. Diferentemente de uma arquitetura de solução concreta, uma arquitetura de referência não está vinculada a uma implementação específica, mas serve como um modelo conceitual que captura as melhores práticas, padrões arquiteturais, componentes e suas interações para resolver problemas comuns dentro de um contexto.

Segundo Bass, Clements e Kazman (2012), arquiteturas de referência são abstrações de arquiteturas concretas, que identificam e incorporam elementos comuns, padrões e processos recorrentes em um domínio específico. Elas servem como ponto de partida para o desenvolvimento de novas soluções, proporcionando consistência, padronização e maior agilidade na implementação de sistemas.

As principais características de uma arquitetura de referência incluem:

- **Generalização:** Abstrai detalhes específicos de implementação, focando em componentes e relações fundamentais.
- **Reutilização:** Permite a aplicação dos mesmos princípios e estruturas em múltiplos contextos e projetos.
- **Padronização:** Estabelece convenções e padrões que garantem consistência em diferentes implementações.
- **Evolução:** É refinada continuamente com base nas experiências e lições aprendidas em implementações concretas.
- **Documentação:** Inclui diretrizes, princípios, restrições e justificativas para as decisões arquiteturais.

#### **4.2.2. A PROPOSTA COMO ARQUITETURA DE REFERÊNCIA**

A arquitetura de referência para gestão de resíduos organiza-se em camadas lógicas que interagem entre si para fornecer todas as funcionalidades necessárias, independentemente da tecnologia específica utilizada para implementação. Estas camadas encapsulam grupos de componentes com responsabilidades similares, criando uma estrutura modular e extensível.

Na camada de apresentação, encontramos as interfaces através das quais os usuários interagem com o sistema. O aplicativo móvel oferece uma experiência intuitiva para os usuários finais, permitindo o agendamento de coletas, acompanhamento de histórico e visualização de métricas ambientais. Uma camada de distribuição de conteúdo garante que estes componentes de interface sejam entregues de forma eficiente e rápida aos usuários, independentemente de sua localização geográfica.

Intermediando a comunicação entre as interfaces e os serviços de backend, a camada de acesso e API atua como porteira do sistema. O gateway de API centraliza a gestão, segurança e roteamento de todas as solicitações, aplicando políticas consistentes de autenticação e autorização através do componente de gerenciamento

de identidade e acesso. Mecanismos de limitação de taxa e proteção contra ataques salvaguardam a infraestrutura contra uso indevido, garantindo a disponibilidade para usuários legítimos mesmo em condições adversas.

No coração do sistema, a camada de serviços abriga os componentes funcionais que implementam a lógica de negócio. O serviço de agendamentos gerencia todo o ciclo de vida das coletas, desde a solicitação até a conclusão. Integrado a ele, o serviço de pagamentos processa transações financeiras de forma segura e confiável. O serviço de rastreamento monitora a localização dos veículos em tempo real, permitindo otimização de rotas e notificações precisas aos usuários. O serviço de cálculo de impacto quantifica as métricas ambientais resultantes das coletas, traduzindo volume de resíduos em benefícios ecológicos tangíveis. Complementando estes, o serviço de notificações mantém os usuários informados sobre os eventos relevantes, como confirmações de agendamento ou proximidade do veículo de coleta.

Para preservar dados e estado, a camada de persistência oferece diferentes modalidades de armazenamento, cada uma otimizada para tipos específicos de informação. O armazenamento relacional mantém dados estruturados com relações complexas, como informações de usuários, agendamentos e transações. O armazenamento NoSQL lida com dados não estruturados ou semi-estruturados que exigem alta velocidade de acesso e elasticidade, como telemetria de veículos e registros de eventos. Completando o trio, o armazenamento de objetos gerencia arquivos e mídia, como documentos, imagens e conteúdo estático do aplicativo.

Para garantir a saúde operacional do sistema, a camada de monitoramento e observabilidade mantém vigilância constante sobre todos os componentes. Sistemas de coleta de métricas e logs agregam dados operacionais de toda a infraestrutura, permitindo análises em tempo real e históricas. O sistema de alertas detecta anomalias e condições predefinidas, notificando os responsáveis quando intervenções são necessárias. Dashboards operacionais traduzem dados brutos em visualizações intuitivas, facilitando a compreensão do estado do sistema e a identificação de tendências.

Finalmente, a camada de CI/CD e DevOps fornece a estrutura necessária para o desenvolvimento contínuo e evolução do sistema. O repositório de código centraliza e versiona todos os artefatos de software, enquanto o sistema de build e testes automatiza a validação de novas funcionalidades. Pipelines de implantação coordenam a promoção do código entre ambientes de desenvolvimento, teste e produção. Práticas de infraestrutura como código garantem que toda a configuração de ambiente seja versionada, testável e reproduzível, eliminando inconsistências entre ambientes e facilitando atualizações de infraestrutura.

#### 4.2.3. ADAPTABILIDADE E EXTENSIBILIDADE DA ARQUITETURA

A arquitetura proposta pode ser facilmente adaptada e estendida para atender diferentes requisitos e escalas de operação:

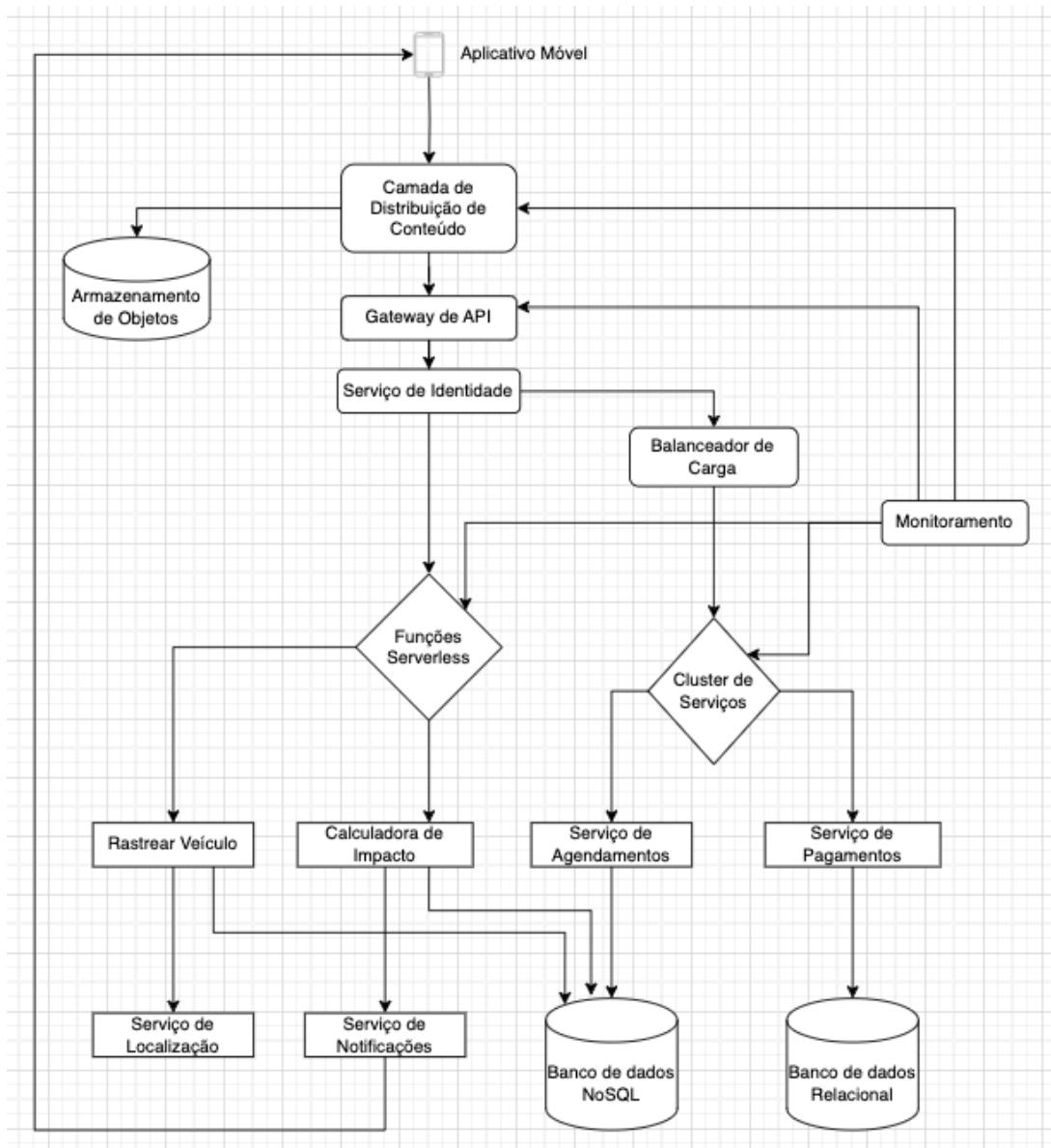
- **Dimensionamento vertical:** Para empresas de pequeno porte, é possível simplificar a arquitetura, consolidando alguns microsserviços e reduzindo a complexidade inicial.
- **Dimensionamento horizontal:** Para operações de grande escala, componentes adicionais podem ser incorporados, como sistemas de IA para otimização de rotas, análise preditiva de volume de resíduos, ou integração com sistemas municipais de gestão ambiental.
- **Extensões funcionais:** A arquitetura comporta facilmente a adição de novas funcionalidades, como marketplace para materiais recicláveis, gamificação para engajamento comunitário, ou integração com dispositivos IoT para automação da coleta.

Ao adotar esta arquitetura de referência, novas implementações podem se beneficiar das decisões arquiteturais já validadas, reduzindo riscos técnicos, acelerando o tempo de desenvolvimento e garantindo uma base sólida para evolução contínua.

#### 4.2.4. DIAGRAMA DA ARQUITETURA DE REFERÊNCIA

O diagrama de arquitetura de referência, apresentado na figura 1, apresenta um modelo conceitual para sistemas de gerenciamento de resíduos que pode ser implementado em qualquer plataforma de nuvem. Ele abstrai os componentes específicos de provedores (como AWS, Azure ou Google Cloud) e foca na estrutura lógica e nos relacionamentos funcionais entre componentes.

Figura 1 - Diagrama da arquitetura de referência



Fonte: o Autor

A arquitetura de referência acima pode ser decomposta nas seguintes camadas:

- **Camada de Frontend e Distribuição**

A camada superior da arquitetura é composta pelo Aplicativo Móvel, que fornece a interface de usuário para os clientes. Estes aplicativos interagem com a Camada de Distribuição de Conteúdo, responsável pela entrega eficiente do conteúdo estático das aplicações. Esta camada de distribuição se conecta ao Armazenamento de Objetos, um repositório que mantém arquivos estáticos como imagens, documentos e recursos da interface, garantindo acesso rápido e confiável a estes elementos.

- **Camada de Gateway e Acesso**

No centro da arquitetura, encontramos o Gateway de API, que atua como ponto central de entrada para todas as solicitações do sistema, gerenciando roteamento, autenticação e aplicação de políticas de acesso. Este componente trabalha em conjunto com o Serviço de Identidade, que administra toda a autenticação e autorização dos usuários, garantindo segurança e controle de acesso apropriado aos recursos. Para distribuir eficientemente a carga de trabalho, o Balanceador de Carga direciona as solicitações entre múltiplas instâncias dos serviços backend, proporcionando escalonabilidade horizontal e alta disponibilidade ao sistema.

- **Camada de Serviços**

O núcleo funcional da arquitetura reside na camada de serviços, começando pelo Cluster de Serviços, uma infraestrutura robusta que hospeda os serviços containerizados do sistema. Dentro deste cluster, o Serviço de Agendamentos gerencia todo o ciclo de vida das coletas, desde sua criação e confirmação até sua conclusão e registro histórico. Paralelamente, o Serviço de Pagamentos processa transações financeiras e gerencia o ciclo de vida das assinaturas dos usuários. Para operações mais específicas e orientadas a eventos, a arquitetura utiliza Funções Serverless, que incluem a Calculadora de Impacto, responsável por quantificar métricas ambientais baseadas nos dados de coleta, e o Rastreador de Veículos, que gerencia a coleta, processamento e armazenamento dos dados de localização da frota.

- **Camada de Dados**

A persistência de informações é gerenciada através de uma abordagem poliglota de bancos de dados. O Banco de Dados Relacional armazena dados altamente estruturados que possuem relações complexas, como informações de usuários, agendamentos e transações financeiras. Complementarmente, o Banco de Dados NoSQL é utilizado para armazenar dados não estruturados ou semiestruturados, como a telemetria dos veículos, logs de eventos e outras informações que se beneficiam de um modelo de dados mais flexível e de alta performance para operações de leitura e escrita intensivas.

- **Componentes de Suporte**

A arquitetura é enriquecida por componentes essenciais de suporte, como o Serviço de Localização, que fornece funcionalidades avançadas de geolocalização, geocodificação e cálculos de rota para otimização das coletas. O Serviço de Notificações atua como o canal de comunicação central, enviando aos usuários confirmações, lembretes e avisos importantes sobre suas coletas programadas. Permeando toda a arquitetura, o sistema de Monitoramento coleta métricas e logs de todos os componentes operacionais, garantindo visibilidade completa do estado do sistema, facilitando diagnósticos e permitindo otimizações proativas baseadas em dados concretos de utilização.

#### **4.2.5. FLUXOS PRINCIPAIS NA ARQUITETURA DE REFERÊNCIA**

A seguir, os fluxos principais do diagrama da figura 1 são explicados:

- **Fluxo de agendamento de coleta**

O fluxo começa quando o usuário solicita uma coleta através do aplicativo. O Gateway de API autentica a solicitação e a encaminha para o Balanceador de Carga, que a direciona ao Serviço de Agendamentos. Este verifica a disponibilidade, confirma o agendamento e armazena os dados no Banco Relacional. Por fim, o Serviço de Notificações envia uma confirmação ao usuário.

- **Fluxo de rastreamento de veículos**

Os veículos enviam periodicamente sua localização para o sistema. O Gateway de API recebe estes dados e os encaminha para o Rastreador de Veículos, que utiliza o Serviço de Localização para processá-los. Estas informações são armazenadas no Banco NoSQL para consulta rápida. Quando um veículo se aproxima do destino, o Serviço de Notificações é acionado para alertar o usuário.

- **Fluxo do cálculo de impacto ambiental**

Após a conclusão de uma coleta, a Calculadora de Impacto é acionada automaticamente. Ela recupera dados da coleta atual e do histórico do usuário no Banco NoSQL, calcula métricas ambientais como CO2 evitado e recursos economizados, e armazena os resultados. Estas informações ficam então disponíveis no aplicativo, permitindo ao usuário visualizar sua contribuição para a sustentabilidade.

#### **4.2.6. CONSIDERAÇÕES SOBRE PORTABILIDADE**

Embora a arquitetura de referência seja adaptável a diferentes provedores, algumas considerações importantes sobre portabilidade devem ser observadas (Opara-Martins, J., Sahandi, R., & Tian, F., 2016):

- **Lock-in de Provedor:** Alguns serviços gerenciados têm APIs proprietárias que podem dificultar a migração.
- **Abstração de Infraestrutura:** O uso de ferramentas como Terraform ou Pulumi pode facilitar a portabilidade da infraestrutura.
- **Contêineres como Unidade de Portabilidade:** A conteinerização dos serviços aumenta significativamente a portabilidade entre nuvens.
- **Estratégia Multi-cloud:** Para aplicações críticas, considerar uma arquitetura multi-cloud desde o início pode reduzir dependências específicas.
- **Camadas de Abstração:** Implementar camadas de abstração nas integrações com serviços específicos de cada nuvem facilita futuras migrações.

### 4.3. IMPLEMENTAÇÃO NA AWS CLOUD

A seção a seguir detalha como a arquitetura de referência pode ser implementada utilizando serviços específicos da Amazon Web Services (AWS), o provedor de cloud mais utilizado atualmente no mundo.

#### 4.3.1. MAPEAMENTO DE COMPONENTES PARA SERVIÇOS AWS

A tabela 5 descreve os componentes da arquitetura deste trabalho e quais serviços AWS equivalentes são utilizados.

Componente da Arquitetura	Serviço AWS	Descrição
Aplicativo Móvel (Frontend)	Amazon S3 + CloudFront	Hospedagem de arquivos estáticos e distribuição de conteúdo
Gateway de API	Amazon API Gateway	Gerenciamento, segurança e roteamento de APIs
Gerenciamento de Identidade	Amazon Cognito	Autenticação e autorização de usuários
Funções Orientadas a Eventos	AWS Lambda	Processamento serverless para funções específicas
Serviço de Agendamentos	Amazon ECS (EC2)	Serviço containerizado para gerenciamento de agendamentos
Serviço de Pagamentos	AWS Fargate	Serviço containerizado serverless para processamento de pagamentos
Armazenamento Relacional	Amazon RDS	Banco de dados relacional para dados estruturados
Armazenamento NoSQL	Amazon DynamoDB	Banco de dados NoSQL para dados de alta velocidade
Rastreamento de Localização	Amazon Location Service	Serviços de geolocalização e rastreamento
Notificações	Amazon SNS	Sistema de notificações para usuários

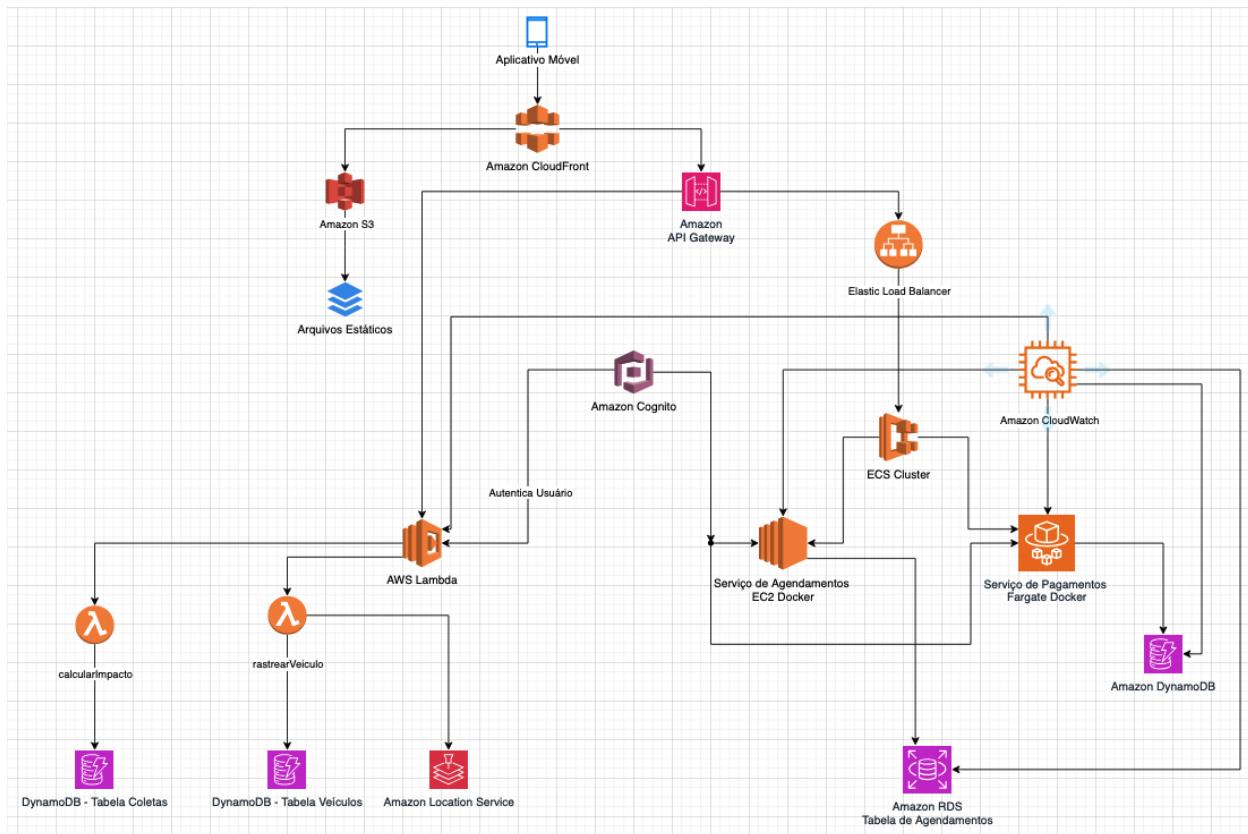
Monitoramento	Amazon CloudWatch	Coleta de métricas, logs e alertas
Pipeline de CI/CD	AWS CodeCommit, CodeBuild, CodeDeploy	Integração e entrega contínuas

Tabela 5 - Componentes para serviços AWS

#### 4.3.2. DIAGRAMA DA IMPLEMENTAÇÃO AWS

Na figura 2, temos a mesma arquitetura utilizada no diagrama de arquitetura de referência, porém dessa vez utilizando os componentes AWS equivalentes.

Figura 2 - Modelo de arquitetura AWS para o aplicativo



Fonte: o autor

Na figura 2, temos os principais componentes da arquitetura:

#### **Aplicativo Móvel:**

O Aplicativo Móvel atua como a interface principal para os usuários, sendo desenvolvido em React Native para compatibilidade com as plataformas iOS e Android. Ele se comunica com o backend utilizando o Amazon CloudFront e o Amazon API Gateway, garantindo rapidez na entrega de conteúdos e segurança na troca de dados.

#### **Amazon CloudFront:**

O Amazon CloudFront é um serviço de rede de entrega de conteúdo (CDN) que acelera a distribuição dos arquivos estáticos do frontend armazenados no Amazon S3. Essa abordagem reduz a latência e melhora a performance do aplicativo, garantindo uma experiência mais fluida para o usuário.

#### **Amazon S3:**

O Amazon S3 serve como repositório de armazenamento escalável e durável para os arquivos estáticos do frontend, como HTML, CSS, JavaScript e imagens. A partir desse serviço, o Amazon CloudFront distribui os conteúdos para os usuários finais de maneira otimizada.

#### **Amazon API Gateway:**

O Amazon API Gateway atua como o principal ponto de entrada para as APIs RESTful do backend, recebendo e encaminhando as solicitações do aplicativo móvel para os serviços adequados. Além disso, ele gerencia processos como autenticação, autorização, controle de taxa de requisições (throttling) e cache.

#### **AWS Lambda:**

No backend, a arquitetura utiliza o AWS Lambda para executar código de forma serverless em resposta a eventos específicos, sem a necessidade de gerenciar servidores. Esse serviço é utilizado para processar solicitações, como a criação de agendamentos, o cálculo do impacto ambiental dos usuários e o rastreamento de

veículos, interagindo com outros serviços da AWS, como DynamoDB e Amazon Location Service.

### **Elastic Load Balancer:**

Para garantir escalabilidade e alta disponibilidade, a aplicação também conta com um Elastic Load Balancer, que distribui as solicitações recebidas do API Gateway para instâncias do Amazon ECS. Isso assegura que os serviços backend permaneçam disponíveis e com bom desempenho, mesmo em momentos de alta demanda.

### **Amazon ECS Cluster:**

O Amazon ECS Cluster gerencia a execução dos contêineres do backend, permitindo a orquestração eficiente dos serviços. Os contêineres Docker são responsáveis pelo funcionamento de microsserviços essenciais, como o serviço de agendamentos e o serviço de pagamentos. O ECS também cuida da escalabilidade, garantindo que novos contêineres sejam iniciados conforme a necessidade.

### **Serviço de Agendamentos (EC2 - Docker):**

O Serviço de Agendamentos é um microsserviço específico que gerencia os agendamentos de coletas. Ele é implantado como um contêiner Docker no Amazon ECS e interage diretamente com o banco de dados Amazon RDS para armazenar e recuperar informações sobre os agendamentos.

### **Serviço de Pagamentos (Fargate - Docker):**

O Serviço de Pagamentos, por sua vez, lida com o processamento de transações financeiras dentro do aplicativo. Implantado no AWS Fargate, um serviço serverless para execução de contêineres, ele se comunica com o Amazon DynamoDB para registrar e recuperar dados de pagamento de forma eficiente e segura.

### **Amazon RDS:**

O Amazon RDS desempenha um papel fundamental na arquitetura ao fornecer um banco de dados relacional gerenciado, utilizado para armazenar informações sobre

os agendamentos e demais operações críticas do aplicativo. Sua escalabilidade e alta disponibilidade garantem um funcionamento confiável.

#### **Amazon DynamoDB:**

Além disso, o Amazon DynamoDB é utilizado como um banco de dados NoSQL altamente escalável para armazenar dados sobre coletas, veículos e pagamentos. Ele oferece alto desempenho e uma estrutura serverless que facilita a manutenção do sistema.

#### **Amazon Location Service:**

Para o rastreamento de veículos, a arquitetura inclui o Amazon Location Service, que fornece recursos avançados de geolocalização. Esse serviço permite que as funções Lambda responsáveis pelo rastreamento de veículos processem e armazenem informações de localização, garantindo a precisão do acompanhamento em tempo real.

#### **Amazon Cognito:**

O Amazon Cognito gerencia a autenticação e autorização de usuários no aplicativo, garantindo um processo seguro de login e controle de acesso. Ele se integra com outros serviços AWS para fornecer autenticação robusta e simplificada.

#### **Amazon CloudWatch:**

O Amazon CloudWatch é responsável pelo monitoramento do sistema, coletando métricas, logs e eventos de diversos componentes. Com isso, é possível detectar anomalias, definir alarmes e otimizar o desempenho do aplicativo continuamente.

#### **Amazon SNS e Amazon CloudWatch Events:**

Para o envio de notificações em tempo real sobre a chegada dos veículos de coleta, a solução utiliza o Amazon SNS e o Amazon CloudWatch Events, garantindo que os usuários sejam informados de maneira eficiente e imediata.

### 4.3.3. COMPONENTES ESPECÍFICOS DA AWS

Nesta seção, alguns dos componentes mais importantes da AWS são detalhados.

#### 4.3.3.1. FUNÇÕES LAMBDA

O uso de funções Lambda da AWS desempenha um papel fundamental na arquitetura do aplicativo. As funções Lambda são um serviço de computação sem servidor que permite executar código em resposta a eventos ou solicitações específicas, sem a necessidade de provisionar ou gerenciar infraestrutura.

No contexto do projeto, as funções Lambda foram utilizadas pelos seguintes motivos:

- **Processamento orientado a eventos:** Algumas funcionalidades do aplicativo, como o cálculo do impacto ambiental e o rastreamento de veículos em tempo real, são acionadas por eventos específicos. As funções Lambda são ideais para lidar com esse tipo de processamento orientado a eventos, pois podem ser acionadas automaticamente quando um evento ocorre, como uma atualização de localização de um veículo.
- **Escalabilidade automática:** As funções Lambda são automaticamente escaladas pela AWS com base no número de solicitações recebidas. Isso significa que o aplicativo pode lidar com picos de demanda sem a necessidade de provisionar manualmente recursos adicionais.
- **Redução da complexidade operacional:** Ao utilizar funções Lambda, a equipe de desenvolvimento pode se concentrar na lógica de negócios e no código da aplicação, sem se preocupar com o gerenciamento da infraestrutura subjacente.
- **Integração com outros serviços da AWS:** As funções Lambda se integram perfeitamente com outros serviços da AWS, como o Amazon DynamoDB para armazenamento de dados e o Amazon Location Service para recursos de geolocalização.
- **Eficiência de custos:** Com as funções Lambda, você paga apenas pelo tempo de computação real consumido durante a execução das funções. Isso significa que não há custos associados a recursos ociosos ou subutilizados.

O uso de funções Lambda no projeto oferece benefícios significativos em termos de escalabilidade, flexibilidade, eficiência operacional e integração com outros serviços da AWS. Essas funções permitem que o aplicativo processe eventos de forma eficiente, escale automaticamente para atender à demanda e reduza a complexidade operacional, ao mesmo tempo em que oferece uma arquitetura de aplicativo moderna e econômica.

#### **4.3.3.2. ELASTIC CONTAINER SERVICE (ECS)**

O uso do Amazon Elastic Container Service (ECS) para os serviços de agendamentos e pagamentos na arquitetura do aplicativo de coleta de resíduos oferece várias vantagens significativas. O ECS é um serviço de orquestração de contêineres altamente escalável e totalmente gerenciado que permite executar, gerenciar e escalar aplicativos em contêineres de forma eficiente.

Os principais motivos para a utilização do ECS Cluster para os serviços de agendamentos e pagamentos são:

- **Escalabilidade e Elasticidade:** O ECS permite que os serviços de agendamentos e pagamentos sejam altamente escaláveis e elásticos. Com o ECS, é possível ajustar automaticamente o número de instâncias de contêiner com base na demanda, garantindo que os serviços possam lidar com picos de tráfego sem degradação do desempenho.
- **Alta Disponibilidade:** O ECS Cluster distribui automaticamente os contêineres em várias instâncias do Amazon EC2, garantindo alta disponibilidade e resiliência. Se uma instância falhar, o ECS redistribui automaticamente os contêineres para outras instâncias saudáveis, minimizando o impacto na disponibilidade do serviço.
- **Gerenciamento Simplificado:** O ECS simplifica significativamente o gerenciamento dos contêineres e da infraestrutura subjacente. Com o ECS, não é necessário gerenciar manualmente a infraestrutura do servidor, pois o serviço cuida do provisionamento, dimensionamento e monitoramento das instâncias do EC2.

- **Implantação e Atualizações Contínuas:** O ECS facilita a implantação e as atualizações contínuas dos serviços de agendamentos e pagamentos. Os contêineres podem ser facilmente atualizados com novas versões do código ou configurações, sem causar tempo de inatividade significativo.
- **Integração com Outros Serviços da AWS:** O ECS se integra perfeitamente com outros serviços da AWS, como o Elastic Load Balancing para distribuição de tráfego e o AWS Auto Scaling para dimensionamento automático.
- **Portabilidade e Consistência:** Ao usar contêineres Docker com o ECS, os serviços de agendamentos e pagamentos podem ser facilmente portados entre diferentes ambientes, como desenvolvimento, teste e produção. Os contêineres encapsulam todas as dependências e configurações necessárias, garantindo consistência e eliminando problemas de compatibilidade entre ambientes.

O uso do ECS Cluster para os serviços de agendamentos e pagamentos oferece escalabilidade, alta disponibilidade, gerenciamento simplificado e integração perfeita com outros serviços da AWS. Essa abordagem permite que esses serviços críticos sejam executados de forma eficiente, confiável e escalável, proporcionando uma experiência de usuário aprimorada e atendendo às demandas crescentes do aplicativo. O ECS simplifica a implantação, o gerenciamento e a atualização dos contêineres, permitindo que a equipe se concentre no desenvolvimento de recursos valiosos para os usuários.

#### 4.3.3.3. AMAZON EC2 E AWS FARGATE

O AWS Fargate e o Amazon EC2 são dois serviços distintos da AWS que podem ser utilizados para executar aplicativos em contêineres. No entanto, cada um deles possui características e benefícios específicos que os tornam adequados para diferentes tipos de cargas de trabalho.

Para o serviço de pagamentos, optou-se pelo uso do AWS Fargate. O Fargate é um serviço serverless que permite a execução de contêineres sem a necessidade de gerenciar a infraestrutura subjacente. Isso significa que não é preciso provisionar, configurar ou escalar instâncias do EC2 manualmente. O Fargate cuida

automaticamente do provisionamento e do gerenciamento da infraestrutura necessária para executar os contêineres do serviço de pagamentos. Essa abordagem simplifica a implantação e o gerenciamento do serviço, permitindo que a equipe se concentre no desenvolvimento da aplicação em si.

Além disso, o Fargate oferece escalabilidade automática, ajustando dinamicamente os recursos alocados com base na demanda. Isso é especialmente importante para o serviço de pagamentos, que pode experimentar picos de tráfego em determinados momentos. O Fargate garante que o serviço seja capaz de lidar com aumentos repentinos na carga de trabalho sem a necessidade de intervenção manual.

Outra vantagem do Fargate é a integração perfeita com outros serviços da AWS, como o Amazon ECS para orquestração de contêineres e o AWS Identity and Access Management (IAM) para controle de acesso e segurança. Essa integração facilita a implementação de práticas recomendadas de segurança e o gerenciamento eficiente dos contêineres.

Por outro lado, para o serviço de agendamentos, decidiu-se utilizar o Amazon EC2. O EC2 oferece um controle mais granular sobre a infraestrutura subjacente, permitindo a personalização das instâncias de acordo com os requisitos específicos do serviço. Com o EC2, é possível escolher o tipo de instância mais adequado, definindo a quantidade de CPU, memória e armazenamento necessária. Isso é particularmente útil para o serviço de agendamentos, que pode exigir recursos dedicados e configurações específicas para atender aos requisitos de desempenho e confiabilidade.

Além disso, o uso do EC2 permite a instalação e configuração de bibliotecas, dependências e software adicional necessário para o funcionamento adequado do serviço de agendamentos. Essa flexibilidade é importante para garantir que o serviço seja executado de forma otimizada e compatível com outras partes do sistema.

O EC2 também oferece opções de escalabilidade, como o Auto Scaling, que permite ajustar automaticamente o número de instâncias com base na demanda. Isso garante que o serviço de agendamentos seja capaz de lidar com aumentos na carga de trabalho, mantendo a disponibilidade e o desempenho.

Outra vantagem do EC2 é a possibilidade de integração com outros serviços da AWS, como o Amazon RDS para armazenamento de dados relacionais. Essa integração é facilitada por meio de redes privadas virtuais (VPC) e grupos de segurança, permitindo uma comunicação segura e eficiente entre os componentes do sistema.

A escolha do AWS Fargate para o serviço de pagamentos e do Amazon EC2 para o serviço de agendamentos foi baseada nas características e requisitos específicos de cada serviço. O Fargate oferece simplicidade, escalabilidade automática e integração perfeita com outros serviços da AWS, sendo ideal para o serviço de pagamentos que requer alta disponibilidade e facilidade de gerenciamento. Já o EC2 proporciona maior controle, personalização e flexibilidade, sendo adequado para o serviço de agendamentos que pode exigir configurações específicas e recursos dedicados. Essa combinação permite aproveitar os pontos fortes de cada serviço, oferecendo uma arquitetura eficiente e escalável para o aplicativo de coleta de resíduos.

#### **4.4. CONTRIBUIÇÃO DO APLICATIVO**

O aplicativo de coleta de resíduos proposto tem como principal objetivo otimizar e modernizar o serviço atual da empresa, proporcionando benefícios tanto para os clientes quanto para a gestão interna. A digitalização dos processos torna a experiência do usuário mais conveniente, permitindo que ele agende coletas, realize pagamentos e acompanhe informações relevantes de maneira centralizada e digital.

Além disso, a empresa poderá alcançar maior eficiência operacional por meio da automação de processos e da melhoria na gestão financeira. O planejamento das rotas de coleta também será aprimorado, reduzindo custos operacionais e aumentando a produtividade da equipe.

Outro aspecto relevante do aplicativo é a promoção da consciência ambiental. Os usuários terão acesso a dados sobre seu impacto positivo na reciclagem, o que incentivará um maior engajamento na prática sustentável. A escalabilidade da solução, viabilizada pelo uso da computação em nuvem, permitirá que o serviço atenda a uma

demanda crescente e possibilite a expansão futura do negócio sem grandes investimentos iniciais.

Por fim, o aplicativo também fornecerá a funcionalidade de rastreamento em tempo real da localização dos veículos de coleta. Dessa forma, os usuários poderão visualizar o trajeto do veículo e obter uma estimativa do horário de chegada, aumentando a transparência e a eficiência do serviço.

#### **4.5. FUNCIONALIDADES DO APLICATIVO**

##### **4.5.1. AGENDAMENTO DE COLETAS**

Uma das principais funcionalidades é o agendamento de coletas, permitindo que os usuários programem a retirada de seus resíduos recicláveis e orgânicos diretamente pelo aplicativo. Eles podem selecionar o tipo de resíduo, escolher a data e o horário preferencial para a coleta e, assim que a solicitação for confirmada, receber notificações e lembretes para garantir que o descarte seja realizado corretamente.

##### **4.5.2. HISTÓRICO DE COLETAS**

Além do agendamento, o aplicativo mantém um histórico de coletas, onde os usuários podem acessar informações detalhadas sobre todas as coletas já realizadas. Esse histórico inclui a data, o tipo de resíduo descartado e a quantidade coletada, possibilitando que os usuários acompanhem sua participação no processo de reciclagem e identifiquem seus hábitos ao longo do tempo.

##### **4.5.3. CÁLCULO DO IMPACTO AMBIENTAL**

Outra funcionalidade relevante é o cálculo do impacto ambiental, que fornece métricas sobre o impacto positivo gerado pela reciclagem. O aplicativo apresenta dados como a quantidade de resíduos reciclados, a redução na emissão de gases de efeito estufa e a economia de recursos naturais. Essas informações visam conscientizar os usuários sobre a importância da reciclagem e incentivá-los a manter esse的习惯 sustentável.

#### **4.5.4. RASTREAMENTO DE VEÍCULOS DE COLETA**

Para melhorar a experiência do usuário, o aplicativo também disponibiliza o rastreamento dos veículos de coleta em tempo real. Os usuários podem acompanhar a localização do veículo responsável pela coleta, visualizar sua rota e obter uma estimativa do horário de chegada. Essa funcionalidade aumenta a transparência do serviço, permitindo que os clientes se organizem melhor para a entrega dos resíduos.

#### **4.5.5. PAGAMENTO E ASSINATURAS**

O aplicativo também oferece uma solução completa para a gestão de pagamentos e assinaturas. Os usuários podem optar por planos de assinatura mensais ou anuais, de acordo com suas necessidades. O pagamento é processado de forma segura dentro do próprio aplicativo, garantindo praticidade e controle sobre as transações.

#### **4.5.6. NOTIFICAÇÕES E LEMBRETES**

Para assegurar que os usuários estejam sempre informados sobre seus serviços, o aplicativo conta com um eficiente sistema de notificações e lembretes. As notificações alertam sobre coletas agendadas, chegada iminente do veículo de coleta e outras informações relevantes, mantendo os usuários organizados e engajados no processo de reciclagem.

#### **4.5.7. SUPORTE AO CLIENTE**

Por fim, a plataforma inclui um recurso de suporte ao cliente, oferecendo canais de atendimento como perguntas frequentes, chat ao vivo e opções de contato direto. Isso permite que os usuários esclareçam dúvidas e resolvam eventuais problemas de forma ágil e eficiente, garantindo uma experiência mais satisfatória e confiável.

## 4.6. FLUXO DE DADOS E INTERAÇÕES

Agora, vamos examinar como os diferentes componentes da arquitetura (mostrado no diagrama da figura 2) interagem e como os dados fluem entre eles para cada funcionalidade principal:

### **Interação do Usuário:**

O usuário interage com o aplicativo móvel, que é entregue pelo Amazon CloudFront a partir dos arquivos estáticos armazenados no Amazon S3. As ações realizadas podem incluir login, agendamento de coleta, visualização do histórico, rastreamento de veículos, entre outras funcionalidades.

### **Autenticação e Autorização:**

Ao fazer login, o usuário tem suas credenciais validadas pelo Amazon Cognito. Se as credenciais forem aceitas, o Cognito retorna tokens de acesso e ID, que são utilizados nas solicitações futuras ao backend. Esses tokens garantem um processo seguro de autenticação e são incluídos nos cabeçalhos das requisições enviadas ao Amazon API Gateway, assegurando a autorização do usuário.

### **Solicitações de API:**

Toda vez que o usuário realiza uma ação que requer processamento no backend, como o agendamento de uma coleta, o aplicativo móvel envia uma solicitação HTTP ao Amazon API Gateway. Esse serviço recebe a requisição, valida a autenticação e roteia a solicitação para o serviço correspondente, garantindo um fluxo seguro e eficiente de dados.

### **Processamento de Agendamentos:**

Quando um usuário agenda uma coleta, o Amazon API Gateway encaminha a solicitação ao Elastic Load Balancer, que a direciona para o serviço de agendamentos em execução no Amazon ECS. O contêiner Docker do serviço processa a requisição e armazena os dados no Amazon RDS. Após a confirmação do agendamento, a resposta percorre o caminho inverso, retornando ao aplicativo móvel.

### **Processamento de Pagamentos:**

Se a funcionalidade envolver um pagamento, como a confirmação de uma assinatura, a solicitação segue um fluxo semelhante. O Amazon API Gateway recebe a requisição e a encaminha ao Elastic Load Balancer, que a distribui para o serviço de pagamentos hospedado no Amazon ECS via AWS Fargate. O processamento da transação ocorre de forma segura, com interação com gateways de pagamento externos e armazenamento dos registros no Amazon DynamoDB. O status da operação é então retornado ao usuário pelo mesmo fluxo de comunicação.

### **Cálculo do Impacto Ambiental:**

Para fornecer métricas sobre a reciclagem realizada pelo usuário, o aplicativo envia uma requisição ao Amazon API Gateway, que invoca uma função Lambda específica para o cálculo do impacto ambiental. Essa função recupera informações sobre as coletas no Amazon DynamoDB, executa os cálculos necessários e retorna os resultados ao aplicativo, permitindo que o usuário visualize dados sobre sua contribuição sustentável.

### **Rastreamento de Veículos:**

O rastreamento de veículos ocorre de maneira contínua, garantindo que os usuários tenham acesso à localização atualizada dos caminhões de coleta. Durante o trajeto, os veículos enviam periodicamente suas posições ao Amazon API Gateway, que aciona uma função Lambda para processar os dados de geolocalização. A função armazena o histórico de posições no Amazon DynamoDB e consulta o Amazon Location Service para informações detalhadas de localização. Quando um usuário solicita a posição atual de um veículo, o aplicativo acessa o Amazon API Gateway, que retorna a última localização registrada, garantindo transparência e previsibilidade no serviço.

### **Monitoramento:**

O Amazon CloudWatch monitora continuamente todos os componentes da arquitetura, coletando métricas, logs e eventos. Ele acompanha a utilização de CPU e

memória no Amazon ECS, a duração das funções Lambda, a latência e as taxas de erro do Amazon API Gateway, além de gerar logs detalhados sobre acessos e execuções dos serviços. Com essas informações, é possível detectar problemas de desempenho, definir alarmes e implementar ações corretivas para manter a estabilidade do sistema.

#### **4.7. NOTIFICAÇÃO DE CHEGADA DO VEÍCULO DE COLETA**

Para notificar os clientes sobre a chegada iminente do veículo de coleta, usaremos o Amazon Simple Notification Service (SNS) em conjunto com o AWS Lambda e o Amazon CloudWatch Events.

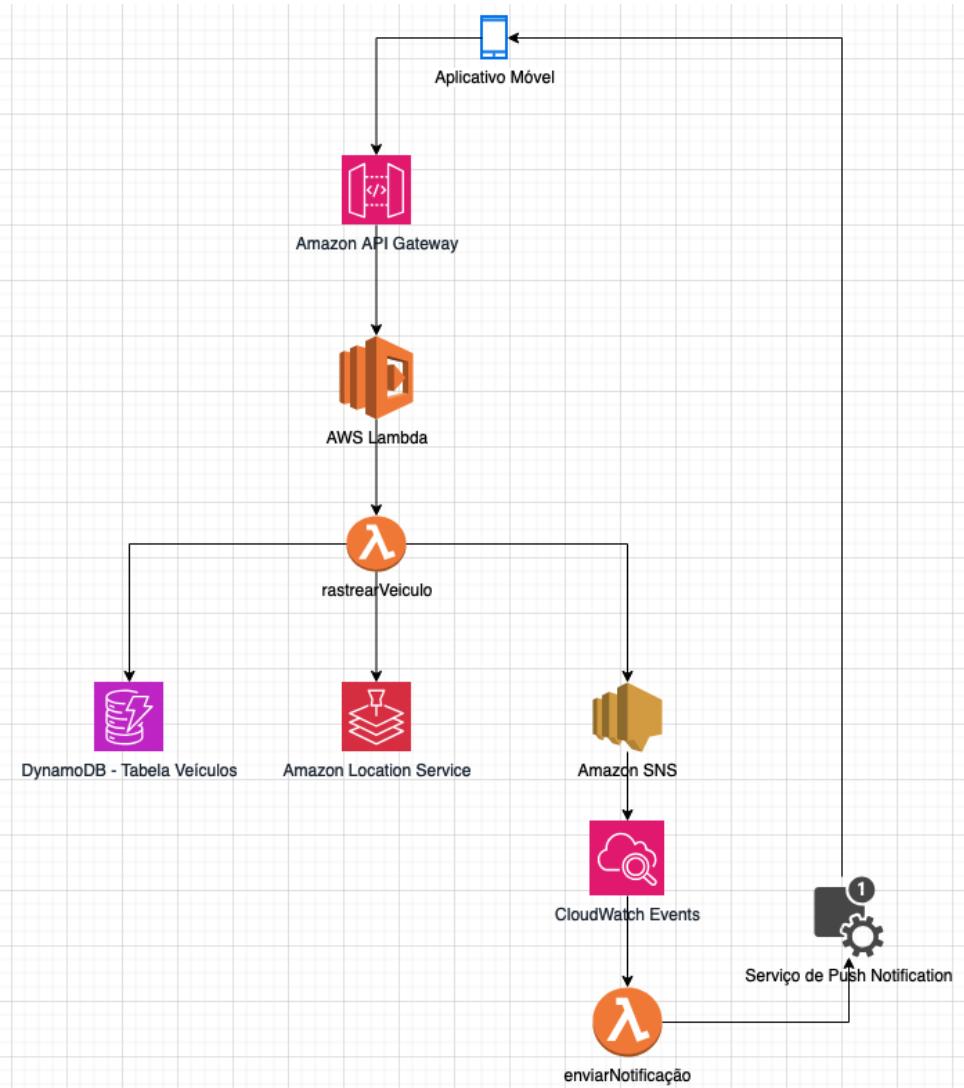
A seguir, o fluxo de notificação:

1. O aplicativo móvel do motorista enviará atualizações periódicas da localização do veículo para o backend usando o endpoint apropriado do API Gateway.
2. O API Gateway acionará a função Lambda `rastrearVeiculo`, que processará a atualização de localização e a armazenará na tabela `Veiculos` do DynamoDB.
3. A função Lambda `rastrearVeiculo` também calculará a distância ou o tempo estimado até o endereço de coleta do cliente com base na localização atual do veículo.
4. Se a distância ou o tempo estimado estiver abaixo de um limite predefinido (por exemplo, 1 km ou 5 minutos), a função Lambda publicará uma mensagem no tópico do Amazon SNS relacionado a notificações de chegada.
5. O Amazon CloudWatch Events será configurado para acionar outra função Lambda, `enviarNotificacao`, sempre que uma mensagem for publicada no tópico do SNS.
6. A função Lambda `enviarNotificacao` usará os endpoints de push notification (por exemplo, APNs para iOS e FCM para Android) para enviar uma notificação push para o dispositivo móvel do cliente, informando que o veículo de coleta está próximo.

7. O aplicativo móvel do cliente receberá a notificação push e exibirá uma mensagem apropriada na interface do usuário, informando que o veículo de coleta está a caminho.

A seguir, na figura 3, temos o arquitetura dessa funcionalidade:

Figura 3 - Diagrama de arquitetura para a funcionalidade de rastreamento em tempo real



Fonte: Autor

## 4.8. PIPELINE DE DESENVOLVIMENTO E DEPLOY

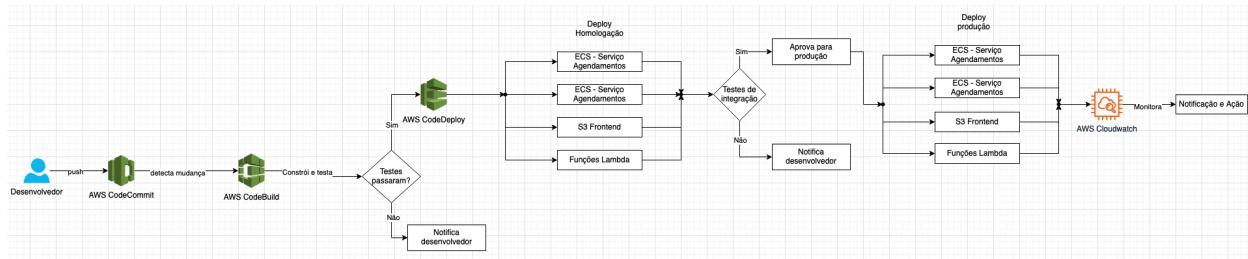
Por fim, para garantir um desenvolvimento ágil e entregas frequentes, será utilizado um pipeline de CI/CD (Continuous Integration / Continuous Deployment) utilizando os serviços AWS CodeCommit, AWS CodeBuild e aws CodeDeploy, conforme apresentado, abaixo, na figura 4.

O código do aplicativo ficará armazenado em um repositório no CodeCommit. A cada novo commit, o CodeBuild Actions será acionado para executar os testes automatizados, compilar o código e gerar os artefatos prontos para deploy.

Caso os testes passem, o CodeDeploy automaticamente fará o deploy dos artefatos nos ambientes adequados, seja em staging para testes adicionais ou em produção.

Esse pipeline automatizado permite um ciclo de desenvolvimento e lançamento rápido e confiável, garantindo a qualidade do aplicativo.

Figura 4 - Modelo de arquitetura para o pipeline



Fonte: Autor

Descrição do fluxo:

1. O desenvolvedor faz push do código para o AWS CodeCommit, que é um serviço de controle de versão totalmente gerenciado.
2. O AWS CodeCommit detecta o novo push e aciona automaticamente o AWS CodeBuild.
3. O AWS CodeBuild compila o código, executa testes unitários e empacota a aplicação.

4. Se os testes passarem, o AWS CodeBuild aciona o AWS CodeDeploy para implantar a aplicação nos ambientes de homologação. Caso contrário, o desenvolvedor é notificado sobre a falha nos testes.
5. O AWS CodeDeploy implanta os serviços de agendamentos e pagamentos no Amazon ECS, o frontend no Amazon EC2 e no Amazon S3, e as funções Lambda no AWS Lambda, todos no ambiente de homologação.
6. Após a implantação em homologação, testes de integração são executados nos serviços, no frontend e nas funções Lambda.
7. Se os testes de integração passarem, a implantação em produção é aprovada. Caso contrário, o desenvolvedor é notificado.
8. Se aprovado, o AWS CodeDeploy implanta os serviços, o frontend e as funções Lambda nos ambientes de produção (ECS, EC2, S3 e Lambda).
9. O Amazon CloudWatch monitora todos os componentes implantados em produção, gerando notificações e acionando ações com base em métricas e eventos predefinidos.

## 4.9. MOCKUP DO APLICATIVO MOBILE

### 4.9.1. TELA INICIAL

Figura 5 - Tela inicial do aplicativo



Fonte: Autor

Figura 6 - Tela do menu lateral do aplicativo



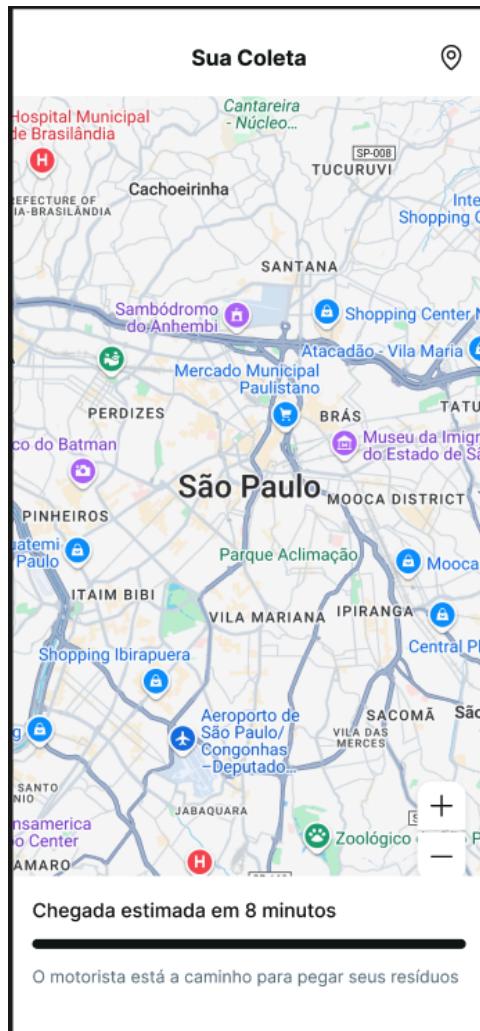
Fonte: Autor

Quando a tela é carregada, o aplicativo móvel envia uma solicitação ao Amazon API Gateway para obter os dados do usuário e as informações de coleta. O API Gateway, por sua vez, aciona a função Lambda apropriada para recuperar os dados do usuário e os detalhes da próxima coleta agendada do Amazon DynamoDB. A função Lambda também calcula o total de lixo orgânico e recicláveis coletados no mês atual, buscando essas informações no DynamoDB. Em seguida, os dados são retornados ao aplicativo móvel e exibidos na tela, conforme figura 5, incluindo as quantidades de

coleta e a data da próxima coleta. Quando o usuário clica no botão "Agendar Coleta", conforme figura 6, uma nova solicitação é enviada ao API Gateway, que aciona o serviço de agendamentos no Amazon ECS para criar um novo agendamento no banco de dados Amazon RDS.

#### 4.9.2. TELA "SUA COLETA"

Figura 7 - Tela de rastreamento em tempo real



Fonte: Autor

Ao acessar a tela "Sua Coleta" para rastreamento da coleta em andamento, o aplicativo móvel faz uma solicitação ao Amazon API Gateway para obter a localização atual do veículo de coleta. O API Gateway aciona a função Lambda responsável por rastrear veículos, que consulta o DynamoDB para obter a última localização conhecida do veículo. A função Lambda também interage com o Amazon Location Service para realizar cálculos de geocodificação reversa e estimar o tempo de chegada com base nas localizações do veículo e do usuário. As informações de localização e tempo estimado são então retornadas ao aplicativo móvel e exibidas no mapa e na mensagem de status. Periodicamente, o aplicativo móvel envia solicitações ao API Gateway para atualizar a localização do veículo e o tempo estimado de chegada, mantendo o usuário informado sobre o progresso da coleta, conforme figura 7.

#### 4.9.3. TELA DO CÁLCULO DO IMPACTO

Figura 8 - Tela do cálculo de impacto ambiental

**Calculadora de impacto**

Informe o peso aproximado dos materiais que você reciclou. Depois é só compartilhar o resultado com os amigos!

Insira a quantidade em quilos

	Papel e Papelão	0,00 Kg
	Vidro	0,00 Kg
	Alumínio	0,00 Kg
	Outros Metais	0,00 Kg
	Plástico	0,00 Kg
	Embalagem longa vida	0,00 Kg
	Baterias, pilhas, eletrônicos em geral	0,00 Kg

**Ver Resultado**

Fonte: Autor

Figura 9: Tela de resultados do impacto ambiental



Fonte: Autor

Na tela "Calculadora de impacto" (figura 8), quando o usuário insere as quantidades de materiais recicláveis (em kg) e clica em "Ver Resultado", o aplicativo móvel envia uma solicitação para a função Lambda "calcularImpacto" por meio do Amazon API Gateway. Essa função Lambda é responsável por processar os dados fornecidos pelo usuário e realizar os cálculos para determinar o impacto ambiental.

O cálculo do impacto ambiental no aplicativo pode ser realizado com base nas quantidades de resíduos reciclados pelos usuários, associando essas quantidades a fatores de equivalência que estimam os benefícios ambientais gerados pela reciclagem. Para isso, utiliza-se dados consolidados em estudos ambientais que atribuem valores médios de economia de recursos e redução de poluentes para cada tipo de material reciclado. Para cada tipo de material reciclado, aplica-se a seguinte fórmula:

$$ImpactoAmbiental = QuantidadeReciclada (kg) \times FatorDeEquivalência$$

Cada material possui um fator de equivalência específico, que representa a média dos benefícios obtidos ao evitar que o resíduo seja enviado para aterros sanitários ou descarte inadequado. Segundo dados do Instituto Akatu (2022) e da EPA (Environmental Protection Agency, 2021), os seguintes valores médios podem ser aplicados:

Material	Redução de CO <sub>2</sub> (kg por kg reciclado)	Economia de Energia (kWh)	Água Economizada (litros)
Papel	1,5 kg	4,1 kWh	26 litros
Plástico	1,5 kg	5,7 kWh	2 litros
Alumínio	9,0 kg	14,0 kWh	4 litros
Vidro	0,3 kg	0,7 kWh	1 litro
Orgânico *	0,25 kg (metano evitado)	-	-

Tabela 6 – Fatores de Equivalência para Materiais Recicláveis

\* Para resíduos orgânicos destinados à compostagem, considera-se a redução da emissão de metano ( $\text{CH}_4$ ), um gás de efeito estufa com potencial de aquecimento global significativamente superior ao do dióxido de carbono.

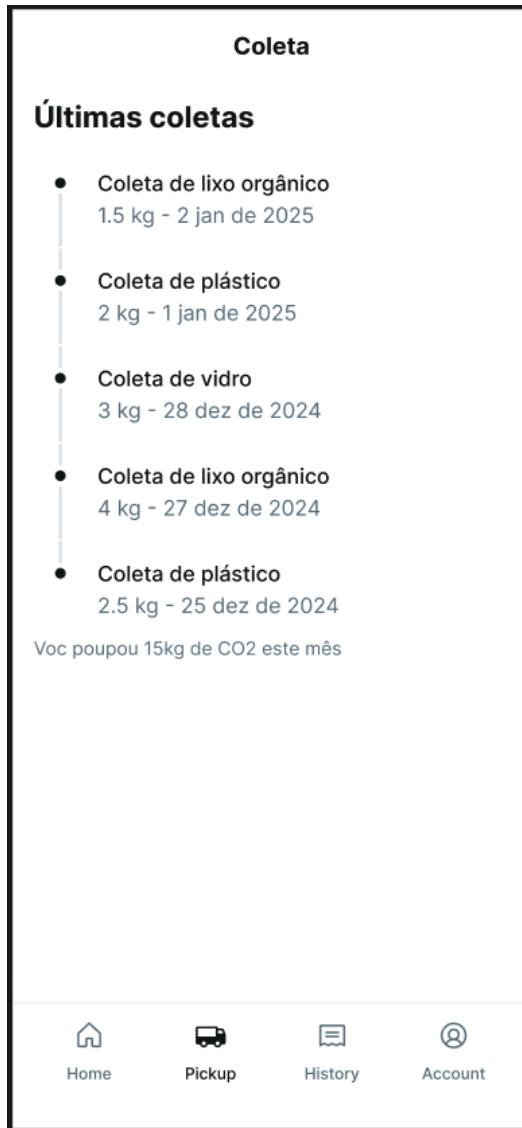
Dessa forma, ao registrar a quantidade de resíduos reciclados no aplicativo, o usuário recebe um retorno imediato sobre seu impacto positivo no meio ambiente. Por exemplo, ao reciclar 5 kg de papel, 2 kg de plástico e 1 kg de alumínio em um mês, o sistema calcula automaticamente a redução total de emissões de  $\text{CO}_2$ :

- Papel:  $5 \text{ kg} \times 1,5 \text{ kg CO}_2 = 7,5 \text{ kg CO}_2$  evitados
- Plástico:  $2 \text{ kg} \times 1,5 \text{ kg CO}_2 = 3 \text{ kg CO}_2$  evitados
- Alumínio:  $1 \text{ kg} \times 9 \text{ kg CO}_2 = 9 \text{ kg CO}_2$  evitados

Totalizando 19,5 kg de  $\text{CO}_2$  evitados no período. Após realizar os cálculos, a função Lambda retorna os resultados para o aplicativo móvel por meio do Amazon API Gateway. O aplicativo, por sua vez, exibe os resultados na tela "Resultados de Reciclagem", conforme figura 9.

#### 4.9.4. TELA DO HISTÓRICO

Figura 10: Tela de histórico de coletas



Fonte: Autor

Quando o usuário acessa a tela "Coleta" para visualizar o histórico de coletas, o aplicativo móvel envia uma solicitação ao Amazon API Gateway para obter os dados das coletas anteriores. O API Gateway aciona a função Lambda responsável por recuperar o histórico de coletas do usuário do DynamoDB, e o resultado é mostrado na tela, conforme figura 10.

## **5. ANÁLISE DE CUSTOS DA IMPLEMENTAÇÃO NA AWS CLOUD**

### **5.1. INTRODUÇÃO**

Este capítulo apresenta uma estimativa de custos baseada no uso dos principais serviços AWS descritos no projeto, levando em conta cenários de demanda inicial e de expansão futura.

### **5.2. SERVIÇOS AWS NECESSÁRIOS**

Os principais serviços AWS para a implementação do aplicativo são:

1. Amazon EC2: Usado para hospedar o serviço de agendamentos em contêineres Docker.
2. Amazon ECS: Utilizado para orquestrar e gerenciar os contêineres Docker dos serviços de agendamentos e pagamentos.
3. AWS Fargate: Empregado para executar os contêineres do serviço de pagamentos sem a necessidade de gerenciar a infraestrutura subjacente.
4. AWS Lambda: Utilizado para executar funções serverless, como cálculo do impacto ambiental e rastreamento de veículos.
5. Amazon API Gateway: Atua como o ponto de entrada para todas as solicitações do aplicativo móvel e gerencia a integração com os serviços de backend.
6. Amazon DynamoDB: Usado como banco de dados NoSQL para armazenar informações sobre coletas, usuários e veículos.
7. Amazon RDS: Empregado como banco de dados relacional para armazenar dados dos agendamentos.
8. Amazon S3: Utilizado para armazenar os arquivos estáticos do frontend do aplicativo.
9. Amazon CloudFront: Empregado como CDN para entregar o conteúdo estático do frontend de forma rápida e eficiente.
10. Amazon Cognito: Utilizado para autenticação e autorização dos usuários.

11. Amazon CloudWatch: Usado para monitorar os recursos e serviços da AWS, coletar métricas e logs.

### **5.3. ESTIMATIVA DE CUSTOS MENSAIS**

Para calcular os custos da implementação do aplicativo na AWS, foram considerados dois cenários: o de lançamento (com um número reduzido de usuários e operações) e o de expansão (com crescimento significativo na base de usuários e no volume de transações). As estimativas a seguir foram calculadas utilizando o AWS Pricing Calculator, com base nos serviços necessários para o projeto e configurações típicas.

Se cada veículo de coleta transmitir sua localização a cada 30 segundos durante um turno médio de 8 horas por dia e houver, por exemplo, 10 veículos operando simultaneamente, teremos:

- Cálculo das Requisições de Rastreamento Diárias por Veículo:
  - 8 horas = 28.800 segundos
  - $28.800 \text{ segundos} \div 30 \text{ segundos} = 960 \text{ atualizações por veículo por dia.}$
- Requisições Totais por Mês (30 dias):
  - $960 \text{ atualizações diárias por veículo} \times 10 \text{ veículos} \times 30 \text{ dias} = 288.000 \text{ requisições/mês.}$

Se adicionarmos 10.000 requisições mensais para outras funcionalidades (como agendamentos, pagamentos, e consultas gerais no aplicativo), o total mensal seria 298.000 requisições.

Portanto, para um cenário inicial de 1000 usuários ativos por mês, teremos:

Serviço	Uso Estimado	Custo Mensal (USD)
Amazon API Gateway	298.000 requisições	\$10,43
AWS Lambda	288.000 execuções (duração média de 200 ms)	\$0,42
Amazon DynamoDB	10 GB de armazenamento + 1000 unidades de leitura/escrita	\$10,00
Amazon S3	50 GB de armazenamento + 100.000 solicitações	\$2,50
Amazon Cognito	10.000 usuários ativos	\$0,00
Amazon Location Service	50.000 chamadas de rastreamento	\$4,50
Amazon CloudWatch	Logs e métricas adicionais	\$5,00
Amazon SNS	10.000 notificações push	\$1,00
Total		\$33,85

Tabela 7 - Custo estimado mensal dos serviços da AWS

#### 5.4. BENEFÍCIOS ECONÔMICOS DA AWS

- **Escalabilidade de Custos**

A AWS permite começar com custos mínimos (aproveitando o Free Tier, que cobre muitos serviços no primeiro ano) e expandir os serviços conforme a demanda aumenta, evitando despesas iniciais excessivas.

- **Modelo Pay-as-you-go**

Os custos são diretamente proporcionais ao uso, garantindo que a empresa pague apenas pelos recursos efetivamente consumidos.

- **Créditos AWS para Startups**

A AWS oferece créditos para startups e projetos inovadores, que podem ser utilizados para cobrir custos iniciais, reduzindo ainda mais o impacto financeiro no lançamento. É possível aproveitar incentivos como créditos para startups, dependendo do programa da AWS em que sua empresa se inscreve, como o AWS Activate.

## 6. CONCLUSÃO

O presente trabalho apresentou uma proposta de arquitetura de software para um aplicativo de reciclagem, com foco na utilização de tecnologias de computação em nuvem para resolver desafios relacionados à gestão de resíduos sólidos. A partir de um diagnóstico das limitações enfrentadas por empresas de coleta de resíduos, especialmente no contexto brasileiro, identificou-se a oportunidade de modernizar operações por meio da digitalização e automação de processos.

A escolha da AWS como base tecnológica mostrou-se fundamental para a solução, devido às suas características de escalabilidade, flexibilidade, segurança e custo-benefício. Os serviços AWS, como Lambda, DynamoDB, ECS, RDS, Cognito e Location Service, foram detalhados e integrados em uma arquitetura serverless projetada para atender às demandas de desempenho, confiabilidade e experiência do usuário. Além disso, funcionalidades como agendamento de coletas, notificações em tempo real e análise de impacto ambiental foram incorporadas para oferecer uma solução completa e alinhada aos princípios de sustentabilidade.

O desenvolvimento de um aplicativo com essas características transcende o objetivo de oferecer conveniência aos usuários. Ele atua como um instrumento de conscientização ambiental, incentivando práticas sustentáveis e promovendo um impacto positivo na sociedade. A integração da tecnologia com a sustentabilidade, como demonstrado, não apenas melhora a eficiência operacional, mas também estimula um comportamento ambientalmente responsável, tanto em nível individual quanto coletivo.

A análise de custos confirmou a viabilidade econômica do projeto, especialmente em cenários de expansão, onde o modelo "pay-as-you-go" da AWS se destaca como uma vantagem competitiva. Adicionalmente, o uso de créditos para startups e a flexibilidade de adaptação às demandas futuras tornam essa abordagem altamente escalável e economicamente sustentável.

Além disso, a adoção desta arquitetura possibilita à empresa coletora acessar o mercado de créditos de carbono, ao permitir a quantificação precisa e certificada das reduções de emissões resultantes das práticas de reciclagem e compostagem. O aplicativo fornece a infraestrutura necessária para documentar estas reduções, transformando benefícios ambientais em potencial fonte de receita adicional e fortalecendo ainda mais a viabilidade econômica do projeto.

Conclui-se que a arquitetura proposta não apenas atende às necessidades imediatas da empresa de coleta de resíduos, mas também estabelece um modelo replicável e escalável que pode ser aplicado em outras regiões e contextos. Assim, este trabalho contribui para a interseção entre tecnologia e sustentabilidade, oferecendo uma solução prática para um dos maiores desafios ambientais da atualidade: a gestão eficiente de resíduos sólidos.

## 7. REFERÊNCIA BIBLIOGRÁFICA

- AMAZON WEB SERVICESa. Amazon API Gateway. Disponível em:  
<https://aws.amazon.com/api-gateway/>
- AMAZON WEB SERVICESb. Amazon CloudFront. Disponível em:  
<https://aws.amazon.com/cloudfront/>
- AMAZON WEB SERVICESc. Amazon CloudWatch. Disponível em:  
<https://aws.amazon.com/cloudwatch/>
- AMAZON WEB SERVICESd. Amazon Cognito. Disponível em:  
[https://aws.amazon.com/cognito/.](https://aws.amazon.com/cognito/)
- AMAZON WEB SERVICESe. Amazon DynamoDB. Disponível em:  
[https://aws.amazon.com/dynamodb/.](https://aws.amazon.com/dynamodb/)
- AMAZON WEB SERVICESf. Amazon EC2. Disponível em:  
[https://aws.amazon.com/ec2/.](https://aws.amazon.com/ec2/)
- AMAZON WEB SERVICESg. Amazon ECS. Disponível em:  
<https://aws.amazon.com/ecs/>
- AMAZON WEB SERVICESh. Amazon Location Service. Disponível em:  
[https://aws.amazon.com/location/.](https://aws.amazon.com/location/)
- AMAZON WEB SERVICESi. Amazon RDS. Disponível em:  
[https://aws.amazon.com/rds/.](https://aws.amazon.com/rds/)
- AMAZON WEB SERVICESj. Amazon S3. Disponível em: <https://aws.amazon.com/s3/>
- AMAZON WEB SERVICESl. AWS Fargate. Disponível em:  
<https://aws.amazon.com/fargate/>
- AMAZON WEB SERVICESm. AWS Lambda. Disponível em:  
[https://aws.amazon.com/lambda/.](https://aws.amazon.com/lambda/)

BUYYA, R.; YEO, C. S.; VENUGOPAL, S. Market-oriented cloud computing: Vision, hype, and reality for delivering IT services as computing utilities. In: IEEE INTERNATIONAL CONFERENCE ON HIGH PERFORMANCE COMPUTING AND COMMUNICATIONS, 10., 2008, Dalian. Proceedings [...]. Dalian: IEEE, 2008. p. 5-13.

DOCKER. What is a Container? Disponível em: <https://www.docker.com/resources/what-container>

KUMAR, R.; GOYAL, R. On cloud computing with AWS: a review on Amazon Web Services. In: IEEE INTERNATIONAL CONFERENCE ON INTELLIGENT COMPUTING AND CONTROL SYSTEMS (ICICCS), 4., 2020, Madurai. Proceedings [...]. Madurai: IEEE, 2020. p. 388-393.

LIU, F. et al. A survey on edge computing systems and tools. *Proceedings of the IEEE*, v. 107, n. 8, p. 1537-1562, 2019.

MELL, P.; GRANCE, T. The NIST definition of cloud computing. Gaithersburg: National Institute of Standards and Technology, 2011. (NIST Special Publication, 800-145).

ROBERTS, M.; CHAPIN, J. *What is Serverless?* Sebastopol: O'Reilly Media, 2017. Disponível em: <https://www.oreilly.com/library/view/what-is-serverless/9781491984178/>.

ROBINSON, D. *AWS Lambda in Action: Event-driven serverless applications*. Shelter Island: Manning Publications, 2017.

SBARSKI, P.; KROONENBURG, S. *Serverless Architectures on AWS: With examples using AWS Lambda*. 2. ed. New York: Manning Publications, 2021.

VARGHESE, B.; BUYYA, R. Next generation cloud computing: new trends and research directions. *Future Generation Computer Systems*, v. 79, p. 849-861, 2018.

INSTITUTO AKATU. Reciclagem e seus benefícios. Disponível em: <https://www.akatu.org.br/>

U.S. ENVIRONMENTAL PROTECTION AGENCY (EPA). Greenhouse Gas Equivalencies Calculator, 2021. Disponível em:  
<https://www.epa.gov/energy/greenhouse-gas-equivalencies-calculator>

ABRELPE. Panorama dos Resíduos Sólidos no Brasil 2022. Disponível em:  
<https://abrelpe.org.br/>

Bass, L., Clements, P., & Kazman, R. (2012). Software Architecture in Practice (3rd ed.). Addison-Wesley Professional.

Fehling, C., Leymann, F., Retter, R., Schupeck, W., & Arbitter, P. (2014). Cloud Computing Patterns: Fundamentals to Design, Build, and Manage Cloud Applications. Springer.

Opara-Martins, J., Sahandi, R., & Tian, F. (2016). Critical analysis of vendor lock-in and its impact on cloud computing migration: a business perspective. Journal of Cloud Computing.