

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA
PARAÍBA CAMPUS CAJAZEIRAS
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO
DE SISTEMAS**

**CLOUD COMPARE: UMA FERRAMENTA DE COMPARAÇÃO DE CUSTOS DE
IMPLANTAÇÃO DE SOFTWARE EM NUVENS DO TIPO PLATAFORMA COMO
SERVIÇO (PaaS)**

CAMILA CASIMIRO CALIXTO

**Cajazeiras
2024**

CAMILA CASIMIRO CALIXTO

**CLOUD COMPARE: UMA FERRAMENTA DE COMPARAÇÃO DE CUSTOS DE
IMPLANTAÇÃO DE SOFTWARE EM NUVENS DO TIPO PLATAFORMA COMO
SERVIÇO (PaaS)**

Trabalho de Conclusão de Curso apresentado junto ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - Campus Cajazeiras, como requisito à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador Prof. MSc. Paulo Ewerton
Gomes Fragoso.

Cajazeiras

2024

IFPB / Campus Cajazeiras
Coordenação de Biblioteca
Biblioteca Prof. Ribamar da Silva
Catalogação na fonte: Cícero Luciano Félix CRB-15/750

| | |
|-------|---|
| C154c | <p>Calixto, Camila Casimiro. Cloud Compare : uma ferramenta de comparação de custos de implantação de software em nuvens do tipo plataforma como serviço (PaaS) / Camila Casimiro Calixto. – 2024.</p> <p>38f. : il.</p> <p>Trabalho de Conclusão de Curso (Tecnólogo em Análise e Desenvolvimento de Sistemas) - Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, Cajazeiras, 2024.</p> <p>Orientador(a): Prof. Me. Paulo Ewerton Gomes Fragoso.</p> <p>1. Desenvolvimento de sistemas. 2. Plataforma Cloud compare. 3. Computação em nuvem. 4. Comparação de preço. I. Instituto Federal de Educação, Ciência e Tecnologia da Paraíba. II. Título.</p> |
|-------|---|



MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA

CAMILA CASIMIRO

**CLOUD COMPARE: UMA FERRAMENTA DE COMPARAÇÃO DE CUSTOS DE
IMPLANTAÇÃO DE SOFTWARE EM NUVENS DO TIPO PLATAFORMA COMO SERVIÇO
(PaaS)**

Trabalho de Conclusão de Curso apresentado junto ao
Curso Superior de Tecnologia em Análise e
Desenvolvimento de Sistemas do Instituto Federal de
Educação, Ciência e Tecnologia da Paraíba - Campus
Cajazeiras, como requisito à obtenção do título de
Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador

Prof. Me. Paulo Ewerton Gomes Fragoso

Aprovada em: **18 de Outubro de 2024.**

Prof. Me. Paulo Ewerton Gomes Fragoso - Orientador

Prof. Me. Diogo Dantas Moreira - Avaliador

IFPB - Campus Cajazeiras

Prof. Esp. Asheley Emy Lacerda Alves

IFPB - Campus Cajazeiras

Documento assinado digitalmente

gov.br

ASHELEY EMMY LACERDA ALVES
Data: 21/10/2024 11:37:42-0300
Verifique em <https://validar.iti.gov.br>

Documento assinado eletronicamente por:

- **Paulo Ewerton Gomes Fragoso**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 19/10/2024 14:08:51.
- **Diogo Dantas Moreira**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 20/10/2024 15:25:13.

Este documento foi emitido pelo SUAP em 19/10/2024. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código 622363

Verificador: f95cc945e6

Código de Autenticação:



Rua José Antônio da Silva, 300, Jardim Oásis, CAJAZEIRAS / PB, CEP 58.900-000
<http://ifpb.edu.br> - (83) 3532-4100

RESUMO

A Plataforma como Serviço (PaaS) é um modelo de computação em nuvem que oferece agilidade para realizar a implantação das aplicações, ao reduzir a responsabilidade do desenvolvedor em tarefas de configuração de infraestrutura, o que contribui para um ganho de tempo e melhor foco no desenvolvimento do sistema. O modelo PaaS ainda promove flexibilidade para dimensionar e escalar recursos, contribuindo para a redução de custos, uma vez que permite pagar apenas pelo que é utilizado, evitando desperdício e uso inadequado de recursos. Assim, com a popularização da computação em nuvem entre empresas e desenvolvedores interessados neste tipo de serviço para implantar seu sistema e, considerando a disponibilidade crescente desse tipo de plataforma, esta pesquisa propôs-se a implementar uma aplicação web de comparação de custos entre um subconjunto de empresas que oferecem o serviço de PaaS a saber: Heroku, Render e Railway. A ideia é que a ferramenta contribua com as demandas de empresas de pequeno porte e desenvolvedores independentes, realizando uma comparação de preços entre plataformas de serviços PaaS populares selecionadas, a partir de métricas de recursos computacionais, tais como: CPU, memória, rede e armazenamento de banco de dados relacionais. Ao inserir as métricas desejadas, a aplicação desenvolvida em forma de assistente fornecerá uma comparação aproximada, ordenada por plataforma, plano e preço, facilitando a escolha dos recursos de nuvem mais adequados e otimizando o processo de tomada de decisão.

Palavras-Chaves: Computação em Nuvem; Plataforma como Serviço (PaaS); Comparação de preço.

ABSTRACT

The Platform as a Service (PaaS) is a cloud computing model that offers agility to carry out a deployment of the applications, since it removes its responsibility for infrastructure configurations, letting them have a gain of time and focus on the development and deployment of the system. It also has the flexibility to scale resources as well, contributing to cost reduction, as it allows you to pay only for what is being used, avoiding waste and inappropriate use of resources. In this way, with the popularization of cloud computing among companies and developers who are interested in this service to deploy their system, and the various platforms as a service, the project proposes to implement a web system, which seeks to meet the demands of small businesses and independent developers, performing a price comparison between several PaaS service platforms, based on computational resource metrics, such as: CPU, memory, network, and relational database storage. Inserting some metrics, the developed application as assistant will provide an approximated comparison ordered by platform, plan, and price, so having an easier choice of properly cloud resources and optimizing the decision making process.

Keywords: Cloud Computing; Platform as a Service (PaaS); Price comparison.

LISTAS DE FIGURAS

| | |
|---|----|
| Figura 2 - Arquitetura da aplicação..... | 24 |
| Figura 3 - Modelo Conceitual..... | 25 |
| Figura 4 - Código..... | 27 |
| Figura 5 - Swagger..... | 28 |
| Figura 6 - Rota configList..... | 29 |
| Figura 7 - Home..... | 30 |
| Figura 8 - Informar quantidade de RAM..... | 30 |
| Figura 9 - Informar quantidade de vCpu..... | 31 |
| Figura 10 - Armazenamento banco de dados..... | 32 |
| Figura 11 - Recomendações..... | 32 |

LISTA DE TABELAS

| | |
|---|----|
| Tabela 1: Comparação entre serviços de computação em nuvem..... | 18 |
| Tabela 2: Descrição das User Stories..... | 22 |
| Tabela 3: Cronograma..... | 22 |

LISTAS DE ABREVIATURAS E SIGLAS

| | |
|------|--|
| IaaS | Infraestrutura como Serviço |
| IP | Protocolo Internet |
| PaaS | Plataforma como Serviço |
| RAM | Memória de Acesso Aleatório |
| SaaS | Software como Serviço |
| SQL | Linguagem de Consulta Estruturada |
| TCC | Trabalho de Conclusão de Curso |
| TCP | Protocolo de Controle de Transmissão |
| vCPU | Unidade Central de Processamento Virtual |

SUMÁRIO

| | |
|---|-----------|
| 1 INTRODUÇÃO | 9 |
| 1.1 OBJETIVOS..... | 9 |
| 1.1.1 Objetivo Geral..... | 9 |
| 1.1.2 Objetivos Específicos..... | 10 |
| 1.2 JUSTIFICATIVA..... | 10 |
| 2 FUNDAMENTAÇÃO TEÓRICA | 11 |
| 2.1 HISTÓRICO DA COMPUTAÇÃO EM NUVEM..... | 11 |
| 2.2 MODELOS DE DISPONIBILIZAÇÃO DE COMPUTAÇÃO EM NUVEM..... | 13 |
| 2.2.1 Infraestrutura como Serviço (IaaS):..... | 14 |
| 2.2.2 Plataforma como Serviço (PaaS):..... | 15 |
| 2.2.3 Software como Serviço (SaaS):..... | 15 |
| 2.3 INTRODUÇÃO A NUVENS DO TIPO PaaS..... | 16 |
| 2.3.1 Levantamento de custos associados ao uso de plataformas PaaS para implantação de software..... | 17 |
| 3 METODOLOGIA | 20 |
| 3.1 FERRAMENTAS RELACIONADAS..... | 23 |
| 4 RESULTADOS DA SOLUÇÃO PROPOSTA | 25 |
| Fonte: Autor..... | 34 |
| 5 CONCLUSÃO | 35 |
| REFERÊNCIAS | 37 |

1 INTRODUÇÃO

Com o surgimento e popularização da internet houve aumento de volume de dados, informações e serviços disponibilizados, como também a crescente necessidade de armazenamento de dados e processamento. Este cenário se tornou propício para a computação em nuvem.

A computação em nuvem, possui como principais características a escalabilidade e flexibilidade, e por isso, vem sendo amplamente difundida e utilizada pelas empresas e desenvolvedores para suprir suas demandas de infraestrutura e desenvolvimento de sistemas.

Segundo, Santos(2013, p.10) , a computação em nuvem:

[...] é um tipo de sistema distribuído e paralelo que consiste de uma coleção de computadores virtualizados e interconectados. Esses computadores são provisionados dinamicamente e apresentados como um ou mais recursos computacionais em níveis de serviço estabelecidos através de negociação entre o provedor e seus consumidores.

Nesse contexto, a computação em nuvem conta com diversos modelos de disponibilização de serviços, dentre eles, podemos citar a Plataforma como Serviço (PaaS). Esse modelo de serviço, busca simplificar a implantação de sistemas, uma vez que reduz a responsabilidade de configurações de infraestrutura (rede, armazenamento e entre outros), fazendo com que se ganhe mais tempo e agilidade no desenvolvimento e implantação do sistema.

Dito isto, o referido trabalho propõe realizar uma comparação de custos entre as diversas provedoras de serviços PaaS, a partir de métricas relacionadas aos seguintes recursos computacionais: CPU, memória, rede e armazenamento de banco de dados relacionais. A comparação tem como base atender empresas de pequeno porte e desenvolvedores independentes, impulsionando-os a tomar decisões mais direcionadas.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

O objetivo geral deste TCC consiste no desenvolvimento de um ferramenta que proporcione aos seus usuários uma forma simples de visualizar a comparação de preços de serviços de nuvem do tipo Plataforma como Serviço (PaaS).

1.1.2 Objetivos Específicos

O trabalho tem os seguintes objetivos específicos:

- Levantar as características das plataformas de nuvem modelo PaaS;
- Implementar um mecanismo de comparação de preços a partir de um subconjunto das plataformas elencadas no levantamento supramencionado;
- Disponibilizar o mecanismo de comparação de preços por meio de uma aplicação Web de software livre formada por uma API REST e um cliente Web.

1.2 JUSTIFICATIVA

Como justificativa deste trabalho podemos citar o potencial de auxiliar as pequenas empresas, empresas juniores e desenvolvedores independentes que possuem restrições financeiras na implantação de sistemas. Além disso, a proposta do sistema *web* ser um *software* livre, permite que estudantes e desenvolvedores interessados possam realizar modificações no código-fonte e aprender sobre algoritmos de comparação e computação em nuvem.

Um outro ponto a ser destacado, é o fato de o sistema condensar as informações das plataformas e os custos por serviço, uma vez que existe uma dificuldade inerente de obter informações nas plataformas.

2 FUNDAMENTAÇÃO TEÓRICA

A implantação de um *software*, segundo Sommerville (2011), é a fase final de um processo de desenvolvimento que envolve algumas configurações do sistema para funcionar de forma adequada no ambiente de produção. Essa fase do processo é desafiadora e complexa, uma vez que os desenvolvedores podem se deparar com ambientes diferentes do esperado. Portanto, pode ser preciso realizar ajustes para adaptar o ambiente em que será implantado o *software*.

A implantação é composta por um conjunto de atividades inter-relacionadas, que envolvem configurações e instalações do *software* e até mesmo a *release* do *software* no final do desenvolvimento. Além disso, engloba atividades pós-instalação, tais como monitoramento contínuo, atualização para novas versões, adaptação para atender às modificações no ambiente e, caso seja necessário, a desinstalação do *software* (DEARLE, 2007).

Nesse sentido, no processo de implantação de *software*, a computação em nuvem (NASCIMENTO, 2020) pode ser considerada uma solução mais simples e rápida para entrega e acesso aos recursos de *software* e *hardware*.

A computação em nuvem surgiu como resultado de uma evolução tecnológica, cuja emergência advém do desenvolvimento de sistemas de virtualização e redes de computadores altamente escaláveis (SANTOS, 2013). Dito isto, para entender a computação em nuvem, suas vantagens e serviços, faz-se necessário pontuar alguns momentos históricos.

2.1 HISTÓRICO DA COMPUTAÇÃO EM NUVEM

Um dos fatores que contribuíram para a evolução da computação foi o surgimento de uma arquitetura baseada em grandes computadores: os *Mainframes*. Segundo SANTOS (2013), mainframes são computadores com grande capacidade de processamento de dados, que foram introduzidos na década de 1960. Utilizando mainframes, os serviços de *software*, armazenamento e processamento eram acessados de forma centralizada, ou seja, somente a partir de terminais via redes privadas. Essa forma de computação possuía custos

significativos de aquisição e manutenção, o que dificultou a sua popularização.

O surgimento de computadores pessoais, sistemas operacionais para estes e redes locais, bem como a possibilidade de conectar esses computadores em servidores de rede, na década de 80, emergiu uma nova forma de computação baseada em cliente/servidor. Essa arquitetura possui um modelo de processamento de dados distribuídos em que parte dos dados processados executam no servidor e a outra parte no cliente através da rede (VERAS, 2012).

Com o surgimento da internet, tem-se um avanço na padronização de protocolos de rede para o TCP/IP (VERAS, 2012) e aumento na disponibilidade de banda. Além disso, com a popularização da internet, houve aumento de volume de dados, informações e serviços disponibilizados e processados em computadores remotos. Isso exigiu uma mudança arquitetural que associava os benefícios e eficiência dos mainframes à redução de custos e flexibilidade proporcionadas pelo modelo cliente-servidor (MEDEIROS et al., 2015).

A arquitetura em nuvem transformou a forma de operar sistemas, saindo de um modelo baseado na aquisição de equipamentos para um modelo baseado na aquisição de serviços. Os recursos são disponíveis para *hardware* e *software*, plataformas de desenvolvimento e serviços podendo ser adquiridos de acordo com a necessidade do usuário e, assim, otimizando o uso (VERAS, 2012).

Para Mell e Grance (2011, tradução nossa), a computação em nuvem:

“[...] é um modelo para permitir acesso onipresente, conveniente e sob demanda à rede a um conjunto compartilhado de recursos de computação configuráveis (por exemplo, redes, servidores, armazenamento, aplicativos e serviços) que podem ser rapidamente provisionados e liberados com mínimo esforço de gerenciamento ou interação com o provedor de serviços.”

Os autores ainda apresentam algumas características que consideram essenciais para a computação em nuvem, dentre elas estão:

- *Autoatendimento sob demanda*, a capacidade de fornecer ao consumidor a opção de aumentar e diminuir recursos computacionais, como armazenamento de rede, tempo de servidor, sem a necessidade de interação humana com o provedor de serviços.

- *Ampla acesso a rede*, onde os recursos estão amplamente disponíveis na rede podendo ser acessados por diferentes dispositivos com acesso a internet, como smartphones, tablets, computadores;
- *Elasticidade rápida*, os recursos podem ser aumentados ou diminuídos de acordo com a necessidade ou demanda, sendo que na maioria das vezes são realizados de forma automática;
- *Agrupamento de recursos*, o provedor utiliza um modelo multilocatário para agrupar recursos, com o intuito de atender diversos clientes, onde de acordo com a demanda os recursos físicos e virtuais são atribuídos e reatribuídos de forma dinâmica.
- *Serviço mensurado*, os serviços podem ser controlados e monitorados pelos sistemas em nuvem, possibilitando uma maior transparência tanto para o fornecedor quanto para o consumidor.

As nuvens possuem modelos de implantação sendo divididas em nuvens privadas, comunitárias, públicas e híbridas (MELL; GRANCE, 2011). Nas *nuvens privadas*, a infraestrutura é gerenciada e operada pela própria organização ou terceiros e a utilização dos serviços é exclusiva de uma única organização. Nas *nuvens comunitárias* a infraestrutura é destinada e usada por organizações que possuem interesses ou negócios em comum e pertencem a uma comunidade específica. Neste modelo, o gerenciamento e operação pode ficar sob responsabilidade das organizações pertencentes à comunidade ou por terceiros. Nas *nuvens públicas* a infraestrutura compartilhada é disponibilizada para o uso do público em geral, onde os recursos são fornecidos por meio do modelo “*pay-per-use*”, ou seja, são pagos apenas pelo que é utilizado. O modelo de *nuvens híbridas* é uma combinação de duas ou mais nuvens (pública, privada ou comunitária), que permanecem em entidades separadas, mas permite a portabilidade de aplicações e dados.

2.2 MODELOS DE DISPONIBILIZAÇÃO DE COMPUTAÇÃO EM NUVEM

Os serviços de nuvem podem ser classificados de acordo com o nível de controle do usuário e disponibilidade de poder computacional (ARAÚJO, 2017). Os três principais modelos de serviços são: Infraestrutura como Serviço (IaaS), Plataforma como Serviço (PaaS) e Software como Serviço (SaaS). Nas seções

2.2.1, 2.2.2 e 2.2.3 a seguir, encontram-se brevemente conceituados estes modelos de serviços, incluindo alguns exemplos de soluções encontradas no mercado. O objetivo é melhorar o entendimento do funcionamento de cada um dos modelos.

2.2.1 Infraestrutura como Serviço (IaaS):

Fornece uma infraestrutura de armazenamento e processamento de forma transparente. O usuário é capaz de configurar e gerenciar parte dos recursos, como aplicativos, armazenamento, máquinas virtuais e até mesmo a escolha do sistema operacional, todavia, não é capaz de controlar a camada física. Além disso, os recursos podem ser aumentados ou diminuídos de forma dinâmica de acordo com a necessidade das aplicações (MELL; GRANCE, 2011).

A Amazon Web Services surgiu em 2006, com o objetivo de fornecer serviços de infraestrutura de TI por meio da computação em nuvem (AMAZON, 2023). Dentre os serviços de IaaS oferecidos pela AWS, destaca-se o Amazon Elastic Compute Cloud (Amazon EC2). Este serviço permite que as empresas possam criar e configurar sua infraestrutura de nuvem com uma capacidade de computação segura e redimensionável. Alguns dos recursos oferecidos pelo Amazon EC2 são:

“Ambientes de computação virtual, conhecidos como instâncias; Os modelos pré-configurados para suas instâncias, conhecidos como Imagens de máquina da Amazon (AMIs), que empacotam os bits de que você precisa para seu servidor (incluindo o sistema operacional e software adicional); Várias configurações de capacidade de CPU, memória, armazenamento e redes para suas instâncias, conhecidas como tipos de instância; Um firewall que permite especificar os protocolos, portas e intervalos de IPs de origem que podem acessar suas instâncias usando grupos de segurança.”

A Microsoft Azure oferece uma variedade de serviços em computação em nuvem seguros, disponíveis e escaláveis (AZURE, 2023). Entre os serviços de Infraestrutura como Serviço (IaaS), pode-se destacar o Windows Server. Este é um sistema operacional em nuvem que dispõe de recursos que facilitam o gerenciamento de VMs, possui suporte nativo para SQL Server e aplicativos .NET, como também permite a integração com outros recursos da Azure, como serviço de Kubernetes do Azure (AKS).

2.2.2 Plataforma como Serviço (PaaS):

O PaaS oferece um modelo de computação em nuvem onde o usuário pode controlar toda a parte de implantação e desenvolvimento, incluindo bibliotecas, serviços e um conjunto de ferramentas compatíveis com o provedor, sem se preocupar com configurações e monitoramento da infraestrutura subjacente (rede, armazenamento e entre outros) (MELL; GRANCE, 2011).

O Heroku é uma plataforma como serviço que oferece um ambiente pronto para implantar e executar aplicações (HEROKU, 2023). Foi criada em 2007, com suporte apenas para linguagem de programação Ruby, mas ampliou o suporte para outras linguagens, como Node, Java, PHP, Python, Go, Scala, após ser adquirida pela Salesforce em 2010. Os aplicativos são executados e dimensionados dentro de contêineres Linux que são chamados de *dynos*.

Um outro exemplo de solução PaaS é o Google App Engine. Lançado em 2008 pela Google como uma plataforma de hospedagem de aplicativos em nuvem. O App Engine suporta várias linguagens de programação e oferece um ambiente padrão pré-configurado baseado em instâncias de contêineres que são executados na infraestrutura do Google. Com relação a segurança dos aplicativos, o serviço permite a definição de regras do firewall e utilização de certificados SSL/TLS gerenciados sem custo adicional para os domínios personalizados (GOOGLE CLOUD, 2023). Na seção 2.3, encontra-se um detalhamento das nuvens tipo PaaS, pois estas são de especial interesse para este trabalho.

2.2.3 Software como Serviço (SaaS):

No modelo de nuvem SaaS, as aplicações são fornecidas por meio de um *host* e podem ser acessadas por meio de um navegador *Web* ou aplicativos específicos. Nesse modelo de nuvem, tem-se a vantagem dos *softwares* poderem ser acessados por meio de qualquer dispositivo com internet, bem como toda parte de manutenção e suporte do *software* fica sob responsabilidade do provedor da infraestrutura (MELL; GRANCE, 2011).

Salesforce é uma das principais provedoras de serviços de software como serviço. A empresa foi fundada em 1999, ano em que lançaram um sistema de gerenciamento de relacionamento com o cliente (CRM), oferecendo produtos e

serviços relacionados a vendas, marketing, e-commerce, atendimento ao cliente entre outros, que podem ser acessados por meio de um navegador web (SALESFORCE, 2023).

Outro provedor de serviços do modelo SaaS é o Google Workspace. Anteriormente conhecido como G Suíte, o Google Workspace é um conjunto de aplicativos e serviços integrados baseados em nuvem, que tem como principal objetivo aumentar a produtividade e colaboração de indivíduos e empresas. O Google Workspace oferece uma variedade de ferramentas, tais como Gmail, Google Drive, Google chat, Google Meet, Google Docs e outros (GOOGLE WORKSPACE, 2023).

2.3 INTRODUÇÃO A NUVENS DO TIPO PaaS

Com base em uma comparação inicial entre ferramentas oferecidas no modelo de serviço PaaS, no contexto deste trabalho, é possível resumir características comuns a estas plataformas.

A primeira característica que é possível citar é a redução de configurações: no modelo PaaS, o usuário não precisa se preocupar em manter servidores, configurar *hardware*, sistemas operacionais e redes, nem com atualização de *softwares*. Esses recursos são fornecidos e mantidos pelos provedores (RED HAT, 2023).

As plataformas PaaS oferecem recursos pré-configurados, possibilitando agilidade e redução no tempo de desenvolvimento. Isso permite que os desenvolvedores possam concentrar seus esforços em criar, testar, implantar e gerenciar as aplicações (AZURE, 2023).

Uma outra característica é a capacidade de viabilizar o desenvolvimento de novos recursos e implantação de aplicações com várias linguagens de programação, como Java¹, Node.js², Python³, Go⁴ e outras. Além disso, possuem

¹ Disponível: <<https://www.java.com/>>

² Disponível: <<https://nodejs.org/>>

³ Disponível: <<https://www.python.org/>>

⁴ Disponível: <<https://go.dev/>>

ferramentas de gerenciamento por meio de interface Web (IBM, 2023), onde os desenvolvedores podem colaborar e testar novos recursos de qualquer lugar com conexão à Internet. Outra característica é a integração com outros serviços e tecnologias, conectando os aplicativos, por exemplo, com bancos de dados e serviços de mensagens (ORACLE, 2023).

A escalabilidade é uma importante vantagem das plataformas PaaS. Os recursos de capacidade de armazenamento, processamento e rede podem ser aumentados ou diminuídos de acordo com a demanda. Essa capacidade de dimensionar e escalar recursos, é um dos fatores que contribui para redução de custos, uma vez que paga-se apenas pelo o que é utilizado, evitando o desperdício e uso inadequado de recursos. Um exemplo é que, durante períodos de picos de tráfego, faz-se necessário adicionar mais recursos computacionais, mas esses ficam ociosos durante períodos de baixa demanda e, portanto, podem ser reduzidos (IBM, 2023).

Diante de todas as características, as nuvens tipo PaaS se tornaram bastante populares entre os desenvolvedores e empresas que desejam implantar o seu sistema. Todavia, com a variedade de plataformas que oferecem serviços em nuvem surge a necessidade de comparar custos, no que diz respeito ao *hardware* virtual cobrado a cada uma das plataformas.

2.3.1 Levantamento de custos associados ao uso de plataformas PaaS para implantação de software

Para comparação das plataformas foram utilizadas as seguintes métricas: CPU, memória, rede e armazenamento de banco de dados relacionais. As plataformas escolhidas para a comparação foram: *Heroku, Railway e Render*.

A escolha dessas plataformas foi baseada em suas características semelhantes no suporte ao desenvolvimento, como a compatibilidade com diversas linguagens de programação, capacidade de implantação de aplicações, escalabilidade, gerenciamento de bancos de dados relacionais e facilidade de uso. Além disso, as informações e documentação sobre os recursos e planos de cada plataforma facilitou a coleta de dados e a realização das comparações.

Para implementar um sistema nestas plataformas de *deploy*, o processo geralmente segue algumas etapas comuns a todas ferramentas abordadas neste trabalho:

1. Hospedar o projeto em um repositório de controle de versão, como GitHub.
2. Registrar-se na plataforma escolhida, selecionar o plano que atenda às necessidades da aplicação e iniciar um novo projeto, conectando o repositório à plataforma.
3. Configurar o projeto: isso envolve definir os comandos de *build* e *start*, adicionar as variáveis de ambiente para configurar banco de dados e quaisquer serviços adicionais que o sistema precisa para funcionar corretamente.

Com todas essas configurações ajustadas, o processo de *deploy* é iniciado, durante o qual a plataforma gerencia automaticamente a construção e a implantação do aplicativo. É possível monitorar o status da aplicação para verificar que tudo esteja funcionando como esperado e se necessário realizar ajustes. Por fim, é fornecida uma URL pela plataforma para visualização da aplicação.

Para a análise foram selecionados planos básicos e intermediários que, geralmente, atendem às necessidades de pequenas e médias aplicações em ambientes de produção. Dessa forma, apresentaremos os planos escolhidos para cada plataforma, juntamente com seus respectivos custos.

Na plataforma *Heroku* foi escolhido o plano *Standard 2X*, que disponibiliza 2vCPU e 1GB de memória RAM pelo custo de 50,00 USD. Para o armazenamento de banco de dados é necessário incluir um outro plano, o *Standard 0*, que possui 64GB de armazenamento com o custo de 50,00 USD por mês. Já em relação aos serviços de rede o acesso é ilimitado.

Na plataforma *Railway* foi escolhido o plano *Hobby*. Nesse plano é cobrado uma taxa de assinatura mensal de 5,00 USD mais o adicional de taxas padrão: 1 vCPU por 20,00 USD e 1GB de memória por 10,00 USD. Assim como a *Heroku*, o acesso à rede é ilimitado.

Na plataforma *Render* foi escolhido o plano *Team*, onde tem-se o custo de 19,00 USD por mês mais os custos de computação. Dessa forma, para os serviços

de computação foi escolhido o plano *Standard* com 1 vCPU e 2GB de memória por 25,00 USD por mês. Com relação ao armazenamento é necessário assinar um outro plano também *Standard* com custo de 20,00 USD por mês para 16GB de memória. Já o acesso à rede já é incluso no plano tendo o limite de 500GB.

A Tabela 1 abaixo ilustra os preços médios totais das plataformas por recursos na data de 19 de Junho de 2023.

| Plataforma | vCPU | Memória (RAM) | Rede | Armazenamento | Preço (USD) |
|-------------------|-------------|----------------------|--------------------|----------------------|--------------------|
| Heroku | 2 | 1GB | Acessos ilimitados | 64GB | \$100 |
| Railway | 1 | 1GB | Acessos ilimitados | incluso até 100GB | \$35 |
| Render | 1 | 2GB | 500GB | 16GB | \$64 |

Tabela 1: Comparação entre serviços de computação em nuvem

3 METODOLOGIA

O *Scrum* é um *framework* de gerenciamento de projetos baseado em uma metodologia ágil, com uma abordagem iterativa e incremental, que possibilita que as equipes façam um melhor uso dos recursos, e de tempo de desenvolvimento, para uma entrega contínua de valor ao cliente (SOMMERVILLE, 2011).

O Scrum adere aos princípios do manifesto ágil⁵ para guiar ou orientar o desenvolvimento dos projetos, incorporando atividades metodológicas como requisitos, análise, projeto, evolução e entrega. Cada uma dessas atividades ocorre em períodos de tempo, com duração de 2 a 4 semanas, chamados de Sprints (PRESSMAN e MAXIM, 2010).

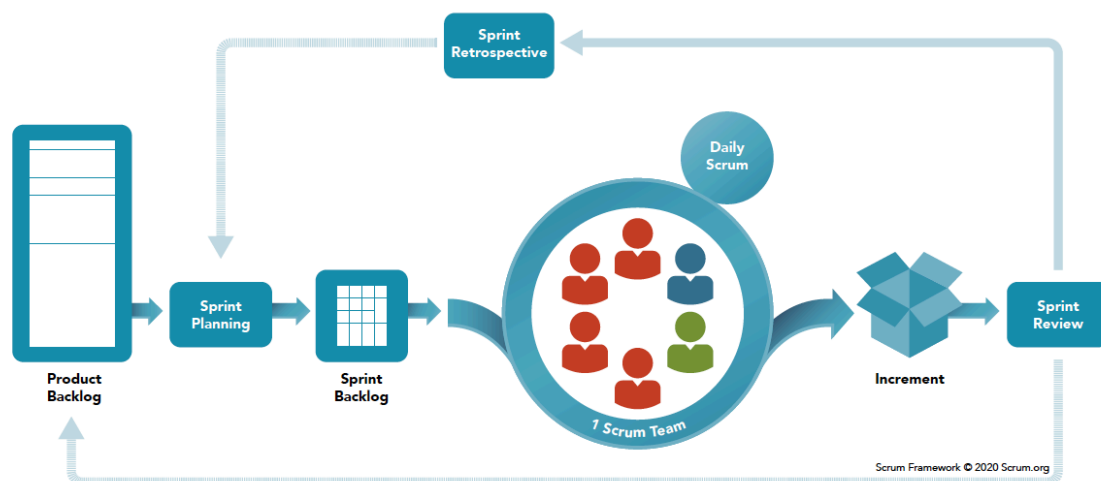
Além das Sprints, o Scrum define os eventos de *Sprint Planning*, onde são definidos e planejados o desenvolvimento para a próxima sprint. O *Daily Scrum*, que é um evento de 15 minutos para possibilitar a equipe sincronizar ideias, identificar impedimentos e se planejar para as atividades das próximas 24hrs. A *Sprint Review*, é uma reunião, que ocorre a cada final de sprint, para demonstrar e revisar o trabalho concluído. E por fim a *Sprint Review*, também realizada ao final de cada sprint, onde toda a equipe Scrum faz uma reflexão sobre o trabalho desenvolvido e para identificar melhorias (PRESSMAN e MAXIM, 2010)..

O Scrum também define time e artefatos. Em relação ao time, essa é composta por uma equipe interdisciplinar, auto organizáveis e multifuncionais, sendo um *Product Owner*, tem o papel de representar os interesses dos clientes e definir e priorizar os itens do *backlog* do produto; um *Scrum Master*, é um facilitador para os membros da equipe e também responsável por identificar e solucionar impedimentos; e três a seis desenvolvedores, sendo os responsáveis pelo desenvolvimento e entrega a cada *Sprint*. Já os artefatos são *backlog* do produto, sendo uma lista de requisitos, funcionalidades que foram priorizadas; o *backlog* da *sprint*, é um conjunto de itens selecionados do backlog do produto para o desenvolvimento e incremento de código para a sprint atual. Na Figura 1 está ilustrado o processo do Scrum (PRESSMAN e MAXIM, 2010)..

⁵ Disponível em: <https://agilemanifesto.org/iso/ptbr/manifesto.html>

Figura 1 - Fluxo do processo do Scrum

SCRUM FRAMEWORK



Fonte: (SCRUM, 2024)

Disto isto, no contexto deste projeto, será utilizada uma metodologia Scrum adaptada para o desenvolvimento do *software*, conhecida como Scrum Solo. Nessa abordagem, alguns conceitos do Scrum são mantidos, mas moldados para atender as necessidades de desenvolvedores que trabalham individualmente. Por exemplo, sprints mais curtas, backlogs das funcionalidades ou tarefas a serem realizadas. Embora não haja reuniões diárias (daily) o orientador, desenvolvedor e grupo de validação podem se reunir conforme necessário para refletir sobre o progresso e entrega de novas funcionalidades. Essa flexibilidade permite uma gestão ágil do projeto, mesmo com um único indivíduo envolvido no desenvolvimento (PAGOTTO, 2016).

Seguiremos então uma abordagem de levantamento de requisitos por meio de *user stories* e o processo de desenvolvimento em sprints. O projeto foi desenvolvido em algumas sprints, sendo necessário retomar funcionalidades já desenvolvidas para realizar correções ou ajustes com base na identificação de

problemas.

| Numeração | Descrição | Critério de Aceitação |
|-----------|--|---|
| S1 | Como usuário gostaria de obter os dados de RAM, vCPU's, armazenamento de banco e rede da plataforma Render. | Os dados sobre planos e custos devem ser fornecidos na estrutura de dados JSON. |
| S2 | Como usuário gostaria de obter os dados de RAM, vCPU's, armazenamento de banco e rede da plataforma Railway. | Os dados sobre planos e custos devem ser fornecidos na estrutura de dados JSON. |
| S3 | Como usuário gostaria de obter os dados de RAM, vCPU's, armazenamento de banco e rede da plataforma Heroku. | Os dados sobre planos e custos devem ser fornecidos na estrutura de dados JSON. |
| S4 | Como usuário, gostaria que houvesse integração dos dados por meio de uma API. | A API deve fornecer endpoints de busca de dados de de RAM, vCPU's, Armazenamento de banco e rede das plataformas. |
| S5 | Como usuário, gostaria que fosse corrigido a exibição dos dados. | Os dados capturados das plataformas devem ser retornados de forma correta. |

| | | |
|----|---|--|
| S6 | Como usuário gostaria de uma interface que permita comparar os recursos desejados, como RAM, vCPU's, armazenamento de banco de dados e capacidade de rede. | Deverá aparecer na página de resultados listagem o nome da provedora de serviço, memória RAM, vCPU, armazenamento e preço. |
| S7 | Como usuário gostaria que fosse realizado um ajuste no interface que compara os recursos desejados, como RAM, vCPU's, armazenamento de banco de dados e capacidade de rede. | Após informar os dados, o resultado deve ser exibido por plataforma e preço de forma crescente. |

Tabela 2: Descrição das User Stories

As atividades desenvolvidas para o desenvolvimento do projeto podem ser visualizadas na Tabela 3, a seguir.

| Atividades | Mar. | Abr. | Mai. | Jun. | Jul. | Ago. | Set. | Out. |
|----------------------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| Planejamento TCC II | X | X | | | | | | |
| Revisão bibliográfica | X | X | X | X | X | X | | |
| Implementação do <i>Software</i> | | X | X | X | X | X | X | |
| Revisão Final do Trabalho | | | | | | | | X |
| Depósito do TCC II | | | | | | | | X |
| Defesa do TCC II | | | | | | | | X |

Tabela 3: Cronograma

3.1 FERRAMENTAS RELACIONADAS

À medida que a computação em nuvem se torna cada vez mais predominante

no cenário tecnológico atual, as ferramentas que auxiliem na comparação de preços de servidores e recursos em nuvem se tornam cruciais para que empresas e indivíduos possam escolher a melhor opção dentro de seus orçamentos e necessidades. Dentre as ferramentas que podemos citar, as mais conhecidas hoje no mercado são: Amazon Web Services (AWS)⁶, Microsoft Azure⁷, Google Cloud Platform⁸. Estas fornecem calculadoras especializadas que permitem aos usuários estimar os custos de serviços, como instâncias de computação, armazenamento de dados e transferência de dados. Embora possam abranger diferentes categorias de serviços, como IaaS, PaaS e SaaS, essas calculadoras são projetadas para os planos específicos oferecidos por cada provedor.

Outra ferramenta é o Clouorado⁹, que oferece uma plataforma para comparar preços em nuvem de uma variedade de provedores. Com uma interface intuitiva, a plataforma apresenta filtros de busca por tipo de serviços, região, rede, segurança, banco de dados, sistema operacional e outros. Além disso, disponibiliza uma seção com informações detalhadas sobre os provedores de serviços de computação em nuvem, facilitando a compreensão das ofertas de cada um. Até o momento do desenvolvimento deste trabalho, o Clouorado não oferece comparação para provedores de PaaS (Plataforma como Serviço).

É importante ressaltar que as ferramentas acima mencionadas comparam preços de recursos e serviços de mais de uma categoria de serviços (IaaS, PaaS e SaaS). No entanto, não foi encontrada uma ferramenta específica, até o momento da pesquisa, que se concentrasse em comparar apenas Plataformas como Serviço, conforme proposto neste trabalho.

⁶ https://calculator.aws/#/?nc2=h_ql_pr_calc

⁷ <https://azure.microsoft.com/pt-br/pricing/calculator>

⁸ https://cloud.google.com/products/calculator?hl=pt_br

⁹ <https://www.clouorado.com/>

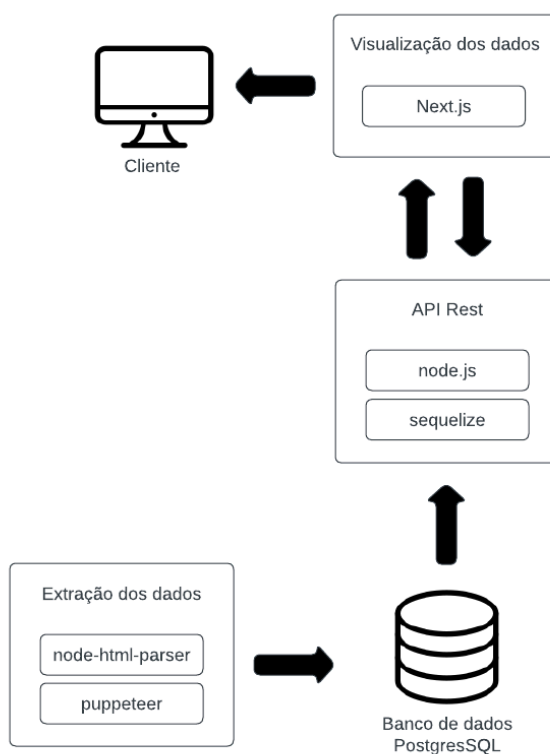
4 RESULTADOS DA SOLUÇÃO PROPOSTA

Neste capítulo, abordaremos as etapas do desenvolvimento e implementação da ferramenta de comparação de preços de serviços de cloud do tipo PaaS (Platform as a Service), proposta neste trabalho de conclusão de curso.

Como prova de conceito desenvolvemos uma aplicação web em forma de assistente, permitindo aos usuários especificar os recursos desejados, tais como RAM, vCPU's, armazenamento de banco de dados e capacidade de rede. Essa funcionalidade retorna um resultado que mais se aproxima das necessidades do usuário.

Para a aplicação web que denominamos de “*Compare Cloud*”, foi pensado uma arquitetura que separasse as responsabilidades dos componentes, buscando facilitar a manutenção e adição de novas funcionalidades sem afetar a aplicação como um todo.

Figura 2 - Arquitetura da aplicação

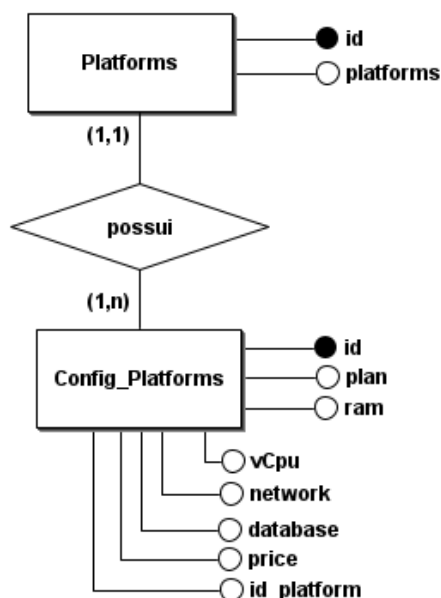


Fonte: Autor

No módulo de banco de dados, optamos por utilizar o PostgreSQL¹⁰, por ser um sistema de gerenciamento de banco de dados relacional de código aberto, ideal para processar e armazenar grandes volumes de dados e suporta consultas rápidas e otimizadas.

Para melhor organizar e estruturar os dados da aplicação, decidimos dividi-los em duas tabelas distintas. Na primeira tabela, armazenamos os dados das plataformas: identificador único (id) e nome da plataforma. Na segunda tabela, armazenamos os dados específicos das configurações das plataformas, sendo eles: os planos disponíveis, as informações de RAM, vCPUs, capacidade de rede, armazenamento de banco de dados, preço e o identificador da plataforma correspondente. Essa estrutura permite que cada registro contenha informações detalhadas sobre as opções de planos oferecidos por cada provedor, facilitando as comparações.

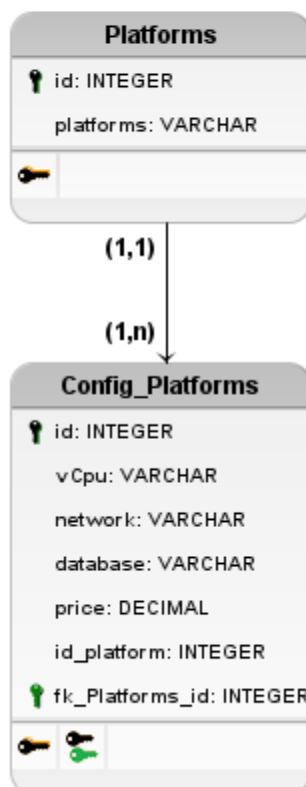
Figura 3 - Modelo Conceitual



Fonte: Autor

¹⁰ <https://www.postgresql.org/>

Figura 3 - Modelo Lógico



Fonte: Autor

Conforme foi mencionado neste projeto, as comparações de dados foram realizadas utilizando os dados das plataformas: *Render*, *Railway* e *Heroku*. Para essa coleta e extração, utilizamos o *Web Scraping*, que nos permite obter dados diretamente das páginas web das plataformas, analisando o conteúdo para identificar e reunir as informações relevantes e organizá-las conforme as necessidades. Essa técnica permite automatizar a extração de dados e também assegura que os dados coletados estejam sempre atualizados (TAVARES; CUNHA, 2021).

Utilizamos duas bibliotecas de *Web Scraping*: o *node-html-parse* e *puppeteer*. Com o *node-html-parse*¹¹ coletamos os dados a partir do HTML das páginas web. Já o *puppeteer*¹², foi utilizado para realizar interações dinâmicas com a página, como clicar em botões.

Após a coleta e o tratamento dos dados, estes serão armazenados em um

¹¹ <https://www.npmjs.com/package/node-html-parser>

¹² <https://pptr.dev/>

banco de dados, onde foi escolhido o SGBD PostgreSQL, conforme mencionado anteriormente. Dessa forma, utilizamos o Sequelize, uma biblioteca ORM (Object-Relational Mapping) para o Node.js, que facilita a modelagem dos dados e as operações de inserção, busca e outros. A modelagem de dados com o *Sequelize* permite a definição clara das tabelas, seus respectivos campos e relações entre as entidades, assegurando a integridade e a consistência dos dados.

Para o processo de inserção, implementamos agendamentos (schedules) que realizarão a coleta dos dados dos sites a cada 30 dias, garantindo que as informações no banco de dados estejam sempre atualizadas. A inserção de dados é efetuada através do método “*bulkCreate*” do *Sequelize*, que permite a inserção de múltiplos registros, assim, otimizando o processo.

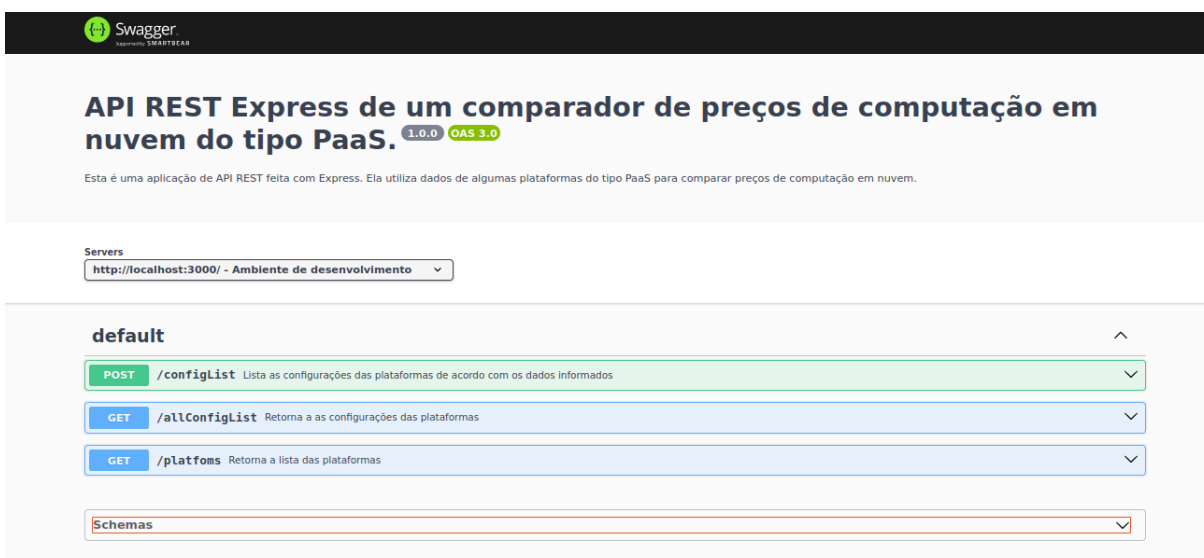
Algoritmo 1 - Código de inserção de dados

```
1  async function insertPlatform() {
2    try {
3      const namePlatforms = await populateTables();
4      const conditions = [];
5      for (const platform in namePlatforms) {
6        if (namePlatforms[platform] != null) {
7          conditions.push({
8            platforms: namePlatforms[platform].platformName,
9          });
10     }
11     const existingRecords = await Platforms.findAll({
12       where: {
13         [Sequelize.Op.or]: conditions,
14       },
15     });
16     if (existingRecords.length === 0) {
17       const platformsCreate = conditions.map(condition => ({
18         platforms: condition.platforms,
19       }));
20       await Platforms.bulkCreate(platformsCreate);
21       console.log('Múltiplas inserções realizadas com sucesso.');
```

Fonte: Autor

Para busca destes dados, implementamos controllers que possuem uma lógica de recuperação aproximada dos dados que serão informados. Estes *controllers* são acessíveis por meio das rotas bem definidas, que foram documentadas utilizando a ferramenta *Swagger*¹³ para facilitar o entendimento e utilização da API.

Figura 5 - Swagger



Fonte: Autor

Vale ressaltar que as rotas `"/allConfigList"` e `"/plafoms"` servem apenas para teste ou para fornecer uma visão geral dos dados disponíveis. Por outro lado, a rota `"/configList"` é a principal utilizada no projeto, sendo responsável por retornar todas as informações detalhadas relacionadas às plataformas, conforme os dados fornecidos.

¹³ <https://swagger.io/>

Figura 6 - Rota configList

POST /configList Lista as configurações das plataformas de acordo com os dados informados

Parameters Try it out

No parameters

Request body application/json

Example Value | Schema

```
{
  "ram": "4 GB ou 4MB",
  "cpu": "4",
  "network": "500",
  "database": "4 GB ou 4MB"
}
```

Responses

| Code | Description | Links |
|------|----------------------------|----------|
| 200 | Item retornado com sucesso | No links |
| 500 | Internal Server Error | No links |

Media type: application/json

Controls Accept header.

Example Value | Schema

```
{
  "id": 1,
  "ram": "4 GB ou 4MB",
  "cpu": "4",
  "network": "500",
  "database": "4 GB ou 4MB",
  "id_pltform": 2,
  "createdAt": "2024-01-15T01:55:12.672Z",
  "updatedAt": "2024-01-15T01:55:12.672Z",
  "Platform": {
    "platforms": "Render"
  }
}
```

Fonte: Autor

Os dados serão apresentados ao usuário de forma interativa e intuitiva por meio do frontend. Este módulo foi desenvolvido utilizando o framework *Next.js*¹⁴, que é baseado em *React.js*, facilita a criação de aplicações web dinâmicas e oferece funcionalidades de renderização do lado do servidor.

Dito isto, a aplicação web foi desenvolvida em forma de assistente, onde o usuário informa a quantidade de recursos de RAM, vCPU, velocidade de rede e armazenamento de dados e por fim é devolvido o resultado aproximado.

A primeira tela implementada é a **Home**, nela o usuário terá uma breve descrição sobre a aplicação. Ao clicar no botão "Iniciar Assistente" será direcionado para a próxima tela.

¹⁴ <https://nextjs.org/>

Figura 7 - Home

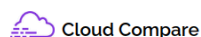


Fonte: Autor

Após ser redirecionado o usuário deverá fornecer informações específicas relacionadas aos recursos que deseja. Essas informações incluem:

- **RAM:** A quantidade de memória RAM em gigabytes (GB) ou megabytes (MB).

Figura 8 - Informar quantidade de RAM



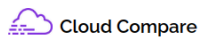
1. Olá! Forneça a quantidade de RAM desejada:

Next

Fonte: Autor

- **vCPU:** o número de unidades de processamento virtual.

Figura 9 - Informar quantidade de vCpu



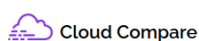
2. Ótimo! Agora, informe a quantidade desejada para a vCPU:

Next

Fonte: Autor

- **Velocidade de rede:** capacidade de transferência de dados.

Figura 10 - Informar velocidade de rede



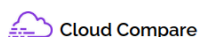
3. Informe a velocidade de rede desejada:

Next

Fonte: Autor

- **Armazenamento de Banco de Dados:** capacidade disponível para armazenar dados em um banco de dados, medida em gigabytes (GB) ou megabytes (MB).

Figura 10 - Armazenamento banco de dados



4. Por último, informe o tamanho desejado para a base de dados:

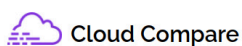
Digite aqui... Ex: 10 GB OU 10 MB

Next

Fonte: Autor

Na página de **Recomendações** é possível visualizar a comparação de preços aproximada aos dados que foram informados pelo usuário em forma de tabela, ordenados por plataforma, plano e preço de forma crescente.

Figura 11 - Recomendações - 1



Recomendações

Render

| Plan | Ram | CPU | Network | Database | Price |
|------|--------|-----|---------|----------|--------|
| Team | 512 MB | 0.5 | 500 | 1 GB | 33.00 |
| Team | 512 MB | 0.5 | 500 | 16 GB | 46.00 |
| Team | 512 MB | 0.5 | 500 | 96 GB | 121.00 |
| Team | 512 MB | 0.5 | 500 | 256 GB | 211.00 |
| Team | 2 GB | 1 | 500 | 1 GB | 51.00 |
| Team | 2 GB | 1 | 500 | 16 GB | 64.00 |
| Team | 2 GB | 1 | 500 | 96 GB | 139.00 |

Fonte: Autor

*Figura 12 - Recomendações - 2***Heroku**

| Plan | Ram | CPU | Network | Database | Price |
|------------------|--------|------|-----------------|----------|---------|
| Standard 1X | 512 MB | 1 | Acesso Limitado | 768 GB | 775.00 |
| Standard 2X | 1 GB | 2 | Acesso Limitado | 768 GB | 800.00 |
| Performance M | 2.5 GB | 100% | Acesso Limitado | 768 GB | 1450.00 |
| Performance L | 14 GB | 100% | Acesso Limitado | 768 GB | 1700.00 |
| Performance LRAM | 30 GB | 100% | Acesso Limitado | 768 GB | 1700.00 |

Railway

| Plan | Ram | CPU | Network | Database | Price |
|-------|-------|-----|---------|----------|-------|
| Hobby | 8 GB | 8 | 500 | 100 GB | 55.00 |
| Pro | 32 GB | 32 | 500 | 100 GB | 70.00 |

Fonte: Autor

5 CONCLUSÃO

Com o aumento exponencial do volume de dados e a crescente demanda por processamento e armazenamento eficientes, a computação em nuvem tornou-se uma ferramenta indispensável no cenário tecnológico atual. Dentro desse contexto, ela oferece uma variedade de modelos de disponibilização de serviços, especialmente o modelo de Plataforma como Serviço (PaaS).

A partir disso, o presente trabalho propôs o desenvolvimento de uma ferramenta de comparação de preços que consolida informações sobre Plataformas como Serviço (PaaS). Essa ferramenta avalia os custos de RAM, vCPU, rede e armazenamento de dados de acordo com os planos oferecidos pelas plataformas, reduzindo a complexidade de acesso a essas informações.

O intuito da ferramenta é ajudar pequenas empresas, empresas juniores e desenvolvedores independentes a escolher a plataforma mais adequada às suas necessidades e capacidades financeiras, facilitando a tomada de decisões na implantação de sistemas. Além disso, a disponibilização da ferramenta de comparação como um software¹⁵ livre traz um valor educacional significativo, permitindo que estudantes e desenvolvedores explorem e modifiquem o código-fonte para melhor compreensão dos algoritmos de comparação.

É importante destacar os desafios enfrentados durante o desenvolvimento do projeto. Ao utilizarmos Web Scraping, enfrentamos problemas devido a atualizações nas páginas da web, o que fazia com que nosso sistema deixasse de funcionar. Em vários momentos, tivemos que retornar ao código para realizar ajustes e corrigir esses problemas.

Como trabalho futuro, planeja-se expandir a diversidade de plataformas do tipo PaaS, além de integrar os modelos de serviços IaaS e SaaS da computação em nuvem. Essa ampliação permitirá uma comparação mais abrangente, oferecendo uma visão completa das opções disponíveis e facilitando a escolha da solução mais adequada para diferentes necessidades e cenários.

¹⁵ <https://github.com/camilacasimiro/tcc-compare-cloud>

Além da expansão das plataformas, almeja-se implementar um conversor de moedas para o resultados das comparações dos preços. Essa funcionalidade permitirá que os usuários possam visualizar os valores das comparações de custos em diferentes moedas.

REFERÊNCIAS

AMAZON WEB SERVICES. **O que é o Amazon EC2?**. Disponível em: <https://docs.aws.amazon.com/pt_br/AWSEC2/latest/UserGuide/concepts.html>. Acesso em: 29 de abril de 2023.

ARAÚJO, Caio César Lopes. **COMPUTAÇÃO EM NUVEM: um estudo sobre a distribuição da produção de artigos publicados no período de 2007 a 2016**. Monografia (Bacharelado em Administração) – Universidade de Brasília. Brasília, 2017.

AZURE. **O que é PaaS?** Disponível em: <https://azure.microsoft.com/pt-br/resources/cloud-computing-dictionary/what-is-paas>. Acesso em: 06 de maio de 2023.

DEARLE, Alan. **Software Deployment, Past, Present and Future**. *Futuro da Engenharia de Software* (FOSE '07), MN, EUA, 2007, pp. 269-284. Disponível em: <https://ieeexplore-ieee-org.ez291.periodicos.capes.gov.br/document/4221626>. Acesso em: 13 de maio de 2023.

GOOGLE CLOUD. **App Engine**. 2023. Disponível em: <https://cloud.google.com/appengine>. Acesso em 13 de maio de 2023.

GOOGLE WORKSPACE. 2023. Disponível em: <https://workspace.google.com/?hl=pt-br>. Acesso em: 14 de maio de 2023.

HEROKU. **A Plataforma Heroku**. 2023. Disponível em: <https://www.heroku.com/platform>. Acesso em: 30 de abril de 2023.

IBM. **What is Platform-as-a-Service (PaaS)?**. Disponível em: <https://www.ibm.com/topics/paas>. Acesso em: 06 de maio de 2023.

MEDEIROS, B. C.; NETO, M. V. S.; DANJOUR, M. F. **Computação em Nuvem: uma análise bibliométrica dos estudos publicados em eventos e periódicos no Brasil**. Botucatu, SP, Tekne e Logos, v. 6, n. 1, p. 60-76, jun., 2015.

MELL, Peter; GRANCE, Timothy. **The NIST Definition of Cloud Computing**. September, 2011. Disponível em: <http://faculty.winthrop.edu/domanm/csci411/Handouts/NIST.pdf>. Acesso em: 20 de abril de 2023.

NASCIMENTO, Gustavo Henrique Alvim. **COMPUTAÇÃO EM NUVEM COM A PLATAFORMA MICROSOFT AZURE**. 2020. 46 f. Monografia (Bacharel) - Curso de Engenharia de Computação, Pontifícia Universidade Católica de Goiás, Goiânia, 2022. Disponível em: <https://repositorio.pucgoias.edu.br/jspui/bitstream/123456789/1070/1/TCC%20-%20>

Gustavo%20Henrique%20%28vers%C3%A3o%20final%29.pdf. Acesso em: 20 abr. 2023.

ORACLE. **Por que o PaaS é importante?**. Disponível em: <https://www.oracle.com/br/cloud/what-is-paas/benefits-of-paas/#role-of-database>. Acesso em : 13 de maio de 2023.

PAGOTTO, Tiago. et al. **Scrum Solo: Processo de software para desenvolvimento individual**. 2016. Disponível em: <https://enghariasoftware.wordpress.com/wp-content/uploads/2016/04/scrum-solo.pdf>. Acesso em: 01 de maio de 2024.

PRESSMAN, Roger S.; MAXIM, Bruce R. **Engenharia de Software: Uma Abordagem Profissional**. 9. ed. São Paulo: McGraw-Hill, 2020.


RED HAT. **PaaS | Plataforma como serviço**. Disponível em: <https://www.redhat.com/pt-br/topics/cloud-computing/what-is-paas>. Acesso em: 06 de maio de 2023.

SALESFORCE. **O que é a Salesforce?**. 2023 Disponível em : <https://www.salesforce.com/br/products/what-is-salesforce>. Acesso em: 30 de abril de 2023.

SANTOS, Fabio André Garaluz dos. **Fundamentos da Computação em Nuvem com Abordagem das Plataformas Eucalyptus e OpenNebula**. 44 f. Monografia (Tecnólogo) - Tecnologia em Sistemas para Internet. Universidade Tecnológica Federal do Paraná. Campo Mourão, 2013. Disponível em : https://repositorio.utfpr.edu.br/jspui/bitstream/1/6458/2/CM_COINT_2013_1_02.pdf. Acesso em: 20 de abril de 2023.

SOMMERVILLE, Ian. **Engenharia de Software**. 9ª edição. São Paulo: Pearson Prentice Hall, 2011. Tradução de Ivan Bosnic e Kalinka G. de O. Gonçalves, revisão técnica de Kechi Hiramã.

SCRUM. **The Scrum Framework Poster**. Disponível em: <https://www.scrum.org/resources/scrum-framework-poster>. Acesso em: 8 de Junho de 2024.

| | |
|---|---|
|  | INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA |
| | Campus Cajazeiras - Código INEP: 25008978 |
| | Rua José Antônio da Silva, 300, Jardim Oásis, CEP 58.900-000, Cajazeiras (PB) |
| | CNPJ: 10.783.898/0005-07 - Telefone: (83) 3532-4100 |

Documento Digitalizado Ostensivo (Público)

TCC

| | |
|-----------------------------|---------------------|
| Assunto: | TCC |
| Assinado por: | Camila Casimiro |
| Tipo do Documento: | Tese |
| Situação: | Finalizado |
| Nível de Acesso: | Ostensivo (Público) |
| Tipo do Conferência: | Cópia Simples |

Documento assinado eletronicamente por:

- Camila Casimiro, ALUNO (201822010008) DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - CAJAZEIRAS, em 22/10/2024 07:00:33.

Este documento foi armazenado no SUAP em 22/10/2024. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 1286555

Código de Autenticação: 45971d6fda

