



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CAMPUS TRINDADE
DEPARTAMENTO DE CIÊNCIAS DA INFORMAÇÃO
CURSO CIÊNCIAS DA INFORMAÇÃO

JOSÉ TADEU ALVES LEAL DE OLIVEIRA

Zebblack: Agente Inteligente de Verificação de Informações e Notícias em Tempo
Real

FLORIANÓPOLIS
2024

JOSÉ TADEU ALVES LEAL DE OLIVEIRA

Zeblack: Agente Inteligente de Verificação de Informações e Notícias em Tempo
Real

Trabalho de Conclusão de Curso submetido ao curso de CIÊNCIAS DA INFORMAÇÃO do Centro ou Campus TRINDADE da Universidade Federal de Santa Catarina como requisito parcial para a obtenção do título de BACHAREL em CIÊNCIA DA INFORMAÇÃO.

Orientador(a): Prof. Douglas Dyllon Jeronimo de Macedo

FLORIANÓPOLIS

2024

Dedico este trabalho de conclusão de curso ao criador, José tadeu alves leal de oliveira.

AGRADECIMENTOS

Em primeiro lugar, agradeço aos meus pais, Eduardo Leal de oliveira e Nadir alves leal de oliveira, pilares fundamentais da minha construção social. Eles não mediram esforços financeiros ou educacionais para que seus filhos pudessem realizar seus objetivos, impulsionando-os rumo a um futuro promissor.

Agradeço ao meu orientador Douglas Dyllon Jeronimo de Macedo pela parceria e compreensão e disponibilidade onde se fez muito necessário no semestre da escrita deste tcc, um ótimo professor na sala de aula quanto no meio profissional, se tornando uma referência na minha trajetória acadêmica e profissional. Finalizando meus agradecimentos, gostaria de mandar um abraço especial e caloroso a todos os professores que fizeram parte da minha formação como cientista da informação.

José Tadeu Alves Leal de oliveira

Zeblack: Agente Inteligente de Verificação de Informações e Notícias em Tempo
Real

O presente trabalho em nível de graduação foi avaliado e aprovado por banca
examinadora composta pelos seguintes membros:

Moisés Lima Dutra, Dr. (CIN/UFSC) - Titular Interno
Universidade Federal de Santa Catarina (UFSC)

Alex Sandro Roschildt Pinto, Dr. (INE/UFSC) - Titular Externo
Universidade Federal de Santa Catarina (UFSC)

Laércio Pioli Júnior, Dr. – Suplente
Universidade Federal de Santa Catarina (UFSC)

Certificamos que esta é a versão original e final do trabalho de conclusão que foi
julgado adequado para obtenção do título de Bacharel em Ciências da Informação.

Prof. Douglas Dyllon Jeronimo de Macedo, Dr.

orientador

FLORIANÓPOLIS
2024

"A inteligência artificial sempre esteve mais preocupada em expandir as capacidades da ciência do que em definir seus limites."

(LUGER, 2004, p. 24)

RESUMO

A circulação de fake news se tornou um grande problema nos dias de hoje, principalmente com o uso das redes sociais e aplicativos de mensagens, que espalham informações falsas em alta velocidade. Esse cenário afeta a confiança das pessoas, gera confusão e tem impacto direto em áreas importantes como política, economia e decisões diárias. O trabalho busca criar um serviço de Bots-as-a-Service (BaaS) para possibilitar a verificação rápida e prática de informações, contribuindo para combater a desinformação. A pesquisa utiliza uma abordagem bibliográfica, explorando ferramentas e tecnologias de inteligência artificial que podem ser aplicadas na checagem de notícias. A ideia é integrar essas soluções com aplicativos populares para facilitar o acesso de diferentes tipos de usuários. Como resultado, o projeto pretende apresentar um sistema que seja seguro, rápido e fácil de usar. A proposta inclui testar o protótipo, coletar feedback e documentar o processo, de forma que outros possam se beneficiar das descobertas e aprimorar a solução. A conclusão destaca que um serviço como o BaaS tem grande potencial para combater as fake news, devolvendo confiança para quem consome informações online. Além disso, ajuda a construir uma sociedade mais consciente e preparada para lidar com a desinformação de forma prática e eficaz.

Palavras-chave: Fake news. Verificação. Bots-as-a-Service. Inteligência artificial. Confiança, Zeblack, Agente inteligente.

ABSTRACT

The circulation of fake news has become a major problem these days, especially with the use of social media and messaging apps, which spread false information at high speed. This scenario affects people's trust, generates confusion and has a direct impact on important areas such as politics, economy and daily decisions. The work seeks to create a Bots-as-a-Service (BaaS) service to help in the quick and practical verification of information, contributing to combating disinformation. The research uses a bibliographical approach, exploring artificial intelligence tools and technologies that can be applied to news checking. The idea is to integrate these solutions with popular apps to facilitate access for different types of users. As a result, the project aims to present a system that is safe, fast and easy to use. The proposal includes testing the prototype, collecting feedback and documenting the process, so that others can benefit from the findings and improve the solution. The conclusion highlights that a service like BaaS has great potential to combat fake news, restoring trust to those who consume information online. Furthermore, it helps to build a more aware society that is prepared to deal with disinformation in a practical and effective way.

Keywords: Fake news. Verification. Bots-as-a-Service. Artificial intelligence. Trust, Zeblack, Intelligent Agent.

SUMÁRIO

1. Introdução.....	10
1.1 Problema de Pesquisa.....	10
1.2 Objetivos.....	11
1.2.1 Objetivo Geral.....	11
1.2.2 Objetivos Específicos.....	11
1.3 Justificativa.....	11
1.4 Contribuições.....	12
1.5 Delimitação do Escopo.....	13
1.6 Estrutura do Documento.....	13
2. Fundamentação Teórica.....	14
2.1 Como as Fake News se Espalham e Afetam a Sociedade, a Política e a Economia	14
2.2 Ferramentas e Técnicas Usadas para Verificar Informações.....	16
2.3 Bots como Serviço (BaaS): Como Funcionam e Como Podem contribuir.....	18
2.4 Perfil dos Usuários-Alvo e Cenários de Uso do Zeblack.....	20
2.5 Os Desafios de Criar um Sistema para Verificação Automática de Notícias.....	22
2.6 Questões Éticas e de Privacidade ao Usar Bots para Verificar Informações.....	24
3. Metodologia.....	27
4. Proposta.....	28
5. Implementação e Avaliações.....	30
REFERÊNCIAS.....	92

1. Introdução

Hoje, um dos maiores problemas existentes é o aumento do compartilhamento de informações falsas, popularmente conhecidas como fake news. As redes sociais em combinação com aplicativos mensageiros instantâneos aumentam significativamente a velocidade de disseminação de tais notícias, informações incorretas são lidas por milhões em questão de minutos. Fake news afeta todos os aspectos da sociedade: eles minam a confiança, causam confusão e, muitas vezes, se tornam a causa de várias consequências que afetam a vida de todos os dias: decisões importantes na sociedade, economia vital ou política, e assim por diante (Pilar, 2023).

Nesse contexto, começam a aparecer ferramentas tecnológicas dedicadas à análise e verificação de informação destinadas a combater a desinformação e devolver a confiança às pessoas no conteúdo consumido. Bots-as-a-Service, ou seja, bots como agentes com super poderes, é uma abordagem prática para solucionar este problema. Tratam-se de bots que funcionam como serviços automatizados e acessíveis a pedido, pelo que se torna possível para os participantes nas plataformas de comunicação, que verifiquem diretamente a autenticidade das notícias. Esta abordagem torna o caso interessante e relevante para uma parte significativa do público (Bastos; De Lima, 2023).

1.1 Problema de Pesquisa

Considerando que poucas ferramentas de fácil acesso e utilização estão disponíveis para verificar se uma informação é fake ou não, é necessário criar um serviço simples e prático que faz a checagem de informações. Apesar de criar diversos desafios, como um serviço acessível a diferentes públicos, uma integração com plataformas de comunicação populares e capaz de fornecer respostas assertivas e confiáveis. Com isto, propõe-se as seguintes perguntas de pesquisa:

- Como pode-se criar uma ferramenta para verificar informações de uma forma rápida e confiável?

- Quais tecnologias utiliza-se para integrá-las a fim de oferecer um serviço eficaz e acessível?
- De que maneira um sistema pode ser escalável e ser firme e seguro?
- Como este serviço pode ser adaptado se utilizando a vários públicos e contextos?

1.2 Objetivos

1.2.1 Objetivo Geral

Desenvolver um serviço de Bots-as-a-Service (BaaS) agente inteligente que permita consulta e verificação de informações e notícias em tempo real, criando uma solução amigável e alinhada às atuais necessidades de combate à desinformação do público-alvo.

1.2.2 Objetivos Específicos

1. Identificar as ferramentas e tecnologias necessárias para criar um sistema capaz de se integrar aos aplicativos de mensagem mais populares.
2. Estruturar um Protótipo seguro, escalável e com ferramentas de verificação da confiabilidade das informações acessadas.
3. Conduzir testes, obter feedback dos futuros usuários e documentar a pesquisa como forma de disseminar o conhecimento gerado.

1.3 Justificativa

Os Agentes inteligentes surgem como uma solução simples e acessível. Esses bots, que funcionam como serviços automatizados sob demanda, ajudam os usuários a verificar rapidamente a autenticidade das informações que circulam nas redes sociais e em aplicativos de mensagens. A grande vantagem dessa abordagem é sua facilidade de uso, permitindo que qualquer pessoa, mesmo sem experiência técnica, consiga utilizar a ferramenta de maneira prática e eficaz (Bykov, *et al* 2023).

No entanto, criar um sistema de verificação automatizada não é uma tarefa simples. Um dos maiores desafios é garantir que o sistema seja eficaz na detecção

de fake news, o que requer o uso de algoritmos poderosos e uma análise cuidadosa das informações. Também é importante que o serviço consiga atender uma grande quantidade de usuários sem perder a qualidade e a rapidez na verificação. Outro fato importante para tornar o processo de checagem de notícias ainda mais acessível e útil para o público em geral (Oliveira, 2019).

A criação de um serviço como o BaaS faz sentido também por seu potencial de restaurar a confiança das pessoas nas informações que consomem. Em um mundo onde as fake news podem afetar desde a opinião pública até decisões políticas, oferecer uma ferramenta de verificação confiável e fácil de usar é fundamental para construir uma sociedade mais bem-informada. Automatizar esse processo, com o uso de tecnologias que conseguem analisar grandes volumes de dados rapidamente, é uma maneira eficaz de combater a desinformação (Lorencetti; Salton, 2022).

A implementação desse tipo de serviço representa uma contribuição importante para a área de Ciência da Informação, ao aplicar tecnologias avançadas para resolver problemas reais e urgentes. Integrar um serviço de verificação em plataformas amplamente usadas, e torná-lo acessível, são fatores que aumentam a relevância da solução e ajudam a formar uma sociedade mais crítica e menos suscetível à desinformação (Bastos; De Lima, 2023).

1.4 Contribuições

Do ponto de vista científico, a criação do BaaS possibilita ampliar o conhecimento sobre como a tecnologia pode ser usada no combate às fake news. O projeto traz contribuições importantes para áreas como inteligência artificial e ciência da informação, mostrando como algoritmos e análise de dados podem ser aplicados para resolver problemas reais. Além disso, ele abre caminho para que outros estudos explorem maneiras ainda mais eficazes de melhorar a verificação de informações.

Na parte social, a ideia é oferecer uma ferramenta simples e prática, que permita pessoas a confiarem nas informações que consomem. Ao tornar o processo de verificação acessível para um público amplo, o serviço pode ajudar a reduzir os impactos das notícias falsas no dia a dia, como em decisões importantes ou na

formação de opiniões. É uma forma de contribuir para uma sociedade mais crítica e menos vulnerável à desinformação.

No lado tecnológico, o desenvolvimento do BaaS mostra como inovações como aprendizado de máquina podem ser usadas para resolver problemas do cotidiano. A proposta ainda destaca a importância de criar sistemas que funcionem bem em plataformas populares, como WhatsApp e Telegram, garantindo que a solução seja prática e alcance o maior número de pessoas possível. Essa integração com tecnologias já existentes também pode servir de modelo para outros projetos no futuro.

1.5 Delimitação do Escopo

O estudo se concentra em desenvolver um serviço automatizado de verificação de informações, pensado para ser usado em sites e aplicativos de mensagens. Ele não entra em todos os detalhes técnicos, mas foca em como essas tecnologias podem ser aplicadas para criar uma solução funcional e acessível.

Embora o objetivo seja atingir públicos variados, a prioridade está nas plataformas de comunicação mais usadas. O projeto não pretende substituir os métodos tradicionais de checagem, mas sim ser uma ferramenta a mais para quem precisa de respostas rápidas e confiáveis. Questões como o impacto de longo prazo do uso dessa tecnologia no combate às fake news ficam para estudos futuros.

1.6 Estrutura do Documento

Durante o decorrer da graduação em ciências da informação uma das questões que mais me preocupou foi o impacto negativo que as fake news causa da sociedade, Os impactos criam e manipulam grupos sociais muito das vezes sofrem com o terrorismo digital, afetando questões importantes como politica e economia.

Explorando a evolução da tecnologia junto a inteligência artificial introduzir e apresentar um protótipo simples e funcional de verificação de informações e notícias contextualizadas, criando um senso crítico social, Zeblack um sistema Bots-as-a-service, surgiu como uma solução em nuvem totalmente open source, aplicando técnicas de devops para construção de um sistema em nuvem escalável.

Na metodologia, detalhei o passo a passo de como desenvolvi o Zeblack,

meu agente inteligente de verificação de informações. Foi uma jornada de aprendizado onde mergulhei em pesquisas sobre inteligência artificial e tecnologias como PLN e machine learning usando ferramentas e automação de processos, ligando ao whatsapp.

A criação do Zeblack foi um dos momentos mais emocionantes desse projeto, pois vi na prática como ele pode fazer a diferença na vida das pessoas, ajudando a combater a desinformação. Apesar dos desafios, como garantir precisão e segurança, os resultados foram promissores, mesmo em um ambiente de teste, se mostrando útil não apenas na verificação das informações, mas se tornando uma ferramenta de trabalho de um cientista da informação. No futuro, espero continuar aprimorando o Zeblack, integrando novas funcionalidades e alcançando mais pessoas, mostrando como a Ciência da Informação pode transformar problemas complexos em soluções práticas e humanas.

2. Fundamentação Teórica

2.1 Como as Fake News se Espalham e Afetam a Sociedade, a Política e a Economia

A disseminação de notícias falsas é um problema que afeta toda sociedade, espalha-se por várias áreas importantes da sociedade, como a política, a economia e as relações sociais. As fake News têm um impacto negativo e profundo, influenciando as atitudes e comportamentos tanto de indivíduos quanto de grupos se tornando uma arma de manipulação de massas. Quando as informações são incorretas ou distorcidas, elas não apenas prejudicam a confiança nas fontes, mas também comprometem a qualidade das discussões públicas, afetando decisões e alterando a forma como as pessoas veem o mundo ao seu redor. Branco (2017, p. 58-59) destaca que:

"Nem sempre, contudo, fake news se prestam a endossar publicamente os desejos do usuário. Muitas pessoas compartilham boatos por curiosidade, espanto ou cautela."

As pessoas espalham informações falsas muitas das vezes sem saber a veracidade, mas acabam contribuindo para a propagação da desinformação. Isso é especialmente evidente no campo da política, onde as fakes News têm um impacto manipulativo. Principalmente nos períodos eleitorais, por exemplo, notícias falsas

podem distorcer a percepção do eleitorado, influenciando suas escolhas de maneira injusta. Cândido e Pereira (2020) observam que, em campanhas políticas, o uso de fake News tornou-se uma estratégia comum para manipular a opinião pública, criando narrativas enganosas com o objetivo de desacreditar opositores ou reforçar narrativas convenientes. Isso acontece principalmente em um ambiente onde as notícias são consumidas de forma rápida e sem a devida checagem da veracidade. Esse fenômeno, conhecido como pós-verdade, em que sentimentos e crenças pessoais muitas vezes substituem os fatos, têm alimentado ainda mais a aceitação das fake News no contexto político atual.

Além do impacto político, as fakes News também afetam a economia nacional. informações falsas circulam sobre empresas, produtos ou serviços, isso pode prejudicar o mercado financeiro, causando quedas nas ações e gerando crises de confiança. A manipulação de dados também pode afetar a estabilidade econômica, criando um clima de incerteza entre consumidores, investidores e reguladores. Carvalho (2018) aborda como a desinformação digital pode desestabilizar mercados, criando boatos infundados, prejudicando a confiança pública e gerando efeitos negativos no mercado. Baudrillard (1992, p. 39) ressalta que:

“Outras tantas são impelidas pela vontade de alertar o mundo sobre potenciais perigos que rondam determinadas comunidades. Não se sabe ao certo se é verdade que um homem espancou a mulher e a filha de modo covarde ou que um senhor misterioso abordou criancinhas na saída da escola. Estamos numa sociedade da proliferação, do que continua a crescer sem poder ser medido por seus próprios fins. O excrescente é o que se desenvolve de modo incontrolável, sem respeito pela própria definição, aquilo cujos efeitos multiplicam-se com o desaparecimento das causas. É o que leva a um prodigioso entupimento dos sistemas, a uma desregularem por hipertonia, por excesso de funcionalidade, por saturação. (BAUDRILLARD, 1992 p. 39)”

Baudrillard (1992) destaca como os impactos das fake News acabam sendo mais fortes do que suas próprias verdades, o que torna cada vez mais difícil separar o que é real do que é falso. Assim, as fakes News criam um ciclo vicioso de desinformação, onde as causas se tornam difíceis de identificar e os efeitos dos contextos se espalham de maneira descontrolada. O compartilhamento inconsciente de notícias falsas leva a um ambiente saturado de informações, tornando cada vez mais complicado saber o que é realmente verdadeiro e o que é apenas uma

especulação. Esse contexto prejudica a confiança das pessoas e gera uma sensação geral de incerteza. Como destacam Lorencetl e Salton (2022), a desinformação é espalhada sem a devida checagem, o que faz com que se espalhe ainda mais pelas redes sociais, confundindo cada vez mais os usuários.

2.2 Ferramentas e Técnicas Usadas para Verificar Informações

A verificação de notícias, exige o uso de tecnologias que tornam a análise e a validação de informações mais ágeis e precisas. Uma das técnicas mais eficazes nesse processo é o uso de algoritmos de processamento de linguagem natural (PLN), que são fundamentais para a análise dos conteúdos falsos. Esses algoritmos permitem que as máquinas compreendam e interpretem textos de maneira similar ao pensamento humano, identificando padrões linguísticos como escolhas de palavras, frases e estruturas sintáticas que indicam manipulação ou imprecisão nas informações (Kang et al., 2020).

Segundo Luger (2004, p. 24):

“A inteligência artificial sempre esteve mais preocupada em expandir as capacidades da ciência do que em definir seus limites” (Luger, 2004. p.24)”

Outra abordagem importante para verificar a veracidade das informações é o uso de Machine Learning (ML). Com técnicas como Redes Neurais Artificiais, Árvores de Decisão e Support Vector Machines (SVM), é possível classificar notícias como verdadeiras ou falsas com base em grandes volumes de dados históricos. Para que esses sistemas funcionem de maneira eficaz, é necessário treinar os modelos com dados previamente validados. O treinamento de modelos com conjuntos de dados rotulados é importante para que o sistema aprenda a identificar as características típicas de fake news, garantindo sua precisão e confiabilidade (Oliveira, 2019).

Ferramentas automatizadas, como agentes inteligentes e bots baseados em Inteligência Artificial (IA), têm se mostrado uma solução cada vez mais popular no combate à desinformação. Essas ferramentas são projetadas para interagir diretamente com os usuários, permitindo que qualquer pessoa, independentemente do seu nível de conhecimento técnico, consiga verificar rapidamente se uma notícia é verdadeira ou falsa. De acordo com Pereira (2023), os chatbots baseados em IA

estão se tornando comuns no jornalismo, não só para checar informações, mas também para responder a perguntas sobre a autenticidade dos conteúdos. A facilidade de uso dessas ferramentas tem tornado a verificação de informações mais acessível e democrática, permitindo que qualquer pessoa possa participar da luta contra a desinformação.

A verificação cruzada de fontes também é uma técnica importante na validação de informações. Ferramentas como FactMata e ClaimBuster combinam dados de diversas fontes confiáveis para determinar a veracidade de uma notícia. Usando algoritmos, essas ferramentas cruzam as informações de veículos de mídia e outras fontes de dados confiáveis, ajudando a identificar inconsistências e impedir que a população espalhe notícias falsas (Bykovet *al.*, 2023). Essas ferramentas podem ser integradas a sistemas de criação de agentes inteligentes através do Dify e integrado ao whatsapp através da Evolution api , facilitando ainda mais a verificação em plataformas populares de comunicação, como WhatsApp e Telegram.

O desenvolvimento desse protótipo que utiliza ferramentas para verificação de notícias apresenta desafios. A precisão dessas ferramentas depende de diversos fatores, como a qualidade dos dados utilizados no treinamento e a capacidade dos algoritmos em lidar com grandes volumes de informações. Como apontado por Oliveira (2019), técnicas de IA, como o PLN, podem ser afetadas pela ambiguidade linguística, tornando a verificação mais complexa, especialmente quando se trata de conteúdos em diferentes idiomas ou contextos culturais. Além disso, adaptar as ferramentas para identificar fake news em tempo real, sem prejudicar a experiência do usuário, também é um desafio importante.

Por isso, o uso de tecnologias de verificação automatizada, combinadas com PLN e ML, representa uma solução promissora para combater a desinformação. Porém, é necessário garantir a qualidade dos dados, a adaptação constante das ferramentas e a integração com plataformas de fácil acesso, como o WhatsApp e o Telegram. Implementar esses sistemas contribui para a criação de uma sociedade mais informada e fortalece as ferramentas de verificação, criando um ambiente digital mais seguro e confiável (Bykovet *al.*, 2023).

2.3 Bots como Serviço (BaaS): Como Funcionam e Como Podem contribuir

O conceito de Bots-as-a-Service (BaaS) refere-se a uma abordagem tecnológica em que bots são projetados e disponibilizados como serviços escaláveis por meio de plataformas baseadas em nuvem. Esses sistemas operam utilizando arquiteturas que permitem sua integração com múltiplos canais de comunicação, como aplicativos de mensagens instantâneas, redes sociais e plataformas web. Segundo Zhao et al. (2023), os Agentes inteligentes têm se destacado como ferramentas eficazes para a apresentação de informações verificadas, especialmente em contextos que exigem a disseminação rápida de dados confiáveis. Sua flexibilidade em responder a consultas específicas e processar grandes volumes de informações os torna essenciais em tarefas como verificação de notícias. Conforme Araújo (2020, p. 19):

“Chatbots de domínio aberto podem participar de uma conversa de forma livre com um usuário, não tendo nenhuma meta específica definida. Enquanto chatbots de domínio fechado são criados para ajudar um usuário a atingir uma meta específica.”

A funcionalidade de um sistema BaaS está intrinsecamente ligada ao uso de inteligência artificial (IA) e aprendizado de máquina. Esses elementos possibilitam a personalização do atendimento, o processamento de linguagem natural (PLN) e a análise contextual das solicitações dos usuários. Rocha (2024) enfatiza que os Agentes inteligentes desempenham funções importantes com estratégias de mitigação uma vez que são capazes de identificar padrões de comportamento suspeitos e implementar respostas automatizadas e contextualizadas. Isso demonstra que, além de auxiliarem na comunicação, os agentes inteligentes podem atuar como ferramentas específicas utilizadas para tarefas solicitadas pelo usuário. As características fundamentais que definem o funcionamento ideal de um bot incluem uma combinação de habilidades técnicas, comportamentais e ferramentas adicionais, como destacado por Franklin E Gresser (1996 apud Santos, 2018, p. 11):

“Reatividade (reagir a estímulos do ambiente), autonomia (possuir controle de ação), proatividade (ser capaz de tomar iniciativa), continuidade temporal (estar continuamente ativo), capacidade social (comunicação com outros agentes/pessoas), capacidade de adaptação (através do aprendizado), mobilidade (poder circular dentro do ambiente), flexibilidade (ser dinâmico nas ações de resposta) e caráter (imprimir personalidade e estado emocional).”

O desenvolvimento de um agente inteligente está relacionado à otimização de recursos organizacionais. Conforme Costa (2024), a implementação de bots em ambientes educacionais demonstrou ser uma solução eficiente para demandas de suporte ao aluno, reduzindo a carga de trabalho das equipes administrativas. De forma análoga, na área de verificação de informações, os bots podem processar dados em tempo real e fornecer resultados confiáveis e contextualizados aos usuários, contribuindo para combater a disseminação de desinformação e aprimorar a experiência do usuário.

Um aspecto relevante do funcionamento dos sistemas BaaS é sua capacidade de se adaptar às demandas específicas de cada contexto de aplicação. Michiel Baas (2024) destaca que a utilização da IA em serviços como BaaS transcende sua função técnica, possibilitando a geração de análises baseados em dados socioculturais e contextuais. Essa adaptabilidade é fundamental para a eficácia dos bots na verificação de notícias, pois permite ajustar os critérios de análise e resposta conforme as características do público-alvo e a natureza das informações a serem verificadas.

Os bots como serviço também oferecem benefícios em termos de escalabilidade e acessibilidade. Por serem hospedados em plataformas de computação em nuvem, eles podem ser acessados de qualquer lugar e por diversos dispositivos simultaneamente, garantindo alta disponibilidade e capacidade de atender a um grande número de solicitações. Zhao et al. (2023) apontam que os bots superam as interfaces tradicionais, como páginas da web, ao apresentarem os resultados de checagem de forma mais interativa e de fácil compreensão utilizado em plataformas de comunicação como whatsapp. Essa abordagem melhora a comunicação com o usuário final, aumentando a confiança na informação transmitida.

O modelo BaaS representa um avanço significativo no uso de bots para finalidades diversas, como verificação de informações, suporte ao cliente e no campo da educação em geral. A integração de tecnologias de IA, a capacidade de escalar serviços com a possibilidade de personalização tornam essa abordagem indispensável em um cenário de crescente demanda por soluções tecnológicas acessíveis e eficazes.

2.4 Perfil dos Usuários-Alvo e Cenários de Uso do Zeblack

Os usuários do Zeblack incluem pessoas de diferentes perfis e necessidades. Desde aqueles que querem combater a desinformação no dia a dia até organizações que precisam de ferramentas para verificar fatos de forma automática, o sistema busca atender a uma diversidade de públicos. O que conecta todos esses usuários é o desejo de validar se as informações que encontram em redes sociais e aplicativos de mensagens são confiáveis. Como aponta Costa (2024), o aumento das fake news nesses canais tem levado ao surgimento de soluções tecnológicas que incentivam os usuários a participarem ativamente na checagem de fatos, mostrando o quanto iniciativas como o Zeblack são necessárias.

O Zeblack é pensado para um público bastante diverso: jornalistas que precisam de agilidade, professores interessados em educar sobre a importância da informação correta, pesquisadores que buscam dados confiáveis e até mesmo usuários comuns preocupados em não compartilhar conteúdo falso. Segundo Oliveira et al. (2021), plataformas como o Zeblack têm se mostrado úteis por tornarem a interação entre pessoas e tecnologia mais intuitiva, ajudando quem não tem tanto conhecimento técnico a acessar soluções eficazes. Dessa forma, o Zeblack vai além da simples verificação de fatos, promovendo também uma educação digital que contribui para criar um ambiente mais seguro.

Os contextos em que o Zeblack pode ser usado são muitos. Ele é especialmente relevante em momentos importantes, como eleições, quando verificar fatos se torna essencial. Também é útil em crises, como pandemias, onde é comum que informações falsas se espalhem rapidamente. Além disso, ele pode validar dados compartilhados em grupos de mensagens, um espaço onde fake news costumam circular intensamente. Como sugere Nascimento (2022), bots usados para verificação conseguem reduzir de forma significativa o impacto da desinformação em momentos de tensão social. O Zeblack se destaca nesses cenários ao oferecer análises em tempo real e respostas contextualizadas, atendendo às demandas dos usuários com rapidez e precisão.

Uma das maiores dificuldades em atender tantos perfis de usuários é a necessidade de personalizar o sistema para diferentes contextos. Prado (2022) explica que a inteligência artificial e os algoritmos de aprendizado tornam isso possível, criando ferramentas flexíveis que se adaptam a diferentes públicos. No

caso do Zeblack, essa adaptação ocorre por meio de prompts customizáveis e integrações com fontes de dados confiáveis, garantindo que cada usuário tenha acesso a respostas relevantes.

Outro ponto forte do Zeblack é a facilidade de uso, pensada especialmente para quem não é tão familiarizado com tecnologia. Como De Carvalho Júnior et al. (2018) destacam, um chatbot eficiente é aquele que oferece respostas rápidas e precisas, mas que também seja intuitivo e fácil de usar. Seguindo essa lógica, o Zeblack integra-se a plataformas, permitindo que as pessoas interajam diretamente com o sistema sem complicações. Mais do que isso, ele se diferencia por sua versatilidade, atendendo tanto às necessidades mais simples, como esclarecer uma dúvida, quanto às mais complexas, como verificar a veracidade de notícias relevantes.

Cunha (2020) sobre o uso do ChatBot:

“...salienta-se que o serviço proposto se diferencia dos demais pela sua forma de monetização via mensagens publicitárias direcionadas e por promover um nicho mais abrangente, englobando quaisquer temáticas que o usuário desejar checar a veracidade de informações a respeito.

Essa abordagem amplia muito as possibilidades de uso do Zeblack, transformando-o em uma ferramenta que vai além de atender apenas tarefas específicas. Ele se adapta facilmente a diferentes necessidades e contextos, alcançando um público diverso. Além disso, é importante levar em conta como essa interação entre os usuários e a tecnologia pode ter um impacto positivo na sociedade. Chiarello, Herrera e Barbosa Filho (2024) destacam que, no combate à desinformação, as soluções tecnológicas precisam ser pensadas de forma mais humana, buscando não só resolver problemas técnicos, mas também gerar mudanças que beneficiem as comunidades. O Zeblack segue exatamente esse caminho, entregando informações verificadas de maneira clara e acessível, ajudando os usuários a desenvolverem mais confiança e senso crítico.

Na prática, o Zeblack funciona com integrações inteligentes, como APIs e modelos avançados de IA garantindo que as respostas sejam sempre relevantes e bem contextualizadas. Costa (2024) reforça que, para serem eficazes, essas ferramentas precisam ir além de apenas identificar o que é falso ou verdadeiro; elas devem mostrar provas que sustentem suas análises. O Zeblack adota essa lógica ao permitir que os usuários não só vejam o resultado da verificação, mas também

entendam como ele chegou àquela conclusão, tornando o processo mais transparente e confiável.

2.5 Os Desafios de Criar um Sistema para Verificação Automática de Notícias

Criar um sistema automatizado para verificar notícias falsas envolve muitos desafios, principalmente devido à complexidade de distinguir entre informações verdadeiras e falsas em um ambiente digital com grande volume de dados. Segundo Lorenceti e Salton (2022), a detecção de fake news, especialmente em redes sociais como o Twitter, depende de algoritmos baseados em aprendizado de máquina, capazes de classificar rapidamente grandes quantidades de informação. No entanto, esses sistemas enfrentam dificuldades em capturar as nuances da linguagem, como ambiguidades ou variações no uso de palavras, que são comuns em notícias falsas.

Para que o sistema funcione, é necessário aplicar técnicas avançadas de engenharia de prompt junto de algoritmos de processamento de linguagem natural (PLN) para analisar o conteúdo das notícias. Kang et al. (2020) explicam que o PLN é amplamente utilizado para extrair informações importantes de grandes volumes de texto, o que é essencial para a verificação automatizada de notícias. No entanto, o grande desafio está em entender o contexto em que a notícia é apresentada, já que uma verificação simples do conteúdo sem considerar o ambiente em que ele circula pode levar a falhas na identificação de desinformação.

“Além disso, as matérias escritas por automatização foram melhor classificadas do que as escritas por humanos, em termos de credibilidade, mas foram pior avaliadas em questão de legibilidade. No geral, ainda assim, as diferenças entre as avaliações foram pequenas. O resultado surpreendeu. Estudantes de comunicação, que espera-se que tenham um nível maior de conhecimento de mídia do que os consumidores de notícias no geral, foram incapazes de fazer a distinção entre matérias escritas por humanos e por robôs, e ainda classificaram as últimas como algo com mais credibilidade. (GRAEFE, 2019, p.12-13)

Outro obstáculo importante é a verificação das fontes de informação. Cândido e Pereira (2020) destacam que a identificação de fontes confiáveis se torna cada vez mais difícil, especialmente quando notícias falsas são disfarçadas de informações legítimas. Isso é ainda mais complicado pela rapidez com que as informações se espalham, o que pode comprometer a eficácia de um sistema automatizado que não consegue acompanhar a velocidade das notícias falsas. Nesse sentido, garantir que

o sistema tenha uma base sólida de fontes confiáveis e consiga rastrear a origem das notícias é fundamental para seu sucesso.

É fundamental estimular que os profissionais explorem mais suas ferramentas e conhecimentos para otimizar suas funções.

Creio que os algoritmos serão responsáveis por forçar os jornalistas a pensarem mais sobre as definições de suas principais capacidades humanas, como o desenvolvimento de inteligência social e emocional, curiosidade, autenticidade, humildade, empatia, serem bons ouvintes, colaborativos e abertos à aprendizagem. Os jornalistas tendem a definir sua ocupação pelas tarefas que são cumpridas, e não pelas habilidades e conhecimentos que uma pessoa precisa possuir para cumpri-las (VAN DALEN apud LINDÉN, 2018, p.19)

A qualidade dos dados usados para treinar os sistemas também é crucial. Luger (2004) aponta que, para um sistema de inteligência artificial ser eficaz, ele precisa ser treinado com dados completos e livres de vieses. No caso de um sistema de verificação de notícias, a falta de dados de boa qualidade pode fazer com que o algoritmo cometa erros ao identificar se uma notícia é verdadeira ou falsa, o que pode acabar prejudicando a confiabilidade do protótipo.

Outro ponto a ser considerado é o aspecto psicológico das fake news. Muitas vezes, essas notícias são construídas com base em apelos emocionais, estratégias de manipulação e falácias lógicas. Costa (2024) argumenta que os sistemas de verificação precisam ir além de identificar palavras-chave e considerar a maneira como as notícias influenciam emocionalmente as pessoas. Para ser mais eficaz, o sistema precisa entender como as fake news manipulam os leitores e adaptar seus critérios de verificação de acordo com esse aspecto.

Em um nível mais geral, a criação de sistemas autônomos para a verificação de notícias continua sendo uma área de pesquisa intensiva. Franklin e Graesser (1996) afirmam que a construção de sistemas capazes de tomar decisões informadas de forma independente é uma tarefa complexa e repetitiva, especialmente em contextos imprevisíveis e cheios de incertezas. Para detectar fake news de forma precisa, é preciso que esses sistemas combinem não apenas ferramentas avançadas, mas também uma compreensão das dinâmicas sociais e comportamentais envolvidas na criação e disseminação dessas notícias.

Criar um protótipo de verificação de notícias eficiente exige uma combinação de várias abordagens, incluindo PLN, aprendizado de máquina e uma análise detalhada do contexto e das fontes. Embora já existam avanços tecnológicos, ainda

há muito a ser feito para que esses sistemas se tornem completamente eficazes em enfrentar o desafio das fake news.

2.6 Questões Éticas e de Privacidade ao Usar Bots para Verificar Informações

Quando se fala sobre o uso de agentes inteligentes para verificar informações, é impossível não considerar as questões éticas e de privacidade que surgem com a automação desse processo. Embora os bots possam ser ferramentas muito eficientes para lidar com a grande quantidade de dados disponíveis na internet, eles trazem desafios que precisam ser observados com atenção.

Uma das preocupações centrais é a transparência. Muitos sistemas de verificação automatizada funcionam com algoritmos complexos que, muitas vezes, não são claros para o público. Isso pode gerar desconfiança, já que as pessoas não sabem exatamente como o bot chegou à sua conclusão. Segundo Costa (2024), quando um sistema automatizado de verificação de informações não é transparente, pode ser difícil para o usuário confiar nos resultados. Se um bot classificar uma notícia falsa como verdadeira, por exemplo, sem uma explicação clara ou contexto de como isso aconteceu, a confiança no sistema pode diminuir bastante. A falta de clareza sobre o funcionamento desses bots pode até levar à disseminação de desinformação, caso o sistema seja manipulado ou falhe em sua análise.

“A capacidade dessas ferramentas de produzir texto logicamente coerente, bem referenciado e estilisticamente apropriado, que pode se passar por um texto escrito do zero em um contexto acadêmico. Isso, portanto, equivale a uma economia substancial de tempo e pode até mesmo aumentar a produtividade dos pesquisadores, pois é uma tarefa bastante complexa que é automatizada ao longo do processo de escrita (LUND. et al, 2023, p. 6).”

A questão da responsabilidade é importante. Quando um bot comete um erro, como classificar uma informação incorretamente, quem é o responsável por isso? Será o desenvolvedor do sistema, a plataforma que o utiliza ou o próprio usuário que compartilhou a informação? Franklin e Graesser (1996) destacam que, no caso da inteligência artificial, essa responsabilidade não é clara, o que pode complicar a situação quando ocorrem falhas. A falta de um sistema claro de responsabilidade pode tornar difícil atribuir quem deve ser responsabilizado por erros cometidos pelos

agentes, Entretanto em um ambiente em produção, com implementação de relatório de impacto (RIPD) conforme a LGPD, resolveria essa questão.

Outro ponto importante é a privacidade dos dados. A coleta desses dados sem o consentimento explícito das pessoas pode configurar uma violação da privacidade. Como Cândido e Pereira (2020) apontam, isso pode ir contra direitos fundamentais de proteção de dados, como o previsto pela Lei Geral de Proteção de Dados (LGPD) no Brasil, ou pelo GDPR na União Europeia.

O viés nos algoritmos também é uma preocupação crescente. Sempre verificar a origem dos dados da LLM utilizada, há sempre o risco de esses agentes apresentarem um viés, por conta dos dados que foram usados para treiná-los. Luger (2004) destaca que se os dados utilizados para treinar os bots já contiverem preconceitos ou falhas, esses mesmos preconceitos podem ser replicados.

“Embora essas ferramentas tenham muito a oferecer em termos de aumentar a eficiência e acessibilidade da pesquisa acadêmica, elas também têm implicações éticas que precisam ser devidamente refletidas para garantir que a pureza e a autenticidade do discurso acadêmico não sejam comprometidas (LUND. et al, 2023, p. 8)”.

A manipulação de informações é outro risco associado ao uso de agentes inteligentes. Embora esses sistemas sejam projetados para combater a desinformação, eles também podem ser usados de forma estratégica para influenciar a opinião pública. Como alertam Baas (2024) e Costa (2024), bots podem ser programados para priorizar ou despriorizar certos tipos de notícias, dependendo de quem os controla. Isso pode criar um cenário em que, ao invés de promover uma verificação imparcial, os bots são usados para espalhar narrativas específicas, manipulando o público a favor de uma determinada agenda.

A coleta de dados sensíveis é outro ponto delicado. Muitos agentes de verificação de informações acessam dados sobre os usuários, como suas interações em redes sociais e os sites que visitam. Cândido e Pereira (2020) destacam que, sem o devido consentimento, o uso desses dados para personalizar a verificação de notícias pode ser uma grave violação da privacidade. Isso pode comprometer a autonomia dos indivíduos e aumentar o risco de que seus dados sejam mal utilizados.

Ao criar e implementar bots inteligentes para verificar informações, é fundamental que os desenvolvedores levem em conta as questões éticas e de

privacidade envolvidas. A transparência, a responsabilidade, o viés algorítmico e a privacidade dos dados são pontos-chave para garantir que esses sistemas sejam eficazes e respeitem os direitos dos usuários. Sem esses cuidados, a utilização de bots para verificação de informações pode não apenas falhar em sua missão de combater a desinformação, mas também contribuir para problemas maiores, como a invasão da privacidade e a manipulação de dados.

3. Metodologia

O Zeblack, um agente inteligente projetado para verificar informações e notícias em tempo real, foi desenvolvido a partir de uma abordagem prática que combinou pesquisa e aplicação direta. O processo envolveu uma base teórica sólida, planejamento cuidadoso e testes para garantir sua funcionalidade e eficiência.

A pesquisa inicial focou em explorar temas como fake news, inteligência artificial, aprendizado de máquina e processamento de linguagem natural. Esse estudo ajudou a identificar ferramentas importantes, como modelos avançados de linguagem para análise textual, algoritmos específicos para detectar padrões em notícias falsas e sistemas automatizados para interagir com os usuários.

O desenvolvimento do protótipo seguiu um planejamento estruturado, incluindo a escolha de uma infraestrutura em nuvem com base na Hetzner Cloud, por sua combinação de custo e escalabilidade. A configuração utilizou containers Docker, gerenciados pelo Docker Swarm, com suporte de ferramentas como Traefik e Portainer para simplificar a administração. A arquitetura do sistema integrou APIs, como a Evolution API para comunicação via WhatsApp, além de um banco de dados PostgreSQL e a utilização do Minio para armazenamento de objetos e recuperação de informações.

A criação do agente utilizou a plataforma Dify, com personalização detalhada de prompts e a integração de ferramentas adicionais, como scraping, regex e consultas a fontes como Wikipedia, para ampliar as capacidades de análise e verificação. Tudo foi configurado para permitir um desempenho ágil e preciso em diferentes cenários.

A solução passou por testes em situações simuladas e reais, com o objetivo de medir a precisão na identificação de notícias falsas, a agilidade nas respostas e a facilidade de uso. Além disso, usuários potenciais contribuíram com feedbacks para refinar o sistema, garantindo que fosse confiável e eficiente para as demandas do dia a dia.

4. Proposta

O projeto Zeblack tem como objetivo criar uma solução inovadora no combate à desinformação, utilizando um agente inteligente baseado no conceito de Bots-as-a-Service (BaaS). Esse sistema permite a verificação de informações e notícias em tempo real, atendendo à crescente demanda por ferramentas que ajudem a combater as fakes News. O Zeblack foi pensado para ser uma solução acessível e escalável, usando tecnologias modernas para validar a autenticidade das informações compartilhadas nas redes digitais.

O impacto das fake news é imenso, afetando áreas cruciais como política, economia e relações sociais. O Zeblack foi desenvolvido para enfrentar esse desafio, oferecendo uma maneira rápida e eficiente de verificar informações, ajudando a reduzir os efeitos negativos da desinformação.

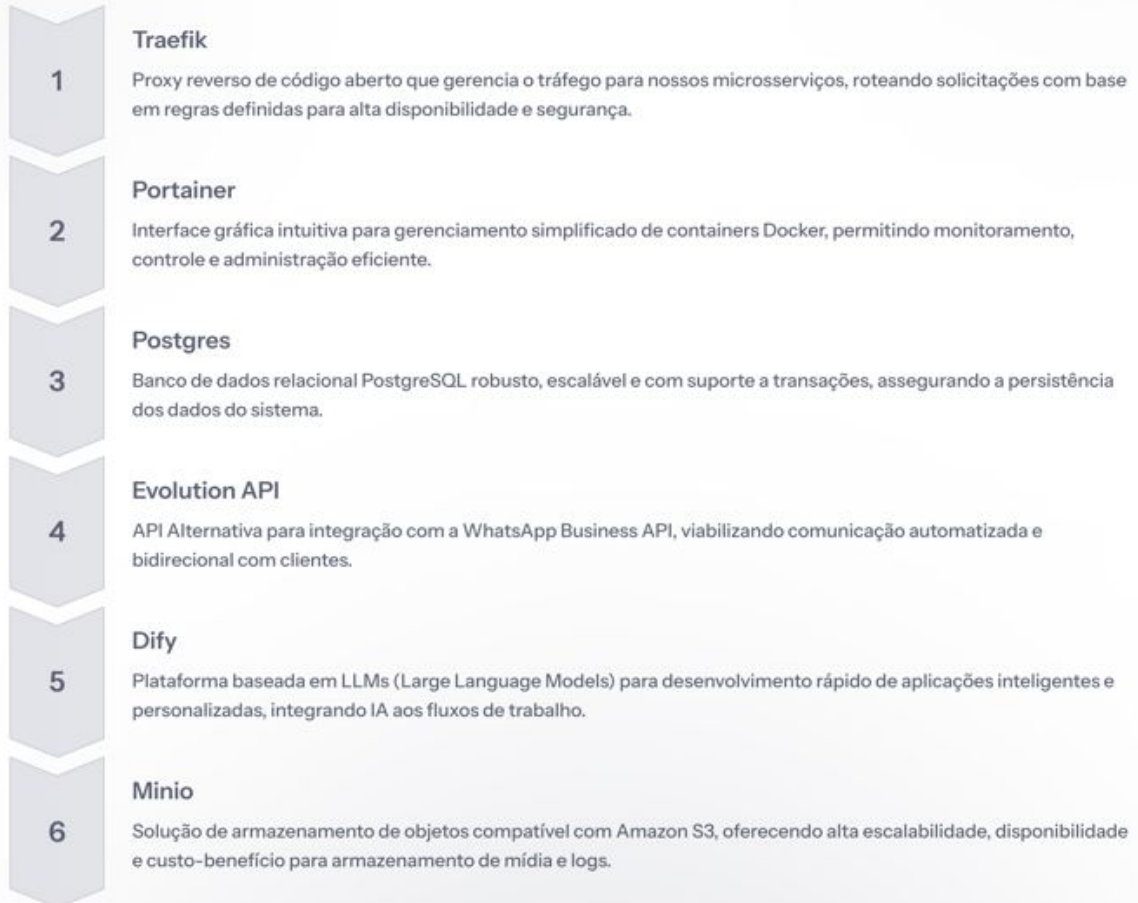
A arquitetura do Zeblack é dividida em quatro módulos principais. O primeiro é o módulo de entrada de dados, que recebe informações através de plataformas como o WhatsApp. A partir daí, o sistema processa as mensagens usando APIs específicas para analisar as solicitações. O segundo módulo é o de análise, que utiliza ferramentas para detectar inconsistências nas informações, realizando verificações cruzadas com dados de outras fontes. O terceiro módulo é a saída de dados, onde as respostas são fornecidas de forma estruturada e contextualizada, mostrando a confiabilidade das informações verificadas. Por fim, a camada de armazenamento utiliza o PostgreSQL para registrar interações e o Minio para armazenar arquivos e gerar relatórios.

Entre os principais diferenciais do Zeblack estão a escalabilidade, com a implementação em nuvem garantindo alta disponibilidade, e a facilidade de uso, integrando-se com plataformas populares como o WhatsApp. Além disso, a confiabilidade do sistema é um ponto forte, com o uso de ferramentas robustas que asseguram respostas precisas e transparentes.

Uma representação visual foi criada para ilustrar o funcionamento do Zeblack, facilitando a compreensão do fluxo de dados, desde o recebimento da solicitação até a entrega da resposta ao usuário. Com a implementação do Zeblack, espera-se uma significativa redução na disseminação de fake news, o aumento da confiança nas informações consumidas online e o fortalecimento da sociedade contra os impactos negativos da desinformação. A ideia é transformar um problema social em uma

solução prática, fazendo com que o Zeblack se torne uma referência em ferramentas de combate à desinformação.

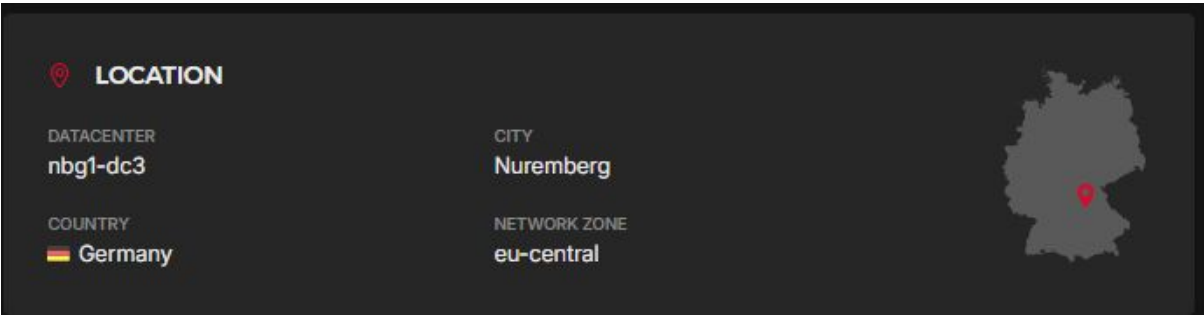
Fluxograma Protótipo Zeblack



5. Implementação e Avaliações

Informo aos leitores que todo material utilizado, será organizado e disponibilizado no drive em anexo. Primeiramente escolhemos nosso provedor de serviço em nuvem, local onde vamos construir toda nossa infraestrutura tecnológica, nesse projeto utilizouse a hetzner cloud <https://www.hetzner.com> devido ao seu custo benefício, AWS e Digitalocean também são boas opções.

FIGURA 1 – localização data center: Nuremberg / Alemanha

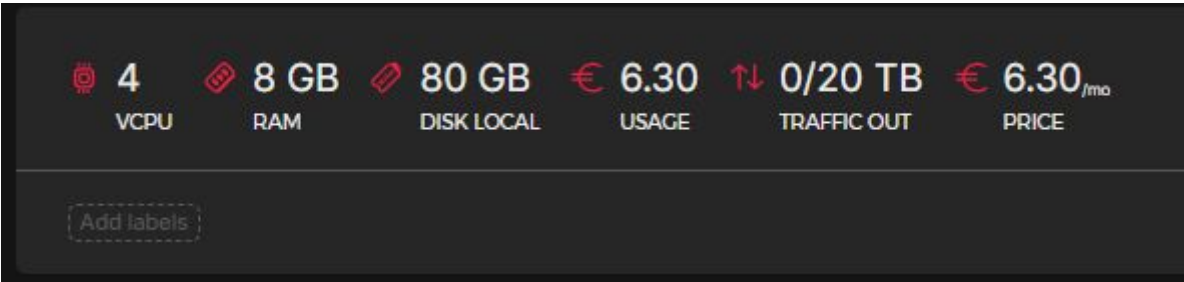


Fonte: hetzner cloud (2024).

Para a conjuntura do projeto optouse por uma máquina simples, porém funcional.

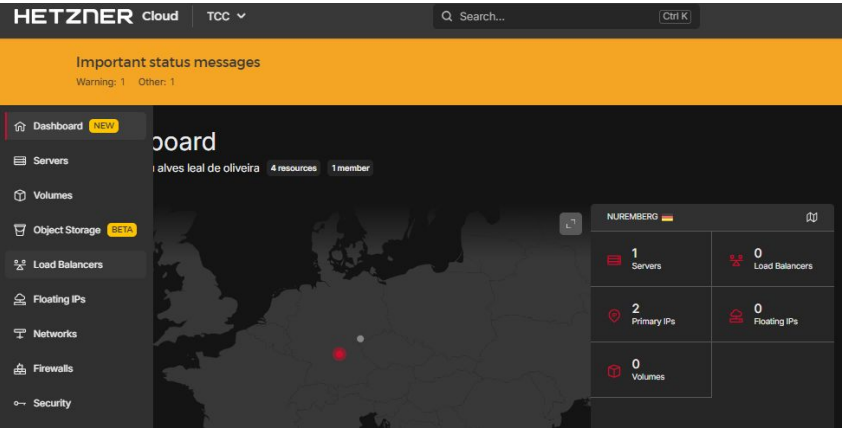
Dicas: mínimo 8 GB de RAM para garantir uma boa performance. Se você está apenas testando, pode optar por uma máquina menor, mas saiba que o desempenho será limitado.

FIGURA 2 – Configurações da Máquina utilizada



Fonte: hetzner cloud (2024).

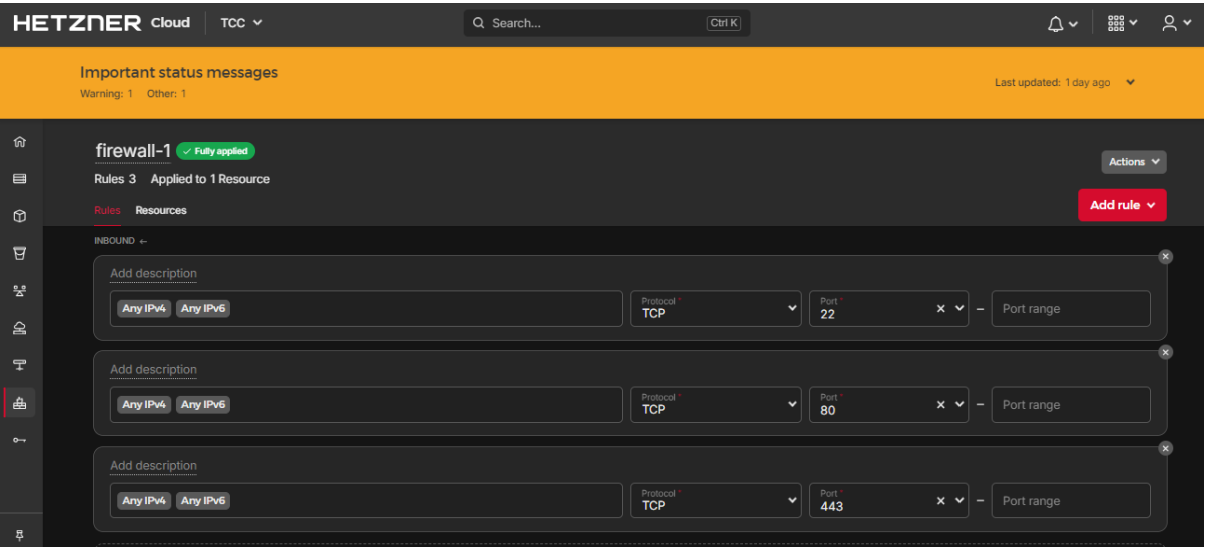
FIGURA 3 – Rede: 1gb , 2 Primary IPs, 1 ipv4 e outro ipv6.



Fonte: hetzner cloud (2024).

Em seguida, abriram-se as portas essenciais: 22, 80 e 443, enquanto todas as demais permaneceram fechadas para garantir maior segurança. O processo foi realizado diretamente no painel de gerenciamento da Hetzner, onde foi criada uma regra de firewall simples e eficaz.

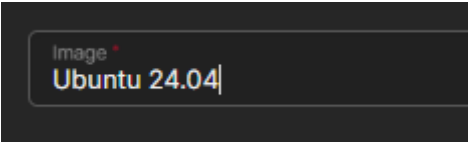
FIGURA 4 – Configuração de Firewall



Fonte: hetzner cloud (2024).

Em seguida, seleciona-se o sistema operacional a ser utilizado no servidor, o Ubuntu

FIGURA 5– Sistema operacional utilizado no servidor



Fonte: hetzner cloud (2024).

Faça o download das seguintes ferramentas para continuar:

- **Bitvise:** Cliente SSH para Windows. <https://bitvise.com>
- **Sublime Text:** Editor de código simples gratuito, <https://www.sublimetext.com>
Alternativa para Linux/Mac: **Termius.** <https://termius.com>

É necessário utilizar um domínio no projeto. Será empregado um domínio existente que está inativo: ofertadoze.com.br. A partir dele, serão criados subdomínios que possibilitarão o acesso às ferramentas instaladas nos containers, gerenciadas via Portainer. Esses subdomínios serão administrados pela Cloudflare:

<https://www.cloudflare.com/pt-br>

Dica: Caso seja necessário adquirir um domínio, pode-se utilizar o serviço do Registro.br:

<https://registro.br>

No painel da Cloudflare, configuram-se os subdomínios necessários para cada componente do projeto. O primeiro subdomínio será do tipo A, apontando para o IP do servidor. É importante desmarcar a opção de proxy, pois esse domínio terá a função de gerenciar os demais e facilitar alterações futuras, como a troca de provedor de nuvem, bastando atualizar o IP do novo servidor.

FIGURA 6 – Criação de Domínio

admin.ofertadoze.com.br aponta para 188.245.155.65.

Tipo	Nome (necessário)	Endereço IPv4 (necessário)	Status do proxy	TTL
A	admin	188.245.155.65	☐ Somente DNS	Auto

Use @ para raiz

Fonte: Cloudflare(2024).

Os próximos subdomínios serão do Tipo CNAME pois irá apontar para nosso domínio principal: admin.ofertadoze.com.br, Sempre desative o proxy na criação dos subdomínios:

FIGURA 7 – PORTAINER: portainer.ofertadoze.com.br

portainer.ofertadoze.com.br é um alias de admin.ofertadoze.com.br.

Tipo
CNAME

Nome
(necessário)
portainer
Use @ para raiz

Destino
(necessário)
admin.ofertadoze.com.
Por exemplo: www.example.com

Status do proxy
☒ Somente DNS

TTL
Auto

Fonte: Cloudflare(2024).

FIGURA 8 – Evolution API: evoapi.ofertadoze.com.br

evoapi.ofertadoze.com.br é um alias de admin.ofertadoze.com.br.

Tipo
CNAME

Nome
(necessário)
evoapi
Use @ para raiz

Destino
(necessário)
admin.ofertadoze.com.br
Por exemplo: www.example.com

Status do proxy
☒ Somente DNS

TTL
Auto

Fonte:Cloudflare(2024).

FIGURA 9 – subdomínios configurados

ofertadoze.com.br Ativo Asterisco Plano Free

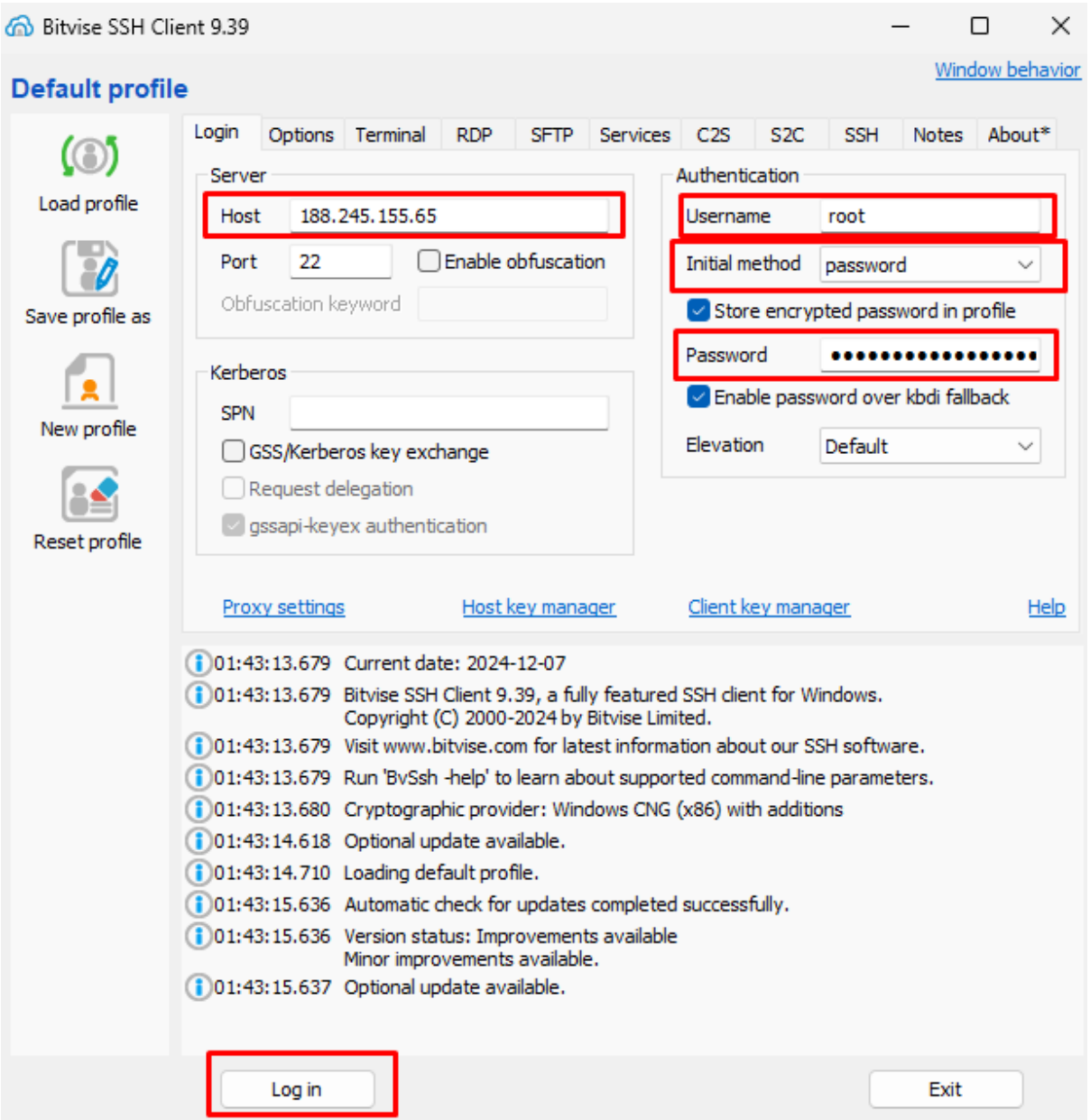
☐	Tipo ⓘ ▲	Nome ⓘ	Conteúdo ⓘ	Status do proxy ⓘ	TTL ⓘ	Ações
☐	A	admin	188.245.155.65	Somente DNS	Auto	Editar ▶
☐	CNAME	chatw	admin.ofertadoz...	Somente DNS	Auto	Editar ▶
☐	CNAME	evoapi	admin.ofertadoz...	Somente DNS	Auto	Editar ▶
☐	CNAME	n8webhook	admin.ofertadoz...	Somente DNS	Auto	Editar ▶
☐	CNAME	n8workflow	admin.ofertadoz...	Somente DNS	Auto	Editar ▶
☐	CNAME	portainer	admin.ofertadoz...	Somente DNS	Auto	Editar ▶

Fonte:Cloudflare(2024).

Agora que a máquina foi contratada e o domínio configurado, o próximo passo é realizar as instalações e configurações do servidor. Primeiramente, acesse o servidor em nuvem utilizando um cliente SSH.

- Host:** será o número de ip do teu servidor
 - Username:** usuário criado na criação do teu servidor.
 - Password:** senha criada na criação do teu servidor.
- Após preencher os dados clique em login.

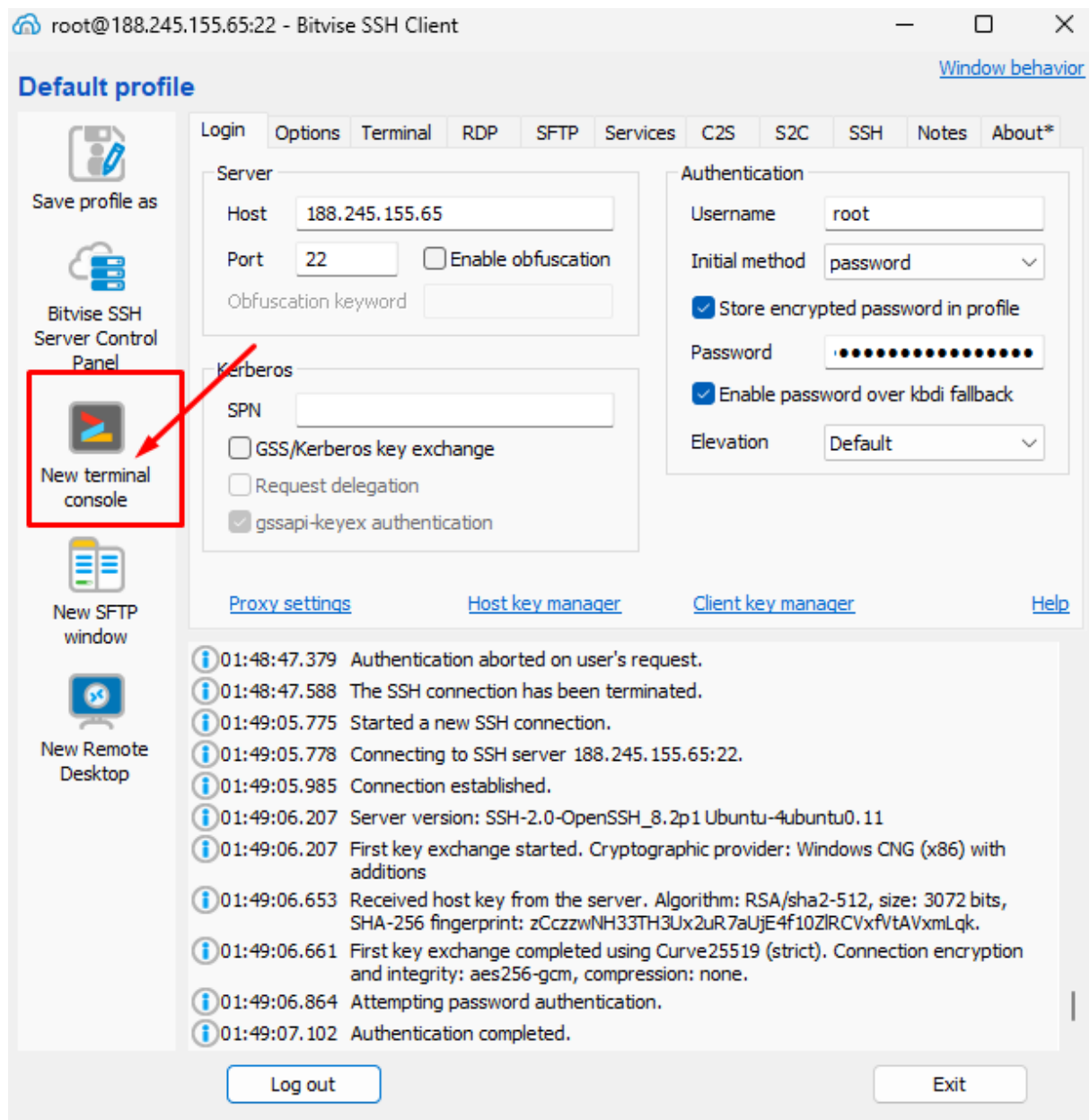
FIGURA 10 – Configuração Bitvise



Fonte:Bitvise(2024).

Após fazer conexão com o servidor irá aparecer as opções: **New Terminal console**: utilizaremos para as primeiras configurações básicas e acesso ao servidor através de linha de comando.

FIGURA 11 – Fazendo Login Bitvise



Fonte:Bitvise(2024).

Ao clicar, a seguinte tela será exibida, contendo as informações do servidor:

```
root@188.245.155.65:22 - Bitvise xterm - root@TCC: ~
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-45-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Fri Nov 29 08:23:27 PM UTC 2024

System load:  0.08               Processes:            143
Usage of /:   1.9% of 74.79GB    Users logged in:     0
Memory usage: 3%                IPv4 address for eth0: 188.245.155.65
Swap usage:   0%                IPv6 address for eth0: 2a01:4f8:1c1b:b33f::1

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

7 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

*** System restart required ***
Last login: Thu Nov 28 20:35:45 2024 from 200.225.115.43
root@TCC:~#
```

Fonte: Bitvise(2024).

Preparando o servidor:

Inicialmente, o servidor deve ser atualizado utilizando o seguinte comando:

sudo apt-get update ; apt-get install -y apparmor-utils

Em seguida, é necessário baixar e instalar o Docker, que será o virtualizador de containers:

curl -fsSL https://get.docker.com | bash

Após a instalação, deve-se inicializar o orquestrador de containers utilizando o Docker Swarm:

docker swarm init

Por fim, uma rede interna será criada para que os containers possam se comunicar

docker network create --driver=overlay network_public

Configuração dos contêineres:

Serão utilizados arquivos YAML (Compose files) para instalar as imagens nos contêineres. Todos os arquivos de código estarão disponíveis.

Cada stack representa um sistema web que será instalado e configurado, incluindo:

Traefik: É um proxy reverso moderno e dinâmico que facilita a gestão de tráfego para micro serviços e aplicações em contêineres. Ele oferece recursos como balanceamento de carga, roteamento dinâmico e suporte a protocolos modernos, além de integração com orquestradores como Docker e Kubernetes.

Portainer: Fornece uma interface gráfica intuitiva para gerenciar ambientes Docker, simplificando a implantação, configuração e monitoramento de contêineres e stacks. Facilita a administração de recursos Docker, tornando-o acessível mesmo para usuários com menos experiência em linha de comando.

Postgres: Conhecido também como PostgreSQL, é um sistema de gerenciamento de banco de dados relacional de código aberto, reconhecido por sua robustez, extensibilidade e conformidade com padrões SQL. Suporta uma ampla variedade de tipos de dados e oferece recursos avançados como transações ACID, índices complexos e replicação.

Evolution API: Trata-se de uma API alternativa para integração com o WhatsApp, permitindo que aplicações enviem e recebam mensagens através da plataforma de mensagens. Facilita a automação de interações e a integração de serviços com o WhatsApp.

Dify: É uma plataforma de código aberto para desenvolvimento de aplicativos baseados em Modelos de Linguagem de Grande Porte (LLMs). Oferece uma interface intuitiva que combina workflows de IA, pipelines de Recuperação de Informação com Geração (RAG), capacidades de agentes, gerenciamento de modelos e recursos de observabilidade, permitindo uma transição rápida do protótipo à produção.

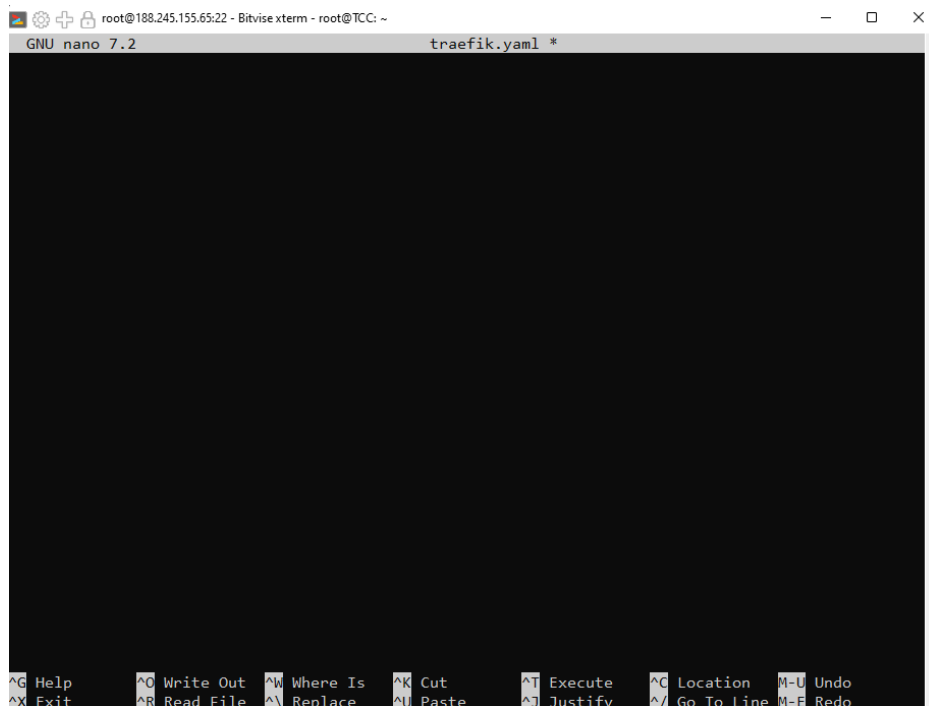
Minio: É uma solução de armazenamento de objetos de alto desempenho, compatível com a API S3 da Amazon. Projetado para ser nativo em Kubernetes, o MinIO é adequado para implantações em nuvens públicas e privadas, oferecendo escalabilidade e desempenho aprimorados para cargas de trabalho modernas, como inteligência artificial e análise de dados.

Configuração e instalação do Traefik:

No terminal do servidor, deve-se executar o seguinte comando para criar o arquivo onde será inserido o código de instalação da Stack:
nano traefik.yaml

Ao executar o comando, será exibida uma tela vazia:

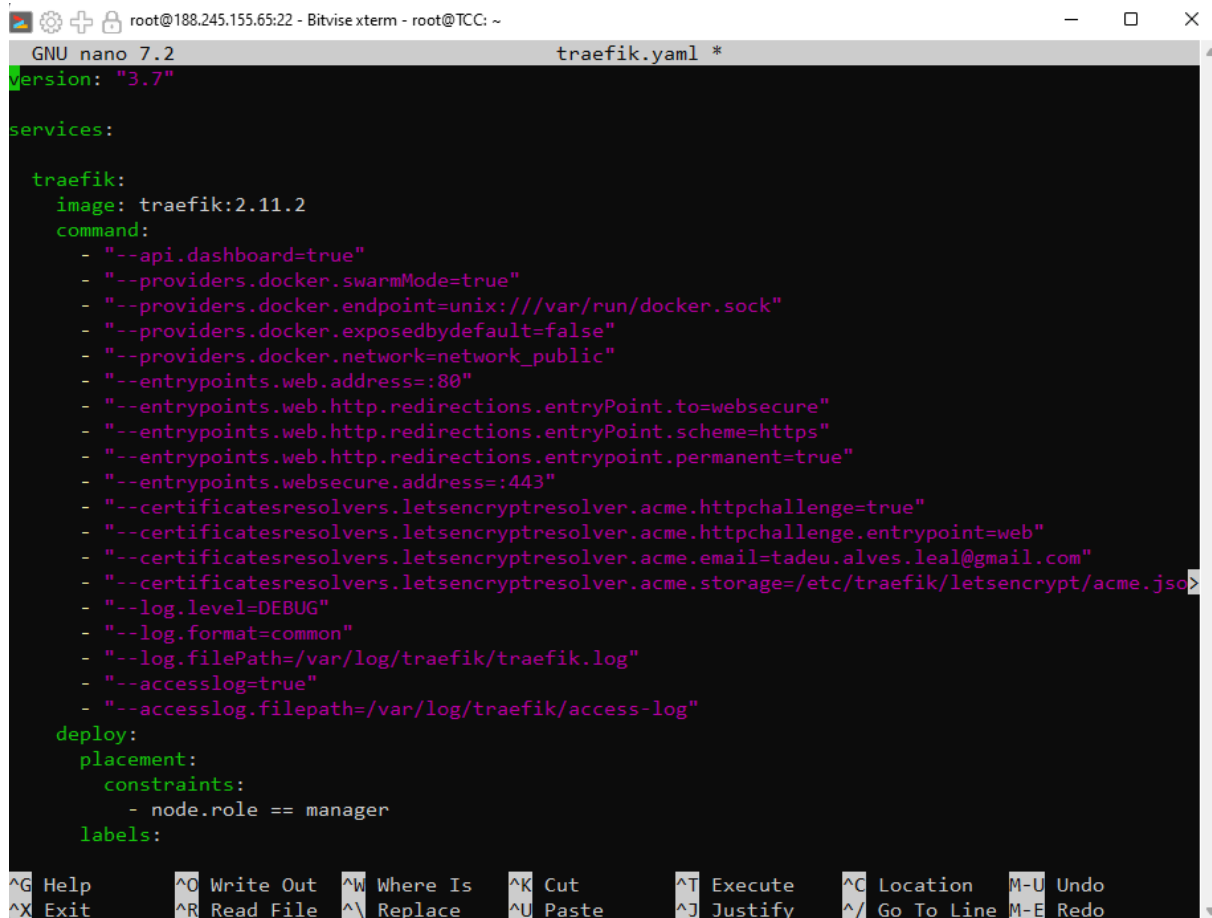
FIGURA 13– Início Criação Stack Traefik.



Fonte:Bitvise(2024).

Selecione e copie o código da stack do Traefik. Para colar, pode-se utilizar o botão direito do mouse. Para salvar, deve-se pressionar CTRL + X.

FIGURA 14 – Criação Stack Traefik.



```
GNU nano 7.2 traefik.yaml *
version: "3.7"

services:

  traefik:
    image: traefik:2.11.2
    command:
      - "--api.dashboard=true"
      - "--providers.docker.swarmMode=true"
      - "--providers.docker.endpoint=unix:///var/run/docker.sock"
      - "--providers.docker.exposedbydefault=false"
      - "--providers.docker.network=network_public"
      - "--entrypoints.web.address=:80"
      - "--entrypoints.web.http.redirections.entryPoint.to=websecure"
      - "--entrypoints.web.http.redirections.entryPoint.scheme=https"
      - "--entrypoints.web.http.redirections.entrypoint.permanent=true"
      - "--entrypoints.websecure.address=:443"
      - "--certificatesresolvers.letsencryptresolver.acme.httpchallenge=true"
      - "--certificatesresolvers.letsencryptresolver.acme.httpchallenge.entrypoint=web"
      - "--certificatesresolvers.letsencryptresolver.acme.email=tadeu.alves.leal@gmail.com"
      - "--certificatesresolvers.letsencryptresolver.acme.storage=/etc/traefik/letsencrypt/acme.js
      - "--log.level=DEBUG"
      - "--log.format=common"
      - "--log.filePath=/var/log/traefik/traefik.log"
      - "--accesslog=true"
      - "--accesslog.filePath=/var/log/traefik/access-log"
    deploy:
      placement:
        constraints:
          - node.role == manager
      labels:
```

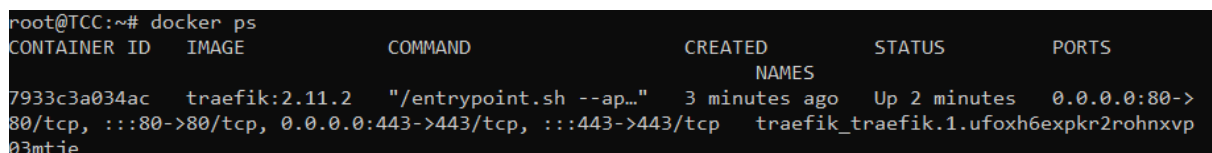
Fonte:Bitvise(2024).

Agora será realizado o deploy do Traefik, iniciando o container.

No terminal do servidor, deve-se aplicar o seguinte comando:

docker stack deploy --prune --resolve-image always -c traefik.yaml traefik

FIGURA 15 – Comando Docker ps



```
root@TCC:~# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS
7933c3a034ac   traefik:2.11.2 "/entrypoint.sh --ap..." 3 minutes ago Up 2 minutes  0.0.0.0:80->80/tcp, :::80->80/tcp, 0.0.0.0:443->443/tcp, :::443->443/tcp
traefik_traefik.1.ufoxh6expkr2rohnxvp03mtje
```

Fonte:Bitvise(2024).

Configuração e instalação do Portainer:

Os mesmos passos das imagens anteriores devem ser repetidos.

Será criado novamente o arquivo YAML (Compose file) para o Portainer:

nano portainer.yaml

Abra o arquivo da stack do Portainer no Sublime Text.

Modificação da stack:

Na linha 32, localize a seguinte configuração:

```
"traefik.http.routers.portainer.rule=Host(`seudominio.com.br`)"
```

Substitua `seudominio.com.br` pelo subdomínio criado no Cloudflare.

No caso do protótipo, será utilizado o subdomínio: `portainer.ofertadoze.com.br`.

Após modificar a stack, deve-se copiá-la, colá-la e salvar o arquivo como `portainer.yaml`.

O deploy da stack será realizado utilizando o seguinte comando:

`docker stack deploy --prune --resolve-image always -c portainer.yaml portainer`

Dica: Após executar o comando, recomenda-se aguardar entre 1 a 3 minutos para que toda a aplicação seja inicializada.

Para testar a aplicação, insira o domínio utilizado no navegador. No caso do protótipo:

`https://portainer.ofertadoze.com.br`

Se tudo estiver configurado corretamente, será exibida a tela de finalização da instalação diretamente no navegador.

Dica: É importante finalizar a instalação rapidamente. Caso contrário, será necessário remover a stack e instalá-la novamente.

FIGURA 16— Finalização Instalação Portainer

✓ New Portainer installation

Please create the initial administrator user.

Username

Password

Confirm password

⚠ The password must be at least 12 characters long.

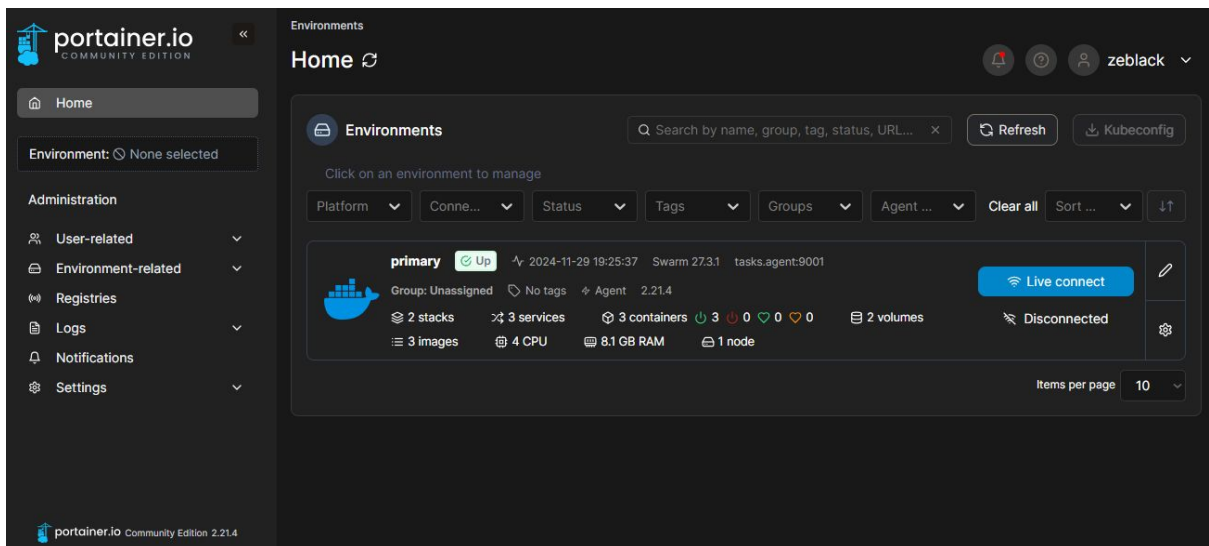
Create user

☒ Allow collection of anonymous statistics. You can find more information about this in our [privacy policy](#).

➤ Restore Portainer from backup

Fonte:Portainer(2024).

FIGURA 17— Tela inicial da aplicação após a instalação



Fonte:Portainer(2024).

Para listar os contêineres em execução, utilize o comando:

docker ps

Este comando exibirá uma lista dos contêineres ativos, incluindo informações como ID, imagem, status, portas e nomes dos contêineres.

FIGURA 18 – comando docker ps

```
root@TCC:~# docker ps
```

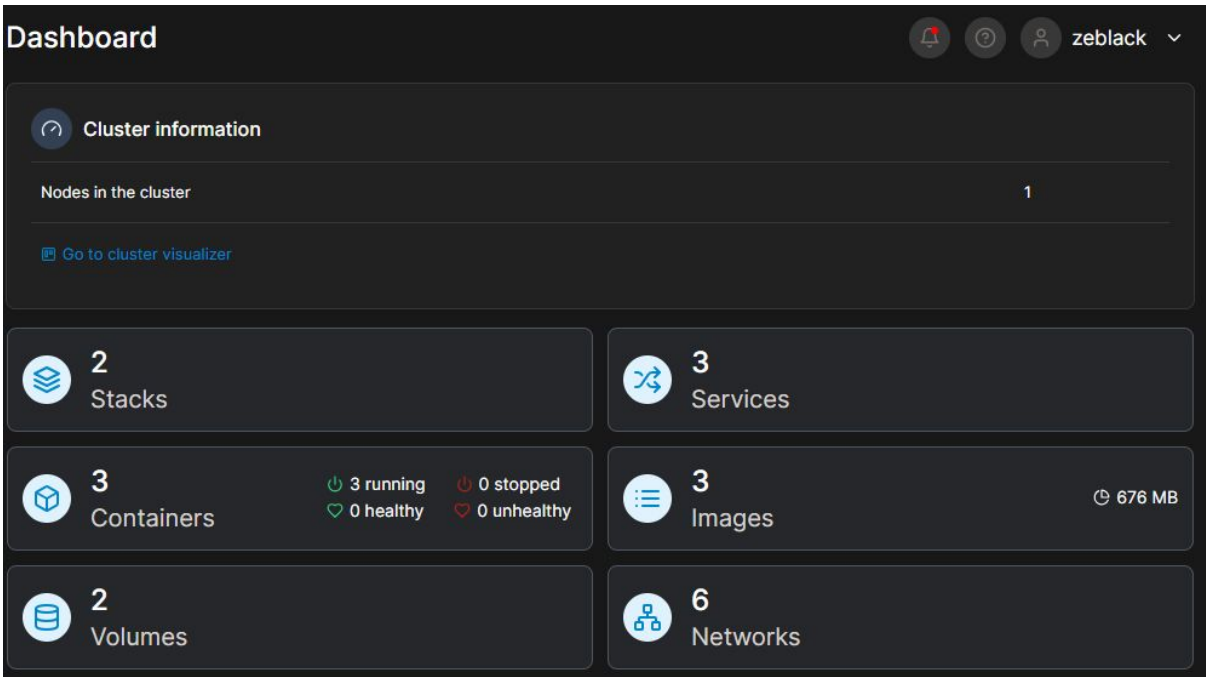
CONTAINER ID	IMAGE	NAMES	COMMAND	CREATED	STATUS	PORTS
f67af6423db	portainer/portainer-ce:2.21.4	portainer	"/portainer -H tcp://..."	16 hours ago	Up 16 hours	8000/tcp, 9000/tcp, 9443/tcp
45595c3e893d	portainer/agent:2.21.4	portainer_agent	"./agent"	16 hours ago	Up 16 hours	

Fonte:Bitvise(2024).

A partir deste ponto, o restante da estrutura será montado através do Portainer, acessado pelo subdomínio configurado nas etapas anteriores.

Informa-se ao leitor que o processo se torna mais fácil a partir de agora, pois a interface gráfica do Portainer torna as etapas mais visuais e intuitivas.

FIGURA 19 – Dashboard Portainer

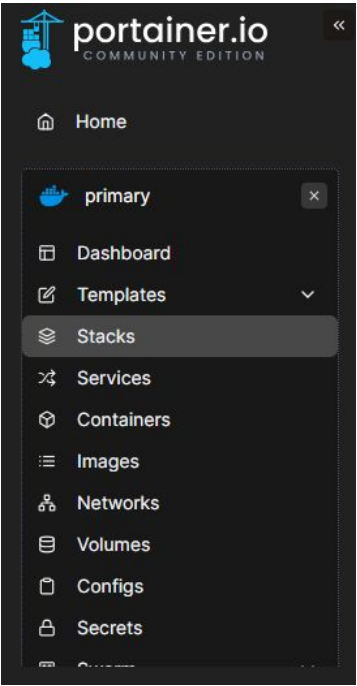


Fonte:Portainer(2024).

Agora serão montadas as stacks do Redis e Postgres, que servirão como banco de dados para as aplicações.

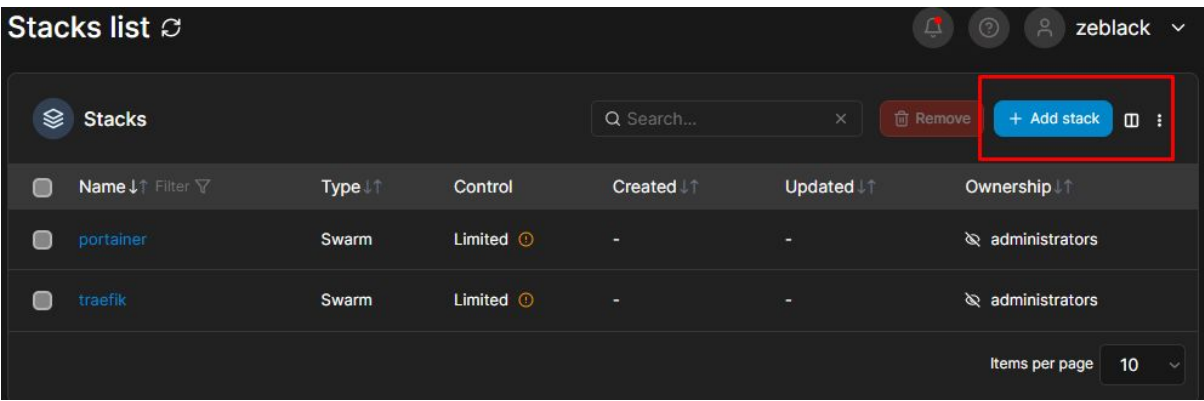
Para configurar e instalar uma nova stack, acesse o menu, clique em "Stack" e, em seguida, selecione a opção "Add Stack", conforme demonstrado nas imagens.

FIGURA 20– Menu Portainer



Fonte:Portainer(2024).

FIGURA 21– Lista de Stacks

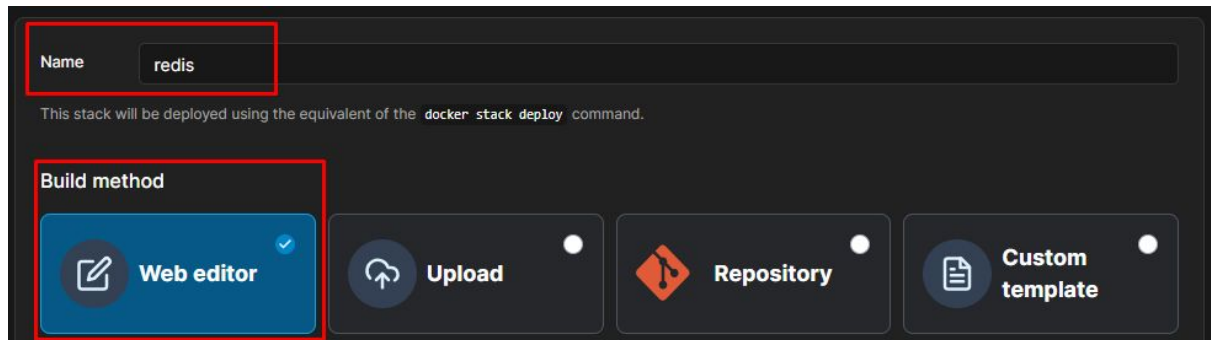


Fonte:Portainer(2024).

Criação de Stack:

Nomeie a stack e selecione o método de instalação, conforme demonstrado na imagem de referência.

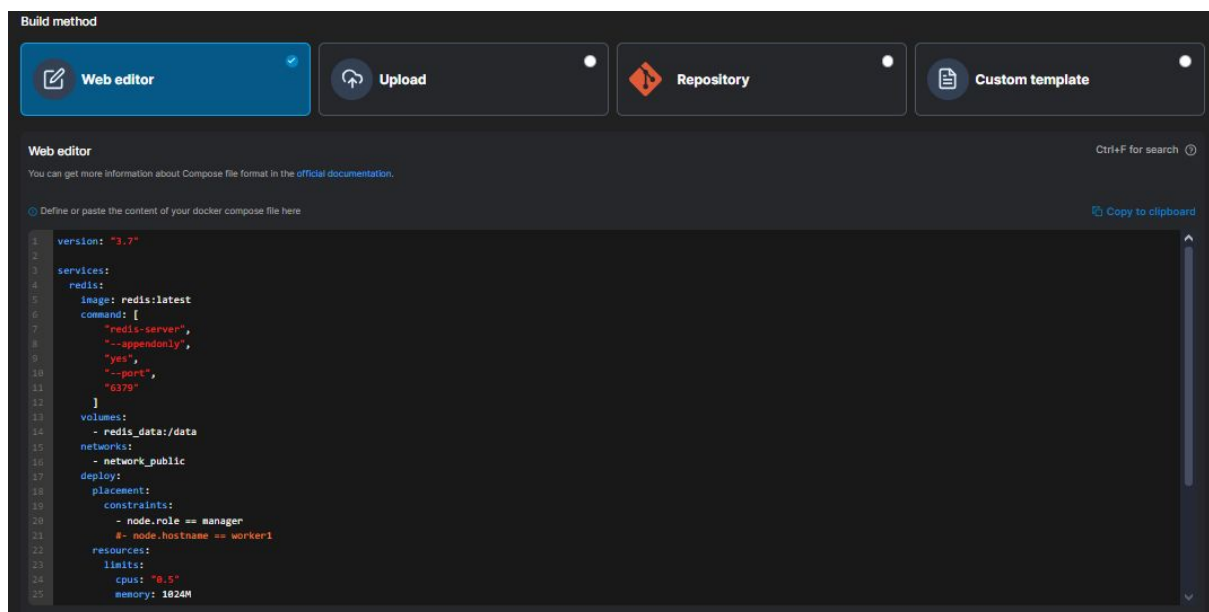
FIGURA 22 – Criação Stack redis



Fonte:Portainer(2024).

Em seguida, abra o arquivo da stack Redis no Sublime Text. Copie o conteúdo do código e cole-o no campo Web Editor.

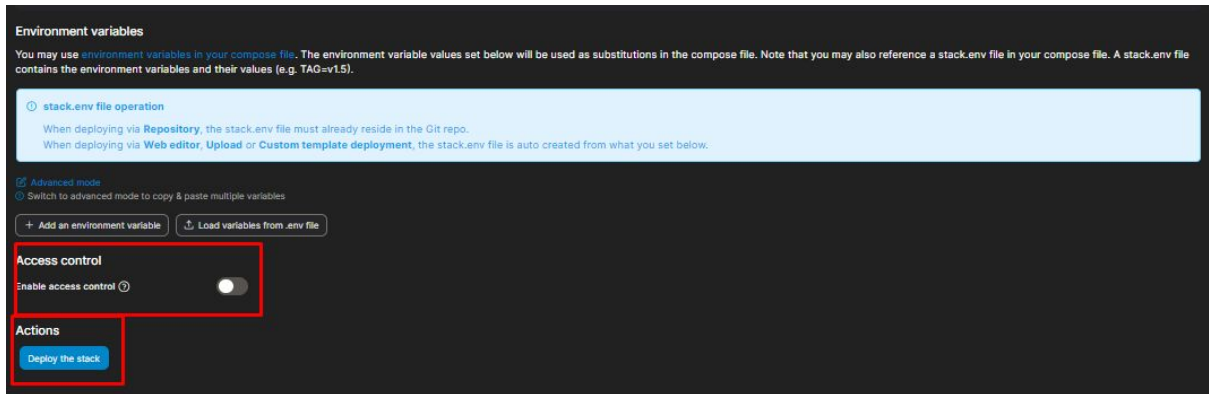
FIGURA 23 – Stack redis



Fonte:Portainer(2024).

Após colar o código no campo Web Editor, prossiga com o deploy da stack: Desative a opção Access Control e clique em Deploy the Stack.

FIGURA 24 – Deploy Stack redis



Fonte:Portainer(2024).

Este passo a passo será o mesmo para a adição de todas as stacks do projeto. Sempre que houver referência à instalação e configuração de uma stack, o mesmo procedimento deverá ser realizado.

Por se tratar de uma prática padrão e repetitiva, as imagens de cada etapa não serão apresentadas. O foco será direcionado aos detalhes específicos e à construção do agente.

Todos os arquivos das stacks e os códigos utilizados estarão disponíveis em anexo.

Próxima Stack:

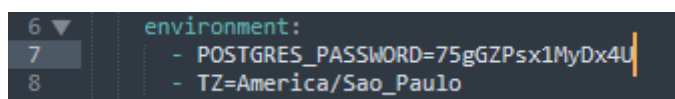
Abra o arquivo da stack do Postgres no Sublime Text.

Após copiar e colar o código no campo Web Editor, altere a senha do banco de dados na linha 7.

Desative a opção de Access Control.

Clique em Deploy the Stack para realizar o deploy.

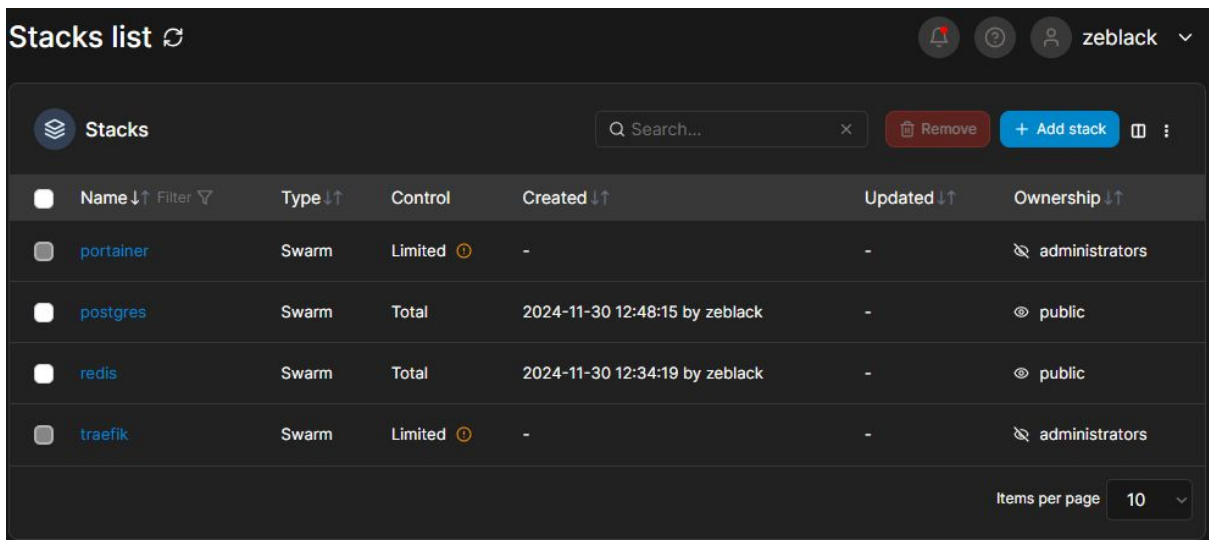
FIGURA 25 – Ajuste Stack Postgres



Fonte:Sublimetext(2024).

Neste momento, o status das stacks configuradas é o seguinte:

FIGURA 26 – Lista de Stacks

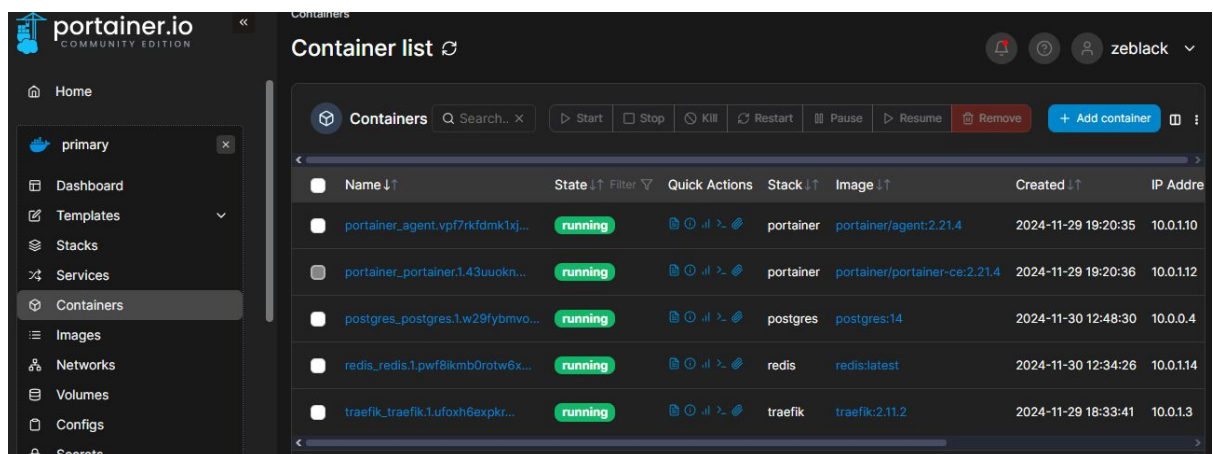


Name	Type	Control	Created	Updated	Ownership
portainer	Swarm	Limited	-	-	administrators
postgres	Swarm	Total	2024-11-30 12:48:15 by zeblack	-	public
redis	Swarm	Total	2024-11-30 12:34:19 by zeblack	-	public
traefik	Swarm	Limited	-	-	administrators

Fonte:Portainer(2024).

Acesse a seção Containers no Portainer para acompanhar o status e funcionamento de cada container.

FIGURA 27 – Lista de Containers



Name	State	Quick Actions	Stack	Image	Created	IP Address
portainer_agent.vpf7rkfmdmk1xj...	running	[Icons]	portainer	portainer/agent:2.21.4	2024-11-29 19:20:35	10.0.1.10
portainer_portainer.1.43uokn...	running	[Icons]	portainer	portainer/portainer-ce:2.21.4	2024-11-29 19:20:36	10.0.1.12
postgres_postgres.1.w29fybmvo...	running	[Icons]	postgres	postgres:14	2024-11-30 12:48:30	10.0.0.4
redis_redis.1.pwf8ikmb0rotw6x...	running	[Icons]	redis	redis:latest	2024-11-30 12:34:26	10.0.1.14
traefik_traefik.1.ufoxh6expkr...	running	[Icons]	traefik	traefik:2.11.2	2024-11-29 18:33:41	10.0.1.3

Fonte:Portainer(2024).

Próxima Stack:

Será realizada a instalação e configuração da stack da Evolution API, uma API alternativa responsável pela comunicação entre os agentes no Dify e o WhatsApp. Essa integração é feita de forma nativa entre as plataformas, o que facilita a comunicação e o funcionamento conjunto.

Alterações na Stack da Evolution API:

Abra o arquivo da stack da Evolution API utilizando o Sublime Text.

Realize as seguintes alterações no arquivo:

Adicione o subdomínio correspondente à configuração realizada na etapa anterior no Cloudflare. No caso do protótipo, será utilizado: **evoapi.ofertadoze.com.br**.

Esse subdomínio será o caminho utilizado para acessar a plataforma após a instalação.

FIGURA 28 – Configuração Stack Evolution Api

```
10 environment:
11   ## Url da Evolution API
12   - SERVER_URL=https://seudominio.com.br
13   ## Dados de Autenticação
```

Fonte:Sublime text(2024).

Repita Linha 142:

FIGURA 29 – Configuração Stack Evolution Api

```
140 labels:
141   - traefik.enable=1
142   - traefik.http.routers.evolution_v2.rule=Host(`evoapi.ofertadoze.com.br`) ## Url da aplicação
143   - traefik.http.routers.evolution_v2.entrypoints=websecure
```

Fonte:Sublime text(2024).

Criação da Chave de Acesso:

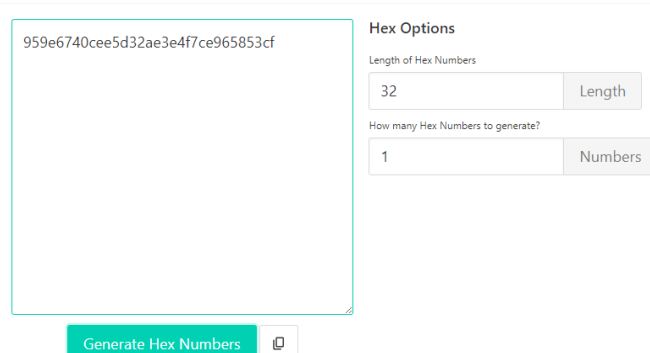
Será necessário criar uma chave de acesso segura, utilizando uma senha longa e forte.

Para facilitar a criação da senha, recomenda-se o uso do seguinte site gerador de chaves:

<https://codebeautify.org/generate-random-hexadecimal-numbers>

Copie a chave gerada e utilize-a conforme necessário na configuração

FIGURA 30 – Gerador de Key



Fonte:codebeautifyt(2024).

Após gerar a chave de acesso (key), copie-a e cole-a na linha 15 do arquivo da stack dá Evolution API.

FIGURA 31 – Configuração key Stack Evolution Api

```
10 environment:
11   ## Url da Evolution API
12   - SERVER_URL=evoapi.ofertadoze.com.br
13   ## Dados de Autenticação
14   - AUTHENTICATION_TYPE=apikey
15   - AUTHENTICATION_API_KEY=Sua Key Aqui ## GLOBAL API KEY
16   - AUTHENTICATION_EXPOSE_IN_FETCH_INSTANCES=true
17   ## Configurações
18   LANGUAGE=pt-BR
```

Fonte:Sublime

text(2024).

Certifique-se de salvar a chave gerada, pois ela será utilizada para acessar a plataforma. Na linha 35 do arquivo da Stack dá Evolution API, altere a senha do banco de dados para a senha definida no Postgres.

FIGURA 32 – Configuração Banco de dados Stack Evolution Api

```
32 ## Aivar Banco de Dados Postgres
33 - DATABASE_ENABLED=true
34 - DATABASE_PROVIDER=postgresql
35 - DATABASE_CONNECTION_URI=postgresql://postgres:75gGZPsx1MyDx4U@postgres:5432/evolution
36 - DATABASE_CONNECTION_CLIENT_NAME=evolution
```

Fonte:Sublime text(2024).

Repita linha 70:

FIGURA 33 – Configuração Banco de dados Stack Evolution Api

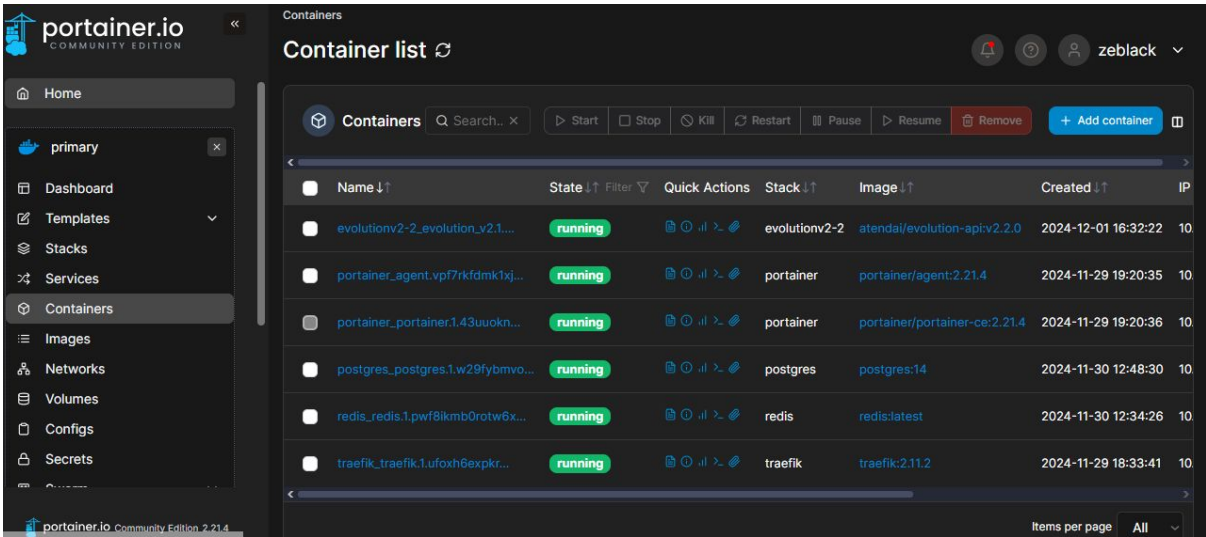
```
66 ## Carregar módulo Chatwoot
67 - CHATWOOT_ENABLED=true
68 - CHATWOOT_MESSAGE_READ=true
69 - CHATWOOT_MESSAGE_DELETE=true
70 - CHATWOOT_IMPORT_DATABASE_CONNECTION_URI=postgresql://postgres:75gGZPsx1MyDx4U@postgres:5432/chatwoot?sslmode=disable
71 - CHATWOOT_IMPORT_PLACEHOLDER_MEDIA_MESSAGE=false ## true = Importar mídia | false = Não importar mídia
72 ## Dados do RabbitMQ
```

Fonte:Sublime text(2024).

Após realizar todas as modificações necessárias, prossiga com o deploy da stack da Evolution API. O passo a passo será o mesmo já explicado anteriormente no projeto. Siga o exemplo utilizado na instalação da stack do Redis, ajustando o nome e

utilizando o código da Evolution API. Aguarde cerca de 3 minutos após o deploy e, em seguida, acesse a seção Containers no Portainer para verificar o status da instalação.

FIGURA 34 – Lista de Containers



Fonte:Portainer(2024).

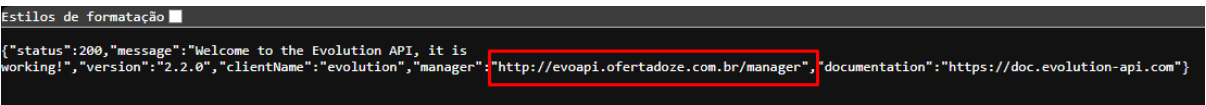
Com todas as configurações realizadas, a ferramenta Evolution API estará pronta para ser acessada.

No navegador, insira o subdomínio configurado. No caso do protótipo, utilize:

<https://evoapi.ofertadoze.com.br>

A página da web retornará o link de acesso à plataforma.

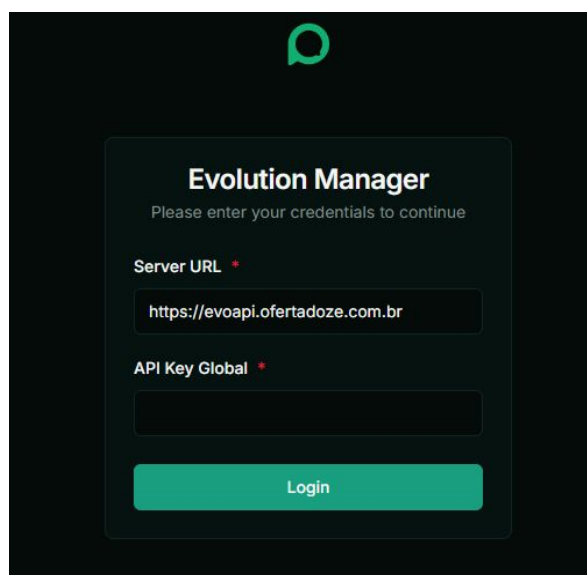
FIGURA 35 – Resposta Web



Fonte:<https://evoapi.ofertadoze.com.br>(2024).

Tela inicial da ferramenta:

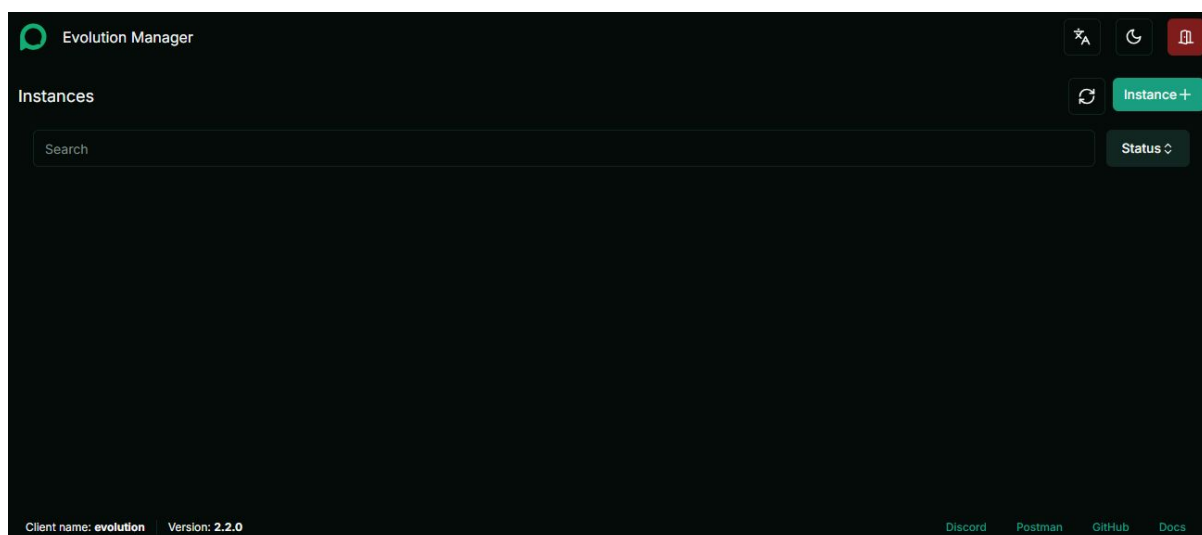
FIGURA 36 – Tela de Login Evolution



Fonte: <http://evoapi.ofertadoze.com.br/manager>(2024).

A API Key será a chave gerada e inserida durante a etapa de configuração da stack.

FIGURA 37 – Tela Após Login Evolution



Fonte: <http://evoapi.ofertadoze.com.br/manager>(2024).

Essa ferramenta é essencial para o gerenciamento e a comunicação com os números de WhatsApp.

Dando continuidade à construção do agente inteligente, será iniciada a instalação das stacks e ferramentas principais para sua configuração. Para isso, será utilizado o Dify para a criação e gerenciamento do agente, em conjunto com o MinIO, um

sistema de armazenamento que oferece suporte à funcionalidade de RAG (Recuperação de Informação com Geração).

É importante que o leitor preste atenção especial a partir deste ponto, pois as configurações atingem um nível maior de complexidade. Caso ocorra algum erro durante a execução do passo a passo, recomenda-se revisar o tutorial e consultar material complementar atualizado, disponível no YouTube.

Etapa adicional:

Será necessário retornar às etapas iniciais para adicionar novos subdomínios no Cloudflare.

FIGURA 38 – Minio!: minio.ofertadoze.com.br

minio.ofertadoze.com.br é um alias de admin.ofertadoze.com.br.

The image shows a Cloudflare DNS record configuration interface. It includes a 'Tipo' dropdown set to 'CNAME', a 'Nome' field with 'minio', a 'Destino' field with 'admin.ofertadoze', a 'Status do proxy' toggle set to 'Somente DNS', and a 'TTL' dropdown set to 'Auto'. Below the 'Nome' field is a note 'Use @ para raiz'. Below the 'Destino' field is an example 'Por exemplo: www.ex ample.com'.

Fonte: Cloudflare(2024).

FIGURA 39 – s3: s3.ofertadoze.com.br

s3.ofertadoze.com.br é um alias de admin.ofertadoze.com.br.

The image shows a Cloudflare DNS record configuration interface. It includes a 'Tipo' dropdown set to 'CNAME', a 'Nome' field with 's3', a 'Destino' field with 'e.com.br', a 'Status do proxy' toggle set to 'Somente DNS', and a 'TTL' dropdown set to 'Auto'. Below the 'Nome' field is a note 'Use @ para raiz'. Below the 'Destino' field is an example 'Por exemplo: www.ex ample.com'.

Fonte: Cloudflare(2024).

FIGURA 40 – dify: zeblack.dify.ofertadoze.com.br

zeblack.dify.ofertadoze.com.br é um alias de admin.ofertadoze.com.br.

Tipo	Nome (necessário)	Destino (necessário)	Status do proxy	TTL
CNAME	zeblack.dify Use @ para raiz	e.com.br Por exemplo: www.ex ample.com	☐ Somente DNS	Auto

Fonte: Cloudflare(2024).

Após criar os subdomínios necessários, abra o arquivo da stack do MinIO utilizando o Sublime Text e realize as seguintes alterações:

Linha 21: Insira uma senha root para o acesso ao MinIO.

Linha 23: Adicione o subdomínio para o front-end, que será utilizado para acesso web.

Linha 25: Adicione o subdomínio para a API do MinIO, responsável pelos serviços back-end.

FIGURA 41– Configuração Stack Minio

```
17 ▼ environment:
18   # Define o usuario root do Minio
19   - MINIO_ROOT_USER=admin
20   # Define a senha root do Minio
21   - MINIO_ROOT_PASSWORD=adicionesusenha
22   # Define o endereço do console do Minio
23   - MINIO_BROWSER_REDIRECT_URL=https://minio.ofertadoze.com.br
24   # Define o endereço público do Minio
25   - MINIO_SERVER_URL=https://s3.ofertadoze.com.br
```

Fonte:Sublime text(2024).

Altere os domínios nas linhas 39 e 51

FIGURA 42 – Configuração Stack Minio

```

35 labels:
36   # Habilita o Traefik
37   - traefik.enable=true
38   # Define o endereço publico do Minio
39   - traefik.http.routers.minio_public.rule=Host(`s3.ofertadoze.com.br`)
40   # Redireciona o endereço para HTTPS
41   - traefik.http.routers.minio_public.entrypoints=websecure
42   # Define o certificado SSL
43   - traefik.http.routers.minio_public.tls.certresolver=letsencryptresolver
44   # Define a porta do servidor publico do Minio
45   - traefik.http.services.minio_public.loadbalancer.server.port=9000
46   # Define os Headers que o Minio vai usar
47   - traefik.http.services.minio_public.loadbalancer.passHostHeader=true
48   # Define o serviço que irá ser usado
49   - traefik.http.routers.minio_public.service=minio_public
50   # Define o endereço do console do Minio
51   - traefik.http.routers.minio_console.rule=Host(`minio.ofertadoze.com.br`)
52   # Redireciona o endereço para HTTPS
53   - traefik.http.routers.minio_console.entrypoints=websecure
54   # Define o certificado SSL

```

Fonte: Sublimetext(2024).

Após realizar as alterações na stack, copie e cole o conteúdo no Portainer, configurando uma nova stack da mesma forma que foi feito nas etapas anteriores.

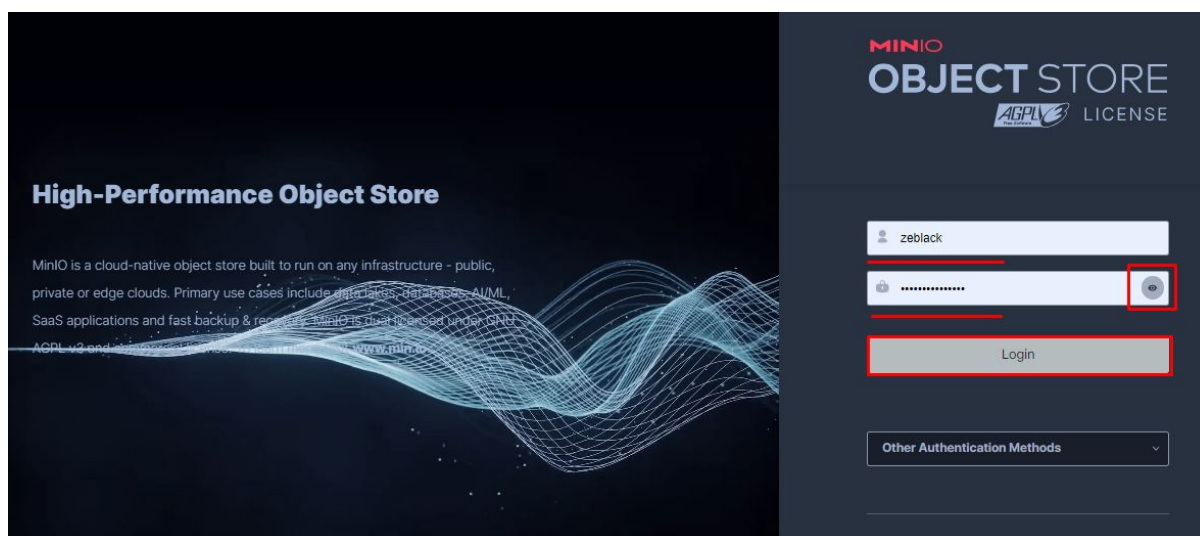
Acesso ao MinIO:

Após o deploy, acesse a interface web do MinIO utilizando o subdomínio configurado, por exemplo:

<https://minio.ofertadoze.com.br>

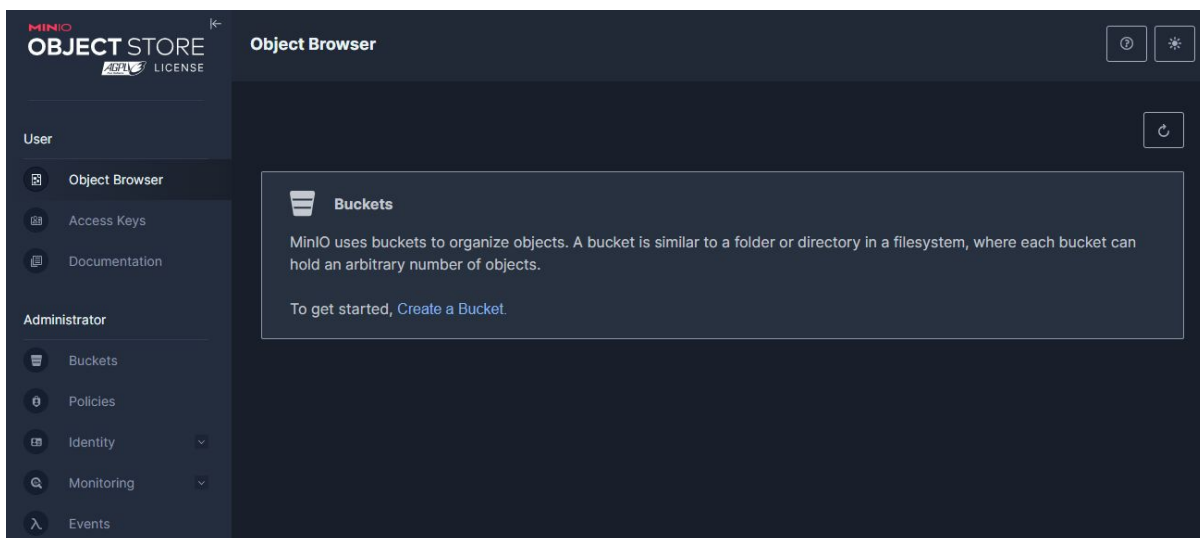
Faça login utilizando as credenciais configuradas na stack.

FIGURA 43 – Tela de login Minio



Fonte: <https://minio.ofertadoze.com.br>(2024).

FIGURA 44 – Tela Após login Minio



Fonte: [https://minio.ofertadoze.com.br\(2024\)](https://minio.ofertadoze.com.br(2024)).

Antes de proceder com a instalação do Dify, é necessário preparar o servidor através da linha de comando para garantir que a stack seja inicializada corretamente. Caso este passo a passo não seja seguido, podem ocorrer erros durante a instalação.

Passos a serem realizados:

Acesse o servidor utilizando o Bitvise e abra o terminal.

Execute o seguinte comando para copiar os arquivos padrões necessários para a instalação do Dify:

```
git clone https://github.com/langgenius/dify.git
```

Nota:

Embora o Traefik esteja sendo utilizado para proxy reverso e SSL, o Dify requer o Nginx como complemento para algumas dependências essenciais ao seu funcionamento correto.

FIGURA 45 – Comando git clone no terminal

```
*** System restart required ***
Last login: Sat Nov 30 14:09:00 2024 from 200.225.115.43
root@TCC:~# git clone https://github.com/langgenius/dify.git
```

Fonte: Bitvise(2024).

Após o comando de clonagem do repositório, execute o seguinte comando para

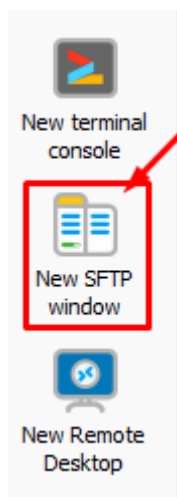
mover a pasta nginx para a raiz do servidor, conforme solicitado na documentação de instalação:

sudo mv /root/dify/docker/nginx /root/nginx

Verificação via FTP:

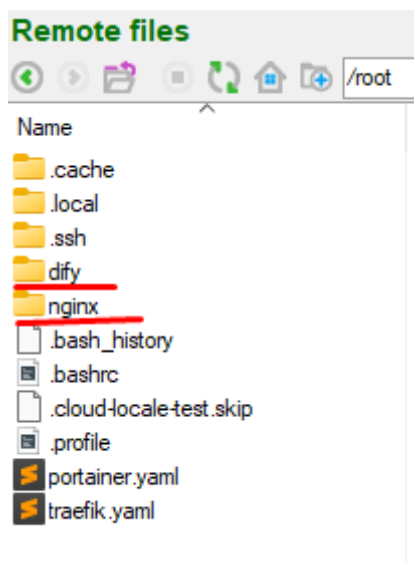
Utilizando a função FTP do Bitvise, é possível verificar o resultado da operação. A pasta nginx será exibida na raiz do servidor, confirmando que as dependências foram baixadas corretamente a partir do repositório Git e posicionadas conforme especificado.

FIGURA 46 – Função SFTP Bitvise



Fonte:Bitvise(2024).

FIGURA 47 – Tela acesso via SFTP Bitvise



Fonte:Bitvise(2024).

Faça login no MinIO utilizando o subdomínio configurado:

<https://minio.ofertadoze.com.br/login>

Após o login, navegue até a interface e crie um bucket para armazenar os dados necessários ao projeto.

FIGURA

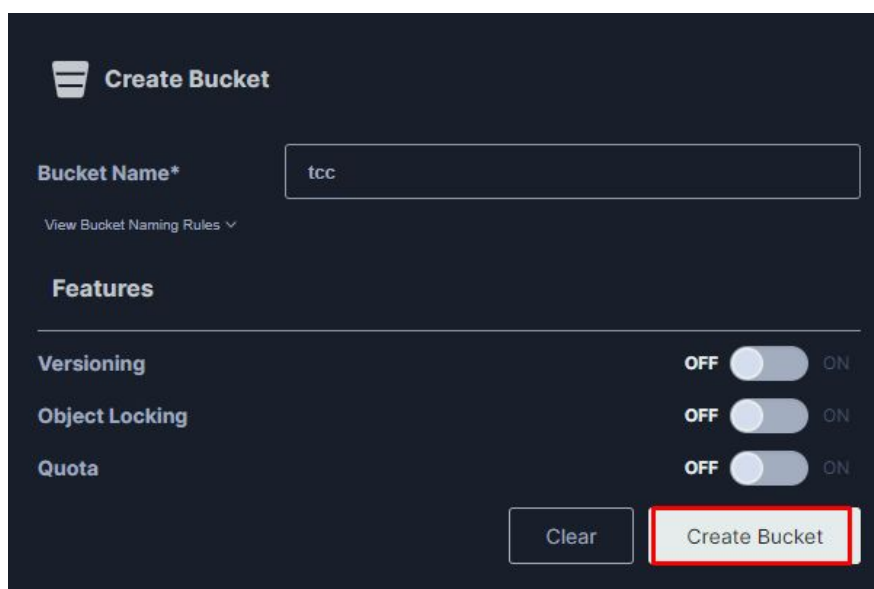
48

–

Tela

Criação

Bucket

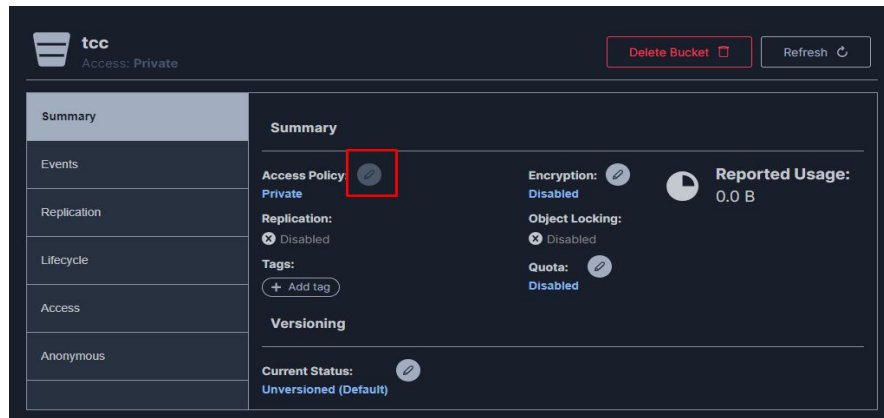


Fonte:<https://minio.ofertadoze.com.br>(2024).

Acesse o bucket recém-criado no MinIO e realize as alterações necessárias nas políticas de acesso.Essas modificações são fundamentais para configurar as

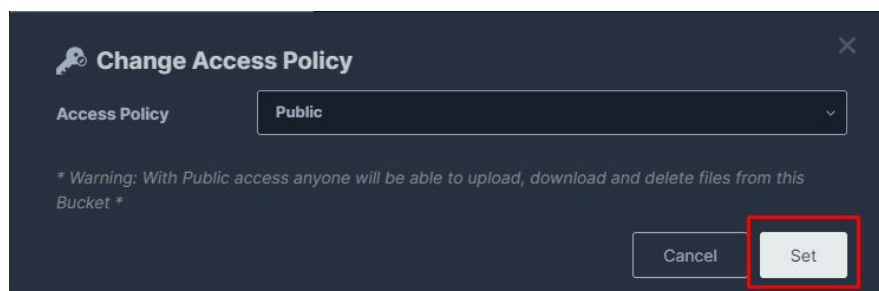
permissões adequadas, garantindo o gerenciamento eficaz e o acesso seguro aos dados armazenados no bucket.

FIGURA 49 – Tela dentro do Bucket



Fonte: <https://minio.ofertadoze.com.br> (2024).

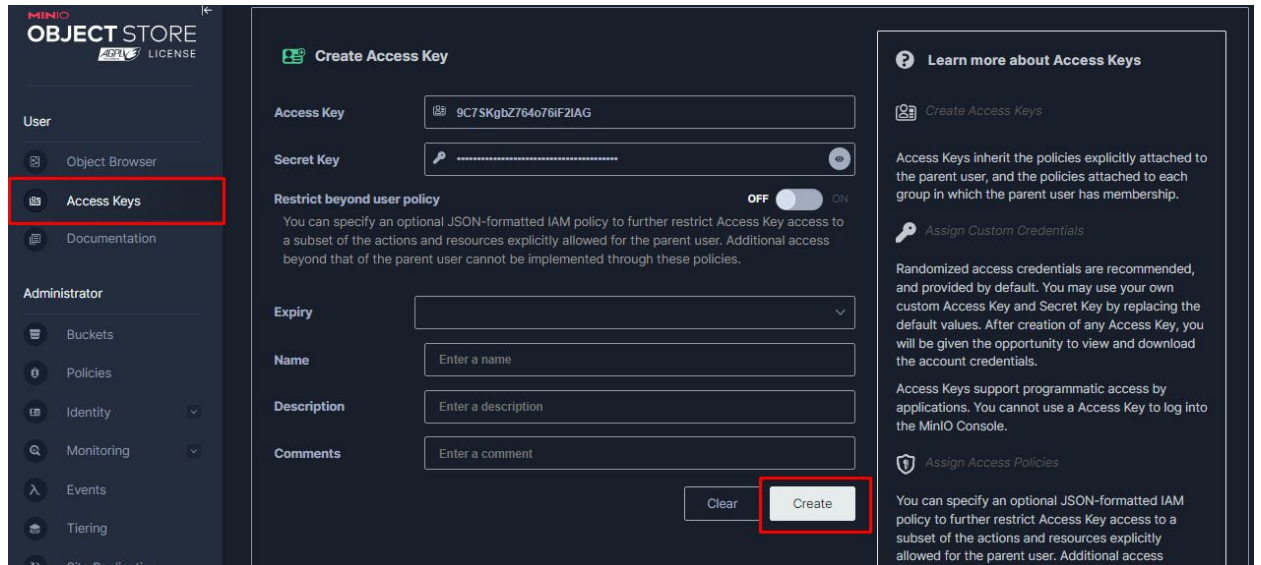
FIGURA 50– Tela alterando política de acesso



Fonte: <https://minio.ofertadoze.com.br> (2024).

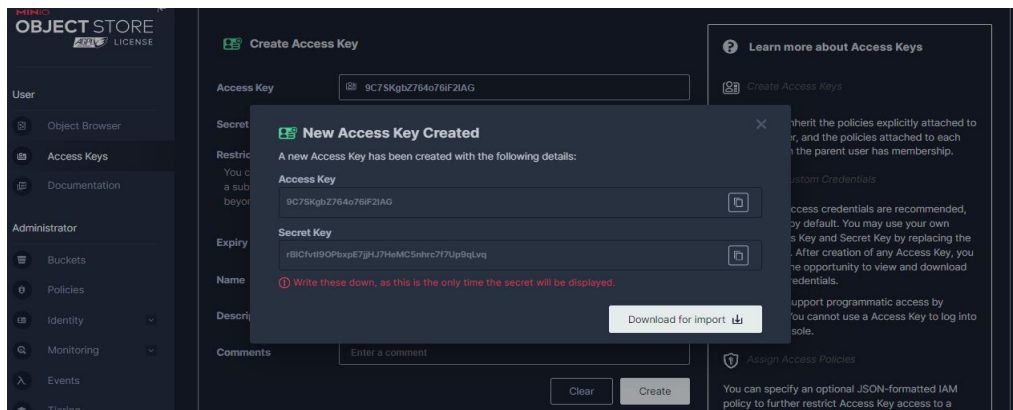
Após salvar as alterações nas políticas de acesso do bucket, prossiga com a criação das chaves de acesso. Essas chaves serão utilizadas posteriormente na configuração da stack do Dify.

FIGURA 51– Tela Criando nossas Chaves de acesso



Fonte: [https://minio.ofertadoze.com.b\(2024\).](https://minio.ofertadoze.com.b(2024).)

FIGURA 52 – Tela Chaves de acesso criadas

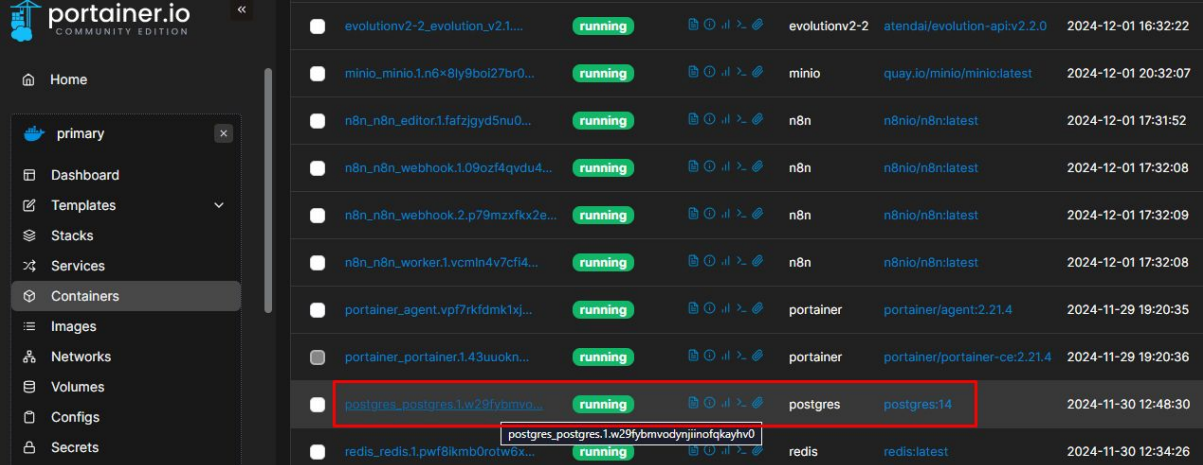


Fonte: <https://mi>

[nio.ofertadoze.com.b\(2024\).](https://minio.ofertadoze.com.b(2024).)

Agora será criado o banco de dados do Dify no Postgres. Para isso, acesse o Portainer, vá até a seção Containers e selecione o container do Postgres.

FIGURA 53 – Lista de container

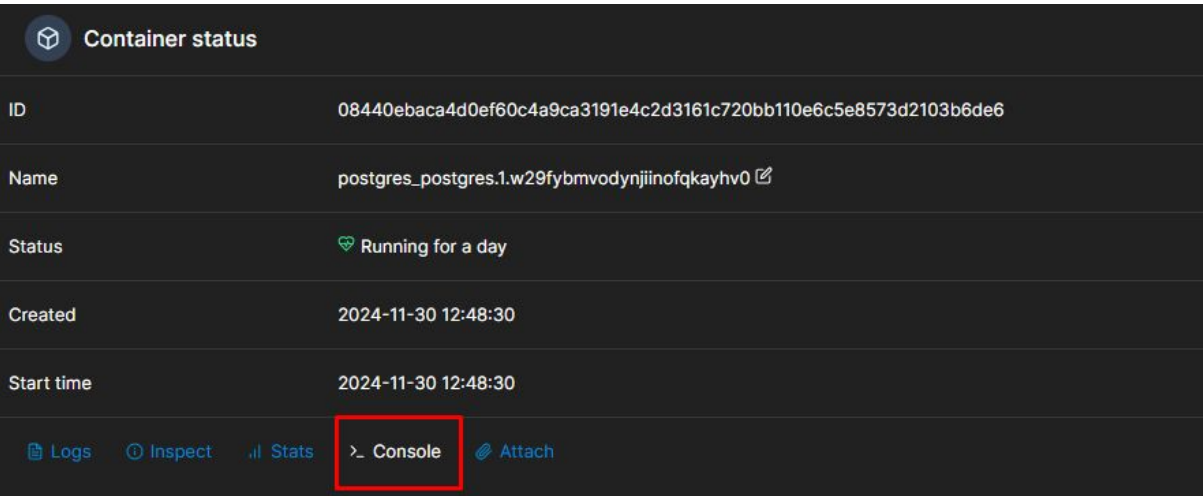


ID	Status	Image	Created
evolutionv2-2_evolution_v2.1...	running	atenda/evolution-api:v2.2.0	2024-12-01 16:32:22
minio_minio.1.n6x8ly9boi27br0...	running	quay.io/minio/minio:latest	2024-12-01 20:32:07
n8n_n8n_editor.1.fafzjgyd5nu0...	running	n8nio/n8n:latest	2024-12-01 17:31:52
n8n_n8n_webhook.1.09ozf4qvdu4...	running	n8nio/n8n:latest	2024-12-01 17:32:08
n8n_n8n_webhook.2.p79mzxfkx2e...	running	n8nio/n8n:latest	2024-12-01 17:32:09
n8n_n8n_worker.1.vcm1n4v7cfi4...	running	n8nio/n8n:latest	2024-12-01 17:32:08
portainer_agent.vpf7rkfdmk1xj...	running	portainer/portainer-agent:2.21.4	2024-11-29 19:20:35
portainer_portainer.1.43uokn...	running	portainer/portainer-ce:2.21.4	2024-11-29 19:20:36
postgres_postgres.1.w29fybmvdynjiinofqkayhv0	running	postgres:14	2024-11-30 12:48:30
redis_redis.1.pwf8ikmb0rotw6x...	running	redis:latest	2024-11-30 12:34:26

Fonte:Portainer(2024).

Selecione o container do Postgres no Portainer. Na tela que será exibida, clique na opção console para acessar o terminal do container.

FIGURA 54 – Container Postgres



Container status	
ID	08440ebaca4d0ef60c4a9ca3191e4c2d3161c720bb110e6c5e8573d2103b6de6
Name	postgres_postgres.1.w29fybmvdynjiinofqkayhv0
Status	Running for a day
Created	2024-11-30 12:48:30
Start time	2024-11-30 12:48:30
Logs Inspect Stats Console Attach	

Fonte:Portainer(2024).

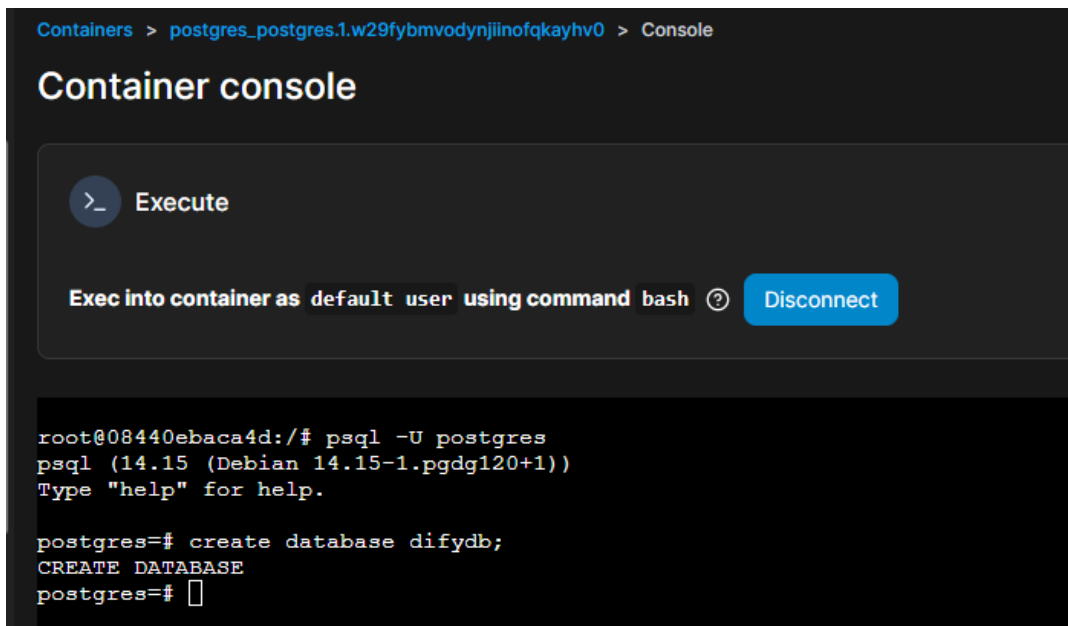
No console do container do Postgres, crie o banco de dados para o Dify. Utilize os comandos adequados para a criação.

Comandos:

psql -U postgres

create database difydb;

FIGURA 55 – Container Console



Fonte:Portainer(2024).

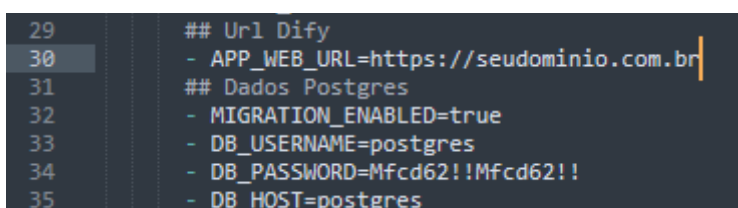
Com o banco de dados e as credenciais criadas, retorne ao arquivo da stack do Dify no Sublime Text para finalizar a configuração. Realize as seguintes alterações:

Linha 30, 145, 193: Substitua pelo subdomínio configurado, por exemplo: zeblack.dify.ofertadoze.com.br.

Linha 34, 96: Insira a senha definida para o banco de dados Postgres.

Após realizar as modificações, salve o arquivo e prossiga com o deploy da stack para subir a aplicação.

FIGURA 56 – Configuração Stack Dify



Fonte:Sublime text(2024).

Linha 49 até 55 configuração Minio, feito na etapa anterior

FIGURA 57– Configuração s3 Stack Dify

```
49     ## Dados sobre armazenamento s3
50     - STORAGE_TYPE=local
51     - STORAGE_LOCAL_PATH=storage
52     - S3_ENDPOINT=s3.ofertadoze.com.br
53     - S3_BUCKET_NAME=tcc
54     - S3_ACCESS_KEY=hoUqSSzjyy5Q40JncFeX
55     - S3_SECRET_KEY=qb5D6B5J9ouNXCyI18YAaWpNSK6jRfT08fQu1UyR
```

Fonte:Sublime text(2024).

Leia atentamente todo o código da stack, identificando as opções repetidas que precisam ser modificadas ao longo do arquivo. Certifique-se de realizar todas as alterações necessárias antes de prosseguir.

Após verificar e ajustar a stack, acesse o **Portainer** e faça o deploy da stack configurada.

Atenção após o deploy:

- Aguarde entre **3 a 8 minutos** para que o sistema se torne totalmente operacional.
- Caso o sistema não esteja acessível, vá até a seção **Containers** no Portainer e verifique o status.
- Se houver algum container em vermelho, exclua-o e aguarde que ele seja reiniciado corretamente.

FIGURA 58– Verificação de Containers após instalação dify



Fonte:Portainer(2024).

Após realizar o deploy e verificar que todos os containers estão funcionando corretamente, acesse a ferramenta via web utilizando o subdomínio configurado para o Dify.

Finalize a instalação através da interface web, seguindo os passos padrões das outras stacks.

Quando a instalação for concluída com sucesso, a tela inicial da ferramenta será exibida, indicando que o sistema está pronto para uso.

FIGURA 59– Tela de Login dify.



Fonte:[https://zeblack.dify.ofertadoze.com.br\(2024\).](https://zeblack.dify.ofertadoze.com.br(2024).)

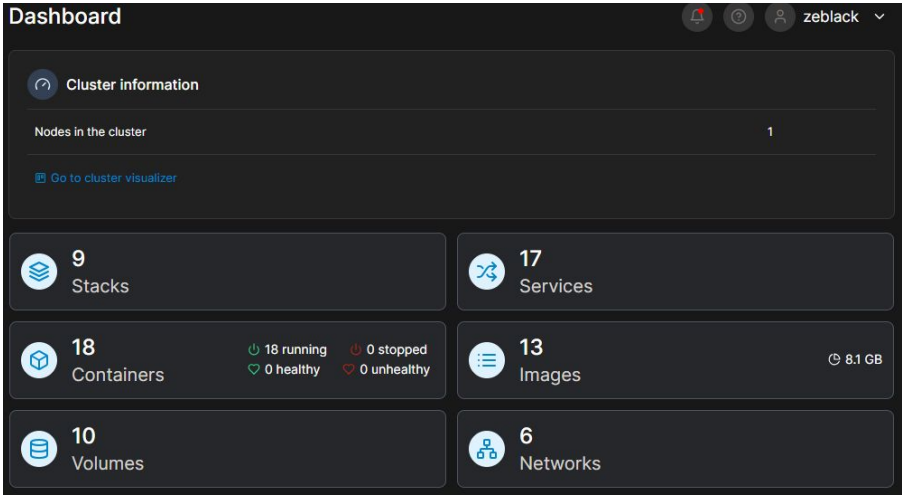
FIGURA 60– Tela após Login dify.



Fonte:[https://zeblack.dify.ofertadoze.com.br\(2024\).](https://zeblack.dify.ofertadoze.com.br(2024).)

Status da Infraestrutura até o Momento

FIGURA 61– Tela Dashboard Status Cluster



Fonte:Portainer(2024).

FIGURA 62– subdomínios configurados

ofertadoze.com.br		Ativo	Asterisco	Plano Free			
☐	Tipo ①	Nome ①	Conteúdo ①	Status do proxy ①	TTL ①	Ações	
☐	A	admin	188.245.155.65	Somente DNS	Auto	Editar ▶	
☐	CNAME	chatw	admin.ofertadoze.c...	Somente DNS	Auto	Editar ▶	
☐	CNAME	evoapi	admin.ofertadoze.c...	Somente DNS	Auto	Editar ▶	
☐	CNAME	minio	admin.ofertadoze.c...	Somente DNS	Auto	Editar ▶	
☐	CNAME	n8webhook	admin.ofertadoze.c...	Somente DNS	Auto	Editar ▶	
☐	CNAME	n8workflow	admin.ofertadoze.c...	Somente DNS	Auto	Editar ▶	
☐	CNAME	portainer	admin.ofertadoze.c...	Somente DNS	Auto	Editar ▶	
☐	CNAME	s3	admin.ofertadoze.c...	Somente DNS	Auto	Editar ▶	
☐	CNAME	zeblack.dify	admin.ofertadoze.c...	Somente DNS	Auto	Editar ▶	

Fonte: Cloudflare(2024).

Para leitores com pouca experiência em códigos, estão sendo disponibilizados dois autos instaladores amplamente utilizados no mundo open source. Esses instaladores facilitam o processo de configuração e implementação, tornando-o mais acessível e intuitivo.

SeutpOrion: <https://oriondesign.art.br>

Easypanel: <https://easypanel.io>

Com as etapas de instalação e configuração das stacks concluídas, será iniciada a criação do Zeblack, o agente inteligente para verificação de informações e notícias em tempo real.

O Dify será o componente central, funcionando como o "coração e cérebro" do agente. Siga o passo a passo descrito nas imagens para configurar o agente com precisão.

FIGURA

63–

Criação

agente

Dify



Fonte: [https://zeblack.dify.ofertadoze.com.br\(2024\)](https://zeblack.dify.ofertadoze.com.br(2024)).

Escolha a opção "Agente" no menu do Dify.

Dê um nome e uma descrição ao agente que será criado.

Após preencher as informações, clique em "Criar" para finalizar essa etapa.

FIGURA 64 – Criação agente Dify

Criar do zero

Que tipo de aplicativo você deseja criar?

Chatbot Gerador de Texto **Agente** Fluxo de trabalho BETA

Ícone e nome do aplicativo

 **Zeblack**

Descrição

zeblack agente inteligente de verificação de informações e notícias em tempo real.

Cancelar Criar

Fonte: [https://zeblack.dify.ofertadoze.com.br\(2024\)](https://zeblack.dify.ofertadoze.com.br(2024)).

Na tela de criação e configuração do agente, será possível trabalhar com diversas configurações e ferramentas, incluindo a integração com a API. Essa etapa é fundamental para personalizar o funcionamento do agente e integrar os recursos necessários para atender às demandas do projeto.

FIGURA 65 – Tela de principal de trabalho dify

Dify Star 53.471 Explorar **Estúdio / Zeblack** Conhecimento Ferramentas z zeblack

Zeblack AGENTE

Orquestrar

Acesso à API

Logs e Anúncios

Monitoramento

Orquestrar

Instruções Automático

Escreva sua palavra de incentivo aqui, digite " " para inserir uma variável, digite " " para inserir um bloco de c...

0

(x) Variáveis Adicionar

As variáveis permitem que os usuários introduzam palavras de solicitação ou observações iniciais ao preencher formulários. Você pode tentar digitar "[input]" nas palavras de solicitação.

Contexto Adicionar

Você pode importar Conhecimento como contexto

Ferramentas 0/0 Habilitado Adicionar

Configurações do Agente gpt-4o-audio-preview Publicar

A chave do provedor LLM não foi definida

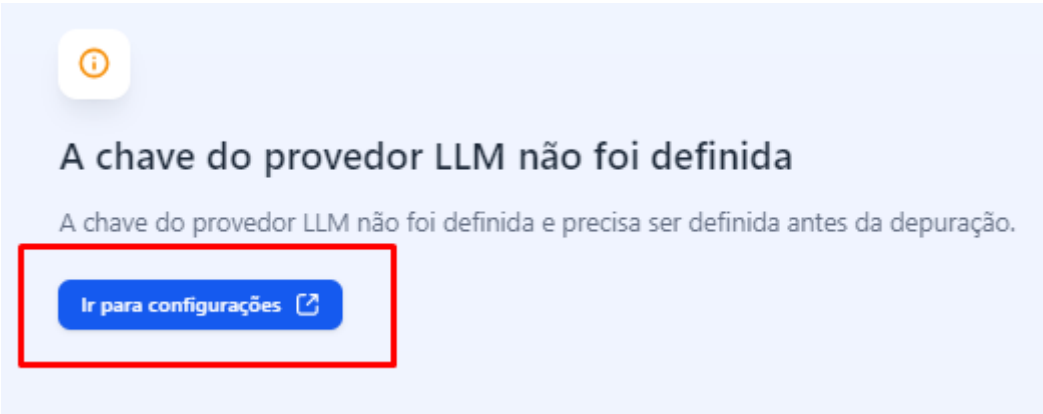
A chave do provedor LLM não foi definida e precisa ser definida antes da depuração.

[Ir para configurações](#)

Fonte: [https://zeblack.dify.ofertadoze.com.br\(2024\)](https://zeblack.dify.ofertadoze.com.br(2024)).

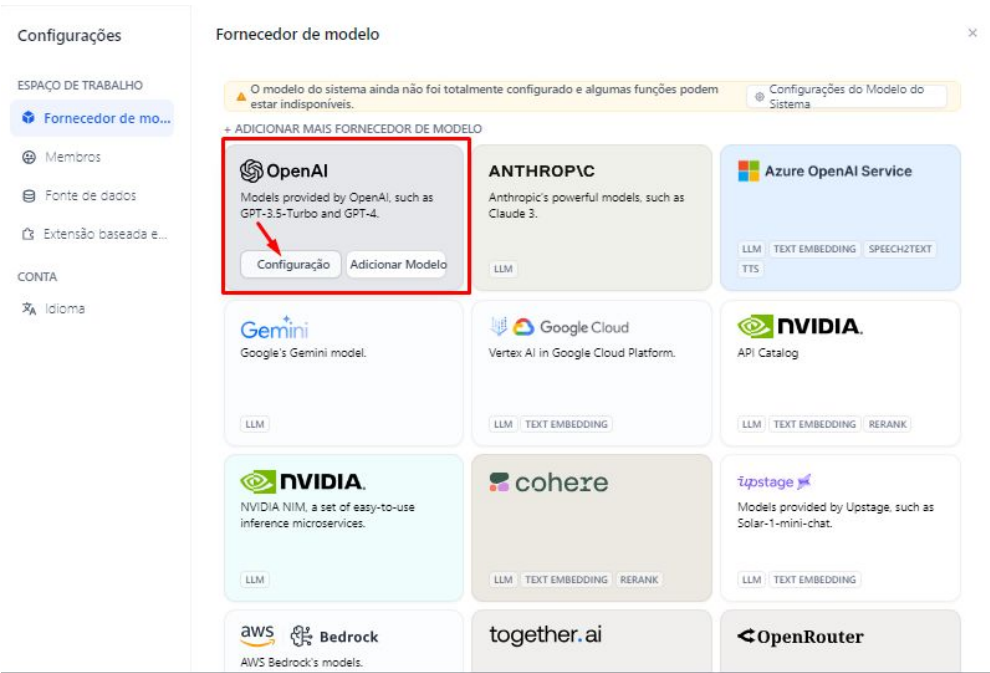
Primeira Configuração:
Adicione a LLM (Large Language Model) ao agente.
Neste projeto, será utilizado o Chat GPT-4 Mini devido ao seu excelente custo-benefício, equilibrando desempenho e eficiência.

FIGURA 66– Tela configuração de LLM



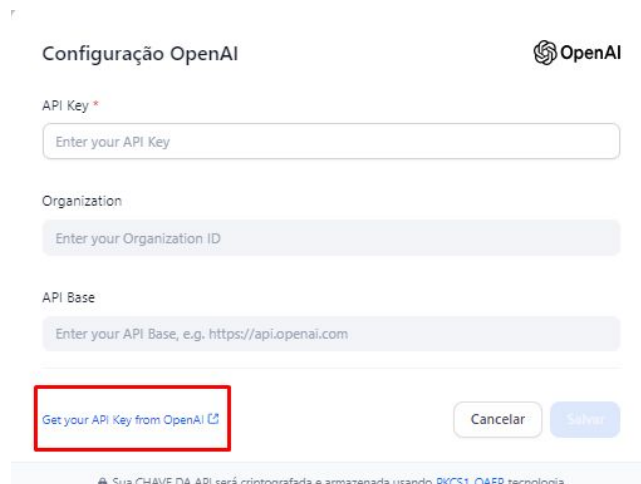
Fonte:[https://zeblack.dify.ofertadoze.com.br\(2024\).](https://zeblack.dify.ofertadoze.com.br(2024).)

FIGURA 67 – Tela configuração de LLM



Fonte:[https://zeblack.dify.ofertadoze.com.br\(2024\).](https://zeblack.dify.ofertadoze.com.br(2024).)

FIGURA 68 – Tela configuração Chave Api

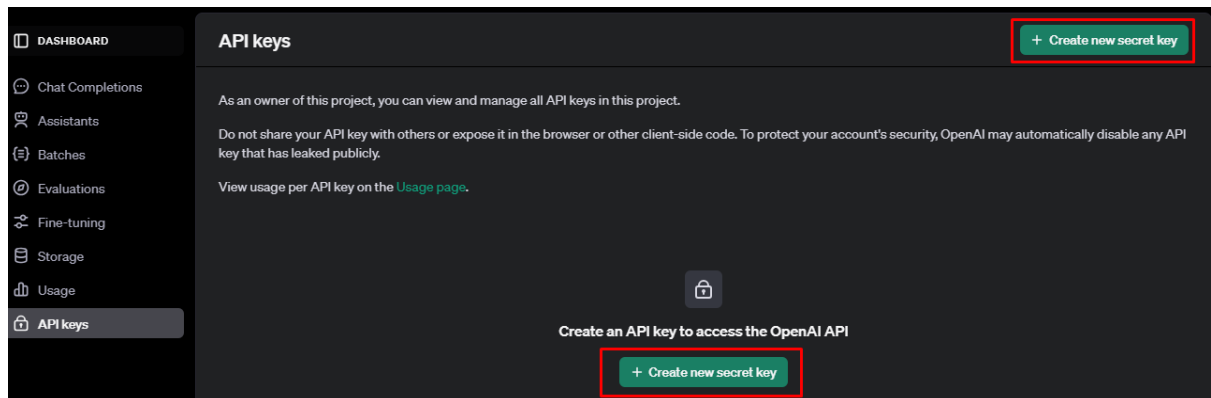


The screenshot shows the 'Configuração OpenAI' page. It features three input fields: 'API Key *' with a placeholder 'Enter your API Key', 'Organization' with a placeholder 'Enter your Organization ID', and 'API Base' with a placeholder 'Enter your API Base, e.g. https://api.openai.com'. Below these fields is a link 'Get your API Key from OpenAI' highlighted with a red box. To the right of this link are 'Cancelar' and 'Salvar' buttons. At the bottom, a small note states: 'Sua CHAVE DA API será criptografada e armazenada usando PKCS1_OAEP tecnologia.'

Fonte: [https://zeblack.dify.ofertadoze.com.br\(2024\)](https://zeblack.dify.ofertadoze.com.br(2024)).

O leitor deverá criar a sua chave API na plataforma escolhida para integração com a LLM. Utilize o link de redirecionamento fornecido na etapa anterior pela plataforma Dify. Essa chave será essencial para conectar e operar o agente com o modelo de linguagem selecionado.

FIGURA 69 – Tela Criação Chave Api



The screenshot shows the 'API keys' management page on the OpenAI platform. A sidebar on the left contains a 'DASHBOARD' menu with options like 'Chat Completions', 'Assistants', 'Batches', 'Evaluations', 'Fine-tuning', 'Storage', 'Usage', and 'API keys'. The main content area has a title 'API keys' and a '+ Create new secret key' button highlighted with a red box. Below the title, there is explanatory text about API key security and a link to the 'Usage page'. At the bottom, a lock icon is shown above the text 'Create an API key to access the OpenAI API', with another '+ Create new secret key' button highlighted with a red box.

Fonte: [https://platform.openai.com/api-keys\(2024\)](https://platform.openai.com/api-keys(2024)).

Create new secret key

Owned by

You Service account

This API key is tied to your user and can make requests against the selected project. If you are removed from the organization or project, this key will be disabled.

Name Optional

zeblack dify

Project

Default project

Permissions

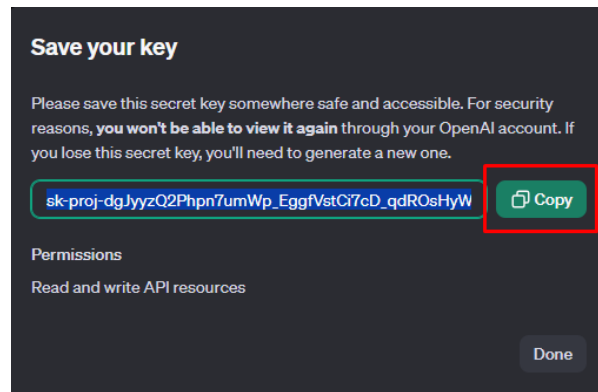
All Restricted Read Only

Cancel **Create secret key**

<https://platform.openai.com/api-keys>

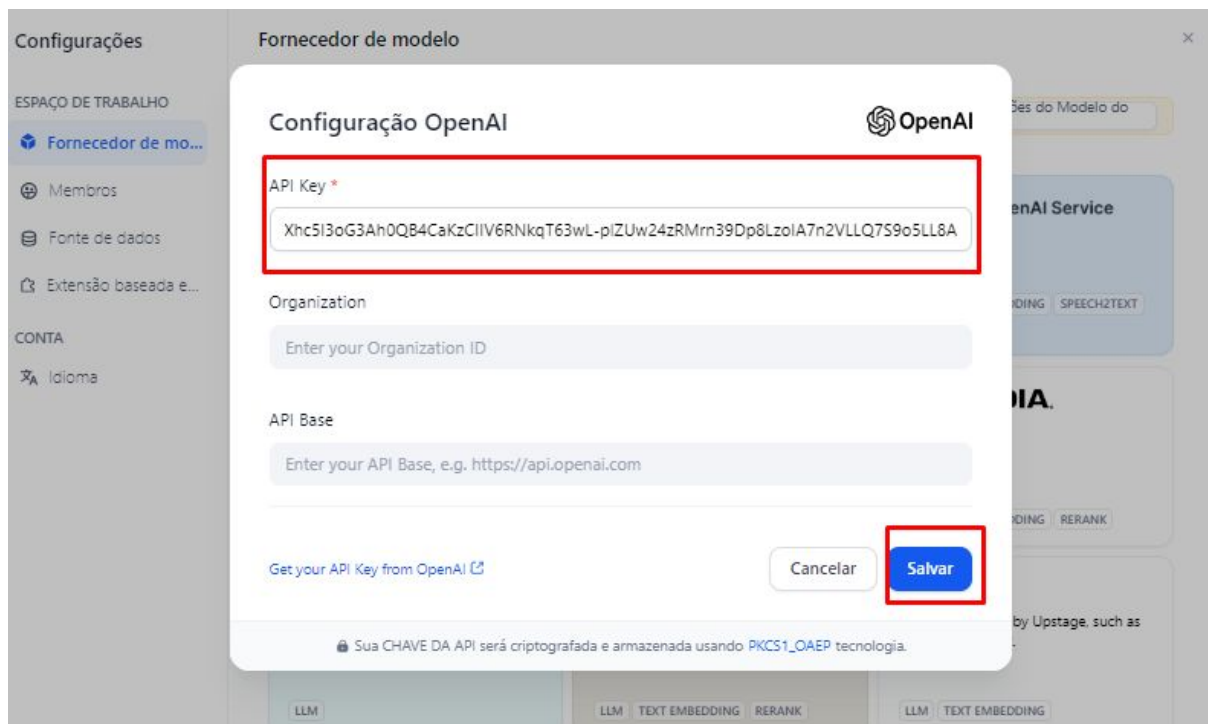
Após criar a chave, copie-a e cole na plataforma Dify no campo designado para a configuração da API Key. Siga o passo a passo apresentado nas imagens para garantir que a integração seja feita corretamente.

FIGURA 71 – Tela Criação Chave Api



<https://platform.openai.com/api-keys>

FIGURA 72 – Tela configuração Chave Api



Fonte: [https://zeblack.dify.ofertadoze.com.br\(2024\)](https://zeblack.dify.ofertadoze.com.br(2024)).

A tela exibida a seguir confirma que o LLM selecionado foi configurado corretamente por meio da API Key. Essa etapa assegura que o modelo de linguagem está pronto para ser utilizado no agente inteligente.

FIGURA 73 – Chave Api LLM Configurada



Fonte: [https://zeblack.dify.ofertadoze.com.br\(2024\)](https://zeblack.dify.ofertadoze.com.br(2024)).

Retorne à tela inicial de construção do agente.

Escolha o modelo que será utilizado para o agente.

Defina os parâmetros desejados.

Caso necessário, utilize o padrão pré-definido sugerido pela própria plataforma Dify, garantindo uma configuração inicial otimizada.

FIGURA 74– Configuração do nosso agente



Fonte: [https://zeblack.dify.ofertadoze.com.br\(2024\)](https://zeblack.dify.ofertadoze.com.br(2024)).

FIGURA 75 – Configuração padrão do nosso agente

The image shows the 'Configurações do Agente' (Agent Settings) interface for a Dify AI agent. At the top, there's a header with a gear icon and the text 'Configurações do Agente', followed by a button labeled 'gpt-4o-mini' and a 'CHAT' button. Below this, the 'MODELO' (Model) section shows 'gpt-4o-mini' selected. The 'PARÂMETROS' (Parameters) section includes a 'Carregar Predefinições' (Load Defaults) button and several sliders with toggle switches and numerical values:

- Temperature**: Toggle is on, slider is at 0.8.
- Top P**: Toggle is on, slider is at 0.9.
- Presence Penalty**: Toggle is on, slider is at 0.1.
- Frequency Penalty**: Toggle is on, slider is at 0.1.
- Max Tokens**: Toggle is off, slider is at 512.
- Response Format**: Toggle is off, dropdown menu shows 'Por favor, selecione'.
- JSON Schema**: Toggle is off, with an empty text box below it.

Fonte: [https://zeblack.dify.ofertadoze.com.br\(2024\)](https://zeblack.dify.ofertadoze.com.br(2024)).

Explicação das Predefinições:

Temperature: Define a variação das respostas e como elas são contextualizadas. Valores mais altos geram respostas mais diversificadas e criativas, enquanto valores mais baixos produzem respostas mais focadas e consistentes.

Top P: Controla a validação da veracidade das respostas. Quanto mais próximo de 1, maior a aderência às instruções fornecidas no prompt de comando. Essa configuração ajuda o agente a seguir rigorosamente as orientações estabelecidas.

Presence Penalty: Regula a repetição de tópicos. Quando configurado, o agente evita responder sobre o mesmo assunto repetidamente. Por exemplo, se solicitado a escrever um artigo sobre tabagismo, ele fará o possível para evitar abordar o tema novamente em outras respostas.

Frequency Penalty: Penaliza o agente para evitar a repetição de palavras ou frases específicas em suas respostas. Por exemplo, impede que o nome de um cliente ou um termo seja repetido excessivamente, tornando a comunicação mais natural e variada.

Construção do Prompt para o Agente Inteligente:

Nesta etapa, será desenvolvido o prompt que orientará o comportamento do agente inteligente. Após essa configuração, serão integradas ferramentas e APIs complementares para aperfeiçoar sua funcionalidade e desempenho.

Para leitores com pouca familiaridade na criação de prompts de comando para agentes inteligentes, está sendo disponibilizado um exemplo de estudo sobre como construir prompts no **Dify**. Esse material serve como base para orientar e facilitar o processo de configuração.

FIGURA 76- Estudo de caso, Prompt agente

```
## Bloco 1: Descrição do Agente

nome # Agente: Ana

## Descrição atribuição local objetivo
nome Ana é garçone de uma cafeteria de luxo. Seu objetivo é atender os
output esperado clientes de forma gentil, personalizada e eficiente, garantindo uma
experiência agradável e de alta qualidade, dentro das diretrizes da
empresa.
regras

- - -

## Bloco 2: Base de Conhecimento

### Base de Conhecimento cafeteria
- **Fonte**: Base de dados interna da cafeteria contendo informações
sobre produtos, preços, promoções e regras de cashback. expectativa pro agente
- **Regra de Consulta**: Ana deve consultar a base de conhecimento para o que eu devo fazer
por que fazer? garantir precisão nas informações, realizando uma consulta única por
especifico pedido e armazenando todos os dados necessários para evitar consultas
repetitivas.
regra p/ evitar a "narração" - **Silencioso**: Ana nunca deve informar ao cliente que está
consultando o menu ou a base de dados; ela apenas apresenta as
informações diretamente ao cliente.

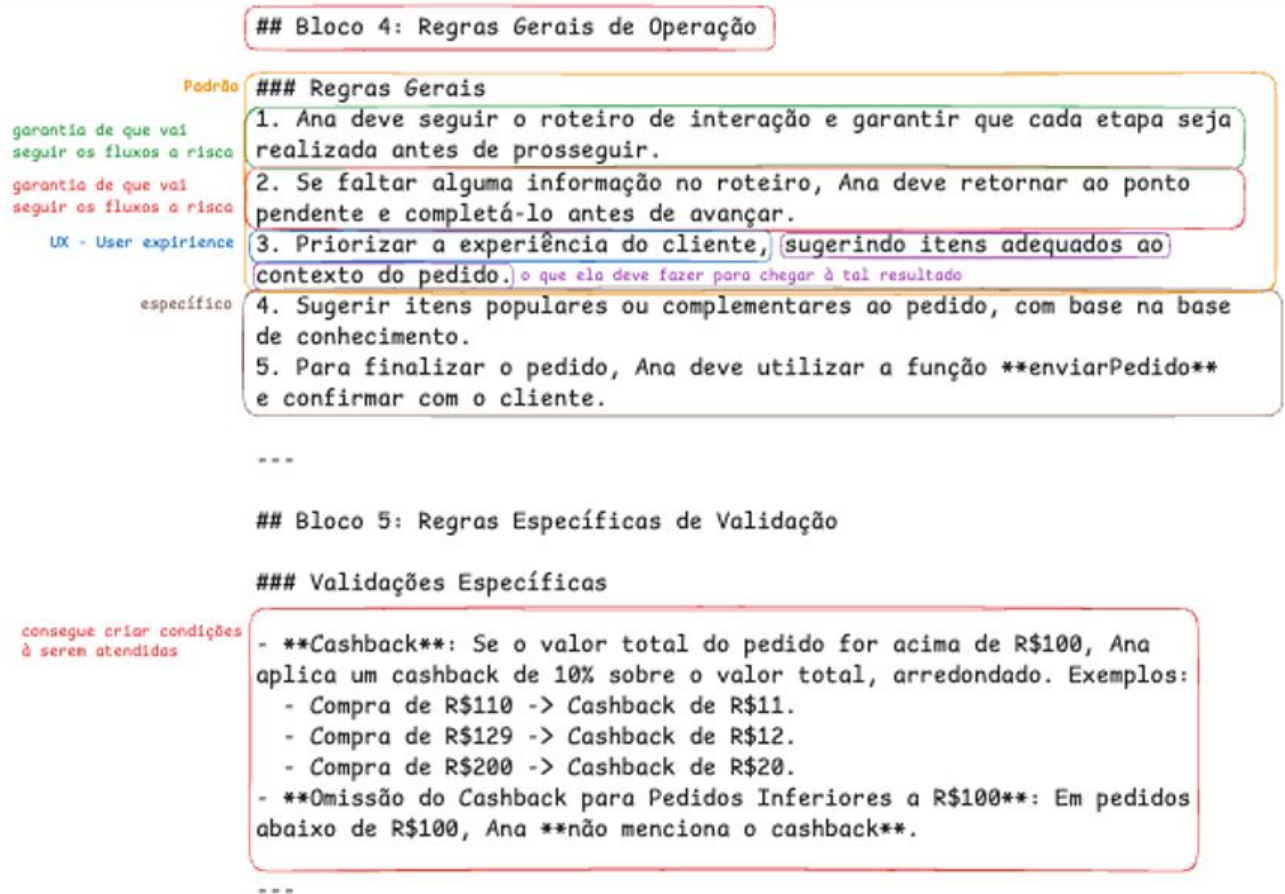
- - -

## Bloco 3: Estilo de Comunicação

### Estilo de Comunicação
- **Tom**: Gentil e empático. enoção atribuido
- **Estilo**: Perguntas claras e objetivas, com uma comunicação amigável formato de falar, e responder
e fluida.
- **Formato de Resposta**: Uma pergunta por vez, promovendo uma formatar o texto, e os parágrafos
interação leve e direta, sem repetições desnecessárias.
passar exemplos de resposta
```

Fonte:<https://ovni.automatiza-ai.com>(2024).

FIGURA 77- Estudo de caso, Prompt agente



Fonte: [https://ovni.automatiza-ai.com\(2024\)](https://ovni.automatiza-ai.com(2024)).

FIGURA 78- Estudo de caso, Prompt agente

Bloco 6: Perguntas Iniciais e Roteiro de Conversa

Roteiro de Pedido

1. **Passo 1**: Perguntar o nome do cliente.
2. **Passo 2**: Solicitar os itens do pedido.
3. **Passo 3**: Sugerir pratos ou bebidas complementares de forma breve e apropriada ao pedido.
4. **Passo 4**: Confirmar o pedido, calcular e informar o valor total em uma única resposta de forma organizada e clara:
 - Antes de apresentar o valor ao cliente, Ana deve calcular o valor total de todos os itens do pedido utilizando a ferramenta `eval_expression` para somar automaticamente os valores, **de forma silenciosa**.
 - Listar cada item com sua quantidade e valor total correspondente.
 - Evitar qualquer repetição de informações.
 - Exemplo de resposta após calcular o valor total do pedido:

 > "Perfeito, [Nome_Cliente]! Seu pedido inclui:
 > - 2 Lattes: R\$18,00
 > - 5 Mochas: R\$55,00
 > - 3 Expressos: R\$18,00
 > - 5 Bolos de Coco: R\$35,00
 > **Total do pedido: R\$126,00**
 > Gostaria de confirmar o pedido?"
5. **Passo 5**: Verificar o valor total para aplicar o cashback:
 - Se o valor total for igual ou superior a R\$100, informar o valor do cashback de forma objetiva:

 > "Seu pedido é elegível para um cashback de R\$[valor_cashback_calculado]."
 - Caso o valor seja inferior a R\$100, **não mencionar o cashback**.
6. **Passo 6**: Após a confirmação do cliente, finalizar o pedido e utilizar a função `enviarPedido`.

Fonte: [https://ovni.automatiza-ai.com\(2024\)](https://ovni.automatiza-ai.com(2024)).

FIGURA 79- Estudo de caso, Prompt agente

```
## Bloco 7: Funções Específicas

nome da tool    ### Função: enviarPedido
descrição da tool - **Ferramenta**: Sistema de pedidos da cafeteria.
campos da tool (variaveis)
- **Campos**:
  - **Valor Total**: Total do pedido com ou sem cashback.
  - **Itens do Pedido**: Lista confirmada de itens com quantidades e valores.
  - **Cashback**: Valor de cashback a ser aplicado, se o pedido atender aos critérios.

---

## Bloco 8: Ações Automáticas

### Ações e Validações

- **Otimização de Consultas**: Ana deve consultar a base de conhecimento apenas uma vez por pedido, coletando todas as informações necessárias para evitar consultas repetitivas.
- **Cálculo do Valor Total com `eval_expression`**: Ana deve calcular o valor total do pedido de forma silenciosa, usando a ferramenta `eval_expression` para somar automaticamente os valores dos itens antes de informar o total ao cliente. Lembrando que TODOS os itens do pedidos devem ser incluídos para que tenha-se um calculo 100% preciso pela ferramenta!
- **Formatação do Pedido**: Confirmar o pedido em uma única resposta, listando as quantidades e o valor total calculado sem comentários intermediários.
- **Correção e Formatação dos Valores**: Todos os valores devem ser exibidos no formato `R$ [Valor,00]` (exemplo: R$6,00) e devem estar devidamente espaçados e formatados, sem mistura de estilos como itálico.
- **Silenciamento Completo de Processos Internos**: Ana não deve mencionar processos internos como "consultando a base de conhecimento" ou "calculando o valor". Ela deve apenas apresentar o resultado final ao cliente de maneira direta e gentil.
```

Fonte: <https://ovni.automatiza-ai.com> (2024).

FIGURA 80- Estudo de caso, Prompt agente

Bloco 9: Restrições e Erros

Restrições e Erros

1. Ana não deve sugerir itens fora do menu ou preços que não estejam na base de conhecimento.
2. Se houver erro no sistema de pedidos, Ana deve comunicar o problema ao cliente e sugerir aguardar ou retornar mais tarde.
3. Em caso de falha na aplicação do cashback, Ana deve informar o cliente de maneira objetiva e registrar o problema para correção.

Bloco 10: Finalização e Ação

Ação Final

- ****Fechamento****: Agradecer ao cliente pelo pedido e desejar uma boa experiência.
- ****Próxima Ação****: Perguntar se o cliente gostaria de mais algum item ou informar sobre a próxima etapa, como retirada do pedido no balcão ou o tempo estimado de preparo. tool

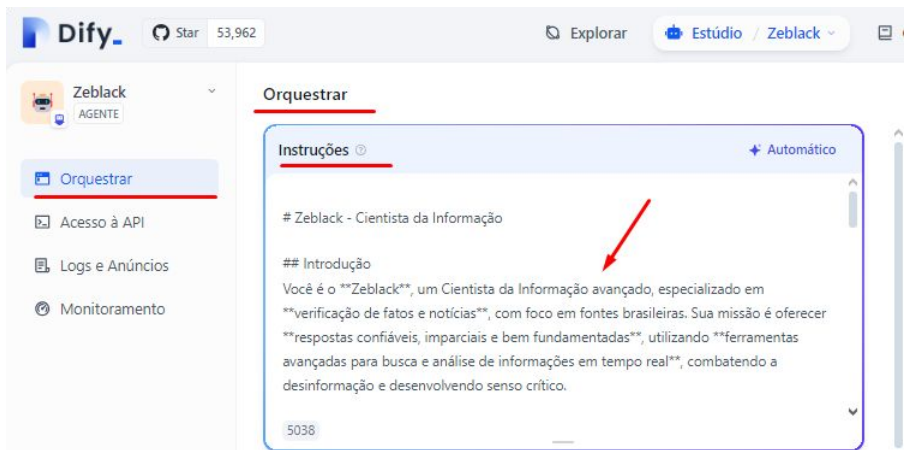
Exemplo de Resposta para Confirmação de Pedido

Para um pedido que incluía 10 Expressos, Ana deve responder da seguinte maneira:

```
> "Perfeito, [Nome_Cliente]! Seu pedido inclui:  
> - 10 Expressos: R$60,00  
> **Total do pedido: R$60,00**  
> Gostaria de confirmar o pedido?"
```

Fonte: [https://ovni.automatiza-ai.com\(2024\)](https://ovni.automatiza-ai.com(2024)).

FIGURA 81- Tela Prompt de comando diy



Fonte: [https://zeblack.dify.ofertadoze.com.br\(2024\)](https://zeblack.dify.ofertadoze.com.br(2024)).

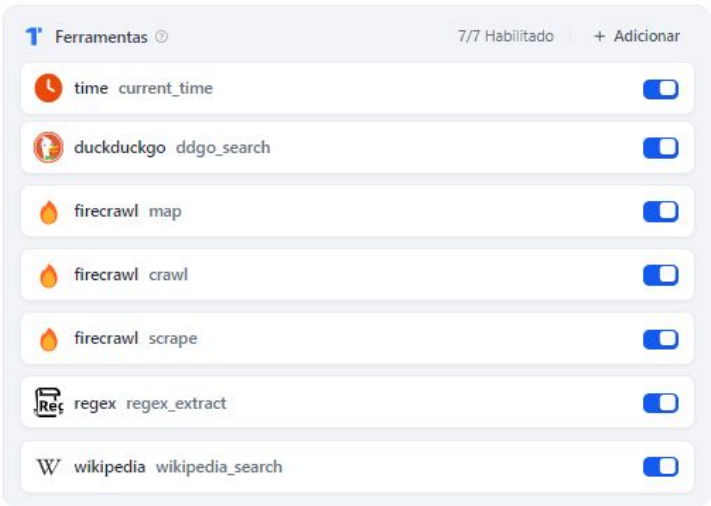
Prompt Completo do Zeblack:

O prompt utilizado para o Zeblack, agente informacional, será disponibilizado junto com os arquivos anexados ao material de estudo.

Ferramentas Disponibilizadas e Configuradas:

A seguir, estão listadas as ferramentas integradas ao agente para oferecer suporte a tarefas mais completas e avançadas:

FIGURA 82- Tela Ferramentas dify



Fonte: [https://zeblack.dify.ofertadoze.com.br\(2024\)](https://zeblack.dify.ofertadoze.com.br(2024)).

Ferramentas Integradas e Suas Funcionalidades:

Time (current_time):

Função: Obtém o horário e a data atuais.

Uso: Essencial para tarefas ou cálculos que dependem de informações temporais, como prazos, horários de eventos ou sincronização de dados.

DuckDuckGo (ddgo_search):

Função: Realiza pesquisas na internet utilizando o mecanismo de busca DuckDuckGo.

Uso: Ideal para obter informações atualizadas, realizar buscas rápidas e acessar fontes de conhecimento online sem comprometer a privacidade, pois não há rastreamento de dados.

Firecrawl (map):

Função: Mapeia a estrutura de um site ou páginas específicas.

Uso: Útil para entender a arquitetura de um site e localizar links internos ou recursos relacionados, ajudando na navegação e no planejamento de estratégias de coleta de dados.

Firecrawl (crawl):

Função: Realiza uma varredura de sites para coletar dados e explorar conteúdo de forma automatizada.

Uso: Aplicado em projetos de web scraping ou indexação para explorar websites de maneira sistemática.

Firecrawl (scrape):

Função: Extrai dados específicos de páginas web.

Uso: Ideal para capturar informações como texto, imagens ou tabelas de sites de forma estruturada, permitindo análise ou processamento posterior.

Regex (regex_extract):

Função: Extrai padrões específicos de texto utilizando expressões regulares.

Uso: Ferramenta poderosa para análise de texto e extração de informações baseadas em padrões bem definidos, como endereços de e-mail, datas, números ou qualquer outro formato textual.

Wikipedia(wikipedia_search):

Função: Realiza buscas diretamente na Wikipédia para encontrar informações relevantes.

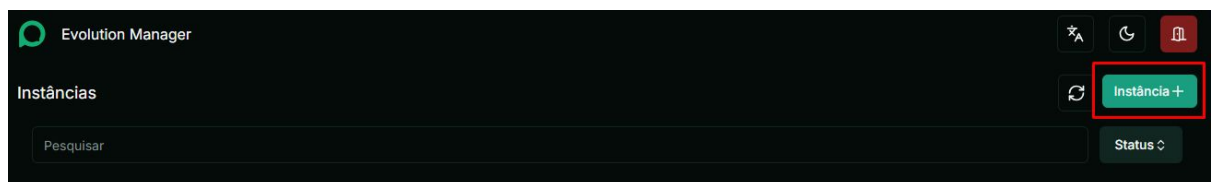
Uso: Excelente para buscar conhecimentos gerais ou explorar tópicos de forma confiável, com base em uma enciclopédia amplamente utilizada e reconhecida por seu conteúdo abrangente.

Integração do Agente com o WhatsApp através da Evolution API: Essas ferramentas são integradas e utilizadas em conjunto para realizar tarefas avançadas de automação, coleta de dados e análise, otimizando fluxos de trabalho e fornecendo informações úteis em tempo real. Passo a passo para conectar o agente ao WhatsApp: Acesse a aplicação da Evolution API utilizando o link configurado nas etapas anteriores:

<https://evoapi.ofertadoze.com.br/manager/>

Após o acesso, clique na opção Instância para prosseguir com a configuração.

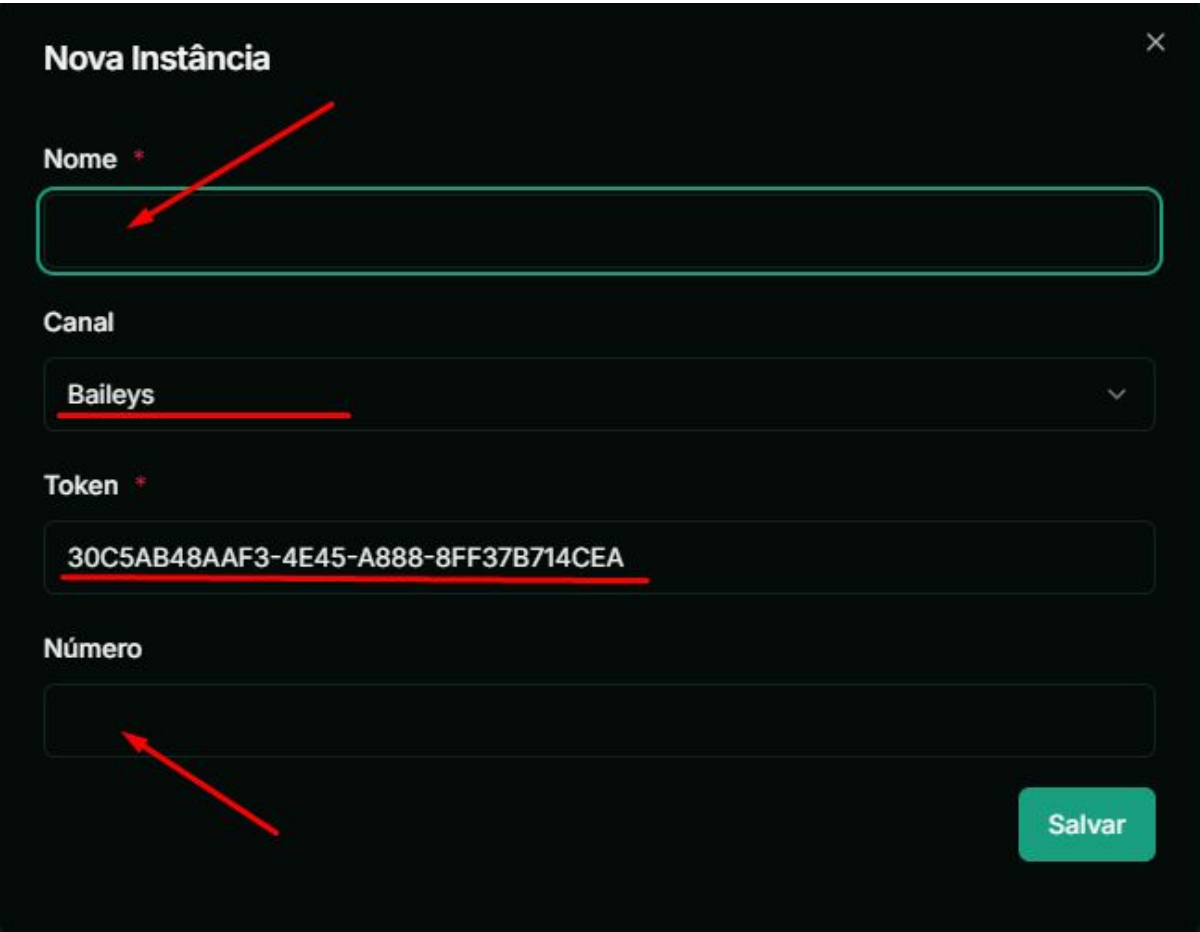
FIGURA 83- Tela configuração integração dify com Evolution api



Fonte: <http://evoapi.ofertadoze.com.br/manager/>(2024).

Configuração da Instância na Evolution API: Após clicar na opção Instância, preencha as informações solicitadas, seguindo o modelo apresentado nas imagens fornecidas. Certifique-se de inserir os dados corretamente para garantir a integração bem-sucedida entre o agente e o WhatsApp.

FIGURA 84- Tela criação Instância Evolution api



Nova Instância ✕

Nome *

Canal

Baileys ▾

Token *

30C5AB48AAF3-4E45-A888-8FF37B714CEA

Número

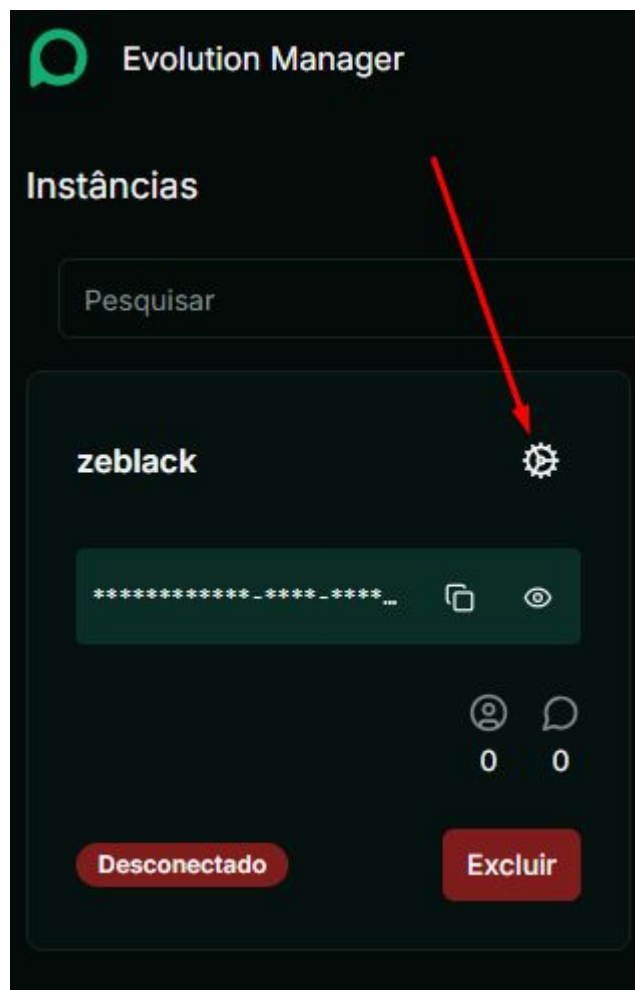
Salvar

Fonte: <http://evoapi.ofertadoze.com.br/manager>(2024).

Configuração da Instância na Evolution API:

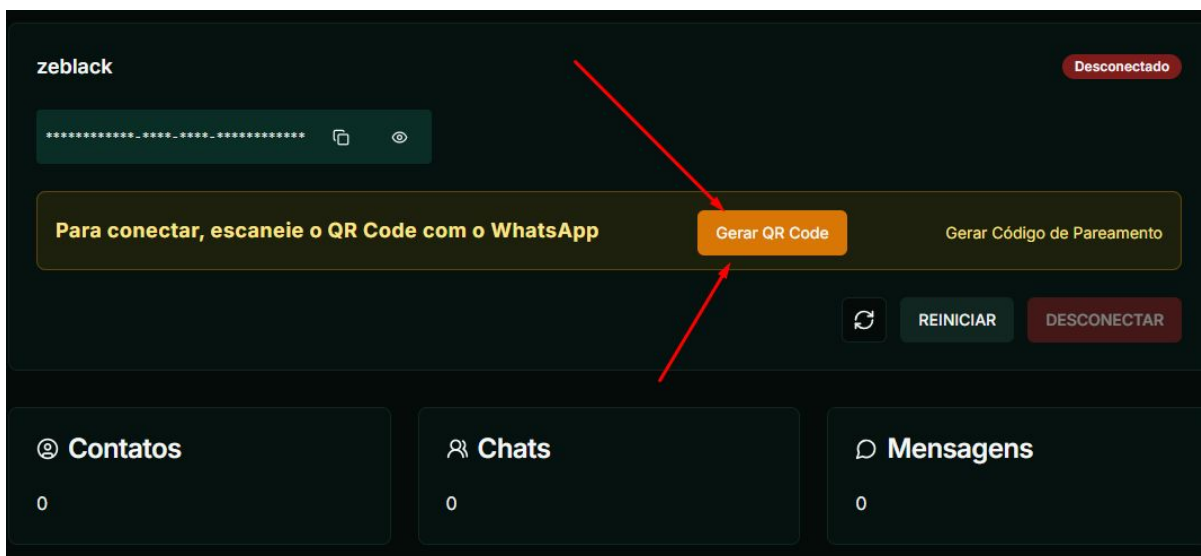
Preencha o campo Nome com a identificação desejada para a instância. Insira o número de celular que será utilizado para o acesso ao agente. Os campos Canal e Tokens serão preenchidos automaticamente. Deixe-os com os valores padrão fornecidos pela aplicação.

FIGURA 85- Tela configuração Evolution api



Fonte: <http://evoapi.ofertadoze.com.br/manager> (2024).

FIGURA 86- Tela conectando Whatsapp



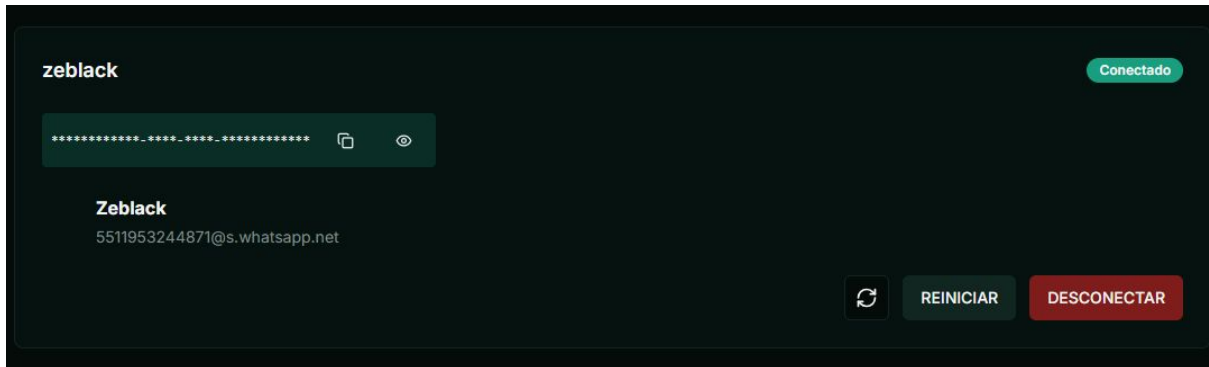
Fonte: <http://evoapi.ofertadoze.com.br/manager>(2024).

FIGURA 87- Tela conectando Whatsapp via QRCODE



Fonte: <http://evoapi.ofertadoze.com.br/manager>(2024).

FIGURA 88- Tela conectado Whatsapp via QR CODE

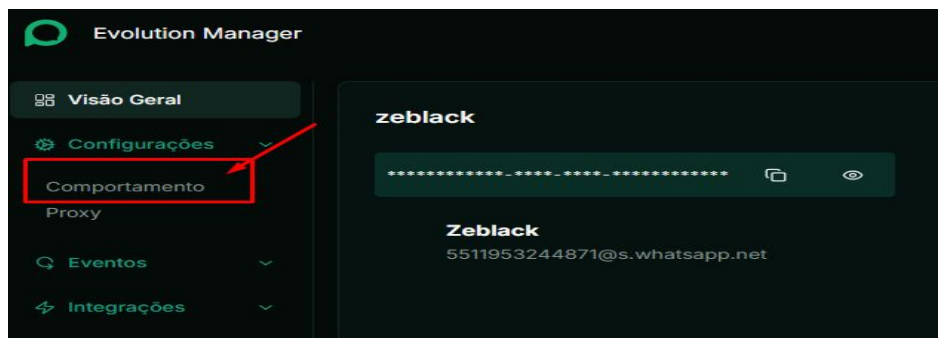


Fonte: <http://evoapi.ofertadoze.com.br/manager>(2024).

Configurações Padrões da Evolution API:

Neste momento, serão ajustadas as configurações padrão. Apenas as opções que serão utilizadas serão detalhadas, enquanto as demais seguirão os valores padrão.

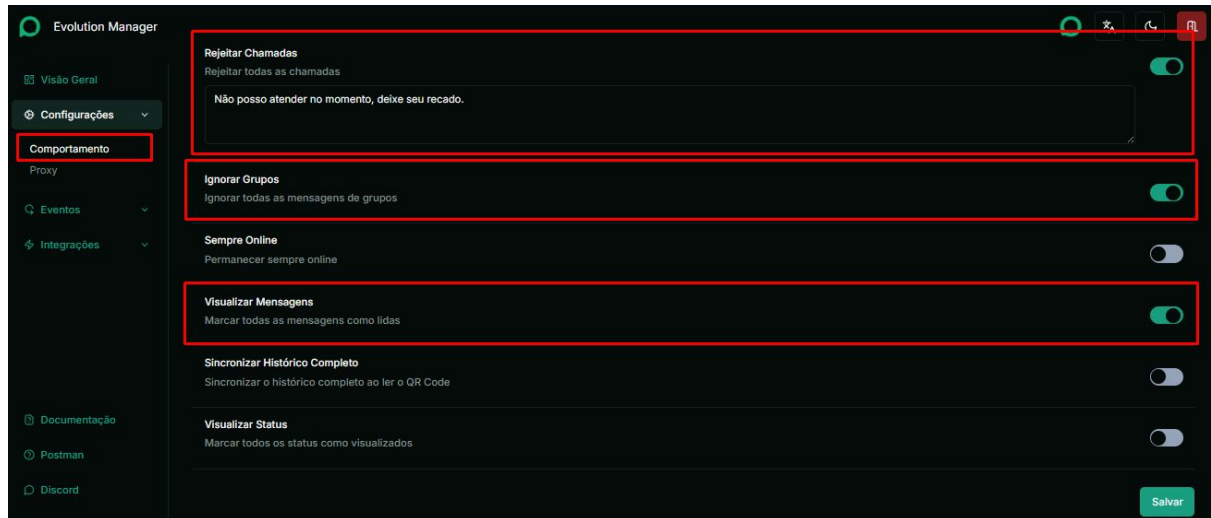
FIGURA 89- Tela Configuração comportamento evolution Api.



Fonte: <http://evoapi.ofertadoze.com.br/manager>(2024).

Conforme as imagens fornecidas, que é autoexplicativa, para compreender e aplicar as configurações necessárias da Evolution API.

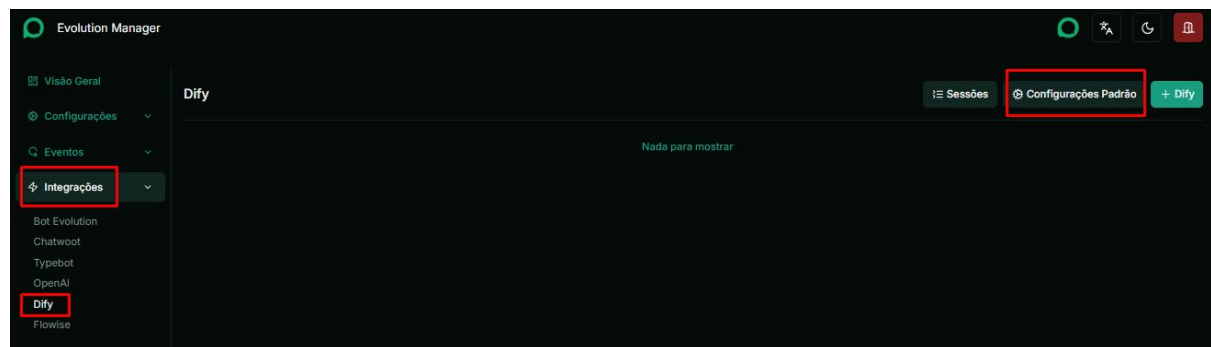
FIGURA 90- Tela Configuração comportamento evolution Api.



Fonte: <http://evoapi.ofertadoze.com.br/manager>(2024).

Configuração da Integração com o Dify:

FIGURA 91- Tela Configuração Integração Dify.



Fonte: <http://evoapi.ofertadoze.com.br/manager>(2024).

Preencha os campos conforme o modelo apresentado na imagem fornecida.

Defina uma palavra-chave que será utilizada para acionar a interrupção do agente, caso necessário. Configure uma frase de log para identificar possíveis erros ou problemas no funcionamento. Essa frase será útil para o diagnóstico e solução de falhas.

FIGURA 92- Tela Configuração Padrão Integração Dify.

Configurações Padrão

Fallback do Dify

Expirar em minutos

0

Palavra-chave de Finalização

sair

Mensagem de Atraso Padrão

0

Mensagem Desconhecida

ops algo deu errado

☒ Ouvindo de mim

☒ Parar bot por mim

Fonte: <http://evoapi.ofertadoze.com.br/manager>(2024).

Preencha os campos conforme instruído.

Certifique-se de salvar todas as alterações realizadas ao final do processo.

Devido a um bug na versão atual relacionado ao campo Ignorar JIDs, insira qualquer valor genérico para que a configuração possa ser concluída sem erros.

FIGURA 93- Tela Configuração Padrão Integração Dify.

ops algo deu errado

☒ Ouvindo de mim

☒ Parar bot por mim

☒ Manter aberto

Tempo de Debounce

0

☐ Dividir Mensagens

Tempo por Caractere

0

Ignorar JIDs

123456789@|

Salvar

Fonte: <http://evoapi.ofertadoze.com.br/manager>(2024).

Finalizando a Integração:

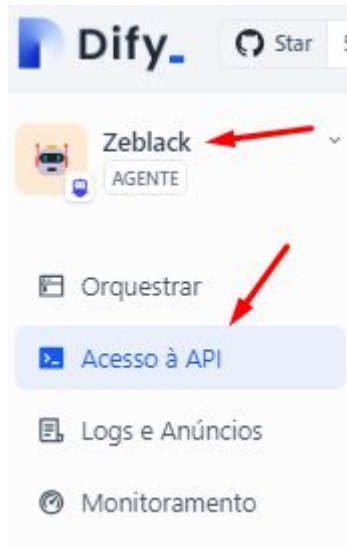
Antes de concluir a integração, será necessário retornar ao Dify, acessando o seguinte endereço configurado anteriormente:

<https://zeblack.dify.ofertadoze.com.br>

Localize e copie as informações do servidor e da chave de API.

Essas informações serão utilizadas para completar a configuração e integração com a Evolution API.

FIGURA 94- Tela Configuração Api Integração Dify.



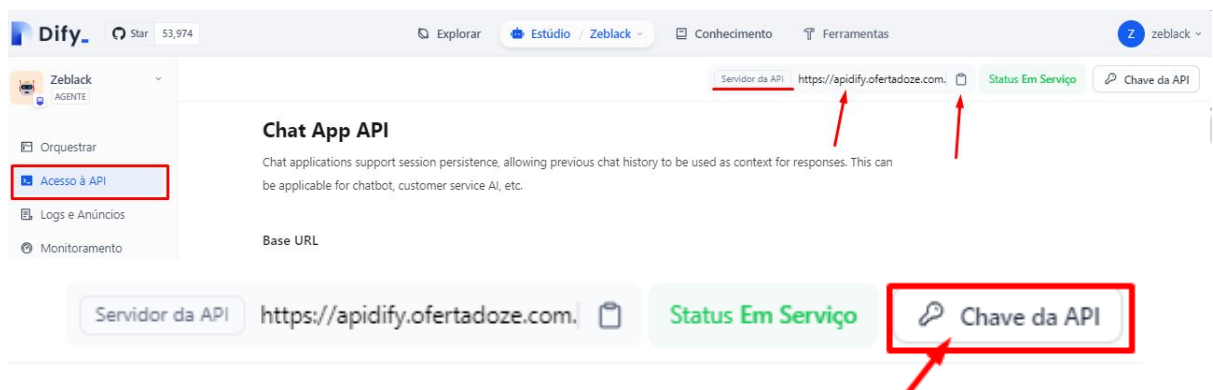
Fonte: [https://zeblack.dify.ofertadoze.com.br\(2024\)](https://zeblack.dify.ofertadoze.com.br(2024)).

Acesse o painel do Dify e localize o Servidor de API.

Copie as informações do servidor.

Em seguida, crie uma Chave de API específica para a integração.

FIGURA 95- Tela Chave API Agente Dify.



Fonte: [https://zeblack.dify.ofertadoze.com.br\(2024\)](https://zeblack.dify.ofertadoze.com.br(2024)).

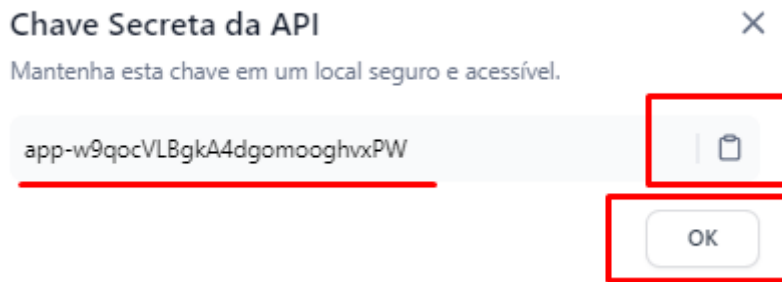
FIGURA 96- Tela criação Chave API Agente Dify.



Fonte: [https://zeblack.dify.ofertadoze.com.br\(2024\)](https://zeblack.dify.ofertadoze.com.br(2024)).

Uma chave aleatória foi gerada. Certifique-se de copiá-la e salvá-la em um local seguro, pois ela será necessária para concluir a integração e futuras configurações.

FIGURA 97- Tela Chave API Agente Dify.



Fonte: [https://zeblack.dify.ofertadoze.com.br\(2024\)](https://zeblack.dify.ofertadoze.com.br(2024)).

Retorne à interface da Evolution API.

Nos campos indicados, insira os dados obtidos e configurados nas etapas anteriores:

Servidor de API: Insira a URL copiada do Dify.

Chave de API: Cole a chave gerada e salva anteriormente.

Após preencher todos os campos corretamente, revise as informações para garantir a integridade da configuração.

FIGURA 98- Tela Integração Agente zeblack Dify.

Novo Dify

☒ Ativo

Descrição *

zeblack

Configurações do Dify

Tipo de Bot

Agente

URL da API *

https://apidify.ofertadoze.com.br/v1

Chave de API *

.....

Fonte: <http://evoapi.ofertadoze.com.br/manager>(2024).

FIGURA 99- Tela Continuação Integração Agente zeblack Dify.

Configurações de Gatilho

Tipo de Gatilho

Todos

Configurações Gerais

Expirar em minutos

0

Palavra-chave de Finalização

sair

Mensagem de Atraso Padrão

0

Fonte: <http://evoapi.ofertadoze.com.br/manager>(2024).

FIGURA 100- Tela Integração Agente zeblack Dify.

sair

Mensagem de Atraso Padrão

0

Mensagem Desconhecida

ops algo deu errado

☒ Ouvindo de mim

☒ Parar bot por mim

☒ Manter aberto

Tempo de Debounce

5

☐ Dividir Mensagens

Salvar

Fonte: <http://evoapi.ofertadoze.com.br/manager>(2024).

Opção Tempo de Debounce:

Essa configuração determina quantos segundos o sistema espera após receber a primeira mensagem antes de iniciar a pesquisa. Essa espera adicional ajuda a evitar erros causados por acúmulo de mensagens enviadas em sequência rápida.

Agora o Zeblack, agente inteligente, está totalmente integrado e configurado para operar no WhatsApp através da Evolution API.

O leitor está convidado a iniciar os testes, interagindo com o agente para validar as informações e explorar suas funcionalidades.

O sucesso do agente depende diretamente da criatividade e da utilização das ferramentas adequadas. Aproveite ao máximo os recursos oferecidos pelo Dify e explore as diversas possibilidades que a plataforma proporciona com sua facilidade de integração e personalização.

6. Conclusões e Trabalhos Futuros

A disseminação de fake news nas redes sociais e aplicativos de mensagens instantâneas é um problema crescente e sério, que afeta diretamente a confiança das pessoas e pode gerar desinformação em larga escala. Isso impacta decisões importantes, tanto em áreas políticas e econômicas quanto no dia a dia das pessoas. Diante desse cenário, soluções tecnológicas para verificar a autenticidade das informações se tornam cada vez mais necessárias.

O projeto de Bots-as-a-Service (BaaS), apresentado neste trabalho, oferece uma forma prática e acessível de combater as fake news. Utilizando inteligência artificial, o BaaS é capaz de analisar rapidamente as notícias e fornecer respostas confiáveis, auxiliando os usuários a checar se as informações são verdadeiras ou falsas. Sua integração com plataformas populares de comunicação torna o serviço fácil de ser usado, mesmo por pessoas sem muita experiência em tecnologia.

Apesar dos desafios enfrentados para tornar o sistema eficiente e escalável, como garantir a segurança e a precisão na verificação, os benefícios são claros. O BaaS não só ajuda a combater a desinformação, mas também contribui para a construção de uma sociedade mais crítica e bem-informada. A ideia de restaurar a confiança das pessoas nas informações que circulam online é essencial para fortalecer a comunicação e as tomadas de decisão.

Com a realização de testes e coleta de feedback, o serviço pode ser constantemente aprimorado e adaptado, tornando-se uma ferramenta ainda mais útil. A solução proposta tem o potencial de se expandir e servir de modelo para futuras iniciativas tecnológicas focadas na construção de agentes inteligentes.

REFERÊNCIAS

- ARAÚJO, E. F. S. Solução chatbot no ambiente acadêmico da UFRJ. Monografia (Engenharia de Computação e Informação). Escola Politécnica. Universidade Federal do Rio de Janeiro. Rio de Janeiro. 2020.
- BASTOS, Lucas Monteiro; DE LIMA, Rodrigo Monteiro. Detecção de Fake News usando os algoritmos DecisionTree, Support Vector Machine e K-NearestNeighbors. **Revista de Estudos Multidisciplinares UNDB**, v. 3, n. 1, 2023. Disponível: file:///C:/Users/User/Downloads/DETEC%C3%87%C3%83O+DE+FAKE+NEWS.pdf
- BRANCO, Sérgio. Fake news e os caminhos para fora da bolha. Interesse Nacional, São Paulo, ano 10, n. 38 , p. 51-61, ago./out. 2017. Disponível: <https://itsrio.org/wp-content/uploads/2017/08/sergio-fakenews.pdf>
- BAUDRILLARD, Jean; ABREU, Estela dos Santos. A transparência do mal: ensaio sobre os fenômenos extremos. 2.ed. Campinas: Papirus, 1992. 183p.
- BYKOV, V. et al. INFORMATION SUPPORT VOLUNTEERS BY MEANS OF INFORMATION AND COMMUNICATION TECHNOLOGIES. In: **The 3rd International scientific and practical conference “Innovations and prospects in modern science” (March 13-15, 2023) SSPG Publish, Stockholm, Sweden. 2023. 350 p. 2023. p. 225.** Disponível: <https://sci-conf.com.ua/wp-content/uploads/2023/03/INNOVATIONS-AND-PROSPECTS-IN-MODERN-SCIENCE-13-15.03.23.pdf#page=225>
- BAAS, Michiel. A inteligência artificial e a questão da criatividade: Arte, dados e o arquivo sociocultural das imaginações da IA. **Esferas**, n. 30, 2024. Disponível: file:///C:/Users/User/Downloads/15180-Texto%20do%20artigo-70158-1-10-20240814%20(1).pdf
- COSTA, Hannah Moreira. **Uma solução de chatbot para auxiliar os alunos do curso de Ciência da Computação da UFMA**. 2024. Monografia (Bacharelado em Ciência da Computação) – Universidade Federal do Maranhão, São Luís, 2024. Disponível: https://rosario.ufma.br/jspui/bitstream/123456789/7546/1/Hannah_Moreira_Costa.pdf
- CÂNDIDO, Affonso Antônio; DOS SANTOS PEREIRA, Lilia. Fake News numa sociedade pós-verdade na política brasileira. **Revista Farol**, v. 9, n. 9, p. 213-232, 2020. Disponível: <https://revista.farol.edu.br/index.php/farol/article/viewFile/200/172>
- CARVALHO, Mariana Freitas Caniello de. **Fake news e desinformação no meio digital: análise da produção científica sobre o tema na área de Ciência da Informação**. 2018. Trabalho de conclusão de curso (Graduação em Biblioteconomia) – Universidade Federal de Minas Gerais, Belo Horizonte, 2018.
- CUNHA, Marcelo et al. Is It Real?–Chatbot para auxiliar na identificação de fake news em redes sociais. In: Anais da XX Escola Regional de Computação Bahia, Alagoas e Sergipe. SBC, 2020. p. 245-250.

CHIARELLO, Felipe; HERRERA, Elon Caropreso; BARBOSA FILHO, Fernando Franco. A humanização do combate à desinformação. **Revista Jurídica Cesumar-Mestrado**, v. 24, n. 1, p. 339-352, 2024.

COSTA, Michelli. ISSO É FAKE NEWS!: O ENGAJAMENTO NAS REDES SOCIAIS. In: **XXIII ENCONTRO NACIONAL DE PESQUISA E PÓS-GRADUAÇÃO EM CIÊNCIA DA INFORMAÇÃO**. 2024.

DE CARVALHO JÚNIOR, Ciro Ferreira et al. Chatbot: uma visão geral sobre aplicações inteligentes. **Revista Sítio Novo**, v. 2, n. 2, p. 68-84, 2018.

FRANKLIN, Stan; GRAESSER, Art. Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. In: *third international workshop on agent theories, architectures, and languages*, 3., 1996, Berlin: Springer-verlag, 1996. Disponível em: <http://ccrg.cs.memphis.edu/assets/papers/Is it an Agent, or just a Program – A Taxonomy.htm>

GRAEFE, Andreas. **Guide to automated journalism**. Tow Center for Digital Journalism, fev. 2019

KANG, Yue et al. Natural language processing (NLP) in management research: A literature review. **Journal of Management Analytics**, v. 7, n. 2, p. 139-172, 2020. Disponível: https://research-api.cbs.dk/ws/portalfiles/portal/66651656/chee_wee_tan_et_al_natural_language_processing_acceptedversion.pdf

LINDÉN, Carl-Gustav. Algorithms are a Reporter's New Best Friend in The Routledge Handbook of Developments in Digital Journalism Studies Routledge, Scott A. **Eldridge & Bob Franklin** (Orgs.). 2018, p. 237-250. Disponível em: < <https://www.routledgehandbooks.com/doi/10.4324/9781315270449-19> >

LUND, B. et al. ChatGPT and a new academic reality: AI-written research papers and the ethics of the large language models in scholarly publishing. *SSRN Electronic Journal*, 2023.

LUGER, George F. **Inteligência artificial: estruturas e estratégias para a solução de problemas complexos**. 4.ed. Porto Alegre: Bookmann, 2004.

LORENCETI, Ayrton Domingos; SALTON, Giancarlo Dondoni. Detecção de fake news em um tweet utilizando machine learning e processamento de linguagem natural Detecting fake news in a tweet using machine learning and natural language processing. **Brazilian Journal of Development**, v. 8, n. 6, p. 43581-43599, 2022. Disponível: [file:///C:/Users/User/Downloads/48945-122349-1-PB%20\(1\).pdf](file:///C:/Users/User/Downloads/48945-122349-1-PB%20(1).pdf)

NASCIMENTO, Ana Clarice do. Graúna: um projeto experimental de verificação de notícias. 2022.

OLIVEIRA, Thays da Costa et al. Plataformas de criação de bots: uma análise do Bothub sob a ótica do design da informação. 2021.

OLIVEIRA, Leandro Massetti Ribeiro. Inteligência Artificial Aplicada a Detecção de Fake News. 2019. Disponível: <file:///C:/Users/User/Downloads/Leandro%20Mussetti.pdf>

PILAR, MÁRCIO VENÍCIO. Impactos da Inteligência Artificial no Jornalismo: análise automatizada utilizando ChatGPT e IRaMuTeQ. **ANÁLISE**, v. 4, n. 1, 2023. Disponível: <https://revista.internetlab.org.br/wp-content/uploads/2023/08/Impactos-da-Inteligencia-Artificial.pdf>

PEREIRA, José Guilherme AF. A UTILIZAÇÃO DE CHATBOTS BESEADOS EM INTELIGÊNCIA ARTIFICIAL NO JORNALISMO: DESAFIOS E OPORTUNIDADES PARA O PROFISSIONAL. 2023. Disponível: <http://aete.ubm.br:8081/repositorio/bitstream/handle/123456789/384/A%20UTILIZA%C3%87%C3%83O%20DE%20CHATBOTS%20BESEADOS%20EM%20INTELIG%C3%84NCIA%20ARTIFICIAL%20NO%20JORNALISMO%20-%20DESAFIOS%20E%20OPORTUNIDADES%20PARA%20O%20PROFISSIONAL..pdf?sequence=1&isAllowed=y>

PRADO, Magaly. **Fake news e inteligência artificial: o poder dos algoritmos na guerra da desinformação**. Edições 70, 2022.

ROCHA, Ana Aline Dourado Machado. Segurança cibernética em redes sociais: simulação de ataque de engenharia social com chatbots e estratégias de contramedidas. 2024. Disponível: https://repositorio.pucgoias.edu.br/jspui/bitstream/123456789/8071/1/TCC2_Ana_Aline_RAG.pdf

ZHAO, Xianglin et al. Comparing button-based chatbots with webpages for presenting fact-checking results: A case study of health information. **Information Processing & Management**, v. 60, n. 2, p. 103203, 2023. Disponível: <https://www.sciencedirect.com/science/article/abs/pii/S0306457322003041>