

# **Implementação de infraestrutura em nuvem para viabilizar solução escalável para aplicações em R com Docker e virtualização**

**Wesley Lourenco Barbosa<sup>1</sup>**

## **Resumo**

A linguagem R, com sua rica variedade de pacotes estatísticos e de análise de dados, oferece um ambiente poderoso para a criação de aplicações interativas. O Shiny, uma estrutura de desenvolvimento web para R, simplifica a construção de aplicativos interativos e sua integração com a nuvem. No entanto, a escalabilidade e a implantação eficiente de aplicativos Shiny em ambientes de nuvem podem ser desafiadoras, principalmente em um contexto de múltiplos usuários. Assim, o objetivo deste trabalho é investigar como implementar soluções escaláveis Shiny R na nuvem. A metodologia foi realizada a partir de uma revisão da literatura científica para identificar como o tema vem sendo investigado na academia. A alternativa mais interessante para escalabilidade de aplicações Shiny R identificada foi o ShinyProxy. Essa solução permite a implantação de aplicativos Shiny em contêineres e torna a disponibilização na nuvem uma tarefa mais eficaz. A convergência de tecnologias voltadas à computação em nuvem como contêineres, Docker, Kubernetes e microsserviços, podem ser combinados para possibilitar a criação e o compartilhamento de aplicações Shiny R escaláveis na nuvem. No entanto, temas relacionados a otimização de desempenho, gerenciamento de recursos computacionais, estratégias multicloud e fluxos de implementação contínua ainda carecem de investigação mais aprofundada da comunidade científica.

**Palavras-chave:** Shiny R. Cloud computing. Docker. ShinyProxy. Shiny Server

---

<sup>1</sup> Mestrando em Engenharia Elétrica pela Escola Politécnica da USP. E-mail: wesleyloubar@usp.br

## Introdução

Com os quase 20.000 pacotes disponíveis, a linguagem de programação R é uma ferramenta computacional amplamente utilizada para análise estatística e geração de gráficos (PARRA et al., 2023). Inicialmente desenvolvida por estatísticos para estatísticos, ela se tornou uma linguagem de programação muito utilizada pelos mais distintos campos das ciências. A sintaxe da linguagem permite aos pesquisadores importarem, limparem e analisarem facilmente dados de uma ampla variedade de fontes além de oferecer recursos para plotar dados e criar visualizações (PARRA et al., 2023). O R é usado para diversos fins, incluindo inferência estatística, análise de dados, aprendizado de máquina e execução de simulações científicas. Os milhares de pacotes úteis oferecidos pelo ecossistema R, como pacotes de regressão linear e não linear, árvores de decisão e classificação, florestas aleatórias, redes neurais, fazem com que o R permita implementar soluções de aprendizado de máquina em diversas áreas como varejo, marketing, medicina e finanças. No que tange a aspectos relacionados a big data, com grandes volumes de dados de inteligência artificial (IA), o R possui pacotes dedicados a sistema de computação distribuída usado para processamento e análise de big data, como Apache Spark, bibliotecas e frameworks de IA também estão disponíveis para a linguagem como o Keras, TensorFlow, Torch.

Por conter um ecossistema tão diverso, o R oferece uma vantagem na condução de análises de dados rápidas, juntamente com uma visualização de resultados. Além disso, a natureza open-source do licenciamento para criar, usar ou distribuir aplicativos R, contribuiu para a sua aceitação tanto no meio acadêmico como nas empresas. A implantação de aplicativos R oferece benefícios como permitir a automação de tarefas, integração com outras ferramentas, reprodutibilidade e colaboração, e interatividade.

Automatizar tarefas ou fluxos de trabalho repetitivos pode economizar tempo e esforço das equipes de cientistas e engenheiro de dados, eliminando a necessidade de processamento e análise manual de dados (DE BIE et al., 2022). Ao automatizar essas tarefas, os profissionais podem se concentrar em análises mais complexas ou em outros aspectos importantes do trabalho. Aplicações desenvolvidas em R podem ser integradas a outras ferramentas e tecnologias, como bancos de dados, APIs ou outras linguagens de programação, permitindo aproveitar recursos complementares de

diferentes sistemas e criar soluções mais robustas (HAGHISH, 2019; OANCEA; DRAGOESCU, 2014). Membros da equipe das empresas ou pesquisadores, podem trabalhar de forma colaborativa, compartilhando suas soluções de modo que outras pessoas possam acessar e reutilizar facilmente o código, facilitando o trabalho conjunto em projetos e o compartilhamento de ideias (KASPRZAK et al., 2021). E, finalmente, destaca-se a implantação de aplicativos com criação de interfaces Web, viabilizado por meio do pacote Shiny, que facilita a interação de usuários não técnicos com as análises e visualizações criadas, sendo particularmente útil ao apresentar resultados a partes interessadas ou clientes que podem não estar familiarizados com R ou programação (ELLIOTT; ELLIOTT, 2020).

Embora a natureza interativa e colaborativa das aplicações implementadas em linguagem R possam ser uma maneira de compartilhar as análises de dados e trabalho de modelagem, ainda há alguns desafios relacionados à distribuição de soluções criadas utilizando essa ferramenta. Em geral, as organizações iniciam iniciativas de ciência de dados como projetos exploratórios que buscam obter novos insights a partir dos dados (SARKER, 2021). O R, com as milhares de bibliotecas voltadas ao tratamento e análise de dados, juntamente com o pacote Shiny, acelerada a visualização de resultados. O próximo passo lógico nesse fluxo seria iniciar uma implementação da solução em produção, convertendo as análises exploratórias e modelagens realizadas na máquina do desenvolvedor em uma aplicação de nível organizacional que a torne mais visível e amplamente utilizada. Mas viabilizar aplicações R em produção tem limitações que precisam ser superadas. O primeiro aspecto é que o R é uma linguagem single-thread, o que significa que ele só pode lidar com um usuário por vez utilizando um único processo R para execução da aplicação (CHENG et al., 2016). Isso pode ser um problema quando vários usuários acessam o aplicativo simultaneamente. À medida que o número de usuários aumenta, evidencia-se problemas com o desempenho e a escalabilidade de aplicativos Shiny R. Além disso, o R não possui uma forma padronizada de implantar aplicativos, sendo necessário, portanto, que a equipe de desenvolvimento selecione e configure sua própria infraestrutura, seja um servidor, um contêiner Docker ou um serviço em nuvem. Assim,

este trabalho tem o objetivo de investigar abordagens de disponibilização em nuvem de aplicações multiusuários em R.

## **História da Linguagem R**

A linguagem de programação R foi criada por Ross Ihaka e Robert Gentleman na Universidade de Auckland, Nova Zelândia, no início dos anos 1990, com a primeira versão lançada em 1995. O R foi o sucessor da linguagem S, desenvolvida pela AT&T para computação estatística na década de 70 sob supervisão de John Chambers (CHAMBERS, 2020).

Desenvolvido sob licença pública GNU, a linguagem R ganhou popularidade tanto na academia quanto na indústria devido à sua versatilidade e à disponibilidade de inúmeros pacotes que implementam técnicas estatísticas e de análise de dados. Ao longo dos anos, R continuou a evoluir e melhorar. Atualizações regulares e novos pacotes são lançados todos os dias para aprimorar sua funcionalidade e atender às necessidades dos usuários. A comunidade R permanece ativa, com usuários contribuindo para o desenvolvimento de novos pacotes, compartilhando conhecimento através de fóruns e conferências e fornecendo suporte a outros usuários.

## **Aplicabilidade da Linguagem R**

Um dos principais pontos fortes do R é sua extensa coleção de pacotes. Esses pacotes são contribuídos por uma comunidade grande e ativa de usuários e desenvolvedores e fornecem funcionalidades e ferramentas adicionais para vários domínios, como manipulação de dados, aprendizado de máquina, visualização e muito mais. O Comprehensive R Archive Network (CRAN) é o repositório principal para pacotes R, hospedando milhares de pacotes que podem ser facilmente instalados e usados dentro do R. Com isso, a linguagem de programação R encontra aplicabilidade em diversas áreas das ciências e indústrias. Em finanças, o R é utilizado para tarefas como modelagem de risco (PFAFF, 2016), análise de portfólio (BOHMAN; FASTH, 2019; CAO, 2021; PICERNO, 2018), análise de séries temporais (PALOMAR, 2020) e econometria (CHEN; CHEN; YAO, 2020; GAO; SHI, 2021; ZHOU; KANG, 2021)).

Oferece pacotes para manipulação de dados financeiros e modelagem econométrica (LEE, 2022; LEE; LEE, 2022; RYAN et al., 2023). No campo da bioinformática, focado em análise de dados biológicos, como sequenciamento de DNA (CHAO et al., 2021), expressão gênica (PARADIS, 2022) e estrutura de proteínas (LILES, 2022). Com recursos de visualização de dados, o R é aplicado em áreas que exigem visualização de dados, como marketing (YILDIRIM; KÜBLER, 2023), jornalismo (MACHLIS, 2019) e inteligência de negócios (FERNANDES; ALMEIDA; MENDES, 2019). Com o foco em análise de dados e modelagem estatística, o R é aplicado na áreas de Agricultura (PISSINATO et al., 2022), Segurança pública (ROZZI, 2021; WANG et al., 2023), Transportation (GOLISZEK, 2021), Cyber segurança (WANG; JONES, 2021), Engenharia (KESEK, 2021; RIZA et al., 2021), Ciências sociais (DION et al., 2021; PURZYCKI; JAMIESON-LANE, 2017; ZHANG et al., 2022), História (KHANNA; BHARDWAJ; KHURANA, 2019). Estes são apenas alguns exemplos, e R também é usado em outras áreas, como ciências ambientais (SILLERO et al., 2023), análise geoespacial (KURIHARA; ISHIOKA, 2023) e pesquisa em saúde (DAVIS; SAUNDERS, 2020; ZHAO; LI, 2023). Assim, sua versatilidade e amplo ecossistema de pacotes o tornam uma escolha para tratamento, análise e visualização de dados em diversos domínios de aplicação.

## **Shiny R**

Shiny é um framework de desenvolvimento que possibilita criar aplicativos web interativos usando a linguagem R, sem exigir conhecimento de HTML, CSS ou JavaScript (KASPRZAK et al., 2021). Esses aplicativos são projetados para permitir que analistas de dados, pesquisadores e desenvolvedores compartilhem seus insights, visualizações de dados e modelos estatísticos com um público mais amplo de maneira interativa e fácil de usar. Os aplicativos Shiny R podem ser acessados por meio de navegadores da web, facilitando a interação dos usuários com os dados e a exploração de vários cenários sem a necessidade de conhecer a programação R.

A estrutura de um aplicativo Shiny R começa pela definição da interface do usuário (IU), empregando funções R que criam controles de entrada, como caixas de texto e botões, e elementos de saída, como gráficos, tabelas e texto. A lógica do

servidor é onde se define como o aplicativo responde às entradas do usuário e gera saídas. É nesse escopo que entra em jogo o paradigma de programação “reativa” (SALVANESCHI; MARGARA; TAMBURRELLI, 2015), que é um conceito central no Shiny (KASPRZAK et al., 2020). Expressões e funções reativas garantem que as saídas sejam atualizadas automaticamente sempre que as entradas forem alteradas. Assim, quando os usuários interagem com o aplicativo ajustando os controles de entrada, as expressões reativas na lógica do servidor detectam automaticamente essas alterações e acionam as atualizações apropriadas nos elementos de saída. Isso cria uma experiência de usuário dinâmica e responsiva. Com isso, é possível criar visualizações de dados interativas, gráficos dinâmicos que permite ajuste de parâmetros e filtros, gerar relatórios dinâmicos onde os usuários podem escolher quais dados e visualizações incluir, adaptando o relatório às necessidades. Dessa forma, o Shiny R preenche a lacuna entre a análise, exploração, visualização e modelagem de dados e a interação do usuário. A Figura 1 apresenta a interface de um aplicativo Shiny R.

Figura 1: Interface de Aplicativo criado em Shiny R



## Metodologia

Para investigar abordagens de disponibilização em nuvem de disponibilização de aplicações em R, realizou-se uma revisão da literatura científica nas bases de dados da Web of Science e Scopus. A escolha dessas bases de dados foi baseada em critérios de qualidade (ŠUBELJ et al., 2015) e relevância para o campo da computação e completude dos metadados retornados por esses repositórios. A questão direcionadora é: como a linguagem R está sendo utilizada em conjunto com tecnologias de implementação de aplicação em nuvem? A motivação é identificar plataformas de software que permite criar, testar e implantar aplicativos R na nuvem. Por isso, construiu-se as *strings* de busca apresentadas na Tabela 1.

Tabela 1: *Strings* de busca

# String	String de busca	Justificativa
1	R AND (programming OR comput* OR software OR package* OR librar*) AND (docker* OR container* OR cloud* OR virtual machine*)	Busca geral para entender o contexto amplo de associação da linguagem R com tecnologias de implementação de aplicação em nuvem.
2	R AND (programming OR comput* OR software OR package* OR librar*) AND (docker* OR container* OR cloud* OR virtual machine*) AND deploy*	O termo deploy* foi adicionado para direcionar os resultados focados na implementação das soluções desenvolvidas.

Foi feita uma tentativa de incluir termos relacionado a multiusuários, mas as buscas tentadas retornaram 0 resultados. A Tabela 2 apresenta os critérios de inclusão e exclusão definidos. O intervalo de tempo de quase 6 anos foi definido para garantir que os resultados descrevam o cenário mais atual e o amadurecimento das tecnologias.

Tabela 2: Critérios de inclusão e exclusão

Critérios de inclusão		Critérios de exclusão
1	Publicados entre janeiro 2017 e agosto 2023	Publicações fora do intervalo de tempo definido
2	Texto completo disponíveis	Não escritos em inglês
3	Publicados em revistas científicas e conferências	Livros

As buscas por tópico da primeira *string* de buscas, aplicando os critérios de inclusão e exclusão, retornaram 2016 artigos na Web of Science e 447 na Scopus, com 226 artigos repetidos no cruzamento dos resultados. A segunda *string* de busca, mais restritiva, resultou em 138 na Web of Science e 39 na Scopus, com uma intersecção de 23 artigos.

Para a análise dos resultados, fez-se uma avaliação dos metadados dos resultados. Ordenou-se os artigos a partir dos critérios de relevância específica de cada base e analisou-se título, resumo e termos chave dos 50 artigos mais bem ranqueados em cada base. Em virtude da restrição de tempo, poucos artigos foram selecionados para leitura completa. Mas para realizar um levantamento do panorama geral dos resultados das buscas, utilizou-se a análise de redes dos termos chave dos autores, e plotou-se os resultados utilizando a biblioteca *igraph* da linguagem R.

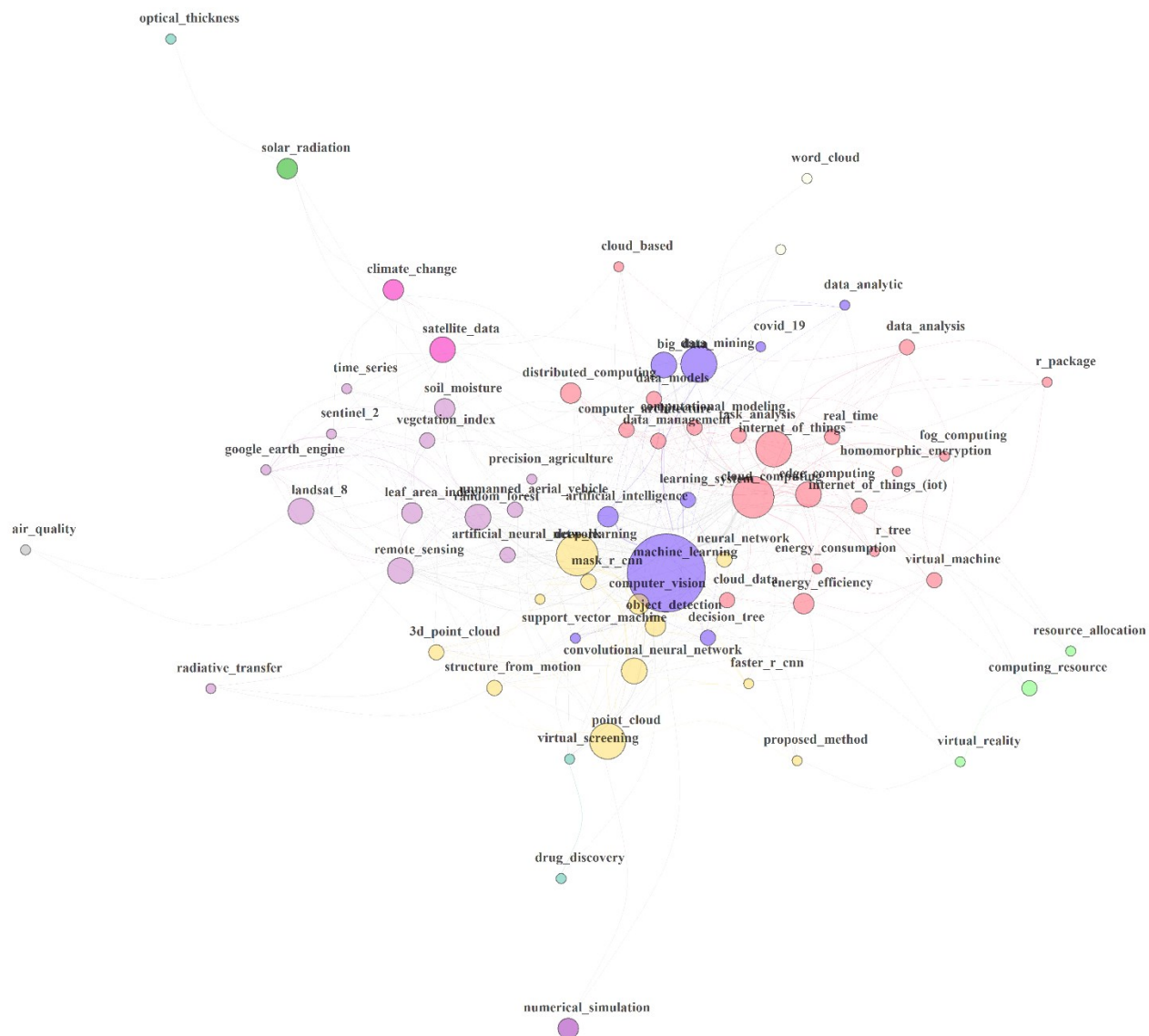
Para complementar os resultados da literatura científica, optou-se por incluir literatura cinzenta incorporada a partir de critérios de relevância para a comunidade de pesquisadores e desenvolvedores R. Assim, também se investigou artigos publicados no portal oficial R Project, no repositório R Bloggers e no provedor oficial de soluções R, Posit Cloud.

## **Análise de Redes**

A análise de redes a partir do agrupamento de termos chaves dos autores é uma forma de mapear e identificar tendências de pesquisas da literatura científica (LEE; CHUNG, 2022; YAN; ZHIPING, 2023). Assim, a Figura 2 mostra a rede gerada a partir dos resultados da *string* de busca 1.



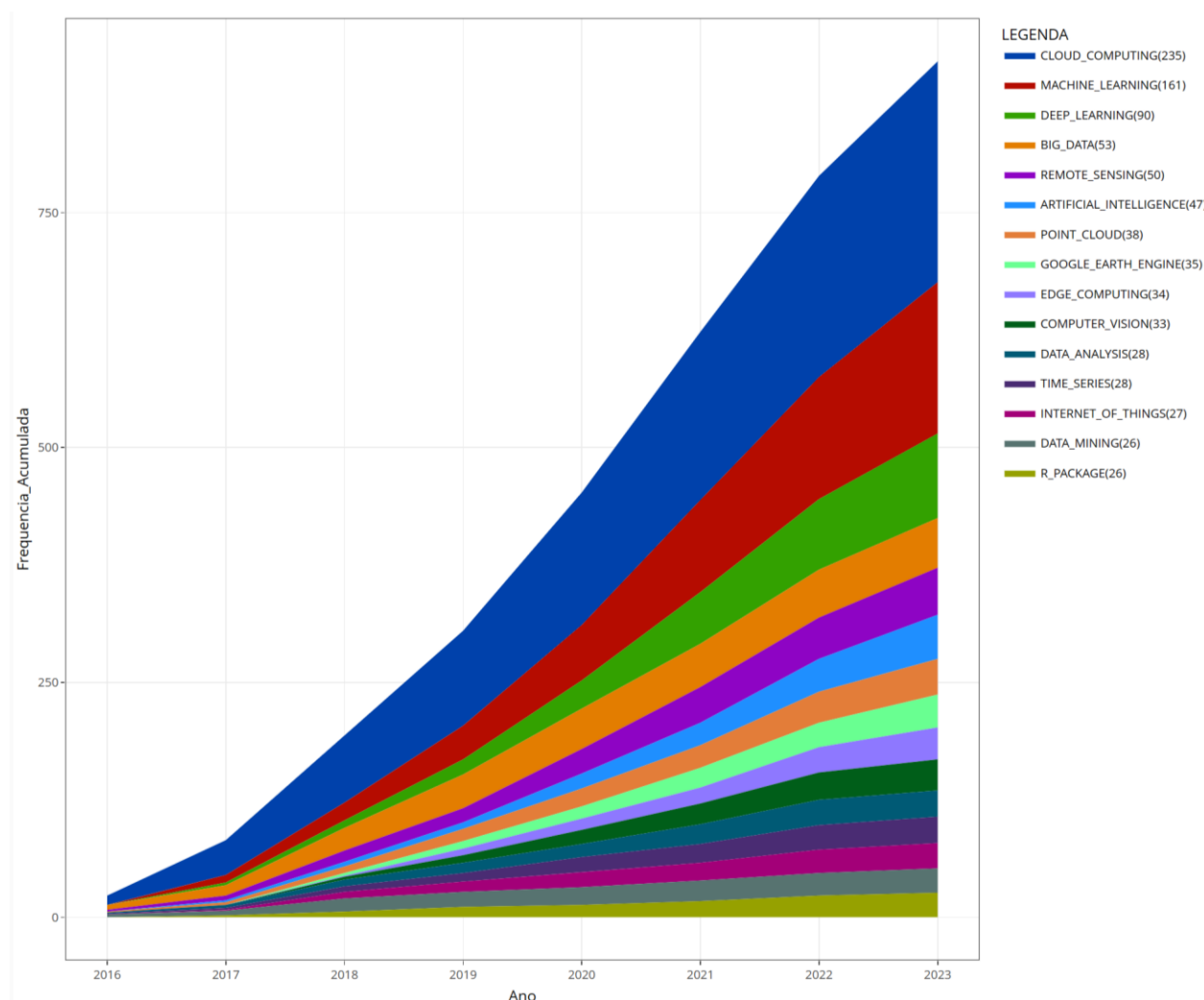
Figura 2: Rede de coocorrência de termos chaves dos autores



A rede apresenta alguns agrupamentos temáticos. Analisando a rede, percebe-se uma tendência de os artigos serem mais focados na aplicabilidade da linguagem R sendo implementada na nuvem voltando a um contexto de pesquisa específico. Por isso, há termos de áreas tão distintas como `solar_radiation` (climatologia), `drug_delivery` (farmacêutica), `energy_efficiency` (engenharia), `precision_agriculture` (agricultura), `3d_point_cloud` (mapeamento tridimensional). Os artigos focados mais na parte prática de implementação de soluções R na nuvem, foram poucos. A tendência era mesmo a

de utilizar a linguagem R para algum tipo de análise específica do contexto de aplicação, como em (CROCI; CAMPANER, 2023; FLORENSA et al., 2023; JIA et al., 2022; KASS et al., 2023), ou com eventual disponibilização na nuvem, como nos trabalhos de (LI, 2020; MULLER et al., 2022). A Figura 3 apresenta um gráfico de frequência acumulada dos 15 termos chave mais relevantes nos resultados da segunda *string* de busca.

Figura 3: Frequência acumulada dos termos chave dos autores



Os resultados da *string* 2 ficaram mais focados nos aspectos técnicos de implementação de soluções em nuvem, com alguns artigos discutindo virtualização (BHARDWAJ; RAMA KRISHNA, 2022) desempenho (BRONDOLIN; SANTAMBROGIO,

2020), arquitetura (KRISHNA; SRINIVAS; REDDY, 2023) e segurança (MANGALAGOWRI; VENKATARAMAN, 2023). Os trabalhos mais alinhados ao objetivo do trabalho com a intersecção da implementação de aplicações Shiny R e disponibilização em nuvem e containers são (HU; PAUNIC, 2020; LI, 2020; NUST et al., 2020).

## **Disponibilização de aplicações Shiny R**

O compartilhamento tradicional de um aplicativo Shiny como página Web exige que o usuário tenha o R e Shiny instalados em sua máquina local. No entanto, é importante viabilizar o compartilhamento dos resultados de uma aplicação para pessoas que não possuem o R e não tem intenção de instalá-lo. Para isso é necessário disponibilizar as aplicações em servidores que gerenciem as requisições de acesso dos usuários. Identificou-se na literatura 4 maneiras principais de hospedar o aplicativo Shiny como uma página web na nuvem:

### **Shinyapps.io**

Serviço de plataforma como serviço (PaaS) do RStudio para aplicativos Shiny que permite carregá-los diretamente da sessão R para o servidor de hospedagem (KIPP, 2017). Com o Shinyapps.io, a equipe de desenvolvimento não precisa se preocupar com instalação ou configuração de servidor, pois a manutenção do servidor é feita pelo próprio provedor, que fornece até um painel para monitorar as métricas/logs do aplicativo hospedado. A desvantagem dessa abordagem é que no plano mais robusto, o servidor atende a 10 mil horas ativas mensais, que é o tempo de execução das aplicações. Uma organização com muitos usuários e muitas aplicações simultâneas, pode ultrapassar esse limiar dentro do período de 30 dias.

### **Shiny Server**

Shiny Server é um programa servidor gratuito, de código aberto, que cria um servidor web projetado para hospedar aplicativos Shiny R (WICKHAM, 2021). Para usar o Shiny Server, é necessário um servidor Linux com suporte para Ubuntu ou CentOS. É possível hospedar vários aplicativos Shiny em várias páginas da web com o mesmo

servidor Shiny e proteger os aplicativos por meio de um firewall (POSIT, 2022). Com o Shiny Server, cada aplicativo tem seu próprio endereço da web e iniciará automaticamente o aplicativo quando um usuário visitar o endereço.

Utilizando o Shiny Server é possível hospedar quantos aplicativos forem necessários e o número de horas de atividade é irrestrito. No entanto, só é possível atender 20 sessões por vez, o que é um problema quando se trabalha com equipes maiores. A utilização do Shiny Server demanda também um entendimento sobre configuração de servidores.

### **POSIT Connect**

Em um contexto corporativo, é interessante disponibilizar os aplicativos Shiny R empregando ferramentas como autenticação de senha, suporte SSL e ferramentas de administrador. Para esse cenário tem-se o POSIT Connect, uma plataforma de colaboração e publicação de trabalho de análise de dados, voltado a equipes que constroem aplicativos em R, publica relatórios em RMarkdown, e criam dashboards, gráficos e Jupyter notebooks (POSIT, 2023). Com essa solução comercial, o servidor inicia um processo R para cada solicitação do usuário. A implementação permite o dimensionamento elástico e o disparo de vários processos Shiny para os usuários simultâneos. Como não há limitação no número de processos, é possível utilizar todos os recursos da máquina. Essas vantagens vêm com um custo elevado. As assinaturas de ofertas comerciais podem incorrer em taxas de licenciamento que partem de US\$ 23 mil/ano para 20 usuários simultâneos.

### **ShinyProxy**

ShinyProxy é uma solução de código aberto para disponibilização de aplicativos Shiny cuja arquitetura é baseada em contêineres docker, que isolam o ambiente da aplicação. A principal diferença em relação ao Shiny Server é que o ShinyProxy inicia uma nova instância de aplicativo para cada novo usuário permitindo escalar uma mesma aplicação para atender muitos usuários simultaneamente. Assim, essa alternativa permite uma hospedagem ilimitada de aplicativos e um número ilimitado de usuários simultâneos, sendo altamente escalável. Porém, como para cada usuário o

ShinyProxy inicializa um contêiner, ele pode consumir muita memória RAM do sistema/servidor e o usuário pode ter que esperar até que o contêiner seja inicializado e a aplicação carregada (SHINYPROXY, 2022).

### **Escalabilidade de aplicações Shiny R na nuvem**

Analizando as opções de disponibilização de aplicações Shiny R, percebe-se o destaque à limitação de escalabilidade que essa tecnologia possui. O kernel da linguagem R é de thread único, monotarefa, ou seja, não possui paralelismo e aciona apenas um núcleo de CPU por sessão (NAGLER, 2021). Em R, uma sessão refere-se ao período de tempo durante o qual o código R é executado. Durante uma sessão, um processo de execução atende o usuário para executar código R, carregar pacotes, definir variáveis e realizar diversas tarefas de análise e visualização de dados. Uma aplicação típica em Shiny R instancia um único processo para atender a todas as solicitações do usuário, o que restringe o número de usuários simultâneos que a aplicação pode suportar. Por isso, essa é uma limitação inerente da linguagem que impacta a escalabilidade dos aplicativos.

Componentes característicos de soluções em nuvem, como contêineres, máquinas virtuais, Dockers, Kubernetes e microsserviços podem ser empregados para construir aplicações modulares e disponibilizar infraestrutura escalável para atender aplicações Shiny R. Uma arquitetura nativa de nuvem oportuniza o aproveitamento do poder da virtualização para lidar com o aumento da carga de trabalho em função do número de usuários e do processamento de dados.

Os contêineres são uma forma de tecnologia de virtualização que permite o empacotamento, distribuição e execução de aplicativos e suas dependências em ambientes isolados (SILVA; KIRIKOVA; ALKSNIS, 2018). De acordo com (PATRA; SAHOO; TURUK, 2022), os contêineres isolam os aplicativos e seus ambientes de execução uns dos outros e do sistema hospedeiro, encapsulando tudo que um aplicativo precisa para ser executado, incluindo código, bibliotecas do sistema e configurações. Como os contêineres compartilham o kernel do sistema operacional hospedeiro, eles se tornam leves e eficientes em termos de recursos em comparação com máquinas virtuais (VMs) tradicionais (ABDULLAH; IQBAL; BUKHARI, 2019). Essa

eficiência permite a execução de vários contêineres no mesmo host sem sobrecarregar sobremaneira o servidor. Os contêineres usam um formato padronizado para empacotar aplicativos e suas dependências. Este formato, muitas vezes baseado no formato de imagem Docker, inclui tudo o que é necessário para executar a aplicação, garantindo um comportamento consistente (NÜST et al., 2020b).

A plataforma de containerização mais conhecida é o Docker, que popularizou o conceito de contêineres e introduziu o formato de imagem Docker. Ele usa tecnologia de containerização para agrupar um aplicativo e suas dependências em um único objeto chamado contêiner. Os contêineres podem ser facilmente replicados e orquestrados para dar suporte ao escalonamento horizontal de aplicativos (ROSSI; NARDELLI; CARDELLINI, 2019), o que é particularmente útil para aplicativos R que as vezes precisam processar grandes conjuntos de dados ou oferecer suporte a muitos usuários. E como os contêineres são leves e iniciam rapidamente se comparados às VMs, tornam-se ideais para implantações onde os aplicativos precisam ser iniciados e interrompidos dinamicamente em resposta às demandas dos usuários. Para aplicativos Shiny, o projeto Rocker fornece um conjunto de imagens Docker especificamente adaptadas para R (NÜST et al., 2020a).

Kubernetes é uma ferramenta de orquestração utilizada para gerenciar e dimensionar aplicativos em contêineres. A orquestração simplifica tarefas como escalabilidade, balanceamento de carga e recuperação automática, facilitando o gerenciamento de aplicações containerizadas em escala (YEPURI et al., 2023). Já a arquitetura de microsserviços é um estilo arquitetural no qual um aplicativo é estruturado como uma coleção de serviços fracamente acoplados e implementáveis de forma independente (MASSAGA; KOUAMOU, 2021).

Combinando todos esses componentes, é possível projetar uma arquitetura de solução para aplicativos Shiny R que seja multiusuário e escalável. Se a aplicação Shiny R for complexa e consistir em vários componentes que podem ser desenvolvidos e implantados de forma independente, pode-se optar por estruturá-la como um conjunto de microsserviços. Cada microsserviço pode ser implantado em seu próprio contêiner, o que fornece isolamento e permite que cada microsserviço seja dimensionado de forma independente. Já o Kubernetes pode dimensionar automaticamente os aplicativos Shiny

R com base na carga de trabalho. Isso é particularmente útil se o aplicativo tiver tráfego variável. Com isso garante-se que o aplicativo permaneça responsivo durante horários de pico, sem provisionar recursos em excesso fora dos horários de pico. O Kubernetes pode reiniciar automaticamente contêineres que falham, substituir e reprogramar contêineres quando os nós morrem e eliminar contêineres que não respondem à verificação de integridade. Os contêineres proporcionam a reprodutibilidade, portabilidade e escalabilidade necessárias para as aplicações Shiny R.

## **Conclusão**

A natureza open-source do R e seu ecossistema diversificado proporcionam vantagens significativas na análise ágil de dados e na visualização de resultados. Por isso ele é utilizado em diversas áreas que precisam processar e analisar dados, como a área científica, varejo, marketing, medicina e finanças. No entanto, a distribuição de soluções desenvolvidas com R enfrenta desafios, especialmente devido à sua natureza single-thread, que limita a capacidade de atender a múltiplos usuários simultaneamente.

Neste trabalho, apresentou-se 4 alternativas para hospedar os aplicativos Shiny R na nuvem: Shinyapps.io, Shiny Server, POSIT Connect e ShinyProxy. Destas alternativas, o ShinyProxy é a opção open-source não comercial que viabiliza a disponibilização dos aplicativos multiusuários em nuvem a partir de containers. Os elementos típicos de soluções em nuvem, como contêineres, Docker, Kubernetes e microsserviços, podem ser utilizados para construir uma arquitetura de implementação que garanti a escalabilidade de aplicações Shiny R.

Na busca e leitura da literatura científica, ainda que não tenha sido sistemática, identificou-se algumas oportunidades de pesquisa que poderiam ser exploradas para melhorar a compreensão sobre a disponibilização de aplicações implementadas em R na nuvem. Uma área de pesquisa com foco no desenvolvimento está relacionada a otimização de desempenho de aplicativos Shiny R em ambientes containerizados. Investigar técnicas para reduzir o consumo de recursos computacionais, melhorar a responsividade e reduzir os tempos de carregamento pode resultar em aplicações mais eficientes e rápidas, garantindo uma melhor experiência dos usuários. Um outro

aspecto está relacionado ao gerenciamento de recursos, explorando tópicos relacionados a alocação dinâmica de recursos, escalabilidade automática com base na demanda e otimização de custos para garantir um uso eficiente dos recursos da nuvem. A implantação multicloud é uma área de pesquisa que tem ganhado tração nos últimos anos. Investigar estratégias de implantação multicloud para aplicativos Shiny R pode ser uma área promissora. Isso permitiria a distribuição de aplicativos em várias nuvens, oferecendo flexibilidade para escolher a infraestrutura mais adequada e de melhor custo-benefício. E, finalmente, uma área com foco na implementação da infraestrutura deve envolver a integração de pipelines de CI/CD (Continuous Integration/Continuous Deployment) específicos para aplicativos Shiny e a automação de processos de escalonamento horizontal.

## Referências

- ABDULLAH, M.; IQBAL, W.; BUKHARI, F. Containers vs Virtual Machines for Auto-scaling Multi-tier Applications Under Dynamically Increasing Workloads. Em: [s.l: s.n.]. p. 153–167.
- BHARDWAJ, A.; RAMA KRISHNA, C. A Container-Based Technique to Improve Virtual Machine Migration in Cloud Computing. **IETE Journal of Research**, v. 68, n. 1, p. 401–416, 2 jan. 2022.
- BOHMAN, S.; FASTH, T. A web-based visualization tool for exploring stakeholder conflicts in land-use planning. **Transactions in GIS**, p. tgis.12527, abr. 2019.
- BRONDOLIN, R.; SANTAMBROGIO, M. D. A Black-box Monitoring Approach to Measure Microservices Runtime Performance. **ACM Transactions on Architecture and Code Optimization**, v. 17, n. 4, p. 1–26, 31 dez. 2020.
- CAO, M. Predicting the Link between Stock Prices and Indices with Machine Learning in R Programming Language. **Journal of Mathematics**, v. 2021, p. 1–10, dez. 2021.
- CHAMBERS, J. M. S, R, and data science. **Proceedings of the ACM on Programming Languages**, v. 4, n. HOPL, 12 jun. 2020.
- CHAO, K.-H. et al. sangeranalyseR: Simple and Interactive Processing of Sanger Sequencing Data in R. **Genome Biology and Evolution**, v. 13, n. 3, 1 mar. 2021.



CHEN, J.; CHEN, D.; YAO, A. Trade development between China and countries along the Belt and Road: A spatial econometric analysis based on trade competitiveness and complementarity. **Pacific Economic Review**, v. 25, n. 2, p. 205–227, 28 maio 2020.

CHENG, X. et al. Research on the System of Parallel Computing in R. **International Conference on Electrical, Mechanical and Industrial Engineering**, 2016.

CROCI, O.; CAMPANER, S. ChroKit: a Shiny-based framework for interactive analysis, visualization and integration of genomic data. **Nucleic Acids Research**, v. 51, n. W1, p. W83–W92, 5 jul. 2023.

DAVIS, J. A.; SAUNDERS, R. Impact of weight trajectory after bariatric surgery on co-morbidity evolution and burden. **BMC Health Services Research**, v. 20, n. 1, p. 278, 3 dez. 2020.

DE BIE, T. et al. Automating data science. **Communications of the ACM**, v. 65, n. 3, p. 76–87, 23 mar. 2022.

DION, K. et al. La modélisation par équations structurelles -- Un guide d'accompagnement pour l'interface R. **The Quantitative Methods for Psychology**, v. 17, n. 3, p. 198–271, 1 set. 2021.

ELLIOTT, M. S.; ELLIOTT, L. M. Developing R Shiny Web Applications for Extension Education. **Applied Economics Teaching Resources**, v. 2, n. 4, out. 2020.

FERNANDES, B.; ALMEIDA, J. A.; MENDES, M. **Near Real Time Business Intelligence Framework using R Shiny**. ago. 2019.

FLORENSA, D. et al. Exploring Cancer Incidence, Risk Factors, and Mortality in the Lleida Region: Interactive, Open-source R Shiny Application for Cancer Data Analysis. **JMIR Cancer**, v. 9, p. e44695, 20 abr. 2023.

GAO, Z.; SHI, Z. Implementing Convex Optimization in R: Two Econometric Examples. **Computational Economics**, v. 58, n. 4, p. 1127–1135, 2 dez. 2021.

GOLISZEK, S. GIS tools and programming languages for creating models of public and private transport potential accessibility in Szczecin, Poland. **Journal of Geographical Systems**, v. 23, n. 1, p. 115–137, 26 jan. 2021.

HAGHISH, E. F. Seamless interactive language interfacing between R and Stata. **The Stata Journal: Promoting communications on statistics and Stata**, v. 19, n. 1, p. 61–82, 14 mar. 2019.

HU, C.; PAUNIC, V. **Building Forecasting Solutions Using Open-Source and Azure Machine Learning**. Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. **Anais...**New York, NY, USA: ACM, 23 ago. 2020.

JIA, L. et al. Development of interactive biological web applications with R/Shiny. **Briefings in Bioinformatics**, v. 23, n. 1, 17 jan. 2022.

KASPRZAK, P. et al. Six Years of Shiny in Research - Collaborative Development of Web Tools in R. **The R Journal**, v. 12, n. 2, p. 155, 2020.

KASPRZAK, P. et al. Six Years of Shiny in Research-Collaborative Development of Web Tools in R. **The R Journal**, 2021.

KASS, J. M. et al. wallace 2: a shiny app for modeling species niches and distributions redesigned to facilitate expansion via module contributions. **Ecography**, v. 2023, n. 3, 16 mar. 2023.

KĘSEK, M. Analysing Data with the R Programming Language to Control Machine Operation. **Inżynieria Mineralna**, v. 1, n. 1, 24 abr. 2021.

KHANNA, S.; BHARDWAJ, S.; KHURANA, A. Titanic Data Analysis by R Data Language for Insights and Correlation. Em: [s.l: s.n.]. p. 73–79.

KIPP, A. **Shiny - Getting started with shinyapps.io**. Disponível em: <<https://shiny.posit.co/r/articles/share/shinyapps/>>. Acesso em: 1 set. 2023.

KRISHNA, D. S.; SRINIVAS, G.; REDDY, P. V. G. D. P. Novel private cloud architecture: A three tier approach to deploy private cloud using virtual machine manager. **Intelligent Decision Technologies**, v. 17, n. 2, p. 275–285, 15 maio 2023.

KURIHARA, K.; ISHIOKA, F. Echelon analysis and its software for spatial lattice data. **WIRES Computational Statistics**, v. 15, n. 1, 12 jan. 2023.

LEE, J. Y.; CHUNG, E. Mapping open science research using a keyword bibliographic coupling analysis network. **Information Research: an international electronic journal**, v. 27, n. 4, 15 dez. 2022.

LEE, S. **Package ‘finreportr’**. Disponível em: <<https://cran.r-project.org/web/packages/finreportr/finreportr.pdf>>. Acesso em: 29 ago. 2023.

LEE, S.; LEE, M. S. Type Package Title Financial Data from U.S. Securities and Exchange Commission. 2022.

LI, Y. Towards fast prototyping of cloud-based environmental decision support systems for environmental scientists using R Shiny and Docker. **Environmental Modelling & Software**, v. 132, p. 104797, out. 2020.

LILES, S. **Perform Analysis and Create Visualizations of Proteins**. [s.l: s.n.].

MACHLIS, S. **Practical R for mass communication and journalism**. London, England: CRC Press, 2019.

MANGALAGOWRI, R.; VENKATARAMAN, R. Randomized MILP framework for Securing Virtual Machines from Malware Attacks. **Intelligent Automation & Soft Computing**, v. 35, n. 2, p. 1565–1580, 2023.

MASSAGA, A.; KOUAMOU, G. E. Towards a Framework for Evaluating Technologies for Implementing Microservices Architectures. **Journal of Software Engineering and Applications**, v. 14, n. 08, p. 442–453, 2021.

MULLER, C. et al. The Image Data Explorer: Interactive exploration of image-derived data. **PLOS ONE**, v. 17, n. 9, p. e0273698, 15 set. 2022.

NAGLER, T. R -Friendly Multi-Threading in C++. **Journal of Statistical Software**, v. 97, n. Code Snippet 1, 2021.

NÜST, D. et al. The Rockerverse: Packages and Applications for Containerisation with R. **The R Journal**, v. 12, n. 1, p. 437, 2020a.

NUST, D. et al. The Rockerverse: Packages and Applications for Containerisation with R. **R JOURNAL**, v. 12, n. 1, p. 437–461, jun. 2020.

NÜST, D. et al. Ten simple rules for writing Dockerfiles for reproducible data science. **PLOS Computational Biology**, v. 16, n. 11, p. e1008316, 10 nov. 2020b.

OANCEA, B.; DRAGOESCU, R. Integrating R and Hadoop for Big Data Analysis. **Romanian Statistical Review**, p. 83–94, set. 2014.

PALOMAR, D. P. **Time Series Modeling of Financial Data with R**. Hong Kong: [s.n.]. Disponível em: <[https://palomar.home.ece.ust.hk/MAFS5310\\_lectures/Rsession\\_time\\_series\\_modeling.html](https://palomar.home.ece.ust.hk/MAFS5310_lectures/Rsession_time_series_modeling.html)>. Acesso em: 29 ago. 2023.

PARADIS, E. **Population genomics with R**. Philadelphia, PA: Chapman & Hall/CRC, 2022.

PARRA, M. I. et al. Using R for teaching and research. 21 jun. 2023.

PATRA, M. K.; SAHOO, B.; TURUK, A. K. Containerization in cloud computing for OS-level virtualization. Em: **Advances in Cyber Security and Intelligent Analytics**. Boca Raton: CRC Press, 2022. p. 261–276.

PFAFF, B. **Financial Risk Modelling and Portfolio Optimization with R**. 2nd edition ed. Southern Gate, Chichester, West Sussex, UK: Wiley, 2016.

PICERNO, J. **Quantitative Investment Portfolio Analytics In R: An Introduction To R For Modeling Portfolio Risk and Return**. Erscheinungsort nicht ermittelbar: CreateSpace Independent Publishing Platform, 2018.

PISSINATO, B. et al. Using Time Series and New Information Technologies for Forecasting Sugarcane Production Indicators. Em: [s.l: s.n.]. p. 238–248.

POSIT. **Shiny - Welcome to Shiny**. Disponível em: <<https://shiny.posit.co/r/getstarted/shiny-basics/lesson1/index.html>>. Acesso em: 1 set. 2023.

POSIT. **Introduction - Posit Package Manager**. Disponível em: <[https://rstudio-package.ap.datim.org/\\_\\_docs\\_\\_/admin/](https://rstudio-package.ap.datim.org/__docs__/admin/)>. Acesso em: 1 set. 2023.

PURZYCKI, B. G.; JAMIESON-LANE, A. AnthroTools. **Cross-Cultural Research**, v. 51, n. 1, p. 51–74, 30 fev. 2017.

RIZA, L. S. et al. DEVELOPMENT AND EXPERIMENTATION OF R PACKAGE “metaheuristicOpt” ON CONTINUOUS OPTIMIZATION. **JOURNAL OF ENGINEERING SCIENCE AND TECHNOLOGY**, v. 16, n. 2, p. 1006–1018, abr. 2021.

ROSSI, F.; NARDELLI, M.; CARDELLINI, V. **Horizontal and Vertical Scaling of Container-Based Applications Using Reinforcement Learning**. 2019 IEEE 12th International Conference on Cloud Computing (CLOUD). **Anais...IEEE**, jul. 2019.

ROZZI, G. C. njtr1: An R package for researching road safety in New Jersey using open crash data. **Software Impacts**, v. 10, p. 100176, nov. 2021.

RYAN, J. A. et al. **CRAN - Package quantmod**. Disponível em: <<https://cran.r-project.org/web/packages/quantmod/index.html>>. Acesso em: 29 ago. 2023.

SALVANESCHI, G.; MARGARA, A.; TAMBURRELLI, G. **Reactive Programming: A Walkthrough**. 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering. **Anais...IEEE**, maio 2015.

SARKER, I. H. Data Science and Analytics: An Overview from Data-Driven Smart Computing, Decision-Making and Applications Perspective. **SN Computer Science**, v. 2, n. 5, p. 377, 12 set. 2021.

SHINYPROXY. **Getting Started | ShinyProxy**. Disponível em: <<https://www.shinyproxy.io/documentation/getting-started/>>. Acesso em: 1 set. 2023.

SILLERO, N. et al. A curated list of R packages for ecological niche modelling. **Ecological Modelling**, v. 476, p. 110242, fev. 2023.

SILVA, V. G. DA; KIRIKOVA, M.; ALKSNIS, G. Containers for Virtualization: An Overview. **Applied Computer Systems**, v. 23, n. 1, p. 21–27, 1 maio 2018.

ŠUBELJ, L. et al. Quantifying the Consistency of Scientific Databases. **PLOS ONE**, v. 10, n. 5, p. e0127390, 18 maio 2015.

WANG, J. et al. Research and Application of Improved Multiple Imputation Based on R Language in Fire Prediction. **Fire**, v. 6, n. 6, p. 235, 13 jun. 2023.

WANG, L.; JONES, R. Big Data Analytics in Cyber Security: Network Traffic and Attacks. **Journal of Computer Information Systems**, v. 61, n. 5, p. 410–417, 3 set. 2021.

WICKHAM, H. Mastering Shiny: build interactive apps, reports, and dashboards powered by R. p. 348, 8 jun. 2021.

YAN, L.; ZHIPING, W. Mapping the Literature on Academic Publishing: A Bibliometric Analysis on WOS. **SAGE Open**, v. 13, n. 1, p. 215824402311585, 4 jan. 2023.

YEPURI, V. K. et al. Containerization of a polyglot microservice application using Docker and Kubernetes. 30 abr. 2023.

YILDIRIM, G.; KÜBLER, R. **Applied marketing analytics using R**. London, England: SAGE Publications, 2023.

ZHANG, W. et al. Statistical Assumptions and Reproducibility in Psychology: Data Mining Based on Open Science. **Frontiers in Psychology**, v. 13, 30 maio 2022.

ZHAO, J.; LI, M. Worldwide trends in prediabetes from 1985 to 2022: A bibliometric analysis using bibliometrix R-tool. **Frontiers in Public Health**, v. 11, 13 fev. 2023.

ZHOU, S.; KANG, C. Research on the Influencing Factors of Russian Foreign Trade Based on R Language Regression Analysis. **Mathematical Problems in Engineering**, v. 2021, p. 1–11, 27 dez. 2021.