

## Aula 02

*Banco do Brasil (Escriturário - Agente de  
Tecnologia) Desenvolvimento de  
Software - 2023 (Pós-Edital)*

Autor:  
**Raphael Henrique Lacerda, Paolla  
Ramos e Silva**

05 de Janeiro de 2023

# Índice

1) Noções Iniciais de DevOps .....	3
------------------------------------	---



## Noções de DevOps

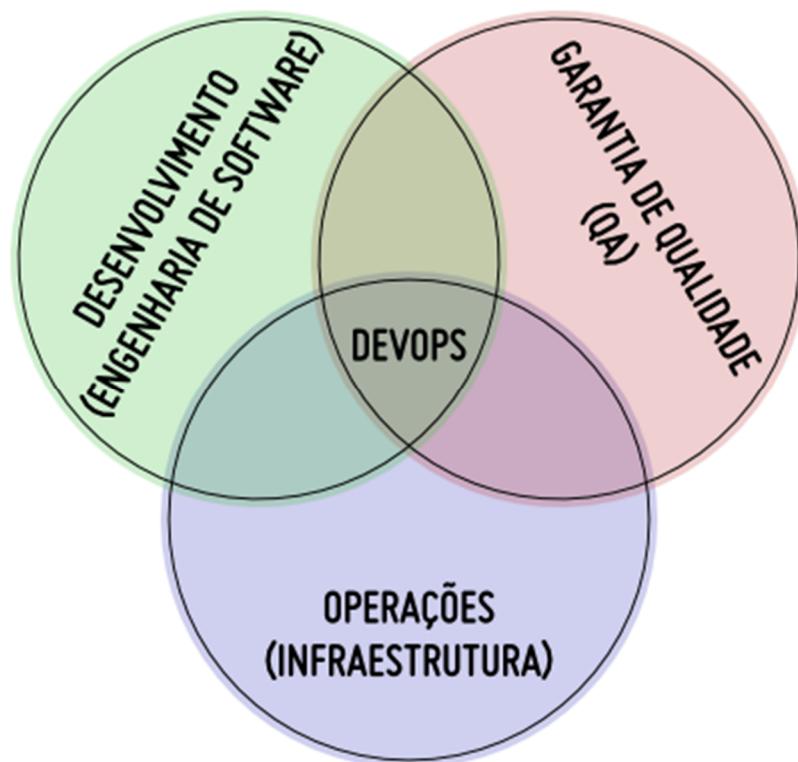
### Conceitos Básicos

INCIDÊNCIA EM PROVA: MÉDIA

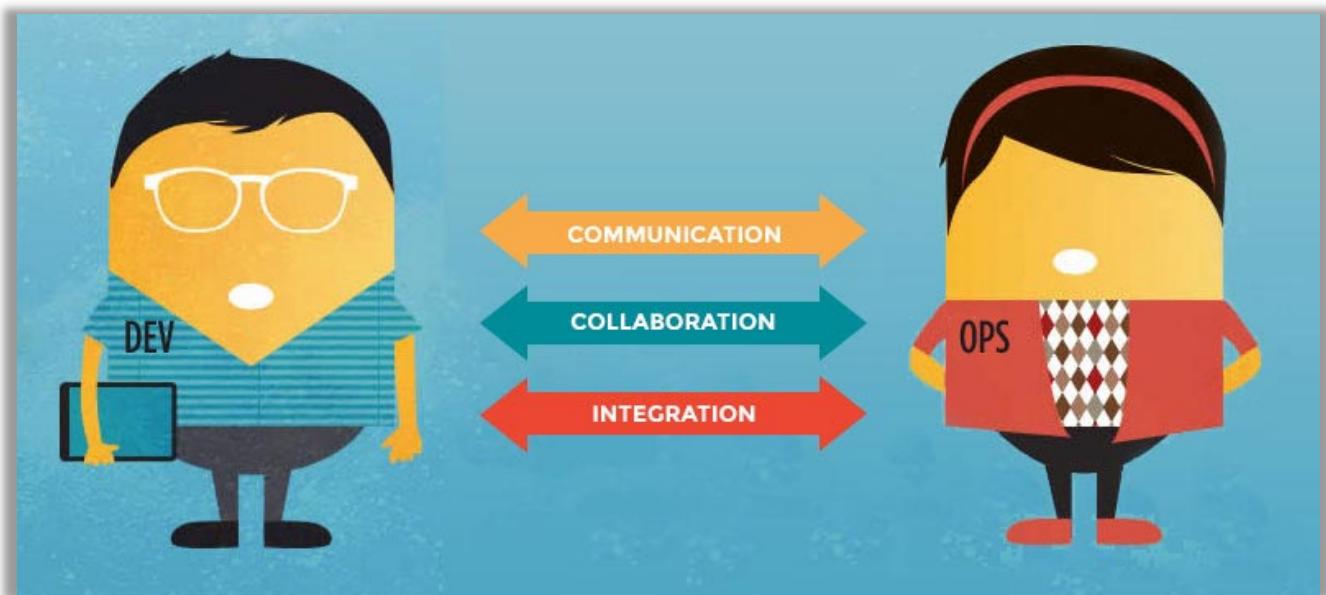
De um tempo para cá, nós temos vivenciado novos paradigmas em tecnologias de software. **As metodologias ágeis, por exemplo, vieram com um conjunto de práticas que desencadearam diversas outras práticas, de forma a obter software de maneira mais ágil e mais adaptável a possíveis mudanças.** No entanto, o grande lance é que as mudanças ocorrem com um intervalo de tempo cada vez menor.

Vocês já devem ter percebido isso! Antigamente, softwares lançavam atualizações em intervalos de 18 a 24 meses (ou até mais). **Atualmente, a dinâmica de consumo de aplicações de tecnologia da informação sofreu uma reviravolta e a demanda dos clientes é insana.** As empresas de software atualmente são duramente pressionadas para lançar e atualizar aplicativos no mercado o mais rápido possível.

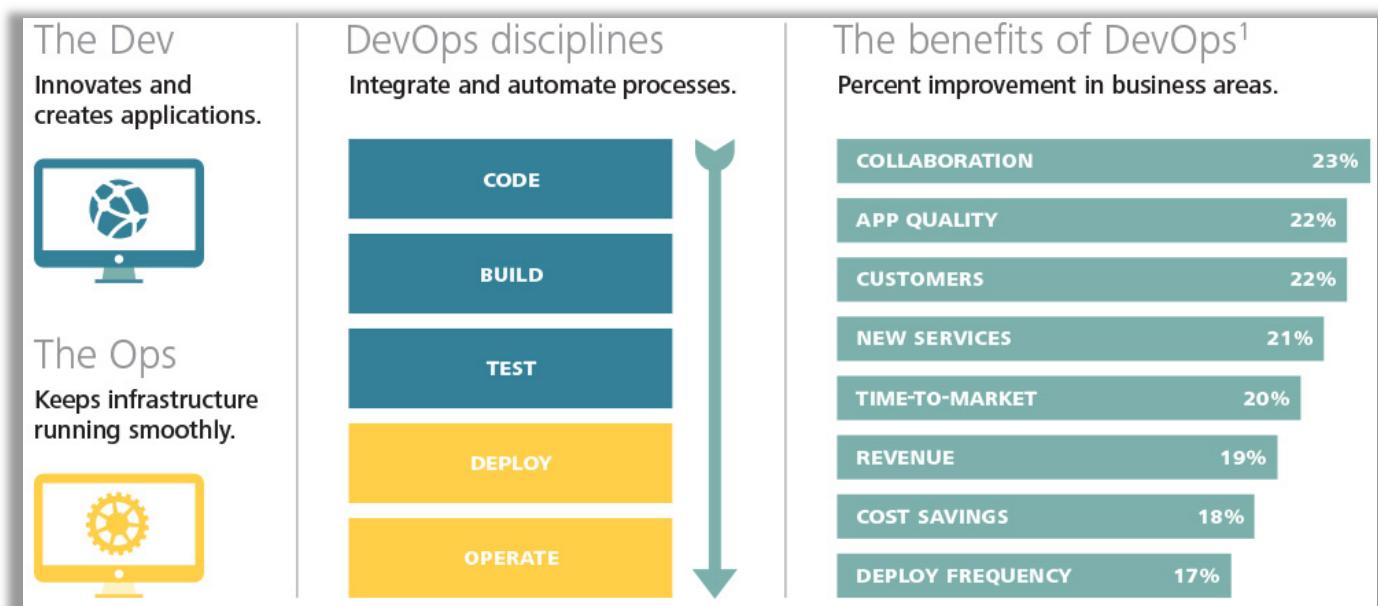
Pois é, o mundo mudou! O ciclo para a criação de novos aplicativos de software dura cerca de três meses para uma versão inicial e mais seis meses para o conjunto completo de recursos. **Não só o ciclo de vida foi encurtado, mas os aplicativos se tornaram muito mais complexos e exigem colaboração e integração cruzada entre os diversos componentes de tecnologia da informação, como Dev, Ops e Q&A.**



Professor, espera um pouco aí! O que acontece se eu juntar conceitos de Desenvolvimento de Software, Operação de Sistemas e Garantia de Qualidade? Surgirá, então, o conceito de DevOps! **Trata-se de um conceito muito simples, como apresenta a imagem acima. Essas ideias são tratadas em conjunto, em vez de separadas.** O lance é ter uma maior comunicação, colaboração e integração entre essas áreas.



Professor, o que é exatamente DevOps? É uma cultura? É uma técnica? É uma metodologia de desenvolvimento de software? Ainda não existe uma resposta precisa para essas perguntas! Por que, professor? **Porque foi um movimento que começou ao mesmo tempo em diversos lugares diferentes, tratando de infraestrutura e desenvolvimento, mas não houve um manifesto formal, como o manifesto ágil.**



**Parece simples fazer a infraestrutura conversar de forma harmônica com o desenvolvimento, mas não é tão fácil!** Qual é o papel da infraestrutura? É sustentar os sistemas em produção; monitorar o funcionamento e a performance; cuidar da estabilidade, segurança, níveis de serviço; planejar mudanças, minimizando riscos; entre outros. O que acontece se uma aplicação em produção parar de funcionar?



**Isso pode significar prejuízo financeiro ou institucional. Em suma, podemos dizer que a infraestrutura se preocupa em proteger o valor de negócio. E o desenvolvimento?** Esses caras se preocupam com inovação e criatividade, baseado nos requisitos do usuário. Desenvolvedor fica louco quando sai uma biblioteca, componente ou tecnologia nova. A consequência disso é que cada inovação significa um novo Deploy (feito pela rapaziada da infraestrutura).

**E se ocorrer algum problema?** Deve ser realizado um rollback (também pelo pessoal da infraestrutura). **Podemos afirmar, então, que o desenvolvedor se preocupa em aumentar o valor do negócio.** Vocês já devem ter notado que há um conflito interessante nessa conversa. Ora, o desenvolvedor quer colocar suas aplicações no ar o mais rápido possível para que fique disponível para o cliente.

**No entanto, a galera da infraestrutura quer ter certeza de que a aplicação está suficientemente estável para ir para produção sem gerar incidentes que parem o que já está funcionando. O que ocorria antigamente?** As empresas permitiam apenas um deploy por semana ou por mês! **Tem coisa mais não-ágil que isso?** Vai totalmente de encontro aos ideais do manifesto ágil. Lembram-se dos conceitos de entrega, integração e teste contínuos?

Pois é, a infraestrutura teve que se adaptar a realizar deploys diáridos. No entanto, os desenvolvedores – muitas vezes – se esqueciam de considerar algumas diferenças importantes entre ambientes de desenvolvimento e produção. **Isso gerava alguns incidentes, o cliente reclamava e começava uma briga muito comum representada pelas duas imagens anteriores.** Sim, galera... rola essa briga!





Desenvolvedores afirmam que a Infraestrutura é engessada, lenta e que não oferece um ambiente adequado para o desenvolvimento de aplicações; já a Infraestrutura afirmava que os desenvolvedores faziam código ruim e instável, e que a culpa não era deles. **Pessoal, voltem agora para a primeira imagem e percebam o que DevOps tenta integrar: Desenvolvimento, Infraestrutura e Qualidade!**

Parece briga de marido e mulher, mas ambos os lados têm que reconhecer seus erros, ceder e se adaptar! **O Desenvolvimento precisa pensar mais na Infraestrutura e controlar as fases de deploy (Ex: Deployment Pipeline). Já a Infraestrutura tem que evoluir para o mundo ágil.** Começar a trabalhar de forma automatizada e dinâmica, ser mais veloz para subir ambientes; reconstruir ou duplicar ambientes de acordo com as necessidades do Desenvolvimento.

**Galera, as principais características do DevOps são:** colaboração entre equipes; fim de divisões; relação saudável entre áreas; teste, integração e entrega contínuos; automação de deploy; controle e monitoração; gerenciamento de configuração; orquestração de serviços; avaliação de métricas e desempenho; logs e integração; velocidade de entrega; feedback intenso; e comunicação constante.

Em suma, podemos dizer que ele é um movimento, um conceito, uma cultura, uma abordagem que trata do feedback, comunicação e colaboração entre áreas de tecnologia da informação com o objetivo de garantir qualidade do software, com menor custo, mais rapidez, menor risco e maior eficiência. **É como se fosse a utilização de metodologias ágeis no desenvolvimento e na infraestrutura. Bacana?**

**Para o DevOps, o código gerado pela infraestrutura é apenas mais um artefato qualquer de desenvolvimento e, não, uma parte separada.** Trata-se de uma prática importante, já que não basta somente o desenvolvedor escrever o código do sistema, é necessário que a área de infraestrutura também atue para liberar, controlar e entregar a versão do sistema de forma contínua, periódica e preferencialmente automática.

# TODAY'S DEVOPS

DEV + OPS

PART DEV

PART OPS

## Application Performance

Modern developers use APM tools to decrease latency, have complete visibility into code, databases, caches, queues, and third-party services.

## End User Analytics

A great developer understands end users have the best feedback and analytics play an enormous part of understanding users. Developers are constantly monitoring end user latency and checking performance by devices and browsers.

## Quality Code

Developers need to ensure their deployments and new releases don't implode or degrade the overall performance.

## Code-Level Errors

When you have a large distributed application it is vital to lower MTTR by finding the root cause of errors and exceptions.



## Application Availability

The applications need to be up and running and it's Ops responsibility to ensure uptime and SLAs are in order.

## Application Performance

Classic Ops generally rely on infrastructure metrics - CPU, memory, network and disk I/O, etc. Modern Ops correlate all of those metrics with application metrics to solve problems 10x faster.

## End User Complaints

The goal is to know about and fix problems before end users complain, reduce the number of support tickets, and eliminate false alerts.

## Performance Analytics

Automatically generated baselines of all metrics help Ops understand what has changed and where to focus their troubleshooting efforts. Alerts based upon deviation from observed baselines improve alert quality and reduce alert noise.



Vamos resumir: **a preocupação é com os objetivos quase diametralmente opostos da galera de Desenvolvimento (DEV) e Operações (OPS)**. Os desenvolvedores querem programar novos recursos, melhorar o produto; corrigir bugs, etc. As operações querem colocar tudo em funcionamento e nunca mudar, já que as alterações causam novos erros, bugs, problemas de desempenho, entre outros.

O objetivo do DevOps é aliviar a tensão entre esses dois campos! **Ter pessoas de Operações nas trincheiras de Desenvolvimento é a principal maneira de alcançar esse objetivo**. Seu trabalho é facilitar ao máximo que desenvolvedores e operações façam o que precisam fazer. *Como assim?* Por exemplo: fornecendo ambientes idênticos de desenvolvimento, testes, homologação, *staging* e qualidade; configuração de pipelines de teste e implementação automáticos.

Além de estar envolvido no processo de desenvolvimento para que eles estejam mais preparados para lidar com erros de produção. **Tradicionalmente, as operações são quase alheias à base de código, uma vez que se preocupam apenas com sua infraestrutura**. O objetivo é tornar todo o processo transparente de forma que você possa fazer milhares de implantações de produção por dia; ao contrário dos métodos tradicionais de configuração de uma janela de implantação uma vez a cada trimestre ou por mais tempo.

**Claro que para fazer isso, é necessário utilizar um conjunto de práticas e ferramentas.** *Quais, professor?* Ferramentas de Controle de versão (Git, CVS, Tortoise), Servidores de Integração Contínua (Jenkins, Bamboo, Travis), Docker/Vagrant, Gerenciamento de Configuração (SaltStack, Chef, Puppet). Isso reduz a dor de cabeça tanto para os desenvolvedores quanto para os operadores e reduz a quantidade de problemas de desempenho e erros de codificação.

**(STF – 2013)** Integração contínua, entrega contínua, teste contínuo, monitoramento contínuo e feedback são algumas práticas do DevOps.

**Comentários:** as principais características do DevOps são: colaboração entre equipes; fim de divisões; relação saudável entre áreas; teste, integração e entrega contínuos; automação de deploy; controle e monitoração; gerenciamento de configuração; orquestração de serviços; avaliação de métricas e desempenho; logs e integração; velocidade de entrega; feedback intenso; e comunicação constante (Correto).

**(STF – 2013)** Teste contínuo é uma prática do DevOps que, além de permitir a diminuição dos custos finais do teste, ajuda as equipes de desenvolvimento a balancear qualidade e velocidade.

**Comentários:** teste, integração e entrega contínuos são realmente práticas do DevOps que permitem reduzir custos de teste e ajuda a equipe de desenvolvimento a balancear a qualidade e velocidade (Correto).

**(TCU – 2015)** De acordo com a abordagem DevOps (development – operations), os desafios da produção de software de qualidade devem ser vencidos com o envolvimento



dos desenvolvedores na operação dos sistemas com os quais colaboraram no desenvolvimento.

**Comentários:** essa questão permite duas interpretações de 'envolvimento': (1) no sentido de interação e colaboração entre equipes; (2) no sentido de mão na massa mesmo - na operação dos sistemas. Por conta da ambiguidade, a questão foi anulada (Anulada).

**(TRE-PE – 2017)** O DevOps consiste em:

- a) um processo similar ao IRUP (IBM Rational Unified Process), que tem como objetivo dividir o processamento em fases e disciplinas de software para paralelizar as ações de desenvolvimento e de manutenção das soluções.
- b) uma plataforma aberta cuja função é substituir a virtualização de aplicações e serviços em containers e, com isso, agilizar a implantação de soluções de software.
- c) um aplicativo que permite o gerenciamento de versões de códigos-fonte e versões de programas, bem como a implantação da versão mais recente de um software em caso de falha.
- d) um processo de promoção de métodos que objetivam aprimorar a comunicação, tornando a colaboração eficaz especialmente entre os departamentos de desenvolvimento e teste e entre os departamentos de operações e serviço para o negócio.
- e) uma metodologia ágil que, assim como a XP (extreme programming) e o Scrum, tem foco na gestão de produtos complexos relativos à equipe de desenvolvimento.

**Comentários:** DevOps não é um processo similar ao iRUP, não é uma plataforma aberta, não é um aplicativo e não é uma metodologia ágil. DevOps é um processo de promoção de métodos que objetivam aprimorar a comunicação, tornando a colaboração eficaz especialmente entre os departamentos de desenvolvimento e teste e entre os departamentos de operações e serviço para o negócio (Letra D).

**(TRT-PA e AP – 2016)** Acerca de DevOps, assinale a opção correta.

- a) O DevOps concentra-se em reunir diferentes processos e executá-los mais rapidamente e com mais frequência, o que gera baixa colaboração entre equipes.
- b) O DevOps tem como princípio produzir, a partir da avaliação dos times de desenvolvimento do serviço, grandes mudanças e farta documentação com valor agregado para os usuários, assemelhando-se, por isso, com objetivos dos métodos iterativos e em cascata.



c) A infraestrutura de nuvem de provedores internos e externos vem restringindo o uso de DevOps pelas organizações.

d) O DevOps parte da premissa de adoção de grandes equipes de especialistas, com a menor interação possível, visando à padronização de processos e à mínima automação de atividades.

e) Atividades típicas em DevOps compreendem teste do código automatizado, automação de fluxos de trabalho e da infraestrutura e requerem ambientes de desenvolvimento e produção idênticos.

**Comentários:** (a) Errado, isso gera alta colaboração entre equipes; (b) Errado, não se assemelha com objetivos de métodos em cascata, visto que essa metodologia possui entregas menos frequentes e é menos dinâmica; (c) Errado, é justamente o inverso – elas usam com cada vez mais frequência; (d) Errado, recomenda-se a maior interação possível entre as equipes; (e) Correto, testes automatizados, automação de fluxos e infraestrutura são atividades frequentes que realmente exigem ambientes de desenvolvimento e produção idênticos (Letra E).

**(STJ – 2015)** DevOps é um conceito pelo qual se busca entregar sistemas melhores, com menor custo, em menor tempo e com menor risco.

**Comentários:** perfeito, perfeito, perfeito – todas são características do DevOps (Correto).

**(STJ – 2015)** O profissional especialista em DevOps deve atuar e conhecer as áreas de desenvolvimento (engenharia de software), operações e controle de qualidade, além de conhecer, também, de forma ampla, os processos de desenvolvimento ágil.

**Comentários:** ele é a combinação entre desenvolvimento, operação e controle de qualidade – portanto questão perfeita (Correto).

**(SLU-DF – 2019)** Em DevOps, o princípio monitorar e validar a qualidade operacional antecipa o monitoramento das características funcionais e não funcionais dos sistemas para o início do seu ciclo de vida, quando as métricas de qualidade devem ser capturadas e analisadas.

**Comentários:** o princípio monitorar e validar a qualidade operacional transfere o monitoramento para o início do ciclo de vida do software, para identificar antecipadamente problemas de qualidade que podem ocorrer no desenvolvimento. Na implantação e teste, as métricas de qualidade são capturadas e analisadas, sendo que essas métricas devem ser entendidas por todos os stakeholders (Correto).

**(MPE-PI – 2018)** A infraestrutura como código é uma prática DevOps caracterizada pela infraestrutura provisionada e gerenciada por meio de técnicas de desenvolvimento de código e de software, como, por exemplo, controle de versão e integração contínua.



**Comentários:** trata-se realmente de uma prática em que a infraestrutura provisionada e gerenciada por meio de técnicas de desenvolvimento de código e de software – ela oferece controle de versão, entrega e integração contínua (Correto).

**(STJ – 2018)** Apesar de ser um processo com a finalidade de desenvolver, entregar e operar um software, o DevOps é incompatível com a aplicação de métodos ágeis como o Scrum ou, ainda, com o uso de ferramentas que permitam visualizar os fluxos do processo.

**Comentários:** pelo contrário, é completamente compatível com as aplicações de métodos ágeis como o Scrum ou, ainda, com o uso de ferramentas que permitam visualizar os fluxos do processo (Errado).

**(STJ – 2018)** O gerenciamento de desenvolvimento de software por meio do Scrum pode ser combinado com o ciclo de vida do DevOps, haja vista que o DevOps combina práticas e ferramentas que aumentam a capacidade de uma organização de distribuir aplicativos e serviços; logo, a integração contínua do software pode ser realizada na sprint do Scrum junto com a operação dos serviços da organização.

**Comentários:** ele realmente combina práticas e ferramentas que aumentam a capacidade de uma organização de distribuir aplicativos e serviços e pode ser realizado na sprint do Scrum por meio da integração contínua (Correto).



# ESSA LEI TODO MUNDO CONHECE: PIRATARIA É CRIME.

Mas é sempre bom revisar o porquê e como você pode ser prejudicado com essa prática.



1

Professor investe seu tempo para elaborar os cursos e o site os coloca à venda.



2

Pirata divulga ilicitamente (grupos de rateio), utilizando-se do anonimato, nomes falsos ou laranjas (geralmente o pirata se anuncia como formador de "grupos solidários" de rateio que não visam lucro).



3

Pirata cria alunos fake praticando falsidade ideológica, comprando cursos do site em nome de pessoas aleatórias (usando nome, CPF, endereço e telefone de terceiros sem autorização).



4

Pirata compra, muitas vezes, clonando cartões de crédito (por vezes o sistema anti-fraude não consegue identificar o golpe a tempo).



5

Pirata fere os Termos de Uso, adultera as aulas e retira a identificação dos arquivos PDF (justamente porque a atividade é ilegal e ele não quer que seus fakes sejam identificados).



6

Pirata revende as aulas protegidas por direitos autorais, praticando concorrência desleal e em flagrante desrespeito à Lei de Direitos Autorais (Lei 9.610/98).



7

Concursado(a) desinformado participa de rateio, achando que nada disso está acontecendo e esperando se tornar servidor público para exigir o cumprimento das leis.



8

O professor que elaborou o curso não ganha nada, o site não recebe nada, e a pessoa que praticou todos os ilícitos anteriores (pirata) fica com o lucro.



Deixando de lado esse mar de sujeira, aproveitamos para agradecer a todos que adquirem os cursos honestamente e permitem que o site continue existindo.