# OBJETIVO:

O objetivo desse documento é descrever as etapas para o Deploy, sugerir o ambiente para publicação e descrever as ferramentas de acompanhamento que serão utilizadas.

# DEPLOY:

Antes da etapa Deploy e colocar o código em um servidor onde todos tem acesso, existem ambientes que um Deploy deve ser feito e fases que essa aplicação deve pasar antes de chegar na etapa final.

Essas fases são os ambientes de desenvolvimento, testes/homologação e produção.

Utilizaremos o Git para versionamento do código, isso ajuda se houver algum problema podemos voltar a versão anterior ou até mesmo testar uma versão específica para achar erros.

Criação das *branches* para trazer mais segurança no desenvolvimento, pois ajuda a montar um histórico do que foi feito no código e mantem organizado, isso é importante quando se tem um time trabalhando na mesma aplicação

# AMBIENTES:

## Desenvolvimento:

Nossa aplicação será desenvolvida localmente, utilizando ferramentas como VsCode, Node.JS e MySQL.

## Homologação:

Nesta fase é feito o primeiro Deploy, a aplicação será disponibilizada para que alguns usuários específicos façam testes.

Esse ambiente não é totalmente público, somente pessoas autorizadas podem testá-lo.

## Produção:

Depois que a aplicação foi desenvolvida e todos os testes foram realizados, é feito o último Deploy para o servidor, colocando a aplicação no ar.

Utilizaremos o SaaS da Azure que é o software como serviço e toda a parte de infraestrutura ficará por conta da Azure, onde eles cuidarão da sustentação da aplicação no ar, para que o nosso foco seja na aplicação.

# PRIVACIDADE E SEGURANÇA:

A Azure oferece recursos de defesa na nuvem. Uma ferramenta orientada por IA que auxilia a descobrir falhas na configuração da nuvem, melhora a postura de segurança contra ameaças cibernéticas.

# RECURSOS BÁSICOS:

Um dos pontos que nos fez escolher a Azure, foi a localização dos servidores, A disponibilidade e O alcance global. Eles possuem servidor no Brasil e isso diminui bastante a latência no recebimento dos dados, uma vez que a maior parte dos usuários que irão utilizar a aplicação se encontram no Brasil.

Além disso, é possível mover dados entre sites sem sair da rede. Isso ajuda a reduzir a latência, custos e melhora a segurança.

Fazendo a análise da aplicação que criamos para o cliente, a Azure atende os requisitos que precisamos para hospedagem dos nossos serviços/aplicações na nuvem. Trabalharemos com um banco de dados relacional que se conectará com uma API que receberá os dados e irá armazenar no banco.

A Azure oferece também um repositório próprio, onde podemos realizar o versionamento, montar as *branches*, realizar o Deploy de nossa aplicação e as 5 primeiras licenças desse repositório são gratuitas, sendo cobrado somente a partir da 6 licença.

E o melhor de tudo, a Azure oferece suporte para os produtos que iremos utilizar. E como nosso cliente já utiliza a Azure e ferramentas Microsoft como SharePoint, Teams e etc, resolvemos escolher esse ambiente, pois já há familiaridade no uso, nossa cliente já conhece a plataforma.

# PREÇO:

Em um comparativo, escolhemos a Azure pois ela está entre a AWS e Google Cloud, por seus preços medianos e pela ampla quantidade de serviços e recursos empresariais que são oferecidos.

A AWS tem uma grande quantidade de serviços, porém o preço é mais elevado que a Azure. E o Google Cloud é uma ferramenta mais barata, porém, não tem uma ampla gama de recursos como a Azure.

Além do modelo de cobrança "pagamento por uso" com o cartão de crédito, os clientes com contas corporativas podem comprar pré-assinaturas como parte de suas renovações anuais. Sendo útil para caso o cliente deseje orçamentar os gastos anuais com antecedência.

# MONITORAMENTO:

Escolhemos a ferramenta *DataDog* para realizar o monitoramento da nossa aplicação. Ela é uma plataforma SaaS. Essa ferramenta é um dos parceiros da Azure. Isso já é mais um ponto positivo para escolha dessa ferramenta, pois a integração é feita de forma rápida e os custos da plataforma são disponibilizados por meio da fatura da Azure.

Ela nos possibilita monitorar toda a parte de infraestrutura, aplicação e experiencia do usuário em um único painel. Nela conseguimos montar nosso próprio *dashboard* e colocar todas as métricas da nossa aplicação como APM, infraestrutura, logs, testes sintéticos e performance da internet. Isso possibilita uma visibilidade total da aplicação sem precisar navegar por várias páginas.

Com ela, conseguimos medir a latência, verificar as requisições pra API e o tempo de resposta, verificar as experiências do usuário, entre outros.

Com a integração do sistema na *DataDog*, é criado um *service map* onde mostra todos os serviços, o que consome e as respectivas dependências. E pelo monitor ele mostra os alertas através de cores diferentes para que seja investigado.

É possível visualizar também em formato de lista, onde mostra tempo das requisições, latência, erros, infraestrutura e o status pelo monitor. Consegue entrar também em cada um dos serviços e ter visibilidade completa. Disponibiliza também um painel com requisições e erros, tempo gasto na requisição, todos os Deploys que foram feitos, *endpoints*, mapa de dependências, possibilita verificar os rastros do que aconteceu durante uma requisição, consegue fazer filtros, entre outros.

A *DataDog* nos alerta quando há um alto tempo de resposta, um status inesperado ou um recurso quebrado.

Conseguimos identificar, por exemplo, se um carregamento lento é um problema com o código, com a infraestrutura ou a rede.

É possível identificar quais segmentos de clientes estão enfrentando erros em tempo real usando tags para dispositivo, sistema operacional e país.

*DataDog* oferece também testes sintéticos de HTTP, de browser e API. Verifica se há algum erro de JavaScript, desempenho os acessos ao servidor, erro no *front-end* como erro de endereço, dando a possibilidade de ver todo o site *front-end* funcionando e monitorar a experiência do usuário e se houver erros conseguir atuar.