**GRUPO ETULAS**

Eduardo Bezerra – RM: 98890

Jefferson Mendes de Farias Lima – RM: 552052

João Vitor Vicente Benjamin – RM: 98938

Silas Henrique da Silva Oliveira – RM: 98965

**PROJETO ETULAS**

SÃO PAULO/SP – 2024

Sumário

[Introdução 3](#_Toc181777955)

[Desenvolvimento da Arquitetura 4](#_Toc181777956)

[Nuvem Pública (SaaS e IaaS) 4](#_Toc181777957)

[Escalabilidade e Flexibilidade (IaaS) 4](#_Toc181777958)

[Custos Otimizados (IaaS) 4](#_Toc181777959)

[Acessibilidade Global (SaaS) 5](#_Toc181777960)

[Inovação e Agilidade (PaaS) 5](#_Toc181777961)

[Nuvem Privada (IaaS) 6](#_Toc181777962)

[Segurança e Conformidade (IaaS) 6](#_Toc181777963)

[Desempenho e Confiabilidade (IaaS) 6](#_Toc181777964)

[Integração com Sistemas Existentes (IaaS) 7](#_Toc181777965)

[Recursos Cloud Computing 8](#_Toc181777966)

[Servidores Virtuais (IaaS): 8](#_Toc181777967)

[Armazenamento em Nuvem (IaaS): 8](#_Toc181777968)

[Plataformas de Comunicação e Colaboração (SaaS): 9](#_Toc181777969)

[Desenho da Arquitetura 10](#_Toc181777970)

[Fluxo de dados 10](#_Toc181777971)

[Desenho Pipeline 13](#_Toc181777972)

[1. Integração Contínua (CI) 13](#_Toc181777973)

[2. Entrega Contínua (CD) 14](#_Toc181777974)

# Introdução

Nesta etapa crucial do projeto, a equipe se reúne para embarcar em uma jornada estratégica com o objetivo de definir a arquitetura ideal para o Etulas. A missão principal é identificar a combinação perfeita de nuvem, tipos de serviço, recursos e desenho arquitetônico que impulsionará o sucesso do projeto e atenderá às necessidades específicas do cliente.

# Desenvolvimento da Arquitetura

A solução Etulas apresenta características que se beneficiam de um modelo de nuvem híbrida, combinando os pontos fortes de ambientes públicos e privados, assim como também utiliza conceitos de serviço do tipo SaaS, PaaS e IaaS.

## Nuvem Pública (SaaS e IaaS)

### Escalabilidade e Flexibilidade (IaaS)

A alta demanda por triagem virtual exige escalabilidade rápida e flexível. A nuvem pública, como o Microsoft Azure ou Google Cloud Platform, oferece provisionamento instantâneo de recursos computacionais e armazenamento, atendendo às flutuações na demanda sem investimentos em hardware físico. Serviços como Azure Virtual Machines e Google Cloud Platform Compute Engine são exemplos de IaaS na nuvem pública.

### Custos Otimizados (IaaS)

A natureza cíclica da demanda por triagem virtual permite otimizar custos utilizando a nuvem pública. Serviços como Azure Virtual Machines e Google Cloud Platform Compute Engine são cobrados sob demanda, evitando custos desnecessários em períodos de baixa demanda.

### Acessibilidade Global (SaaS)

A nuvem pública garante acesso ao sistema Etulas de qualquer lugar, facilitando o uso por pacientes e profissionais de saúde em diferentes localidades. A solução pode ser acessada através de um site ou aplicativo móvel, utilizando serviços como Azure App Service ou Google Cloud Platform App Engine. Estes serviços configuram-se como SaaS, pois oferecem a aplicação pronta para uso, sem necessidade de gerenciamento da infraestrutura.

### Inovação e Agilidade (PaaS)

A nuvem pública oferece ferramentas inovadoras para desenvolvimento e testes de aplicações, como o Docker. Isso permite que a equipe Etulas desenvolva e teste novas funcionalidades de forma rápida e eficiente, utilizando serviços como Azure DevOps ou Google Cloud Build. O PaaS provê a plataforma para desenvolvimento e testes, sem a necessidade de gerenciar infraestrutura e software básico.

## Nuvem Privada (IaaS)

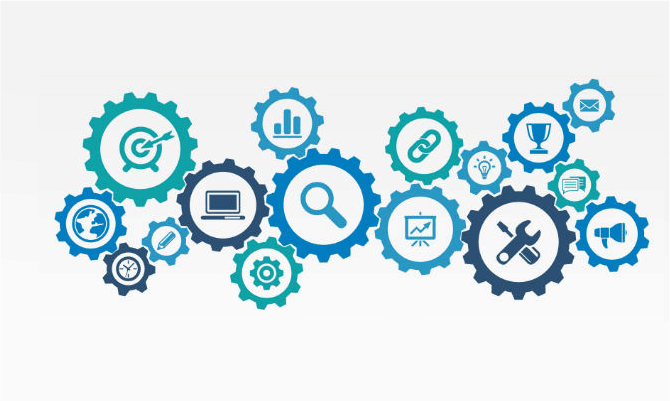
### Ícone de segurança (símbolo png) azulSegurança e Conformidade (IaaS)

A nuvem privada garante maior controle sobre os dados dos pacientes, atendendo às rigorosas leis e regulamentações de privacidade de dados da área da saúde. Serviços como Azure Virtual Machines e Google Cloud Platform Compute Engine em ambientes privados garantem a segurança e o isolamento dos dados. A infraestrutura de IaaS na nuvem privada é dedicada ao cliente, o que garante maior controle e segurança.

### Desempenho e Confiabilidade (IaaS)

A nuvem privada oferece maior controle sobre a infraestrutura, permitindo otimizar o desempenho e a confiabilidade da solução Etulas. Isso é crucial para garantir o acesso rápido e confiável ao sistema, especialmente em momentos de alta demanda. O IaaS na nuvem privada permite que o cliente configure a infraestrutura de acordo com suas necessidades específicas, o que pode resultar em melhor desempenho e confiabilidade.

### Integração com Sistemas Existentes (IaaS)

A nuvem privada facilita a integração com os sistemas legados dos hospitais, como prontuários eletrônicos e sistemas de gestão hospitalar. Serviços como Azure Virtual Network e Google Cloud VPC permitem a criação de redes virtuais seguras e isoladas para conectar diferentes sistemas de forma eficiente. A infraestrutura de IaaS na nuvem privada pode ser facilmente integrada com os sistemas legados do cliente, sem a necessidade de grandes modificações.

# Recursos Cloud Computing

Para o projeto Etulas, três recursos de cloud computing são essenciais: servidores virtuais (IaaS), armazenamento em nuvem (IaaS), e plataformas de comunicação e colaboração (SaaS). Cada um desses recursos será detalhado a seguir:

## Servidores Virtuais (IaaS):

Servidores virtuais permitem a rápida provisionamento de recursos de computação, essencial para lidar com a demanda flutuante de pacientes. Isso evita a necessidade de investimentos em hardware físico que consomem muita energia, oferecendo uma infraestrutura escalável e flexível. Os servidores virtuais podem suportar os sistemas de triagem virtual, o processamento de dados de pacientes e outras aplicações críticas do projeto. São exemplos de servidores o Microsoft Azure Virtual Machines e o Google Cloud Platform Compute Engine.

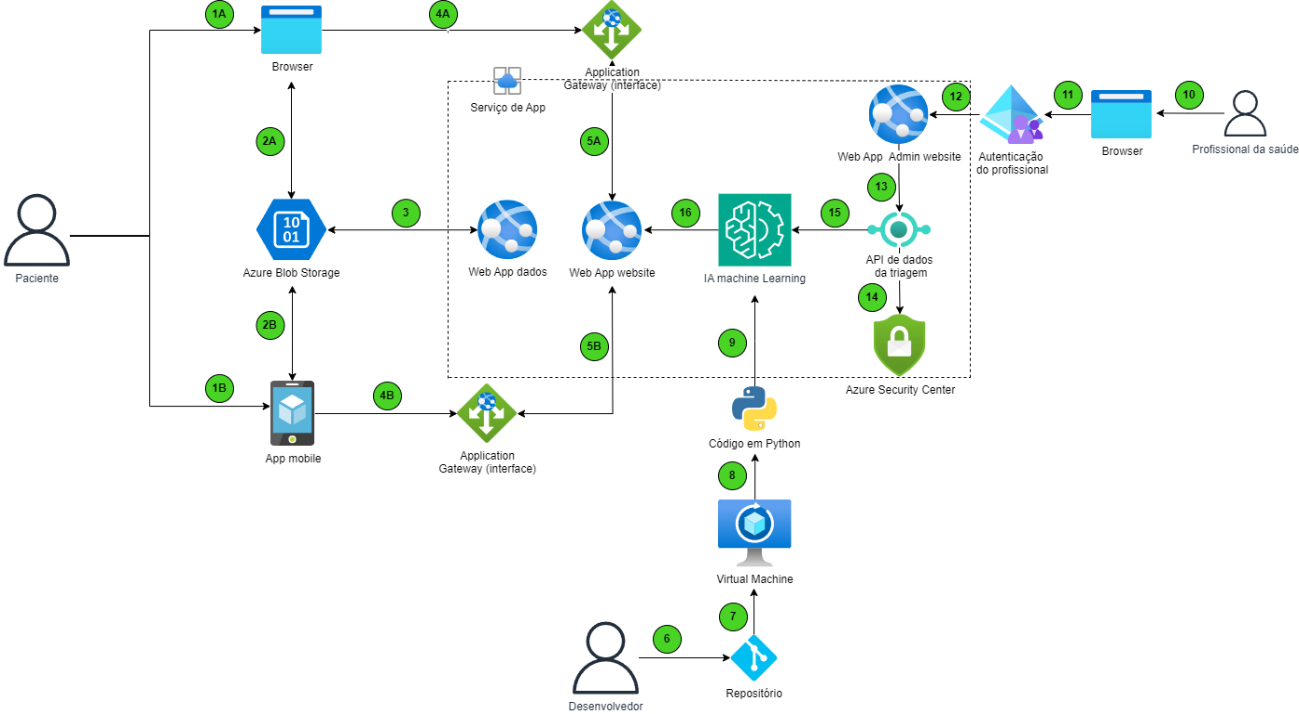
## Armazenamento em Nuvem (IaaS):

O armazenamento em nuvem proporciona uma solução segura e escalável para gerenciar grandes volumes de dados, incluindo prontuários eletrônicos, imagens médicas e resultados de exames. Esse recurso é vital para garantir a disponibilidade e a segurança dos dados dos pacientes, permitindo um acesso rápido e eficiente às informações necessárias para a tomada de decisões médicas. São exemplos de armazenamentos o Google Cloud Storage e o Microsoft Azure Blob Storage.

## Plataformas de Comunicação e Colaboração (SaaS):

Ferramentas de comunicação em tempo real são cruciais para a colaboração entre profissionais de saúde de diferentes departamentos e unidades hospitalares. Essas plataformas facilitam o compartilhamento instantâneo de informações e a coordenação de cuidados, otimizando a tomada de decisões e melhorando a qualidade do atendimento ao paciente. Além disso, a integração de prontuários eletrônicos unificados via SaaS permite acesso seguro e rápido a todas as informações do paciente, independentemente de sua localização no hospital. São exemplos de plataforma de comunicação o Microsoft Teams e o Slack.

# Desenho da Arquitetura



## Fluxo de dados

Paciente -> Aplicação Web/App Mobile:

1A e 1B: O paciente inicia a interação inserindo informações de triagem através de um navegador web ou um aplicativo móvel.

Aplicação Web/App Mobile -> Azure Blob Storage:

2A e 2B: As informações de triagem do paciente são enviadas e armazenadas no Azure Blob Storage.

Azure Blob Storage -> Web App Dados:

3: Os dados armazenados são recuperados pelo serviço Web App de dados para processamento.

Web App Dados -> Application Gateway (Interface):

4A e 4B: Os dados processados são encaminhados para a interface do Application Gateway, que serve como um ponto de acesso para a aplicação.

Application Gateway -> Web App Website:

5A e 5B: O Application Gateway encaminha os dados para o Web App website para exibição e interação adicional.

Desenvolvedor -> Repositório:

6: O desenvolvedor realiza o desenvolvimento e testes do código, que é armazenado em um repositório (GitHub).

Repositório -> Virtual Machine:

7: O código do repositório é implantado em uma Virtual Machine para execução.

Virtual Machine -> Azure Security Center:

8: A Virtual Machine é monitorada pelo Azure Security Center para garantir a segurança e conformidade dos dados.

Virtual Machine -> Código em Python:

9: A aplicação Python na Virtual Machine processa os dados de triagem utilizando machine learning.

Profissional de Saúde -> Browser:

10: O profissional de saúde acessa o sistema através de um navegador web para visualizar e interagir com os dados dos pacientes.

Browser -> Autenticação do Profissional:

11: O profissional de saúde se autentica no sistema para acessar as informações.

Autenticação do Profissional -> Web App Admin Website:

12: Após a autenticação, o profissional de saúde acessa o Web App Admin website para gerenciar os dados e monitorar os pacientes.

Web App Admin Website -> API de Dados da Triagem:

13: O Web App Admin website se comunica com a API de dados da triagem para recuperar e atualizar informações dos pacientes.

API de Dados da Triagem -> Azure Security Center:

14: A API de dados da triagem é monitorada pelo Azure Security Center para garantir segurança e integridade dos dados.

API de Dados da Triagem -> IA Machine Learning:

15: Os dados de triagem são analisados pela IA Machine Learning para melhorar a precisão da triagem e recomendações.

IA Machine Learning -> Web App Dados/Website:

16: As previsões e análises da IA são enviadas de volta para o Web App de dados e o Web App website para exibição e ação.

# Desenho Pipeline

Linha do tempo

Descrição gerada automaticamente  
  
Etapas da Pipeline de CI/CD

### Integração Contínua (CI)

**Etapa 1: Check-out do Código**

A pipeline inicia com a extração do código do repositório de controle de versão, como o GitHub, GitLab, ou Azure Repos. Esta etapa garante que a versão mais recente do código seja usada em todas as etapas subsequentes.

**Etapa 2: Restaurar Dependências**

Utilizando o gerenciador de pacotes NuGet, as dependências do projeto são restauradas para garantir que todas as bibliotecas externas estejam disponíveis. Isso previne problemas relacionados a bibliotecas ausentes.

**Etapa 3: Compilação (Build)**

O código é compilado para verificar erros de sintaxe e inconsistências. Essa etapa é realizada utilizando o .NET CLI (por exemplo, dotnet build) e garante que o código está apto para ser executado.

**Etapa 4: Execução de Testes Unitários**

Os testes unitários, desenvolvidos com xUnit para o HospitalController e outros serviços, são executados. O sucesso nesta etapa indica que as funcionalidades básicas estão se comportando conforme o esperado, enquanto qualquer erro impede o avanço do processo.

**Etapa 5: Análise de Qualidade de Código (opcional)**

Ferramentas como o SonarQube podem ser utilizadas para analisar o código e garantir que ele segue padrões de qualidade, verificando complexidade, cobertura de código e possíveis vulnerabilidades.

### Entrega Contínua (CD)

**Etapa 6: Publicação do Build**

Se os testes forem bem-sucedidos, o código compilado é empacotado para ser distribuído nos ambientes de homologação ou produção. O .NET CLI é usado para criar o pacote de deploy, geralmente com o comando dotnet publish.

**Etapa 7: Implantação em Ambiente de Homologação**

O código é implantado em um ambiente de homologação (staging), que simula o ambiente de produção, permitindo que o time de QA ou outros membros do projeto testem novas funcionalidades sem interferir no ambiente de produção.

**Etapa 8: Testes de Integração e Validação de API**

Nesta etapa, testes de integração, como os HospitalControllerIntegrationTests, são executados para verificar se a API funciona conforme o esperado. Testes de integração validam a interação entre diferentes componentes, e testes de API verificam os endpoints documentados com Swagger para garantir sua operação adequada.

**Etapa 9: Aprovação Manual para Produção (opcional)**

Para ambientes mais sensíveis, como o de produção, pode ser necessária uma aprovação manual para liberar o código. Essa etapa garante que um responsável revise o que será implantado antes de o código ser disponibilizado aos usuários finais.

**Etapa 10: Implantação em Produção**

Após a aprovação, o código é implantado no ambiente de produção, onde ficará disponível para os usuários finais. Nessa etapa, ferramentas de automação, como o Azure DevOps ou GitHub Actions, podem ser usadas para realizar o deploy automatizado.