



Guia de Implantação

Versão <1.0>

<CHP> - <Chamado Pro>

HISTÓRICO DE REVISÕES

Data	Versão	Descrição	Autor
10/11/2025	1.0	<i>Primeira versão instável para desenvolvimento. Ambiente em Docker e pronto para subir em uma instância AWS.</i>	Gustavo Alonso / João Paulo Yudi

SUMÁRIO

1. INTRODUÇÃO	4
1.1. Referências.....	4
2. PLANEJAMENTO DE IMPLANTAÇÃO.....	4
2.1. Responsabilidades	5
2.2. Cronograma.....	5
3. RECURSOS NECESSÁRIOS PARA IMPLANTAR O PRODUTO.....	6
3.1. Ambiente.....	6
3.2. Hardware	6
3.3. Software.....	6
3.4. Pessoas	7
4. ROTEIRO DE IMPLANTAÇÃO	7
5. TREINAMENTO E SUPORTE	8

1. INTRODUÇÃO

O presente documento tem como objetivo descrever as etapas e procedimentos necessários para a implantação do sistema **Chamado Pro**, uma aplicação desenvolvida para gerenciar e atender chamados de clientes de forma eficiente e integrada.

O **Chamado Pro** possui arquitetura baseada em microserviços, com **backend** implementado em **Java (Spring Boot)** e **frontend** desenvolvido em **Angular**, garantindo escalabilidade, segurança e boa experiência do usuário.

A implantação do sistema será realizada utilizando **Docker** para conteinerização dos serviços e **AWS (Amazon Web Services)** como infraestrutura de nuvem, assegurando alta disponibilidade, flexibilidade e facilidade de manutenção.

Este documento fornece uma visão geral das atividades envolvidas no processo de implantação, incluindo planejamento, testes, empacotamento, instalação, configuração, treinamento e suporte técnico pós-implantação.

1.1. Referências

- Documentação oficial do **Spring Boot**: <https://spring.io/projects/spring-boot>
- Documentação oficial do **Angular**: <https://angular.io/docs>
- Documentação do **Docker**: <https://docs.docker.com>
- Documentação da **AWS**: <https://docs.aws.amazon.com>
- Documento de Requisitos do Sistema Chamado Pro (versão 1.0)
- Documento de Arquitetura do Sistema Chamado Pro (versão 1.0)

2. PLANEJAMENTO DE IMPLANTAÇÃO

O processo de implantação do **Chamado Pro** envolve uma série de atividades organizadas em etapas, visando garantir uma transição estável e segura para o ambiente de produção. As principais fases são descritas a seguir:

2.1. Planejamento

- Definição do **ambiente de destino** (servidores EC2, RDS, S3, etc.) na AWS.
- Criação de **imagens Docker** separadas para o backend e frontend.
- Configuração de **variáveis de ambiente e segredos** (credenciais do banco de dados, chaves de API, etc.) usando o **AWS Secrets Manager**.
- Definição do cronograma de implantação e das responsabilidades de cada membro da equipe.

2.2. Testes

- Realização de **testes de integração** e **testes de carga** no ambiente de homologação.
- Validação de comunicação entre o backend e o frontend via endpoints REST.
- Teste do fluxo completo de abertura e acompanhamento de chamados.
- Verificação do consumo de recursos e comportamento dos containers Docker.

2.3. Preparação e Empacotamento

- Construção da imagem Docker do backend a partir do Dockerfile Java/Spring Boot.
- Construção da imagem Docker do frontend em Angular.
- Criação do arquivo docker-compose.yml para orquestração local e verificação pré-deploy.
- Upload das imagens para o **Amazon Elastic Container Registry (ECR)**.

2.4. Implantação e Configuração

- Provisionamento da infraestrutura via **AWS Elastic Beanstalk ou ECS (Elastic Container Service)**.
- Configuração do **Load Balancer** e **Auto Scaling Group** para o backend.
- Deploy do frontend em **AWS S3** com **CloudFront** para distribuição global.
- Configuração do banco de dados **AWS RDS (PostgreSQL/MySQL)** e teste de conexão.

2.5. Treinamento e Suporte

- Treinamento dos usuários-chave e equipe de suporte técnico sobre o uso do sistema e procedimentos básicos de manutenção.
- Elaboração de **manual de operação** e **documentação de endpoints** da API.
- Disponibilização de suporte técnico durante o período inicial pós-implantação (fase de estabilização).

2.1. Responsabilidades

A implantação do **Chamado Pro** requer a colaboração entre a equipe de desenvolvimento e o cliente, garantindo que todos os requisitos técnicos e operacionais sejam atendidos antes da liberação do sistema em ambiente de produção.

Equipe de Desenvolvimento

Responsabilidades:

- Planejar e executar todas as etapas técnicas da implantação.
- Criar e validar as imagens Docker do backend (Java/Spring Boot) e frontend (Angular).
- Configurar os serviços necessários na AWS (ECR, ECS, RDS, S3 e CloudFront).
- Realizar testes de integração, desempenho e segurança.
- Corrigir eventuais discrepâncias ou falhas encontradas durante os testes.
- Documentar todo o processo e prestar suporte técnico durante o período inicial pós-implantação.

Integrantes da equipe técnica:

- Gustavo Alonso
- Pedro Fukuya
- Gabriel Oliveira
- Gustavo Romano
- João Paulo Yudi

2.2. Cronograma

Data	Atividade	Responsáveis
10/11/25	Início do planejamento da implantação e definição do ambiente AWS	Gustavo Alonso, João Paulo Yudi
13/11/25	Criação das imagens Docker do backend (Java) e frontend (Angular)	Gustavo Alonso, João Paulo Yudi

15/11/25	Testes locais com docker-compose e validação de comunicação entre os containers	Pedo Fukuya, Gabriel Oliveira
16/11/25	Upload das imagens para o Amazon ECR e configuração do ECS/EKS	Gustavo Romano, Pedro Fukuya
17/11/25	Configuração do banco de dados no RDS e integração com o backend	Gustavo Alonso, Gustavo Romano
18/11/25	Deploy do frontend no S3 com CloudFront	João Paulo Yudi, Gabriel Oliveira
19/11/25	Testes de integração e carga no ambiente de homologação	Toda a equipe
20/11/25	Treinamento dos usuários e entrega da documentação final	Gustavo Romano, Pedro Fukuya
21/11/25	Liberação oficial do Chamado Pro em produção	Toda a equipe

3. RECURSOS NECESSÁRIOS PARA IMPLANTAR O PRODUTO

A implantação do Chamado Pro requer um servidor capaz de executar Docker, suporte ao banco PostgreSQL e ambiente com Java 17 e Node.js 20. Também são necessários Docker Compose e acesso à rede para comunicação entre frontend, backend e banco de dados.

3.1. Ambiente

A implantação do Chamado Pro ocorre em ambiente de nuvem AWS, utilizando os seguintes componentes:

- AWS EC2: Instância para execução do ambiente Docker contendo backend, frontend e PostgreSQL.
- AWS VPC: Rede privada garantindo isolamento entre os serviços.
- AWS IAM: Controle de permissões e políticas de acesso.

3.2. Hardware

Servidor Backend:

- vCPU: 2 a 4 vCPUs
- RAM: 4–8 GB
- Disco: 20 GB
- Rede: 100–300 Mbps

Banco de Dados (AWS RDS):

- Engine: PostgreSQL 16
- Tipo: db.t3.micro
- Armazenamento: 20 GB (auto-scale habilitado)

3.3. Software

Backend:

- Java 17
- Spring Boot 3.x
- Maven 3.9.x

- PostgreSQL 17

Frontend

- Angular 17
- Node.js 20 LTS
- NPM 10.x
- NGINX (em caso de deploy via container)

DevOps / Infraestrutura

- Docker 24+
- Docker Compose 2+
- AWS CLI v2
- GitHub para versionamento
- CloudWatch (logs e monitoramento)

3.4. Pessoas

Implantador: Realiza a instalação, ajusta variáveis de ambiente, executa o Docker e valida funcionamento.
Desenvolvedor Full-Stack: Suporte técnico em casos de incompatibilidade, correções rápidas e ajustes no código.

DevOps: Configura AWS, CI/CD, RDS, VPC e ECR.

Tester: Realiza testes após a implantação para garantir estabilidade.

4. ROTEIRO DE IMPLANTAÇÃO

1) Preparação do Ambiente

- Criar repositório no AWS ECR para backend e frontend. (t3.micro)
- Criar e configurar a VPC:
 - Criar VPC dedicada (CIDR ex.: 10.0.0.0/16).
 - Criar subnets públicas para a EC2 e Internet Gateway.
 - Associar rota 0.0.0.0/0 para permitir acesso externo.
 - Criar Security Groups permitindo:
 - Porta 22 (SSH) — opcional e restrita ao IP do administrador.
 - Porta 80 (HTTP) para o frontend.
 - Porta 8080 para comunicação com o backend.
 - Acesso ao Postgres apenas pela própria EC2 ou containers internos.
- Validar conectividade entre os serviços.

2) Build das Imagens Docker

- Instalar na instância as dependências (Git, Docker, Docker-Compose)
 - sudo yum update -y
 - sudo yum install git -y
 - sudo yum install docker -y
 - sudo systemctl enable docker
 - sudo systemctl start docker
 - sudo usermod -aG docker ec2-user
 - sudo curl -L "https://github.com/docker/compose/releases/download/v2.24.0/docker-compose-\$(uname -s)-\$(uname -m)" -o /usr/local/bin/docker-compose
 - sudo chmod +x /usr/local/bin/docker-compose
- Clonar projeto dentro da instância:
“git clone --recurse-submodules https://github.com/guualonso/chamado-pro.git”
- Acessar a pasta do projeto e buildar as imagens:
cd chamado-pro/CHP/3.Implementacao
docker-compose up -d –build

3) Configuração Final

- Configurar DNS (Route 53).
- Ativar HTTPS com ACM.
- Validar logs no CloudWatch.

4) Validação

- Teste completo de fluxo de manter chamado.
- Login e autenticação JWT.
- Teste de permissões (ADMIN, USUARIO).
- Teste de performance e latência.

5. TREINAMENTO E SUPORTE**1) Treinamento**

Um treinamento deve ser aplicado aos usuários finais cobrindo:

- Abertura de chamados.
- Atualização de status.
- Consulta de informações.
- Uso do painel administrativo.
- Boas práticas de uso do sistema.

Material entregue:

- Manual do Usuário (PDF)
- Vídeo de demonstração
- Slides de apresentação do sistema

2) Suporte Pós-Implantação

Período de Estabilização (primeiros 15 dias)

- Suporte dedicado da equipe de desenvolvimento.
- Monitoramento ativo via CloudWatch (CPU, memória, logs).
- Ajustes rápidos em containers e configurações AWS.

Suporte Contínuo

- Atualizações trimestrais do sistema.