**PLANO DE PROJETO**

**YOUTUBE**



[**1. SOBRE A EMPRESA 2**](#_heading=h.tyjcwt)

[**2. ROTEIRO DO PROJETO 3**](#_heading=h.3dy6vkm)

[Etapa 1 - Sobre o App 3](#_heading=h.1t3h5sf)

[Etapa 2 - Padrões de código e nomenclatura utilizados no desenvolvimento 3](#_heading=h.4d34og8)

[Etapa 3 - Padrões e nomenclatura utilizados nas issues 3](#_heading=h.2s8eyo1)

[**3. TESTES 3**](#_heading=h.17dp8vu)

[Etapa 4 - Definição de execução 3](#_heading=h.3rdcrjn)

[Etapa 5 - Ferramentas de automação 3](#_heading=h.26in1rg)

[Etapa 6 - Estratégias de Deploy 3](#_heading=h.lnxbz9)

[Etapa 7 - CODE REVIEW 3](#_heading=h.35nkun2)

[Etapa 8 - TIMELINE DE DESENVOLVIMENTO 4](#_heading=h.1ksv4uv)

# **SOBRE A EMPRESA**

# **ROTEIRO DO PROJETO**

## **Sobre o App**

YouTube é uma plataforma de vídeos online, fundada em 2005, por meio do qual seus usuários podem consumir, criar e compartilhar vídeos pela internet. Por meio desta plataforma, seus usuários podem consumir informações dos mais variados assuntos gratuitamente, garantindo também que possa ser usada como uma plataforma de aprendizado.

## **Padrões de código e nomenclatura utilizados no desenvolvimento**

Por ser uma plataforma web

## **Padrões e nomenclatura utilizados nas issues**

# **TESTES**

## **Definição de execução**

## **Ferramentas de automação**

## **Estratégias de Deploy**

## **CODE REVIEW**

O code review faz parte do processo de desenvolvimento, é uma rotina que deve ser adotada a fim de colaborar na entrega de um bom código, com qualidade, para seguir para as próximas etapas de desenvolvimento do software. O code review é uma boa prática, onde se faz revisões do código de dada aplicação, em nosso caso, a revisão do código de desenvolvimento desta plataforma.

Em geral, o code review é realizado a cada commit, pois estes dizem respeito às alterações que o código sofre ao longo da atividade de desenvolvimento, seja por ajustes ou implementação de uma nova funcionalidade da aplicação.

É importante manter a prática de realizar code review, pois nesta fase, em geral, possíveis bugs, falhas ou pontos de melhorias são levantados e ajustados antes mesmo do código chegar em produção. É uma atividade rápida de se realizar e que enriquece a qualidade do código sem afetar o tempo de entrega.

Durante a atividade usar checklist, automatizar os testes, ficar atento a possíveis impactos e novas ameaças, são práticas que podem ser adotadas para auxiliar sua execução.

## **TIME LINE DE DESENVOLVIMENTO**

Planejamento

Desenvolvimento

Geração de Build

Deploy em ambiente de homologação

Deploy em produção

OBS: Expliquem onde o software poderia ser testado e de que forma poderia ser testado.

* Definição de estratégia de CI/CD

OBS: descrever brevemente como faria para automatizar cada passo.

* Diagrama de componentes que a solução usaria em um provedor cloud

OBS: Recomenda-se usar o draw.io e salvar no github (opcional, pontos extras).

ATENÇÃO:

Prepar um documento breve de disaster recovery. Este documento deve conter 5 casos, as formas de como identificá-los e as estratégias para recuperar o ambiente.

ATIVIDADE BÔNUS:

Tentem montar uma pipeline básica com o Jenkins. O job deve baixar algo do github e executar alguma operação. Pode ser a execução de testes automáticos. Outra opção seria somente um ajuste de versão (editar um arquivo) e fazer o push da mudança para o github. A entrega pode ser os prints das telas de configuração do job e dos outputs. Esta atividade conta pontos extra, não é obrigatória.