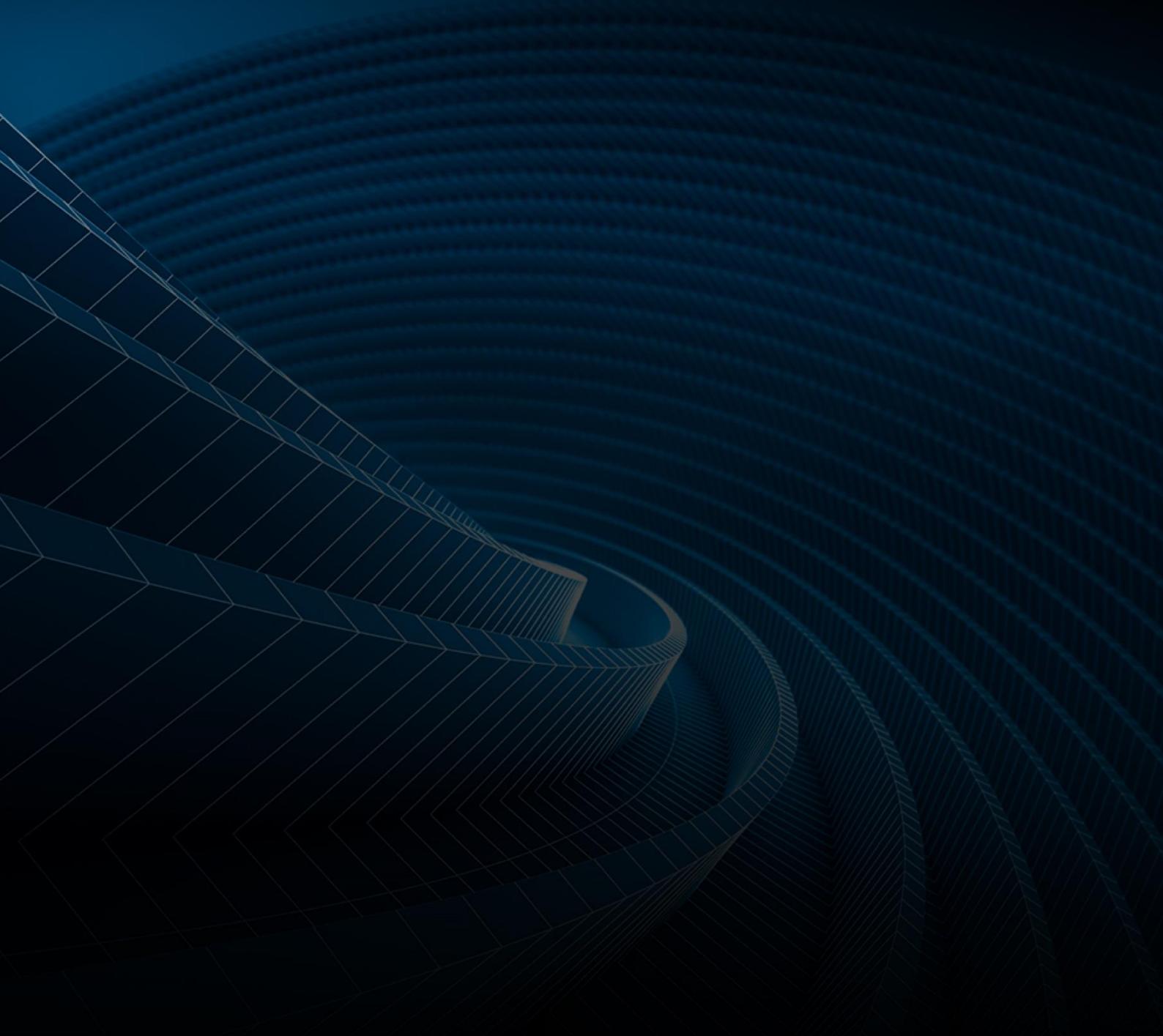


POSTECH



---

## Tech Challenge

Tech Challenge é o projeto da fase que engloba os conhecimentos obtidos em todas as disciplinas dela. Esta é uma atividade que, a princípio, deve ser desenvolvida em grupo. É importante atentar-se ao prazo de entrega, uma vez que essa atividade é obrigatória e vale 90% da nota de todas as disciplinas da fase.

### O problema

O **FIAP Cloud Games (FCG)** continua sua jornada de evolução! Após a migração para microsserviços e a adoção de serverless, agora o desafio é garantir que a comunicação entre os serviços seja eficiente e que o sistema possa escalar horizontalmente sem comprometer o desempenho.

Nesta fase, aplicaremos os conceitos de Mensageria, Docker e Kubernetes para garantir um ambiente altamente disponível, resiliente e pronto para produção.

### Desafio

A FIAP está sofrendo com a escalabilidade dos microsserviços: alguns não estão aguentando o alto número de requisições e isso tem afetado a imagem da **FCG**, já que muitos alunos e alunas não conseguem fazer o login e ficam muito tempo em fila para iniciar seus jogos.

Os arquitetos da FIAP tiveram a ideia de melhorar esses problemas utilizando Kubernetes e HPA para garantir sua escalabilidade em caso de necessidade. Isso acaba impactando diretamente os custos, uma vez que a quantidade de alunos jogando vai ditar a quantidade de recursos que a cloud deve alocar; em caso de baixo tráfego, os recursos serão diminuídos juntamente com o custo.

## Funcionalidades obrigatórias

### Comunicação assíncrona entre Microsserviços:

- Garantir que a comunicação seja assíncrona entre os microsserviços para garantir que, enquanto um microsserviço escala, ele segure a comunicação e envie quando estiver dimensionado corretamente.

### Melhorar imagens Docker:

- Com o intuito de melhorar a escalabilidade e a correção de problemas, precisamos garantir que o deploy seja mais eficiente utilizando imagens base menores e descartando configurações desnecessárias.

### Orquestrar containers com Kubernetes:

- Para garantir a alta escalabilidade dos microsserviços precisamos realizar a orquestração dos containers, visto que agora a quantidade de containers será muito variável e fica inviável controlá-los manualmente.

### Monitoramento:

- Precisamos monitorar nosso cluster Kubernetes para garantir que as métricas para o autoscaling estejam corretas e podemos utilizá-las para este fim, diminuindo os problemas e erros dos nossos microsserviços.

## Requisitos técnicos

### Comunicação Microsserviços:

- Implementar RabbitMQ, Apache Kafka ou AWS SQS para comunicação entre os microsserviços.
- Criar eventos assíncronos para operações críticas, como pagamentos e notificações.
- **(Opcional)** Garantir retry e dead-letter queues para mensagens que falharem.

## Tech Challenge

### **Containerização com Docker:**

- Criar Dockerfiles para todos os microsserviços.
- Criar imagens otimizadas e seguras para evitar desperdício de recursos.

### **Orquestração com Kubernetes:**

- Criar um cluster Kubernetes para gerenciar os microsserviços na cloud de sua preferência (AWS, Azure, Oracle, Google).
- Utilizar Helm Charts ou Kubernetes YAML Manifests para definir os deployments.
- Configurar Auto Scaling (HPA) para escalar os serviços conforme a demanda.
- Empregar boas práticas utilizando ConfigMaps e Secrets.

### **Monitoramento**

- **(OPCIONAL)** Implementar Prometheus e/ou Zabbix, e Grafana para métricas de infraestrutura.
- Implementar APM para garantir performance dos microsserviços.

### **Entregáveis da Fase 4**

- **Vídeo de até 15 minutos** demonstrando todos os requisitos. Ele pode ser em grupo ou individual (um integrante do grupo grava ou é possível se dividir entre si e apresentar).
  - O projeto deve rodar na cloud (à sua escolha), apresentando os requisitos anteriores.
  - Se o requisito técnico estiver com a flag **(Opcional)**, isso significa que caso ele não seja implementado não descontaremos pontos.
  - A infraestrutura não precisa ficar em pé até a avaliação: após gravar o vídeo, ela deve ser excluída para evitar gastos.
- **Documentação** (pode ficar no README/Miro/Imagen):
  - Fluxo de comunicação assíncrona dos microsserviços.

## Tech Challenge

- Desenho de arquitetura representando o fluxo de funcionamento no Kubernetes.
- **Código-fonte** no repositório (público ou privado), incluindo:
  - APIs conforme requisitos separados em microsserviços.
  - Dockerfile para cada microsserviço.
  - Manifestos kubernetes (YAML).
  - README.md completo com instruções de uso e objetivos.
- **Relatório de entrega (PDF ou TXT)** – esse arquivo deve ser postado na data da entrega, contendo:
  - Nome do grupo.
  - Participantes e usernames no Discord.
  - Link da documentação.
  - Link dos repositórios.
  - Link do vídeo salvo no Youtube ou lugar de sua preferência.

Lembramos que caso você tenha qualquer dúvida, é só nos chamar no Discord!

The background features a dark blue gradient with a subtle, glowing grid pattern. A prominent, wavy, curved line runs diagonally across the center, creating a sense of depth and motion.

**POSTECH**