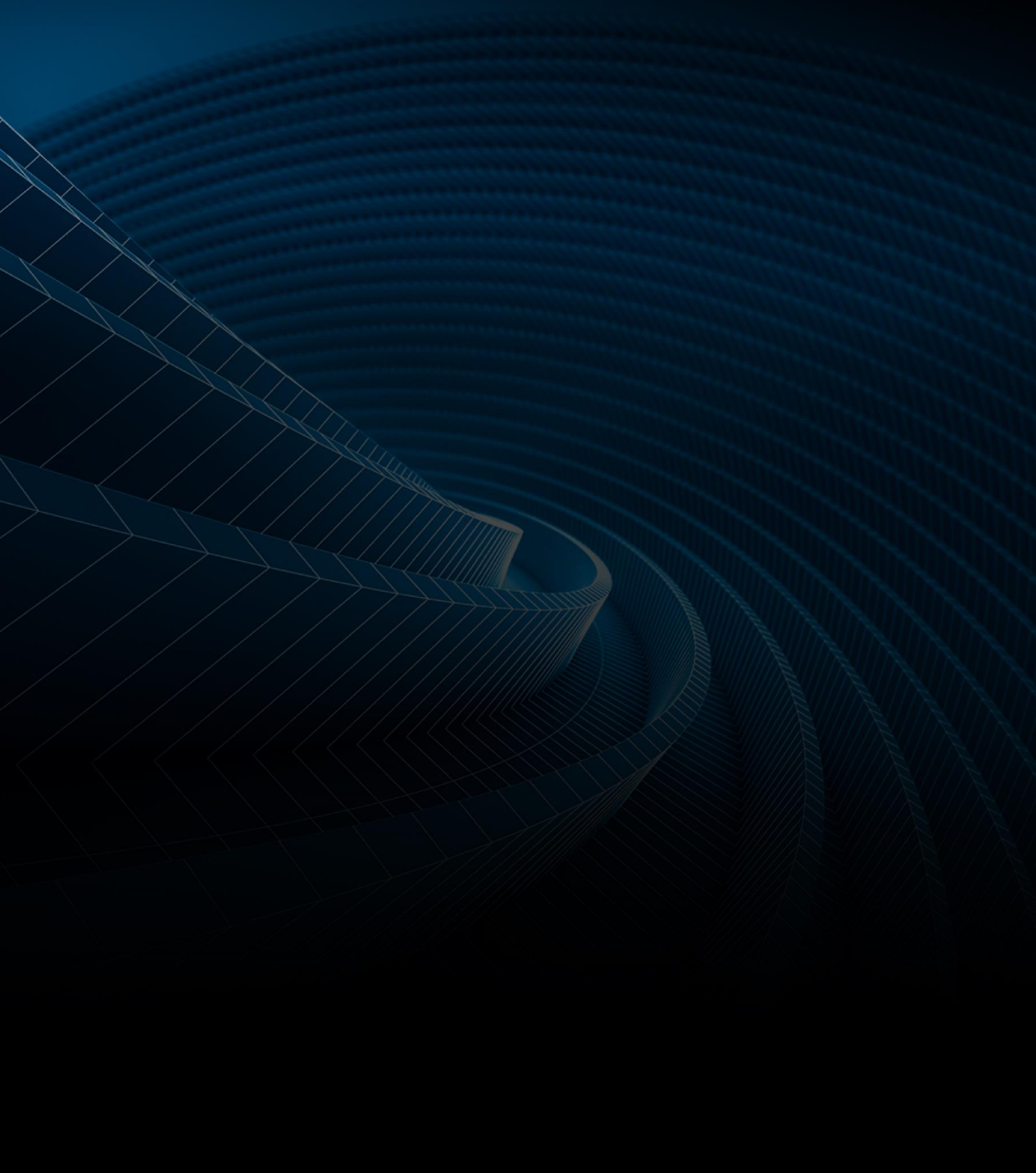


POSTECH



## 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

A **FIAP Cloud Games (FCG)** segue sua evolução! Nesta fase, o foco será a migração para microsserviços, a otimização da busca de jogos e a adoção de soluções serverless para eficiência operacional.

O desafio desta fase foi estruturado para aplicar os conhecimentos adquiridos nas disciplinas da fase, como Elasticsearch, Serverless, API Gateway, Microsserviços, Arquitetura de Software e Monitoramento e Acesso.

### Desafio

Após a FIAP ter conseguido identificar o valor no MVP construído anteriormente e lançarmos a primeira versão para os alunos e alunas de forma totalmente automática e acessível via cloud, precisamos deixar nosso sistema mais robusto e modular, a fim de identificar problemas em módulos.

Sem deixarmos a aplicação totalmente fora do modelo anterior, no qual seguimos com a arquitetura monolítica, vamos dar robustez a nossa aplicação e separar em módulos lógicos, além de seguirmos uma abordagem serverless, consolidando nossa escalabilidade.

### Funcionalidades obrigatórias

#### Migração para Microsserviços:

- Separar a API em três microsserviços principais. A seguir temos **exemplos** para a criação/separação de microsserviços:
  - **Usuários:** cadastro, login e gerenciamento de perfis.
  - **Jogos:** listagem, compra e recomendação de jogos.
  - **Pagamentos:** processamento e status de transações.

## Tech Challenge

### Elasticsearch:

- Implementação:
  - Armazenar e indexar os dados dos jogos para busca eficiente.
  - Criar consultas avançadas para sugerir jogos baseados no histórico do usuário.
  - Implementar agregações para métricas, como os jogos mais populares.

### Utilizar Serverless:

- Criar funções serverless para processos assíncronos, como envio de notificações e processamento de pagamentos.
- Configurar gatilhos em eventos para acionar funções de forma automática.
- Implementar um API Gateway para gerenciar requisições dos microsserviços.
- Garantir segurança entre os acessos para os microsserviços

### Arquitetura:

- Implementar event sourcing para registrar todas as mudanças no estado do sistema.
- Melhorar a observabilidade com logs e rastreamento distribuído (Traces).

### Requisitos técnicos

#### Microsserviços:

- Criar ao menos três microsserviços, separados em repositórios diferentes. A seguir, uma sugestão para a separação (este é somente exemplo; a forma com que ela será feita fica à sua escolha, mas é obrigatório que siga a separação em três microsserviços):
  - Usuários.
  - Jogos.

## Tech Challenge

- Pagamentos.

### Elasticsearch:

- Indexar dados dos jogos.
- Criar consultas avançadas e agregações para recomendações e métricas.

### Serverless:

- Criar funções AWS Lambda ou Azure Functions para operações específicas.
- Configurar triggers para acionar funções automaticamente.
- Implementar API Gateway para gerenciar e proteger os microsserviços.
- **(Opcional)** Configurar a autenticação JWT e o controle de requisições.
- **(Opcional)** Aplicar rate limiting para proteger os serviços contra sobrecarga.

### Arquitetura

- Implementar event sourcing para registrar todas as mudanças no estado do sistema.
- Melhorar a observabilidade com logs e rastreamento distribuído (Traces).

### Entregáveis da Fase 3

- **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.

## Tech Challenge

- 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/Imagem):
  - Fluxo de comunicação dos microsserviços.
  - Desenho de arquitetura representando o fluxo de funcionamento.
- **Código-fonte** no repositório (público ou privado), incluindo:
  - APIs conforme requisitos separados em microsserviços.
  - Arquivo de Pipeline CI (testes) escrita (fase anterior).
  - Arquivo de Pipeline CD (deploy) escrita (fase anterior).
  - Para o deploy da lambda, é possível utilizar Serverless Framework, CLI da cloud escolhida, terraform ou cloudformation.
  - 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 is a dark blue gradient. It features a series of concentric, slightly wavy circles that create a sense of depth and movement. Overlaid on these circles is a grid of thin, light blue lines that form a perspective of a 3D surface, possibly a ramp or a series of steps, leading towards the center.

POSTECH