

## PROJETO SEMESTRAL

A Expo Tech é um evento acadêmico da UniFECAF onde alunos do curso de Análise e Desenvolvimento de Sistemas (ADS) apresentam projetos que integram teoria e prática, focando em soluções inovadoras para aplicações móveis e desenvolvimento de jogos. O objetivo é estimular a criatividade, o trabalho em equipe e a aplicação de tecnologias emergentes.

### **Projeto:**

Os alunos podem escolher entre duas áreas:

- Mobile Development: Desenvolvimento de aplicativos móveis com integração de IoT, IA e Big Data.
- Game Development: Criação de jogos multiplayer ou com realidade aumentada (AR).

### **Requisitos:**

Planejamento de Requisitos: Levantamento de requisitos funcionais e não funcionais.

### **Implementação:**

- Aplicativos móveis: Desenvolvidos em React Native.
- Jogos: Desenvolvidos em Godot ou Unity.
- Banco de Dados e IoT (para Mobile): Integração com dispositivos IoT (ESP32, sensores) e armazenamento em bancos SQL ou NoSQL.
- Big Data e IA (opcional): Uso de Machine Learning e Big Data para análise de dados.
- **Quality Assurance (QA): Cenários de Testes, Testes unitários, de interface (UI Tests) e de desempenho com Blaze Meter / JMeter. Configuração de pipeline de CI/CD no GitHub Actions para execução de build, testes e deploy do projeto.**

Conceitos técnicos:

- Mobile Development: React Native, integração com IoT (sensores, Bluetooth, Wi-Fi).
- Game Development: Unity ou Unreal Engine, multiplayer via Bluetooth, realidade aumentada (AR) e IA para NPCs inteligentes.
- **Quality Assurance: Cenários de Testes, Testes automatizados, CI/CD, relatórios de desempenho.**
- Disruptive Architectures: IoT, Big Data, Machine Learning e APIs inteligentes.

Tecnologias Envolvidas:

- Mobile: React Native, Firebase, SQLite/PostgreSQL.
- Games: Godot, Unity.
- QA: GitHub Actions, JMeter, Jest, Cypress, Blaze Meter.
- IoT e IA: ESP32, TensorFlow, OpenAI API, Vuforia.
- Back-End e APIs: Node.js, Express, NestJS.
- Banco de Dados: PostgreSQL, MongoDB, Firebase.

Sugestão de Temas:

**-Mobile Development:**

- "Aplicativo de Monitoramento IoT para Sensores Ambientais".
- "Sistema de Controle de Acesso via Reconhecimento Facial".
- "Plataforma de Telemetria para Agricultura Inteligente".

**-Game Development:**

"Jogo de Corrida Multiplayer com Controles Bluetooth".

"Jogo de Aventura em Realidade Aumentada (AR)".

"Jogo de Sobrevivência com IA Adaptativa".

**Formato:**

- Equipes de 3 a 5 alunos.
- Uso de metodologias ágeis.
- Apresentação do projeto no evento Expo Tech.

O foco é transformar conhecimento em inovação, aplicando tecnologias modernas para criar soluções práticas e criativas.

**Cronograma de Q.A.:**



**Semana 1 (01/04 - 05/04) - Planejamento e Definição de Cenários (Vale 0,25 pt)**

**Ações:**

✓ Reunião inicial: Alinhamento sobre escopo de testes e ferramentas.

✓ Definir documental de 10 cenários de teste no mínimo:

- 4 cenários de testes unitários
- 4 cenários de teste de usabilidade.
- 2 cenários de performance (ex.: carga em login, consulta ao banco de dados).

Entregável: Documento padrão com 10 cenários de teste



**Semana 2 (08/04 - 12/04) - Escrever Testes nos cenários e testar a ferramenta JMETER (Vale 0,25 pt)**

**Ações:**

✓ Instalar e configurar Jmeter

✓ Descrever os 10 cenários de teste no mínimo:

- 4 cenários de testes unitários
- 4 cenários de teste de usabilidade.

- 2 cenários de performance (ex.: carga em login, consulta ao banco de dados).

Entregável: Documento com 10 cenários de teste

### **Semana 3 (15/04 - 19/04) - Testes de Performance (JMeter) (Vale 0,50 pt)**

Ações:

- ✓ Criar scripts no JMeter:
- ✓ Implementar testes unitários:

Cada grupo desenvolve testes unitários para validar funções críticas.

- ✓ Testes manuais de usabilidade:

Gravar 2 fluxos de usabilidade no BlazeMeter (ex.: cadastro de usuário, processo de compra).

- ✓ Relatório de bugs: Registrar falhas em planilha compartilhada.
- ✓ Executar testes e analisar resultados:

Tempo de resposta, taxa de erro, consumo de CPU/memória (se aplicável).

Entregável:

Arquivo JMeter (.jmx) + relatório de performance (gráficos e análise).

Código dos testes unitários (GitHub).

### **Semana 4 (22/04 - 26/04) - Automação Básica e Validação (Vale 0,50 pt)**

Ações:

- ✓ Automatizar processos repetitivos:  
Ex.: Script em Python para rodar testes unitários automaticamente.
- ✓ Integrar ao GitHub Actions :

Executar testes unitários a cada commit.

- ✓ Revisão com o professor: Validar cobertura e resultados.

Entregável:

Script de automação + print de execução.

### **Semana 5 (29/04 - 03/05) - Testes Finais e Consolidação (Vale 0,50 pt)**

Ações:

- ✓ Repetir testes críticos:

Usabilidade (BlazeMeter).

Performance (JMeter).

✓ Corrigir bugs pendentes.

✓ Gerar relatório consolidado:

Resultados de usabilidade, performance e unitários.

Entregável: Relatório final (máx. 3 páginas) com:

- Métricas de performance.
- Gravações de usabilidade.
- Status dos testes unitários.



#### **Semana 6 (06/05 - 17/05) - Preparação para Apresentação (Vale 0,50 pt)**

Ações:

✓ Relatório resumindo:

- Problemas encontrados.
- Melhorias aplicadas.

Dados de performance/usabilidade.

✓ Simulação de apresentação.

Entregável: Demonstração da execução dos testes



#### **17/05 - Apresentação Final (a confirmar) (Vale 1,50 pt)**

✓ Demonstração:

Mostrar gravações de usabilidade (BlazeMeter).

Exibir relatórios de performance (JMeter).

Código dos testes unitários.

✓ Avaliação:

Qualidade dos cenários.

Resultados práticos.

**Ferramentas e Critérios:**

- Testes Unitários: conforme execução.
- Usabilidade: BlazeMeter (gravação de telas).
- Performance: JMeter.
- Automação: Scripts simples (GitHub Actions).