



FIAP GRADUAÇÃO

GLOBAL SOLUTION 2025/1

2º ANO

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Turmas de Fevereiro

DESAFIO FIAP: EVENTOS EXTREMOS

A natureza está cheia de surpresas. Ventos poderosos, chuvas intensas, tremores inesperados, calor de derreter ideias e frio de congelar pensamentos. Os eventos extremos estão aí e fazem parte do nosso novo cenário global. Mas em vez de temê-los, que tal enxergar tudo isso como uma grande oportunidade de criar algo incrível?

Na FIAP, acreditamos que cada desafio é uma chance de inovar, aprender e transformar. E quando a tecnologia se junta à criatividade, surgem soluções que fazem a diferença de verdade.



DESAFIO FIAP: EVENTOS EXTREMOS

Agora é a sua vez. Imagine um cenário impactado por um evento extremo da natureza. Seu papel é propor uma solução que use tecnologia, inovação e boas ideias para ajudar pessoas, proteger o meio ambiente ou prevenir problemas ainda maiores.

Vale usar tudo o que você sabe e explorar o que ainda está aprendendo. Crie protótipos, pesquise, conecte dados, crie narrativas e pense fora da caixa. O que você criaria para enfrentar um super calor? Ou uma cidade isolada por um terremoto? Como a tecnologia pode ser usada de forma prática e surpreendente?

Você está no comando. Solte a criatividade e mostre como grandes ideias nascem em momentos de grandes desafios.

DESAFIO FIAP: EVENTOS EXTREMOS

A inovação pode surgir no design de soluções acessíveis, no uso de sensores conectados à internet das coisas, na análise de grandes volumes de dados em tempo real ou na criação de plataformas que integram pessoas, serviços e recursos de forma ágil e eficaz.

Com criatividade, é possível simular cenários, treinar comunidades por meio de experiências imersivas, construir modelos de negócio sustentáveis e fortalecer a resiliência das cidades e das pessoas. Áreas como tecnologia, design, engenharia e gestão se conectam para enfrentar desafios reais e criar respostas que protegem, aproximam e salvam vidas.

Neste desafio, o mais importante não é apenas o que você sabe, mas o que você é capaz de imaginar e colocar em prática. Afinal, quando ideias ganham propósito, elas têm o poder de mudar o mundo.



VEJA ALGUMAS SUGESTÕES PARA VOCÊ SE INSPIRAR:

- Sistemas de alerta antecipado para desastres naturais
- Aplicativos de evacuação e rotas seguras em situações de risco
- Drones para busca, resgate e entrega de suprimentos em áreas isoladas
- Estações móveis de energia limpa para abrigos temporários
- Plataformas de comunicação offline para comunidades sem acesso à internet
- Monitoramento de encostas, barragens e zonas de risco com sensores IoT
- Reutilização de água da chuva e soluções portáteis de purificação
- Tecnologias para reconstrução rápida e sustentável de moradias
- Gamificação para educação sobre prevenção de desastres



VEJA ALGUMAS SUGESTÕES PARA VOCÊ SE INSPIRAR:

- Inteligência artificial para prever padrões de eventos extremos
- Gestão de abrigos e recursos em tempo real com uso de dashboards
- Apoio psicológico remoto via apps em contextos de calamidade
- Impressão 3D de estruturas de emergência para zonas afetadas
- Sistemas de energia solar para manter hospitais móveis funcionando
- Mapeamento colaborativo de áreas atingidas usando dados da comunidade

INDICAÇÃO DE FONTE

<https://disasterscharter.org/>

<https://www.youtube.com/watch?v=ZvExM-Z3E2w&t=5s>

A Carta Internacional Space and Major Disasters é uma iniciativa global que reúne agências espaciais de diversos países com o objetivo de fornecer dados de satélite gratuitos e em tempo real para apoiar ações de resposta a desastres naturais.

Sempre que ocorre um grande desastre, como enchentes, terremotos, furacões ou incêndios florestais, a Carta pode ser acionada por governos e instituições autorizadas. A partir disso, satélites são mobilizados para capturar imagens da região afetada, permitindo o monitoramento, a análise dos impactos e a tomada rápida de decisões em situações de emergência.

INDICAÇÃO DE FONTE

No site, é possível acessar:

- Casos recentes de desastres monitorados
- Mapas e imagens de satélite com áreas impactadas
- Tipos de desastres acompanhados
- Ferramentas de visualização e relatórios técnicos

É uma ferramenta poderosa para pesquisadores, estudantes e desenvolvedores que queiram trabalhar com dados reais e atuais em projetos voltados à gestão de crises, cidades resilientes, tecnologias emergenciais e soluções para eventos extremos.



PROGRAMAÇÃO

DATA	EVENTO
26/05	LIVE DE ABERTURA
26/05	CONTEÚDO PARA ALUNOS
06/06 até 23h55	ENTREGA NO PORTAL FIAP
Até 13/06	CORREÇÃO DOS PROFESSORES



REGRAS GERAIS

- Equipe: o desafio pode ser realizado individualmente ou em grupo de até 3 integrantes (sem exceções). Os grupos podem ser formados com alunos de turmas e turnos diferentes.
- Haverá chamada nos dias de aula para todas as disciplinas (mantendo os dias presenciais e remotos).

ENTREGAS

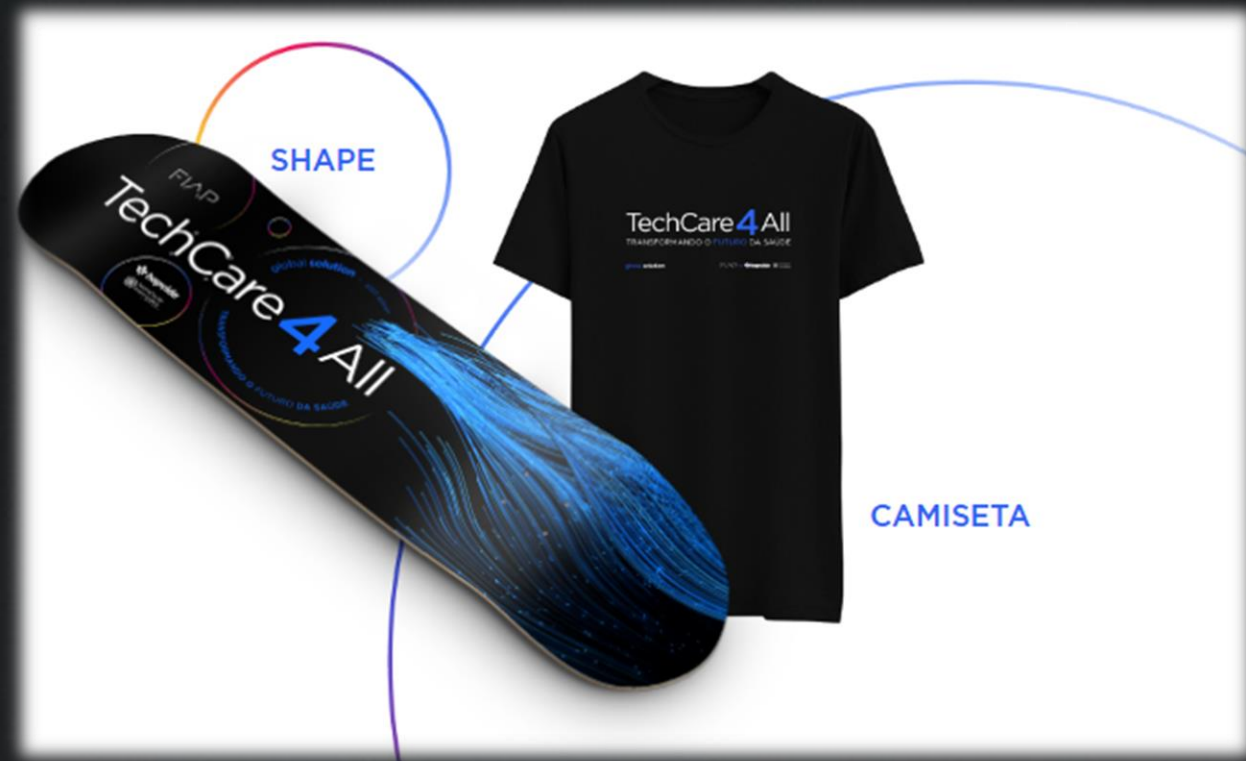
- As entregas de todas as disciplinas serão realizadas até o dia **06/06/25 até às 23h55** na área de trabalhos, no portal do aluno. *(Instruções de entrega estão no final do documento).*
- Entrega de um txt. no .zip de cada entrega: Esse txt deve ter o RM, nome do aluno e turma de cada integrante.



Cadastro dos grupos (prazo até : 30/05): <https://forms.office.com/r/8gXnbZ0jMh>



- Os grupos serão avaliados além das notas por uma comissão de professores;
- O grupo que obter as melhores notas em todas as disciplinas, junto com a melhor avaliação do vídeo Pitch será o grande vencedor (shape e camisetas exclusivas);
- O grupo que obter nota igual ou maior que 90 em todas as disciplinas, junto com a nota do vídeo Pitch da matéria de **Java Advanced**, que também tem que ser uma nota igual ou maior que 90, então esses grupos serão analisados pelos Scrum Master, Professores e o Coordenador, para a escolha de um único grupo vencedor.



ENTREGAS

Descrição das entregas por disciplina



DESAFIO

- O Grupo deve propor uma solução para o **tema da Global Solution**.
- Essa solução deve ser utilizada para desenvolver as entregas de **todas as disciplinas**.



ADVANCED BUSINESS DEVELOPMENT WITH .NET

O projeto em Advanced Business Development with .NET deve abranger uma solução criativa e inovadora para o tema proposto nesta GS, criando basicamente uma API REST para tratar de problemas críticos e auxiliar as pessoas em períodos de extrema urgência. Para tanto, com base nas aulas deste semestre, o grupo deve desenvolver a solução atendendo aos seguintes requisitos:

Requisitos do projeto:

- API REST que atenda a boas práticas de programação / arquitetura;
- Persistência em Banco de Dados relacional;
- Pelo menos um relacionamento 1:N;
- Documentação com o Swagger;
- Aplicação de Razor e TagHelpers;
- Uso correto da Migration no projeto.



ADVANCED BUSINESS DEVELOPMENT WITH .NET

A **avaliação** será da seguinte forma:

- Viabilidade e Inovação; **(10 pontos)**
- Cumprimento dos requisitos técnicos e boas práticas; **(70 pontos)**
- Documentação com Swagger e Github. **(20 pontos)**

A **entrega** deve conter os seguintes artefatos:

- Link do Github contendo repositório com os códigos fontes e ReadMe;
- O ReadMe deve ter diagramas, desenvolvimento e parte de testes;
- Além disso, o ReadMe deve ter instruções para acesso e exemplos de testes;
- Vídeo demonstração da solução completa (máx 8 minutos);
- Vídeo Pitch (máx. 3 minutos).



COMPLIANCE, QUALITY ASSURANCE & TESTS

Considerando a sua proposta de projeto da Global Solution, documente:

Entrega 1 (80 pts): arquitetura da solução, aplicando o modelo TOGAF, com a ferramenta ARCHI, contendo:

- Visão de arquitetura com indicação de stakeholders, objetivos e requisitos (peso 20);
- Arquitetura de negócio com indicação de processos, funções, atores e seus papéis (peso 20);
- Arquitetura de sistemas com indicação de componentes de aplicação, componentes de dados, em camadas de serviços (peso 20);
- Arquitetura de tecnologia com detalhes da forma de conexão em rede para uso das aplicações, devices de acesso dos usuários e equipamentos servidores, com dados de quais softwares cada máquina precisa ter instalado (peso 20).



Entrega 2 (20 pts): documento de apresentação do projeto que permita a validação da proposta de arquitetura:

- Apresentação do nome do projeto, da composição da equipe (nome e RM) e descrição de qual o desafio/problema escolhido para ser resolvido (10 pontos);
- Definição do público alvo que usará e se beneficiará da sua solução, com estimativas sobre o impacto da solução/produto para o público alvo, fundamentado em estudos de dados de mercado/estatísticas (10 pontos).

OBS: Entregue um arquivo .ZIP com o documento descritivo do projeto, o arquivo .ARCHI de desenho da arquitetura e um arquivo .PDF com o mesmo desenho de arquitetura (PDF gerado a partir do ARCHI). O descumprimento de qualquer requisito de entrega implicará em desconto de 10 pontos.

DEVOPS TOOLS & CLOUD COMPUTING 1/4

1. Objetivo

Sua equipe DevOps deverá **containerizar** a API desenvolvida na disciplina de  **JAVA ADVANCED** ou de  **ADVANCED BUSINESS DEVELOPMENT WITH .NET**, utilizando **Dockerfile** e seguindo as boas práticas de infraestrutura como código.

Montar um ambiente com pelo menos **dois containers** Docker **integrados**, sendo:

- ✓ Container para rodar a **Aplicação**: projeto em **Java** ou **.NET**.
- ✓ Container do **Banco de Dados**: escolha livre entre Oracle, PostgreSQL, MySQL, MongoDB etc. (desde que seja em container para persistência dos dados).

Não será aceito o Banco H2 na escolha

DEVOPS TOOLS & CLOUD COMPUTING 2/4

2. Regras para os Containers

- Container da **Aplicação**: projeto em **Java** ou **.NET**.
 - ✓ Deve ser construído via **Dockerfile!** (obrigatório gerar sua imagem personalizada)
 - ✓ Executar com **usuário não root** (definir o nome no Dockerfile – escolha livre).
 - ✓ Definir um **diretório de trabalho** (definir no Dockerfile – nomeação livre).
 - ✓ Usar pelo menos uma **variável de ambiente**.
 - ✓ **Expor** uma porta para acesso à aplicação.
 - A aplicação deve conter um **CRUD completo** (Create, Read, Update, Delete) e **persistir** os dados em **uma** tabela/coleção no Container de Banco.
 - Pode ser uma **API** ou aplicação **Web**.
- Container do **Banco de Dados**:
 - ✓ Pode usar uma imagem **pública** e ser executado sem **Dockerfile**, direto via **docker run**.
 - Deve conter:
 - ✓ **Volume** *nomeado* ou *anônimo* para persistência.
 - ✓ Pelo menos uma **variável** de ambiente
 - ✓ Porta **exposta** para acesso externo ao Banco.
 - ✓ Não será aceito o Banco H2 na entrega

Utilizar imagens **oficiais** do **Docker Hub** ou da **Oracle Container Registry**
(<https://container-registry.oracle.com>)

DEVOPS TOOLS & CLOUD COMPUTING 3/4

2.1 Regras para os Containers (Gerais)

Ambos os containers devem ser executados em **modo background (Segundo Plano)**.

Exibir os **logs** de **ambos** os **containers** no terminal após a execução.

Evidências **obrigatórias** via **terminal** (não usar *interface gráfica* do Docker Desktop).

3. ENTREGA

3.1 Repositório no GitHub (obrigatório)

- Repositório com tudo que é necessário para rodar o projeto:
 - ✓ Código-fonte da aplicação.
 - ✓ Arquivos Dockerfile.
 - ✓ Scripts ou JSONs usados nos testes do CRUD (no caso de APIs).

****Sem os arquivos Código Fonte e Dockerfile, o projeto não será avaliado.***

3.2 Vídeo Demonstrativo (obrigatório)

- ✓ O funcionamento **completo** da aplicação, desde o funcionamento do APP e a persistência de dados, mostrando:
 - Execução de todas as **operações** do **CRUD**
 - A **persistência** no **banco de dados após cada operação**.
- ✓ O vídeo **deve** conter:
 - **Explicação** em voz ou legenda clara de **cada** passo.
 - Boa qualidade visual e sonora (mínimo 740p).
 - Pode usar ferramentas como OBS, Clipchamp, etc.

DEVOPS TOOLS & CLOUD COMPUTING 4/4

Atenção!

A pontuação será baseada na **completude e qualidade da entrega**. Serão considerados: o uso correto do projeto de uma das disciplinas, a execução completa do CRUD, a apresentação das evidências no banco de dados, o cumprimento das regras técnicas solicitadas e a clareza do vídeo demonstrativo — com explicações por voz ou legenda sobre cada etapa apresentada.

- Não serão aceitas entregas com código **incompleto**, sem **Dockerfile** ou sem **vídeo** explicativo.
- O **uso** de Interface gráfica (Docker Desktop) para evidências **não será aceito**.

4.1 - Containerização de **JAVA ADVANCED** ou de **ADVANCED BUSINESS DEVELOPMENT WITH .NET** utilizando um banco de dados containerizado. (até 80 pontos)

4.2 - Projeto **completo** no GitHub. (até 20 pontos)

Siga todas as etapas e boas práticas.
Organização e clareza contam **pontos**!
Boa atividade!



DISRUPTIVE ARCHITECTURES: IOT, IOB & GENERATIVE IA

DESAFIO: SOLUÇÃO BASEADA EM IOT

Objetivo: Integrar a sua solução proposta a um sistema IoT que integre sensores e/ou atuadores para coletar, processar e exibir os dados em tempo real, utilizando plataformas e protocolos adequados para a comunicação entre os dispositivos.



DISRUPTIVE ARCHITECTURES: IOT, IOB & GENERATIVE IA

REQUISITOS TÉCNICOS MÍNIMOS:

- 1. Sistema IoT:** O sistema IoT deve possuir, no mínimo, três dispositivos (sensores, atuadores ou ambos), que devem coletar, processar ou interagir com base nos dados coletados.
- 2. Hardware:** A plataforma utilizada deve ser uma das abordadas durante as aulas (Arduino/ESP32). A solução pode ser desenvolvida e testada no simulador Wokwi ou em hardware real.
- 3. Dashboard:** Implementação de um painel de controle para monitorar e/ou controlar os dispositivos em tempo real.
- 4. Gateway:** Utilizar Node-RED, Thinger.io, ThingSpeak ou plataformas similares como gateway. O gateway deve gerenciar a comunicação entre os dispositivos IoT e o dashboard, incluindo fluxos de processamento, análise e visualização de dados.
- 5. Protocolos:** A comunicação deve ser feita utilizando o protocolo MQTT e/ou HTTP no padrão JSON para a troca de mensagens entre os dispositivos e o gateway. Utilize tópicos personalizados e únicos.



DISRUPTIVE ARCHITECTURES: IOT, IOB & GENERATIVE IA

ENTREGÁVEIS:

A entrega da solução IoT deve ser composta por um conjunto de elementos que comprovem a funcionalidade, a aplicabilidade e o impacto da proposta desenvolvida. Cada item entregue será avaliado considerando a clareza das instruções, a organização dos arquivos e a apresentação das funcionalidades implementadas, destacando o potencial de aplicação em situações reais de desastres ou emergências urbanas.

Para documentar e demonstrar a solução proposta, os seguintes elementos devem ser entregues, garantindo clareza, organização e replicabilidade dos resultados.



DISRUPTIVE ARCHITECTURES: IOT, IOB & GENERATIVE IA

1. Protótipo Funcional (até 50 pontos):

- Apresentação do protótipo operacional que demonstre a funcionalidade da solução IoT, incluindo a integração dos sensores e/ou atuadores e a comunicação entre dispositivos.
- A solução deve ser testada e validada em um ambiente simulado (Wokwi) ou em hardware real, comprovando a eficácia na coleta e transmissão de dados.
- É fundamental que o protótipo apresente uma aplicação prática alinhada e integrada ao contexto da sua solução proposta.

2. Documentação (até 30 pontos):

A entrega deve incluir um repositório GitHub bem estruturado, contendo todos os arquivos do projeto e o código-fonte devidamente comentado e organizado, com estrutura clara para replicação. A documentação deve conter:

- ⑩ Descrição completa da solução, com textos explicativos e imagens ilustrativas;
- ⑩ Instruções didáticas no README para configuração, execução e testes da aplicação;
- ⑩ Detalhamento dos fluxos implementados no Node-RED, quando aplicável, evidenciando a comunicação entre dispositivos e gateway;
- ⑩ Apresentação clara do código-fonte.



DISRUPTIVE ARCHITECTURES: IOT, IOB & GENERATIVE IA

3. Vídeo de Apresentação (até 20 pontos):

- Criação de um vídeo de até 3 minutos, apresentando de forma clara e objetiva o funcionamento da solução IoT, destacando as funcionalidades implementadas, a motivação do projeto e os benefícios previstos em cenários de risco ou emergência.

Arquivo de Entrega:

Um arquivo .txt contendo:

- Nomes completos dos integrantes do grupo;
- Link do repositório no GitHub;
- Link do vídeo no YouTube.

O arquivo deve ser organizado de forma clara para facilitar a identificação dos links e dos responsáveis pelo projeto.

Observações:

Atenção com os anexos. Entregas com arquivos corrompidos não serão considerados. Não serão considerados arquivos enviados pelo Teams.



JAVA ADVANCED ^{1/2}

Considerando o desafio proposto, você deve apresentar uma proposta de solução que utilize tecnologia e inovação para resolver ou mitigar os impactos dos eventos extremos na população. Uma parte desta solução deve ser uma API Rest para concentração dos dados e das regras de negócio da aplicação. Utilizando Spring Boot, você e sua equipe devem construir uma API seguindo os padrões de princípios apresentados nas aulas do primeiro semestre. A entrega deve contemplar os seguintes requisitos técnicos:

- API Rest atendendo as boas práticas da arquitetura
- Persistência em Banco de Dados Relacional utilizando Spring Data JPA
- Mapeamento de relacionamento entre entidades
- Validação com Bean Validation
- Paginação, ordenação e filtros
- Documentação da API com Swagger
- Autenticação com JWT
- Deploy em nuvem



JAVA ADVANCED 2/2

A entrega será **avaliada** da seguinte forma:

- Cumprimento dos requisitos técnicos e boas práticas (70 pontos)
- Viabilidade e Inovação (10 pontos)
- Documentação e Apresentação (20 pontos)

O documento de **entrega** deve conter os seguintes artefatos:

- Link dos repositórios com os códigos fontes
- Link dos deploys
- Instruções para acesso e testes
- Vídeo demonstração da solução completa (máx 10 minutos)
- **Vídeo pitch (máx. 3 minutos)**



MASTERING RELATIONAL AND NON-RELATIONAL DATABASE

1 Modelagem Relacional (3FN) – 20 pontos

- Diagrama lógico e físico com entidades bem normalizadas

2 Criação das Tabelas com Restrições – 10 pontos

- Tabelas criadas com PRIMARY KEY, FOREIGN KEY, NOT NULL, CHECK, etc.

3 Procedures DML por Tabela – 25 pontos

- Para cada tabela do sistema, criar:
 - 1 procedure de inserção
 - 1 procedure de atualização (update)
 - 1 procedure de exclusão (delete)
 - Executar as procedures para inserção de no mínimo 5 linhas, hard insert não serão considerados.

4 Funções para Retorno de Dados Processados – 10 pontos

- Criar pelo menos 2 funções
 - Ex: calcular risco médio, retornar total de ocorrências por região



MASTERING RELATIONAL AND NON-RELATIONAL DATABASE

5 Blocos Anônimos com Consultas Complexas – 10 pontos

- Criar **2 blocos anônimos distintos** que obedeça sua regra de negócio
 - Utilizando: JOIN, GROUP BY, HAVING, ORDER BY, subqueries, IF/ELSE, LOOP

6 Cursores Explícitos – 10 pontos

- Utilização de cursores para leitura de dados com OPEN, FETCH, CLOSE
- Uso dentro de um LOOP em um dos blocos anônimos ou procedures

7 Consultas SQL Complexas (relatórios) – 10 pontos

- Criar pelo menos 5 SELECTs que usem:
 - JOIN entre múltiplas tabelas
 - Agregações (SUM, COUNT, AVG)
 - GROUP BY, HAVING, subqueries e ORDER BY

8 Integração com Projeto Java (Framework Application) – 5 pontos

- Demonstrar a utilização do banco de dados Oracle no projeto desenvolvido em Java



MASTERING RELATIONAL AND NON-RELATIONAL DATABASE

Entrega Final (em PDF)

O grupo deverá entregar:

- Modelo relacional (diagrama lógico)
- Scripts SQL:
 - Criação de tabelas
 - Inserts
 - Procedures (INSERT, UPDATE, DELETE)
 - Funções
 - Blocos anônimos
 - Cursores
 - Consultas SQL
- Prints de execução dos códigos
- Explicação da integração com o Java
- **Vídeo** com demonstração do projeto funcionando



MOBILE APPLICATION DEVELOPMENT

A partir da solução escolhida na **Global Solution**, seu desafio é desenvolver um aplicativo usando React Native.

Entregáveis:

- Arquivo texto com o endereço do repositório no GitHub Classroom
- Inclua o nomes dos integrantes do grupo, o link pro vídeo no YouTube e a descrição da solução da **Global Solution** no arquivo README.md do repositório.

Requisitos para Avaliação

- (10 pts) Criar no mínimo 5 telas e utilizar navegação entre elas (React Navigation ou Expo Router).
- (40 pts) Implementar um CRUD utilizando a API criada na disciplina JAVA ou .NET (Axios ou Fetch).
- (10 pts) Estilização do aplicativo, incluindo cores, fontes, e imagens personalizadas.
- (20 pts) Arquitetura: será avaliada a organização dos arquivos, nomes das variáveis, funções e componentes utilizados.
- (20 pts) Elabore uma gravação de um vídeo de todas as funcionalidades do App



MOBILE APPLICATION DEVELOPMENT

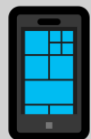
1. Telas e Navegação (React Navigation ou Expo Router) **10 pontos**

Requisitos:

- O aplicativo deve conter, no mínimo, 5 telas distintas com navegação fluida entre elas.
- A navegação deve seguir boas práticas de usabilidade e acessibilidade.

Avaliação:

- Pontos por navegação sem bugs e bem integrada com as funcionalidades.
- Avaliação da clareza e consistência da experiência de navegação.



MOBILE APPLICATION DEVELOPMENT

2. CRUD com API (Java/.NET) usando Axios ou Fetch **40 pontos**

Requisitos:

- Implementar operações de **Create, Read, Update, Delete** usando uma API RESTful desenvolvida nas disciplinas de backend (Java ou .NET).
- Utilização de bibliotecas adequadas (Axios ou Fetch) com tratamento de erros e feedback visual.

Avaliação:

- Os dados devem ser armazenados e manipulados via API, nunca apenas no dispositivo.
- Interface deve fornecer feedbacks visuais em caso de erro (mensagens, loaders, alertas).
- Avaliação da qualidade e semântica da integração com a API.



MOBILE APPLICATION DEVELOPMENT

3. Estilização com identidade visual personalizada **10 pontos**

Requisitos:

- Personalização de cores, fontes e imagens de acordo com o tema do app.
- Uso consistente do design, respeitando padrões de usabilidade.

Avaliação:

- Aplicação das guidelines da Apple/Google.
- Criatividade e identidade visual clara e funcional



MOBILE APPLICATION DEVELOPMENT

4. Arquitetura do Código **20 pontos**

Requisitos:

- Organização lógica de arquivos, pastas e componentes.
- Nomeação clara e padronizada de variáveis, funções e componentes.
- Separação adequada de responsabilidades (componentes, serviços, estilos, rotas, etc.).

Avaliação:

- Código limpo, legível e bem estruturado.
- Indentação correta e formatação padronizada (uso consistente de espaçamentos, quebras de linha, nomes descritivos, etc.).
- Adoção de boas práticas de desenvolvimento com React Native.
- Uso adequado de ferramentas de lint/formatter (como ESLint, Prettier ou configurador do próprio editor).
- Código mal formatado, com indentação irregular ou difícil leitura poderá resultar em perda de pontos.
- Bibliotecas utilizadas são relevantes, atualizadas e realmente necessárias. Pontos podem ser descontados por uso excessivo, desatualizado ou sem justificativa clara.



MOBILE APPLICATION DEVELOPMENT

5. Gravação de vídeo com todas as funcionalidades **20 pontos**

Requisitos:

- Apresentar o app em funcionamento real (emulador ou dispositivo).
- Demonstrar claramente as principais funcionalidades: navegação, CRUD, usabilidade, etc.
- Narração opcional, mas a apresentação deve ser clara.
- Duração máxima de 5 minutos.

Avaliação:

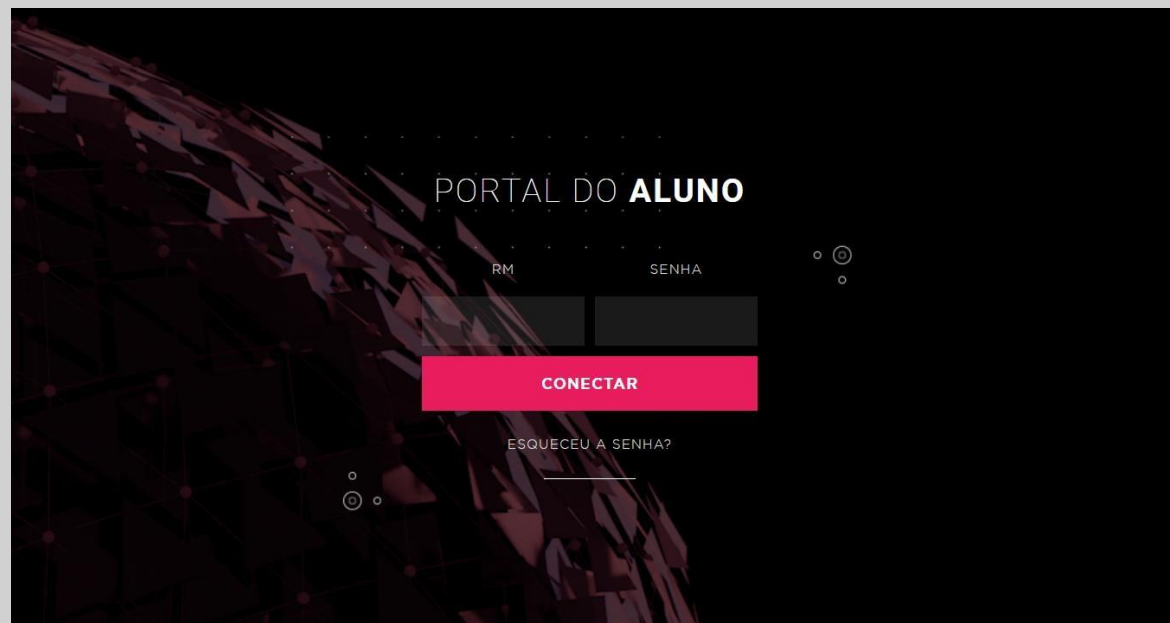
- Clareza na apresentação das funcionalidades.
- Demonstração da criatividade na resolução do problema proposto.
- Qualidade do vídeo e da demonstração geral do produto.

ENTREGA (PASSO A PASSO)

Como fazer as entregas da Global Solution?

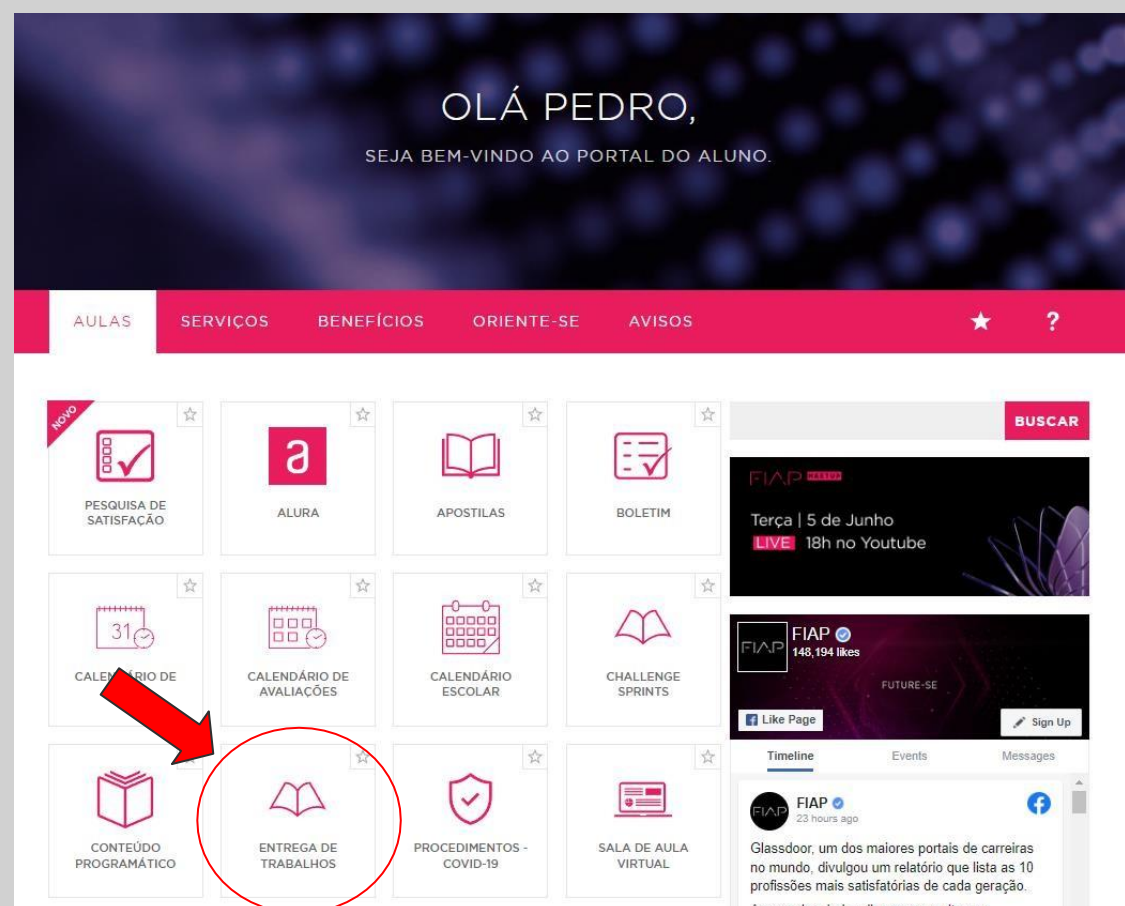
1.0 Acesse o Portal do Aluno FIAP

www2.fiap.com.br



ENTREGA (PASSO A PASSO)

2.0 Em Aulas, Clique na opção “Entrega de Trabalhos”



ENTREGA (PASSO A PASSO)

3.0 Clique em um trabalho referente a Global Solution

LISTA DE TRABALHOS		
1TDSB		
<div>07/06/2023 23:55:00</div> <div>NÃO ENTREGUE</div>	<div>BUILDING RELATIONAL DATABASE</div> <div>GLOBAL SOLUTION - BUILDING RELATIONAL DATABASE</div>	
<div>07/06/2023 23:55:00</div> <div>NÃO ENTREGUE</div>	<div>COMPUTATIONAL THINKING USING PYTHON</div> <div>GLOBAL SOLUTION - OBJETIVO DE DESENVOLVIMENTO SUSTENTÁVEL</div>	
<div>07/06/2023 23:55:00</div> <div>NÃO ENTREGUE</div>	<div>DOMAIN DRIVEN DESIGN</div> <div>GLOBAL SOLUTIONS - DDD</div>	
<div>07/06/2023 23:55:00</div> <div>NÃO ENTREGUE</div>	<div>RESPONSIVE WEB DEVELOPMENT</div> <div>GLOBAL SOLUTION 2023 - RESPONSIVE WEB DEVELOPMENT</div>	
<div>07/06/2023 23:55:00</div> <div>NÃO ENTREGUE</div>	<div>SOFTWARE DESIGN & TOTAL EXPERIENCE</div> <div>GLOBAL SOLUTION - SOFTWARE DESIGN & TOTAL EXPERIEN</div>	



ENTREGA (PASSO A PASSO)

4.0 Anexe o arquivo do seu projeto referente a entrega escolhida

Na página de entrega, você pode conferir o seu grupo, a data de vencimento, e a descrição da entrega.

ENTREGA DE TRABALHOS

INFORMAÇÕES DO TRABALHO

ANO	TURMA	DISCIPLINA
2023	ITDSB	SOFTWARE DESIGN & TOTAL EXPERIENCE

TEMA

GLOBAL SOLUTION - SOFTWARE DESIGN & TOTAL EXPERIENCE

DATA DE ENTREGA

07/06/2023 23:55

DESCRIÇÃO

GLOBAL SOLUTION - SOFTWARE DESIGN & TOTAL EXPERIENCE

INTEGRANTES

• 98043 - PEDRO CARVALHO PACHECO

COMENTÁRIOS

[Anexar Arquivos](#)

CADASTRAR COMENTÁRIO

ENTREGA DO TRABALHO

ARQUIVO

Tamanho Máximo: 50 MB.

Escolher arquivo

Nenhum arquivo escolhido

LINK DO ARQUIVO

Caso prefira ou o arquivo tenha mais que 50 MB, você pode fazer o upload do trabalho em um site de hospedagem de arquivos e enviar o link dele.

ENVIAR

Repita este mesmo processo para todas as outras entregas referentes que constam como **Global** Solution.



ENTREGA (PASSO A PASSO)

5.0 Confira o seu arquivo anexado.

O arquivo que você enviar na entrega fica registrado, você pode conferi-lo depois do envio.

ARQUIVOS ANEXADOS

- [Global Solution - Software Design & TX](#)

ENTREGA DO TRABALHO

ARQUIVO

[52ED5F5B-71FE-48CB-A3DC-D294B435F3E3.zip](#) (Entregue pelo(a) aluno(a) PEDRO CARVALHO PACHECO no dia 04/06/2023 às 07:28)

