



# FIA/P GRADUAÇÃO

# GLOBAL SOLUTION 2025/2

2º ANO

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

*Turmas de Fevereiro*

# O FUTURO DO TRABALHO

## **O futuro do trabalho já começou.**

Avanços tecnológicos, como a inteligência artificial, a robótica e a automação, estão transformando profissões, criando novas oportunidades e também desafios inéditos. Ao mesmo tempo, mudanças demográficas, sociais e ambientais estão redesenhando a forma como vivemos, trabalhamos e nos relacionamos.

## **Como serão os empregos em 2030? E 2050?**

O avanço da tecnologia está disruptando profissões e negócios, gerando muitas oportunidades e muitas ameaças e apenas por meio da educação e de boas ideias conseguiremos um futuro justo e próspero para todos.

# O FUTURO DO TRABALHO

**Segundo a ONU e a OIT**, até 2030, milhões de empregos poderão ser extintos, mas outros tantos serão criados, exigindo requalificação constante, adaptação ágil e a busca por competências humanas cada vez mais valorizadas — como criatividade, pensamento crítico, empatia e colaboração. O Fórum Econômico Mundial estima que quase um quarto das profissões deve mudar radicalmente até 2027, impulsionadas por tecnologias emergentes, transições verdes e modelos de negócio digitais.

**Na FIAP**, acreditamos que esse cenário não deve ser encarado como ameaça, mas como uma oportunidade única de reimaginar o trabalho e criar soluções que tornem o mundo mais inclusivo, justo e sustentável. Quando conectamos tecnologia, inovação e propósito, abrimos caminho para transformar vidas, organizações e sociedades inteiras.



# O DESAFIO FIAP

**Agora é a sua vez.** Imagine como será o futuro do trabalho em um cenário onde a tecnologia está em todos os lugares e as relações humanas se tornam cada vez mais centrais. Seu papel é propor uma solução que use inovação, tecnologia e boas ideias para melhorar a vida das pessoas, preparar organizações para novos tempos e criar oportunidades mais justas, inclusivas e sustentáveis.

Explore temas como novas formas de aprendizagem, saúde e bem-estar no trabalho; modelos de qualificação e requalificação (reskilling); uso de IA como parceira do ser humano; ambientes híbridos e imersivos, economia verde e novas formas de inclusão produtiva.



# O DESAFIO FIAP

**Vale usar tudo o que você sabe** — e explorar o que ainda está aprendendo. Crie protótipos, investigue tendências, conecte dados, explore IA, gamifique experiências e pense fora da caixa. O que você criaria para preparar jovens para carreiras que ainda não existem? Como reinventar ambientes de trabalho em um mundo remoto, híbrido ou imersivo? Que soluções podem garantir bem-estar, inclusão e desenvolvimento contínuo para todos? Quais profissões (e ferramentas que ainda não existem) precisarão ser criadas? Você está no comando. Solte a criatividade e mostre como grandes ideias nascem em momentos de grandes transformações.

# POR QUE ISSO IMPORTA?

**Porque diante de mudanças tão rápidas,** o conhecimento e a inovação se tornam as ferramentas mais poderosas de adaptação. A tecnologia pode personalizar jornadas de aprendizagem, automatizar tarefas repetitivas, criar plataformas de colaboração global ou até projetar ambientes virtuais de trabalho imersivos. Enquanto põe em cheque profissões de baixa especialização, abrem portas para outras oportunidades.

**A inovação pode estar no design de experiências inclusivas,** no uso de inteligência artificial para apoiar decisões, na criação de sistemas de saúde mental corporativa, na análise de dados para prever tendências de carreira ou no desenvolvimento de soluções que unem propósito, impacto e resultados.

# POR QUE ISSO IMPORTA?

**Com criatividade**, é possível repensar modelos de negócio, treinar equipes em ambientes virtuais, construir redes de colaboração global e transformar o futuro do trabalho em um espaço de realização, diversidade e progresso. Áreas como tecnologia, gestão, design e ciências humanas se conectam para criar respostas que aproximam, empoderam e abrem novos caminhos.

Quem sabe uma nova profissão não esteja surgindo e você estará na vanguarda de seu desenvolvimento?





# INSPIRAÇÕES PARA VOCÊ COMEÇAR

- Plataformas de upskilling e reskilling baseadas em IA
- Ambientes de trabalho imersivos em realidade virtual ou aumentada
- Ferramentas de monitoramento de bem-estar e saúde mental no trabalho
- Aplicativos para conciliar vida pessoal e profissional em regimes híbridos
- Sistemas de recrutamento ético e inclusivo apoiados por dados
- Soluções gamificadas para engajamento e motivação em equipes distribuídas
- Bots e agentes de IA como parceiros no dia a dia de trabalho
- Modelos de trabalho baseados em impacto social e sustentabilidade
- Comunidades de aprendizagem colaborativa e global
- Plataformas que conectam talentos a projetos de propósito

# CONEXÃO COM OS ODS DA ONU

Este desafio está alinhado com os Objetivos de Desenvolvimento Sustentável (ODS), especialmente:

ODS 8: Trabalho decente e crescimento econômico

ODS 4: Educação de qualidade

ODS 10: Redução das desigualdades

ODS 9: Indústria, inovação e infraestrutura

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 reinventar o futuro do trabalho — e transformar o mundo.



# MAIS INFORMAÇÕES SOBRE O ASSUNTO

A discussão sobre o futuro do trabalho tem ganhado espaço crescente nas agendas internacionais, motivada pelos impactos da automação, inteligência artificial, mudanças demográficas, desigualdades e pela urgência de assegurar condições dignas de trabalho para todos. Organismos como a ONU (Organização das Nações Unidas), a OIT (Organização Internacional do Trabalho) e diversas instituições dedicadas ao desenvolvimento econômico têm mapeado riscos, oportunidades e recomendações para políticas públicas, governança corporativa e ação social.

# 1. DISRUPÇÃO TECNOLÓGICA VS. MANUTENÇÃO DO EMPREGO DECENTE

A OIT, por meio da Comissão Global sobre o Futuro do Trabalho, aponta que as tecnologias disruptivas — automação, robótica, inteligência artificial — provocarão tanto a criação quanto a destruição de empregos. Em muitos casos, haverá uma compensação entre os dois efeitos, mas nem sempre de forma justa ou uniforme.

Estudos mais recentes, inclusive da comunidade acadêmica, sugerem que uma fração relevante de tarefas humanas já pode ser automatizada. Por exemplo, uma pesquisa publicada mostra que cerca de 25,1% das ocupações analisadas estão em risco substancial de automação, considerando a granularidade das tarefas.

## 2. A NECESSIDADE URGENTE DE REQUALIFICAÇÃO CONTÍNUA

Tanto a OIT como o Fórum Econômico Mundial enfatizam que a requalificação e o aprendizado ao longo da vida serão centrais para que os trabalhadores permaneçam relevantes num mercado de trabalho em rápida mutação. No relatório *Futuro do Trabalho 2025*, estima-se que 65% dos trabalhadores consideram essencial se requalificar e adquirir novas competências para manter sua empregabilidade.

As habilidades mais demandadas são frequentemente aquelas ligadas à tecnologia — como inteligência artificial, análise de dados, segurança cibernética — mas também competências “humanas” (comunicação, criatividade, pensamento crítico, empatia) ganham importância.

### 3. FLEXIBILIDADE, TRABALHO HÍBRIDO E MODELOS DIVERSIFICADOS

Estudos sinalizam que regimes híbridos e trabalho remoto tendem a se consolidar como padrão em muitas áreas. A preferência por empresas que oferecem flexibilidade cresce e já é fator relevante de retenção de talentos. Isso implica uma transformação não apenas nas políticas de RH, mas também em infraestrutura, cultura organizacional e regulação trabalhista.

## 4. RISCOS DE DESIGUALDADE E EXCLUSÃO

A ONU e a OIT alertam que o futuro do trabalho pode exacerbar desigualdades, caso as vantagens das novas tecnologias sejam concentradas em quem já detém recursos. Em especial, populações vulneráveis — jovens, mulheres, pessoas com menor escolaridade — correm maior risco de serem deixadas para trás.

Nesse sentido, a ONU incentiva programas de formação e requalificação profissional amplos, com políticas que visem garantir oportunidades para todos.

## 5. REGULAÇÃO, PROTEÇÃO SOCIAL E NOVAS FORMAS DE GARANTIA

Para que o trabalho do futuro seja digno, a OIT enfatiza a necessidade de fortalecer regulação trabalhista, tripartismo (governo, empregadores e trabalhadores) e redes de proteção social adaptadas aos novos modelos de emprego (ex.: trabalho por plataforma).

Além disso, autores jurídicos têm defendido a criação de instrumentos legais que coíbam a exclusão de trabalhadores frente à IA e automação, como fundos de amparo ou políticas de renda mínima para quem perder rendimento em transições tecnológicas.



## 6. TRANSIÇÕES “JUSTAS” E SUSTENTABILIDADE

A OIT sugere que a transição para novos modelos produtivos deve ser "justa", ou seja, planejada de modo que não penalize os mais vulneráveis, mas inclua programas de apoio, diálogo social e medidas compensatórias. Também se propõe que se alinhe com os Objetivos de Desenvolvimento Sustentável (ODS) da ONU, sobretudo ao **ODS 8 — Trabalho Decente e Crescimento Econômico**.

# PROJEÇÕES E TENDÊNCIAS

- O relatório *Futuro do Trabalho 2025* prevê que 170 milhões de novos empregos poderão ser criados entre 2025 e 2030, impulsionados por tecnologia, transformação verde e reconfigurações econômicas globais.
- Por outro lado, outro estudo do Fórum Econômico Mundial indica que cerca de 23% das profissões podem se transformar radicalmente até 2027.
- O potencial de automação de tarefas humanas, segundo pesquisas de IA, pode atingir um patamar em que 40% das tarefas sejam automatizadas nos próximos 5 anos e até 60% em 10 anos.

Esses números mostram que não se trata de um futuro distante, mas de uma transição já em curso.

# DESAFIOS E PARADOXOS

- **Quem vai possuir as máquinas?** O debate não é só sobre substituição de trabalho, mas sobre quem detém o capital tecnológico e como isso influencia distribuição de renda e poder. Será que soluções low cost e edge podem virar esse jogo?
- **Velocidade vs. preparação:** A adoção tecnológica muitas vezes é mais rápida do que a capacidade de adaptação de trabalhadores, de políticas públicas e sistemas educacionais.
- **Dilema da “formas de trabalho emergentes”:** Trabalhos por aplicativos, freelancing e gig economy ganham relevância, mas frequentemente sem proteção social, benefícios ou estabilidade.
- **Ética e dignidade:** Como assegurar que as máquinas aumentem a capacidade humana em vez de degradar o papel humano? A regulamentação deve acompanhar os avanços.
- **Inclusão:** A transição só será legítima se todos tiverem acesso à requalificação e não apenas aqueles em núcleos privilegiados.

# UM OLHAR ESPERANÇOSO: O QUE PODE DAR CERTO

- Mesmo com desafios significativos, boa parte da literatura aponta caminhos promissores:
- Políticas públicas integradas que combine educação, emprego, inovação e proteção social
- Adoção de plataformas de aprendizado adaptativo e acesso universal à formação contínua
- Incentivo à economia de plataformas justas e cooperativas que distribuam valor
- Fortalecimento de alianças tripartites para elaboração de normas trabalhistas inovadoras
- Investimento em infraestrutura digital, conectividade e inclusão digital
- Promoção de “transições justas” para garantir que ninguém fique para trás
- Modelos híbridos de trabalho que equilibrem flexibilidade, saúde mental e produtividade
- Enfoque em trabalho com propósito e impacto social, mais do que mera utilidade econômica



# PROGRAMAÇÃO

DATA	EVENTO
11/11	LIVE DE ABERTURA
07/11	CONTEÚDO PARA ALUNOS
19/11 até 23h55	ENTREGA NO PORTAL FIAP
Até 28/11	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.
- **Não** haverá chamada nos dias de aula, porém os professores estarão disponíveis durante as aulas para suporte, respeitando os dias das aulas remotas e presenciais.



# ENTREGAS

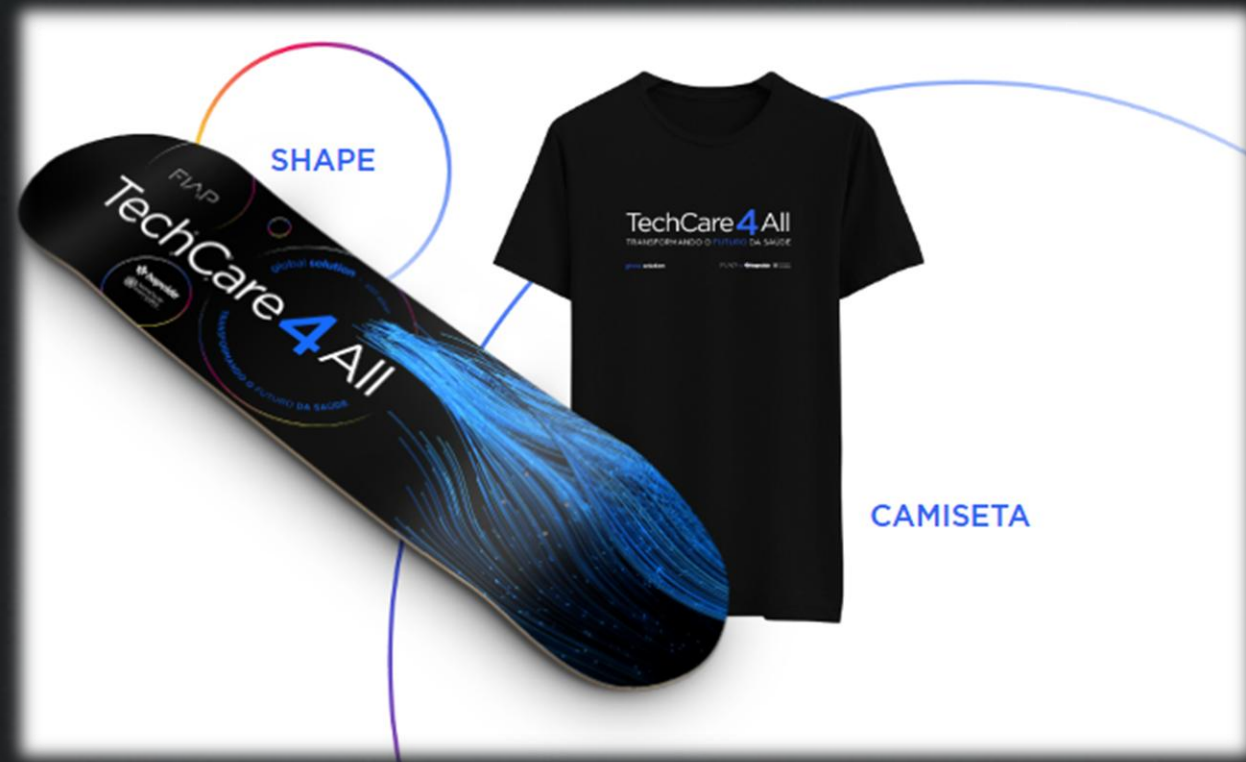
- As entregas de todas as disciplinas serão realizadas até o dia **19/11/25 até às 23h55** na área de trabalhos, no portal do aluno. *(Instruções de entrega estão no final do documento).*
- Entrega de um txt. no .zip de cada entrega: Esse txt deve ter o RM, nome do aluno e turma de cada integrante.



**Cadastro dos grupos (prazo até : 19/11):** <https://forms.cloud.microsoft/r/24AubMQb3x>



- 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 noventa 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 noventa, 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

Desenvolver uma API RESTful em .NET que represente uma solução tecnológica voltada ao tema “O Futuro do Trabalho”;

- **1- Boas Práticas REST (30 pts)**
  - Implementar paginação, HATEOAS e status codes adequados.
  - Uso correto dos verbos HTTP (GET, POST, PUT, DELETE).
- **2 - Monitoramento e Observabilidade API (15 pts)**
  - Implementar Health Check, Logging e Tracing
- **3 - Versionamento da API (10 pts)**
  - Estruturar versões da API (ex: /api/v1, /api/v2) com controle adequado em rotas.
  - Explicitar no README
- **4 - Integração e Persistência (30 pts)**
  - Integração com banco de dados relacional ou não relacional (Oracle, SQL Server ou MongoDB).
  - Utilização do Entity Framework Core com Migrations.
- **5 - Testes Integrados (15 pts)**
  - Implementar testes automatizados (unitários ou de integração) com **xUnit**;



# ADVANCED BUSINESS DEVELOPMENT WITH .NET

## Itens opcionais

- **ML.NET**
  - Integração de um modelo de aprendizado de máquina via ML.NET, demonstrando uma aplicação prática
- **Autenticação e Segurança**
  - Implementar autenticação via JWT ou API Key (opcional conforme o contexto do grupo).
  - Caso o projeto não possua usuários ou claims, pode-se optar apenas pela API Key.
- **Forma de Entrega**
  - Entregar via portal um arquivo com os seguintes itens
    - Link do repositório do código fonte no github ou azure Repos;
    - Links dos Deploys da API, com instruções para acessos e testes se aplicável (usuário, senha, url do banco, api key, etc).
    - Link do vídeo (Youtube ou equivalente) demonstrando o funcionamento da solução de forma integrada (máximo 5 minutos).



# COMPLIANCE, QUALITY ASSURANCE & TESTS

Crie um plano de projeto Scrum, no Azure DevOps Organizations com:

- 1- **(30%)** Features, todas elas com descrição e critério de aceite documentados. Tire um “print screen” da tela com a relação de Features e as telas com os detalhes de cada Feature, colando esses “prints” em um documento de resposta da prova. Inclua no documento de resposta da prova o link para a página inicial do seu projetos no Azure DevOps Services (Menu lateral -> Overview -> Summary)
- 2- **(40%)** Crie os casos de testes de sistema manuais dentro do Azure, controlando os dados de input e output esperados de cada teste e indicando a preparação a ser feita para que cada teste execute como previsto. Cada Feature deve ter ao menos 1 caso de teste de sistema manual. As entradas e saídas devem usar a funcionalidade "**Param Values**". O sumário deve conter os pré requisitos do teste (preparação a ser feita para que cada teste execute como previsto). Tire um “print screen” das telas que mostram a configuração do sumário e dos passos dos testes criados, colando esses “prints” em um documento de resposta da prova.



# COMPLIANCE, QUALITY ASSURANCE & TESTS

Crie um plano de projeto Scrum, no Azure DevOps Organizations com:

- 3- (**30%**) Execute os casos de testes usando o Azure e mostre os painéis de execução passo a passo e o painel de resumo dos testes realizados. Para cada teste o painel de execução passo a passo deve ser acessado em More options > View Test Results. Para cada Test Plan deve ser gerado um Gráfico de **Outcome**. Para cada Test Plan deve ser gerado um Gráfico de **Run by**. Tire um “print screen” das telas mencionadas e cole esses “prints” em um documento de resposta da prova.
- INSTRUÇÕES DE ENTREGA: Gere um .ZIP com o arquivo PDF gerado a partir do documento de resposta da prova e entregue na área de entrega conforme já orientado.

## ATENÇÃO!

- O professor que vai corrigir deve ser membro administrador do projeto [https://dev.azure.com/\\_settings/teams](https://dev.azure.com/_settings/teams);
- Entregas fora das especificações implicarão em perda de 10% na nota final da avaliação.



# DEVOPS TOOLS & CLOUD COMPUTING

## GS - Objetivo

Implementar **em nuvem** a solução de uma das disciplinas a seguir:

- **JAVA ADVANCED**
- **ADVANCED BUSINESS DEVELOPMENT WITH .NET**

A solução deve utilizar **Azure DevOps**, integrando as ferramentas:

- **Azure Boards**
- **Azure Repos**
- **Azure Pipelines**

A **infraestrutura** deve ser **provisionada via Azure CLI (Script)**.



# DEVOPS TOOLS & CLOUD COMPUTING

## Tarefas Obrigatórias

### 1) Provisionamento em Nuvem (Azure CLI):

- Crie via script os recursos necessários (ex.: Resource Group, Banco, Web App, Container etc.)

O Build e o deploy da Release devem utilizar esses recursos criados.

### 2) Projeto no Azure DevOps:

- Crie um novo projeto.
- Convide o professor (confirme o e-mail do seu professor) com as permissões:
  - Organização: **Basic**
  - Projeto: **Contributor**

### 3) Código no Azure Repos:

- Importe ou crie o repositório com o código-fonte da disciplina com a solução escolhida.

### 4) Azure Boards:

- Crie uma Tarefa inicial.
- Vincule commits, branches e pull requests a ela.





# DEVOPS TOOLS & CLOUD COMPUTING

## Tarefas Obrigatórias

### 5) Pipeline de Build:

- Crie no **Azure Pipelines** (YAML ou Classic).
- Deve rodar **automaticamente** a cada commit na branch principal.
- Publique artefatos e testes (JUnit, Xunit, Selenium etc.).

### 6) Pipeline de Release:

- Crie o fluxo de **deploy automático** na nuvem.
- Deve rodar automaticamente após a Build gerar um novo artefato.



# DEVOPS TOOLS & CLOUD COMPUTING

## GS - Requisitos da Implementação

### Requisitos da Implementação

- 1) Projeto **privado** e com **Git** para versionamento.
- 2) Azure **Boards** vinculado ao **Repos** (Commits, Branches, PRs).
- 3) Branch principal **protegida**:
  - Revisor obrigatório
  - Vinculação de Work Item
  - Revisor padrão (seu RM)
- 4) Build deve ser acionado somente **após Merge via PR**.
- 5) O aluno pode **aprovar sua própria PR** (simulação).
- 6) Release **executa automaticamente** após novo artefato.
- 7) **Deploy** pode ser feito por:
  - Container (ACI/ACR)
  - Web App PaaS



# DEVOPS TOOLS & CLOUD COMPUTING

## Requisitos da Implementação

8) Banco de dados em:

- **Container (ACI/ACR)** ou
- **Serviço PaaS (Azure SQL, MySQL ou Postgres)**

⚠ **Não misturar Containerização e PaaS!** O artefato e resultado dos testes (JUnit, Xunit, Selenium, Plan Tests etc.) devem estar publicados na Pipeline de Build (exceção para projetos em ACR/ACI que é obrigatório somente a publicação dos Testes)

9) Usar **imagens oficiais** (Microsoft, Oracle etc.).

10) Scripts de infraestrutura devem estar no repositório.

11) Arquivo script-bd.sql na pasta **/scripts**.

12) Scripts Azure CLI com prefixo **script-infra**.

13) Dockerfiles (caso use Docker) na pasta **/dockerfiles**.

14) Arquivo **azure-pipeline.yml** na raiz do repositório (YAML).

15) CRUD exposto em **JSON** no README do projeto (para APIs).

16) Utilizar **variáveis de ambiente** e proteger dados sensíveis.

17 Incluir **desenho macro da arquitetura**. - Ferramenta sugerida: [Visual Paradigm Azure Diagram](#)



# DEVOPS TOOLS & CLOUD COMPUTING

## Gravação do Vídeo (Explicativo e Avaliativo)

### Requisitos técnicos:

- Resolução mínima: 720p
- Áudio claro
- Explicação narrada por voz (sem legendas)

### Etapas:

Apresente o README explicando o conceito e arquitetura.

Mostre no Portal Azure os recursos criados pelos scripts.

No Azure DevOps, crie uma nova tarefa no Boards.

Faça o Merge e execute as Pipelines (Build + Release).

Mostre a execução completa.

Destaque o artefato e os testes CI.

Demonstre as alterações publicadas em nuvem.

Execute testes de CRUD (Create, Read, Update, Delete) em pelo menos duas tabelas.

Finalize mostrando a tarefa concluída com links (Commits, PR etc.).



# DEVOPS TOOLS & CLOUD COMPUTING

## Dicas

- 01) Apresente a execução de forma detalhada: inicie pela criação da Tarefa no Azure Boards, mostre a atualização do código fonte e percorra cada etapa e tarefa das Pipelines de Build e Release. Destaque os artefatos gerados e os testes executados que foram publicados no CI
- 02) Demonstre todas as operações de CRUD (Create, Read, Update, Delete) para cada tabela do banco de dados. Grave de forma contínua, sem cortes, e exiba os resultados diretamente no banco para comprovar que as operações foram executadas com sucesso
- 03) Antes de enviar a GS, assista ao vídeo completo e já postado no Youtube, garantindo se todos os requisitos foram atendidos, incluindo qualidade técnica (720p, áudio claro), explicações narradas e demonstração completa do fluxo de trabalho

## Entrega

Arquivo PDF com o nome: GS\_<nomeGrupo>.pdf

Inclua no arquivo PDF:

Folha de rosto com o nome do grupo, RM e nome completo do integrantes

Breve descrição da solução proposta

Link da Organização no Azure DevOps

Link do Vídeo no Youtube



# DEVOPS TOOLS & CLOUD COMPUTING

## Pontuação (**dependendo da completude da entrega**)

- 01) Desenho da Arquitetura Macro - até 10 pontos (desenho, fluxo, legenda etc)
- 02) Utilização do Azure Boards - até 10 pontos (Histórico, Links etc)
- 03) Utilização do Azure Repos - até 10 pontos (Branch, Commit, Merge etc)
- 04) Utilização do Azure Pipelines - Build - Até 35 pontos (Etapas, Tarefas, Testes etc)
- 05) Utilização do Azure Pipelines - Release - Até 35 pontos (Etapas, Tarefas etc)

## Critérios de Penalização

### **Penalidades que inviabilizam a correção (nota zero):**

- 01) Ausência do código fonte do projeto no repositório
- 02) Ausência do arquivo azure-pipeline.yml (quando utilizada implementação via YAML)
- 03) Vídeo sem narração por voz ou com uso de legendas em substituição à explicação falada
- 04) Aplicativo ou banco de dados entregues em localhost (não hospedados em nuvem)
- 05) Projeto sem acesso configurado ao Azure Boards, Repos ou Pipelines (qualquer dessas ferramentas)



# DEVOPS TOOLS & CLOUD COMPUTING

- 06) Projeto sem utilização do Azure DevOps
- 07) A solução apresentada deve ser de autoria exclusiva da equipe. Qualquer identificação de plágio ou cópia entre equipes resultará em nota zero para todos os envolvidos
- 08) Entregas fora do horário estipulado para a GS não serão corrigidas

## Penalidades parciais

- 09) Ausência do arquivo script-bd.sql: -5 pontos
- 10) Ausência de Dockerfiles (quando a solução utilizar containerização): -30 pontos
- 11) Ausência dos arquivos JSON para execução dos CRUDs de cada tabela: -10 pontos
- 12) Ausência de scripts do Azure CLI para criação de recursos: -10 pontos por script faltante
- 13) Omissão de itens obrigatórios no vídeo (conforme instruções de gravação): -20 pontos por item não apresentado (fora a penalidade de não completude da entrega)
- 14) Vídeo sem qualidade mínima de 720p ou com áudio sem clareza: -20 pontos
- 15) Ausência de demonstração de operações CRUD de cada tabela no banco em nuvem: -30 pontos (na falta de qualquer uma das operações)



# DEVOPS TOOLS & CLOUD COMPUTING

- 16) Falta de variáveis de ambiente ou exposição de senhas desprotegidas nas Pipelines: -20 pontos
- 17) Ausência de execução ou publicação de testes automáticos na Pipeline de Build: -15 pontos
- 18) Mistura de soluções de Containerização (ACI/ACR) com PaaS (Web App/Banco PaaS): -50 pontos





# DISRUPTIVE ARCHITECTURES: IOT, IOB & GENERATIVE IA

Desenvolva uma **aplicação baseada em Deep Learning** para resolver um problema apresentado, de forma **integrada** às demais disciplinas (como Desenvolvimento Web e Mobile).

O projeto deve demonstrar **integração com as outras disciplinas**, com interface funcional que consome os resultados do modelo de IA (por exemplo, via REST API, Firebase AI Logic, Hugging Face e outras).

## Requisitos técnicos

A solução deve obrigatoriamente envolver pelo menos **um** dos seguintes componentes de IA:

- **API de Visão Computacional**, por exemplo:
  - Classificação ou detecção de objetos em imagens/vídeos.
  - Reconhecimento facial, OCR, segmentação ou análise de padrões visuais.
  - Modelos pré-treinados (como MobileNet, YOLO...) ou modelos próprios.

## E/OU

- **API de IA Generativa**, por exemplo:
  - Geração de texto, imagens, música ou código com base em prompts.
  - Implementação via modelos como Gemini, GPT, Stable Diffusion ou outros.
  - Aplicação de técnicas de Prompt Engineering ou Fine-tuning.



# DISRUPTIVE ARCHITECTURES: IOT, IOB & GENERATIVE IA

## Critérios de Avaliação

- [até 60 pontos] Cumprimento INTEGRAL dos requisitos técnicos. - Implementação técnica da solução em Deep Learning, incluindo funcionamento da IA, integração da API, documentação do modelo e aderência aos requisitos obrigatórios.
- [até 20 pontos] Integração entre IA e outras disciplinas. - Grau de integração efetiva entre a solução de Deep Learning e as outras disciplinas, além de coerência na arquitetura geral do sistema.
- [até 10 pontos] Boas práticas de código. - Organização, clareza, documentação (Github README, comentários, etc.).
- [até 10 pontos] Apresentação (vídeo). - Clareza, objetividade, com ênfase na demonstração funcional da IA e integração interdisciplinar.



# DISRUPTIVE ARCHITECTURES: IOT, IOB & GENERATIVE IA

## Critérios de Avaliação

### Descontos e Penalizações

- [- 40 pontos] API de **IA sem integração** real com outras disciplinas (funciona isolada)
- [- 20 pontos] Código **sem README** explicativo ou sem instruções de execução
- [- 50 pontos] Código **não executável** (erro crítico sem solução)
- [- 60 pontos] **Não tem vídeo** de apresentação da solução
- [-35 pontos] Apresentação **sem demonstração funcional da IA**



# DISRUPTIVE ARCHITECTURES: IOT, IOB & GENERATIVE IA

## Forma de Entrega

Entregar via portal um arquivo com os seguintes itens

- Link do repositório do código fonte no github;
- Links dos Deploys da API, com instruções para acessos e testes se aplicável (usuário, senha, url do banco, api key, etc).
- Link do vídeo (Youtube ou equivalente) demonstrando o funcionamento da solução de forma integrada (máximo 5 minutos).



# JAVA ADVANCED 1/3

## Requisitos da Entrega

De acordo com o problema apresentado, sugira, analise e desenvolva uma solução web que colabore de forma impactante e viável para o tema.

A aplicação pode ser uma API ou um WebApp (full MVC), utilizando Spring Framework. Caso opte pela criação de uma API, a mesma deve ter um frontend web e/ou mobile.

A entrega deve atender aos seguintes requisitos:

- Utilização de anotações do Spring para configuração de beans e injeção de dependências
- Camada model / DTO com utilização correta dos métodos de acesso
- Persistência de dados com Spring Data JPA
- Validação com Bean Validation
- Aplicação adequada de caching para melhorar a performance
- Internacionalização dando suporte à pelo menos duas línguas
- Opção de paginação para recursos com muitos registros
- Spring Security para controle de autenticação e autorização
- Tratamento adequado dos erros e exceptions
- Mensageria com filas assíncronas
- Recursos de Inteligência Artificial Generativa com Spring AI
- Deploy em nuvem
- Para API REST: utilização adequada dos verbos HTTP e códigos de status



# JAVA ADVANCED 2/3

## Critérios de Avaliação

- Cumprimento dos requisitos técnicos: 60 pontos
  - O código entregue atende aos requisitos técnicos?
  - O código entregue utiliza boas práticas e apresenta um bom design e arquitetura?
- Relevância e Inovação: 10 pontos
  - O projeto aborda um problema ou desafio significativo na área energética?
  - A solução proposta tem potencial para solucionar o problema endereçado?
  - O projeto apresenta uma ideia inovadora ou uma abordagem criativa para resolver o problema?
  - A solução proposta incorpora tecnologias modernas e emergentes de forma eficaz?
  - Qual é o potencial impacto positivo sobre o tema ?
- Viabilidade e Usabilidade: 10 pontos
  - A solução proposta é tecnicamente viável?
  - Os alunos demonstraram compreensão das tecnologias utilizadas e sua aplicação no projeto?
  - A solução é fácil de usar pelos usuários e outros stakeholders?
- Demonstração: 10 pontos
  - A apresentação da proposta está clara e objetiva?
- Apresentação (vídeo pitch): 10 pontos
  - Os alunos apresentaram eficazmente o projeto e explicaram sua solução de forma compreensível?



# JAVA ADVANCED 3/3

## Forma de Entrega

Entregar via portal um arquivo com os seguintes itens

- Código fonte do software (github)
- Links dos Deploys em nuvem, com instruções para acessos e testes (usuário, senha, url do banco, api key, etc)
- Vídeo Pitch (youtube ou equivalente)
- Vídeo demonstrando o software funcionando com duração máxima de 10 minutos. O vídeo deve ter pelo menos o áudio com a explicação das funcionalidades apresentadas. (Youtube ou equivalente).



# MASTERING RELATIONAL AND NON-RELATIONAL DATABASE

## 1. Modelagem de Banco de Dados Relacional (10 Pontos)

- Desenvolver o modelo lógico e físico do banco de dados, garantindo que:
- O modelo esteja na **3ª Forma Normal (3FN)**, com todas as cardinalidades corretas.
- Cada entidade e relacionamento estejam coerentes com o tema “**O Futuro do Trabalho**” e com a solução proposta pelo grupo.
- O modelo seja implementado no **banco de dados relacional Oracle**.
- Criar todos os objetos necessários (**tabelas, chaves primárias e estrangeiras, constraints, etc.**).





# MASTERING RELATIONAL AND NON-RELATIONAL DATABASE

## 2. Procedures e Funções (20 Pontos)

- Criar **procedures** para realizar os *inserts* no banco de dados, utilizando parâmetros e garantindo consistência dos dados.
- Criar **duas funções personalizadas**, descritas a seguir, aplicadas ao contexto do tema “O Futuro do Trabalho”:
  - **Função 1:** responsável por converter dados relacionais em **JSON manualmente construído**, sem uso de funções internas do Oracle.
  - **Função 2:** responsável por **validações e cálculos lógicos**, utilizando expressões regulares e tratamento de exceções.
- Criar **triggers de auditoria** para registrar transações em cada tabela (**INSERT, UPDATE, DELETE**).
- Inserir **no mínimo 10 registros** em cada tabela utilizando as *procedures* de inserção — assegurando que os dados façam sentido com o contexto da aplicação (dados genéricos ou fora do tema serão desconsiderados).
- Implementar **tratamento de exceções e expressões regulares**, conforme boas práticas de **PL/SQL**.
- Além disso:
  - Criar uma **procedure** para exportar um **dataset no formato JSON**, contendo dados relevantes da aplicação (usuários, vagas, competências, cursos etc.).
  - Esse dataset poderá ser utilizado para **alimentar uma aplicação de Inteligência Artificial (IA)**, caso o grupo deseje implementar essa integração.
- **Nota Importante:** Procedures com *hard inserts* ou valores que não pertençam ao contexto do tema “O Futuro do Trabalho” serão desconsideradas.



# MASTERING RELATIONAL AND NON-RELATIONAL DATABASE

## Sobre a Função 1 — Conversão Manual para JSON (15 Pontos)

- Crie uma **Função** que receba dados relacionais e retorne uma **string no formato JSON**.
- A **lógica de conversão** deve ser desenvolvida integralmente pelo grupo, utilizando concatenação e manipulação de strings para construir o JSON.
- **Proibido o uso de funções internas/built-in do Oracle**, como TO\_JSON, JSON\_OBJECT, JSON\_VALUE, JSON\_QUERY, JSON\_TABLE ou similares.
- A função deve **tratar no mínimo 3 exceções distintas**, com mensagens personalizadas e registradas via DBMS\_OUTPUT ou em uma tabela de log.
- O JSON gerado deve conter dados coerentes com o contexto da aplicação — por exemplo, **informações de profissionais, competências e vagas** relacionadas ao tema *O Futuro do Trabalho*.
- *Exemplo de aplicação:* Gerar um JSON com o perfil de um profissional, suas competências e cursos recomendados, que poderá ser exportado e utilizado posteriormente em uma aplicação de IA ou módulo de recomendação de carreira.



# MASTERING RELATIONAL AND NON-RELATIONAL DATABASE

- Sobre a Função 2 — Validação e Cálculo de Dados (15 Pontos)
  - Crie uma **Função** responsável por realizar **validações ou cálculos lógicos** dentro do contexto da solução.
  - Utilize **expressões regulares (REGEXP)** para verificar formatação de campos (exemplo: e-mail corporativo, CPF, código de curso etc.).
  - Implemente **tratamento de exceções** com mensagens amigáveis, garantindo que o sistema não quebre em casos de erro de entrada ou ausência de dados.
  - A lógica deve refletir a proposta do tema “**O Futuro do Trabalho**”, como:
    - Cálculo de **nível de compatibilidade** entre perfil do candidato e competências exigidas em uma vaga;
    - Validação de **dados cadastrais** antes da inserção no banco;
    - Verificação da **aderência de um curso de requalificação** às competências em alta no mercado.
  - *Exemplo de aplicação:* Calcular automaticamente o percentual de compatibilidade entre um usuário e uma vaga, retornando uma mensagem JSON com o resultado da análise.



# MASTERING RELATIONAL AND NON-RELATIONAL DATABASE

- Empacotamento de Objetos de Banco de Dados (10 Pontos)
- Todos os objetos (**procedures** e **funções**) devem estar **empacotados** para garantir modularidade e reuso do código.
- O empacotamento deve seguir **boas práticas**, agrupando logicamente os objetos conforme suas funções (ex.: pacotes de usuários, competências, auditoria etc.).



# MASTERING RELATIONAL AND NON-RELATIONAL DATABASE

- Integração com Outras Linguagens do Curso (Java, C#, Mobile) (10 Pontos)
- A base de dados relacional deverá ser usada como **backend da aplicação**, desenvolvida em **Java, C# ou plataforma mobile**.
- As *procedures* criadas devem ser chamadas pela aplicação e sua execução demonstrada no vídeo de apresentação.



# MASTERING RELATIONAL AND NON-RELATIONAL DATABASE

- 5. Importação para MongoDB (10 Pontos)
- O dataset exportado em JSON deverá ser **importado para o MongoDB**, criando uma estrutura de dados coerente com o modelo **NoSQL**.
- A estrutura MongoDB deve respeitar os princípios de **organização flexível e eficiente de documentos**, facilitando consultas e integrações com APIs ou IA.



# MASTERING RELATIONAL AND NON-RELATIONAL DATABASE

- Demonstração em Vídeo (10 Pontos)
- Gravar um vídeo demonstrativo apresentando:
  - A execução das *procedures* sendo utilizadas no backend da aplicação.
  - A inserção de dados no banco relacional via *procedures*.
  - A exportação do dataset para JSON e sua importação no MongoDB.
  - Uma explicação de como essa integração contribui para o tema “**O Futuro do Trabalho**” — mostrando como a solução pode apoiar empresas e pessoas na era da automação, IA e requalificação profissional.



# MASTERING RELATIONAL AND NON-RELATIONAL DATABASE

- Entregáveis

- Modelos **Lógico e Físico** do banco de dados relacional desenvolvido usando o Oracle Data Modeler (em .pdf).
- Um arquivo .jpg com o modelo **Lógico** em notação Information Engineering (IE).
- Um arquivo .jpg com o modelo **Físico** em notação Information Engineering (IE).
- **Procedures e Funções** personalizadas com tratamento de exceções (em .sql).
- **Arquivo JSON** gerado a partir do banco relacional.
- **Estrutura MongoDB** (todos os código-fonte usados).
- **Vídeo demonstrativo** com a execução das etapas acima.





# MOBILE APPLICATION DEVELOPMENT<sub>1/9</sub>

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 6 telas e utilizar navegação entre elas (React Navigation ou Expo Router).
- (30 pts) Implementar um CRUD utilizando a API criada na disciplina JAVA ou .NET (Axios ou Fetch).
- (20 pts) Implementar autenticação no aplicativo
- (5 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.
- (10 pts) Elabore uma gravação de um vídeo de todas as funcionalidades do App
- (5 pts) Publicação do app



# MOBILE APPLICATION DEVELOPMENT<sub>2/9</sub>

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

### Requisitos:

- O aplicativo deve conter, no mínimo, 6 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<sub>3/9</sub>

## 2. CRUD com API (Java/.NET) usando Axios ou Fetch **30 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<sub>4/9</sub>

## 3. Sistema de Autenticação **20 pontos**

### Requisitos:

- Tela de cadastro (signup)
- Tela de login com autenticação real
- Logout funcional, limpando sessão e bloqueando rotas protegidas

### Avaliação

- Funcionamento completo do fluxo de autenticação, clareza e segurança mínima no processo.
- Validação dos formulários, tratamentos de erros e indicadores de carregamento devem estar presentes.
- Pode utilizar Firebase Authentication ou API desenvolvida nas disciplinas de JAVA ou .NET
- Proteção de rotas (telas restritas só acessíveis após login)



# MOBILE APPLICATION DEVELOPMENT<sub>5/9</sub>

## 4. Estilização com identidade visual personalizada **5 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<sub>6/9</sub>

## 5. 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<sub>7/9</sub>

## 6. Gravação de vídeo com todas as funcionalidades **10 pontos**

### Requisitos:

- Apresentar o app em funcionamento real (emulador ou dispositivo).
- Demonstrar claramente as principais funcionalidades: autenticação, 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.



# MOBILE APPLICATION DEVELOPMENT<sub>8/9</sub>

## 6. Publicação do app **5 pontos**

### Requisitos:

- O aplicativo deve estar publicado no Firebase App Distribution.
- O e-mail do professor deve ser adicionado como tester.
- Criar uma tela "Sobre o App", que exiba o hash do commit de referência.

### Avaliação

- A versão publicada deve corresponder exatamente ao código-fonte enviado.
- A tela "Sobre o App" não será contabilizada para o requisito 1.





# MOBILE APPLICATION DEVELOPMENT<sub>9/9</sub>

## Penalidades

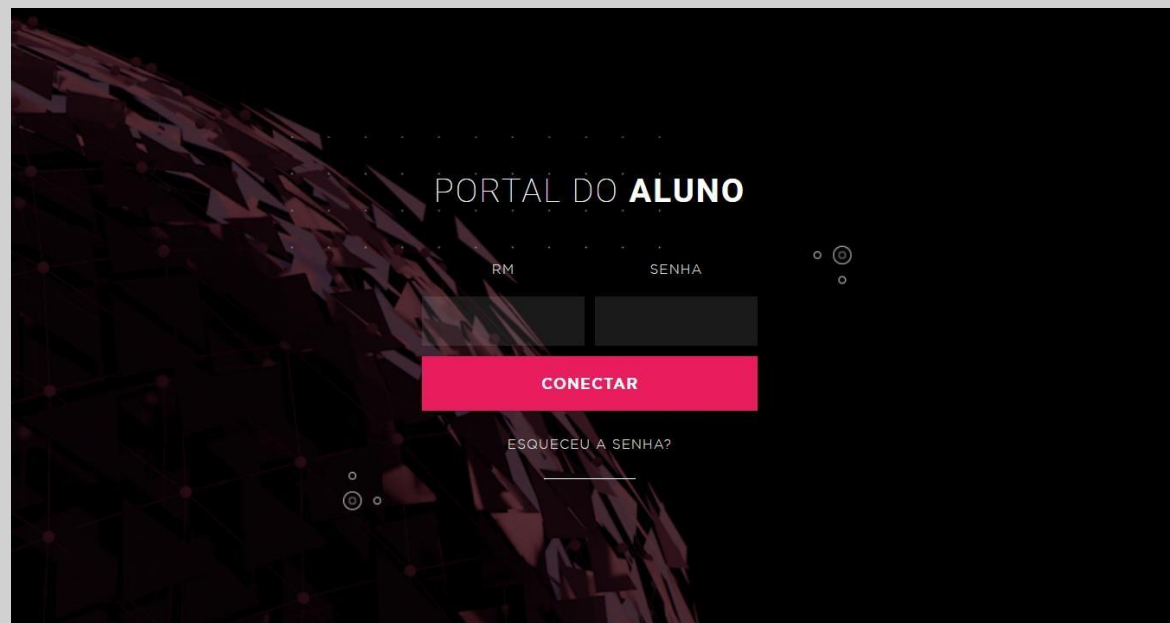
- I. Não entregar via GitHub Classroom (-20 pontos)
- II. Ausência do arquivo README.md (-10 pontos)
- III. Aplicativo fora do escopo das aulas (-60 pontos)
- IV. Histórico do Git incoerente ou confuso (-50 pontos)
  - É esperado que o repositório tenha uma árvore de commits sequencial e evolutiva, com mensagens claras e representando a construção real do app. Repositórios que não demonstrem envolvimento prático ou com uso superficial do Git poderão ser penalizados.
- V. Não publicar o aplicativo ou aplicativo publicado diferente do código. (-20 pontos)
- VI. Não conter o mínimo de 6 telas. (-10 pontos)
- VII. Ausência de estilização ou visual incompatível com um aplicativo. (-20 pontos)
- VIII. Não entregar o link para o vídeo no YouTube. (-20 pontos)

# ENTREGA (PASSO A PASSO)

## Como fazer as entregas da Global Solution?

### 1.0 Acesse o Portal do Aluno FIAP

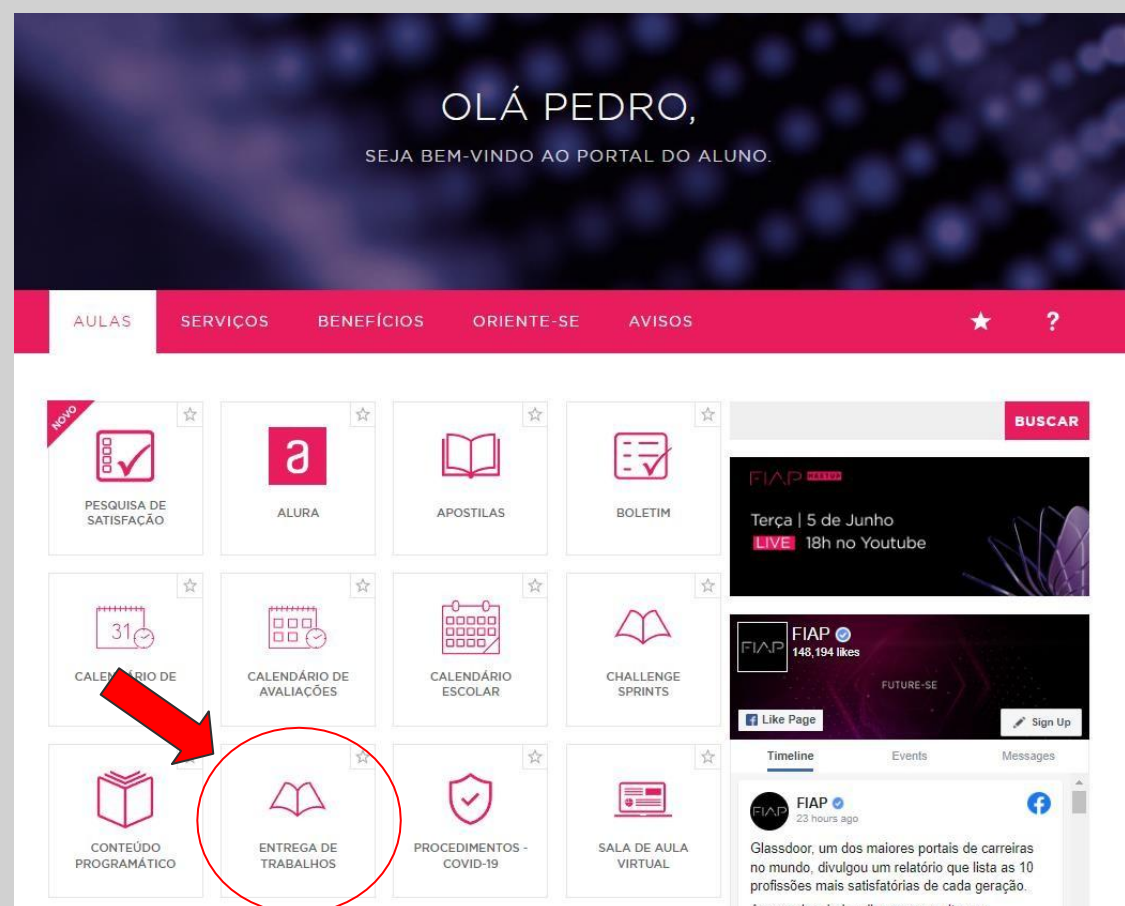
[www2.fiap.com.br](http://www2.fiap.com.br)





# 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><b>BUILDING RELATIONAL DATABASE</b></div> <div>GLOBAL SOLUTION - BUILDING RELATIONAL DATABASE</div>	
<div>07/06/2023 23:55:00</div> <div>NÃO ENTREGUE</div>	<div><b>COMPUTATIONAL THINKING USING PYTHON</b></div> <div>GLOBAL SOLUTION - OBJETIVO DE DESENVOLVIMENTO SUSTENTÁVEL</div>	
<div>07/06/2023 23:55:00</div> <div>NÃO ENTREGUE</div>	<div><b>DOMAIN DRIVEN DESIGN</b></div> <div>GLOBAL SOLUTIONS - DDD</div>	
<div>07/06/2023 23:55:00</div> <div>NÃO ENTREGUE</div>	<div><b>RESPONSIVE WEB DEVELOPMENT</b></div> <div>GLOBAL SOLUTION 2023 - RESPONSIVE WEB DEVELOPMENT</div>	
<div>07/06/2023 23:55:00</div> <div>NÃO ENTREGUE</div>	<div><b>SOFTWARE DESIGN &amp; TOTAL EXPERIENCE</b></div> <div>GLOBAL SOLUTION - SOFTWARE DESIGN &amp; 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)

