



# FIAP GRADUAÇÃO

# GLOBAL SOLUTION 2025/2

---

**1º 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 poe 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

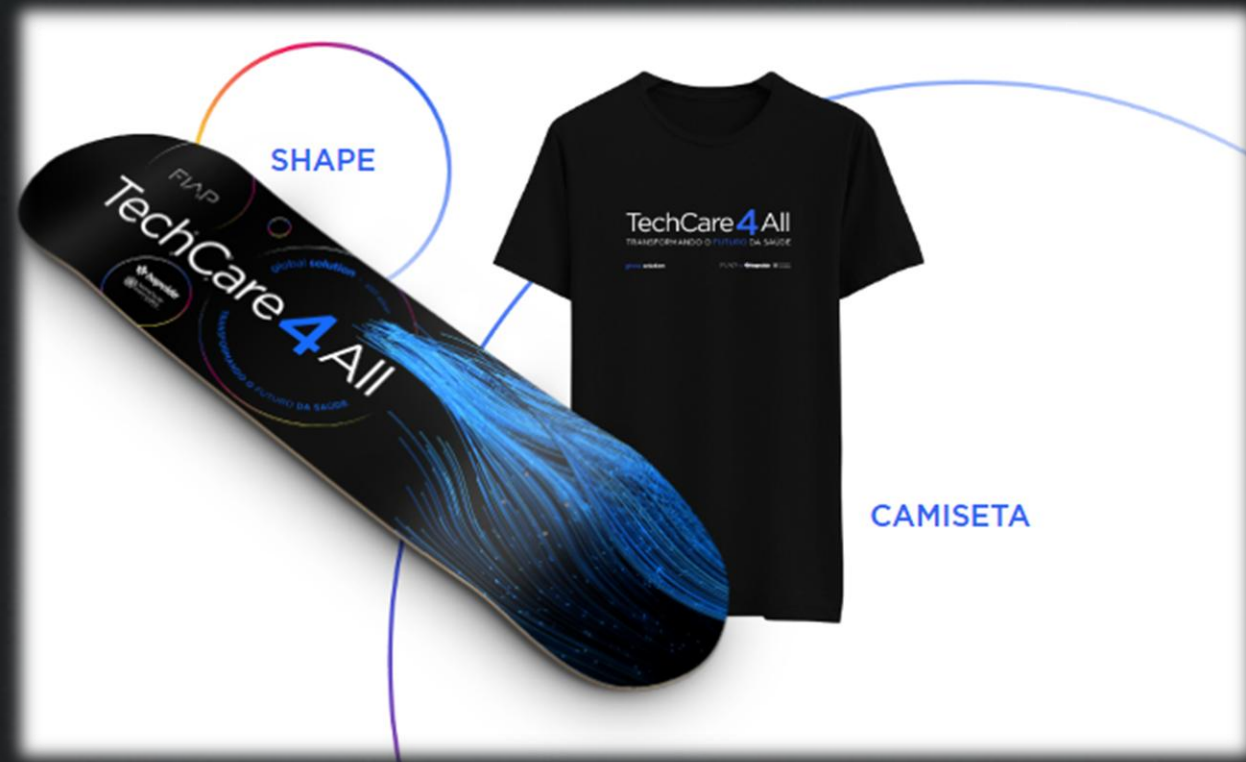
- As entregas de todas as disciplinas serão realizadas até o dia **19/11/25 até às 23h55** na área de trabalhos, no portal do aluno. *(Instruções de entrega estão no final do documento).*
- Entrega de um txt. no .zip de cada entrega: Esse txt deve ter o RM, nome do aluno e turma de cada integrante.



**Cadastro dos grupos (prazo até : 19/11):** <https://forms.cloud.microsoft/r/3temSvyu96>



- 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 nove em todas as disciplinas, junto com a noventa do vídeo Pitch da matéria de **SOFTWARE ENGINEERING AND BUSINESS MODEL**, 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**.



# SOFTWARE ENGINEERING AND BUSINESS MODEL

(Página 1 de 3)

Documento PDF único com identidade visual (fontes e cores) do grupo contendo:

- **Capa**
  - Nome da Startup
  - Nome completo e RM dos integrantes
- **Sumário (índice)**
- **[5 pontos] Sumário Executivo (resumo de cada tópico do documento)**
- **[20 pontos] Sobre o Negócio:**
  - [5 pontos] Detalhamento da oportunidade identificada
  - [5 pontos] Detalhamento do modelo de negócios da startup
  - [5 pontos] Modelo de relacionamento que será aplicado (B2B, B2C,...)
  - [5 pontos] Detalhamento do tipo de inovação da solução proposta
- **[30 pontos] Pitch de vendas:**
  - Video do Pitch de no máximo 5 minutos (link para visualização no Youtube) contendo material visual de apoio e imagem do aluno



# SOFTWARE ENGINEERING AND BUSINESS MODEL

(Página 2 de 3)

- **[25 pontos] Sobre o Produto:**
  - [5 pontos] Detalhamento das tecnologias aplicadas
  - [5 pontos] Backlog da Sprint (link do Jira ou Trello). Cada história deve conter:
    - História do Usuário no formato 3W (Eu como XXX Quero que XXX Para que XXX)
    - Requisitos Funcionais
    - Requisitos Não Funcionais
    - Regras de Negócio
  - [5 pontos] Diagrama de Casos de Uso da solução completa
  - [5 pontos] Diagrama de Atividades de cada caso de uso
  - [5 pontos] Link para o vídeo explicativo (no Youtube) do protótipo ou frontend final contendo:
    - Detalhamento de como cada heurística de Nielsen foi aplicada no sistema (demonstrar telas e jornada)
    - Detalhamento do tom de voz e boas práticas de UX Writing aplicadas no sistema (demonstrar telas e jornada)



# SOFTWARE ENGINEERING AND BUSINESS MODEL

(Página 3 de 3)

- **[5 pontos] Mercado e Competidores:**
  - Mercado-alvo (segmento de clientes)
  - Análise de Concorrentes
  - Vantagem competitiva (diferenciais da solução proposta)
- **[15 pontos] Plano de Investimento:**
  - [5 pontos] Precificação da solução proposta
    - Racional de despesas para formação da hora técnica
    - Racional do cálculo para o valor da Hora/Homem
    - Preço final para apresentação ao cliente
  - [5 pontos] Retorno do Investimento
    - Métricas tangíveis utilizadas
    - Racional do cálculo do ROI
  - [5 pontos] Acordo de Nível de Serviço
    - Determinação dos serviços a serem prestados
    - Determinação dos níveis dos serviços
    - Disponibilidade dos serviços (datas, horários, etc)
    - Determinação dos requisitos de “Help Desk” ou “Service Desk”
    - Informativo sobre multas/descontos em descumprimento dos SLA's



# ARTIFICIAL INTELLIGENCE & CHATBOT

Baseado no desafio, você e seu grupo devem construir pelo menos 2 modelos de IA que envolvam Regressão, Classificação e/ou Agrupamento. Em seguida, criem uma API REST usando Flask para que os modelos preditivos possam ser usados por outros sistemas.

## Entregáveis:

- Dados utilizados para treinar os modelos (arquivo **.csv**, **.xlsx**, ou similar) **[10 pontos]**
- Notebooks **.ipynb** com o desenvolvimento do pipeline de Machine Learning (análise exploratória dos dados, treinamento dos modelos de IA, avaliação de desempenho) **[60 pontos]**
- Modelos pré-treinados (**.pickle**, **.joblib** ou similar) **[10 pontos]**
- Código da API REST (**.py**) e integração **[20 pontos]**
- Nomes do membros do grupo e eventuais explicações adicionais em arquivo **.txt**

**Atenção:** caso o arquivo entregue esteja corrompido ou esteja faltando algum arquivo, o item será zerado.



# BUILDING RELATIONAL DATABASE

- A partir da solução proposta pelo grupo, para o desafio proposto pede-se:
  - Elaborar o modelo de dados lógico utilizando o Data Modeler, contendo:
  - Tabelas.
  - Atributos (pk, fk, nn, uk e ck, pelo menos um na base de dados do projeto);
  - Relacionamentos;
  - Lembrando que deve ser levado em consideração as classes desenvolvidas em backend– Java (anexar o diagrama de classes), para construção desta base de dados e estas serão usadas no processo de CRUD em Python.
- 
- Obs: a solução deverá conter no mínimo 4 tabelas e por tabela 4 atributos.



# BUILDING RELATIONAL DATABASE

- Parte prática - SQL:
- Script DDL da criação da estrutura do BD, contendo PKs, FKs, NN e etc. Utilize nomes significativos para as tabelas, colunas e restrições.
- Script DML para popular as tabelas com no mínimo 10 linhas em cada tabela.
- Script DQL/DRS (Selects) com pelo menos 4 consultas, incluindo texto explicativo do que se deseja recuperar do banco de dados, onde o objetivo é a obtenção de informações para tomada de decisão:





# BUILDING RELATIONAL DATABASE

- Parte prática - SQL:
- Uma consulta simples envolvendo SELECT/FROM/WHERE/ORDER BY.;
- Uma consulta envolvendo uma ou mais junções de tabelas contendo: SELECT/FROM/WHERE/ORDER BY;
- Uma consulta envolvendo função de grupo e agrupamento;
- Uma consulta envolvendo função de grupo, agrupamento com filtro (HAVING) e junção de tabelas



# BUILDING RELATIONAL DATABASE

- Critérios de Avaliação:
- Modelo de Dados Relacional, mais descritivo até 20 pontos;
- SCRIPT DDL até 20 pontos
- SCRIPT DML 20 pontos
- SCRIPT DQL até (40 pontos)



# COMPUTATIONAL THINKING USING PYTHON

## (1/3)

### Objetivo Central:

Implementar um sistema completo em Linguagem Python que realize ao menos, as 4 operações CRUD (Create, Read, Update e Delete) em um banco de dados Oracle. Sua solução deve interagir com as tabelas e o modelo dados previamente definidos no seu projeto.

### Documentação de especificação (.pdf) (40 pontos)

#### 1. Capa e Sumário (5 pontos)

- **Capa (2,5 pontos):** Deve conter o nome da sua solução e o nome completo de todos os integrantes da equipe.
- **Sumário (2,5 pontos):** Na página seguinte à capa, apresente o sumário detalhado, organizando a estrutura do documento com paginação correta para todos os tópicos, subtópicos e seu conteúdo.

#### 2. Especificação da Solução Proposta (35 pontos)

- **Explicação Detalhada (5 pontos):** Apresente um descritivo completo do seu projeto, funcionalidades principais e como ela atende aos desafios propostos.
- **Justificativas e Objetivos (5 pontos):** Inclua justificativas claras e objetivos bem definidos, com um mínimo de quinze linhas, que permitam a compreensão do contexto da sua solução. Este item é crucial para validar se as classes desenvolvidas refletem adequadamente o escopo do projeto.
- **Link para o vídeo Pitch (20 pontos):** Inclua claramente o link do seu vídeo Pitch, confirmando que ele está acessível publicamente.
- **Modelo físico e Scripts do Banco de Dados (5 pontos):** Apresente o modelo físico do banco de dados e inclua os scripts SQL completos para a criação das tabelas e a inserção de dados de exemplo.



# COMPUTATIONAL THINKING USING PYTHON

## (2/3)

### Código-Fonte Python (60 pontos):

1. **Menu Interativo:** O sistema deverá apresentar menu de opções claro e amigável, permitindo ao usuário navegar entre as principais funcionalidades oferecidas (CRUD e outras).
2. **Validação de Entradas:** Todas as entradas de dados fornecidas pelo usuário deverão ser devidamente validadas para garantir a integridade e correção das informações antes de serem processadas ou enviadas ao banco de dados.
3. **Tratamento de Exceções:** Implemente o tratamento adequado de exceções (try-except) para gerenciar erros inesperados (ex: falha de conexão com o banco, entrada de dados inválida, etc.), proporcionando uma experiência robusta e amigável ao usuário.
4. **Estruturas de controle:** Faça uso apropriado de estruturas de decisão (if, elif, else) e estruturas de repetição (for, while) para controlar o fluxo do programa e a lógica das operações.
5. **Modularização em funções:** O código deve ser organizado utilizando funções. Demonstre a passagem de parâmetros para as funções e o retorno de valores, promovendo a reutilização e a legibilidade do código.
6. **Consultas e Exportação em Arquivos JSON:** Realize **pelo menos 3 consultas diferentes** ao banco de dados Oracle, que tragam informações relevantes para a sua solução. Disponibilize ao usuário a opção de **exportar o resultado de cada uma dessas consultas para um arquivo no formato JSON**.



# COMPUTATIONAL THINKING USING PYTHON

## (3/3)

### Entrega:

1. Documentação de especificação (.pdf) (20 pontos)
2. Vídeo Pitch (20 pontos)
3. Código-Fonte Python (60 pontos)
4. Instruções de Instalação (README) - caso o projeto necessite de bibliotecas ou complementos externos que não fazem parte da instalação padrão do Python. Crie um arquivo README.txt ou README.md com as instruções detalhadas para instalação e execução do sistema.



# DOMAIN DRIVEN DESIGN USING JAVA

## (1/4)

### ■ Documentação (em PDF) – [30 pontos]

A documentação deve ser clara, objetiva e conter os seguintes itens:

- **Capa**
  - Nome dos integrantes.
  - Nome da equipe.
  - Nome da solução.
- **Sumário**
  - Lista dos tópicos do documento e numeração de página onde se encontram.
- **Objetivo e escopo do projeto**
  - Descrição objetiva e concisa da solução proposta. Caso a solução não tenha sido completamente implementada, descreva também o que foi possível implementar
- **Breve descrição das principais funcionalidades da solução**
  - Destaque das funcionalidades implementadas.
- **Tabela de Endpoints (API Restful)**
  - URIs (caminhos dos recursos).
  - Verbos HTTP (GET, POST, PUT, DELETE etc.).
  - Códigos de status de resposta esperados (200, 404, 500 etc.).



# DOMAIN DRIVEN DESIGN USING JAVA

FIAP

(2/4)

- **Protótipo – Prints das telas implementadas**
  - Capturas de tela reais do sistema com explicações resumidas.
- **Modelo de Entidade-Relacionamento (MER)**
  - Diagrama do banco de dados com entidades, atributos, chaves e relacionamentos.
- **Diagrama de Classes**
  - Representação das classes com atributos, métodos e relações (interface, herança etc.).

## Projeto Java Finalizado – [70 pontos]

A solução desenvolvida deve ser uma **API Restful** utilizando o **framework Quarkus**. A API entregue deve ser integrada com o Front-End da disciplina Front-End Design Engineering. A API deve conter:

- **Camada Model – [10 pontos]**
  - Contém todas as classes modelo necessárias, representando corretamente o banco de dados.
- **Camada DAO e BO – [20 pontos]**
  - Contém todas as funcionalidades essenciais para o Front-End.
  - O CRUD (Create, Read, Update, Delete) deve estar **funcionando completamente**.
  - Contém classe para conexão com banco de dados (padrões DAO / Connection Factory / Data Source).
  - Contém validações e regras de negócio adequadas.

ENTREGA



# DOMAIN DRIVEN DESIGN USING JAVA

## (3/4)

- **Camada Resource – [20 pontos]**
  - Implementação de todos os endpoints necessários para funcionamento do Front-End.
  - Deve seguir os princípios REST (recursos, verbos HTTP, status de resposta, etc.).
  - Contém a classe/configuração para Cors.
- **Boas Práticas – [10 pontos]**
  - Seguir regras de nomenclatura e organização do código.
  - Tratamento adequado de exceções.
  - Aplicação de padrões de projeto trabalhados em aula (por exemplo: DAO, MVC).
- **Deploy da Aplicação – [10 pontos]**
  - Deve realizar o **Deploy** da aplicação em qualquer servidor de aplicação (render, railway, etc).

### Observação:

Não é necessária uma classe de teste, desde que a API Restful esteja completamente implementada.





# DOMAIN DRIVEN DESIGN USING JAVA

FIAP

(4/4)

## ⚠ Forma de Entrega

Envio do arquivo através do portal do aluno com os seguintes itens:

- **Código Fonte** do software (**Zip e Github**)
- **Documentação** (Arquivo Único em formato **PDF**)
- **Link do Deploy**
- **Link do Vídeo Pitch** (publicado no Youtube ou equivalente)
- **Link do Vídeo de Demonstração:** deve mostrar o software funcionando com duração máxima de 10 minutos. O vídeo deve ser gravado com áudio e câmera aberta com explicação e demonstração das funcionalidades (publicado no Youtube ou equivalente)

### IMPORTANTE:

A não entrega de um dos itens acima, acarretará em um desconto de 20% da nota obtida.

ENTREGA



# FRONT-END DESIGN ENGINEERING

(01/14)

A equipe, nesta **Global Solution**, deverá utilizar **REACT + VITE + TYPESCRIPT**, promovendo uma arquitetura modular e uma aplicação **Single Page Application (SPA)**. Este ambiente permitirá maior **componentização** e eficiência no desenvolvimento, além de fortalecer a escalabilidade do projeto. Para completar nossa implementação é crucial à integração de **APIs** em nosso projeto de **Front-End Design Engineering** que esteja perfeitamente alinhado com o *backend* que consuma o *endpoint* desenvolvido na disciplina de **Domain Drive Design Using Java**, assegurando a integração destas disciplinas uniformemente.

O objetivo é proporcionar uma experiência de usuário **coesa e responsiva** em todos os dispositivos, com foco em acessibilidade e usabilidade. Também será essencial seguir os padrões técnicos estabelecidos, utilizando **GITHUB** para o versionamento. Essa prática garantirá o registro detalhado das alterações e facilitará a colaboração eficaz entre os integrantes da equipe.

**Para esta etapa vamos utilizar REACT + VITE + TYPESCRIPT**



# FRONT-END DESIGN ENGINEERING

ALGUMAS REGRAS (1/3)

(02/14)

## ▪ FRAMEWORK

- ✓ O projeto final deverá ser estruturado utilizando **REACT + VITE + TYPESCRIPT**, obrigatoriamente, onde o roteamento de páginas e outras adaptações necessárias serão implementados.

## ▪ DEPLOY DO PROJETO

- ✓ Além disso é fundamental realizar o *deploy* do projeto na plataforma **VERCEL**. A URL aqui criada, deverá ser disponibilizada ao professor responsável durante o período de avaliação, pois esta aplicação já deve estar consumindo remotamente a API criada na disciplina de Java para que os testes possam ser realizados.

## ▪ LINK DO GITHUB

- ✓ A **EQUIPE** deverá entregar o *link* do repositório ao professor responsável no README do projeto, para que a correção possa ser efetuada.
- ✓ O versionamento deve estar presente no projeto. É obrigatório utilizar o **GIT/GITHUB/GITFLOW** para realizar o versionamento, todos os integrantes devem participar do trabalho, sendo possível ver esta participação através do histórico de *commits*, por isso o **mínimo de *commit* por aluno é 5** e o **mínimo total de *commits* do projeto é 15**.



# FRONT-END DESIGN ENGINEERING

ALGUMAS REGRAS (2/3)

(03/14)

## ▪ APPLICATION PROGRAMMING INTERFACE (API) – JAVA – Obrigatória

- ✓ A **API**, desenvolvida na disciplina de *Domain Drive Design Using Java*, será responsável por enviar e receber dados entre o *backend* e o *frontend*. Ela trará todos os dados coletados no *backend* para o *frontend*, possibilitando também o envio de dados do *frontend* para o *backend*, viabilizando seu armazenamento no banco de dados, conforme abordado na disciplina de *Building Relational Database*. **Lembrando que tanto o front-end quanto o back-end devem estar publicados na nuvem.**

## ▪ PÁGINAS OBRIGATÓRIAS

- ✓ Em seu projeto deve constar estas páginas além das páginas da aplicação — index/home (Página Inicial), integrantes, sobre/about, FAQ (Perguntas Frequentes) ou contato. Os nomes podem variar destes aqui apresentados, desde que representem as mesmas funções.

## ▪ PÁGINA DE INTEGRANTES

- ✓ A página de integrantes deve ser criada como uma página do sistema e deve constar dos links comuns com os seguintes dados.
  - Nome, foto, RM e Turma de cada integrante
  - Link para GITHUB e LINKEDIN de cada integrante



# FRONT-END DESIGN ENGINEERING

ALGUMAS REGRAS (3/3)

(04/14)

## ▪ OBSERVAÇÕES

- ✓ A estilização deve ser realizada utilizando apenas **TAILWINDCSS**.
- ✓ Qualquer estilo em css puro encontrado será penalizado.
- ✓ À entrega do projeto deve ser realizada em formato .zip, sem a pasta **node\_modules** e com o histórico de versionamento, isso quer dizer que deve ser entregue uma cópia do repositório do projeto contendo *todo o histórico de versionamento* e não apenas os arquivos do projetos. **A branch main é obrigatória.**
- ✓ Utilizar o **README.md** do projeto para apresentar as informações importantes de desenvolvimento do projeto, além das informações obrigatórias como:
  - Estilizar com **MARKDOWN** este arquivo para que ele seja o indicador e orientador do projeto no repositório.
  - Crie seções: 1. Título e Descrição, 2. Status do Projeto, 3. Sumário, 4. Sobre o Projeto, 5. Tecnologias Utilizadas, 6. Instalação, 7. Como Usar, 8. Estrutura de Pastas, 9. Endpoints ou Rotas Principais, 10. Autores e Créditos, 11. Screenshots / Demonstração, 12. Contato
  - Estas são as informações que devem ser inseridas em contato: Nome, RM, Turma, Imagem, Links para LINKEDIN e GITHUB de cada integrante.
  - A disponibilização da URL da aplicação deve ser realizada nas seção 7 Como Usar.



# FRONT-END DESIGN ENGINEERING

## CRITÉRIOS DE AVALIAÇÃO E PONTUAÇÃO (1/6)

(05/14)

### 1. CONSTRUÇÃO DO PROJETO (20,0 pontos)

A construção da solução para esta *Global Solution* deverá ser feita utilizando o *framework* **REACT + VITE + TYPESCRIPT**.

#### I. Criação de Rotas estáticas e dinâmicas (passagem de parâmetros) com VITE(10,0 pontos)

##### a. Criação de Rotas e Passagem de Parâmetros (7,5 pontos)

- i. Definição clara e funcional de rotas utilizando, considerando boas práticas e tipagem do código. (2,5 pontos)
- ii. Implementação correta de parâmetros de rota e tratamento de rotas dinâmicas. (5,0 pontos)

##### b. Navegação (1,0 ponto)

- i. Navegação fluida e intuitiva entre as páginas, proporcionando uma experiência de usuário consistente. (1,0 ponto)

##### c. Redirecionamento (1,5 ponto)

- i. Utilização eficaz de redirecionamentos e *feedbacks* ao usuários com mensagens personalizadas.

#### II. Utilização de Tipos Específicos (10,0 Pontos)

##### a. Criação de Tipos de dados (**number, string, boolean, object**) (4,0 Pontos)

##### b. Tipos Avançados (**UnionTypes e Intersection**) (4,0 Pontos)

##### c. Interface. (2,0 Pontos)



# FRONT-END DESIGN ENGINEERING

(06/14)

## CRITÉRIOS DE AVALIAÇÃO E PONTUAÇÃO (2/6)

### 2. ESTILIZAÇÃO E RESPONSIVIDADE COM TAILWIND (35,0 pontos)

Este item avalia a qualidade visual da aplicação, verificando se o layout foi bem estilizado com Tailwind CSS e se a interface funciona corretamente em diferentes tamanhos de tela (responsividade).

#### a. Responsividade (15,0 pontos)

A responsividade devem ser representadas em diferentes resoluções de tela ( 05 ): eXtra Small devices ; SMall devices; MeDium devices; Large Devices e eXtra-Large devices.

#### a. Adaptabilidade (15,0 pontos)

- i. Funcionalidade e aparência adequada em diferentes dispositivos. (5,0 pontos)
- ii. Adequação do *layout* e conteúdo para diferentes tamanhos de tela / @media-queries. (10,0 pontos)

#### b. Tema Escuro/Claro (10,0 pontos)

A criação de um thema global é importante para dar opções aos usuários e também aqueles com dificuldades de enxergar tanto no Tema Claro, quanto no Tema Escuro. É obrigatório o uso de *Context* / *useContext* na aplicação dessa estratégia para evitar o prop-drilling.

- a. **Tema- Escuro / Claro:** Utilizando a Context Api do React (*useContext*), aplicar os referidos temas na aplicação inteira, tendo o controle de mudança, visível ou automático de acordo com o sistema do usuário. (10,0 pontos)



# FRONT-END DESIGN ENGINEERING

## CRITÉRIOS DE AVALIAÇÃO E PONTUAÇÃO (3/6)

(07/14)

### 3. DEPLOY DO PROJETO PARA PLATAFORMA VERCEL (10,0 pontos)

Avalia se o projeto foi publicado corretamente na plataforma Vercel e se o link de acesso está funcionando de forma pública e estável.

#### I. Sucesso no Deploy (10,0 pontos):

- Disponibilização de uma URL funcional para acesso ao projeto no arquivo **README.md**. (5,0 pontos)
- Efetivação bem-sucedida do *deploy* na plataforma Vercel. (5,0 pontos)

### 4. INTEGRAÇÃO ADEQUADA DAS APIs (20,0 pontos)

Avalia se as APIs foram integradas corretamente ao projeto, garantindo comunicação eficiente com os dados e funcionamento adequado das funcionalidades dependentes delas.

**ATENÇÃO: AQUI NESTE PONTO VOCÊ JÁ DEVE TER A URL REMOTA DA API, OU SEJA, A API DE JAVA JÁ DEVE TER SIDO PUBLICADA EXTERNAMENTE!!!**

#### I. Consumo da API (20,0 pontos)

- Consumo de API, utilizando request/response e as suas melhores práticas, manipulando os verbos HTTP, **GET/POST/PUT/DELETE**, CRUD. (6,6 pontos)
- Manipulação correta dos dados obtidos das requisições. (6,6 pontos)
- Tratamento de erros e respostas inesperadas. (6,6 pontos)





# FRONT-END DESIGN ENGINEERING

CRITÉRIOS DE AVALIAÇÃO E PONTUAÇÃO (4/6)

(08/14)

## 5. VERSIONAMENTO DO PROJETO NO GITHUB (10,0 PONTOS)

Avalia se o projeto foi versionado corretamente no GitHub, incluindo organização de commits, uso adequado do repositório e manutenção do histórico de desenvolvimento, aplicação do GitFlow no desenvolvimento, com a criação da estrutura padrão, utilizando os nomes de branches padrão, como develop e prefixos como feature,bugfix,release,hotfix. O projeto deve ter ao menos 2 (duas tags) geradas.

### I. Criação de repositório (2,5 pontos)

Criação de um repositório no GITHUB, para controle do projeto.

### II. Envio do link ao professor responsável (2,5 pontos)

Envio do link grupo no Item I, ao professor responsável pela disciplina em no arquivo README.MD.

### III. Commits frequentes (2,5 pontos)

Realização de, no mínimo, 05 (cinco) *commits* significativos, por integrante, demonstrando a evolução do projeto.

### IV. Participação da equipe (2,5 pontos)

Participação de todos os integrantes no repositório, com contribuições evidentes de cada membro da equipe.



# FRONT-END DESIGN ENGINEERING

## CRITÉRIOS DE AVALIAÇÃO E PONTUAÇÃO (5/6)

(09/14)

### 6. ARQUIVO README.MD (10,0 PONTOS)

#### I. README.MD (10,0 pontos)

O grupo deve criar um arquivo README.MD, formatado de acordo com MD (MarkDown), apresentando:

##### a. Todas as informações pertinentes para manipular o sistema. (3,4 pontos)

- Título e Descrição (0,28 ponto)
- Status do Projeto (0,28 ponto)
- Sumário (0,28 ponto)
- Sobre o Projeto (0,28 ponto)
- Tecnologias Utilizadas (0,28 ponto)
- Instalação (0,28 ponto)
- Como Usar (0,28 ponto)
- Estrutura de Pastas (0,28 ponto)
- Endpoints ou Rotas Principais (0,28 ponto)
- Autores e Créditos (0,28 ponto)
- Screenshots / Demonstração (0,28 ponto)
- Contato (0,28 ponto)

##### b. Links (6,6 pontos)

- GITHUB (3,3 pontos)
- Vídeo do YOUTUBE (3,3 pontos)



# FRONT-END DESIGN ENGINEERING

CRITÉRIOS DE AVALIAÇÃO E PONTUAÇÃO (6/6)

(10/14)

**7. Vídeo (5,0 pontos)**

Avalia a apresentação em vídeo do projeto, verificando se demonstra claramente o funcionamento da aplicação e explica suas principais funcionalidades e tecnologias utilizadas.

**I. Gravação de vídeo (5,0 pontos)**

Grupo deverá gravar um vídeo de no máximo 3 minutos apresentando os recursos do projeto, telas, *layout*, o mesmo deve ser disponibilizado via *link* e hospedado no Youtube.



# FRONT-END DESIGN ENGINEERING

INTEGRAÇÃO

(11/14)

- I. Building Relational Database
- II. Domain Drive Design Using Java (obrigatória a API estar hospedada remotamente)



# FRONT-END DESIGN ENGINEERING

## PENALIDADES (1/2)

(12/14)

- I. Arquivos maiores que 50 MB (-50,0 pontos)**  
O arquivo .ZIP do projeto não deve exceder 50 MB.
- II. A solução deverá ser entregue utilizando obrigatoriamente o framework VITEJS+TS (-100,0 pontos)**  
Caso a solução seja entregue utilizando outro framework, por exemplo o **NEXTJS**, serão descontados 100(CEM) pontos!
- III. Para cada página OBRIGATÓRIA que faltar no projeto (-20,0 pontos)**  
A ausência de qualquer página obrigatória — **index/home (Página Inicial)**, **sobre/about**, **FAQ (Perguntas Frequentes)** ou **contato** — resultará na perda de 20 pontos por página faltante no projeto.
- IV. Falta de página de identificação dos integrantes (-30,0 pontos)**  
A ausência de uma página com a identificação dos integrantes, incluindo nome, RM e turma, será penalizada.
- V. Para cada item de identificação da página dos integrantes que faltar (-60,0 pontos)**  
A ausência de cada item da página dos integrantes abaixo:
  - **Nome (-10)**
  - **Foto (-10)**
  - **RM (-10)**
  - **Turma(-10)**
  - **Link para GITHUBdo aluno (-10)**
  - **Link para o LINKEDIN do aluno (-10)**



# FRONT-END DESIGN ENGINEERING

## PENALIDADES (2/2)

(13/14)

### I. Não entrega do *link* do GITHUB no README.md (-50,0 pontos)

Sem o *link*, não há como avaliar o versionamento, o trabalho colaborativo e a evolução do projeto.

### II. README.md (-50 pontos)

A não entrega do arquivo **readme.md**, ou a entrega de outro tipo de arquivo em seu lugar acarretará uma penalidade de CINQUENTA pontos ao grupo.

### III. Arquivos do boiler plate ou arquivos estranhos à aplicação:( -10)

A não limpeza da estrutura do projeto e retirada dos arquivos do boiler plate e os arquivos que forem encontrados no projeto, será descontado dez(10) pontos por arquivo que for encontrado.

### IV. Utilização de frameworks, CDNs e/ou qualquer tipo de arquivo externo a solução entregue (-100,0 pontos)

A utilização de frameworks e/ou soluções prontas, como AXIOS, CARROUSEL, ACORDION incluindo o Bootstrap, RADIXUI, ou qualquer outro *framework*, está estritamente proibida neste projeto. Caso o professor responsável pela disciplina identifique a utilização desses recursos, a nota da **GLOBAL SOLUTION** da equipe será automaticamente **ZERADA**.

### V. Não entrega da API (-50,0 pontos)

A não entrega da API ou das APIs, implicará em um desconto de CINQUENTA pontos da nota final.

### VI. Entrega da solução somente com *link* (-100,0 pontos)

Não serão aceitas soluções que contenham apenas o *link* do repositório para que o professor responsável faça o download da solução a partir do repositório. É necessário o envio o arquivo ZIPADO.



# FRONT-END DESIGN ENGINEERING

## ENTREGA

(14/14)

- A equipe deve encaminhar ao professor responsável pela disciplina um arquivo ZIP contendo:
  - Compacte TODA a solução e encaminhe TUDO junto num único arquivo ZIP.
  - Lembre-se que o deve ser enviado é a cópia do repositório contendo todo o versionamento do projeto com as últimas atualizações, somente com *branch main*.
    1. Não serão aceitas soluções que contenham apenas:
    2. O *link* do repositório para que o professor responsável faça o *download* da solução a partir do repositório.
    3. O projeto sem versionamento.

## ONDE DEVE SER ENTREGUE

- Portal do Auno
- Anexe o arquivo do seu projeto referente a entrega escolhida. Lembre-se que somente o **representante** deve enviar o trabalho!!

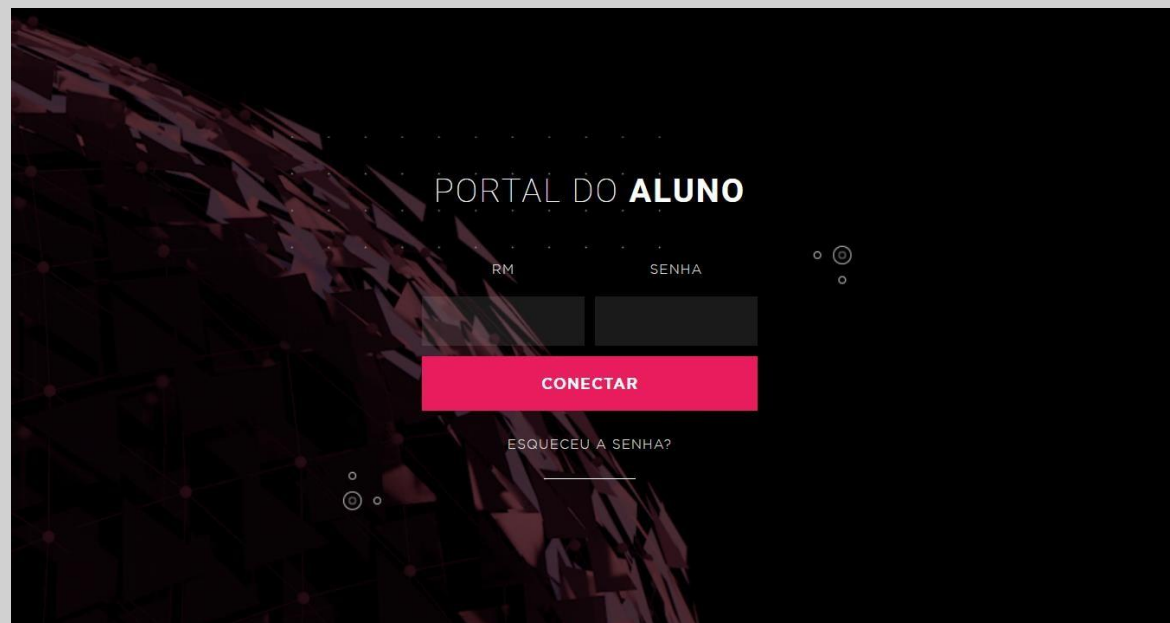
**NÃO SERÃO ACEITAS ENTREGAS PELO TEAMS OU OUTRO MEIO DE COMUNICAÇÃO!!**

# 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)

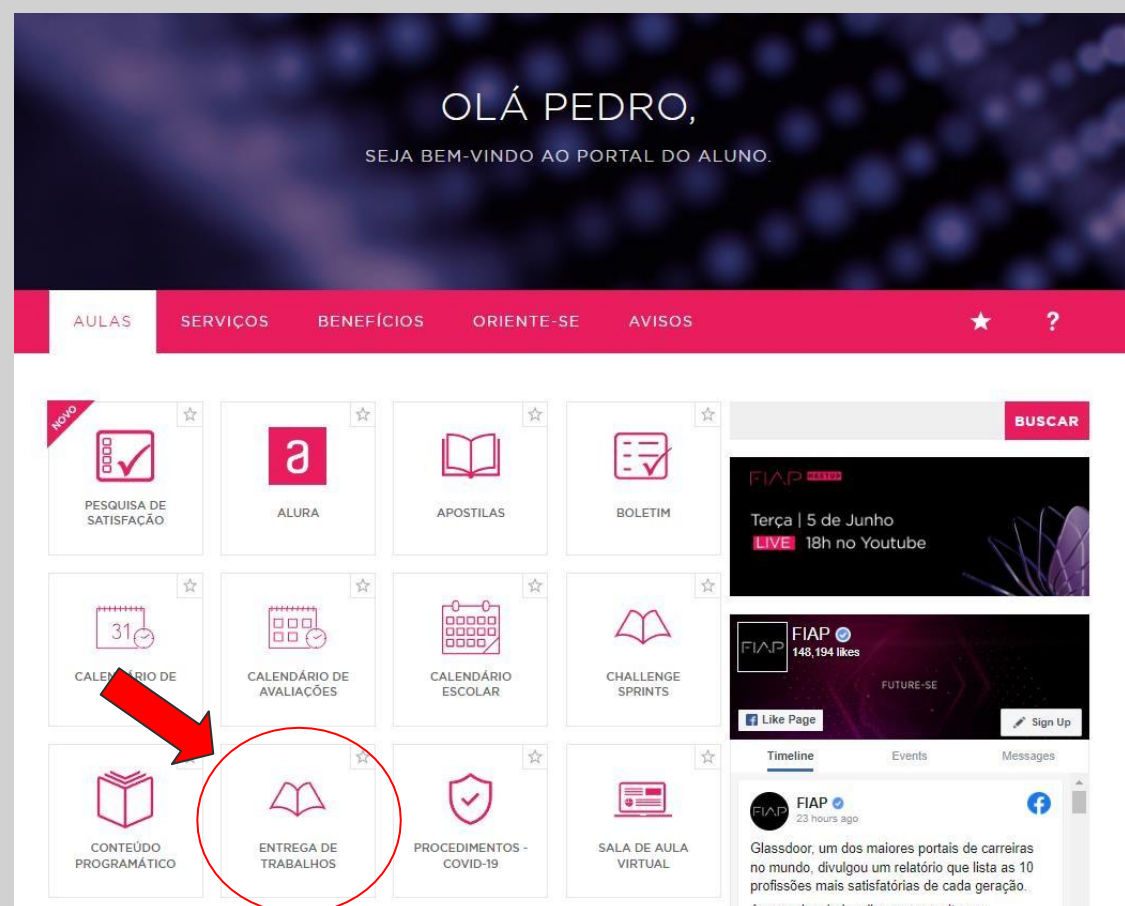






# ENTREGA (PASSO A PASSO)

## 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.

The screenshot displays the 'ENTREGA DE TRABALHOS' interface. The main section, 'INFORMAÇÕES DO TRABALHO', contains a table with the following data:

ANO	TURMA	DISCIPLINA
2023	ITDSB	SOFTWARE DESIGN & TOTAL EXPERIENCE

Below the table, the 'TEMA' is 'GLOBAL SOLUTION - SOFTWARE DESIGN & TOTAL EXPERIENCE' and the 'DATA DE ENTREGA' is '07/06/2023 23:55'. The 'DESCRIÇÃO' section shows the same topic. At the bottom, the 'INTEGRANTES' list includes '98043 - PEDRO CARVALHO PACHECO'.

The sidebar on the right, titled 'COMENTÁRIOS', includes a 'CADASTRAR COMENTÁRIO' button. Below it, the 'ENTREGA DO TRABALHO' section has an 'ARQUIVO' subsection. A red circle highlights the 'Escolher arquivo' button, with a red arrow pointing to it. The text 'Nenhum arquivo escolhido' is next to the button. Below this is a 'LINK DO ARQUIVO' section with a text input field and an 'ENVIAR' button.

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)

