FIAP GRADUAÇÃO

CHALLENGE 2025

1º ANO

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Turmas de Fevereiro – 2º Semestre

CRONOGRAMA

2º SEMESTRE

CRONOGRAMA

DATA	EVENTO	STEAKHOLDER
25/08/2025*	Mentoria com os Professores	Professores
09/09/2025*	Apresentação para Banca de Professores – Seleção dos Projetos	PROFESSORES
Semana do 22/09*	Apresentação dos selecionados para Banca Final HC	HC
28/09/2025	ENTREGA DA SPRINT 3	ALUNO
05/10/2025	Feedback das entregas SPRINT 3	PROFESSORES
09/11/2025	ENTREGA DA SPRINT 4	ALUNO
16/11/2025	Feedback das entregas SPRINT 4	PROFESSORES
08/11/2025	NEXT	FIAP

AGENDA

* As datas poderão ser alteradas

APRESENTAÇÕES – 2º SEMESTRE

- **A primeira apresentação para a banca de professores** têm como objetivo fornecer orientações e direcionamentos para o desenvolvimento dos projetos.
- **A segunda apresentação para a banca de professores** será voltada à avaliação dos projetos que serão **selecionados para a apresentação final à HC**.
- **A apresentação final para a HC** servirá para selecionar os melhores projetos, que participarão do evento NEXT e concorrerão à premiação na competição.

3º ENTREGAS

POR DISCIPLINA

SOFTWARE ENGINEERING AND BUSINESS MODEL

(1 de 2) Documento PDF único com identidade visual (fontes e cores) do grupo contendo:

- **Capa**
  - Nome da Startup
  - Nome completo e RM dos integrantes
- **Sumário**
- **Sobre o Negócio**
  - **[10 pontos]** Detalhamento da oportunidade identificada atualizada (mínimo de 20 linhas)
  - **[10 pontos]** Detalhamento da solução proposta e diferenciais atualizada (mínimo de 20 linhas)
  - **[20 pontos]** Link para visualização vídeo do Pitch (com no máximo 3 minutos, sem uso de IA)

ENTREGA - 3º SPRINT

SOFTWARE ENGINEERING AND BUSINESS MODEL

(2 de 2)

- **Sobre o Produto**
  - **[10 pontos]** Backlog da Sprint atual (link do Jira ou Trello). Cada história deve conter:
    - História do Usuário no formato 3W
    - Requisitos Funcionais
    - Requisitos Não Funcionais
    - Regras de Negócio
  - **[25 pontos]** Diagrama de Casos de Uso da solução completa (no mínimo 3 casos de uso)
  - **[25 pontos]** Para caso de uso da solução proposta:
    - Detalhamento da funcionalidade
    - Especificação do Caso de Uso relacionado
    - Diagrama de Atividades

ENTREGA - 3º SPRINT

ARTIFICIAL INTELLIGENCE & CHATBOT

- **Descrição:** Você e sua equipe deverão levantar uma base de dados para o treinamento de uma IA.
- **Requisitos:**
  - Dados levantados aderentes ao tema do desafio. **[+50 pontos]**
  - Explicação dos dados (fonte ou como foram coletados, o que significam, quantidade e qualidade, características importantes, rótulos, qual o objetivo que se pretende realizar com eles). **[+50 pontos]**
- **Entrega:**
  - Arquivo **.zip** com os dados. Os dados podem ser tabelas em .csv, imagens, áudios, etc., desde que tenham aderência ao tema;
  - Arquivo **.txt** com as explicações e nomes dos membros do grupo.
- **Penalidades:**
  - Entrega fora do prazo. **[-100 pontos]**
  - Entrega de arquivos errados ou corrompidos. **[- pontuação do item do respectivo arquivo faltante]**

ENTREGA - 3º SPRINT

BUILDING RELATIONAL DATABASE

Entregar a atualização da modelagem da Sprint anterior seguindo as correções e sugestões do seu professor

- Desenvolver o Modelo Físico de Dados / Modelo Relacional
- Criar o modelo físico (entregar o arquivo sql obrigatoriamente, com os comandos DDLs).
- O modelo relacional deve apresentar além das restrições (CONTRAINTS) chave primária (PRIMARY KEY), chave estrangeira (FOREIGN KEY) e obrigatória (NOT NULL) , pelo menos uma restrição única/exclusiva UNIQUE e uma restrição de verificação/validação CHECK, de acordo regras de negócio da solução proposta.
- Todos os nomes devem estar padronizados, conforme nomenclatura trabalhada nas aulas: nomes de tabelas, nome de colunas e nome de restrições.
- É, obrigatório, utilizar a ferramenta Oracle Data Modeler para construir o MER (modelo relacional). A não utilização dessas ferramentas irá acarretar desconto na nota.

ENTREGA - 3º SPRINT

BUILDING RELATIONAL DATABASE

ENTREGA - 3º SPRINT

Entregas:

Arquivo pdf com a modelagem conceitual, lógica relacional e física - 15 pontos

Arquivo sql com a codificação de construção das tabelas – 35 pontos

Estrutura com as colunas Pks corretas – 10 pontos

Estrutura com as colunas Fks corretas – 10 pontos

Estrutura com as colunas Not Nulls corretas – 10 pontos

Estrutura com as colunas Unique corretas – 10 pontos

Estrutura com as colunas Check corretas – 10 pontos

Penalidades, com suas pontuações.

Arquivo fora do padrão solicitado, menos 10 pontos

Codificação com erros de execução, ficará sem avaliação.

Qualquer esquecimento de regras será penalizado em, menos 5 pontos

COMPUTATIONAL THINKING USING PYTHON

- PROJETO PYTHON:
- **A partir dos dados coletados nas outras disciplinas, a equipe deverá desenvolver um programa em Python, o qual deverá contemplar os seguintes itens:**
  - [10 pontos] Menu com submenus para uso do sistema e acesso ao CRUD.
  - [30 pontos] definição de funções para organizar, filtrar e armazenar os dados em sequências (listas de listas, lista de tuplas e/ou lista de dicionários) - priorize os conteúdos vistos em sala de aula, **considere as boas práticas, passagem de parâmetro e retorno de funções;**
  - [20 pontos] tratamento de erros para inserção, alteração e exclusão dos dados (try, except, else, finally);
  - [10 pts] onde houver necessidade, deve haver validação de dados (entrada e processamento).
  - [30 pontos] Vídeo explicativo de até 5 minutos contendo os nomes dos integrantes, a ideia, a solução proposta, explicação do código fonte e execução do mesmo.

ENTREGA - 3º SPRINT

COMPUTATIONAL THINKING USING PYTHON

- Entrega:
  - Entrega em formato .zip do código fonte, arquivo contendo os nomes dos participantes e link para o vídeo (youtube)
- Penalidades:
  - [30 pontos] - Funcionamento do CRUD ou incompleto
  - [30 pontos] - Explicação coerente do código fonte (implementação)
  - [20 pontos] - Entrega fora dos padrões solicitados

ENTREGA - 3º SPRINT

DOMAIN DRIVEN DESIGN USING JAVA (1/6)

✓ Descrição

- Esta sprint do projeto Java tem como objetivo avaliar o desenvolvimento técnico e documental da solução proposta pela equipe. A entrega deve conter tanto o projeto Java funcional quanto a documentação detalhada, permitindo ao avaliador compreender claramente o funcionamento, o escopo e a lógica implementada. A entrega é individual por equipe, e deve ser feita no portal do aluno, em formato `.ZIP`, contendo **a documentação em PDF** e **a pasta do projeto Java completo**.

ENTREGA - 3º SPRINT

Requisitos

Documentação (em PDF) [20 pontos]

A documentação deve estar clara, organizada e conter obrigatoriamente os seguintes itens:

- **Capa**
  - Nome dos integrantes.
  - Nome da equipe.
  - Nome da solução (sistema desenvolvido).
- **Sumário**
  - Gerado automaticamente ou manualmente, listando os tópicos da documentação com numeração de páginas.

DOMAIN DRIVEN DESIGN USING JAVA (2/6)

- **Objetivo e escopo do projeto**
  - Texto claro, conciso e objetivo explicando a solução desenvolvida.
  - Deve conter o propósito principal do sistema e os limites de sua atuação.
- **Descrição das funcionalidades**
  - Breve explicação das principais funcionalidades implementadas.
  - Pode ser em forma de lista ou tópicos com descrição.
- **Protótipo do sistema**
  - Telas do sistema (mesmo que não estejam implementadas ainda).
  - Pode ser feito no Figma, PowerPoint, Canva, ou similar.
  - Cada tela deve vir acompanhada de explicações sobre como o usuário irá interagir com ela.
- **Modelo de Entidade-Relacionamento (MER)**
  - Diagrama do banco de dados com entidades, atributos, relacionamentos, e chaves.
- **Diagrama de Classes**
  - Atualizado conforme a estrutura da aplicação.
  - Deve representar as classes, atributos, métodos, relacionamentos (herança, composição etc.).

ENTREGA - 3º SPRINT

DOMAIN DRIVEN DESIGN USING JAVA (3/6)

- **Procedimentos para rodar a aplicação**
  - Lista de instruções claras para executar o projeto.
  - Deve conter:
    - Ferramentas utilizadas (ex: Eclipse, NetBeans, IntelliJ, VS Code).
    - Versões necessárias (Java JDK, banco de dados).
    - Passo a passo da execução.
    - Caso necessário, como importar o projeto.

💻 Projeto Java (Código-Fonte)

O código deve estar funcional e estruturado com base nos padrões trabalhados em sala de aula. Os seguintes requisitos devem ser atendidos:

- **Camada Model [20 pontos]**
  - Classes modelo corretamente estruturadas, com atributos, construtores, métodos getters e setters.
  - Padrões de encapsulamento e boa prática.

ENTREGA - 3º SPRINT

DOMAIN DRIVEN DESIGN USING JAVA (4/6)

- **Métodos com lógica de negócio [20 pontos]**
  - Implementação de **no mínimo 4 métodos relevantes**.
  - Os métodos devem conter **lógica e regras de negócio** condizentes com a proposta do sistema.
  - A complexidade e a criatividade da solução serão consideradas na avaliação.
- **Classe de Teste [10 pontos]**
  - Uma classe com o método `main` apenas para **testar os métodos implementados**.
  - Deve instanciar objetos e simular a utilização da aplicação.
- **Classe de Conexão com o Banco de Dados [10 pontos]**
  - Deve conter a lógica de conexão com o banco.
  - Os dados de conexão (usuário e senha) devem estar **inseridos no código**.
- **Camada DAO e CRUD completo [20 pontos]**
  - Implementação completa da camada DAO.
  - Métodos de Create, Read, Update e Delete devem estar **funcionais**.
  - As operações devem ser testadas e aplicadas às classes modelo.

ENTREGA - 3º SPRINT

DOMAIN DRIVEN DESIGN USING JAVA (5/6)

Pontuação Total: 100 pontos

Item	Descrição	Pontuação
Documentação completa	Itens listados acima	20 pontos
Camada Model	Estrutura correta com boas práticas	20 pontos
Métodos com lógica de negócio	Mínimo 4 métodos relevantes	20 pontos
Classe de Teste	Método main testando funcionalidades	10 pontos
Classe de Conexão	Incluindo usuário e senha no código	10 pontos
Camada DAO com CRUD	Implementação completa e funcional	20 pontos
Total		**100 pontos**

ENTREGA - 3º SPRINT

DOMAIN DRIVEN DESIGN USING JAVA (6/6)

ENTREGA - 3º SPRINT

⚠ Penalidades e Descontos

Situação	Penalidade
Ausência de qualquer item obrigatório na documentação	- até 5 pontos por item, além da pontuação atribuida.
Entrega fora do padrão solicitado	-10 pontos
Código que não compila ou não executa corretamente	-20 a -50 pontos (dependendo da gravidade)
Falta de pelo menos um dos métodos com lógica de negócio	-5 a -10 pontos por método faltante
CRUD incompleto ou com falhas graves	-10 pontos por item
Entrega após o prazo (não serão aceitos)	Penalização total (-100 pontos)

FRONT-END DESIGN ENGINEERING

Para a **Sprint 03** deste Challenge, as páginas desenvolvidas na **Sprint 02** servirão de base para reestruturação e novas funcionalidades.

A equipe 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.

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.

ENTREGA - 3º SPRINT

Para esta etapa vamos utilizar REACT + VITE + TYPESCRIPT.

FRONT-END DESIGN ENGINEERING

- **Criação de Conta de Equipe**
  - ✓ A **EQUIPE** deverá entregar o link do repositório na plataforma **GITHUB** ao professor responsável, para que a correção possa ser efetuada.

- **Definição de Páginas do Projeto**
  - ✓ **TODAS** as **páginas obrigatórias** da **Sprint 02**, além das páginas adicionais criadas pela equipe, deverão ser reestruturadas em **REACT** + **VITE** + **TypeScript** seguindo o conceito de **componentização**. Essa abordagem trará modularidade e escalabilidade ao projeto, permitindo componentes reutilizáveis e uma estrutura de fácil manutenção e crescimento.
  - ✓ O objetivo é transformar o HTML da **Sprint 02** em uma **Single-Page Application (SPA)**, com uma estrutura semântica e responsiva e navegação entre páginas por rotas. Esse projeto seguirá as melhores práticas discutidas em *sala de aula*, o versionamento no **GitHub** documentará o progresso, garantindo uma experiência de usuário coesa em diversos dispositivos.

ENTREGA - 3º SPRINT

A NÃO ENTREGA DE PÁGINAS OBRIGATÓRIAS ACARRETARÁ EM PENALIDADES!

FRONT-END DESIGN ENGINEERING

ALGUMAS REGRAS (2/3)

- **Integrações**
  - ✓ A aplicação **NÃO** necessita, neste momento de integração com uma **API**. Contudo, se houver elementos interativos de entrada de dados do usuário, devem ser obrigatoriamente implementados e validados através de HookForms. Ex: formulários de cadastro, campos de pesquisa etc. Lembrando que agora utilizamos **Typescript** e todo o código deve seguir este padrão.
  - ✓ **Nesta Sprint, não teremos integrações com outras disciplinas.**
- **Proibições**
  - ✓ É estritamente **proibido** o uso de **bibliotecas, frameworks, templates** ou **exercícios fornecidos durante as aulas**. O descumprimento dessa norma resultará automaticamente na atribuição de nota **ZERO** para a equipe. Ex: Axios, BootStrap etc.

ENTREGA - 3º SPRINT

FRONT-END DESIGN ENGINEERING

ALGUMAS REGRAS (3/3)

- **Observações**

  - ✓ A estilização deve ser realizada utilizando apenas **TAILWINDCSS**.

  - ✓ 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 **total de commits do projeto é 15**.

  - ✓ 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 contend a branch main e não apenas os arquivos do projetos.

  - ✓ Utilizar o **README.md** do projeto para apresentar as informações importantes de desenvolvimento do projeto até este, além das informações obrigatórias . Estilizar com **MARKDOWN** este arquivo para que ele seja o indicador e orientador do projeto no repositório.

ENTREGA - 3º SPRINT

FRONT-END DESIGN ENGINEERING

ENTREGA - 3º SPRINT

CRITÉRIOS DE AVALIAÇAO E PONTUAÇÃO (1/5)

1. CONSTRUÇÃO DO PROJETO DE ACORDO COM A SPRINT 02 (40,0 pontos)

- Desenvolvimento dos documentos HTML reestruturados em **REACT + VITE + TYPESCRIPT**.

I. COMPONENTIZAÇÃO, MODULARIDADE E REUTILIZAÇÃO DAS PÁGINAS DA SPRINT 02 (20,0 pontos)

a. Componentização (10,0 pontos)

i. É obrigatório entregar todos os documentos HTML da **Sprint 02** reestruturados utilizando **REACT + VITE** + **TYPESCRIPT**. A reescrita deve seguir os conceitos de componentização e a arquitetura de uma aplicação SPA (***Single Page Application***), garantindo uma estrutura semântica e responsiva, definindo de forma clara e funcional dos componentes/funções principais, considerando boas práticas e tipagem do código. A não entrega dos arquivos da **Sprint 02**, conforme os requisitos estabelecidos, resultará em penalidades, impactando diretamente na avaliação do projeto. (10,0 pontos)

b. Modularidade e Reutilização (10,0 pontos)

i. Identificação e separação clara de componentes, **REACT + VITE** + **TYPESCRIPT**. (3,3 pontos)

ii. Facilidade de reutilização de elementos em diferentes partes do projeto. (3,3 pontos)

iii. Eficiência na manutenção e atualização de componentes. (3,4 pontos)

FRONT-END DESIGN ENGINEERING

ENTREGA - 3º SPRINT

CRITÉRIOS DE AVALIAÇAO E PONTUAÇÃO (2/5)

1. CONSTRUÇÃO DO PROJETO DE ACORDO COM A SPRINT 02 (40,0 pontos)

- Desenvolvimento dos documentos HTML reestruturados em **REACT + VITE + TYPESCRIPT**.

I. **UTILIZAÇÃO DE HOOKS, PROPS E ROTAS** (20,0 pontos)

a. **Hooks, Props, e Rotas** (20,0 pontos)

i. Utilização de useState(). (2,5 pontos)

ii. Utilização de useNavigate(). (2,5 pontos)

iii. Utilização de useParams(). (2,5 pontos)

iv. Utilização de useEffect(). (2,5 pontos)

v. Utilização de Props. (5,0 pontos)

vi. Utilização de rotas estáticas / Dinâmicas. (5,0 pontos)

FRONT-END DESIGN ENGINEERING

ENTREGA - 3º SPRINT

CRITÉRIOS DE AVALIAÇAO E PONTUAÇÃO (3/5)

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

I. Estilização (10,0 pontos)

a. Escolha de Cores e Fontes (5,0 pontos)

i. Coerência e consistência das cores e fontes em todo projeto. (5,0 pontos)

b. Atratividade do Design (5,0 pontos)

i. Facilidade de navegação, interface amigável. (5,0 pontos)

II. Responsividade (25,0 pontos)

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

a. Adaptabilidade (25,0 pontos)

i. Funcionalidade e aparência adequada em diferentes dispositivos. (12,5 pontos)

ii. Adequação do *layout* e conteúdo para diferentes tamanhos de tela. (12,5 pontos)

FRONT-END DESIGN ENGINEERING

(08/12)

ENTREGA - 3º SPRINT

CRITÉRIOS DE AVALIAÇAO E PONTUAÇÃO (4/5)

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

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

(09/12)

ENTREGA - 3º SPRINT

CRITÉRIOS DE AVALIAÇAO E PONTUAÇÃO (5/5)

4. 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)

- Tecnologias
- Integrantes
- Imagens e ícones relacionadas ao projeto
- Estrutura de pastas do projeto

b. Link do GitHub (3,3 pontos)

c. Link do Vídeo do YouTube (3,3 pontos)

5. VÍDEO (5,0 pontos)

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. (5,0 pontos)

FRONT-END DESIGN ENGINEERING

ENTREGA - 3º SPRINT

PENALIDADES (1/2)

I. **Arquivos maiores que 50 MB (-50,0 pontos)**

O arquivo .ZIP do projeto não deve exceder 50 MB.

II. **Para cada página OBRIGATÓRIA que faltar no projeto (-20,0 pontos)**

- A ausência de qualquer página obrigatória — Index/Home (Página Inicial), Integrantes, Sobre/About, FAQ (Perguntas Frequentes) ou Contato — resultará na perda de 20 pontos por página faltante no projeto.

III. **Falta de página de identificação dos integrantes (-20,0 pontos)**

A ausência de uma página com a identificação dos integrantes, incluindo nome, RM e turma, será penalizada.

IV. **Para cada item de identificação da página dos integrantes que faltar (-30,0 pontos)**

A ausência de cada item da página dos integrantes abaixo:

- *Nome* (-10,0 pontos)
- *RM* (-10,0 pontos)
- *Turma* (-10,0 pontos)

FRONT-END DESIGN ENGINEERING

ENTREGA - 3º SPRINT

PENALIDADES (2/2)

V. 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.

VI. Não entrega do link do GITHUB no README.md (-50 pontos)

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

VII. README.md (-50 pontos)

A NÃO ENTREGA DO ARQUIVO README, OU A ENTREGA DE OUTRO TIPO DE ARQUIVO EM SEU LUGAR ACARRETARÁ UMA PENALIDADE DE -50 PONTOS AO GRUPO.

VIII. 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, está estritamente proibida neste projeto.

Caso o professor responsável pela disciplina identifique a utilização desses recursos, a nota da **SPRINT 03** da equipe será automaticamente ZERADA.

FRONT-END DESIGN ENGINEERING

ENTREGA - 3º SPRINT

ENTREGA

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 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!!

4º ENTREGAS

POR DISCIPLINA

SOFTWARE ENGINEERING AND BUSINESS MODEL

(1 de 2) Documento PDF único com identidade visual (fontes e cores) do grupo contendo:

- **Capa**
  - Nome da Startup
  - Nome completo e RM dos integrantes
- **Sumário**
- **Sumário Executivo**
- **[10 pontos] Sobre o Negócio**
  - Detalhamento da oportunidade identificada
  - Detalhamento do modelo de negócios da startup
  - Modelo de relacionamento que será aplicado (B2B, B2C,...)
  - Detalhamento do tipo de inovação da solução proposta
  - BMC atualizado
- **[20 pontos] Mercado e Competidores**
  - Mercado-alvo (Segmento de Clientes)
  - Análise de Concorrentes
  - Vantagem competitiva (diferenciais da solução proposta)

ENTREGA - 4º SPRINT

SOFTWARE ENGINEERING AND BUSINESS MODEL

(2 de 2)

- **[25 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
- **[25 pontos] Retorno do Investimento**
  - Métricas tangíveis utilizadas
  - Racional do cálculo do ROI
- **[20 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

ENTREGA - 4º SPRINT

ARTIFICIAL INTELLIGENCE & CHATBOT

- **Descrição:** você e seu grupo devem usar os dados do Sprint 3 para criar modelos de inteligência artificial.

- **Requisitos:**
  - Usando os dados levantados, crie 2 modelos de aprendizado de máquina. Os modelos podem ser de classificação, regressão e/ou agrupamento. **[+70 pontos]**
  - Pelo menos um modelo treinado deve ser disponibilizado em uma API REST para ser usado pela sua aplicação. **[+30 pontos]**

- **Entrega:**
  - Arquivo **.ipynb** com os modelos de aprendizado de máquina, discussão dos resultados em markdown e todo o código em Python implementado;
  - Arquivo do modelo treinado (várias extensões são possíveis como .joblib, .pickle, .h5,etc.);
  - Arquivo de configuração da API REST (por exemplo, se em Node-RED, o .json, se em Flask, .py).
  - Arquivo **.txt** com nome dos membros e explicações pertinentes.

- **Penalidades:**
  - Entrega fora do prazo. **[-100 pontos]**
  - Entrega de arquivos errados ou corrompidos. **[- pontuação do item do respectivo arquivo faltante]**

ENTREGA - 4º SPRINT

BUILDING RELATIONAL DATABASE

SCRIPT DDL E DML-CRIAÇÃO DA ESTRUTURA E DADOS PARA POPULARAS TABELAS PARA OS TESTES DA APLICAÇÃO

- Arquivo com as instruções DDL (scripts da 3a sprint corrigido) e DMLs referente a criação e carga de dados para testes.
- Cada tabela deve ser preenchida com no mínimo 10 linhas. As tabelas associativas, devem ser preenchidas com no mínimo 20 linhas.
- A massa de dados, deve ser composta por dados válidos, ou seja, não devem ser inseridos: xxxx,11111,teste, ou similar. Trabalhar com dados fictícios mas coerentes.

ENTREGA - 4º SPRINT

BUILDING RELATIONAL DATABASE

SCRIPT DQL- CONSULTAR AS TABELAS PARA OS TESTES DA APLICAÇÃO

- Implementar a criação de relatórios na aplicação, deverá ser implementado os seguintes relatórios:
- Relatório utilizando classificação de dados, a escolha da tabela é decisão do grupo.
- Relatório utilizando alguma função do tipo numérica simples.
- Relatório utilizando alguma função de grupo.
- Relatório utilizando sub consulta.
- Relatório utilizando junção de tabelas
- Pelo menos 2 relatórios de cada versão solicitada.

ENTREGA - 4º SPRINT

BUILDING RELATIONAL DATABASE

Entregas:

Arquivo pdf com a modelagem completa

Arquivo sql com a parte prática na sequência de execução dos comandos

Pontuação:

Modelagem completa – 10 pontos

Comandos DDL estrutura do banco de dados – 20 pontos

Comandos DML dados – 20 pontos

Relatório utilizando classificação de dados, a escolha da tabela é decisão do grupo – 10 pontos

Relatório utilizando alguma função do tipo numérica simples. – 10 pontos

Relatório utilizando alguma função de grupo. – 10 pontos

Relatório utilizando sub consulta. – 10 pontos

Relatório utilizando junção de tabelas – 10 pontos

ENTREGA - 4º SPRINT

BUILDING RELATIONAL DATABASE

Penalidades:

Arquivos fora do padrão, menos 10 pontos

Arquivo DDL com erros, sem avaliação

Arquivo DML com erros, sem avaliação

Cada relatório que não execute o solicitação, sem avaliação

ENTREGA - 4º SPRINT

COMPUTATIONAL THINKING USING PYTHON

- **Conectando ao Banco de dados**
- Tendo como referência o entregável da sprint 3, a equipe deverá realizar a integração com o banco de dados, desenvolvendo um CRUD em Python. Para tanto, os requisitos são os seguintes:
- [10 pontos] Estrutura de menus e submenus para acesso às opções do CRUD (inserir, alterar, excluir e consultar);
- [30 pontos] Realizar ao menos um exemplo de cada operação do CRUD e disponibilizar uma opção, para exportar os dados dessas consulta para um arquivo JSON.
- [20 pontos] Desenvolvimento e/ou consumo de uma API externa pública.
- [10 pontos] As informações colhidas e/ou alteradas pelo programa desenvolvido devem refletir no sistema web (front-end) e vice-versa.
- [30 pontos] Vídeo explicativo de até 7 minutos, contendo os nomes dos integrantes, a ideia, a solução proposta, explicação do código fonte e execução - o mesmo deve estar em conformidade com o front-end.

ENTREGA - 4º SPRINT

COMPUTATIONAL THINKING USING PYTHON

- Entrega:
  - Entrega em formato .zip do código fonte, arquivo contendo os nomes dos participantes e link para o vídeo (youtube)
- Penalidades:
  - [30 pontos] - Funcionamento do CRUD ou incompleto
  - [30 pontos] - Explicação coerente do código fonte (implementação)
  - [20 pontos] - Entrega fora dos padrões solicitados

ENTREGA - 3º SPRINT

FIAP

DOMAIN DRIVEN DESIGN USING JAVA (1/5)

ENTREGA - 4º SPRINT

✓ Descrição

- Esta é a entrega final do projeto Java, que visa avaliar o desenvolvimento completo da aplicação, tanto em nível técnico quanto documental. O projeto deve conter todos os elementos necessários para o funcionamento do sistema de forma integrada com o front-end, além de uma documentação que comprove a estrutura e funcionamento da aplicação.
- A entrega deve ser feita por meio de um **repositório no GitHub**, contendo:
- O **código-fonte Java** atualizado.
- A **documentação final em formato PDF**.

Requisitos

Documentação (em PDF) – [10 pontos]

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

- **Capa**
  - Nome dos integrantes.
  - Nome da equipe.
  - Nome da solução.

FIAP

DOMAIN DRIVEN DESIGN USING JAVA (2/5)

- **Objetivo e escopo do projeto**
  - Descrição objetiva e concisa da solução proposta.
- **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.).
- **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 Atualizado**
  - Representação das classes com atributos, métodos e relações (associação, herança etc.).

ENTREGA - 4º SPRINT

DOMAIN DRIVEN DESIGN USING JAVA (3/5)

💻 Projeto Java Finalizado (Código-Fonte)

O sistema deve estar completo, funcional e estruturado conforme boas práticas:

- **Camada Model – [10 pontos]**
  - Contém todas as classes modelo necessárias, representando corretamente o banco de dados.
- **Camada DAO e Service – [30 pontos]**
  - Contém todas as funcionalidades essenciais para o front-end.
  - O CRUD (Create, Read, Update, Delete) deve estar **funcionando completamente**.
  - Contém validações e regras de negócio adequadas.
- **API Restful – [30 pontos]**
  - Implementação de todos os endpoints necessários para funcionamento do front-end.
  - Deve seguir os princípios REST (recursos, verbos HTTP, status de resposta, etc.).
- **Boas Práticas – [20 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).
- **Observação:**
  Não é necessária uma classe de teste, desde que a API Restful esteja completamente implementada.

ENTREGA - 4º SPRINT

DOMAIN DRIVEN DESIGN USING JAVA (4/5)

🧮 **Pontuação Total: 100 pontos**

Item	Descrição	Pontuação
Documentação final	Incluindo tabela de endpoints e prints de telas	10 pontos
Camada Model	Classes conforme o MER, representando as entidades	10 pontos
Camada DAO e Service	CRUD funcional + integrações + validações	30 pontos
API Restful	Endpoints completos e funcionais para o front-end	30 pontos
Boas práticas	Organização, nomenclatura, tratamento de erros, padrões	20 pontos
Total		**100 pontos**

ENTREGA - 4º SPRINT

DOMAIN DRIVEN DESIGN USING JAVA (5/5)

ENTREGA - 4º SPRINT

⚠ Penalidades e Descontos

Ausência de itens obrigatórios na documentação	**Até -5 pontos por item**
Tabela de endpoints incompleta ou mal formatada	Até -10 pontos
Endpoints da API não funcionais	Até -30 pontos
Falta de integração com o front-end	Até -30 pontos
Ausência de tratamento de exceções	Até -10 pontos
Código desorganizado ou fora do padrão	Até -10 pontos
Entrega após o prazo (não serão aceitos)	Penalização total (-100 pontos)

FRONT-END DESIGN ENGINEERING

Chegou a hora de finalizarmos nosso projeto!

A equipe, nesta **Sprint 04**, 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.

ENTREGA - 4º SPRINT

FRONT-END DESIGN ENGINEERING

ENTREGA - 4º SPRINT

ALGUMAS REGRAS (1/2)

- **FRAMEWORK**
  - ✓ O projeto final deverá ser estruturado utilizando **REACT + VITE + TYPESCRIPT**, 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 na plataforma GITHUB ao professor responsável, 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 **total de commits do projeto é 15**.

FRONT-END DESIGN ENGINEERING

ALGUMAS REGRAS (2/2)

- **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*.

- **OBSERVAÇÕES**
  - ✓ A estilização deve ser realizada utilizando apenas **TAILWINDCSS**.
  - ✓ 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 contend a branch main e não apenas os arquivos do projetos.
  - ✓ Utilizar o **README.md** do projeto para apresentar as informações importantes de desenvolvimento do projeto até este, além das informações obrigatórias. Estilizar com **MARKDOWN** este arquivo para que ele seja o indicador e orientador do projeto no repositório.

ENTREGA - 4º SPRINT

FRONT-END DESIGN ENGINEERING

ENTREGA - 4º SPRINT

CRITÉRIOS DE AVALIAÇAO E PONTUAÇÃO (1/5)

1. CONSTRUÇÃO DO PROJETO DE ACORDO COM A SPRINT 03 (20,0 pontos)

Reestruturação da Sprint 03, para o *framework* **REACT + VITE + TYPESCRIPT**.

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

- a. **Criação de Rotas** (2,5 pontos)
  - i. Definição clara e funcional de rotas utilizando, considerando boas práticas e tipagem do código. (2,5 pontos)
- b. **Navegação** (2,5 pontos)
  - i. Navegação fluida e intuitiva entre as páginas, proporcionando uma experiência de usuário consistente. (2,5 pontos)
- c. **Redirecionamento** (2,5 pontos)
  - i. Utilização eficaz de redirecionamentos e *feedBacks* ao usuários com mensagens personalizadas. (2,5 pontos)
- d. **Parâmetros** (2,5 pontos)
  - i. Implementação correta de parâmetros de rota e tratamento de rotas dinâmicas. (2,5 pontos)

II. **Utilização de Tipos Específicos** (10 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

CRITÉRIOS DE AVALIAÇAO E PONTUAÇÃO (2/5)

2. ESTILIZAÇÃO E RESPONSIVIDADE COM TAILWIND (15,0 pontos)

I. **Responsividade** (15,0 pontos)

A responsividade devem ser representadas em diferentes tamanhos de tela: 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. (7,5 pontos)

ii. Adequação do *layout* e conteúdo para diferentes tamanhos de tela. (7,5 pontos)

ENTREGA - 4º SPRINT

FRONT-END DESIGN ENGINEERING

(06/12)

ENTREGA - 4º SPRINT

CRITÉRIOS DE AVALIAÇAO E PONTUAÇÃO (3/5)

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

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

a. Disponibilização de uma URL funcional para acesso ao projeto no arquivo ***README.md*** . (5,0 pontos)

b. Efetivação bem-sucedida do deploy na plataforma Vercel. (5,0 pontos)

4. INTEGRAÇÃO ADEQUADA DAS APIs (30,0 pontos)

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 (30,0 pontos)**

a. **Consumo da API** (30,0 pontos)

i. Consumo de API, utilizando request/response e as suas melhores práticas, manipulando os verbos HTTP, GET/POST/PUT/DELETE, CRUD. (10,0 pontos)

ii. Manipulação correta dos dados obtidos das requisições. (10,0 pontos)

iii. Tratamento de erros e respostas inesperadas. (10,0 pontos)

FRONT-END DESIGN ENGINEERING

CRITÉRIOS DE AVALIAÇAO E PONTUAÇÃO (4/5)

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

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.

ENTREGA - 4º SPRINT

FRONT-END DESIGN ENGINEERING

(08/12)

ENTREGA - 4º SPRINT

CRITÉRIOS DE AVALIAÇAO E PONTUAÇÃO (5/5)

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)

- Tecnologias
- Integrantes
- Imagens relacionadas ao projeto
- Ícones relacionados ao projeto
- Estrutura de pastas do projeto

b. Link do GitHub (3,3 pontos)

c. Link do Vídeo do YouTube (3,3 pontos)

7. Vídeo (5,0 pontos)

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. (5,0 pontos)

FRONT-END DESIGN ENGINEERING

FIAP

(09/12)

INTEGRAÇÃO

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

ENTREGA - 4º SPRINT

FRONT-END DESIGN ENGINEERING

ENTREGA - 4º SPRINT

PENALIDADES (1/2)

I. Arquivos maiores que 50 MB (-50,0 pontos)

O arquivo .ZIP do projeto não deve exceder 50 MB.

II. Para cada página OBRIGATÓRIA que faltar no projeto (-20,0 pontos)

A ausência de qualquer página obrigatória — Index/Home (Página Inicial), Integrantes, Sobre/About, FAQ (Perguntas Frequentes) ou Contato — resultará na perda de 20 pontos por página faltante no projeto.

III. Falta de página de identificação dos integrantes (-20,0 pontos)

A ausência de uma página com a identificação dos integrantes, incluindo nome, RM e turma, será penalizada.

IV. Para cada item de identificação da página dos integrantes que faltar (-30,0 pontos)

A ausência de cada item da página dos integrantes abaixo:

- *Nome* (-10,0 pontos)
- *RM* (-10,0 pontos)
- *Turma* (-10,0 pontos)

V. 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 - 4º SPRINT

PENALIDADES (2/2)

VI. Não entrega do link do GITHUB no README.md (-50 pontos)

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

VII. README.md (-50 pontos)

A NÃO ENTREGA DO ARQUIVO README, OU A ENTREGA DE OUTRO TIPO DE ARQUIVO EM SEU LUGAR ACARRETARÁ UMA PENALIDADE DE -50 PONTOS AO GRUPO.

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

A utilização de frameworks e/ou soluções prontas, como AXIOS, CARROUSEL, ACORDION incluindo o Bootstrap, está estritamente proibida neste projeto. Caso o professor responsável pela disciplina identifique a utilização desses recursos, a nota da **SPRINT 03** da equipe será automaticamente **ZERADA**.

IX. Não entrega da API (-50,0 pontos)

A não entrega da API ou das APIs, implicará em um desconto de 50,0 pontos da nota final.

X. 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 - 4º SPRINT

ENTREGA

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!!


FIAP