

CHALLENGE 2023

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - 1º ANO (Manhã)

CRONOGRAMA

FIAP

CRONOGRAMA - 2º SEMESTRE



DATA	EVENTO	STEAKHOLDER
25/08	Apresentação Banca de Professores	Professores
28 ou 29 ou 30/08 (Aguardando confirmação da Porto)	Mentoria Online Porto Seguro	Porto
12/09	ENTREGA DA SPRINT 3	Alunos
Semana 18/09	Feedback das entregas SPRINT 3	Professores
15/09	Apresentação Banca de Professores - Filtragem	Professores
26/09	Apresentação Banca Final Porto Seguro	Porto
28/10	NEXT	FIAP
13/11	ENTREGA DA SPRINT 4	Alunos
Semana 13/11	Feedback das entregas SPRINT 4	Professores
*As apresentações não são obrigatórias * Datas sujeitas a alterações, devido a agenda da empresa parceira		

- ▣ **A primeira apresentação para a banca de professores e a mentoria da Porto Seguro** serão somente para orientações e direcionamentos no desenvolvimento dos projetos.
- ▣ **A segunda apresentação para a banca de professores** será para avaliar os projetos aptos a realizarem a apresentação final para a **Porto Seguro**.
- ▣ **A apresentação final para a Porto Seguro** será para avaliar os melhores projetos que participarão do evento NEXT e serão premiados na competição.

As apresentações não são obrigatórias e deve ser um pitch de no máximo 5 minutos.

Nos dias das apresentações não haverá aula.

3º ENTREGAS

FIAP

Sprint #3: Base de Dados para IA

Você e sua equipe deverão levantar uma base de dados para o treinamento de uma IA.

Requisitos:

- Dados levantados aderentes ao tema do desafio, mínimo 100 exemplos; **[50 pontos]**
- Explicação dos dados (fonte ou como foram coletados, o que significam, quantidade e qualidade, rótulos, qual o objetivo que se pretende realizar com eles); **[50 pontos]**

Entregáveis:

- 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.

BUILDING RELATIONAL DATABASE

- Realizar a implementação prática da estrutura de banco dados utilizando a linguagem Sql (create table) plataforma Oracle.
- A partir do Diagrama de classes (BackEnd) que demonstra os objetos criados e usados para armazenamento de dados, estes objetos também devem aparecer na modelagem do projeto: DER ou MER, MLR e MF.
- Ferramenta a ser utilizada: Data Modeler e Sql Developer ou Live Sql.
- **Até 40 pontos**, desde que atenda o solicitado e dentro da metodologia aplicada

BUILDING RELATIONAL DATABASE

- Deverá ser implementada a estrutura usando os comandos DDL da linguagem Sql na plataforma Oracle, lembrando que as tabelas devem corresponder aos objetos criados em BackEnd.?
- **Até 50 pontos**, desde que objetos e tabelas sejam correspondentes?
- **Até 10 pontos**, entregar a seguinte documentação:
- A entrega zipada deverá conter dois arquivos:

1 - Template atualizado (já enviado na Sprint 1). Aqui o grupo deve fazer realizar uma revisão no desenho da modelagem, nos objetos de banco e dicionário de dados com base nos Feedbacks do Professor das Sprints 1 e 2

2 - Arquivo SQL contendo:

- **Identificação dos integrantes do time com RM e Nome Completo.**
- **A codificação SQL solicitada, na sequência de execução.**
- **Antes de execução utilizar comentário para descrever o que será processado.**
- **Iniciar a codificação usando comando drop table para apagar as possíveis tabelas já existentes no ambiente.**
- **O arquivo DEVE seguir a seguinte padronização de nomenclatura:**
- **Exemplo: 1TDSX_2023_Proj_BD.pdf e 1TDSX_2023_CodigoSql.sql**

•Onde:

•X = Turma

Obs: A boa organização do conteúdo do arquivo é de suma importância para apontamento da nota, caso haja falha nesta organização a correção pode vir a ser comprometida acarretando diminuição da nota.

COMPUTATIONAL THINKING USING PYTHON

SPRINT 3

- 1 – Considerando a Sprint 2 devidamente concluída, escolha e codifique ao menos 3 funcionalidades do menu, utilizando os conceitos aprendidos: Funções/Procedimentos, passagem de parâmetros e retorno de funções / Manipulação de estruturas de dados: Listas, Tuplas, Dicionários, Tabelas (listas + dicionários).
- 2 – Printar algumas telas (principais) dos projetos de Java e/ou JavaScript que contenham rotinas que envolvam lógica de programação.
- 3 – A **complexidade** da solução e **organização** do documento serão considerados na avaliação do projeto.
- 4 – **Interdisciplinaridade**: Software Experience & Design (modelagem da solução)

Entrega:

Compactar em um arquivo (extensão .zip):

- Documento texto contendo uma capa com os nomes dos integrantes, nome da equipe e da solução proposta, um sumário, um descritivo explicando a solução proposta com os objetivos e escopo do projeto, além dos prints de código contendo Lógica de Programação e Estruturas implementadas em Java/JavaScript **(20 pontos)**;
- Arquivo .py com o projeto em Python **(80 pontos)**;

Postar o arquivo .zip no Portal da FIAP.

DOMAIN DRIVEN DESIGN



Documento PDF (15 pts) - Os itens abaixo devem constar na documentação

- **Capa** com o nome dos integrantes, nome da equipe e nome da solução.
- **Sumário**
- **Objetivo e escopo do projeto:** descrever a solução proposta pelo grupo de forma objetiva.
- **Breve descrição** das principais funcionalidades da solução.
- **Protótipo** - apresentar as telas do sistema explicando como interagir (pode ser desenvolvida em qualquer ferramenta).
- **Modelo do banco de dados.**
- **Diagrama de classes** atualizada.
- **Procedimentos para rodar a aplicação:**

DOMAIN DRIVEN DESIGN

Projeto Java (85 pts)

- **(25) Camada Model**, com os devidos padrões seguidos em sala de aula.
- **(20) Implementação de (no mínimo) quatro métodos** de relevância, contendo lógica e regras de negócio definidos para o projeto (O grau de dificuldade será considerado na correção deste item)
- **(10)** Implementação de uma **classe de teste** que instancie objetos e teste os métodos implementados no tópico 2.
- **(10)** Camada com a classe de conexão, o usuário e senha tem que estar no código.
- **(20)** Implementação da camada e classe DAO com pelo menos um método insert, funcionando.

Entrega: Projeto Java zipado e entregue no portal do aluno no menu do challenge.

RESPONSIVE WEB DEVELOPMENT

Para esta etapa, vamos criar as telas, nas resoluções desenvolvidas na prototipação da 2ª entrega, seguindo as regras:

- Construir um projeto em REACT nos mesmos moldes que a entrega realizada no (FIGMA).
- Esta nova etapa faremos o projeto em **React** para componentizar a aplicação, utilizando todas as boas práticas aprendidas em aula. Estruture o projeto entregue no primeiro semestre no (FIGMA), de forma que a aplicação se torne single-page.
- O projeto ainda não precisa consumir uma API, mas caso possuir elementos de input de dados do usuário (formulários de cadastro etc.), deve ser apresentado se assim a documentação e a entrega anterior descrever esses elementos.

RESPONSIVE WEB DEVELOPMENT

(continuação)

- O projeto deve ser criado dentro das normas do W3C.
- Não será permitida a utilização de *templates* ou exercícios dados em aula, caso isso aconteça, a equipe terá sua nota automaticamente ZERADA.
- Deverá ser utilizado styled-components ou CSS3 puro.
- A estrutura deve ser semântica e responsiva de acordo com o estipulado na entrega anterior.
- Uma das páginas do projeto deve conter o nome e RM dos integrantes da equipe e o link do repositório.
- O projeto deve ser versionado com *git* e *github* e ter no mínimo 10 *commits* e a participação de todos os integrantes do grupo e estar em um repositório público. (**obrigatório**)

Entrega:

- A equipe deverá entregar uma cópia do repositório do projeto criado compactada em formato .ZIP, **(não esqueça de utilizar .gitignore, principalmente para a pasta node-modules, pois esta não deve ser versionada).** **A não observância desta regra, vai elevar o tamanho do projeto, fazendo com que ultrapasse os 50MB que é o limite do portal da FIAP.**
- O aluno responsável deverá revisar o documento antes do envio para evitar falhas e equívocos na entrega, se necessário teste em mais de uma máquina.
- Somente um aluno do grupo deverá realizar a entrega, caso mais de um aluno realize a entrega, será descontado do grupo 1 ponto por entrega adicional.
- Os alunos do grupo que forem de outras turmas devem estar devidamente identificados com seus RM e turma.

RESPONSIVE WEB DEVELOPMENT



Pontuação:

- Construção do projeto de acordo com a 2ª entrega (20 pontos);
- Componentização das páginas e itens reaproveitáveis (30 pontos);
- Responsividade das páginas: Desktop, Tablet e Mobile. (20 pontos);
- Utilização do Git/Github, projeto com no mínimo 10 commits e participação de todos os integrantes (20 pontos).
- Entrega dentro dos padrões solicitados no documento. (10 pontos).

Documentação funcional do projeto:

1 - Entregar documento em arquivo Word único contendo **(20 pontos)**:

- Capa;
- Sumário (com tópicos das páginas, como documento está organizado);
- Descritivo com explicação do projeto, com justificativa/objetivos de no mínimo 20 linhas para esse contexto, permitindo a validação da entrega e compreensão do projeto;
- Explicar como o seu sistema que estará pronto até a Sprint Quatro, conseguirá atender o challenge da Porto, com no mínimo 10 linhas.

2 - Diagramas de caso de uso que está explicado nos próximos slides **(30 pontos)**.

3 - Diagramas de atividades de uso que está explicado nos próximos slides **(30 pontos)**.

4 - Backlog do Produto e Planejamento das Sprints **(20 pontos)**:

- Essa entrega você terá que entregar a versão final do product backlog priorizado e estimado, já dividido por Sprint e mostrando os respectivos status de cada tarefa, sendo esses: Pendente, Em andamento, Concluído ou Cancelado.

Elaborar os Diagramas completos de Casos de Uso que explicam as funcionalidades do sistema:

Represente nos seus diagramas (Mínimo de dois diagramas de caso de uso):

- Casos de uso;
- Atores;
- Associação de Atores com generalização/especialização, conforme o caso;
- Associação entre Casos de Uso com Include, Extend ou Generalização, conforme o caso.

Elabore os documentos descritivos dos Casos de Uso com detalhes de:

- Identificação do diagrama de caso de uso detalhado e seu objetivo;
- Pré e pós condição de execução;
- Passos da execução do Caso de Uso, com detalhes de qual a operação feita, com quais dados.

Elaborar os Diagramas completos de Atividades:

Terá que mostrar o fluxo de atividades completo do seu sistema, caso o professor avalie que não fez completo, irá verificar o percentual do que refletiu de atividades do sistema, então se por exemplo entregar 40% das atividades do sistema refletido nos diagramas de atividades, então ao invés de 30 pontos para esse tópico, irá receber 12 pontos.

Mínimo de dois diagramas de atividades, mas cuidado, a quantidade é você que irá determinar com base no que fizer que tem que refletir as atividades do sistema.

SOFTWARE DESIGN & TX



Os Diagramas e Documentos devem ser produzidos utilizando o ASTAH.

Após construídos, fazer uma imagem/print dos diagramas e incluir no documento WORD.

Crie um arquivo .ZIP com o .PDF gerado, fazendo a entrega do .ZIP no portal do aluno, na área de Challenge Sprints.

Tem que enviar apenas um arquivo com tudo que foi solicitado, caso envie mais de um arquivo, será descontado 10 pontos, pois a organização e gestão do que deve entregar é o foco dessa matéria.

4º ENTREGAS

FIAP

Sprint #4: Modelos de IA

Nesta etapa, você e seu grupo usarão os dados do Sprint 3 para criar modelos de inteligência artificial.

Requisitos:

- Usando os dados levantados, crie um modelo de aprendizado de máquina. O modelo pode ser de classificação, regressão e/ou agrupamento. **[70 pontos]**
- O modelo treinado deve ser disponibilizado em uma API REST para ser consumido pela sua aplicação. **[30 pontos]**

Sprint #4: Modelos de IA

Nesta etapa, você e seu grupo usarão os dados do Sprint 3 para criar modelos de inteligência artificial.

Entregáveis:

- 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).

BUILDING RELATIONAL DATABASE

Utilizar a estrutura criada e atualizada da versão anterior para implementação de dados, ou seja, utilizar os comandos DML na estrutura, o grupo deverá alimentar cada tabela existente com 7 linhas de dados, exemplificar o uso de comandos de atualização e eliminação de dados, aplicar 3 testes de atualização e eliminação de dados. [50 pontos]

BUILDING RELATIONAL DATABASE

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. [5 pontos]?

Relatório utilizando alguma função do tipo numérica simples. [5 pontos]?

Relatório utilizando alguma função de grupo. [10 pontos]?

Relatório utilizando sub consulta. [15 pontos]?

Relatório utilizando junção de tabelas [15 pontos]?

Pelo menos 1 relatório de cada versão solicitada.

BUILDING RELATIONAL DATABASE

A entrega zipada deverá conter dois arquivos:

1 - Template atualizado (já enviado na Sprint 1). Aqui o grupo deve fazer realizar uma revisão no desenho da modelagem, nos objetos de banco e dicionário de dados com base nos Feedbacks do Professor das Sprints 1 e 2

2 - Arquivo SQL contendo:

Identificação dos integrantes do time com RM e Nome Completo.

A codificação de criação da estrutura atualizada/corrigida.

A codificação SQL solicitada, na sequência de execução.

Antes de execução utilizar comentário para descrever o que será processado.

Iniciar a codificação usando comando drop table para apagar as possíveis tabelas já existentes no ambiente.

BUILDING RELATIONAL DATABASE

O arquivo DEVE seguir a seguinte padronização de nomenclatura:

Exemplo: 1TDSX_2023_Proj_BD.pdf e 1TDSX_2023_CodigoSql.sql

Onde:

X = Turma

Obs: A boa organização do conteúdo do arquivo é de suma importância para apontamento da nota, caso haja falha nesta organização a correção pode vir a ser comprometida acarretando diminuição da nota.

COMPUTATIONAL THINKING USING PYTHON

SPRINT 4

- 1 – Considerando a Sprint 3 devidamente concluída, codifique as funcionalidades do menu em Python utilizando os conceitos vistos até o final do semestre, como arquivos texto, JSON, consumo de API, Conexão com BD Oracle e tratamento de excessões.
- 2 – Fazer um Pitch-vídeo (até 2 minutos) apresentando a aplicação.
- 3 – Printar algumas telas (principais) dos projetos de Java e/ou JavaScript que contenham rotinas que envolvam Conceitos de lógica de programação e manipulação de dados em estruturas de dados.
- 4 – A **complexidade** da solução e **organização** do documento serão considerados na avaliação do projeto.
- 5 – **Interdisciplinaridade**: Banco de Dados (persistência) e Software Experience & Design (modelagem da solução)

Entregas:

Compactar em um arquivo (extensão .zip):

- Documento texto contendo uma capa com os nomes dos integrantes, nome da equipe da solução proposta, um sumário, um descritivo explicando a solução proposta com os objetivos e escopo do projeto, além dos prints de código contendo Lógica de Programação e Estruturas implementadas em Java/JavaScript, e o link do video Pitch **(30 pontos)**;
- Implementação em Python (arquivos .py) **(70 pontos)**

RESPONSIVE WEB DEVELOPMENT



Chegou a hora de finalizarmos o nosso projeto.

- Nossa implementação deve ser finalizada com a integração de uma API em nosso projeto.
- Aqui deverá acontecer a integração de todos os itens do projeto, ou seja, o *frontend* deve estar em alinhamento com o *backend*.
- A API criada na disciplina de **Domain Drive Design** deverá retornar todos os dados colhidos no *backend* ao *frontend* e/ou enviar os dados do *frontend* ao *backend*, para possível armazenamento no banco de dados, disciplina **Building Relational Database**.
- As informações colhidas e/ou alteradas pelo sistema desenvolvido na disciplina de **Computational Thinking Using Phyton** devem refletir no sistema *web* e vice-versa.

RESPONSIVE WEB DEVELOPMENT



(continuação)

- O projeto final deve estar utilizando o framework NextJS para roteamento de páginas e demais ajustes.
- Deve ser feito o deploy do projeto na plataforma Vercel, onde deverá ser disponibilizada uma URL de acesso.

RESPONSIVE WEB DEVELOPMENT

Entrega:

- A equipe deverá entregar uma cópia do repositório do projeto criado compactada em formato .ZIP, **(não esqueça de utilizar .gitignore, principalmente para a pasta node-modules, pois esta não deve ser versionada)**. **A não observância desta regra, vai elevar o tamanho do projeto, fazendo com que ultrapasse os 50MB que é o limite do portal da FIAP.**
- No repositório do projeto, no arquivo README.MD deve ser descrito qualquer detalhe de acesso ao sistema, como usuários e senha para testes.
- O aluno responsável deverá revisar o documento antes do envio para evitar falhas e equívocos na entrega, se necessário teste em mais de uma máquina.
- Somente um aluno do grupo deverá realizar a entrega, caso mais de um aluno realize a entrega, será descontado do grupo 1 ponto por entrega adicional.
- Os alunos do grupo que forem de outras turmas devem estar devidamente identificados com seus RM e turma.

RESPONSIVE WEB DEVELOPMENT

Pontuação:

- Construção do projeto de acordo com a 3ª entrega **(20 pontos)**;
- Criação de rotas e navegação com NEXT.JS **(20 pontos)**;
- Consumo das APIs. **(25 pontos)**;
- Utilização do Git/Github, projeto com no mínimo (novos) 10 commits e participação de todos os integrantes **(10 pontos)**;
- Deploy do *projeto* para a plataforma Vercel **(15 pontos)**;
- Entrega dentro dos padrões solicitados no documento. **(10 pontos)**.

DOMAIN DRIVEN DESIGN



Documento PDF Atualizado (10 pts)

- **Capa:** com o nome dos integrantes, nome da equipe e nome da solução.
- **Objetivo e escopo do projeto:** descrever a solução proposta pelo grupo de forma objetiva.
- **Breve descrição:** das principais funcionalidades da solução.
- **Tabela dos endpoints (API Restful):** contendo as URIs, verbo HTTP e Códigos de status de respostas para a requisição.
- **Protótipo** - prints das telas implementadas.
- **Modelo do banco de dados.**
- **Diagrama de classes** atualizada.
- Procedimentos para rodar a aplicação.

DOMAIN DRIVEN DESIGN



Projeto Java Finalizado

- **(10 pts) Camada Model** com todas as classes necessárias, de acordo com o banco de dados.
- **(30 pts) Camada Dao e Service** (validações), com todas as funcionalidades necessárias para o front-end do projeto.
- **(30 pts) API Restful** com todos os *endpoints* necessários para o funcionamento do *front-end* do projeto.

(20 pts) – Seguir todas as boas práticas trabalhadas em sala de aula, como regras de nomenclatura, tratamento de exceções, padrões de projetos etc.

Obs. Não é necessária uma classe de teste, desde que o projeto implemente API Restful.

Entrega: Link do repositório do GitHub com o Código Java e Documento PDF.

Documento descritivo do projeto em arquivo Word único contendo:

Estrutura do documento **(10 pontos)**:

- Capa com nome dos integrantes e nome do projeto
- Índice
- Explicação do projeto com no mínimo três folhas escritas com a letra tamanho 12 e o tipo de font Arial ou Times New Roman.
- **Print das telas prontas do projeto e caso ainda não esteja pronto alguma tela colocar os protótipos que podem, por exemplo estar no formato foto do caderno ou qualquer outra ferramenta, pois essa Sprint é a última, então caso mais de 10% do projeto não esteja pronto poderá ser descontado até 30 pontos, mesmo que atendam os requisitos abaixo.**

SOFTWARE DESIGN & TX

Versão final do Diagrama de Caso de Uso **(25 pontos)**:

- Nível de implementação e que reflita o projeto do seu grupo para o challenge, tem que ter no mínimo dois diagramas de caso de uso.

FIAP

Diagrama de Atividades **(20 pontos)**:

- Nível de implementação e que reflita o projeto do seu grupo para o challenge, tem que ter no mínimo dois diagramas de atividades que represente a solução completa, então fique à vontade para criar a quantidade necessária para representar as atividades do sistema, se o professor verificar que não entregou o suficiente para representar todo o sistema, terá desconto de pontos.

Diagrama de Sequência **(20 pontos)**:

- Nível de implementação e que reflita o projeto do seu grupo para o challenge, tem que ter no mínimo dois Diagramas de sequência que represente a solução completa, então fique à vontade para criar a quantidade necessária para representar o sistema, se o professor verificar que não entregou o suficiente para representar todo o sistema, terá desconto de pontos.

Obs.: Os diagramas incluídos no descritivo é obrigatório que sejam gerados com a utilização da plataforma Astah.

- A organização e o formato para a entrega deve ser produzida e entregue no formato da versão final do documento do projeto.

SOFTWARE DESIGN & TX

Backlog do Produto e Planejamento das Sprints **(10 pontos)**:

- Essa entrega você terá que entregar a versão final do product backlog priorizado e estimado, já dividido por Sprint e mostrando os respectivos status de cada tarefa, sendo esses: Pendente, Em andamento, Concluído ou Cancelado.

O vídeo deve ser publicado do Youtube **(15 pontos)**:

- Com permissão de acesso para que os professores possam visualizar e fazer suas avaliações. Um link de acesso deve ser gerado e entregue em um arquivo .TXT que contenha também a lista de RM e nomes dos integrantes do grupo. Se não postar no youtube, pode utilizar outra plataforma, porém se o professor não conseguir acessar no momento da avaliação, então não receberá essa pontuação.
- Nesse vídeo você terá que mostrar a solução funcionando, no formato de demonstração da entrega final de um sistema, com tempo mínimo de 5 minutos e máximo de 8 minutos.

Os Diagramas e Documentos devem ser produzidos utilizando o ASTAH.

Após construídos, fazer uma imagem/print dos diagramas e incluir no documento WORD.

Crie um arquivo .ZIP com o .PDF gerado, fazendo a entrega do .ZIP no portal do aluno, na área de Challenge Sprints.

Tem que enviar apenas um arquivo com tudo que foi solicitado, caso envie mais de um arquivo, será descontado 15 pontos, pois a organização e gestão do que deve entregar é o foco dessa matéria.

DÚVIDAS?!

OBRIGADO!!

