

# AI & CHATBOT (1/1)

Baseado no rascunho inicial e nos feedbacks recebidos da entrega da Sprint 1, chegou a hora de implementar o seu chatbot.

## Requisitos:

- 5 intenções, cada uma com pelo menos 12 exemplos; **[20 pontos]**
- 5 entidades, com possíveis sinônimos; **[15 pontos]**
- Lembrar do que foi dito em algum fluxo: uso de variável de contexto; **[10 ponto]**
- Integração via Webchat (somente texto); **[10 ponto]**
- Integração com o Telegram por texto e áudio usando STT e TTS., **[20 pontos]**
- Vídeo demonstrando a utilização real do chatbot, tanto no Webchat quanto no Telegram (capturando a tela e explicando em áudio a interação e o que de fato sua solução realiza). O vídeo deve estar em um link privado no YouTube; **[25 pontos]**

## Entregável:

- Arquivo do Dialog Skill **.json**;
- Fluxo de integração do Telegram em NODE-RED, arquivo **.json**;
- Arquivo **.txt** com o nome dos membros do grupo, as credencias dos serviços usados e o link do vídeo no YouTube.

# BUILDING RELATIONAL DATABASE (1/2)

Baseado na sua 1ª Entrega (ou Sprint 1) e nos feedback recebidos, continue a implementação do seu Projeto de Banco de Dados Relacional, levando em consideração

- **REQUISITOS**

- O descritivo do seu projeto devidamente atualizado e alinhado ao objetivo da solução definida pelo o grupo
  - Também é necessário indicar quais os dados/atributos necessários (ou importantes) para persistência E Mencionar as Regras de Negócio apuradas e Relacionadas as Entidades. **[14 pontos]**
  - Entregar o Diagrama de Classes da UML desenvolvido na disciplina Domain-Driven Design (DDD). **[14 pontos]**
- Migre o seu Diagrama de Entidade e Relacionamento (DER) ou Diagrama Logical para a Modelagem Lógica (utilize a "Notação de Engenharia - Informações" no Oracle Data Modeler) devidamente atualizado com a:
  - Definição da Chave Primária para cada Entidade; **[18 pontos]**
  - Definição dos Relacionamentos entre as Entidades; **[18 pontos]**
  - Definição da Chave Estrangeira para cada Entidade Relacionada; **[18 pontos]**
  - Definição dos Tipos dos Atributos **[18 pontos]**

# BUILDING RELATIONAL DATABASE (2/2)

- **ENTREGÁVEIS**

- Arquivo **.pdf** sobre o descritivo do seu projeto
  - Arquivo **.pdf** ou **.png** dos seus Diagramas (Modelagem Lógica e Classes)
- Arquivo **.txt** com o Nome e RM dos membros do grupo
  - Sugestão para nomeação dos arquivos:
    - 1TDSX\_2024\_Nome\_Grupo
    - X= turma
    - Nome\_grupo= Caso não tenha, pode deixar vazio

**Observação:** A boa organização do conteúdo dos arquivos 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 (1/2)

## ■ Código Fonte (Total 70 pontos)

- **(30 pontos)** Implementação/atualização do menu de opções (implementado na Sprint 1), com as principais funcionalidades oferecidas pelo sistema, ao menos um MVP dos itens sugeridos. O programa deve permitir ao usuário escolher uma funcionalidade e fazer a chamada da função correspondente, e após a sua execução, o programa deve retornar para o menu principal.
- **(10 pontos)** O programa deve realizar validações nas entradas de dados do usuário e na manipulação do menu.
- Será avaliada a aplicação correta dos conceitos de:
  - **(5 pontos)** Estruturas condicionais e de repetição.
  - **(10 pontos)** Funções com passagem de parâmetro e retorno.
  - **(15 pontos)** Armazenamento e manipulação de dados em Listas

# COMPUTATIONAL THINKING USING PYTHON (2/2)

## ▣ Documentação Atualizada (Total 25 pontos):

- **(10 pontos)** A partir do formato da documentação proposta na Sprint 1, atualizar o Sumário, o Descritivo do Projeto (incluindo as funcionalidades implementadas) e os Anexos que contém o código-fonte e/ou fluxogramas.
- **(15 pontos)** Descrever textualmente as mudanças da Sprint 1 para a Sprint 2, ou seja, o que foi acrescentado, alterado ou retirado do projeto.

## ▣ Formato de Entrega (Total 5 pontos):

- Arquivo compactado (**ZIP ou RAR**) contendo:
  - Arquivo **.pdf** com a documentação.
  - Arquivo **.py** com o código fonte do programa.

# DOMAIN DRIVEN DESIGN (1/2)

## ▣ Documentação atualizada (20 pontos):

- **(2,5 pontos) Capa** contendo o nome da solução e o nome dos integrantes
- **(2,5 pontos) Sumário** (como o documento está organizado)
- **(5 pontos) Descritivo** explicando o projeto, contendo uma justificativa/objetivos de no mínimo quinze linhas para compreensão do contexto, permitindo a validação da entrega (se as classes refletem o contexto do projeto).
- **(10 pontos) Modelagem das classes do projeto** (mínimo de 10 classes)

# DOMAIN DRIVEN DESIGN (2/2)

## ▣ Projeto Java (80 pontos):

- (20 pontos) Criação do projeto Java com todas as classes, atributos e em seus devidos pacotes, que deve estar de acordo com o Diagrama de Classe e o projeto proposto.
- (20 pontos) Conter métodos construtores, Getters e Setters.
- (10 pontos) Conter no mínimo dois métodos diferentes de funcionalidade do sistema.
- (30 pontos) Conter uma classe para executar o programa, instanciar objetos fazendo teste com todas as classes e todas as saídas do projeto.

**Entrega:** Arquivo .pdf para a documentação e um arquivo .zip com a pasta do projeto.

A entrega deverá ser feita exclusivamente pelo portal, em formato ZIP

**Não serão aceitas entregas pelo github.**

## FRONT-END DESIGN ENGINEERING (1/8)



- Para este desafio, vamos implementar o que foi prototipado na primeira sprint.
- O objetivo é garantir uma experiência de usuário coesa e responsiva, adaptada para diferentes dispositivos.
- Garantindo a qualidade e seguindo os padrões de desenvolvimento técnicos, por exemplo utilizando o Github como ferramenta de versionamento.



# FRONT-END DESIGN ENGINEERING (2/8)

- **Estrutura básica HTML (10 pontos)**

- Criação de Arquivos (5 pontos):

- Desenvolver arquivos HTML separados para cada página do protótipo, dentro dos padrões e convenções de nomenclatura já apresentado.

- Estrutura Essencial (5 pontos):

- Implementar a estrutura básica HTML com tags <html>, <head>, e <body> para cada página.
    - Incluir metatags essenciais, como aquelas relacionadas à viewport e outros requisitos específicos.

# FRONT-END DESIGN ENGINEERING (3/8)

- **Semântica e estruturação (20 pontos)**

- Tags Semânticas (10 pontos):

- Utilizar tags semânticas HTML5 para estruturar o conteúdo, como <nav>, <article>, <section>, <header>, e <footer>.

- Hierarquia Clara (5 pontos):

- Garantir uma hierarquia de elementos HTML clara e lógica, refletindo a estrutura do wireframe.

- Atributos Semânticos (5 pontos):

- Utilizar atributos semânticos, como role e aria-\*, para aprimorar a acessibilidade.

# FRONT-END DESIGN ENGINEERING (4/8)

- **Estilo Responsivo com CSS (25 pontos)**

- Implementação de CSS (15 pontos):

- Desenvolver estilos CSS para cada página, mantendo a consistência com o wireframe inicial.

- Media Queries para Desktop (5 pontos):

- Integrar Media Queries para garantir uma apresentação otimizada em telas de desktop.

- Media Queries para Tablets (5 pontos):

- Adaptação do design para telas de tablets usando Media Queries.

# FRONT-END DESIGN ENGINEERING (5/8)

- **Estilo Responsivo para dispositivos móveis (20 pontos)**

- Media Queries para Dispositivos Móveis (5 pontos):
  - Implementar Media Queries para garantir uma experiência de usuário apropriada em dispositivos móveis.
- Aprimoramento Visual para Móveis (10 pontos):
  - Ajustar o estilo para otimizar a apresentação em dispositivos móveis.
- Navegação Intuitiva para Móveis (5 pontos):
  - Garantir uma navegação intuitiva e fácil de usar em dispositivos móveis.

## FRONT-END DESIGN ENGINEERING (6/8)

- **Incorporação de elementos de formulários (10 pontos)**

- Incorporação de Elementos de Input (5 pontos):

- Integração de elementos de input conforme documentação e entregas anteriores.

- Estilo dos Elementos de Input (5 pontos):

- Aplicação de estilos apropriados aos elementos de input para uma apresentação coesa.

# FRONT-END DESIGN ENGINEERING (7/8)

- **Versionamento do projeto (15 pontos)**

- Criar um repositório privado para o projeto **e no dia da entrega dar permissão ao professor** no repositório da disciplina através de seu username do github. (5 pontos):
- Mínimo de 10 commits (5 pontos):

**ATENÇÃO**

- A participação de todos os integrantes do grupo é obrigatória. Caso um ou mais integrantes não apareçam nas métricas do repositório, estes não receberam a pontuação de versionamento.
- **O link do repositório deve ser inserido na página de integrantes devidamente sinalizada.**

- **Entrega do projeto**

- O grupo é encarregado de compactar e submeter o **repositório** do projeto no formato .ZIP, garantindo que o tamanho do arquivo não ultrapasse 50 MB, conforme o limite estabelecido pelo portal da FIAP. O não cumprimento dessa norma resultará em uma penalidade significativa, causando a **perda de cinquenta pontos** para todos os membros do grupo. Certifique-se de ajustar o tamanho do arquivo para atender a esta exigência e evitar quaisquer repercussões negativas.
- O líder eleito pela equipe é responsável por revisar o documento antes da entrega, garantindo a ausência de falhas ou equívocos. Recomenda-se realizar testes em múltiplas máquinas, se necessário.
- A entrega deve ser feita por apenas um aluno do grupo. Caso ocorra a **entrega por mais de um aluno, será descontado UM ponto do grupo para cada entrega adicional.**
- Alunos pertencentes a outras turmas devem ser claramente identificados com seus respectivos RM e turma em uma das páginas do projeto (página de integrantes).

# SOFTWARE DESIGN & TX (1/4)

Estamos elaborando a documentação do seu projeto, portanto, vamos agregar todos os entregáveis abaixo no documento anterior. O grupo deverá complementar o documento entregue na Sprint anterior - PDF os seguintes itens:

## 1. CONTEXTO DO PROBLEMA:

- **Business Model Canvas** da sua solução para a Porto Seguro - (10 pontos);
- **Mapa de Empatia** do público-alvo identificado na solução do grupo (10 pontos).



# SOFTWARE DESIGN & TX (2/4)

## 2. CONTEXTO DA SOLUÇÃO:

- Solução: A partir do feedback recebido pelos professores na Sprint 1 e do resultado dos itens anteriores:
  - **Matriz CSD** - Inclua no documento (PDF anterior) uma Matriz CSD para visibilidade do entendimento atual do grupo (10 pontos);
  - **Sua solução:** Refine e descreva com mais detalhes a solução do grupo para resolver o desafio através das tecnologias que serão vistas durante o curso. A solução proposta será considerada para validar as histórias do Backlog da Sprint 2 – Deve conter as correções solicitadas da Sprint anterior. (10 pontos);
- **Product Backlog da Sprint 2:** Mantenha o link do (Trello/Jira) com acesso liberado para o backlog da Sprint 2. As histórias neste momento precisarão conter os requisitos funcionais, não funcionais e regras de negócio especificados - (20 pontos);

## 2. CONTEXTO DA SOLUÇÃO:

**Diagrama de Caso de Uso** - Elaborar os Diagramas completos de Casos de Uso que explicam as funcionalidades do sistema:

Represente as funcionalidades da sua solução nos seus diagramas no ASTAH (Mínimo de três diagramas de caso de uso)

✓ Utilize todas as ferramentas que foram ensinadas em sala, os documentos descritivos devem acompanhar cada caso de uso. ✓ Não será aceito uso de outras ferramentas. ✓ Não será aceito diagrama sem os documentos descritivos.

✓ O uso correto dos conceitos deve ser praticado, caso contrário não será aceito (20 pontos);

**Pitch:** Inclua no documento um link com acesso liberado para a gravação em vídeo do PITCH da solução atual. O pitch deverá possuir no máximo 3 minutos, ✓ O Uso de IA para geração automática do Pitch não será aceito. ✓ O pitch deve ser produzido com dedicação. ✓ Vídeo com apenas os alunos/aluno explicando verbalmente não é permitido. ✓ Deve ter imagens da solução (reais ou protótipo). ✓ O Link deve ser enviado no arquivo no PDF. ✓ Caso o link esteja corrompido, seja zerado a nota.

## SOFTWARE DESIGN & TX (4/4)

### 2. CONTEXTO DA SOLUÇÃO:

- O grupo deverá gerar um vídeo pitch que sobre solução de uma forma criativa e interessante, não deverá passar os 3 minutos, devem considerar todo o contexto abordado em aula.
- ✓ O Uso de IA para geração automática do Pitch não será aceito.
- ✓ O pitch deve ser produzido com dedicação.
- ✓ Vídeo com apenas os alunos/aluno explicando verbalmente não é permitido.
- ✓ Deve ter imagens da solução (reais ou protótipo).
- ✓ O Link deve ser enviado no arquivo no PDF.
- ✓ Caso o link esteja corrompido, seja zerado a nota. (20 pontos).