



# CHALLENGE 2023

◦ ANÁLISE E  
DESENVOLVIMENTO DE  
SISTEMAS - 1º ANO



# O DESAFIO

- “Criar um novo portal da Salesforce com acessibilidade e que consiga capturar mais clientes”



The screenshot shows the Salesforce website homepage. At the top, there is a navigation bar with the Salesforce logo, links for 'Produtos', 'Indústrias', 'Aprendizado', 'Suporte', and 'Empresa'. On the right side of the navigation bar, there is a search icon, a globe icon, a 'Login' link, and a 'Teste grátis' button. Below the navigation bar, there is a blue banner with the text 'Acompanhe a magia do Dreamforce, ao vivo no Salesforce+, de 12 a 14 de setembro. Registre-se agora'. The main content area features a large headline 'Entregue sucesso agora com o Salesforce Customer 360.' followed by a sub-headline 'O Customer 360 é a nossa suíte de produtos e serviços que ajuda 98% dos clientes a alcançar ou superar suas metas de ROI.' To the right of the text, there is a graphic with a woman smiling and a small robot character. The graphic includes the text '98% of customers meet or exceed their ROI goals'. At the bottom of the main content area, there are two buttons: 'Faça o teste grátis' and 'Saiba mais'. In the bottom right corner, there is a 'CONTATO' button.

salesforce

Produtos Indústrias Aprendizado Suporte Empresa

Entre em contato  
0800 91 1887

Login

Teste grátis

Acompanhe a magia do Dreamforce, ao vivo no Salesforce+, de 12 a 14 de setembro. Registre-se agora

Entregue sucesso agora com o Salesforce Customer 360.

O Customer 360 é a nossa suíte de produtos e serviços que ajuda 98% dos clientes a alcançar ou superar suas metas de ROI.

98% of customers meet or exceed their ROI goals

Faça o teste grátis

Saiba mais

CONTATO

# DESAFIO

As turmas de agosto do curso de Análise e Desenvolvimento de Sistemas terão como desafio proposto pela empresa parceira Salesforce: fazer um novo portal da salesforce, com base em acessibilidade e que consiga atrair ainda mais clientes, os grupos poderão explorar propor um novo site.

O Desenvolvimento para o tema é livre, cabe ao grupo criar um entregável que melhor represente o tema, como por exemplo: deixar o site mais intuitivo, educativo, vinculado aos temas de serviço da empresa, Campanha de Marketing, e até Jornal Digital, esses são apenas alguns exemplos que podem inspirar os grupos.

## Dica

Pense fora da caixa!

Seja diferente!

Encante seu cliente!

# OBJETIVO

- ✓ Capacitar o aluno a **desenvolver um projeto**, simulando a **experiência profissional**, utilizando técnicas, ferramentas, metodologias e boas práticas trabalhadas ao longo do curso de tecnologia além de proporcionar o conhecimento e parceria com uma empresa de tanto prestígio como a Salesforce.



## Dica

A prática é a melhor ferramenta.

# OBJETIVO

- ✓ **O que é um site com acessibilidade?**
- ✓ Acessibilidade em um site se refere à prática de projetar e desenvolver conteúdo digital de forma a torná-lo facilmente utilizável por pessoas com diversos tipos de habilidades, incluindo aquelas com deficiências visuais, auditivas, motoras, cognitivas ou outras limitações. O objetivo da acessibilidade é garantir que todas as pessoas, independentemente de suas capacidades, possam interagir e consumir informações em um site de maneira eficaz e sem barreiras.
- ✓ No próximo slide, alguns tópicos que irá ajudar entender um pouco mais.

## Dica

Entenda o pedido do seu cliente

# OBJETIVO

- ✓ Uma pílula de conhecimento sobre o assunto:

**5.Navegação simples:** Projetar uma estrutura de navegação clara e organizada para que os usuários possam mover-se facilmente pelo site.

**6.Formulários acessíveis:** Criar formulários que possam ser preenchidos e enviados por pessoas que usam tecnologias assistivas ou têm dificuldades motoras.

**7.Tempo de interação:** Evitar limitações de tempo rígidas que possam dificultar a interação para pessoas que precisam de mais tempo para ler, entender ou responder.

**8.Linguagem clara e simples:** Escrever conteúdo de maneira clara e concisa para que seja facilmente compreensível por uma variedade de públicos.

## Dica

A objetividade é admirada por todos.

# OBJETIVO

✓ Uma pílula de conhecimento sobre o assunto:

**1. Compatibilidade com tecnologias assistivas:** Garantir que o site funcione bem com leitores de tela, teclados alternativos, dispositivos de controle por voz e outras tecnologias que as pessoas com deficiências utilizam para acessar a web.

**2. Contraste e legibilidade:** Usar combinações de cores de fundo e texto que sejam fáceis de ler para pessoas com deficiências visuais ou dificuldades de leitura.

**3. Legendas e transcrições:** Incluir legendas em vídeos para pessoas com deficiência auditiva e fornecer transcrições de áudio para tornar o conteúdo acessível.

**4. Descrições de imagens:** Adicionar textos alternativos descritivos para imagens, permitindo que leitores de tela transmitam informações visuais a usuários com deficiência visual.

## Dica

Deve ser agradável para o seu público.



# OBJETIVO

- ✓ Uma pílula de conhecimento sobre o assunto:

**9.Compatibilidade com diferentes dispositivos:** Garantir que o site funcione bem em uma variedade de dispositivos, como computadores, tablets e smartphones, oferecendo uma experiência consistente.

**10.Testes de acessibilidade:** Realizar testes regulares com pessoas com deficiências para identificar problemas e garantir que as melhorias de acessibilidade sejam eficazes.

Dica

Precisa  
funcionar!

# OBJETIVO

- ✓ Por onde começar?

Criar um site para atrair mais clientes com um chatbot e um bom back-end envolve uma combinação de design, desenvolvimento técnico e estratégia de marketing.

## 1.Planejamento

- ✓ Defina seus objetivos: Determine quais são os principais objetivos do seu site, como gerar leads, fazer vendas diretas ou fornecer suporte ao cliente.
- ✓ Identifique seu público-alvo: Compreenda quem são seus clientes ideais para personalizar o conteúdo e as funcionalidades do site.

## Dica

Entenda o pedido do seu cliente

# OBJETIVO

- ✓ Por onde começar?

## **2.Design e Desenvolvimento:**

- ✓ Faça com um design responsivo e foco na interface do usuário (UI) e Experiência do Usuário (UX) e não se esqueça, integrar o site com um chatbot, é uma ideia.

## **3.Marketing:**

- ✓ Faça sugestões que envolve o lado de marketing, isso irá ajudar a diferenciar o seu projeto.

## **4.Pesquisa:**

- ✓ Busque conversar com pessoas que precisam das soluções como a da Salesforce, então escute o que esse profissional, gostaria de ver no site.

## Dica

Entenda o  
pedido do seu  
cliente

# REGRAS BÁSICAS



# REGRAS BÁSICAS

- ✓ Máximo de **3 integrantes**;
- ✓ **Não é permitido** o desenvolvimento individual do projeto;
- ✓ **Os grupos** devem ser formados **por alunos da mesma turma**, exceto, caso o aluno tenha mudado de turma durante o segundo ano (ou seja, para vocês não se aplica pois estão no primeiro ano).
- ✓ **Mudança de grupo** é permitido somente no final do primeiro semestre, no mês de junho, através de uma solicitação formal ao Scrum Master e aprovação de todos os componentes do grupo e somente após a entrega da segunda Sprint.

# GESTÃO DO GRUPO

- ✓ Os **componentes dos grupos poderão indicar a não atribuição da nota ou expulsão** do componente, caso o componente não esteja participando do projeto. Para isso, a maioria do grupo deve estar de acordo e:
  - ✓ **Primeiro**, o componente deve ser notificado pelo grupo; Depois de duas semanas, caso o
    - ✓ componente ainda não esteja participando ativamente do projeto, notifique o Scrum Master;
- ✓ **O grupo deve notificar o Scrum Master**, que entrará em contato com o componente e dará o prazo de uma semana para participar do projeto;
- ✓ **Caso não obtenha resultado**, o grupo pode decidir a não atribuição de nota ou expulsão;
- ✓ **Caso seja um componente reincidente**, o grupo pode notificar o Scrum Master diretamente

# ENTREGAS

- ✓ Os grupos devem **idealizar os projetos** a serem desenvolvidos ao longo do ano letivo, dividido em **entregas parciais durante o 2º semestre de 2023 e 1º.Semestre de 2024;**
- ✓ Haverá um total de 4 sprints (4 entregas ao longo do primeiro ano da graduação), sendo duas no segundo semestre de 2023 e duas no primeiro semestre de 2024;
- ✓ **A entrega** ocorrerá ao final de cada Sprint e será para **todas as disciplinas;**
- ✓ **Cada sprint** terá os seus entregáveis de **cada disciplina**, especificados pelos respectivos professores no **portal e neste documento;**



# RESPONSABILIDADES





# ALUNOS

- ✓ Ter **ciência do documento da Challenge** e cumprir as entregas conforme solicitação dos professores.
- ✓ **Organização e gerenciamento do grupo**, como o planejamento e prazos
  - ✓ das entregas internas.
- ✓ Atentar-se a qualidade no **desenvolvimento dos entregáveis**.

# PROFESSORES

**Responsáveis por orientar o desenvolvimento** do projeto em suas disciplinas;

- ✓ **A correção de cada entregável**, é de responsabilidade do professor da disciplina em cada turma;
- ✓ Disponibilizar **nota individual** referente à disciplina;
- ✓ **Manter informações no grupo de professores relativas a cada grupo** (compartilhar avaliação
  - ✓ dos grupos);
- ✓ Manter **sua solicitação de entregável sempre atualizada** para o grupo de professores e alunos;
- ✓ Explicar **detalhadamente entregável da disciplina** e critérios de avaliação junto aos alunos;
- ✓ Disponibilizar **justificativa de nota aplicada** ao grupo/aluno, quando se aplicar;

# SCRUM MASTER

- ✓ **Comunicação entre os alunos/professores e a(s) empresa(s) parceira(s);**
- ✓ **Manter os dados dos grupos de alunos atualizada** (a gestão e formação dos grupos é de responsabilidade dos alunos);
- ✓ Manter o **documento da challenge** atualizada;
- ✓ **Agendar os eventos** com a empresa parceira/professores, como apresentações, reuniões e treinamentos;
- ✓ SCRUM MASTER – KARINA PALTRINIERI

# PREMIAÇÃO



# PREMIAÇÃO

## Grandes momentos



- NEXT



- PREMIAÇÃO EM DINHEIRO



- MEDALHAS, CAMISETA E SHAPES EXCLUSIVOS



# CRONOGRAMA



## CRONOGRAMA 2º SEMESTRE

DATA	EVENTO	STEAKHOLDER
05/02 a 16/02	Entrega da documentação para os alunos	Professores
18/03	Mentoria Presencial	SALESFORCE
01/04	Entrega Sprint 3	Alunos
10 dias após entrega	Feedback Sprint 3	Professores
22/04	Banca de Professores	Professores
13/05	Mentoria Presencial	SALESFORCE
20/05	Entrega Sprint 4	Alunos
10 dias após entrega	Feedback Sprint 4	Professores
10/06	Entrega do projeto para SALESFORCE (RODANDO)	Alunos



# LINHA DO TEMPO





# 3ª ENTREGAS

FIAP

# AI & CHATBOT

## Ciência de Dados para UI/UX

Nesta etapa usaremos os conceitos de representação e ciência de dados para mapear o fluxo de utilização do frontend. Considere o código de exemplo que coleta cliques e salva em um arquivo .csv (você pode alterar o Javascript para coletar as informações que achar mais relevantes). Colete dados de pelo menos 3 pessoas diferentes usando seu site e analise os dados com os conceitos visto em aula.

### Requisitos:

- Levantamento de dados de navegação na página; **[40 pontos]**
- Análise dos dados visando a produção de gráficos e estatísticas. (Por exemplo, considerem, se capturarem posição do mouse, fazer um mapa de calor da navegação); **[60 pontos]**

### Entregável:

- Arquivo **.csv** com os dados.
- Arquivo **.ipynb** com os códigos em Python, análises e explicações dos resultados.

# BUILDING RELATIONAL DATABASE

## Criação do Banco de Dados Relacional

### Requisitos:

- Por meio do seu Projeto Conceitual/Lógico desenvolvido na Sprint 1 e 2 realizar a implementação prática da estrutura de banco dados utilizando a Linguagem de Definição de Dados (DDL/SQL - Create Table);
- Atenção! Revisem a Modelagem do Projeto, assim como as possíveis correções sugeridas pelos professores.
- A modelagem precisa estar aderente ao Diagrama de Classes da UML (ou vice-versa), uma vez que as classes/objetos criados na Linguagem de BackEnd refletem as tabelas/colunas do banco de dados.

# BUILDING RELATIONAL DATABASE

## Entregáveis:

- **[1] Um Arquivo PDF contendo [30 pontos]:**
  - Uma descrição sobre a solução que foi projetada desde a SPRINT 1;
  - O Diagrama Lógico (ou Diagrama de Entidades de Relacionamento)
  - O Diagrama Relacional
  - O arquivo DEVE seguir a seguinte padronização de nomenclatura:  
Exemplo: 1TDSX\_2023\_Proj\_BD.pdf  
Onde: X = Turma

Uma 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.

# BUILDING RELATIONAL DATABASE

## Entregáveis:

- **[2] Arquivo SQL contendo [70 pontos]:**
  - 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\_CodigoSql.sql  
Onde: X = Turma

**Uma 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:

### Preparação dos dados para IA

A partir dos dados coletados nas outras disciplinas, a equipe deverá desenvolver um programa em Python, o qual deverá contemplar os seguintes itens:

- **[35 pts]** os dados deverão ser filtrados e armazenados em listas de dicionários;
- **[35 pontos]** tratamento de erros para a inserção e alteração dos dados (try...except...else...finally);
- **[30 pts]** menu com submenus para acesso ao CRUD dos dados.

### **Entregáveis:**

- Arquivo **.zip** com os arquivos do projeto do Python.
- Arquivo **.txt** com as explicações e nomes dos membros do grupo.

# DOMAIN DRIVEN DESIGN

## Documentação

Especificação Técnica [20 pontos]

- **Capa:** contendo o nome dos integrantes, nome da equipe e nome da solução. **(1 ponto)**
  - **Sumário:** contendo as sessões e numeração de páginas **(1 ponto)**
  - **Objetivo e escopo do projeto:** descritivo com a solução proposta pelo grupo de forma objetiva. **(1 pontos)**
  - **Descrição** das principais funcionalidades da solução. **(1 pontos)**
  - **Protótipos:** prints das telas implementadas para o front-end com descrição das funcionalidades e como a interação com o usuário é realizada – pode ser desenvolvida em qualquer ferramenta. **(2 pontos)**
  - **Modelagem** do banco de dados (tabelas e relacionamentos). **(2 pontos)**
  - **Diagrama de classes atualizada.** **(10 pontos)**
  - **Procedimentos para rodar a aplicação** **(2 pontos)**
- **Obs.: O projeto e o documento PDF (ambos) devem ser entregues dentro da mesma pasta compactada, porém separado projeto de documentação.**

# DOMAIN DRIVEN DESIGN

## Projeto Java

Especificação Técnica [80 pontos]

- **[20 pontos] Camada beans e model (consumo de API)**, com suas classes e seus devidos padrões seguidos em sala de aula.
- **[20 pontos] 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).
- **[5 pontos]** Camada com a classe de conexão, o usuário e senha tem que estar no código.
- **[25 pontos]** Implementação da camada e classe DAO com pelo menos um método insert, funcionando.
- **[10 pontos]** Dentro do pacote main, a implementação das **classes de execução**, que instancie no mínimo 2 objetos, os métodos implementados no tópico 2, deve constar também a classe de teste de conexão e a classe de Inserir (insert) funcionando. O código Java deve estar integrado com o *front-end* e também com o Banco de Dados.

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

•Obs.: O projeto e o documento PDF (ambos) devem ser entregues dentro da mesma pasta compactada, porém separado projeto de documentação.



# DOMAIN DRIVEN DESIGN

## Entregáveis:

### **Documentação [20 pontos]**

- **Documentação** completa (em um único arquivo) contendo a especificação da solução proposta, protótipos de tela, diagrama de Classes, Modelo de Banco de Dados e demais informações de relevância para o projeto.

- **Obs.:** O documento deve ser entregue em formato PDF.

### **Projeto [80 pontos]**

- **O código do projeto**, deve seguir os padrões de boas práticas de orientação a objetos aprendidos em aula com as classes estando corretamente em seu devidos pacotes.

- **Obs.:** O projeto e o documento PDF (ambos), devem ser entregues dentro da mesma pasta compactada, porém separado projeto de documentação.

- **Entrega:** Pelo 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.

## CONSIDERE TODOS OS ITENS A SEGUIR OBRIGATÓRIOS!

### 1. REGRAS

- Desenvolver um projeto em **REACT + TypeScript** com base na estrutura apresentada no (FIGMA) durante entregas anteriores.
- Esta fase requer a utilização de **REACT + TypeScript** para a componentização da aplicação, seguindo as melhores práticas discutidas em sala de aula.
- Reestruturar o projeto apresentado no (FIGMA) do primeiro semestre, transformando-o em uma aplicação de página única (single-page), com estrutura semântica e responsiva, conforme estabelecido nas entregas anteriores.
- A aplicação não necessita, neste momento, de integração com uma API. Contudo, se houver elementos interativos de entrada de dados do usuário (formulários de cadastro, etc.), estes devem ser incluídos conforme descrito na documentação e nas entregas anteriores.

# RESPONSIVE WEB DEVELOPMENT

## 1. REGRAS (continuação)

- O projeto deve ser desenvolvido em conformidade com as normas estabelecidas pelo W3C.
- É estritamente proibido o uso de templates ou exercícios fornecidos durante as aulas. O descumprimento dessa norma resultará automaticamente na atribuição de nota ZERO para a equipe.
- A estilização deve ser realizada utilizando styled-components ou CSS3 puro.
- Uma das páginas do projeto deverá conter os nomes e RM dos integrantes da equipe. Alunos pertencentes a outras turmas devem ser devidamente identificados com seus RM e turma.
- O projeto deve ser versionado utilizando git e hospedado em um repositório público no GitHub. Repositório no GitHub com no mínimo 10 commits, e o link do repositório no GitHub deve ser disponibilizado na mesma página junto aos nomes e RM dos integrantes.

# RESPONSIVE WEB DEVELOPMENT

## 2. CRITÉRIOS DE AVALIAÇÃO

- **Construção do projeto de acordo com a 2ª entrega (20 pontos)**
  - Escolha de Cores e Fontes (10 pontos)
    - ✓ Coerência e harmonia das cores utilizadas. (2,5 pontos)
    - ✓ Adequação das cores ao tema e propósito do projeto. (2,5 pontos)
    - ✓ Legibilidade e estética das fontes escolhidas. (2,5 pontos)
    - ✓ Consistência no uso das fontes em todo o projeto. (2,5 pontos)
  - Atratividade do Design (10 pontos)
    - ✓ Originalidade e criatividade na disposição de elementos visuais. (2,5 pontos)
    - ✓ Apelo visual para o público-alvo pretendido. (2,5 pontos)
    - ✓ Impacto emocional e estético do design. (2,5 pontos)
    - ✓ Clareza na comunicação visual e facilidade de navegação. (2,5 pontos)

# RESPONSIVE WEB DEVELOPMENT

## 2. CRITÉRIOS DE AVALIAÇÃO (continuação)

- **Componentização das páginas e itens reaproveitáveis (30 pontos)**
  - Modularidade e Reutilização
    - ✓ Identificação e separação clara de componentes. (5,0 pontos)
    - ✓ Facilidade de reutilização de elementos em diferentes partes do projeto. (5,0 pontos)
    - ✓ Eficiência na manutenção e atualização de componentes. (5,0 pontos)
  - Padronização e Consistência
    - ✓ Coerência na estruturação e organização dos elementos. (5,0 pontos)
    - ✓ Consistência no uso de estilos, classes e IDs. (5,0 pontos)
    - ✓ Documentação adequada dos componentes para facilitar o entendimento e reutilização. (5,0 pontos)

# RESPONSIVE WEB DEVELOPMENT

## 2. CRITÉRIOS DE AVALIAÇÃO (continuação)

- **Responsividade das páginas: Desktop, Tablet e Mobile. (20 pontos)**
  - Adaptabilidade
    - ✓ Funcionalidade e aparência adequada em diferentes dispositivos. (3,3 pontos)
    - ✓ Adequação do layout e conteúdo para diferentes tamanhos de tela. (3,3 pontos)
    - ✓ Navegabilidade e usabilidade em todas as plataformas. (3,4 pontos)
  - Performance
    - ✓ Velocidade de carregamento e fluidez da interface em cada dispositivo. (5,0 pontos)
    - ✓ Ausência de problemas de redimensionamento ou sobreposição de elementos. (5,0 pontos)

# RESPONSIVE WEB DEVELOPMENT

## 2. CRITÉRIOS DE AVALIAÇÃO (continuação)

- **Utilização do Git/Github, projeto com no mínimo 10 commits (10 pontos)**
  - Colaboração e Uso do Git
    - ✓ Clareza nos commits e mensagens descritivas. (3,3 pontos)
    - ✓ Demonstração de práticas adequadas de versionamento. (3,3 pontos)
    - ✓ Contribuições distribuídas e evidências de trabalho em equipe. (3,4 pontos)
  - Participação Equitativa
    - ✓ Envolvimento equitativo de todos os membros do grupo. (3,3 pontos)
    - ✓ Evidências de colaboração ativa, revisão de código e resolução de conflitos. (3,3 pontos)
    - ✓ Responsabilidade compartilhada na entrega e qualidade do código. (3,4 pontos)

# RESPONSIVE WEB DEVELOPMENT

## 2. CRITÉRIOS DE AVALIAÇÃO (continuação)

- **Entrega dentro dos padrões solicitados no documento (20 pontos)**
  - Atendimento às Especificações
    - ✓ Cumprimento dos requisitos e funcionalidades solicitadas. (2,5 pontos)
    - ✓ Conformidade com as diretrizes e padrões estabelecidos. (2,5 pontos)
  - Grupo deverá gravar um vídeo de no máximo 3 minutos apresentando os recursos do projeto, telas, layout, o mesmo poderá ser disponibilizado via link e hospedado por exemplo Youtube (10 pontos)
  - Qualidade da Entrega
    - ✓ Coerência entre o produto entregue e o descrito no documento. (2,5 pontos)
    - ✓ Ausência de erros significativos ou falhas na implementação. (2,5 pontos)



# RESPONSIVE WEB DEVELOPMENT

## 3. ENTREGA

- O grupo deve compactar e entregar o repositório do projeto em formato .ZIP, seguindo as diretrizes do .gitignore para garantir a exclusão da pasta node-modules. O não cumprimento desta norma resultará em um arquivo excedendo 50MB, ultrapassando o limite do portal da FIAP, acarretando na **PERDA** de **CINQUENTA PONTOS** para todos os integrantes do grupo.
- Grupo deverá gravar um vídeo de no máximo 3 minutos apresentando os recursos do projeto, telas, layout, o mesmo poderá ser disponibilizado via link e hospedado por exemplo Youtube.
- O aluno encarregado 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)

# RESPONSIVE WEB DEVELOPMENT

## ATENÇÃO!!!

- O grupo deve compactar e entregar o repositório do projeto em formato .ZIP, seguindo as diretrizes do .gitignore para garantir a exclusão da pasta **NODE\_MODULES**. O não cumprimento desta norma resultará em um arquivo excedendo 50MB, ultrapassando o limite do portal da FIAP, acarretando na **PERDA** de **CINQUENTA PONTOS** para todos os integrantes do grupo.
- 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 Alunos pertencentes a outras turmas devem ser claramente identificados com seus respectivos RM e turma em uma das páginas do projeto (página de integrantes).

# RESPONSIVE WEB DEVELOPMENT

## 3. ENTREGA (continuação)

- Local de entrega  
✓ Portal do aluno

## 4. DISCIPLINAS PARA INTEGRAÇÃO

- NÃO HÁ

## 5. ENTREGÁVEIS

- Projeto completo compactado (ZIP).

# SOFTWARE DESIGN & TX

Documentação funcional do projeto:

1 - Entregar documento com identidade visual da empresa/solução em arquivo ppt único contendo **(20 pontos)**:

- Capa e Sumário (com tópicos das páginas, como documento está organizado);
- Descritivo com explicação, com justificativa/objetivos, de no mínimo 20 linhas para esse contexto, permitindo a validação da entrega e compreensão do projeto. (venda da ideia).
- Explicar como o seu sistema que estará pronto até a Sprint Quatro (com os principais VMP's), conseguirá atender o challenge da Salesforce, com no mínimo 10 linhas. (como irá entregar por etapas para o cliente ficar satisfeito desde a primeira entrega)
- Gestão do projeto no Jira/Trello com todos os entregáveis das matérias e os to dos de desenvolvimento da solução. (continuando a entrega da Sprint 2.
- BMC da solução elaborado com as técnicas do design thinking (Realmente a solução atende todos os pontos do BMC).

2 - Diagramas de caso de uso que está explicado nos próximos slides **(20 pontos)**.

3 - Diagramas de atividades de uso que está explicado nos próximos slides **(20 pontos)**.

4 – Elaborar o vídeo pitch com no máximo 3 minutos, detalhado nos próximos slides.**(20 pontos)**

5- Aplicação de TX, Aplicar e trazer os resultados com as respectivas soluções das pesquisas de satisfação e quais os SLAS que irá atender (o grupo pode escolher entre os métodos que o professor ensinou). Incluir os gráficos , forma de levantamento e a análise das respostas.**(20 pontos)**

# SOFTWARE DESIGN & TX

## Elaboração do Pitch:

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.

# SOFTWARE DESIGN & TX

**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

# SOFTWARE DESIGN & TX

## 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 40 pontos para esse tópico, irá receber 10 pontos.

Mínimo de três 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.

Os Diagramas devem ser produzidos utilizando o ASTAH.

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

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.**

# 4ª ENTREGAS

FIAP



# AI & CHATBOT

## Modelos de IA

Nesta etapa, você e seu grupo deverão propor um modelo de IA que possa ser usado pelo ou para o site elaborado. Quando dizemos pelo site, nos referimos a própria aplicação web que usa o modelo de IA para entregar alguma funcionalidade ao usuário do site. Quando dizemos para o site estamos nos referindo a um modelo de IA que ajudará de maneira indireta o site, fazendo previsões ou descrições de potenciais clientes, segmentação de marketing, etc.

### Requisitos:

- Crie um modelo de aprendizado de máquina aderente ao projeto. 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. Considere que caso sua aplicação web não consuma o modelo, ele será consumido por outro membro da equipe da empresa. **[30 pontos]**

### Entregável:

- 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 implementação da API REST (caso usando Flask, **.py**, caso usando Node-RED, **.json**).

# BUILDING RELATIONAL DATABASE

## Populando o Banco de Dados Relacional

### Requisitos:

- Por meio da sua estrutura criada na SPRINT 3, realize as seguintes tarefas:
  - [1] Através da Linguagem de Manipulação de Dados (DML/SQL):
    - Inserção de 7 linhas de dados (INSERT); [15 pontos]
    - 3 exemplos de Atualização de dados (UPDATE); [10 pontos]
    - 3 exemplos de Remoção de dados (DELETE); [10 pontos]
  - [2] Através da Linguagem de Consulta de Dados (DQL/SQL):
    - 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; [10 pontos]
    - Relatório utilizando junção de tabelas; [10 pontos]

# BUILDING RELATIONAL DATABASE

## Entregáveis:

- **[1] Um Arquivo PDF contendo [10 pontos]:**
  - Uma descrição sobre a solução que foi projetada desde a SPRINT 1;
  - O Diagrama Lógico (ou Diagrama de Entidades de Relacionamento)
  - O Diagrama Relacional
  - O arquivo DEVE seguir a seguinte padronização de nomenclatura:  
Exemplo: 1TDSX\_2023\_Proj\_BD.pdf  
Onde: X = Turma

Uma 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.

# BUILDING RELATIONAL DATABASE

## Entregáveis:

- **[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 delete para limpar as possíveis tabelas já existentes no ambiente.
  - O arquivo DEVE seguir a seguinte padronização de nomenclatura:  
Exemplo: 1TDSX\_2023\_CarregaSql.sql  
Onde: X = Turma

**Uma 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.**

# BUILDING RELATIONAL DATABASE

## Entregáveis:

- **[3] Video: [ 15 pontos ]:**
  - Um vídeo demonstrando o funcionando em Java e Python gerado para a entrega da sprint das matérias correspondentes. A aplicação deve utilizar o Banco de dados gerado nas Sprints,
  - O vídeo deve conter pelo menos uma inserção, sendo demonstrado a consulta no banco desta inserção.
  - O vídeo deve mostrar também a execução de um relatório com os dados que constam no script entregável da sprint 4.

**Uma 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.**

# BUILDING RELATIONAL DATABASE

## **Ponto de atenção:**

- O banco de dados tem que atender a necessidade do back-end, se o professor avaliar e justificar no feedback que não está alinhado e integrado, poderá perder até 30 dos 100 pontos.

**Uma 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:**

Tendo como referência o entregável da sprint 3, a equipe deverá realizar a integração com o banco de dados Oracle, desenvolvendo um CRUD em Python. Para tanto, os requisitos são os seguintes:

- [35 pontos] Estrutura de menus e submenus para acesso às opções do CRUD (inserir, alterar, excluir e consultar);
- [40 pontos] Realizar pelo menos 2 consultas utilizando filtros (select...where) e disponibilizar uma opção para exportar os dados dessas consulta para um arquivo JSON.
- [15 pontos] As informações colhidas e/ou alteradas pelo programa desenvolvido devem refletir no sistema web e vice-versa.

## **Entregável:**

Compactar em um arquivo .zip:

- Código fonte em Python (arquivos .py)
- [10 pontos] Documento em pdf contendo uma capa com os nomes dos integrantes, um sumário, uma descrição da solução proposta com os objetivos e escopo do projeto, além dos scripts necessários para criação das tabelas no banco de dados Oracle.

# DOMAIN DRIVEN DESIGN

## Documentação

Especificação Técnica [20 pontos]

- **Capa:** contendo o nome dos integrantes, nome da equipe e nome da solução.
- **Objetivo e escopo do projeto:** descritivo com a solução proposta pelo grupo de forma objetiva.
- **Descrição das principais funcionalidades da solução.**
- **Tabela dos *endpoints* (API Restful):** contendo as URLs, verbo HTTP e Códigos de status de respostas para a requisição.
- **Protótipos:** prints das telas implementadas para o front-end.
- **Modelo do banco de dados.**
- **Diagrama de classes atualizada.**
- **Procedimentos para rodar a aplicação.**
- **O documento de conter o Link do repositório com do código Java do GitHub.**
- **Obs.: O projeto e o documento PDF (ambos) devem ser entregues dentro da mesma pasta compactada, porém separado projeto de documentação.**



# DOMAIN DRIVEN DESIGN

## Projeto Java - versão final

Requisitos Técnicos [80 pontos]

- **[10 pontos] Camada beans e model (consumo de API)** com todas as classes necessárias, de acordo com o banco de dados.
- **[30 pontos] Camada DAO e Service** (validações), com todas as funcionalidades necessárias para o *front-end* do projeto.
- **[30 pontos] API Restful** com todos os *endpoints* necessários para o funcionamento do *front-end* do projeto. A implementação Java deve estar integrada com o *front-end* e também com o banco de dados (solução completa).
- **[10 pontos] Seguir todas as boas práticas** trabalhadas em sala de aula, como regras de nomenclatura, tratamento de exceções, padrões de projetos.

•Obs.: O projeto e o documento PDF (ambos), devem ser entregues dentro da mesma pasta compactada, porém separado projeto de documentação.

# DOMAIN DRIVEN DESIGN

## Entregáveis:

### Documentação [20 pontos]

- **Documentação** completa (em um único arquivo) contendo a especificação da solução proposta, diagrama de Classes, Modelo de Banco de Dados e demais informações importantes do projeto.
- **Obs.:** O documento deve ser entregue em formato PDF.

### Projeto Java [80 pontos]

- **O código do projeto**, deve seguir os padrões de boas práticas de orientação a objetos aprendidos em aula com as classes estando corretamente em seu devidos pacotes.
- **Obs.:** O projeto e o documento PDF (ambos), devem ser entregues dentro da mesma pasta compactada, porém separado projeto de documentação.
- **Entrega:** Pelo Portal do Aluno no menu do *Challenge*.

# DOMAIN DRIVEN DESIGN

## Ponto de atenção:

- O back-end tem que estar alinhado com o banco de dados, se o professor avaliar e justificar no feedback que não está alinhado e integrado, poderá perder até 30 dos 100 pontos.

# RESPONSIVE WEB DEVELOPMENT

Chegou a hora de finalizarmos nosso projeto.

**CONSIDERE TODOS OS ITENS A SEGUIR OBRIGATÓRIOS!**

## 1. REGRAS

Para esta etapa, vamos considerar as regras da 3ª entrega MAIS as regras abaixo.

- Para completar nossa implementação, é crucial a integração de uma **API** em nosso projeto. Essa etapa marca a convergência de todos os elementos do projeto, assegurando que o *frontend* esteja perfeitamente alinhado com o *backend*.
- A API, desenvolvida na disciplina de *Domain Drive Design*, 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, tal como abordado na disciplina de *Building Relational Database*.

# RESPONSIVE WEB DEVELOPMENT

## 1. REGRAS (continuação)

- É essencial que as informações capturadas ou alteradas pelo sistema desenvolvido na disciplina de *Computational Thinking Using Python* e/ou na disciplina de *Domain Drive Design* sejam refletidas no sistema web, e vice-versa, para garantir uma integração plena.
- O projeto final deverá ser estruturado utilizando o **REACT + TypeScript** para o roteamento de páginas e outras adaptações necessárias.
- Além disso, é fundamental realizar o *deploy* do projeto na plataforma *Vercel*, onde será disponibilizada uma URL de acesso para a avaliação.

# RESPONSIVE WEB DEVELOPMENT

## 2. CRITÉRIOS DE AVALIAÇÃO

- **Construção do projeto de acordo com a 3ª entrega**
  - ✓ Estrutura e Organização do Código (7 pontos)
    - Clareza na estruturação do projeto e organização dos arquivos. (2,0 pontos)
    - Consistência na nomenclatura de variáveis, funções e componentes. (2,0 pontos)
    - Modularidade e reutilização de código. (3,0 pontos)
  - ✓ Estilo e Design (7 pontos)
    - Escolha de cores, fontes e elementos visuais alinhados com o propósito do projeto. (3,5 pontos)
    - Adaptação e responsividade do design em diferentes dispositivos. (3,5 pontos)
  - ✓ Funcionalidades Implementadas (6 pontos)
    - ✓ Implementação bem-sucedida das funcionalidades descritas no escopo do projeto. (3,0 pontos)
    - ✓ Ausência de bugs críticos ou erros que impactem a usabilidade. (3,0 pontos)

# RESPONSIVE WEB DEVELOPMENT

## 2. CRITÉRIOS DE AVALIAÇÃO (continuação)

- **Criação de Rotas e Navegação com TYPESCRIPT** (20 pontos)
  - ✓ **Correta Implementação de Rotas** (8 pontos)
    - Definição adequada e funcional de rotas utilizando REACT + TYPESCRIPT. (4,0 pontos)
    - Navegação fluida entre páginas. (4,0 pontos)
  - ✓ **Gerenciamento de Estado da Aplicação** (7 pontos)
    - Utilização eficiente do estado da aplicação para a troca de informações entre as páginas. (3,5 pontos)
    - Manuseio correto de parâmetros de rota, quando aplicável. (3,5 pontos)
  - ✓ **Performance e Otimização** (5 pontos)
    - Desempenho satisfatório na transição entre rotas. (2,5 pontos)
    - Utilização de técnicas para otimização do carregamento das páginas. (2,5 pontos)

# RESPONSIVE WEB DEVELOPMENT

## 2. CRITÉRIOS DE AVALIAÇÃO (continuação)

- **Consumo das APIs** (25 pontos)
  - ✓ Integração Adequada com APIs (10 pontos)
    - Correto uso das chamadas de API conforme a documentação fornecida. (5,0 pontos)
    - Tratamento de erros e respostas inesperadas. (5,0 pontos)
  - ✓ Funcionalidades Implementadas com APIs (8 pontos)
    - Integração bem-sucedida das funcionalidades que dependem das APIs. (4,0 pontos)
    - Manipulação correta dos dados obtidos das requisições. (4,0 pontos)
  - ✓ Segurança e Gerenciamento de Dados (7 pontos)
    - ⑩ Garantia de segurança nas transações de dados entre o *frontend* e o *backend*. (3,5 pontos)
    - ⑩ Adequado tratamento e armazenamento dos dados obtidos das APIs. (3,5 pontos)



# RESPONSIVE WEB DEVELOPMENT

## 2. CRITÉRIOS DE AVALIAÇÃO (continuação)

- **Utilização do Git/Github, Projeto com no Mínimo 10 Commits e Participação de Todos os Integrantes** (5 pontos)
  - ✓ **Colaboração e Participação** (5 pontos):
    - Evidências de colaboração equitativa de todos os membros do grupo. (2,5 pontos)
    - Mensagens descritivas nos *commits* indicando claramente o que foi realizado. (2,5 pontos)
  - ✓ **Estruturação do Repositório** (5,0 pontos)
    - Organização adequada dos arquivos no repositório. (2,5 pontos)
    - Demonstração de uso efetivo de *branches*, merges e resolução de conflitos, se aplicável. (2,5 pontos)
- **Grupo deverá gravar um vídeo de no máximo 3 minutos apresentando os recursos do projeto, telas, layout, API funcionando, o mesmo poderá ser disponibilizado via link e hospedado por exemplo Youtube.(5 pontos)**

# RESPONSIVE WEB DEVELOPMENT

## 2. CRITÉRIOS DE AVALIAÇÃO (continuação)

- **Deploy do Projeto para a Plataforma Vercel** (15 pontos):
  - ✓ **Sucesso no *Deploy*** (8 pontos):
    - Efetivação bem-sucedida do *deploy* na plataforma *Vercel*.  
(4,0 pontos)
    - Disponibilização de uma URL funcional para acesso ao projeto. (4,0 pontos)
  - ✓ **Configuração e Otimização** (7 pontos):
    - Configuração adequada das variáveis de ambiente, se necessárias. (3,5 pontos)
    - Otimização para um carregamento rápido e eficiente do projeto. (3,5 pontos)

# RESPONSIVE WEB DEVELOPMENT

## 2. CRITÉRIOS DE AVALIAÇÃO (continuação)

- **Entrega Dentro dos Padrões Solicitados no Documento** (10 pontos):
  - ✓ **Atendimento às Especificações** (5 pontos):
    - Cumprimento dos requisitos e funcionalidades estabelecidas no documento. (2,5 pontos)
    - Conformidade com as diretrizes e padrões exigidos. (2,5 pontos)
  - ✓ **Qualidade da Entrega** (5 pontos):
    - Coerência entre o produto entregue e o descrito no documento. (2,5 pontos)
    - Ausência de erros significativos ou falhas na implementação. (2,5 pontos)

# RESPONSIVE WEB DEVELOPMENT

## 3. ENTREGA

- O grupo deve compactar e entregar o repositório do projeto em formato .ZIP, seguindo as diretrizes do .gitignore para garantir a exclusão da pasta node-modules. O não cumprimento desta norma resultará em um arquivo excedendo 50MB, ultrapassando o limite do portal da FIAP, acarretando na **PERDA de CINQUENTA PONTOS** para todos os integrantes do grupo.
- Grupo deverá gravar um vídeo de no máximo 3 minutos apresentando os recursos do projeto, telas, layout, o mesmo poderá ser disponibilizado via link e hospedado por exemplo Youtube. (5 pontos)
- O aluno encarregado 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).

# RESPONSIVE WEB DEVELOPMENT

## 3. ENTREGA (continuação)

- Local de entrega  
✓ Portal do aluno

## 4. DISCIPLINAS PARA INTEGRAÇÃO

- Building Relational Database
- Computational Thinking Using Python (opcional)
- Domain Drive Design

## 5. ENTREGÁVIS

- Projeto completo compactado (ZIP).

# RESPONSIVE WEB DEVELOPMENT

## ATENÇÃO!!!

- O grupo deve compactar e entregar o repositório do projeto em formato .ZIP, seguindo as diretrizes do .gitignore para garantir a exclusão da pasta **NODE\_MODULES**. O não cumprimento desta norma resultará em um arquivo excedendo 50MB, ultrapassando o limite do portal da FIAP, acarretando na **PERDA** de **CINQUENTA PONTOS** para todos os integrantes do grupo.
- 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. Alunos pertencentes a outras turmas devem ser claramente identificados com seus respectivos RM e turma em uma das páginas do projeto (página de integrantes).

# RESPONSIVE WEB DEVELOPMENT

## **Ponto de atenção:**

- O front-end tem que estar integrado e funcionando com o back-end, caso não esteja, se o professor avaliar e justificar no feedback que não está alinhado e integrado, poderá perder até 30 dos 100 pontos.

# SOFTWARE DESIGN & TX

Documento descritivo do projeto em arquivo PDF único contendo:

1) Documentação do projeto incluindo o plano de negócios. **(15 pontos):**

Entregar projeto finalizado com identidade visual da empresa/solução em arquivo ppt único contendo

- Capa e Sumário (com tópicos das páginas, como documento está organizado);
- Descritivo com explicação do projeto finalizado, explicando cada entrega das MVP's, de no mínimo 20 linhas para esse contexto, permitindo a validação da entrega e compreensão do projeto. (venda da ideia).
- Gestão do projeto no Jira/Trello com todos os entregáveis das matérias e os to dos de desenvolvimento da solução 100% finalizado.
- BMC da solução elaborado com as técnicas do design thinking (Realmente a solução atende todos os pontos do BMC).

2) Cálculo de precificação da sua solução (durante e após implementação). **(20 pontos)**

Avalie os custos que sua equipe teve e irá ter para entregar a solução funcionando e a permanência dela em perfeitas condições de uso (inclua horas dedicadas, tipos de níveis de profissionais dedicados, materiais, inteligência ). Precisamos entender qual o custo, investimentos e a viabilidade da sua solução. Deve ser entregue todo o racional de cálculo.



# SOFTWARE DESIGN & TX

## 3) Diagrama de Atividades **(15 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.

## 4) Diagrama de Sequência **(15 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.

5) Aplicação de TX, Aplicar e trazer os resultados com as respectivas soluções das pesquisas de satisfação e quais os SLAS que irá atender (o grupo pode escolher entre os métodos que o professor ensinou). Incluir os gráficos, forma de levantamento e a análise das respostas. **(20 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

### 6) Vídeio 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 colocado no PDF, deve conter 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 2 minutos e máximo de 5 minutos.

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.**