

# Challenge

---

2TDS – Turma de Agosto



FIAP

 odontoprev



# DESAFIO



---

Redução de Sinistro com Análise preditiva de atendimento

# Sobre a Odontoprev

A Odontoprev foi fundada em 1987 por cinco cirurgiões-dentistas, que entendiam que era preciso popularizar o acesso à Odontologia de qualidade, tanto à classe dos dentistas, como às camadas da população.

**+8,6 mi**  
de beneficiários

Contamos com  
**+27 mil**  
dentistas credenciados

Presente em  
**+2,5 mil**  
municípios brasileiros

**37**  
Anos de história

Qualidade garantida em  
**100%**  
dos procedimentos realizados

A odontologia  
pode ser mais.

# Nossas Marcas

Possuímos diversas marcas comerciais, dedicadas exclusivamente ao atendimento odontológico.

Cada uma delas possui características específicas que permitem atender aos diferentes públicos de forma diversificada e precisa. Nossos mais de 400 planos oferecem ampla cobertura, além de muitas vantagens e benefícios que garantem uma experiência completa aos nossos beneficiários, sejam eles do segmento corporativo, PME ou de Planos Individuais.

 odontoprev

 odonto  
system

 PRÍVIA  
Seguros e Previdência

 BRASIL DENTAL

 bradesco  
dental

 MOGIDONTO





## Desmistificando o Sinistro

Além da aposta na especialização e na qualidade da oferta, o sucesso do modelo de negócio Odontoprev também se apoia na prevenção como fator chave para o equilíbrio dos custos no decorrer do tempo.

Assim, diferentemente dos planos médicos - hospitalares, a sinistralidade em Odontologia apresenta um padrão distinto, com custos mais elevados no início do contrato devido à demanda imediata por serviços odontológicos.

# O desafio

Os alunos serão desafiados a criar soluções inovadoras focadas na redução de sinistros no setor odontológico. Especificamente, o desafio envolve o uso de análise preditiva para antecipar e mitigar situações de sinistro relacionadas aos atendimentos odontológicos.

O objetivo central é criar soluções que utilizem análise preditiva para reduzir sinistros. Alguns pontos importantes:

**Sinistro em Odontologia:** Sinistros ocorrem quando há materialização do risco, ou seja, o uso do serviço, o sinistro considerado ruim ocorre quando há um alto custo ou uso indevido de serviços.

No contexto odontológico, isso pode envolver excesso de consultas, tratamentos desnecessários ou fraudes.

**Análise Preditiva:** Técnica que utiliza dados e algoritmos para prever padrões futuros. No caso, os alunos deverão desenvolver modelos que possam antecipar atendimentos ou comportamentos que possam resultar em sinistros.

Dados sobre Odontoprev: <https://ri.odontoprev.com.br/>





# O desafio

**Soluções Inovadoras:** As soluções podem incluir o uso de machine learning, inteligência artificial ou análise de dados para identificar sinais de risco antes que se transformem em um sinistro. Por exemplo, prever padrões de comportamento de pacientes ou dentistas que possam indicar necessidade de intervenção.

A ideia é que os alunos apresentem propostas que ajudem a Odontoprev a reduzir custos e aumentar a eficiência, além de melhorar a experiência dos pacientes e garantir a sustentabilidade do negócio. Eles devem pensar em formas de aplicar tecnologia para:

- Identificar riscos em tempo real;
- Antecipar padrões problemáticos;
- Sugerir ações preventivas.





# Cronograma 2º Semestre

| Data   | Evento   | Stakeholder       |
|--------|--|-------------------|
| 20/mar | Entrega da Sprint 3                              | Alunos            |
| 27/mar | Prazo limite para postagem das notas da Sprint 1 | Professores       |
| 07/abr | Mentoria presencial na FIAP                      | Odontoprev e FIAP |
| 15/mai | Entrega da Sprint 4                              | Alunos            |
| 21/mai | Prazo limite para postagem das notas da Sprint 1 | Professores       |
| 09/jun | Banca de Professores                             | Professores       |
| 25/jun | Apresentação Final Odontoprev                    | Odontoprev        |
| 08/nov | NEXT   | FIAP              |

# Entregáveis



# Sprint 3

---





# MASTERING RELATIONAL AND NON RELATIONAL DATABASE

## 1. Correções

- Realizar a **correção dos apontamentos realizados na sprint anterior.**  
**Valor: 5 pontos.**


## 2. Empacotamento de Objetos

- Criar o **empacotamento (Package)** de todos os objetos criados no projeto.  
**Valor: 20 pontos.**

## 3. Criação de Procedures para Relatórios(Consultas)

- Desenvolver **2 procedures** para geração de relatórios e exibição dos dados no front-end da aplicação.  
**Valor: 25 pontos.**

As procedures deverão atender aos seguintes requisitos mínimos:

- Utilizar **2 inner joins.**
  - Utilizar pelo menos **1 left join** ou **1 right join.**
  - Incluir **1 cursor.**
  - Implementar **1 função de agregação.**
  - Aplicar **1 função de tratamento de dados.**
  - Ordenar os dados para melhorar a visualização.
- 



# MASTERING RELATIONAL AND NON RELATIONAL DATABASE

## 4. Criação de Trigger para Auditorias

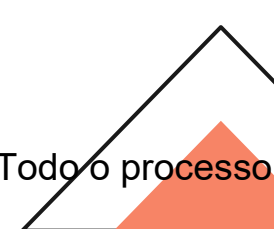
•Desenvolver uma **trigger de auditoria** que registre em uma tabela auxiliar todas as alterações (INSERT, UPDATE e DELETE) realizadas nas **3 principais tabelas do projeto**.

**Valor:** 30 pontos.

## 5. Testes e Demonstração em Vídeo

**Valor:** 20 pontos.

•Gravar um vídeo demonstrando e testando todos os itens entregues, atendendo aos seguintes critérios:

- **Testar as procedures** dentro das packages via aplicação com as seguintes operações:
    - **3 chamadas** para INSERT.
    - **2 chamadas** para UPDATE.
    - **2 chamadas** para DELETE.
    - 2 Exceptions
  - Demonstrar os registros de auditoria na tabela auxiliar.
  - Extrair um relatório na aplicação com os dados processados.
  - **Observação:** Não utilizar Postman ou Swagger para as demonstrações. Todo o processo deve ser realizado diretamente na aplicação.
- 

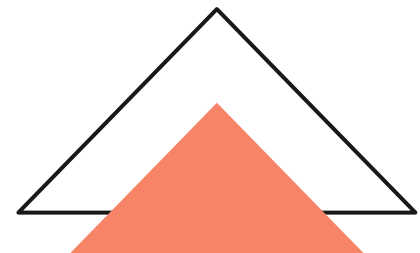


# MASTERING RELATIONAL AND NON RELATIONAL DATABASE

## 6. Entregáveis

- Enviar um arquivo **.zip** contendo os seguintes itens:
  - Um documento **PDF** com a documentação do projeto, incluindo:
    - Descrição detalhada dos componentes desenvolvidos.
    - Localização do vídeo de demonstração.
    - Nome de todos integrantes do projeto
  - Todos os códigos desenvolvidos (procedures, triggers, packages, etc.) organizados.

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







## COMPLIANCE & QUALITY ASSURANCE

**Utilizando a ferramenta de planejamento de projeto ágil em nuvem, a qual foi apresentada em suas aulas, aplicando SCRUM, faça:**

- Criação do backlog de produto, organizado em Épicos (Epic), Funcionalidades (Feature) e Itens de entrega de produto (Product backlog itens) - peso 25%
- Descrição de cada item do planejamento (Epic, Feature, Product backlog itens) - peso 25%
- Critérios de aceite de cada item do planejamento (Epic, Feature, Product backlog itens) - peso 25%
- Organização do backlog na ordem de prioridade de execução/realização do desenvolvimento por SPRINT– peso 25%

### **Entrega:**

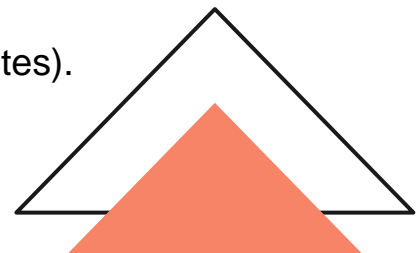
- Coloque o link de acesso ao seu plano de projeto, o qual precisa estar configurado para acesso público, permitindo o acesso integral do seu professor para correção.
- 



## DEVOPS TOOLS E CLOUD COMPUTING 1/3

Utilizando uma das tecnologias aprendidas: Serviços de Aplicativos ou ACR/ACI **(deve-se utilizar ambos ACR e ACI nessa opção)**:

- Implantar a solução ou parte dela (um módulo do sistema) da Odontoprev **em Nuvem**;
- Serão aceitas soluções com Front ou API;
- Obrigatório um Banco de Dados em Nuvem que não seja o H2. Serão aceitos: Oracle, Mysql, SQL da Azure ou PostgreSQL;
- Realize um CRUD (Inclusão, Alteração, Exclusão e Consulta) sobre **pelo menos duas tabelas** com **um relacionamento entre elas**;
- Inclua e movimente no mínimo 5 linhas com conteúdo significativo nessas tabelas (testes).

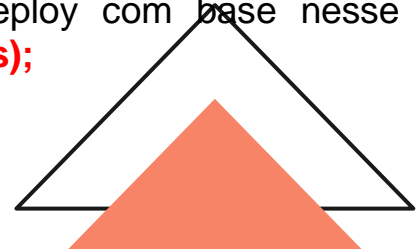




## DEVOPS TOOLS E CLOUD COMPUTING 2/3

### Entregas obrigatórias:

- 1) Uma breve descrição de sua solução; **(0 pontos – Na falta a nota será subtraída em 05 pontos);**
- 2) O desenho da arquitetura da solução proposta (desenho baseado em nossa disciplina, mostrando os recursos, fluxos da informação etc); **(até 20 pontos);**
- 3) Uma breve descrição dos benefícios a serem alcançados em relação ao negócio (com a implantação da solução proposta); **(0 pontos – Na falta a nota será subtraída em 05 pontos);**
- 4) DDL das tabelas (tabelas, colunas, chave primária, comentários etc) - **Criar um arquivo de texto somente com esse DDL; (até 05 pontos);**
- 5) Código-Fonte da aplicação no Github (e com tudo que é necessário para a execução do App/API) com um Readme.md para realizar o Deploy e testes (O Prof. irá reproduzir e testar o deploy com base nesse Readme.md). Incluir os scripts JSON do CRUD, se a solução for uma API; **(até 05 pontos);**





## DEVOPS TOOLS E CLOUD COMPUTING 3/3

### Entregas obrigatórias:

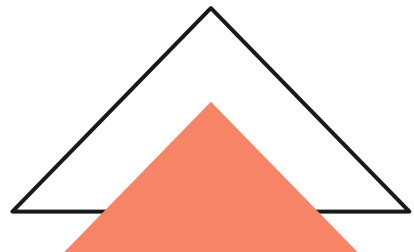
6) Uma gravação de um vídeo mostrando o funcionamento da solução, desde o Clone do Repositório (ou do build da Imagem), mostrando o funcionamento até persistência de dados em Nuvem MS Azure. Inclusive, o vídeo deve mostrar os dados persistidos no banco de dados (CRUD); **(até 70 pontos);**

**Obs: O vídeo precisa ter qualidade de no mínimo 720p, e conter áudio ou legenda explicando o desenvolvimento da solução em nuvem.**

7) Arquivo PDF, contendo o nome e RM dos integrantes do time, **link do Github e Link do Vídeo. (0 pontos);**

**Obs: O professor precisa ter acesso ao vídeo.**

**A PONTUAÇÃO DE CADA QUESITO VAI VARIAR DE ACORDO COM A QUALIDADE DA TAREFA ENTREGUE!**



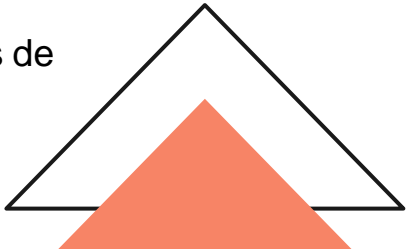


# ADVANCED BUSINESS DEVELOPMENT WITH .NET (1/2)

## Objetivo Geral

Desenvolver uma API utilizando ASP.NET Core Web API, aplicando princípios de arquitetura de software, design patterns, técnicas de documentação, testes e integração com banco de dados.

## Requisitos

- O grupo deve criar os seguintes artefatos:
    - **(até 15 pontos)** - Definir a arquitetura da API, explicando a escolha entre uma abordagem monolítica ou microservices e justificando a decisão.
    - **(até 15 pontos)** - Implementar a API seguindo a arquitetura escolhida e explicar as diferenças.
    - **(até 30 pontos)** - Endpoints CRUD (ORACLE) para os recursos de escolha (ex: produtos, usuários).
- 

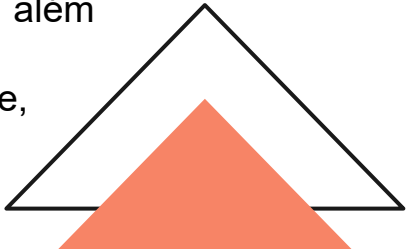


## ADVANCED BUSINESS DEVELOPMENT WITH .NET (2/2)

### Requisitos (continuação)

- O grupo deve criar os seguintes artefatos:
  - **(até 25 pontos)** - Implementar pelo menos um padrão de criação na API (ex: Singleton para o gerenciador de configurações).
  - **(até 15 pontos)** - Configurar a documentação da API utilizando Swagger/OpenAPI, com descrições claras dos endpoints e modelos de dados.

### Entrega

- O grupo deve criar um repositório público no **GitHub** com o código da API e incluir no repositório um README.md com integrantes do grupo, explicação da arquitetura, design patterns utilizados, instruções para rodar a API, além de exemplos de testes.
    - Se os endpoints CRUD não estiverem funcionando corretamente, será aplicada uma penalidade de 40 pontos.
- 





# DISRUPTIVE ARCHITECTURES: IOT, IOB & GENERATIVE IA

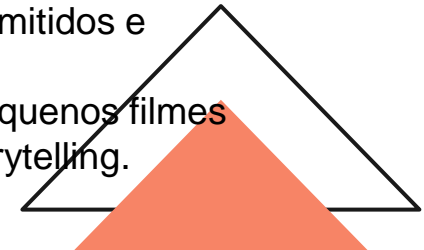
## **Objetivo Principal**

Apresentação do Protótipo Funcional e Análise da Arquitetura de IA.

## **Objetivos Específicos**

- Demonstração do protótipo funcional até o momento: apresentar o estado atual do projeto, demonstrando as funcionalidades implementadas até o momento. Apontar as dificuldades e as conclusões sobre os problemas apurados até o momento deste sprint.
- Detalhamento da arquitetura de IA: descrever a arquitetura da IA utilizada, explicar porque essa arquitetura foi escolhida e como ela foi / será implementada.
- Apresentar a base de dados utilizada para o treinamento/teste da atual versão do projeto.

## **Observações:**

- Não explique os conceitos e como os frameworks funcionam, e sim como poderão ser utilizados no projeto.
  - É FUNDAMENTAL que nessa etapa sejam demonstradas funcionalidades do projeto e evoluções em relação à entrega anterior. Erros de funcionamento podem ser admitidos e devem explicados para as correções a serem efetuadas.
  - O vídeo deverá ser gravado por um (ou mais) integrantes do grupo. Utilizem pequenos filmes funcionais, imagens do produto e seus fluxos podem ser um bom roteiro de storytelling.
- 



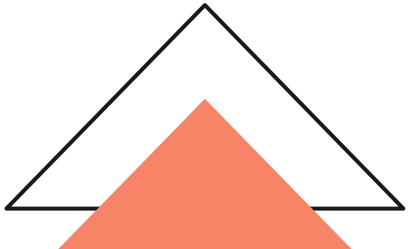
# DISRUPTIVE ARCHITECTURES: IOT, IOB & GENERATIVE IA

## Requisitos

- **[até 40 pontos]** Evoluções do protótipo funcional, com foco na análise detalhada e clara da arquitetura de ML / IA / Análise de Dados e sua implementação.
- **[até 20 pontos]** Integração do modelo ao projeto principal e às demais disciplinas.
- **[até 20 pontos]** Organização e estrutura da documentação do projeto no GitHub.
- **[até 20 pontos]** Criatividade da apresentação e da solução demonstrada.

## Entrega

Arquivos entregáveis (todos obrigatórios):

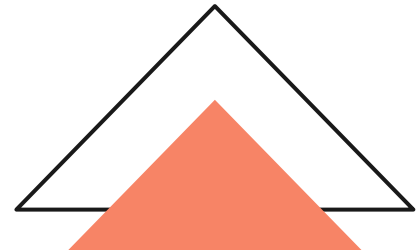
- Link ou arquivo do novo vídeo com a apresentação, conforme objetivos acima.
  - Link para a documentação organizada no GitHub.
- 



# DISRUPTIVE ARCHITECTURES: IOT, IOB & GENERATIVE IA

## Condições de entrega

- A integridade e o conteúdo do arquivo entregue são de responsabilidade dos integrantes do grupo. Arquivos entregues sem conteúdo ou com arquivos corrompidos não serão considerados.
- Não serão aceitos arquivos enviados pelo Teams ou fora do prazo.

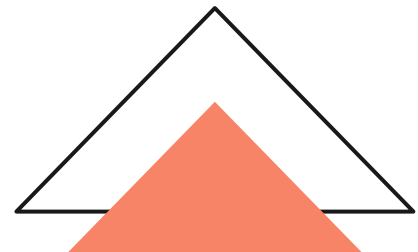




# JAVA ADVANCED (1/2)

## Requisitos

- **Aplicação MVC inicial / parcial para a solução proposta.**
- Nesta 3ª Sprint, a equipe precisa desenvolver uma aplicação web MVC que suporte a solução proposta. O grupo deve criar os seguintes artefatos:
  - **Código fonte da aplicação (70 pts)**
    - A aplicação deve utilizar o Thymeleaf em conjunto com o framework Spring Boot;
    - É importante que a solução de software já apresente templates dinâmicos;
    - Integração com o banco de dados;
    - Realizar pelos menos 2 CRUDs.





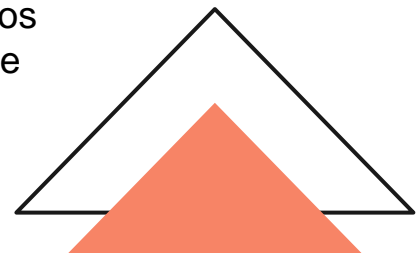
## JAVA ADVANCED (2/2)

### Requisitos (continuação)

- **Vídeo com a apresentação da aplicação funcionando com duração máxima de 10 minutos (30 pts)**
  - O vídeo deve abordar os seguintes tópicos:
    - Objetivo, Arquitetura da solução, Funcionamento da aplicação, Dificuldades encontradas até o momento, Próximos passos.

### Entrega

- A entrega deve ser feita via repositório público do **GitHub** com arquivo ReadMe apresentando toda a solução proposta (nome dos integrantes do grupo, diagramas, desenvolvimento e exemplos de testes).





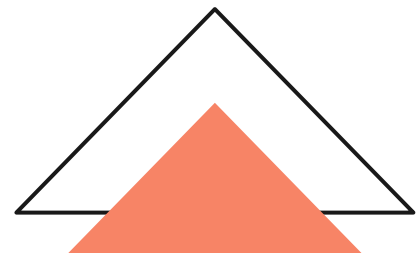
# MOBILE APP DEVELOPMENT

- **Requisitos:**

- Construa um App com 5 telas (40 pontos);
- Crie chamadas as APIs que façam sentido ao teu projeto de App (20 pontos);
- Crie um diagrama de arquitetura (estrutura de pastas) do seu aplicativo (20 pontos);
- Crie recursos atuais de navegabilidade e usabilidade (20 pontos).

- **Entrega:**

- Entregar o caminho do repositório do git contendo no readme.md as informações de nome do grupo, explanação do projeto





# Sprint 4

---





# COMPLIANCE & QUALITY ASSURANCE

**Crie o plano de testes manuais, de validação no nível de sistema para seu projeto:**

- Liste os testes planejados – peso 20%
  - Descreva os dados de entrada para cada teste – peso 20%
  - Descreva os dados de saída para cada teste – peso 20%
  - Descreva o procedimento (passos) de teste aplicado – peso 20%
  - Aponte o status de cada teste (se pendente ou realizado e se realizado, indique quem executou o teste e se passou ou não na avaliação) - peso 20%
- 
- **Obs:** trabalhe com dados controlados nos seus testes, predefinindo os valores de variáveis para input e output esperado.

## **Entrega:**

Gere um arquivo com RM e nome dos integrantes, e o Link do seu projeto no Azure.





# MASTERING RELATIONAL AND NON RELATIONAL DATABASE

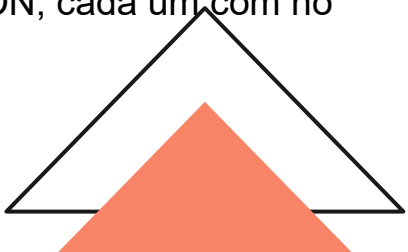
## **Projeto de Banco de Dados NoSQL MongoDB (até 40 pontos):**

**Documento de Projeto:** Apresente uma descrição detalhada do projeto e a justificativa para a escolha do MongoDB como Banco de Dados NoSQL.

**Modelo de Dados e Justificativas:** Descreva os modelos de dados utilizados, explicando como as estruturas de dados atendem aos requisitos do projeto.

**Construção de Dados e Operações (até 60 pontos):** Realize as seguintes tarefas com base no MongoDB:

**Criação de Documentos JSON/BSON:** Crie 10 documentos em formato JSON/BSON, cada um com no mínimo 10 atributos.





# MASTERING RELATIONAL AND NON RELATIONAL DATABASE

**Interface de Consulta de Dados (30 pontos):** Implemente uma interface que permita:

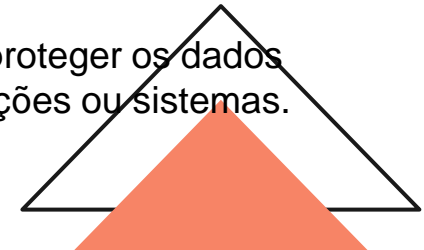
**Consulta de Dados:** Visualização e busca de documentos no MongoDB.

**Operações CRUD (Create, Read, Update, Delete):** Interface para inserir, atualizar, e excluir documentos.

**Exportação de Dataset (20 pontos):** Exporte um dataset dos dados inseridos no MongoDB, demonstrando os documentos criados e sua estrutura.

**Análise de Performance e Escalabilidade:** Forneça uma breve análise sobre as expectativas de performance para as operações de consulta implementadas. Discuta como o sistema pode ser escalado.

**Segurança e Integração:** Elabore sobre medidas de segurança recomendadas para proteger os dados no MongoDB. Descreva também como o MongoDB pode ser integrado a outras aplicações ou sistemas.

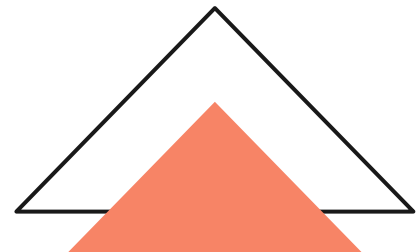




# MASTERING RELATIONAL AND NON RELATIONAL DATABASE

## Entregável:

Arquivo zip contendo o PDF do Projeto de Banco de Dados NoSQL MongoDB e os arquivos de script referentes ao banco de dados escolhido.



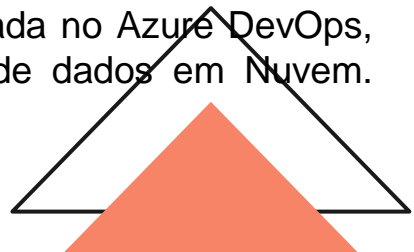


## DEVOPS TOOLS E CLOUD COMPUTING 1/2

Utilizando o conhecimento adquirido em DevOps, é esperado o grupo inserir uma solução ou parte dela (um módulo) em uma típica esteira contínua DevOps (Pipeline), permitindo realizar Deploy de forma mais rápida e com menos custo operacional, melhorando significativamente a qualidade das aplicações:

### Entregas obrigatórias:

- 1) Uma breve descrição de sua solução; **(0 pontos – A falta desse quesito será descontado 05 pontos);**
- 2) O desenho da Pipeline de integração contínua e entrega contínua (CI/CD) com foco na solução definida pelo seu grupo. Inclua o detalhamento de cada etapa criada (dissertação); **(20 pontos);**
- 3) Configuração das pipelines (CI e CD) definido na ferramenta Azure DevOps; **(40 pontos);**
- 4) Uma gravação de um vídeo mostrando o funcionamento da solução, desde a entrada no Azure DevOps, executando a Pipeline e mostrando o funcionamento da solução até persistência de dados em Nuvem. Mostrar os dados persistidos no banco inclusive; **(30 pontos);**







## DEVOPS TOOLS E CLOUD COMPUTING 2/2

### Entregas obrigatórias:

- 5) Código fonte da aplicação no Github, código da Pipeline (e com tudo que é necessário para a execução) com um Read.me para realizar a execução da Pipeline e testes (o Prof irá reproduzir e testar com base nesse Readme.md). Incluir os scripts JSON do CRUD se a solução for uma API; **(até 05 pontos);**
- 6) Deve-se utilizar pelo menos duas tabelas com um relacionamento entre elas e o Banco de Dados em Nuvem; **(Até 05 pontos);**
- 7) Arquivo PDF, contendo o nome e RM dos integrantes do time, link do Github e link do YouTube.

**A PONTUAÇÃO DE CADA QUESITO VAI VARIAR DE ACORDO COM A QUALIDADE DA TAREFA ENTREGUE!**



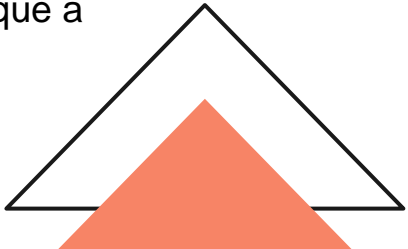


# ADVANCED BUSINESS DEVELOPMENT WITH .NET (1/2)

## Objetivo Geral

Continuação do desenvolvimento de uma API utilizando ASP.NET Core Web API, aplicando princípios de arquitetura de software, design patterns, técnicas de documentação, testes e integração com banco de dados.

## Requisitos

- O grupo deve criar os seguintes artefatos:
    - **(até 15 pontos)** - Implementar a integração com um serviço externo (ex: autenticação, pagamento) utilizando RESTful APIs.
    - **(até 15 pontos)** - Implementar testes unitários, de integração e de sistema abrangentes utilizando xUnit.
    - **(até 20 pontos)** - Aplicar práticas de Clean Code, SOLID. Explique a utilização.
- 

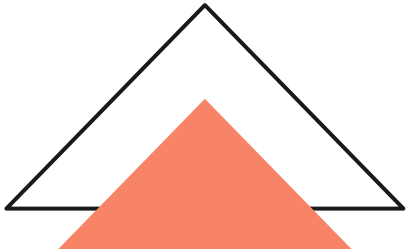


## ADVANCED BUSINESS DEVELOPMENT WITH .NET (2/2)

### Requisitos (continuação)

- O grupo deve criar os seguintes artefatos:
  - **(até 50 pontos)** - Integrar ML.NET na API, criar modelos simples e implementar funcionalidades de IA generativa que agregue valor à API (ex: recomendação de produtos, análise de sentimento).

### Entrega

- O grupo deve:
    - Entregar o repositório público no **GitHub** com o código da API.
    - Atualizar o **README.md** com detalhes sobre os testes implementados, práticas de Clean Code aplicadas e funcionalidades de IA generativa adicionadas.
- 



# DISRUPTIVE ARCHITECTURES: IOT, IOB & GENERATIVE IA

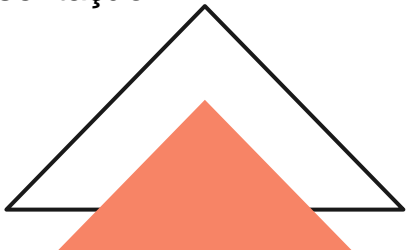
## **Objetivo Principal**

Apresentação final do projeto e autocrítica (reflexões).

## **Objetivos Específicos**

- Demonstração da versão final do projeto.
- Autocrítica (Reflexão) sobre o processo de desenvolvimento: o que funcionou, o que não funcionou, o que foi aprendido e o que fariam de diferente.
- Planos para o futuro do projeto: ideias para expansão ou melhoria

## **Observações:**

- É FUNDAMENTAL que nessa etapa sejam demonstradas as evoluções em relação à entrega anterior.
  - O vídeo deverá ser gravado por um (ou mais) integrantes do grupo. Utilizem apresentações e imagens, mas não utilizem ferramentas geradoras de apresentação com IA.
- 



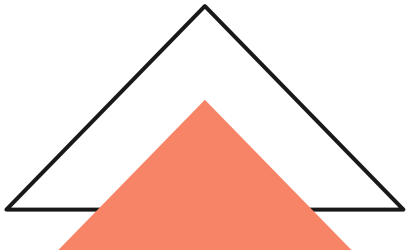
# DISRUPTIVE ARCHITECTURES: IOT, IOB & GENERATIVE IA

## Requisitos

- **[até 50 pontos]** Reflexão profunda e significativa sobre o processo de desenvolvimento e apresentação do projeto final.
- **[até 25 pontos]** Documento com a descrição final do projeto. (.docx, .ppt, .pdf ou README.md).
- **[até 25 pontos]** Organização e estrutura da documentação do projeto final no GitHub.

## Entrega

Arquivos entregáveis (**todos obrigatórios**):

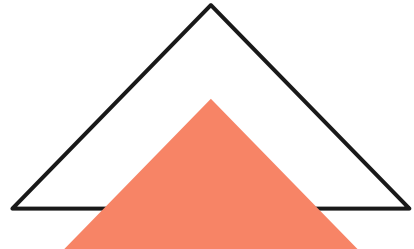
- Link ou arquivo do novo vídeo com a apresentação, conforme objetivos acima.
  - Documento com a descrição final do projeto. (.docx, .ppt, .pdf ou README.md). Formate o documento e evite usar IAs generativas para a criação.
  - Link para a documentação organizada no GitHub.
- 



# DISRUPTIVE ARCHITECTURES: IOT, IOB & GENERATIVE IA

## Condições de entrega

- A integridade e o conteúdo do arquivo entregue são de responsabilidade dos integrantes do grupo. Arquivos entregues sem conteúdo ou com arquivos corrompidos não serão considerados.
- Não serão aceitos arquivos enviados pelo Teams ou fora do prazo.





# JAVA ADVANCED (1/2)

## Requisitos

- **Nesta 4ª Sprint, a equipe deve entregar o projeto finalizado (aplicação full MVC).**
- O grupo deve produzir os seguintes artefatos:
  - **Código fonte do projeto (70 pts)** contemplando os seguintes tópicos:
    - Funcionalidades de autenticação através da implementação do Spring Security (com gestão de perfis de segurança);
    - Implementação de recursos de internacionalização;
    - Configuração de recursos de mensageria (produtores e consumidores);
    - Implementação de monitoramento com o Spring Boot Actuator;
    - Implementação de recursos de inteligência artificial utilizando o framework Spring AI.





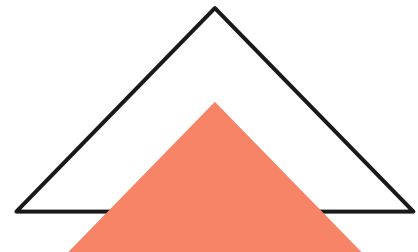
## JAVA ADVANCED (2/2)

### Requisitos (continuação)

- **Vídeo com a apresentação da aplicação funcionando com duração máxima de 10 minutos (20 pts)**
  - O vídeo deve abordar os seguintes tópicos:
    - Objetivo, Arquitetura final da solução, Funcionamento da aplicação, Conclusão (resumo de tudo o que fizeram, erros e acertos ao longo da jornada, e projetos futuros).
- **Viabilidade da solução (10 pts)**
  - Pertinência da solução entregue para atender a necessidade do cliente.

### Entrega

- A entrega deve ser feita via repositório público do GitHub com ReadMe completo.







# MOBILE APP DEVELOPMENT

- **Requisitos:**

- Evolua o App para conter telas completas de CRUD (40 pontos);
- Integre o CRUD com API (40 pontos);
- Implemente funções usando async storage (20 pontos).

- **Entrega:**

- Entregar o caminho do repositório do git contendo no readme.md as informações de nome do grupo, explanação do projeto e demais informações que ampliem o entendimento do projeto

