

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

 PRIVIAN
Simple Solutions

 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.



Regras Básicas



Regras Básicas

- Máximo de **3 integrantes**.
- **Não é permitido** o desenvolvimento individual do projeto;
- **Os grupos** devem ser formados **apenas pelos alunos das salas do 2TDS-Turmas de agosto e apenas do presencial, caso mude a modalidade para o ON, não poderá mais participar desse projeto.**
- Você deverá entrar na aba geral do teams, **baixar o passo a passo de como cadastrar o grupo no portal do challenge e o líder, cadastrar o grupo.**

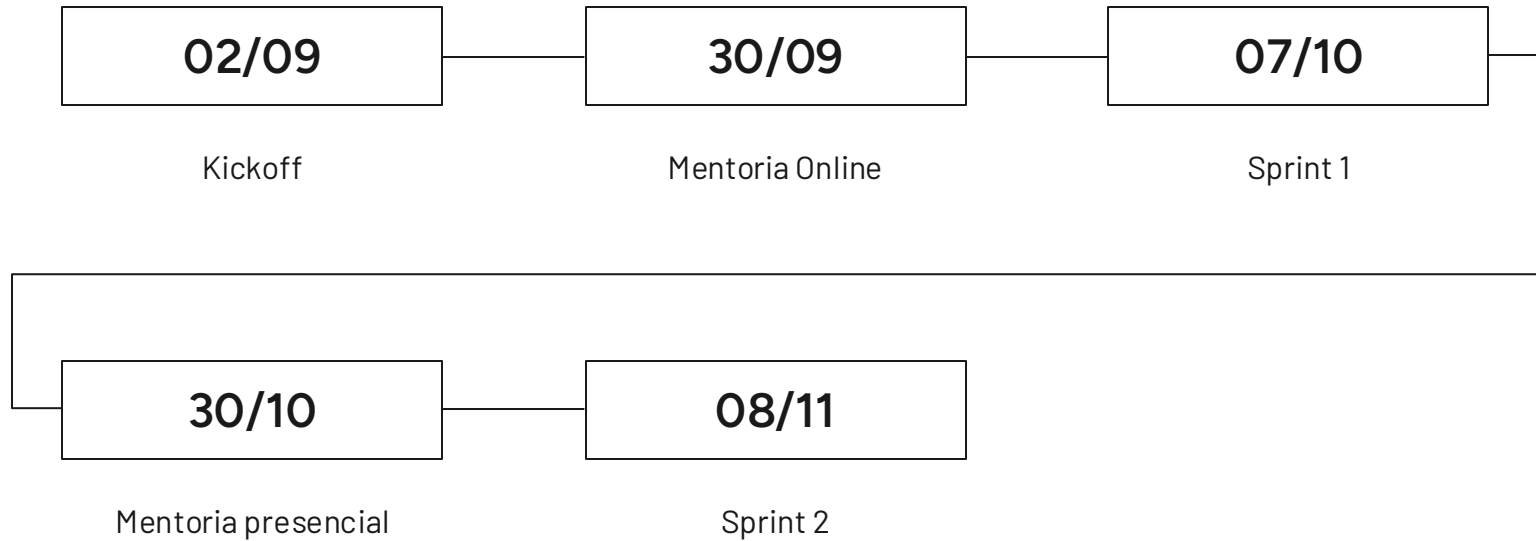
Gestão do grupo

- 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 2024 e 1º. Semestre de 2025.**
- Haverá um total de 4 sprints (4 entregas ao longo do segundo ano da graduação), sendo duas no segundo semestre de 2024 e duas no primeiro semestre de 2025;
- **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;**
- Entrega de todas as disciplinas serão realizadas através do **portal FIAP.**

Linha do tempo



*As datas das mentorias e horários serão confirmadas uma semana antes, pois dependemos da empresa.

Cronograma 1º Semestre

Data	Evento	Stakeholder
02/set	Kickoff Odontoprev – Presencial, Unidade Paulista (Evento único -Paulista 19h)	Odontoprev
15/set	Explicação dos entregáveis em sala	Professores
30/set	Mentoria Online	Odontoprev
07/out	Entrega da Sprint 1	Alunos
18/out	Prazo limite para postagem das notas da Sprint 1	Professores
26/out	NEXT	FIAP
30/out	Mentoria Presencial	Odontoprev
08/nov	Entrega da Sprint 2	Alunos
18/nov	Prazo limite para postagem das notas da Sprint 2	Professores

Encontros com a Odontoprev

- O Kickoff é o momento onde é apresentado o desafio, deve ser aproveitado para tirar dúvidas, entender a empresa, cenários pretendidos e qualquer outra dúvida que somente a Odontoprev poderá ajudar.
- A mentoria online (Setembro) tem como objetivo avaliar se as soluções apresentadas realmente estão alinhadas com o objetivo do challenge. Neste momento, o grupo deve ter um pitch com uma apresentação da sua solução, com ideia, problemática, aplicabilidade etc **(Caso não apresente o pitch irá perder 10 de 100 pontos em todas as matérias).**
- A mentoria presencial (Novembro) acontecerá na Paulista no período noturno, o grupo deverá apresentar a solução em desenvolvimento, já com uma visão preliminar do objetivo.



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**, tirando possíveis dúvidas com os professores em sala de aula.
- Acompanhar o que está sendo desenvolvido em sala x Entregável.
- Entender os critérios de avaliação do seu professor.

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;
- Disponibilizar a correção em até 15 dias corridos a contar do próximo dia após a data limite da entrega de cada Sprint.

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** atualizado;
- **Agendar os eventos** com a empresa parceira/professores, como apresentações, reuniões e treinamentos;
-
- SCRUM MASTER – Antonio Sergio Rodrigues Figueiredo e Karina Paltrinieri

Premiação NEXT



1º lugar

Shape + medalha + camiseta para cada aluno + 5k para o grupo



2º lugar

Shape + medalha + camiseta para cada aluno + 3k para o grupo



3º lugar

Shape + medalha + camiseta para cada aluno + 2k para o grupo



Mentorias

- Caso o grupo não participe de uma mentoria, o grupo será penalizado em 10 pontos em todas as disciplinas (pontuação de 0 a 100);
- Serão duas mentorias neste semestre, se o grupo não participar das duas, será descontado 20 pontos em todas as disciplinas, se não participar de uma, será descontado 10 pontos.
- Não é obrigatória a participação de todos os integrantes do grupo para a mentoria.
- O desconto será para todos do grupo, não é individual, ou seja, se pelo menos um integrante participar da mentoria, o grupo não terá a nota descontada.
- Caso a apresentação para a empresa esteja significativamente abaixo das expectativas, indicando falta de preparação adequada por parte do grupo, poderá haver um desconto na nota.

Entregáveis



Sprint 1



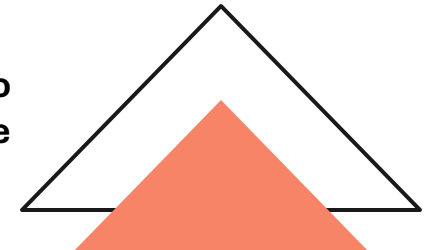


COMPLIANCE & QUALITY ASSURANCE

Desenvolver a declaração de visão e escopo do projeto e vídeo PITCH, contendo:

- Descrição do problema a resolver **(10 pontos)**;
- Descrição dos objetivos da solução idealizada **(10 pontos)**;
- Definição do público alvo que comprará (cliente/pagante) e/ou usará a solução (consumidor) **(10 pontos)**;
- Estudo de produtos semelhantes já existentes no mercado (que solucionam o mesmo problema, mesmo que de forma diferente, parcial ou totalmente) **(10 pontos)**;
- Avaliação do potencial de mercado (fatia de mercado a conquistar) **(10 pontos)**.
- Criação de vídeo pitch (máximo: 7 minutos) com a apresentação macro da ideia do projeto (MVP). O que esperamos do MVP, as telas da aplicação seja em FrontEnd ou Mobile detalhando a solução, caso não tenham as tela construídas, podem utilizar o Figma em substituição. **(50 pontos)**.

Gere um PDF com o material desenvolvido incluindo o link do vídeo compartilhado no Youtube, e suba a sua resposta da challenge sprint - entregas fora do formato perderão 10%.



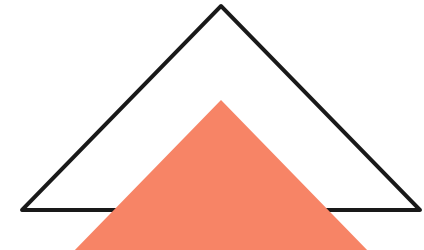


MASTERING RELATIONAL AND NON RELATIONAL DATABASE

1. Modelo Conceitual e Documentação (10 pontos):

- Apresentar o modelo conceitual do projeto, acompanhado de uma documentação que descreva as entidades, atributos e relacionamentos. A documentação deverá explicar a lógica e as decisões tomadas durante o processo de modelagem do banco de dados.

2. Desenvolvimento do DER, MER e Implementação no Banco de Dados (30 pontos):

- Elaborar o Diagrama Entidade-Relacionamento (DER) e o Modelo Entidade-Relacionamento (MER), seguindo rigorosamente a notação de Barker;
 - Garantir que o DER atenda, no mínimo, aos requisitos da 3ª Forma Normal (3FN), respeitando as cardinalidades estabelecidas;
 - Implementar o modelo no banco de dados, criando todos os objetos necessários para assegurar a funcionalidade do sistema.
- 

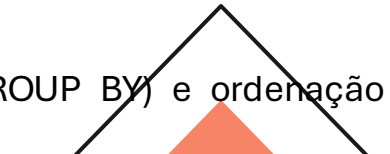


MASTERING RELATIONAL AND NON RELATIONAL DATABASE

3. Inserção de Dados Utilizando Python (20 pontos):

- Utilizar as bibliotecas do Python *cx_Oracle* ou *oracledb* para inserir no mínimo 10 registros em cada tabela criada;
- Assegurar que os dados inseridos estejam em conformidade com as especificações do modelo conceitual e relacional;
- Gravar um vídeo realizando essas operações e disponibiliza-lo em algum *streaming*, e o link deve ser enviado junto aos itens da entrega.


4. Blocos Anônimos com Consultas (20 pontos):

- Desenvolver 3(três) blocos anônimos para exibir os dados inseridos, implementando ao menos três consultas que utilizem diferentes tipos de junções: um INNER JOIN, um LEFT JOIN e um RIGHT JOIN;
 - Cada consulta deve também realizar operações de agrupamento (GROUP BY) e ordenação (ORDER BY), apresentando os resultados de forma clara e estruturada.
- 



MASTERING RELATIONAL AND NON RELATIONAL DATABASE

5. Estruturas de Decisão e Operações DML em Blocos Anônimos (20 pontos):

- Criar 2 (dois) blocos anônimos com estruturas de decisão, sendo um para realizar uma operação de UPDATE em uma tabela e outro para realizar um DELETE em outra tabela;
 - Os valores para a cláusula WHERE devem ser passados por variáveis dentro dos blocos, garantindo flexibilidade e controle sobre as operações realizadas.
-
- ✓ **Entregável** - Um arquivo zipado contendo:
 - ✓ O *.pdf* do Projeto de Banco de Dados Relacional contendo descritivo do projeto, o modelo lógico e físico, além do link do vídeo;
 - ✓ O arquivo *.sql* com os *scripts* separados por SQL e PL/SQL .
 - ✓ **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.
- 

1) A partir da solução identificada pelo seu grupo, justifique como a técnica de Virtualização (Máquinas Virtuais e/ou Docker) pode contribuir com a entrega desse importante projeto.

O que será avaliado:

a) A Organização e a estrutura do material gerado; **(20 pontos)**

Se todo o conhecimento adquirido na teoria e prática sobre Virtualização, ou seja Tipo de **Nuvem**, Tipo de **serviços** de Computação IaaS estão sendo utilizados no material entregue;

b) Justificar qual tipo de Virtualização será adotado na solução **(20 pontos)**





DEVOPS TOOLS E CLOUD COMPUTING 2 | 3

2) Desenhe a arquitetura **macro** de sua solução com o uso desses recursos em nuvem

O que será avaliado:

- A Organização e a estrutura do material gerado **(20 Pontos)**
- Na virtualização do desenho da arquitetura apresentado será avaliado o desenho do fluxo, imagens dos recursos utilizados e a utilização de legendas ou rótulos **(40 Pontos)**

Sugestões de Ferramentas:

- **Ferramenta de diagrama do Azure online**

<https://online.visual-paradigm.com/pt/diagrams/solutions/free-azure-diagram-tool/>

- **Ferramenta de diagrama de arquitetura online da AWS:**

<https://online.visual-paradigm.com/pt/diagrams/features/aws-architecture-diagram-tool/>



DEVOPS TOOLS E CLOUD COMPUTING 3 | 3

- **Oracle Cloud Infrastructure Diagram Software**

<https://www.visual-paradigm.com/features/oracle-cloud-infrastructure-diagram-tool/>

- **Software de diagrama de arquitetura de nuvem da IBM**

<https://online.visual-paradigm.com/pt/diagrams/solutions/free-ibm-cloud-diagram-tool/>

- **Software de design do Google Cloud Platform**

<https://online.visual-paradigm.com/pt/diagrams/solutions/free-google-cloud-diagram-tool/>

- **Software de desenho gráfico multiplataforma**

<https://app.diagrams.net/> - Draw.io

A entrega deve ser efetuada em um arquivo PDF

nomeGrupo_Challenge_1Sem_2TDS.PDF

Obs: A pontuação pode variar de acordo com a qualidade da entrega, ou seja, se o aluno entregar algum entregável com um baixo nível de apresentação poderá sofrer descontos na pontuação.





ADVANCED BUSINESS DEVELOPMENT WITH .NET

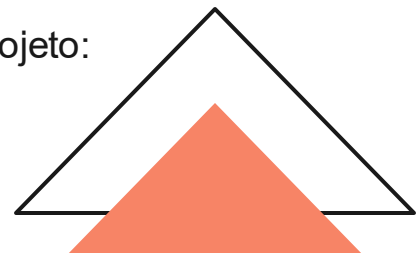
1. Projeto Github com arquivo README (30 pts total)

Definição do Projeto

- **Objetivo do Projeto:** Esclarecer o problema que o projeto pretende resolver.
- **Escopo:** Delimitar o que será desenvolvido e as funcionalidades principais.
- **Requisitos Funcionais e Não Funcionais:** Identificar os requisitos que a aplicação deve cumprir.

Desenho da arquitetura

- **Clean Architecture:** Considerar o uso de uma arquitetura limpa para separar responsabilidades e manter o código desacoplado.
- **Camadas da Aplicação:**
- **Apresentação:** As pastas e camadas que vão separar a aplicação e justificativa do projeto:
 - **Aplicação:** Serviços e casos de uso da aplicação.
 - **Domínio:** Modelos e regras de negócio.
 - **Infraestrutura:** Acesso a dados, integração com outras APIs, etc.





ADVANCED BUSINESS DEVELOPMENT WITH .NET

1. Projeto .NET (Camada Application) (70 pts total)

1. Criação da Solução .NET

2. Implementação do Domínio (20 pts)

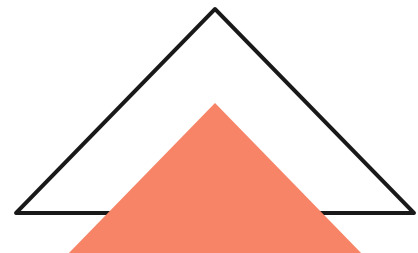
- **Criação das Entidades:** Implementar as entidades principais que representam os conceitos de negócio.
- **Interfaces de Repositório:** Definir interfaces para o acesso aos dados, seguindo o padrão de repositório.
- **Regras de Negócio:** Implementar lógica de negócios dentro das entidades ou em serviços específicos.

3. Implementação da Camada de Aplicação (20 pts)

- **Serviços de Aplicação:** Criar classes que implementam os casos de uso do sistema.
- **Manipulação de Erros:** Adicionar tratamento de exceções e retorno de erros apropriados.
- **DTOs (Data Transfer Objects):** Definir e implementar os DTOs para comunicação entre as camadas e expor via API.

4. Implementação da Camada de Infraestrutura (30 pts)

- **Mapeamento de Entidades (Entity Framework Core):** Configurar o EF Core para mapear as entidades para tabelas do banco de dados.
- **Repositórios Concretos:** Implementar os repositórios com métodos CRUD usando EF Core
- **Integrações Externas:** Implementar clientes HTTP para consumir APIs externas, se necessário.
- **Migrações de Banco de Dados:** Configurar e aplicar migrações para manter o esquema do banco de dados sincronizado com as entidades.





DISRUPTIVE ARCHITECTURES: IOT, IOB & GENERATIVE IA

Objetivos Principais:

- Criação de vídeo *pitch* (máximo: 3 minutos) com a apresentação macro da ideia do projeto.
- Criação de um documento de referência com a descrição da ideia do projeto.

Objetivos Específicos:

- Apresentar o tema escolhido pelo grupo.
- Apresentar a explicação do problema.
- Apresentar as alternativas de solução.
- Descrever os possíveis frameworks/bibliotecas Python que serão utilizados (seja específico, quais e por quais razões).
- Descrever de que forma os conceitos / técnicas de Machine Learning / IA poderão ser utilizados no desenvolvimento do projeto (seja específico neste requisito).

Observação: Não explique os conceitos e como os frameworks funcionam, e sim como poderão ser utilizados no projeto.



DISRUPTIVE ARCHITECTURES: IOT, IOB & GENERATIVE IA

Requisitos

- **(até 40 pontos)** Abrangência, profundidade e coerência para a aplicação de conceitos de Machine Learning / IA com apresentação das possíveis ferramentas / *frameworks* a serem aplicados na solução.
- **(até 30 pontos)** Apresentação do vídeo de forma clara.
- **(até 30 pontos)** Organização e estrutura do material gerado.

Entrega

Arquivos entregáveis **(todos obrigatórios)**

- Arquivo .zip com o link do vídeo carregado no YouTube
- Arquivo de referência, em formato PDF, conforme objetivos relacionados.

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.

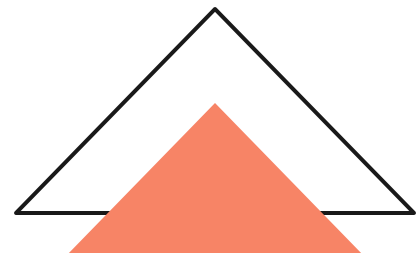


JAVA ADVANCED (1/5)

- Descrição geral:

Produza pelo menos um dos domínios da sua solução tecnológica proposta utilizando conceitos de Java e Spring.

A solução deverá ser capaz de persistir, manter e consultar os dados em um SGBD Relacional (preferencialmente Oracle). Os dados gerados por meio da solução de TI proposta pela equipe será primordial para atender as solicitações do cliente com propostas que ajudem a Odontoprev na redução de custos e aumento da eficiência, além de melhorar a experiência dos pacientes e garantir a sustentabilidade do negócio.



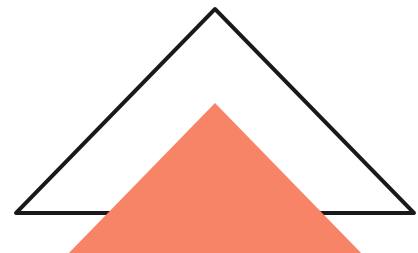


JAVA ADVANCED (2/5)

- Solicitação:

Dentro do contexto apresentado, a equipe deverá:

- ☐ Criar uma aplicação Java com Spring Boot que ajude a resolver um problema dentro do contexto apresentado (utilize a criatividade);
- ☐ A aplicação deverá respeitar os fundamentos da Programação Orientada a Objetos (POO) e as classes de entidade deverão estar relacionadas e mapeadas com JPA;
- ☐ Conter o código que respeite a coesão e o desacoplamento;
- ☐ Respeitar os padrões DTO e utilizar a Bean Validation;
- ☐ Respeitar os conceitos fundamentais do REST (RESTful);



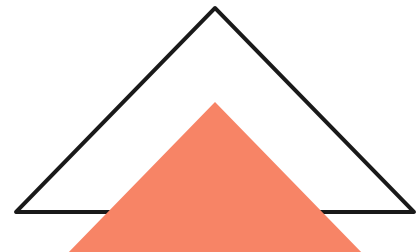


JAVA ADVANCED (3/5)

- Solicitação:

Dentro do contexto apresentado, a equipe deverá:

- ☐ Na primeira Sprint, a API deverá estar de acordo com o modelo de maturidade nível 1;
- ☐ Utilizar os Design Patterns como Singleton, Repository com Generics quando necessário;
- ☐ Poderá utilizar a JPQL e/ou Spring JPA Query Methods.
- ☐ Criar uma Gestão de Configuração dos artefatos de software que deverá ser demonstrada pela equipe, portanto, todos os artefatos produzidos deverão estar no Github.
Os professores deverão ter acesso ao repositório.
- ☐ Apresentar os testes dos endpoints da API (utilize o Postman ou Insomnia).
Exporte as requisições para que o professor possa validar o funcionamento.

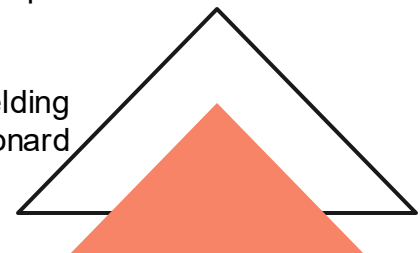




JAVA ADVANCED (4/5)

- Distribuição da pontuação:

- a) **(até 5 Pontos)** Apresentar o cronograma de desenvolvimento e respeitar os prazos. Crie um o documento dizendo quem fará o que e quando a atividade deverá ser ou foi realizada;
- b) **(até 10 Pontos)** Imagens explicativas da arquitetura, definição das classes de domínio da aplicação e o respectivo Diagrama de Classes de Entidade. O Diagrama de Entidade e Relacionamento (DER) e o Diagrama de Classe das Entidades deverão ser coerentes. Se necessário faça uma breve explicação sobre os relacionamentos e as constraints envolvidas;
- c) **(até 40 Pontos)** Implementação das classes de Entidade necessárias para solução do problema que se propõe resolver. Atenção ao correto encapsulamento, à tipagem dos atributos e ao Mapeamento Objeto Relacional com JPA e Hibernate (A equipe deverá demonstrar a cada Sprint que houve evolução em relação à sprint anterior);
- d) **(até 15 Pontos)** A aplicação respeita os conceitos fundamentais do REST criados por Roy Fielding (RESTful) e API está de acordo com o modelo de maturidade nível 1 proposto por Leonard Richardson;





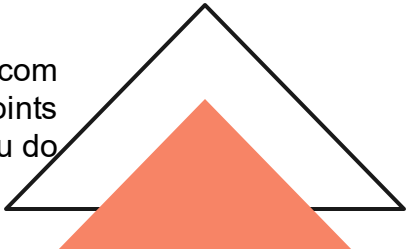
JAVA ADVANCED (5/5)

- Distribuição da pontuação:

e) **(até 10 Pontos)** A Gestão de Configuração dos Artefatos de Software foram demonstrados pela equipe. Ou seja, todos os artefatos produzidos estão no Github e os professores têm acesso ao repositório;

f) **(até 10 pontos)** Envio do link para o projeto público compartilhado no Github contendo entre outros: 1) Toda documentação; 2) Código-fonte; 3) Arquivo README.md com texto contendo: 3.1) Nome da aplicação; 3.2) Nome completo e breve apresentação dos integrantes do Grupo (atividade da qual ficou responsável no projeto); 3.3) instrução de como rodar a aplicação; 3.4) imagem dos diagramas; 3.5) link para vídeo apresentando a Proposta Tecnológica, o público-alvo da aplicação e os problemas que a aplicação se propõe a solucionar; 3.6) Listagem de todos os endpoints (documentação da API);

g) **(até 10 pontos)** Se a equipe demonstrou preocupação em testar a aplicação e provou com documentos. Disponibilizou arquivos para que o professor seja capaz de realizar teste dos endpoints e houve, após os testes, a perfeita persistência e recuperação dos dados. (exporte do Postman ou do Insomnia as requisições aos endpoints). Inclua na pasta documentos do projeto.

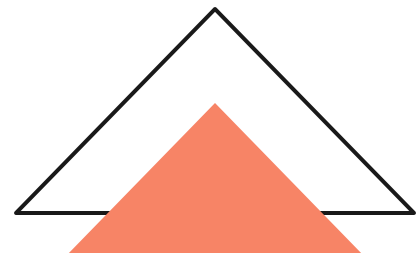




MOBILE APP DEVELOPMENT (1/2)

Descrição:

- Wireframes de uma solução mobile com no mínimo 5 telas, acompanhado de explicações detalhadas para cada tela. **(50 pontos)**
- Protótipo com uma tela implementada (Uma Activity) contendo pelo menos: **(30 pontos)**
 - 1 Componente de saída de informação (TextView, Image, Etc)
 - 1 Componente de entrada de informação (EditText, Spinner, ListBox, etc)
 - 1 Componente ativo de ação de evento (Botão, etc)
- Incluir cenários possíveis de sucesso e erro, no mínimo um de cada. **(20 pontos)**





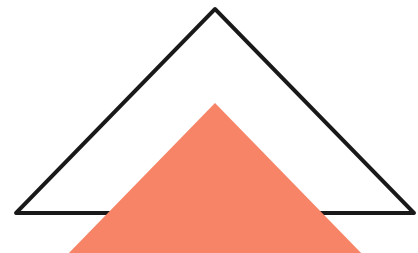
MOBILE APP DEVELOPMENT (2/2)

Formato de Entrega

- Documento do Word com imagens e explicações, contendo o nome completo e RM de todos os integrantes do grupo na capa.
- Repositório no GitHub com o código do aplicativo.
- Um vídeo do protótipo funcionando.
- O APK para instalação.

Dicas de ferramentas

- Software de prototipação como Figma, Protopie, Photoshop, Miro e etc;



Sprint 2





COMPLIANCE & QUALITY ASSURANCE

Faça o desenho da Arquitetura da solução, aplicando TOGAF e usando a ferramenta Archi.

Desenhe em uma mesma página de diagrama no Archi:

- Visão da arquitetura **(25 pontos);**
- Arquitetura de negócio **(25 pontos);**
- Arquitetura de sistemas **(25 pontos);**
- Arquitetura de tecnologia **(25 pontos);**

Anexar na entrega um arquivo ZIP contendo um PDF com a imagem do seu desenho e um arquivo TXT contendo nome e RM dos integrantes.



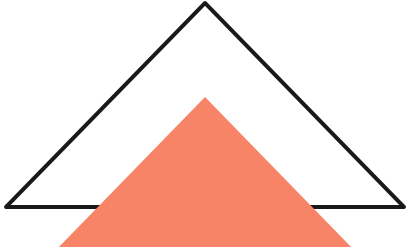


MASTERING RELATIONAL AND NON RELATIONAL DATABASE

1. Aplicação das Correções no Modelo Anterior (5 pontos):

- Implementar as correções sugeridas pelo professor no modelo apresentado na entrega anterior, garantindo que todas as observações sejam integralmente atendidas.

2. Funções de Validação de Entrada de Dados (10 pontos):

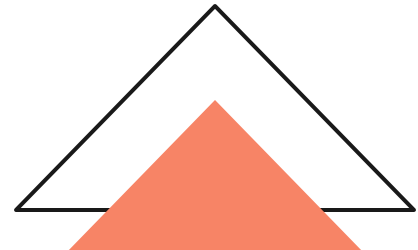
- Desenvolver duas funções para validar a entrada de dados, assegurando que as informações inseridas estejam em conformidade com as regras de integridade e os requisitos específicos do projeto.
- 



MASTERING RELATIONAL AND NON RELATIONAL DATABASE

3. Procedures para Operações CRUD (20 pontos):

- Criar procedures para realizar operações de INSERT, UPDATE e DELETE em cada tabela criada no sprint anterior, automatizando e padronizando a manipulação de dados.
- Garantir que as funções criadas no item 2 sejam utilizadas nas operações de INSERT e UPDATE para validar os dados antes de serem inseridos ou atualizados.
- Incluir tratamento de erros e exceções em todas as procedures para assegurar robustez e confiabilidade nas operações.



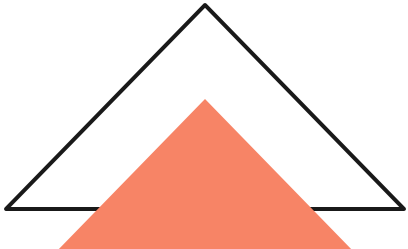


MASTERING RELATIONAL AND NON RELATIONAL DATABASE

4. Execução de Procedures via Aplicação Java com Demonstração em Vídeo (25 pontos):

- Utilizar a aplicação backend desenvolvida na disciplina de Java para executar as procedures criadas, aplicando-as em pelo menos duas tabelas. Em cada tabela, realizar 2 operações de INSERT, 2 de UPDATE e 2 de DELETE.
- Gravar um vídeo demonstrando a execução dessas operações e anexar o link do vídeo na entrega.

5. Função com Cursor e Joins para Relatório Formatado (20 pontos):

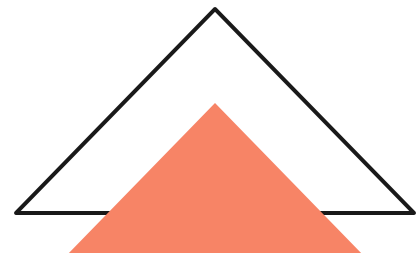
- Criar uma função com retorno do tipo tabela(**table**) para relatório, que utilize um cursor e realize JOINS entre pelo menos duas tabelas do projeto, gerando um relatório formatado com os dados obtidos.
- 



MASTERING RELATIONAL AND NON RELATIONAL DATABASE

6. Função para Relatório com Regra de Negócio (20 pontos):

- Desenvolver uma função do que retorne um tipo tabela que gere um relatório baseado em uma regra de negócio específica, incorporando funções, INNER JOIN, ORDER BY, e agregações como SUM ou COUNT, apresentando os resultados de forma detalhada e organizada.
- ✓ **Entregável:** arquivo zip contendo o pdf com prints de execução dos scripts, link do vídeo e arquivo .sql com a criação dos objetos solicitados.
- ✓ **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.




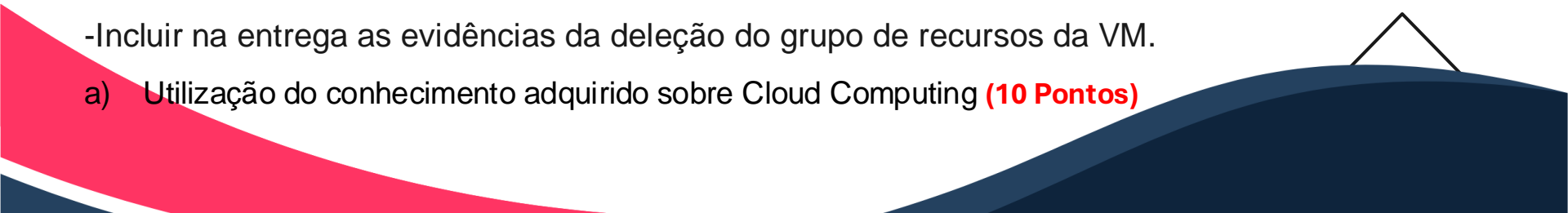


DEVOPS TOOLS E CLOUD COMPUTING 1 | 4

De acordo com a arquitetura e tecnologia da sua solução proposta, é hora de colocarmos em prática o que foi aprendido. Vocês devem provisionar os recursos necessários no **provedor de Nuvem**, realizar o deployment de máquinas virtuais, e utilizar **Dockerfile** e Docker **Compose**.

1) No provedor de Nuvem, provisione máquina virtual (Linux ou Windows) e Assegure-se de que todas as dependências necessárias estão configuradas, incluindo o monitoramento da saúde da VM, desempenho e dependências de rede.

O que será avaliado:

- a) A Organização e a estrutura do material gerado; **(20 Pontos)**
 - b) Deployment de Máquinas Virtuais; **(20 Pontos)**
- Incluir na entrega as evidências da deleção do grupo de recursos da VM.
- a) Utilização do conhecimento adquirido sobre Cloud Computing **(10 Pontos)**
- 
- 

DEVOPS TOOLS E CLOUD COMPUTING 2 | 4

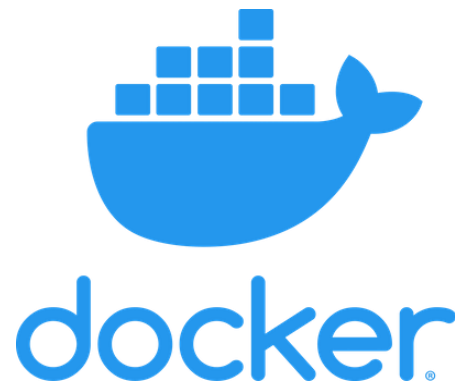
2) Após a criação das máquinas virtuais, instale o Docker. Utilize **Dockerfile** e **Docker Compose** para realizar o deployment de imagens Docker. É permitido utilizar qualquer recurso de IaaS.

Requisitos:

- O projeto deve ser executado em background.
- Utilize imagens Docker que ofereçam melhor desempenho, como aquelas com `slim` no nome ou baseadas em Alpine Linux.
- Defina um diretório padrão para o projeto.
- O aplicativo deve ser executado por um usuário que não possua privilégios administrativos.

O que será avaliado:

- A Organização e a estrutura do material gerado **(20 Pontos)**
- Instalação e uso do Docker, incluindo o deployment de imagens **(20 Pontos)**
- Utilização do conhecimento adquirido sobre Cloud Computing **(10 Pontos)**

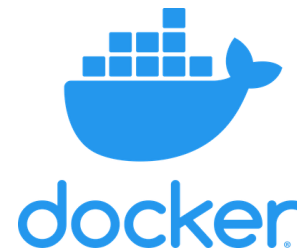


DEVOPS TOOLS E CLOUD COMPUTING 3 | 4

3) O código-fonte da aplicação deve estar disponível em um repositório no **GitHub**, incluindo um arquivo `README.md` com **instruções** claras para o deploy e testes. Inclua também o **Dockerfile** e o arquivo `.yml`.

O que será avaliado:

- a) Código-fonte e README.md; (item obrigatório, a falta desses quesitos será subtraído da nota)
- 4) Grave um vídeo demonstrando o funcionamento da solução, desde a instalação do Docker até a execução da aplicação. Inclua todos os detalhes da criação, implementação e testes, bem como a persistência de dados na Nuvem.

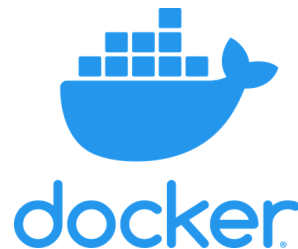


Entrega final:

Entregue um arquivo **PDF** contendo os **nomes** e **RMs** dos integrantes do time, link do repositório **GitHub** e link do vídeo no **YouTube**.

- Atenção: A ausência deste PDF resultará na subtração de pontos e possível impossibilidade de correção.

A pontuação de cada requisito será ajustada de acordo com a qualidade da tarefa **entregue**.





ADVANCED BUSINESS DEVELOPMENT WITH .NET

1. Atualização do README.md (10 pts total)

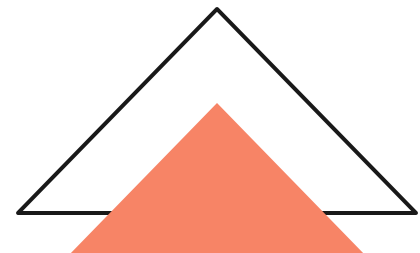
- Visão Geral do Projeto:
- Atualizar a descrição geral do projeto para refletir o progresso e as novas funcionalidades implementadas.
- Instruções de Instalação e Configuração

Implementação da Camada Web (ASP.NET Core) (90 pts total)

• Views e Layouts:

Configurar as rotas padrão para as páginas da aplicação. (10 pts)

- Implementar rotas personalizadas. (10 pts)
- Implementar o layout principal da aplicação, incluindo cabeçalho, rodapé e navegação customizado com bootstrap. (10 pts)
- Desenvolver as views correspondentes às principais funcionalidades da aplicação com validações correspondentes das viewmodels. (10 pts total)
- Criar ViewModels para transferir dados entre a camada de apresentação e a lógica de negócio. (10 pts)
- **Controllers:**
 - Implementar os controladores que manipulam as requisições HTTP e retornam as views ou redirecionam para outras ações e ações CRUD (Create, Read, Update, Delete) conforme necessário para a aplicação, as validações, padrões de projetos utilizados e precisão da criação do controllers conforme as boas praticas do framework precisam ser implementadas. (50 pts)





DISRUPTIVE ARCHITECTURES: IOT, IOB & GENERATIVE IA

Objetivo Principal:

Apresentação de uma versão Beta (simples) envolvendo as principais ideias e demonstrando os recursos / funcionalidades que serão utilizados no projeto.

Objetivos Específicos:

- Apresentar as diferenças entre a etapa atual do projeto e a proposta apresentada na primeira entrega.
- Apresentar, **com mais detalhes em relação à primeira entrega**, os frameworks/bibliotecas Python, APIs e demais recursos/ferramentas que serão (ou poderão ser) utilizados no desenvolvimento.
- Explicar, de forma breve, o funcionamento dos recursos/ferramentas (descritos acima) **dentro da aplicação** que está sendo desenvolvida.
- Descrever resumidamente de que forma os conceitos de Machine Learning / IA estão sendo (ou poderão ser) utilizados no desenvolvimento do projeto.
- Apresentar um repositório com a documentação gerada até o momento.

Observações:

- **Não explique os conceitos e como os frameworks funcionam, e sim como poderão ser utilizados no projeto.**
- **É FUNDAMENTAL** que nessa etapa já sejam demonstradas funcionalidades do projeto.



DISRUPTIVE ARCHITECTURES: IOT, IOB & GENERATIVE IA

Requisitos:

- **(até 40 pontos)** Abrangência, profundidade e coerência para a aplicação de conceitos de Machine Learning / IA com apresentação das possíveis ferramentas / frameworks a serem aplicados na solução.
- **(até 30 pontos)** Organização e a estrutura do material gerado.
- **(até 30 pontos)** Apresentação do vídeo de forma clara.

Entrega:

Arquivos entregáveis **(todos obrigatórios)**

Arquivo .zip com o link do repositório do projeto, contendo o link do novo vídeo e a documentação organizada, conforme descrito nos objetivos.

Condições de entrega:

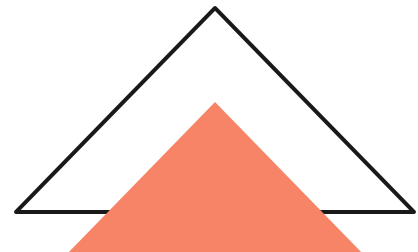
- A integridade e o conteúdo do arquivo entregue são de responsabilidade dos integrantes do grupo. Arquivos entregues sem conteúdo ou com arquivos corrompidos não serão considerados.
- Não serão aceitos arquivos enviados pelo Teams ou fora do prazo.



JAVA ADVANCED (1/5)

- Descrição geral:

- ☐ A equipe deverá demonstrar evolução entre a primeira e a entrega que está sendo realizada nesta segunda Sprint.
- ☐ A equipe deverá portanto aperfeiçoar a solução tecnológica fazendo as devidas refatorações de código visando a evolução da qualidade da aplicação ou a inclusão de novos módulos ou domínios da solução tecnológica proposta utilizando Java e Spring.
- ☐ Esta solução deverá continuar sendo capaz de persistir, alterar e consultar os dados em um SGBD Relacional (preferencialmente Oracle).

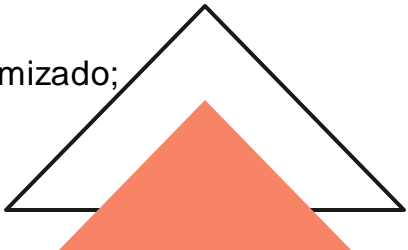




JAVA ADVANCED (2/5)

- Solicitação:

Dentro do contexto apresentado, a equipe deverá:

- ☐ Aperfeiçoar a aplicação Java com Spring Boot em desenvolvimento;
 - ☐ Evidenciar todos os tópicos já abordados na Sprint anterior;
 - ☐ Mostrar em detalhes a evolução do projeto desde a Sprint 1;
 - ☐ Na API, deverá acrescentar a parte relacionada com HATEOAS em nível de maturidade 3 proposto por Leonard Richardson;
 - ☐ Como uma opção, o grupo pode acrescentar o uso de Lombok para um código mais otimizado;
- 

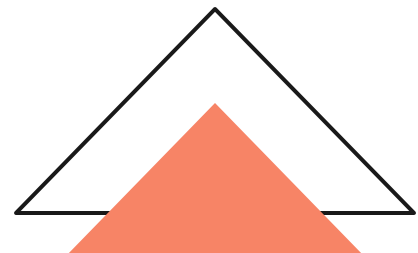


JAVA ADVANCED (3/5)

- Solicitação:

Dentro do contexto apresentado, a equipe deverá:


- ☐ Continuar usando uma Gestão de Configuração dos artefatos de software que deverá ser demonstrada pela equipe, portanto, todos os artefatos produzidos deverão estar no Github. Os professores deverão ter acesso ao repositório.
- ☐ Apresentar os novos testes dos endpoints da API (utilize o Postman ou Insomnia). Exporte as requisições para que o professor possa validar o funcionamento.





JAVA ADVANCED (4/5)

- Distribuição da pontuação:

- a) **(até 5 Pontos)** Apresentar o cronograma de desenvolvimento e respeitar os prazos. Crie um o documento dizendo quem fará o que e quando a atividade deverá ser ou foi realizada;
 - b) **(até 10 Pontos)** Imagens explicativas da arquitetura, definição das classes de domínio da aplicação e o respectivo Diagrama de Classes de Entidade. O Diagrama de Entidade e Relacionamento (DER) e o Diagrama de Classe das Entidades deverão ser coerentes. Se necessário faça uma breve explicação sobre os relacionamentos e as constraints envolvidas;
 - c) **(até 40 Pontos)** Implementação das classes de Entidade necessárias para solução do problema que se propõe resolver. Atenção ao correto encapsulamento, à tipagem dos atributos e ao Mapeamento Objeto Relacional com JPA e Hibernate (A equipe deverá demonstrar a cada Sprint que houve evolução em relação à sprint anterior);
 - d) **(até 15 Pontos)** A aplicação respeita os conceitos fundamentais do REST criados por Roy Fielding (RESTful) e API está de acordo com o modelo de maturidade nível 1 proposto por Leonard Richardson;
- 



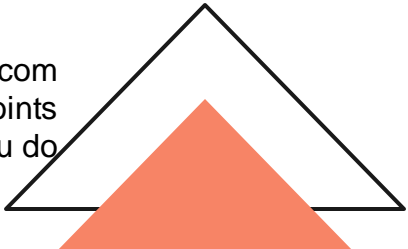
JAVA ADVANCED (5/5)

- Distribuição da pontuação:

e) **(até 10 Pontos)** A Gestão de Configuração dos Artefatos de Software foram demonstrados pela equipe. Ou seja, todos os artefatos produzidos estão no Github e os professores têm acesso ao repositório;

f) **(até 10 pontos)** Envio do link para o projeto público compartilhado no Github contendo entre outros: 1) Toda documentação; 2) Código-fonte; 3) Arquivo README.md com texto contendo: 3.1) Nome da aplicação; 3.2) Nome completo e breve apresentação dos integrantes do Grupo (atividade da qual ficou responsável no projeto); 3.3) instrução de como rodar a aplicação; 3.4) imagem dos diagramas; 3.5) link para vídeo apresentando a Proposta Tecnológica, o público-alvo da aplicação e os problemas que a aplicação se propõe a solucionar; 3.6) Listagem de todos os endpoints (documentação da API);

g) **(até 10 pontos)** Se a equipe demonstrou preocupação em testar a aplicação e provou com documentos. Disponibilizou arquivos para que o professor seja capaz de realizar teste dos endpoints e houve, após os testes, a perfeita persistência e recuperação dos dados. (exporte do Postman ou do Insomnia as requisições aos endpoints). Inclua na pasta documentos do projeto.

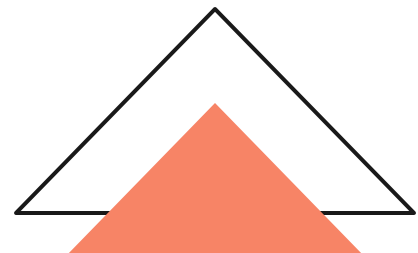




MOBILE APP DEVELOPMENT (1/2)

Descrição:

- Diagrama demonstrando a integração/comunicação do aplicativo com APIs/banco de dados e sistemas externos. **(20 pontos)**
- Programação visual de 5 telas do aplicativo funcional. **(50 pontos)**
- Realizar chamadas a uma API (pelo menos 3 telas fazendo requisições). **(20 pontos)**
- Layout adequado e com bom nível de usabilidade. **(10 pontos)**





MOBILE APP DEVELOPMENT (2/2)

Formato de Entrega

- Documento do Word com imagens e explicações, contendo o nome completo e RM de todos os integrantes do grupo na capa.
- Repositório no GitHub com o código do aplicativo.
- Um vídeo do aplicativo funcionando.
- O APK para instalação.

Dicas de ferramentas

- Android Studio, Activity, Fragments e Views no Android, elementos do JetPack compose
- 