FIAP GRADUAÇÃO

CHALLENGE 2025

2º ANO

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Turmas de Fevereiro

MOTTU

A Mottu é uma startup brasileira fundada em 2020 que oferece soluções de mobilidade por meio do aluguel de motos econômicas, visando atender principalmente entregadores e profissionais que necessitam de transporte ágil. Com presença em mais de 30 cidades no Brasil e no México, a empresa disponibiliza planos acessíveis a partir de R$ 18 por dia, incluindo benefícios como manutenção preventiva, proteção contra terceiros e roubo, além de suporte 24 horas. Recentemente, a Mottu expandiu suas operações para a venda de motocicletas, como a Mottu Sport 110i, reforçando seu compromisso em transformar realidades e gerar oportunidades para seus clientes.

EMPRESA PARCEIRA

DESAFIO

🎯**Desafio:** **Desafio de Mapeamento Inteligente do Pátio e Gestão das Motos.**

🧩 Introdução

A gestão de frotas de motos em pátios de múltiplas filiais é um desafio operacional crítico. Atualmente, a localização das motos dentro desses pátios pode ser controlada de maneira manual, o que gera imprecisões e impacta diretamente nossa eficiência operacional. Com mais de 100 filiais espalhadas por diferentes regiões, é fundamental aprimorar o processo de monitoramento e gestão das motos, garantindo uma operação ágil, segura e escalável. Para resolver essa questão, buscamos uma solução inovadora que se utilize de tecnologia para otimizar o mapeamento e o monitoramento das motos dentro dos pátios da Mottu, desde a identificação até a localização e gestão das unidades.

DESAFIO - OBJETIVOS

OBJETIVOS DO DESAFIO

🚧**Objetivos do Desafio:**

Os alunos devem desenvolver uma solução que permita mapear e monitorar de forma precisa e automatizada as motos dentro dos pátios das nossas filiais, utilizando ferramentas inovadoras. O sistema proposto deve ser capaz de capturar informações sobre a disposição das motos e apresentá-las em uma interface visual intuitiva e fácil de usar. A solução precisa ser capaz de:

- Identificar com precisão a localização das motos dentro dos pátios;
- Fornecer uma visualização em tempo real da disposição das motos;
- Ser escalável, de forma que possa ser implementada nas mais de 100 filiais tanto no Brasil como no México com diferentes layouts e tamanhos de pátios

O QUE É ESPERADO DA SOLUÇÃO

A SOLUÇÃO DEVE CONTER

A solução proposta deve incluir, mas não se limitar a, seguintes componentes:

Visão Computacional para Identificação das Motos:

- Utilizar tecnologias de visão computacional para capturar imagens ou vídeos do pátio e identificar as motos presentes.
- O sistema deve ser capaz de distinguir entre diferentes motos, identificar modelos e
- registrar sua posição no pátio.
- Implementação de técnicas de detecção de objetos, classificação e rastreamento para
- identificar as motos de forma eficaz.

O QUE É ESPERADO DA SOLUÇÃO

A SOLUÇÃO DEVE CONTER

Modelo de Mapeamento Digital do Pátio:

- Criar um modelo digital interativo do pátio que mostre, em tempo real, a disposição exata de cada moto.
- A visualização do pátio deve ser detalhada, permitindo aos operadores identificar facilmente a localização de cada moto.
- O sistema deve ser flexível, permitindo ajustes conforme a necessidade do espaço físico de cada filial.

O QUE É ESPERADO DA SOLUÇÃO

A SOLUÇÃO DEVE CONTER

Desenvolvimento de Interface Web ou App:

- Criar uma interface de usuário intuitiva, que pode ser acessada por operadores do pátio em desktops ou dispositivos móveis;
- A interface deve apresentar uma visão geral clara e em tempo real das motos dentro do pátio, permitindo que os operadores acessem informações sobre cada moto de forma prática;
- Funcionalidades adicionais, como alertas de proximidade ou de localização específica das motos, devem ser consideradas.

O QUE É ESPERADO DA SOLUÇÃO

A SOLUÇÃO DEVE CONTER

Integração com Sensores IoT das Motos:

- Implementar integração com os sensores de IoT instalados nas motos, de forma que a localização dos veículos possa ser atualizada automaticamente, caso os sensores permitam tal funcionalidade.
- Além disso, as informações do sensor devem ser utilizadas para fornecer dados adicionais, como o status da moto (por exemplo, ligado/desligado), condições de manutenção ou qualquer outra informação relevante.

O QUE É ESPERADO DA SOLUÇÃO

A SOLUÇÃO DEVE CONTER

Escalabilidade e Adaptabilidade

- A solução deve ser capaz de ser implantada em diferentes filiais, considerando variações no tamanho e formato dos pátios.
- A infraestrutura de backend deve ser preparada para lidar com grandes volumes de dados, mantendo o desempenho e a eficiência, independentemente da quantidade de motos mapeadas.

CRITÉRIOS PARA ELEGER O VENCEDOR

CRITÉRIOS DE AVALIAÇÃO DA MOTTU

Critérios de Avaliação:

Os projetos serão avaliados com base nos seguintes critérios pela Mottu para eleger o projeto vencedor do Challege:

Precisão do Mapeamento:

- Capacidade de identificar e mapear com precisão a localização de cada moto dentro do pátio.
- Precisão das tecnologias de visão computacional, especialmente em ambientes dinâmicos, com diferentes condições de luz e diferentes tipos de motos.

CRITÉRIOS DE AVALIAÇÃO DA MOTTU

Eficiência da Solução

- Tempo de resposta do sistema e velocidade no mapeamento em tempo real.
- Capacidade de escalar a solução para diversas filiais sem perda significativa de desempenho.
- Otimização do processo de monitoramento, com foco em minimizar os custos operacionais.

CRITÉRIOS DE AVALIAÇÃO DA MOTTU

Facilidade de Uso

- A interface deve ser intuitiva e acessível, de modo que os operadores possam realizar o monitoramento sem necessidade de treinamento extenso. A usabilidade será analisada com base na experiência do usuário final, buscando minimizar a complexidade das operações no pátio.

CRITÉRIOS DE AVALIAÇÃO DA MOTTU

CRITÉRIOS PARA ELEGER O VENCEDOR

Criatividade e Inovação

- Uso de tecnologias e abordagens inovadoras para resolver o problema.
- Criatividade na integração de sistemas de visão computacional, IoT e interface de usuário.
- Propostas de soluções que ofereçam um diferencial em relação ao uso tradicional de sistemas de monitoramento manual.

CRITÉRIOS PARA ELEGER O VENCEDOR

CRITÉRIOS DE AVALIAÇÃO DA MOTTU

Viabilidade de Implementação

- A solução deve ser factível de ser implementada no ambiente real de operações da Mottu.
- Considerações práticas sobre custos de implementação, manutenção e suporte da solução.
- A capacidade da solução de operar com uma base de dados grande, atendendo a múltiplas filiais simultaneamente.

REGRAS DO CHALLENGE

TIMES

COMPOSIÇÃO DOS GRUPOS

- Máximo de **3 integrantes**;
- **Não** é permitido o **desenvolvimento individual do projeto**;
- É recomendado que os grupos sejam **compostos preferencialmente por alunos da mesma turma**. No entanto, é permitido formar grupos com alunos de turmas diferentes, desde que estejam cientes e aceitem os possíveis inconvenientes, como horários de apresentação distintos (manhã/noite), diferentes professores, entre outros.
- **Não é permitido formar grupos com alunos de outros cursos ou anos**.

ENTREGAS

ENTREGAS DAS SPRINTS

- Os grupos devem **idealizar os projetos** a serem desenvolvidos ao longo do ano letivo, dividido em **entregas parciais durante o 1º. e 2º. semestre;**
- Haverá um total de **3 sprints (3 entregas ao longo do ano),** sendo uma no primeiro semestre e duas no segundo semestre;
- 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**;

ENTREGAS

- Entrega de cada **sprint será realizada por disciplina**;
- Cada disciplina disponibilizou os **requisitos no portal e nesta documentação**;
- Entrega de todas as disciplinas serão realizadas através do **portal FIAP**.

Entrega 1° Semestre
23/05/2025

ENTREGAS 1º SEMESTRE

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**.

RESPONSABILIDADES

PROFESSORES

RESPONSABILIDADES

- **Responsáveis por orientar o desenvolvimento** do projeto em suas disciplinas;
- 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;
- A **correção de cada entregável**, é de responsabilidade do professor da disciplina em cada turma;
- Disponibilizar **nota individual** referente à disciplina;
- Disponibilizar **justificativa de nota aplicada** ao grupo/aluno, quando se aplicar;

SCRUM MASTER

RESPONSABILIDADES

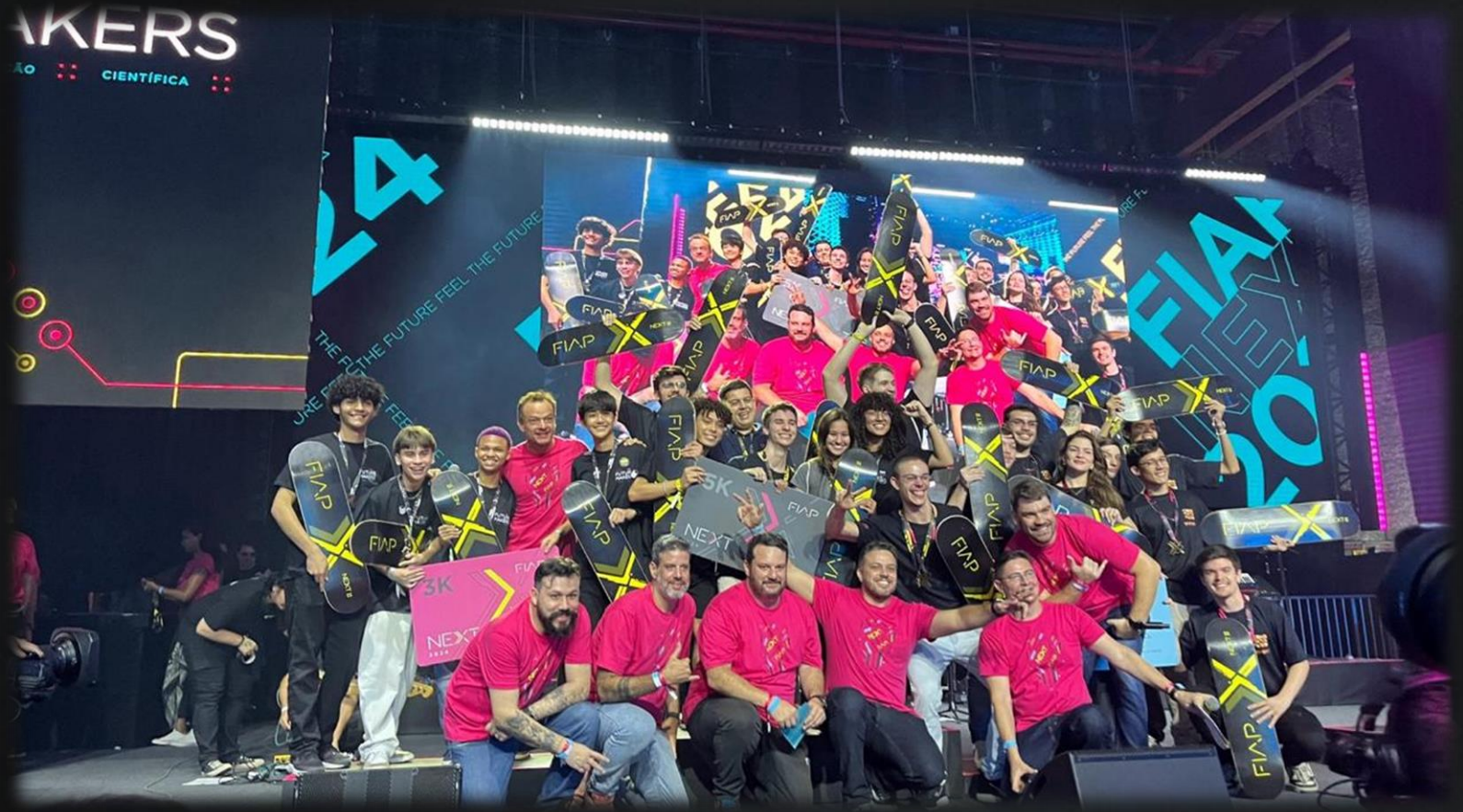
- **Comunicação entre os alunos/professores e a(s) empresa(s) parceira(s);**
- **Manter os dados dos grupos de alunos atualizada** (a gestão e formação dos grupos é de responsabilidade dos alunos);
- Manter o **documento do challenge** atualizada;
- **Agendar os eventos** com a empresa parceira/professores,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

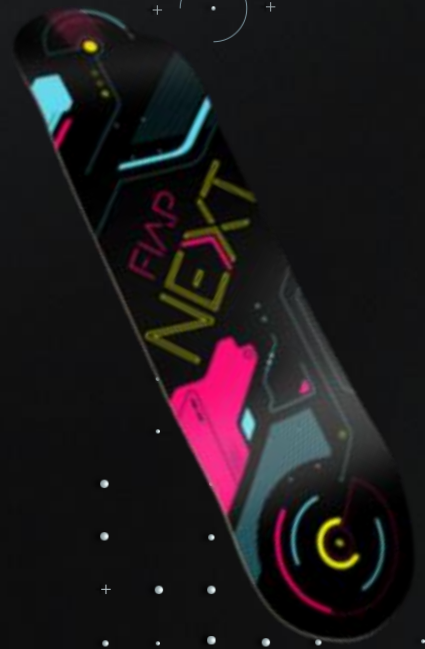
- Scrum Master 2025
  - Professores: Karina Costa, Thiago Keller e Thiago Yamamoto.

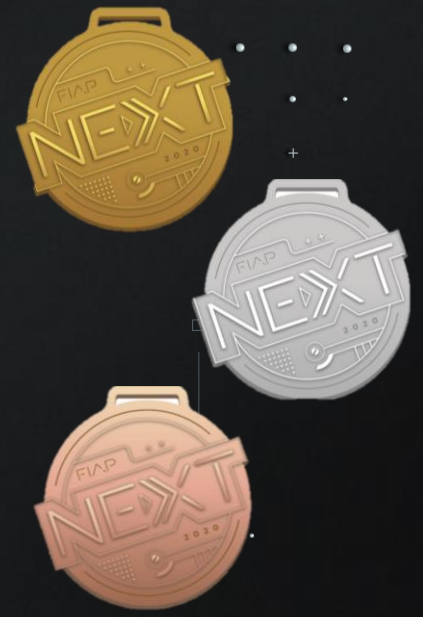
PREMIAÇÃO

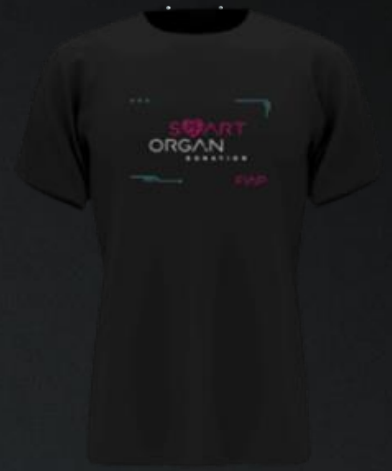
NEXT 2025

PREMIAÇÃO









1º Lugar

R$ 5.000,00

MEDALHA,

SHAPE E CAMISETA

2º Lugar

R$ 3.000,00

MEDALHA,

SHAPE E CAMISETA

3º Lugar

R$ 2.000,00

MEDALHA,

SHAPE E CAMISETA

CRONOGRAMA

1º SEMESTRE

CRONOGRAMA

DATA	EVENTO	STEAKHOLDER
14/04	Abertura do Challenge com a Mottu	Mottu
À definir	Primeira mentoria online com a Mottu	Mottu
Até 23/05	ENTREGA DA SPRINT 1	ALUNO
Até 30/05	Feedback das entregas SPRINT 1	PROFESSORES

AGENDA

CRONOGRAMA – 1º SEMESTRE

- **Abertura do desafio** – Kickoff com a empresa parceira.
- **1º Mentoria** com os profissionais da empresa parceira.
- **Entrega da 1º Sprint.**

AGENDAS DO CHALLENGE

KICKOFF DO CHALLENGE – 14/04

AGENDAS DO CHALLENGE

Fotos: https://photos.app.goo.gl/YWiBdurKo7oj3MiMA

ENTREGAS

1º SPRINT

POR DISCIPLINA

ADVANCED BUSINESS DEVELOPMENT WITH .NET

ENTREGA - 1º SPRINT

- Implementar uma API Restful utilizando ASP.NET Core (Controllers ou Minimal API) (85 pts)
  - Apresentar um CRUD pelo menos (GET (mais de 3 rotas e devidamente parametrizadas com QueryParams ou PathParams), POST, PUT, DELETE)
  - Apresentar os retornos HTTP adequados para cada rota (ok, NotFound, BadRequest, NoConent, Created)
  - Integração do Banco de dados Oracle via EF Core, com utilização de migrations para criação das tabelas
  - Open API Implementada seguindo os padrões para documentação das API's com interface gráfica (Swagger, Redoc ou Scalar)
- ReadMe do projeto e Github (15 pts)
  - Implementação do Readme do projeto apresentando: Descrição do projeto, Rotas, Instalação
  - Obs: a organização e apresentação do trabalho terá impacto na redução da nota ou penalidade adicional no trabalho.

COMPLIANCE, QUALITY ASSURANCE & TESTS

- **Visão e escopo do projeto em formato PITCH, contendo:**
  - Descrição detalhada do problema a resolver (peso 20%)
  - Descrição dos objetivos da solução idealizada(peso 20%)

- **Arquitetura da solução, aplicando TOGAF® e usando a ferramenta ARCHIMATE® (ARCHI).**
  - Desenhe em uma mesma página de diagrama no Archi:
    - Visão da arquitetura (peso 15%)
    - Arquitetura de negócio (peso 15%)
    - Arquitetura de sistema (peso 15%)
    - Arquitetura de tecnologia (peso 15%)

- Gere um PDF com o material desenvolvido (incluindo o diagrama do Archi e suba a sua resposta do Challenge sprint em tarefa específica criada no Teams - entregas fora do formato perderão 10%

ENTREGA - 1º SPRINT

DEVOPS TOOLS & CLOUD COMPUTING 1/4

ENTREGA - 1º SPRINT

Primeira opção:

01) Conteinerização da API solicitada na disciplina ADVANCED BUSINESS DEVELOPMENT WITH .NET em Nuvem (até 85 pontos)

02) Documentação no GitHub (até 15 pontos)

As seguintes regras solicitadas por essa disciplina serão avaliadas para a execução em nuvem:

. Apresentar um CRUD (GET, POST, PUT, DELETE) com pelo menos 5 inserts com conteúdo significativo

. Integração do Banco de dados Oracle

. Open API Implementada seguindo os padrões para documentação das API's com interface gráfica (Swagger, Redoc ou Scalar)

. ReadMe do projeto e Github

. Implementação do Readme do projeto apresentando: Descrição do projeto, Rotas, Instalação, Dockerfile, scripts do Azure CLI

A nota irá variar de acordo com a completude da entrega e, se a entrega não funcionar em nuvem, não poderá ser avaliada, ficando apenas a parte do GitHub para análise e totalização da nota

DEVOPS TOOLS & CLOUD COMPUTING 2/4

ENTREGA - 1º SPRINT

Segunda opção:

01) Conteinerização da API solicitada na disciplina JAVA ADVANCED em Nuvem (até 85 pontos)

02) Documentação no GitHub (até 15 pontos)

As seguintes regras solicitadas por essa disciplina serão avaliadas para a execução em nuvem:

. Spring Web para criação da API

. Spring Data JPA para acesso ao banco de dados

. Banco de dados H2 ou Oracle

. CRUD completo para pelo menos duas entidades e pelo menos 5 inserts com conteúdo significativo em cada entidade

. Relacionamento entre entidade

. O repositório deve conter o código do projeto na raiz, Dockerfile, scripts do Azure CLI

. O repositório deve ter um readme com a descrição do projeto, nomes dos alunos e instruções para executar

A nota irá variar de acordo com a completude da entrega e, se a entrega não funcionar em nuvem, não poderá ser avaliada, ficando apenas a parte do GitHub para análise e totalização da nota

DEVOPS TOOLS & CLOUD COMPUTING 3/4

Tarefas da disciplina (independente da escolha da tecnologia):

01) Provisionar uma Máquina Virtual Linux no Azure via CLI

02) Abrir as portas necessárias ao projeto via CLI

03) Instalar o Docker na VM criada via SSH

04) Executar a entrega de .NET ou Java na VM utilizando Docker

- Projeto executando em background
- Imagens otimizadas (ex: python:3.11-slim, node:18-alpine)
- Rodar aplicação com usuário sem privilégios administrativos

05) Acessar externamente e realizar todos os testes

06) Desenhe a Arquitetura Macro da sua solução na nuvem (ex: fluxo de usuários, front-end, API, banco de dados, VM, containers, etc.). Utilize ferramentas como Draw.io ou Visual Paradigm (links abaixo) e inclua legendas, rótulos, imagens e setas de fluxo para facilitar a compreensão

07) Ao final da entrega delete a Máquina Virtual criada

Obs.: Pode gerar a imagem do Docker em sua máquina de desenvolvimento e subir para o Docker Hub. Assim a equipe somente roda a solução na VM baixando a imagem

ENTREGA - 1º SPRINT

DEVOPS TOOLS & CLOUD COMPUTING 4/4

ENTREGA - 1º SPRINT

Entrega:

Grave um vídeo demonstrando: A Criação da VM via CLI, a aberturas das portas necessárias via CLI, Instalação do Docker, Funcionamento da aplicação com o Docker e Persistência de dados (mostrar cada operação executada no banco)

Entrega final:

Um único arquivo PDF nomeado como: **NomeDoGrupo_Challenge_1Sem_2TDS.pdf**, contendo:

01) Uma folha de rosto com o nome da equipe, RM e nome completo dos alunos. Inclua um índice para organização

02) Desenho da arquitetura com legenda, fluxos etc

03) Link para o repositório no GitHub

04) Link para o vídeo no YouTube

05) Print da evidência da remoção da VM

Sugestões de ferramentas de desenho

- Azure Diagram Tool (Visual Paradigm)
  https://online.visual-paradigm.com/diagrams/features/azure-architecture-diagram-tool/
- Draw.io
  https://app.diagrams.net/

DISRUPTIVE ARCHITECTURES: IOT, IOB & GENERATIVE IA

Objetivos Principais

- Desenvolver um protótipo funcional simples (físico ou simulado) que utilize tecnologias de IoT **e/ou** Visão Computacional para solução proposta.
- Gravar um vídeo pitch de aproximadamente 5 minutos com a demonstração funcional das primeiras implementações técnicas utilizando IoT e/ou Visão Computacional.

Objetivos Específicos

- Apresentar de forma clara o **problema real** que o projeto busca resolver.
- Justificar a **aplicação de tecnologias de IoT e/ou Visão Computacional**.
- Apontar as **tecnologias utilizadas**, explicando **como serão aplicadas na solução**.
- Demonstrar o funcionamento de **componentes-chave** do projeto (por exemplo: sensores conectados, processamento de imagens, modelos de IA, automações etc).
- Evidenciar a **viabilidade técnica inicial** do projeto como prova de conceito.

ENTREGA - 1º SPRINT

DISRUPTIVE ARCHITECTURES: IOT, IOB & GENERATIVE IA

Tarefa

- Caso IoT:
  - Implementar o protótipo, usando montagem física (protoboard) ou simulação (ex.: Wokwi, Tinkercad, etc).
  - Criar um Dashboard com as informações coletadas usando protocolos (HTTP ou MQTT)
- Caso Visão Computacional
  - Implementar um script Python para detecção / classificação e/ou rastreamento de objeto alinhado ao projeto.
  - O output visual do objeto detectado deve ser realizado para demonstrar a detecção / classificação e/ou rastreamento.

Observações importantes:

- Explique como os frameworks / ferramentas serão aplicados no projeto.
- Utilize imagens e vídeos que tentem reproduzir / simular o ambiente real (exemplo: pátio de motos da Mottu).
- Faça a escolhas dos sensores/atuadores e monte o circuito de forma organizada.
- Evite apresentações completamente automatizadas por IA. O uso de recursos visuais é recomendado, mas o protagonismo deve ser do grupo no vídeo.

ENTREGA - 1º SPRINT

DISRUPTIVE ARCHITECTURES: IOT, IOB & GENERATIVE IA

Critérios de Avaliação

(até 60 pontos) Aplicação técnica de conceitos de IoT **e/ou** Visão Computacional

(até 20 pontos) Clareza e didática da apresentação em vídeo

(até 20 pontos) Organização do repositório e documentação técnica

Entregáveis obrigatórios: Arquivo .zip contendo:

- Link do vídeo publicado no YouTube (em modo não listado) com a apresentação da ideia central do projeto, demonstração funcional (real ou simulada) dos principais recursos implementados.
- Link para o repositório no GitHub, apresentando código-fonte e estrutura do projeto e o README com instruções de uso, tecnologias utilizadas e resultados parciais.

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. Sugestão: confira seu anexo antes de publicar.
- Não serão aceitos arquivos enviados pelo Teams ou fora do prazo.

ENTREGA - 1º SPRINT

JAVA ADVANCED 1/2

Nesta entrega você deve criar uma API Rest que dará suporte a sua solução do challenge. A API deve atender os seguintes **requisitos técnicos**:

- Spring Web para criação da API
- Spring Data JPA para acesso ao banco de dados
- Banco de dados H2 ou Oracle
- CRUD completo para pelo menos duas entidades
- Relacionamento entre entidade
- Validação de campos com Bean Validation
- Paginação de resultados
- Ordenação de resultados
- Busca por parâmetros
- Cache para otimizar requisições
- Boas práticas de design REST
- Tratamento centralizado de erros
- Utilização de DTO

ENTREGA - 1º SPRINT

JAVA ADVANCED 2/2

No momento da correção da entrega, serão cobrados os seguintes **critério de avaliação**:

- 70 pontos - Requisitos Técnicos: conforme slide 1
- 10 pontos - Relevância: a proposta do projeto deve ser relevante e estar de acordo com o tema do challenge.
- 10 pontos - Inovação: o projeto deve apresentar uma solução inovadora e criativa para a resolução do challenge.
- 10 pontos - Organização: a gerência dos artefatos de entrega deve atender aos requisitos da sprint.

A **entrega** deve ser realizada dentro do prazo através do portal do aluno, atendendo os seguintes requisitos:

- Arquivo de texto com link do repositório público do github.
- O repositório deve conter o código do projeto na raiz.
- O repositório deve ter um readme com a descrição do projeto, nomes dos alunos e instruções para executar

ENTREGA - 1º SPRINT

MASTERING RELATIONAL AND NON-RELATIONAL DATABASE

ENTREGA - 1º SPRINT

Entregar o Projeto de Banco de Dados Relacional, compondo:

- Modelo descritivo, descrevendo a estrutura de armazenamento. (10 pontos)
  - Os Diagramas que deverão ser construídos no Oracle Data Modeler, cumprindo a notação DER-(Logical Model) e o MER-(Physical Model).
  - Deverá ser utilizado obrigatoriamente a notação de Barker para o DER-(Logical Model) e no mínimo estar na 3ª Forma Normal (3FN)
- Gerar o Modelo Físico e criar os objetos/esquema no banco de dados. (10 pontos)
- Para cada Tabela preencher no mínimo 5 registros de acordo com a especificação do projeto. (10 pontos)

MASTERING RELATIONAL AND NON-RELATIONAL DATABASE

Programação PL/Sql:

- Criar dois blocos anônimos para mostrar os dados inseridos, com pelo menos 3 consultas de junções (Joins) utilizando agrupamento (group by) e ordenação (order by). (30 pontos)

ENTREGA - 1º SPRINT

MASTERING RELATIONAL AND NON-RELATIONAL DATABASE

Programação PL/Sql;

- Criar um bloco que deverá ler os dados de uma tabela e, na mesma linha, mostrar o valor de uma coluna da linha atual, o valor dessa mesma coluna na linha anterior e o valor dessa mesma coluna na próxima linha. Caso a linha anterior ou a próxima linha não existir, apresentar a palavra "Vazio". O relatório deve ter, pelo menos, cinco linhas de dados. A tabela e a coluna a ser exibida fica a cargo do grupo. (40 pontos)

Vide exemplo

Cod_emp	Cod_dep	Anterior	Atual	Próximo
1	10	Vazio	3000	3500
2	10	3000	3500	4000
3	10	3500	4000	Vazio

ENTREGA - 1º SPRINT

MASTERING RELATIONAL AND NON-RELATIONAL DATABASE

Entregável: arquivo zipado contendo o Projeto de Banco de Dados Relacional em pdf e um arquivo sql com os scripts separados por tabela.

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.

ENTREGA - 1º SPRINT

MOBILE APPLICATION DEVELOPMENT

Criar um protótipo funcional de aplicativo em React Native com Expo, simulando o uso do app para o mapeamento inteligente do pátio e gestão das motos.

Requisitos

1. Navegação entre telas (20 pontos)
2. Protótipo visual completo (30 pontos)
3. Formulário com manipulação de estado (20 pontos)
4. Armazenamento local com AsyncStorage (20 pontos)
5. Projeto no GitHub Classroom (10 pontos)

Formato de Entrega

- Arquivo texto com o endereço do repositório no GitHub Classroom

ENTREGA - 1º SPRINT

MOBILE APPLICATION DEVELOPMENT

1. Navegação entre telas (20 pontos)
    - O projeto deve utilizar uma biblioteca de navegação (Expo Router ou React Navigation).
    - O app deve conter ao menos cinco rotas navegáveis.
2. Protótipo visual funcional (30 pontos)
    - Todas as telas devem possuir um layout funcional e coerente com o fluxo de uso, mesmo que o design visual seja simples.
    - É permitido usar dados mockados (fixos ou com useState).
    - As telas devem estar organizadas de acordo com um fluxo lógico de uso.
3. Formulário com manipulação de estado (20 pontos)
    - Pelo menos um formulário deve controlar os campos com useState e reagir às mudanças (ex: salvar, exibir ou limpar os dados).
    - O formulário deve armazenar os dados inseridos enquanto o usuário digita e exibi-los dinamicamente na tela.

ENTREGA - 1º SPRINT

MOBILE APPLICATION DEVELOPMENT

4. Armazenamento local com AsyncStorage (20 pontos)
    - Deve haver pelo menos um tipo de dado salvo utilizando AsyncStorage.
    - Após o reinício do app, esse dado deve ser carregado corretamente, ou seja, o valor salvo anteriormente deve ser restaurado ao abrir o aplicativo novamente.
    - O tipo de dado a ser salvo pode incluir, por exemplo:
        - Informações de um formulário preenchido pelo usuário.
        - Preferências do usuário.
        - Qualquer outro dado relevante para o funcionamento do app.
5. Projeto no GitHub Classroom (10 pontos)
    - Código organizado.
    - O README deve conter nome completo e RM dos integrantes, além de instruções claras sobre como rodar o projeto localmente e a descrição da solução implementada.

ENTREGA - 1º SPRINT

DÚVIDAS

Sobre a entrega da disciplina, procure o professor responsável pela sua turma e disciplina;

Sobre o challenge, procure um Scrum Master.

DÚVIDAS

FIAP