



CHALLENGE XP Inc. – 2025

RECOMENDAÇÃO DE INVESTIMENTOS (ASSESSOR VIRTUAL)

TURMA: ENGENHARIA DE SOFTWARE 3º ANO, INÍCIO: 2025/FEV

SPRINTS SEGUNDO SEMESTRE

Parceria:



Neste segundo semestre, o projeto dará continuidade ao trabalho iniciado no início do ano. No primeiro semestre, tivemos apenas uma entrega, a Sprint, que contou como Sprint 1 e Sprint 2. Agora, teremos duas entregas: **Sprint 3 e Sprint 4.**

Desta vez teremos mais duas disciplinas integradas (C# e IoT & IoB), com solicitações e orientações feitas pelos professores. Assim como antes, as entregas serão comuns para todos os grupos, mas a forma como cada equipe desenvolverá suas soluções poderá variar de acordo com as estratégias e abordagens escolhidas.

Reforçamos a importância de organização e alinhamento dentro de cada grupo, garantindo que as entregas sejam realizadas dentro dos prazos, aproveitando ao máximo a integração entre as disciplinas e o avanço natural do projeto.

Em caso de dúvidas, procurem os professores responsáveis pelas entregas.



RELEMBRANDO OS CASES....

DESAFIO (CASE 1)

APOSTAS COMPULSIVAS

- O crescimento acelerado do mercado de apostas no Brasil, impulsionado pela digitalização e desregulamentação, trouxe à tona uma série de desafios sociais e psicológicos. O aumento das apostas compulsivas tem resultado em sérias consequências financeiras, afetando a saúde mental e as relações interpessoais de muitos brasileiros. A falta de ferramentas eficazes para detectar, inibir e tratar a compulsão por apostas representa uma lacuna significativa que precisa ser abordada de forma urgente e eficaz. Portanto, o desafio é desenvolver uma solução que não apenas identifique os apostadores em risco, mas que também ofereça suporte e alternativas saudáveis.

OBJETIVO

Case I:

- **Deteção e Classificação:** Criar uma solução que utilize algoritmos de machine learning para identificar padrões de comportamento que indiquem predisposição a apostas compulsivas, classificando os usuários em diferentes níveis de risco.
- **Inibição:** Implementar intervenções em tempo real que alertem os usuários sobre comportamentos de apostas prejudiciais e ofereçam alternativas saudáveis.
- **Tratamento:** Estabelecer um módulo de suporte que conecte os usuários a uma rede de apoio, incluindo profissionais de saúde mental e grupos de terapia.

SUGESTÕES

- **Plataforma Mobile Responsiva:** Uma aplicação acessível em dispositivos móveis, permitindo que os usuários acessem suporte e recursos de forma conveniente.
- **Sistema de Coleta de Dados:** Mecanismos éticos para a coleta de dados dos usuários, garantindo consentimento informado e transparência.
- **Modelos Preditivos:** Desenvolvimento de modelos de machine learning que identifiquem usuários em risco de compulsão por apostas, com atualizações regulares baseadas em novos dados.
- **Governança de Dados:** Implementação de práticas rigorosas para garantir a segurança e privacidade dos dados dos usuários, em conformidade com a LGPD.
- **Princípios de IA Ética:** Criação de algoritmos que sejam transparentes, auditáveis e livres de vieses, com revisão por uma equipe multidisciplinar.

EXEMPLOS DE FUNCIONALIDADE

- **Notificações Personalizadas:** Alertas em tempo real para usuários que exibem comportamentos de apostas preocupantes, com recomendações para limitar apostas ou buscar ajuda.
- **Sugestões de Atividades Alternativas:** Oferecimento de opções de atividades saudáveis, como exercícios físicos, meditação ou hobbies, para desviar a atenção das apostas.
- **Acesso à Rede de Apoio:** Funcionalidade que permite aos usuários agendar consultas virtuais com profissionais de saúde mental ou participar de grupos de terapia online. Pode ser feito de forma hiperpersonalizada.
- **Relatórios de Comportamento:** Geração de relatórios personalizados que mostram a evolução do comportamento de apostas do usuário, ajudando na autoavaliação e conscientização.
- **Educação sobre Riscos:** Módulo educativo que fornece informações sobre os riscos associados às apostas e estratégias para um jogo responsável.
- **Simulação de oportunidades:** Enviar um relatório mostrando caminhos alternativos de aplicação do dinheiro gasto com apostas
- **Alertas de conscientização:** Casos e estatísticas que possam alertar e conscientizar as pessoas dos riscos associados à aposta compulsiva.

CONSIDERAÇÕES

- 1- Código: aberto com licença MIT/Apache/BSD
- 2- Datasets: licença aberta (CC0/CC BY). Dados pessoais devem ser tratados.
- 3- Em caso de testes da ferramenta deve se ter o consentimento do usuário

DESAFIO (CASE 2) RECOMENDAÇÃO DE INVESTIMENTOS (ASSESSOR VIRTUAL)

- Com a evolução da tecnologia e o surgimento de diversos canais de acesso, o investimento e os instrumentos financeiros tornaram-se cada vez mais populares. Atualmente, abrir uma conta digital e iniciar investimentos é uma tarefa muito mais acessível. Os clientes vêm de diferentes segmentos e classes sociais, apresentando perfis de investimento variados. No entanto, o aumento e a diversificação dos produtos de investimento tornam esse processo bastante complexo. Diante de tantas opções e variáveis, é desafiador determinar a combinação ideal que atenda às necessidades e objetivos de cada cliente.

EXEMPLOS DE ENTRADAS

- **Lista de ativos disponíveis para investimento:** essa lista deve ser variável e será montada por um assessor humano. Ex.: ele pode colocar na lista 10 ativos de renda fixa, 5 de renda variável, fundos diversos etc. Todos os ativos devem estar segmentados em classes e conter informações detalhadas.
- **Variáveis macroeconômicas:** taxas de juros, câmbio, tributos, etc.
- **Preferências do cliente:** histórico de compras e investimentos, etc.
- **Suitability do cliente;** Conservador, moderado, agressivo.
- **Liquidez do cliente:** quanto em dinheiro ele tem disponível para aplicar?
- **Objetivos:** Objetivos do cliente de curto, médio ou longo prazo.

EXEMPLOS DE SAÍDAS

- **Carteiras recomendadas:** uma ou mais sugestões de carteiras recomendadas dadas as entradas.
- **Explicação:** Explicação de todas as recomendações presentes na carteira recomendada.

SOLUÇÃO

- Sistema que receba as entradas, processe todas as informações e retorne as saídas desejadas com suas devidas explicações. O sistema deve levar em consideração práticas éticas e responsáveis de I.A.: mitigar vieses, garantir conformidade com LGPD, usar técnicas de XAI, etc.

EXEMPLOS DE SITUAÇÕES

- **Cliente Iniciante em Investimentos:**

1. **Cenário:** Maria, uma jovem de 25 anos, nunca investiu antes e deseja começar a aplicar uma pequena quantia que economizou. Ela se sente perdida diante das muitas opções disponíveis e não sabe por onde começar.
2. **Desafio:** O assessor virtual deve entender o perfil conservador de Maria e recomendar uma carteira que inclua ativos de baixo risco, como renda fixa, além de explicar de forma clara os benefícios e riscos de cada ativo.

- **Cliente com Objetivos de Longo Prazo:**

1. **Cenário:** João, um profissional de 40 anos, está interessado em construir uma reserva para a aposentadoria e tem um horizonte de investimento de 20 anos. Ele está disposto a assumir riscos moderados para potencializar seus ganhos.
2. **Desafio:** O assessor virtual precisa sugerir uma carteira diversificada que inclua ações de crescimento e fundos de investimento, além de explicar como essas escolhas se alinham com os objetivos de longo prazo de João e a necessidade de crescimento do capital.

EXEMPLOS DE SITUAÇÕES

- **Cliente com Necessidade de Liquidez:**

1. **Cenário:** Ana, uma empresária de 35 anos, precisa de liquidez para financiar um projeto em breve. Ela tem uma quantia considerável em dinheiro, mas não quer deixar de investir.
2. **Desafio:** O assessor virtual deve montar uma carteira que priorize ativos com alta liquidez, como fundos de renda fixa e títulos públicos, e justificar as escolhas, garantindo que Ana compreenda a importância de manter a liquidez para seu projeto.

- **Cliente com Tolerância ao Risco Variável:**

1. **Cenário:** Carlos, um investidor de 50 anos, tem um perfil que varia entre conservador e moderado, dependendo do desempenho do mercado. Ele já investiu em ações, mas ficou receoso após uma queda significativa.
2. **Desafio:** O assessor virtual deve avaliar o histórico de investimentos de Carlos e suas reações a diferentes cenários de mercado, recomendando uma carteira que equilibre ações e renda fixa, além de explicar como a diversificação pode ajudar a mitigar riscos em períodos de volatilidade.

EXEMPLOS DE SITUAÇÕES

- **Cliente com Impacto Social:**

1. **Cenário:** Luiza, uma investidora de 30 anos, deseja que seus investimentos tenham um impacto social positivo e está interessada em ativos sustentáveis ou de empresas que adotam práticas éticas.
2. **Desafio:** O assessor virtual precisa identificar e recomendar ativos que atendam a critérios de sustentabilidade, explicando como esses investimentos podem não apenas proporcionar retorno financeiro, mas também contribuir para causas sociais e ambientais.

- **Cliente com Variações Macroeconômicas:**

1. **Cenário:** Roberto, um investidor de 45 anos, está preocupado com a inflação crescente e as taxas de juros elevadas. Ele quer proteger seu capital e, ao mesmo tempo, buscar oportunidades de crescimento.
2. **Desafio:** O assessor virtual deve analisar as variáveis macroeconômicas atuais e sugerir uma carteira que inclua ativos que historicamente se saem bem em ambientes inflacionários, como commodities ou imóveis, e explicar como essas escolhas podem ajudar Roberto a preservar seu poder de compra.

CONSIDERAÇÕES

- 1- Código: aberto com licença MIT/Apache/BSD
- 2- Datasets: licença aberta (CC0/CC BY). Dados pessoais devem ser tratados.
- 3- Em caso de testes da ferramenta deve se ter o consentimento do usuário

REGRAS DO PROJETO

- OS GRUPOS DEVEM SER FORMADOS DE 3 A 5 INTEGRANTES, NO MÁXIMO.
- OS GRUPOS DEVEM SER FORMADOS POR ALUNOS DA MESMA TURMA.
- NÃO É POSSÍVEL TROCAR DE GRUPO DURANTE A CHALLENGE NO MESMO SEMESTRE.
- AS TROCAS PODEM OCORRER NA VIRADA DE SEMESTRE SOB ANÁLISE DO SCRUM MASTER.

Em caso de dúvidas, entre em **CONTATO** com o
SCRUM MASTER: Prof. Yan Coelho



proyan.coelho@fiap.com.br



Microsoft Teams

REGRAS DAS ENTREGAS

- SOMENTE O REPRESENTANTE DE CADA GRUPO REALIZA A ENTREGA DAS ATIVIDADES NO TEAMS.
- CHAMAMOS AS ENTREGAS DE “SPRINTS” E CADA UMA DELAS SERÁ DEFINIDA PELO PROFESSOR DA DISCIPLINA NO TEAMS.
- EM CASO DE DÚVIDA CONSULTE O PROFESSOR RESPONSÁVEL PELA DISCIPLINA.
- AS ENTREGAS TEM DATA! O TEAMS NÃO ACEITARÁ ENTREGAS COM ATRASO.

OBJETIVOS DE DESENVOLVIMENTO “DISCIPLINAS”

- **Arquitetura orientada a Servicos e web services.**
- **Mobile Development and IoT.**
- **Operating systems.**
- **Testing, compliance and Quality Assurance.**
- **Cybersecurity**
- **C# SOFTWARE DEVELOPMENT**
- **PHYSICAL COMPUTING: IOT & IOB**

SPRINT 3

- Data de **ENTREGA – 21/09/2025** – Via tarefas **Teams**.

ARQUITETURA ORIENTADA A SERVICOS E WEB SERVICES.

Desenvolva uma aplicação que exponha e consuma serviços via **WebService** (REST ou SOAP), contemplando modelagem, implementação e boas práticas.

Critérios de avaliação

Estruturação do projeto, classes e código limpo (peso 25%)

- Organização do projeto por artefatos ou features.
- Uso de VO (Value Objects), DTOs e enums de forma adequada.
- Separação clara entre camadas (controller, service, repository, etc.).
- Código legível, comentado quando necessário, e aderente a boas práticas.

Mapeamento e manipulação de requisições (peso 20%)

- Implementação correta de endpoints (CRUD ou casos de uso definidos).
- Validações nas entradas (parâmetros, corpo da requisição).
- Padronização dos retornos (uso de `ResponseEntity` ou equivalente).
- Tratamento de erros com mensagens personalizadas e `RestControllerAdvice`.

Conexão com banco de dados e persistência (peso 20%)

- Modelagem correta das entidades.
- Uso de migrações para manter controle das alterações no banco.
- CRUD completo e funcional.
- Garantir integridade e consistência dos dados.

ARQUITETURA ORIENTADA A SERVICOS E WEB SERVICES.

Critérios de avaliação

Interface de acesso ao serviço (peso 15%)

- Pode ser feita via cliente HTTP (Postman, Insomnia), ou integração com uma interface simples (web/console).
- O aluno deve demonstrar o consumo dos serviços implementados.

Documentação do projeto (peso 10%)

- Arquivo README.md contendo:
 - ✓ Descrição do projeto.
 - ✓ Passos de configuração e execução.
 - ✓ Exemplos de requisições e respostas.
 - ✓ Tecnologias utilizadas.

Arquitetura e diagramas (peso 10%)

- Diagramas de arquitetura (camadas e componentes).
- Diagrama de entidades (ER ou UML).
- Representação clara dos casos de uso implementados como serviços.

Observações: Métodos e funções devem estar bem estruturados. O link de acesso ao repositório deve ser entregue, garantindo permissão ao professor.

MOBILE DEVELOPMENT AND IOT.

Baseado na solução escolhida nesta Global Solution, desenvolva um aplicativo utilizando React Native que seja apresentável no emulador mobile Android ou iOS.

Entregáveis

- No sistema da FIAP arquivo com .txt do repositório do github ou github classroom com o CP3.
- No arquivo readme.md no github, colocar nomes dos integrantes do grupo com Nome e RM e as funcionalidades do aplicativo.

Critérios de avaliação

Critérios de avaliação:

(25 pts) Criar uma tela home que utilize navegação: Drawer Navigation ou TabNavigation.

(25pts) Desenvolver Tela de Login com AsyncStorage.

(40pts) Desenvolver de 3 a 5 telas com informações e componentes.

(10 pts) Arquitetura e organização de pastas, componentes e telas.

Penalidades

(-10pts) Falhar na navegação do aplicativo ou não ter navegação entre telas.

(-15pts) Tela de login sem funcionar, falta da informação de usuário/senha para realizar o fluxo de login do aplicativo na documentação do github.

(-15pts) Por tela não implementada ou com falha, erros e ou bugs.

(-60pts) O build não funcionar no emulador ao realizar o gitclone

OPERATING SYSTEMS.

- **25% - Implementar um servidor Linux (Debian ou Ubuntu Server, sem GUI) com Docker, NGINX instalados e acesso à internet liberado e configurado.**
- **25% - Instale uma imagem SQLite dentro de um Contêiner Docker e monte um volume externo (da sua máquina Linux) como caminho onde seu arquivo **.db** vai ser localizado, dessa forma compartilhando o arquivo entre o contêiner e a própria máquina Linux.**
- **25% - Crie um endpoint em Node para receber dados dos usuários enviados pela sua aplicação e inserí-los no banco de dados SQLite.**
- **25% - Crie um **cronjob** simples em Shell Script na máquina Linux responsável por anonimizar dados em colunas com possíveis dados PII (como nome completo, CPF, RG, entre outras informações).**

TESTING, COMPLIANCE AND QUALITY.

Desenvolva o plano detalhado do seu projeto Challenge, usando o AZURE BOARDS, contendo:

- **Estrutura de trabalho em forma de Backlog de Produto, com detalhes de Épicas, Features e Product Backlog Itens, no padrão Scrum (peso 25%);**
- **Documentação de descrição e critérios de aceite, explicando as histórias de usuário formadas pelos Épicas, Features e Product Backlog Itens (peso 25%);**
- **Apontamento da Prioridade, Esforço, Dependência entre itens do backlog, ordenando o backlog na sequência de implementação e entrega (peso 25%);**
- **Detalhamento da Sprint atual do projeto (em curso), com Tarefas detalhadas com suas descrições, pontuações de esforço e dependências técnicas (25%).**

Entregue o link de acesso ao seu plano de projetos na nuvem. Garanta que o professor esteja cadastrado como membro da organização e administrador do projeto.

Entregas fora das especificações implicarão em perda de 10% na nota de avaliação.

CYBERSECURITY

Objetivo da Sprint Integrar testes de segurança automatizados ao pipeline de desenvolvimento (CI/CD), utilizando SAST, DAST e SCA para identificar vulnerabilidades em tempo real, desde o código-fonte até a aplicação em execução.

Tarefa 1: SAST – Análise Estática no Pipeline de CI – 2,5

- Descrição:

Configurar ferramentas de SAST no pipeline de Integração Contínua (CI) para realizar análise estática do código-fonte. O objetivo é detectar vulnerabilidades como:

- Injeção de código (SQL, XSS)
- Uso inseguro de funções
- Falhas de autenticação e autorização
- Exposição de dados sensíveis
- Ferramentas sugeridas: Semgrep, SonarQube, Checkmarx

- Entregável:

Relatório automatizado gerado pelo pipeline CI com as vulnerabilidades encontradas, classificadas por severidade e com recomendações de correção.

CYBERSECURITY

Tarefa 2: DAST – Testes Dinâmicos no Pipeline de CD – 3,0

- Descrição:
Integrar ferramentas de DAST no pipeline de Entrega Contínua (CD) para testar a aplicação em execução em ambiente de staging. O foco é identificar falhas de segurança em tempo de execução, como:
 - Falhas de autenticação
 - Exposição de dados via endpoints
 - Configurações inseguras
 - Comportamentos inesperados sob carga
 - Ferramentas sugeridas: OWASP ZAP, Burp Suite (automatizado), Nikto
- Entregável:
Relatório de varredura dinâmica com evidências de vulnerabilidades exploráveis, incluindo payloads utilizados e sugestões de mitigação.

CYBERSECURITY

Tarefa 3: SCA – Análise de Componentes de Terceiros – 1,5

- Descrição:
Executar SCA para identificar vulnerabilidades em bibliotecas e dependências externas utilizadas no projeto. A análise deve ocorrer automaticamente no pipeline CI, verificando:
 - Versões desatualizadas
 - CVEs conhecidos
 - Licenças incompatíveis
 - Ferramentas sugeridas: Snyk, OWASP Dependency-Check, WhiteSource
- Entregável:
Relatório de dependências com riscos associados, sugestões de atualização e plano de ação para substituição ou correção.

CYBERSECURITY

Tarefa 4: Integração e Monitoramento no CI/CD – 3,0

- Descrição:
Consolidar os três testes (SAST, DAST, SCA) em um pipeline CI/CD unificado, com:
 - Gatilhos automáticos a cada commit/push
 - Bloqueio de deploy em caso de vulnerabilidades críticas
 - Notificações em tempo real para o time de desenvolvimento
 - Dashboards de segurança contínua
- Entregável:
Documentação do pipeline CI/CD com os testes integrados, políticas de segurança aplicadas e exemplos de execução com logs e alertas.

C#

Desenvolva uma aplicação que receba os valores para manipular com banco de dados.

- **Estruturação das classes e código limpo (peso 25%);**
- **Manipulação de arquivos (txt/json) (peso 20%);**
- **Conexão com banco de dados (CRUD completo) (peso 20%);**
- **Interface (console / forms / web) (15%);**
- **Documentação do projeto (10%);**
- **Apresentar arquitetura em diagramas (10%).**

Métodos e funções estruturadas são esperadas para avaliação, legibilidade do código e estruturação. O código deverá estar em um versionador e um arquivo Readme;

Entregue o link de acesso ao seu plano de projetos na nuvem, garantir que o professor tenha acesso ao repositório. Entregas fora das especificações implicarão em perda de 10% na nota de avaliação.

IOT & IOB

Objetivo: desenvolver uma aplicação **local (desktop/notebook)** que realize **reconhecimento/identificação facial do usuário** usando **OpenCV, IA/ML ou Haar Cascade** (qualquer tecnologia que envolva **parâmetros ajustáveis**). **Não precisa** estar conectada à aplicação/solução final.

Requisitos técnicos mínimos

- Usar uma abordagem de detecção/identificação facial (OpenCV + Haar Cascade, MediaPipe, dlib, etc.).
- Demonstrar parâmetros relevantes e como impactam o resultado.
- Exibir a detecção em tela (retângulos, landmarks, identificação).
- Código executável com instruções claras.

Vídeo Explicativo (até 3 min) – 70 pontos

- Problema/contexto.
- Tecnologia escolhida.
- Demonstração de parâmetros e impacto.
- Funcionamento prático.
- Limitações e próximos passos.

Repositório GIT (30 pontos)

- README claro (objetivo, execução, dependências, parâmetros).
- Código organizado.
- Nota ética sobre uso de dados faciais.

SPRINT 4

- Data de **ENTREGA** – **20/10/2025** – Via tarefas **Teams**.

ARQUITETURA ORIENTADA A SERVICOS E WEB SERVICES.

Desenvolva uma aplicação que implemente autenticação/autorização segura e demonstre boas práticas de projeto orientado a objetos, com documentação e testes.

Critérios de avaliação

Estruturação do projeto, código limpo e princípios SOLID (peso 25%)

- Aplicação de interfaces, polimorfismo e despacho dinâmico de métodos.
- Separação clara das responsabilidades em camadas (controller, service, repository).
- Aplicação dos princípios SOLID de forma consistente.
- Código legível, modularizado e aderente a boas práticas.

Configurações de segurança e autenticação (peso 20%)

- Configuração de segurança stateless.
- Uso de JWT (JSON Web Token) para autenticação e autorização.
- Senhas criptografadas com BCryptPasswordEncoder.
- Implementação de filtros para interceptação de requisições.
- Integração com Auth0 ou outra solução equivalente (quando aplicável).

Regras de negócio implementadas como serviços (peso 15%)

- Lógica encapsulada em services, separados do controller.
- Uso adequado de interfaces para permitir extensibilidade e reuso.
- Clareza e coesão na implementação das regras.

ARQUITETURA ORIENTADA A SERVICOS E WEB SERVICES.

Documentação automática da API (peso 15%)

- Configuração do SpringDoc + Swagger/OpenAPI.
- Endpoints documentados com descrições, exemplos de requisições e respostas.
- Organização da documentação em grupos ou tags, se necessário.

Testes automatizados (peso 15%)

- Implementação de testes unitários para classes de serviço.
- Implementação de testes de integração para endpoints da API.
- Uso de mocks para isolar dependências (quando necessário).
- Boa cobertura de código nos testes.

Documentação do projeto (peso 10%)

- Arquivo README.md contendo:
 - ✓ Descrição do projeto.
 - ✓ Instruções de execução (como rodar a aplicação e acessar a API).
 - ✓ Como rodar os testes.
 - ✓ Tecnologias utilizadas.

Observações: O link do repositório deve ser entregue garantindo acesso ao professor. Métodos e classes devem seguir boas práticas de organização e legibilidade.

MOBILE DEVELOPMENT AND IOT.

Baseado na solução escolhida nesta Global Solution, desenvolva um aplicativo utilizando React Native que seja apresentável no emulador mobile Android ou iOS.

Entregáveis

- arquivo com .txt do repositório do github ou github classroom com o CP3.
- No arquivo readme.md no github, colocar nomes dos integrantes do grupo com Nome e RM e as funcionalidades do aplicativo.

Critérios de avaliação

Critérios de avaliação:

- (30pts) CRUD completo nas funcionalidade do aplicativo, consumindo dados de uma API Restfull ou Firebase para obtenção e gestão dos dados.
- (30pts) Validação dos cenários de erro, com feedback claro e informativo para o usuário.
- (30pts) Aplicativo com navegação fluída e com tratamentos de possíveis interrupções.
- (10pts) Arquitetura e organização de pastas, componentes e telas.

Penalidades

- (-20pts) Falhar em conectar a API nas telas do aplicativo.
- (-10pts) Não ter tentar novamente em caos de interrupções.
- (-10pts) Falta de documentação das funcionalidades no github readme.md
- (-10pts) Por falhas, bugs e erros que possam ocorrer no aplicativo.
- (-50pts) O build do App não funcionar no emulador ao realizar o git clone, falhas nas dependências e versões.

OPERATING SYSTEMS.

- **25% - Crie um segundo endpoint em Node para receber logs de Acesso e Operação da aplicação e salve os logs em arquivos no diretório /var/log em arquivos como xp_access.log por exemplo.**
 - **Crie também um registro de logs da execução do script de anonimização. Este não precisa ser via endpoint do Node por não estar fora da máquina Linux.**
 - **25% - Crie usuários específicos e individuais que serão utilizados para a escrita de logs dentro do diretório /var/log, executar o script de anonimização.**
 - **50% - Crie um README.md com a explicação de todos os recursos criados e configurados, e grave um vídeo de no máximo 6 minutos com a explicação completa sobre a estrutura desenvolvida.**
- O README.md será entregue junto com um vídeo de no máximo 6 minutos apresentando as soluções desenvolvidas e suas funcionalidades.**

TESTING, COMPLIANCE AND QUALITY.

PARTE A) Crie o plano de testes manuais, de validação no nível de sistema para seu projeto (as funcionalidades principais do seu sistema precisam estar cobertas), usando a plataforma **AZURE BOARDS**:

- 1) Liste os testes planejados – peso 20%
- 2) Descreva os dados de entrada para cada teste – peso 20%
- 3) Descreva os dados de saída para cada teste – peso 20%
- 4) Descreva o procedimento (passos) de teste aplicado – peso 20%

Obs: trabalhe com dados controlados nos seus testes, predefinindo os valores de variáveis para input e output esperado. Os testes precisam estar de acordo com o plano de release e tarefas de sprints realizadas. Esses itens serão checados e em caso de desrespeito às regras, será aplicada uma penalidade de 10% na nota.

PARTE B) Utilize um software de automação para validar o seu sistema. Pode ser usada automação Record & Playback com Selenium IDE ou Katalon Studio se o seu sistema tiver telas de interface de usuário. Se o seu sistema é composto somente por APIs e componentes de serviços sem interface de usuário, empregue **POSTMAN** ou software equivalente na automação:

Faça ao menos 4 casos de testes automatizados para a sua aplicação – peso (20%)

FORMA DE ENTREGA:

- Crie um repositório no **GIT HUB**, com acesso público e suba na Branch develop: Link de acesso ao **AZURE BOARDS** para correção dos testes manuais, sendo que o professor precisa ser membro da organização e do projeto para poder acessar e corrigir o trabalho;

Crie um repositório no **GIT HUB**, com acesso público e suba na Branch develop:

- a) Link de acesso ao **AZURE BOARDS** para correção dos testes manuais, sendo que o professor precisa ser membro da organização e do projeto para poder acessar e corrigir o trabalho;

- b) Link de um vídeo gerado pela equipe que mostre a configuração e execução dos testes de automação, nas ferramentas escolhidas.

- Entregas fora do formato implicarão em penalidade de 10% na nota.

CYBERSECURITY

Tarefa 1: SSDLC Automatizado e Codificação Segura – 3,0

- Descrição:

Com base nas vulnerabilidades identificadas por SAST, DAST e SCA na Sprint 3, implementar práticas de SSDLC (Secure Software Development Life Cycle) diretamente no pipeline de CI/CD. As práticas devem ser validadas automaticamente e aplicadas às funcionalidades críticas como Login e Cadastro.

- Ações recomendadas:
- Validação e sanitização de entradas com testes automatizados
- Autenticação segura com tokens, expiração e verificação de sessão
- Tratamento seguro de erros para evitar vazamento de informações
- Integração no CI/CD:
- Testes unitários e de segurança como pré-requisitos de build
- Regras de bloqueio para commits que violam padrões seguros

- Entregável:

Relatório técnico com as práticas de SSDLC aplicadas, evidências de validação automatizada e exemplos de código seguro.

CYBERSECURITY

Tarefa 2: Gerenciamento de Vulnerabilidades Contínuo – 4,0

- Descrição:
Estabelecer um processo automatizado de Gerenciamento de Vulnerabilidades, alinhado com os achados da Sprint 3. O processo deve ser integrado ao pipeline e incluir:
- Identificação automática via SAST, DAST e SCA
- Priorização baseada em CVSS, contexto do sistema e impacto
- Mitigação automatizada, como atualizações de dependências e correções de código
- Monitoramento contínuo com dashboards e alertas
- Ferramentas sugeridas: Snyk, SonarQube, OWASP ZAP, GitHub Security Alerts

CYBERSECURITY

Tarefa 3: LGPD & Conformidade com Segurança Automatizada – 3,0

- Descrição:

Desenvolver um plano de conformidade com a LGPD, incorporando controles de segurança automatizados no pipeline. O plano deve abordar:

- Consentimento rastreável via logs e registros de aceite
- Minimização de dados com validações de escopo e coleta
- Criptografia em trânsito e em repouso, validada por testes automatizados
- Controle de acesso baseado em papéis (RBAC) com auditoria contínua
- Direitos do titular com mecanismos automatizados de exclusão e portabilidade
- Integração no CI/CD:
- Testes de conformidade como parte do pipeline
- Alertas para coleta excessiva ou uso indevido de dados

- Entregável:

Relatório de conformidade com evidências técnicas, políticas aplicadas e validações automatizadas no ciclo DevSecOps.

C#

Desenvolva Swagger C# com

- **ASP.NET Core Web API e Entity Framework com CRUD completo (35%);**
- **Pesquisas com LINQ (peso 10%);**
- **Publicação em ambiente Cloud (15%);**
- **Endpoints conectando com outras APIs ou fonte de dados (APIs públicas, ChatGPT etc) (20%);**
- **Documentação do projeto (10%);**
- **Apresentar arquitetura em diagramas (10%).**

Métodos e funções estruturadas são esperadas para avaliação, legibilidade do código e estruturação. O código deverá estar em um versionador e um arquivo Readme;

Entregue o link de acesso ao seu plano de projetos na nuvem, garantir que o professor tenha acesso ao repositório.

Entregas fora das especificações implicarão em perda de 10% na nota de avaliação.

IOT & IOB

Objetivo: evoluir a POC da Entrega 3 para que esteja **integrada ao projeto principal** (de acordo com o Case escolhido pelo grupo). O reconhecimento facial precisa ser **incorporado de alguma forma prática à aplicação final**, ainda que de maneira simples.

Formas possíveis de integração (os alunos podem escolher a mais adequada ao Case):

- Se o Case envolve **Aplicações Web/Mobile/API** → reconhecimento facial pode ser usado para autenticação, registro de presença ou login de usuários.
- Se o Case envolve **Serviços de Dados/BI/IA** → reconhecimento facial gera dados para dashboards, relatórios ou análise de comportamento.
- Integração **mínima exigida**: o módulo de reconhecimento facial deve **enviar, registrar ou disparar algum tipo de ação/fluxo dentro da aplicação final**, provando a conexão entre os dois.

Demonstração (vídeo até 5 min) – 40 pontos

- Arquitetura geral do projeto final (visão dos módulos).
- Demonstração da integração: do reconhecimento facial até a resposta no sistema final.
- Mostrar comportamento em pelo menos **um cenário prático** (ex.: face reconhecida → ação registrada/exibida/iniciada).

Funcionalidade e Integração – 40 pontos

- Reconhecimento facial funcionando de forma consistente.
- Integração prática com a aplicação escolhida (não pode ficar isolado).

Repositório e Documentação – 20 pontos

- README atualizado com instruções completas para rodar a solução final.
- Explicação clara de como o reconhecimento facial está conectado com o restante da aplicação.

ATENÇÃO!!!

- A nota da challenge é a média das entregas e **TODAS** as disciplinas recebem essa nota.

BANCAS AVALIATIVAS

- **Obs:** Neste semestre, teremos duas bancas avaliativas com a empresa parceira. As datas serão enviadas pelo Scrum Master ao longo do semestre([Via Teams](#)), com antecedência, para que possam se organizar.