

Regulamento Escola de TI 2026

Objetivo geral

Entregar software de qualidade que resolvam os problemas mais relevantes de seus usuários, concebido, projetado e desenvolvido seguindo as boas práticas da engenharia de software.

Objetivos específicos

Conduzir o trabalho em equipe de modo respeitoso e colaborativo, promovendo o aprendizado, crescimento e maturidade de todos os seus membros.

Obter a participação ativa de todos os membros do grupo em todos os processos envolvidos, gerando evidências objetivas que permitam constatar esta participação.

Sobre os times

Os times serão formados livremente pelos alunos, e contarão com no mínimo 4 e no máximo 6 integrantes.

Uma vez formados, os times serão mantidos até o final do período letivo, não sendo permitidas trocas de times.

Sobre os projetos

Os times poderão propor seus próprios projetos (sujeito a revisão por parte dos professores) ou aceitar um dos projetos propostos pelos parceiros desta edição. Os projetos devem atender às seguintes restrições:

1. Devem possuir escopo amplo o suficiente para alocar o trabalho dos membros do time durante todo o período letivo.
2. Requisitos arquiteturais:
 - a. *Backend stateless REST*, com controle transacional persistência relacional ou não-relacional.

- b. Aplicação *mobile* e *frontend* web implementados em HTML5+Javascript com *framework* a critério dos times.
- c. Testes automatizados (unitários e de integração).
- d. Arquitetura e requisitos não funcionais bem definidos para um cenários de 1 milhão de requisições de escrita para cada 2 milhões de requisições de leitura
- e. Pelo menos um caso de uso que envolva o protocolo websocket
- f. Integração com uma API de pagamento em ambiente de teste/homologação (A integração fica a critério do time, Mercado Pago, abacate pay, stripe, etc)
- g. Deploy da aplicação (Plataforma fica a critério do time)

As equipes podem utilizar outras tecnologias complementares, adicionais àquelas aqui apresentadas, a seu critério. Geradores de código e *frameworks* que escondem grande parte da construção do software não são permitidos.

Sobre os artefatos de projeto

Os times deverão projetar seus produtos de software com base nas boas práticas da engenharia de software, conforme abordado em seus cursos de graduação, apresentando, no mínimo, os seguintes artefatos:

1. Diagramas de caso de uso e de classes completos.
2. Diagramas de sequência apresentando o esquema geral dos tipos de caso de uso previstos (por exemplo: cadastros básicos, processamentos em *batch*, integração de serviços, emissão de relatórios, etc.).
3. Diagramas de entidade/relacionamento.
4. Glossário dos termos do negócio.
5. System Design
6. Especificação completa dos casos de uso.
7. Critérios de aceitação ou cenários de teste.
8. *Mockups* das telas.
9. Documentação da API REST.

Os artefatos devem ser atualizados em paralelo com a implementação. Não serão admitidos artefatos desenvolvidos *a posteriori*.

Sobre o processo

Nesta edição foi sugerida a utilização de um processo baseado no Kanban, partindo de um *backlog* priorizado a partir do qual as atividades serão atribuídas aos membros da equipe e, no máximo a cada 2 (duas) semanas, os times devem conduzir um encontro para review e retrospectiva das entregas, com possível revisão do backlog.

Os times publicarão no Jira / Confluence todos os artefatos gerados nestas cerimônias, especialmente o resultado da retrospectiva e *review*.

Todo o código e demais artefatos versionados deverão ser gerenciados no Github oficial da Escola de TI.

Todas as tarefas iniciadas devem estar previamente registradas no Jira e, quando iniciadas, deverão ser executadas em uma *branch* separada, derivada de uma *branch* de desenvolvimento. O time deve definir sua estratégia de *merge/release* e aplicar integração contínua (Ex: Jenkins terá suporte oficial) em um servidor do próprio time.

As horas devem ser corretamente apontadas nas respectivas tarefas no Jira.

Os times serão avaliados pela qualidade de seu processo, cabendo a cada time elaborar as métricas, indicadores e processos complementares que julgarem adequado.

Sobre as avaliações

As avaliações serão realizadas com base nas apresentações internas e públicas do projeto, que formam a nota base, a partir da qual:

- i. será aplicado o fator de avaliação do feedback 360°, sendo a pontuação do membro melhor avaliado do time considerada 100% e as demais reduzidas proporcionalmente.
- ii. será deduzido o percentual de 0 a 25% conforme avaliação de implementação de CRUD completo considerando 0 para implementação completa, 15% para implementação parcial e 25% para implementação insuficiente.

Sobre Restrições

Fica restrito o uso abusivo de Inteligência Artificial, agentes de codificação e até mesmo uso de código de terceiros como templates, para o desenvolvimento dos projetos.

O Intuito da matéria é avaliar as capacidades de planejamento, desenvolvimento e execução de projeto de software por parte dos alunos. O uso de PI de terceiros sai do propósito da matéria. Times que não

apresentarem o conhecimento técnico prático necessário e que não demonstrarem o domínio do projeto serão levados à rebanca e em casos extremos, a reprova.

Como Funciona a Rebanca

Por mais que o trabalho seja desenvolvido em grupo, existem pontos de avaliação individual, como a avaliação 360 e nivelamentos. Sendo assim, se um indivíduo não obtiver uma nota média maior ou igual a 60, o time todo irá para a rebanca. Isso não significa que todos os integrantes estão em situação de reprova, porém, o trabalho é em equipe. As atividades devem ser distribuídas de forma que todos desempenhem suas atividades e comprovem suas habilidades para recuperar a nota.

Não-conformidades

- Normais (-0.5 a cada 3 Ncs)
 1. Não criar as branches no formato: "#1332 - efetuando implementações iniciais".
 2. Se os commit's não descreverem uma prévia do que está sendo commitado no referido push.
 3. Ao realizar o merge, não colocar a opção –no-ff.
 4. Inconsistência no apontamento de horas da semana (apontar no dia de atividade, nunca consolidado).
 5. Não alterar o status do issue conforme o trabalho é realizado.
 6. Não entregar as atividades no prazo combinado.
- Graves (-0.5 a cada NC)
 1. Não comparecer nas reuniões agendadas pelo time (máximo de faltas: 1 em 4).
 2. Trabalhar em uma issue sem o branch correspondente.
 3. Ao trabalhar em uma issue, não realizar commits/pushs a cada dia trabalhado.
 4. Não realizar as atividades combinadas com o time.
 5. Não anexar os arquivos às issues que envolvam artefatos que não sejam código-fonte.
 6. Criar issues a revelia (issues devem ser criados no planning ou, posteriormente, pelo scrum master).
 7. Não responder a avaliação 360º ao final de cada sprint
- Gravíssima (-1.0 a cada NC)
 1. Deliberadamente burlar o regulamento da Escola de TI para se auto-beneficiar.

- Imperdoável (perder a nota da equipe no bimestre)
 1. Não comparecer às reuniões bimestrais de apresentação do projeto.

Observações:

1. Não-conformidades apontadas indevidamente pelo auditor são consideradas GRAVES.
2. Não-conformidades não apontadas pelo auditor são consideradas GRAVES.

Reunião com os times → Feedback apresentação

Individual → 360 e avaliação individual

geral → explicar objetivo da avaliação 360, sem autoavaliação

Tarefas

- Atribuir % de perda para as avaliações individuais conforme regulamento
- levantar e avaliar as não conformidades conforme regulamento
- Avaliar peso das notas individuais e de grupo