

Gerenciamento de Configuração de Software

Prof. Daniel Callegari, 2025/2

Trabalho 1 – Prática com Git em Times

O trabalho consiste em observar os requisitos do trabalho, implementar uma solução seguindo os conceitos vistos em aula e produzir um relatório final.

O trabalho deverá ser realizado em times compostos por 8 a 10 membros. Cópias ou tentativas de fraude resultarão em nota zero para todos os envolvidos.

Data de entrega: 03/10/2025

Atividades

Cada time deverá:

1. Cada grupo deve criar um único repositório no GitHub. O repositório deve ser público e seu endereço deve ser copiado para o relatório.
2. Definir e descrever claramente o seu processo de *branching* (fluxo de trabalho), com base em um dos fluxos estudados.
3. Descrever as regras do fluxo de trabalho no relatório e segui-las ao longo da implementação.
 - a. O professor verificará a participação de cada membro do grupo através do log do git no repositório do time. Cada aluno deverá participar de pelo menos duas *features* ou correções de *bug*, com uma contribuição substancial.
 - b. As submissões ao repositório no GitHub deverão ocorrer via *Pull Requests*.
 - c. Os movimentos no repositório deverão acontecer até no máximo a data de entrega. Serão desconsiderados os trabalhos que tiverem movimentos depois da data de entrega.
4. Escrever o software a partir do repositório inicial criado de acordo com os requisitos e com o fluxo de trabalho definido pelo time.
5. Entregar um relatório via Moodle até a data limite. O relatório deverá conter:
 - a. Folha de rosto com nomes completos dos componentes do time e seus nomes de usuário no GitHub.
 - b. Link para o repositório público no GitHub.
 - c. Descrição clara do fluxo de trabalho adotado (pode ser uma variação dos fluxos vistos em aula, adaptado pelo time).
 - d. Demonstração de que o time seguiu o fluxo definido. Deve conter capturas de tela e descrições que deixem isto claro. (sugere-se capturar o Network graph do GitHub: Insights/Network)
 - e. Tabela com os nomes dos componentes do grupo e suas contribuições (ex. links para commits ou pull-requests registrados no GitHub).
 - f. Conclusão do grupo contendo reflexões sobre as dificuldades encontradas, como foram superadas e quais as lições aprendidas. A conclusão deverá conter uma nota de autoavaliação do grupo (0,0 a 10,0). Também deverá listar e comentar claramente a contribuição de cada aluno no trabalho.

Critérios de Avaliação

Avaliação: 30% da nota é resultado da contribuição individual e 70% da nota advém do resultado do grupo. Cada aluno deverá garantir que seu nome/usuário conste (em bom número de ocorrências) nos *commits* e *linhas de código* do histórico do git, confirmando a sua participação no processo de desenvolvimento, seguindo o fluxo de trabalho definido pelo grupo.

Dica: utilize o comando abaixo para uma métrica básica por usuário:

```
git shortlog -s -n --all --no-merges
```

Item	Pontos
Descrição clara do fluxo de trabalho adotado pelo grupo -- nomes de branches, regras de uso, comandos utilizados, exemplos, etc.	3,0
Demonstração de que o grupo seguiu o fluxo de trabalho definido – capturas de tela, descrições, anotações sobre trechos de código, log do git, etc.	4,0
Qualidade geral do relatório, em atendimento os itens listados no item Atividades.	2,0
Conclusão com autoavaliação do grupo.	1,0
TOTAL	10,0

Requisitos Gerais

1. Deseja-se um **sistema para gerenciar eventos**.
2. O sistema deverá ser implementado em **Java Console** /ou/ **Web Puro** (apenas HTML+CSS+Javascript). Não utilizar frameworks ou outras dependências. A ideia é ter a base de código o mais simples possível. Cuidar para que **todos** os membros do time conheçam a(s) linguagem(s) escolhida(s).
3. Não implementar um mecanismo de persistência de dados. O sistema deverá manter dados apenas em memória durante a execução.
4. Não implementar um mecanismo de login.
5. O sistema deverá iniciar com dados já preenchidos (em bom número e de boa qualidade), de forma a facilitar os testes.

Detalhamento

Precisa-se de um sistema para gerenciar eventos, ingressos e participantes.

Para cada evento, deseja-se armazenar o nome do evento, a data de realização, o valor do ingresso e o nome do responsável. Além disso, ao criar um evento, deve-se especificar quantos ingressos estarão disponíveis, reservando 15% para casos especiais. Os ingressos deverão ter uma numeração específica, detalhada no item a seguir.

Cada ingresso vendido deve ser nominal, ou seja, deverá ficar associado a um participante e não poderá ser alterado. Para cada participante deverá ser armazenado o nome completo e o CPF.

Funcionalidades desejadas

- 1) O sistema será operado apenas pelo operador principal. Não é necessário login e senha.
- 2) O sistema deverá permitir ao operador cadastrar novos eventos. Deve-se validar que a data é futura (não se pode cadastrar um evento no dia atual ou no passado). Todos os campos são obrigatórios.
 - a) Quando um evento for cadastrado, o sistema deverá atribuir um código único ao evento.
 - b) Ao cadastrar um evento, o usuário deverá informar quantos ingressos estarão disponíveis.
 - i) O sistema deverá reservar 15% dos ingressos disponíveis para casos especiais (ex. cadeirantes, deficientes físicos e outras situações).
 - ii) Os ingressos deverão ser identificados como {cod-evento}{seq-ingresso}{E}. Por exemplo, para um evento cujo código único é 121, com 100 lugares, os ingressos deverão ser identificados como 121-001; 121-002; 121-003 ... até 121-074 e mais os 15% especiais, identificados como 121-075E, 121-076E ... até 121-100E.
- 3) O sistema deverá permitir ao operador listar todos os eventos
- 4) O sistema deverá permitir ao operador procurar por um evento específico (por parte do nome).
- 5) Depois que o operador abrir os detalhes de um evento, poderá emitir ingressos (dentro do limite especificado na criação do evento)
 - a) O operador deverá indicar o tipo de ingresso (normal ou especial)
 - b) Os ingressos deverão ser emitidos sequencialmente (1, 2, 3 etc e respeitando a quantidade de ingressos normais e especiais)
 - c) Não poderão ser emitidos mais ingressos do que o limite do evento.
- 6) O sistema deverá permitir ao operador (no dia do evento) registrar a entrada de cada participante mediante a apresentação do ingresso. Para cada participante, o sistema deverá registrar a presença, de modo que posteriormente possamos saber quem esteve e quem não esteve no evento (o ingresso foi emitido, mas o participante não compareceu).
- 7) O sistema deverá permitir ao operador consultar os detalhes de um evento específico, incluindo:
 - a) Número total de ingressos disponíveis (lotação máxima)
 - b) Número de ingressos de cada tipo (normal e especial – 15%)
 - c) Número de ingressos vendidos de cada tipo e respectivos percentuais
 - d) Percentual de ocupação total do evento (total de vendidos vs lotação máxima).
- 8) O sistema deverá permitir ao operador gerar um relatório geral por mês:
 - a) Escolher mês e ano
 - b) Mostrar cada evento naquele mês e naquele ano, bem como estatísticas de cada evento.
- 9) Duas funcionalidades a mais, à escolha do grupo.
 - a) Detalhar as funcionalidades no relatório.

Bom trabalho!