

# Regras de Negócio — Campeonato de Xadrez (MVP)

Arquitetura em camadas com MVC na borda • SOLID • DDD leve

## 1) Objetivo do Documento

Definir, de forma enxuta e inequívoca, as regras de negócio do MVP do sistema de campeonato de xadrez. Este documento serve como fonte de verdade para o domínio, independentemente de frameworks ou do banco de dados.

## 2) Escopo do MVP

- Listar turmas (público).
- Exibir ranking por turma (público).
- Admin (professor) gerencia turmas e alunos (CRUD).
- Admin lança pontos (vitória/empate/derrota/bônus/ajuste).
- Login com autenticação; autorização por perfil.

## 3) Papéis e Permissões

- **Público/Aluno:** leitura de turmas e ranking.
- **Admin (Professor):** CRUD de turmas/alunos; lançar pontos; ajustar regras.

## 4) Glossário (Domínio)

- **Ranking:** ordenação de alunos por total de pontos dentro da turma.
- **Lançamento:** registro que altera a pontuação de um aluno (com tipo e valor).
- **Agregado:** conjunto lógico de dados e regras sob uma raiz (Aluno).

## 5) Regras de Pontuação

- Vitória: soma +1,0 ao total do aluno.
- Empate: soma +0,5 ao total do aluno.
- Derrota: soma +0,0 (sem alteração de total).
- Bônus: soma valor positivo configurável.
- Ajuste: valor positivo ou negativo para correções manuais.
- Os valores podem ser configurados pelo Admin; recomenda-se congelar após o início do campeonato.

## 6) Ranking e Critérios de Desempate

- Ranking por turma ordenado por **totalPontos** (descendente).
- Desempate (sugestão): (1) maior número de vitórias; (2) quem atingiu a pontuação primeiro (timestamp do último ponto).

## 7) Regras por Entidade (DDD tático)

## 7.1 Aluno (Agregado raiz)

- Cada Aluno pertence a exatamente **uma** turma.
- **Invariantes:** totalPontos  $\geq 0$  (exceto quando 'ajuste' negativo for permitido explicitamente).
- Operações de negócio: lançar vitória/empate/derrota/bônus/ajuste atualizam total e contadores.

## 7.2 Turma

- Turma possui professor responsável (Admin).
- Estados: ativa/inativa; turmas inativas não aceitam lançamentos.

## 7.3 Lançamento de Pontos

- Tipos válidos: VITÓRIA, EMPATE, DERROTA, BÔNUS, AJUSTE.
- Cada lançamento registra: tipo, valor, aluno, autor (Admin) e data/hora.
- Efeito colateral: atualizar totalPontos e contadores do aluno na mesma transação.

## 7.4 Configurações

- Regras de pontuação e ordem de desempate são configuráveis pelo Admin.
- Mudanças após início do campeonato devem ser registradas e versionadas.

# 8) Casos de Uso (Negócio) — Resumo

- **Listar Turmas:** público e admin visualizam turmas ativas.
- **Ver Ranking da Turma:** público e admin; ordenação conforme regras.
- **Criar/Editar/Arquivar Turma:** apenas admin.
- **Criar/Editar/Remover Aluno:** apenas admin (sempre com turma atribuída).
- **Lançar Pontos:** admin seleciona turma → aluno → tipo/valor → confirma.
- **Definir Regras:** admin ajusta pontuação e desempate (preferencialmente antes do início).

# 9) Validações e Erros (de Negócio)

- Não permitir lançamento em turma inativa.
- Não permitir aluno sem turma.
- Não permitir valores de pontos fora das regras configuradas (exceto ajuste).
- Mensagens de erro claras ao usuário/admin (ex.: 'Turma inativa', 'Aluno sem turma').

# 10) Auditoria e Transparência

- Todo lançamento guarda autor (Admin) e timestamp.
- Histórico consultável por turma/aluno/intervalo.

# 11) Relatórios e Exportações (MVP)

- Exportar CSV do ranking da turma.
- Exportar CSV do histórico de lançamentos (opcional).

# 12) Segurança e Conformidade (Negócio)

- Apenas Admin pode criar/editar/remover turmas, alunos e lançamentos.
- Autenticação por login; autorização por perfil.
- Proteção contra duplicidade acidental: impedir duplo clique no lançamento.

### **13) Critérios de Aceite (Globais)**

- Ranking exibe alunos ordenados corretamente por totalPontos e desempate.
- Ao lançar pontos, o total do aluno é atualizado e registrável no histórico.
- Ações de admin são restritas e auditadas.

### **14) Onde Implementar as Regras (Arquitetura em Camadas + MVC + DDD)**

- **Domínio:** invariantes e cálculos (ex.: atualizar total de pontos do Aluno).
- **Aplicação:** orquestra casos de uso, transações e chamadas a repositórios.
- **Apresentação (MVC):** controllers recebem requisições e retornam JSON (sem regra).
- **Infra:** persistência, cache, logs, autenticação; sem regra de negócio.

### **15) Evolução Futura (Opcional)**

- Introduzir partidas/rodadas e cálculo automático de pontos.
- Eventos de domínio (ex.: PontosLancados) para auditoria e integrações.
- Relatórios avançados (histórico por período, dashboards).