Boa tarde, Danilo! Que pergunta excelente! Uma plataforma de acesso de alunos de robótica pode ser uma ferramenta super poderosa para engajar, acompanhar e desenvolver os alunos. Além de **controle de presença**, **relatórios** e **tarefas**, aqui vão outras funcionalidades que fariam uma grande diferença:

**📚 1. Trilhas de Aprendizagem Personalizadas**

* Organização por níveis: iniciante, intermediário, avançado.
* Caminhos de aprendizagem com desafios progressivos.
* Liberação de conteúdos conforme o aluno avança.

**🤖 2. Área de Projetos e Portfólio**

* Upload de fotos e vídeos dos robôs criados.
* Registro dos projetos com descrição, materiais usados e problemas resolvidos.
* Comentários e avaliações entre colegas (peer review).

**🎓 3. Materiais de Apoio Multimídia**

* Vídeos tutoriais, apostilas, slides.
* Links para simuladores (como Tinkercad, mBlock, etc).
* Conteúdo técnico dividido por temas: programação, mecânica, eletrônica, etc.

**🧠 4. Desafios e Missões Gamificadas**

* Missões com pontos e medalhas virtuais.
* Ranking por participação e solução de problemas.
* Sistema de conquistas (badges).

**📈 5. Dashboard para Mentores e Coordenadores**

* Acompanhamento da evolução individual e por turma.
* Indicadores: participação, entrega de tarefas, desempenho nos desafios.
* Alertas automáticos para alunos inativos.

**👥 6. Fórum ou Comunidade**

* Espaço para dúvidas, troca de ideias, sugestões de projetos.
* Integração entre turmas ou equipes.
* Moderação feita por mentores.

**🧪 7. Simulador Virtual de Robótica (se possível)**

* Permitir testar códigos ou lógicas sem precisar de um robô físico.
* Pode ser algo simples com blocos de programação ou integração com simuladores externos.

**📅 8. Agenda e Calendário Integrado**

* Datas de aulas, eventos, competições.
* Lembretes automáticos por e-mail ou notificação.

**📂 9. Biblioteca de Componentes e Tutoriais Técnicos**

* Banco de dados com explicações de sensores, atuadores, placas, etc.
* Instruções de montagem, esquemas elétricos.

**🔐 10. Controle de Acesso e Perfis Diferenciados**

* Aluno, mentor, coordenador, pais/responsáveis.
* Permissões específicas para cada tipo de usuário.

Se quiser, posso te ajudar a desenhar uma arquitetura básica dessa plataforma, ou até montar um MVP com alguma tecnologia que você já use (como Django, Flask, Node, etc). Me avisa!

Aliás, essa plataforma pode ser uma baita iniciativa para as equipes FRC também 👀

Quer que a gente pense junto nos módulos principais para organizar isso como um projeto real?

4o

**Você disse:**

Seria possível fazer com Java Spring Boot? Lembrando que ficará em um servidor local.

**O ChatGPT disse:**

Sim, Danilo, **é totalmente possível e super viável desenvolver essa plataforma com Java Spring Boot**, especialmente se ela ficará hospedada em um **servidor local** — isso te dá mais controle sobre o ambiente e os dados.

Aliás, o Spring Boot é excelente para esse tipo de sistema, e com a sua bagagem em Java, dá pra fazer um projeto robusto, modular e escalável. Vou te mostrar como isso poderia ser estruturado:

**🔧 Arquitetura Base (Spring Boot Local)**

**🗂️ Camadas recomendadas:**

* **Controller** – Endpoints REST para interação com o front-end.
* **Service** – Lógica de negócios (regras de presença, pontuação, progresso etc).
* **Repository** – Comunicação com banco de dados via Spring Data JPA.
* **Model** – Entidades do banco (Aluno, Turma, Projeto, etc).

**🧱 Módulos Funcionais sugeridos**

| **Módulo** | **Funcionalidades principais** |
| --- | --- |
| **Autenticação & Acesso** | Login, perfis (aluno, mentor, admin), permissões. |
| **Gestão de Turmas** | Criar/editar turmas, atribuir alunos e mentores. |
| **Presença** | Registro de presença por aula ou atividade. |
| **Tarefas & Desafios** | Criação de desafios, entrega de soluções, feedback. |
| **Projetos** | Upload de projetos com descrição e imagens. |
| **Relatórios** | Geração de relatórios de desempenho e participação. |
| **Trilhas de Aprendizagem** | Organização de conteúdo por níveis, com desbloqueio progressivo. |
| **Biblioteca de Conteúdo** | Upload/download de PDFs, vídeos e links úteis. |
| **Gamificação** | Pontuação, conquistas, ranking de alunos. |
| **Fórum/Discussões** | Sistema simples de postagens e comentários. |
| **Calendário** | Eventos, competições, datas de aula com lembretes. |

**🧰 Tecnologias e Dependências Úteis**

* **Spring Boot Starter Web** – API REST.
* **Spring Security** – Controle de autenticação/autorização.
* **Spring Data JPA + H2/PostgreSQL** – Persistência.
* **Thymeleaf (opcional)** – Se quiser uma interface server-side.
* **Lombok** – Para reduzir boilerplate.
* **Swagger** – Para documentar e testar a API localmente.
* **Actuator** – Monitoramento da aplicação local.

**🗄️ Banco de dados sugerido**

* **PostgreSQL** (se já estiver rodando localmente).
* **H2** para desenvolvimento rápido e testes.

**🖥️ Hospedagem Local**

Você pode rodar o Spring Boot como um serviço no servidor local (usando systemd no Linux, por exemplo), e expor via IP local para a rede da escola/lab.

Se quiser, posso te ajudar com um **esqueleto inicial** do projeto em Spring Boot com:

* login de usuários,
* cadastro de turmas e alunos,
* e registro de presença.

Aí vocês vão evoluindo em cima disso.

Quer que a gente comece esse MVP juntos? Posso gerar o código base agora.

--------------------------------- 09/04/2025 ----------------------------------------------------------------

**✅ Já implementado:**

* 📚 CRUD de Student, Team, User, Presence
* 📝 Registro de presença por aluno
* 🔍 Filtros por data e por turma
* 🔄 Evita duplicidade de presença por aluno/dia
* 📊 Relatório individual com percentual
* 📥 Exportação CSV (individual e por turma)

**🔧 Funcionalidades que podem ser adicionadas:**

**1. Relatórios avançados**

* 📈 Percentual de presença por **turma**
* 📅 **Calendário de presença** da equipe ou de um aluno
* 🟢 Relatórios por **mês**, **bimestre**, **ano** com comparação
* 📤 Exportar relatórios em **PDF**
* 📊 Ranking de alunos com mais presenças

**2. Funcionalidades administrativas**

* ✅ **Login** e autenticação de usuários (JWT)
* 🔐 Controle de acesso (ex: ADMIN, MENTOR)
* 🧠 Dashboard com indicadores para o mentor
* 🔄 Registro de múltiplas presenças por turma (marcar vários alunos de uma vez)

**3. Melhoria de estrutura**

* 🔄 Refatorar com **DTOs e Mappers**
* 📦 Organizar services em camadas mais claras (relatórios, autenticação, etc.)
* 🚨 Tratamento global de erros (exception handler)
* ✅ Testes unitários e de integração

**4. Funcionalidades extras (gamificação ou motivacional)**

* 🏆 Sistema de recompensas baseado em presença
* 📣 Notificações por e-mail quando o aluno faltar muito
* 📲 Integração com app mobile (futuramente)