

Projeto de Desenvolvimento Móvel

Relatório

Docente: Pedro Rosa

Integrantes:

Francisco Baptista, 20240788

Francisco Silva, 20241293

Nayuka Jasmin Malebo, 20220386

Yassin Latif, 20240626

GIT : <https://github.com/yxsin720/Projeto-Peerly.git>

Tarefas Principais

Divisão de tarefas					
Tarefa Principal	Francisco Baptista (20240788)	Francisco Silva (20241293)	Yassin Latif (20240626)	Nayuka Malebo (20220386)	Total por tarefa
Análise e Planeamento	25%	25%	25%	25%	100%
Modelação da Base de Dados	40%	10%	10%	40%	100%
Desenvolvimento Backend (Spring Boot)	50%	0%	0%	50%	100%
Desenvolvimento Frontend (Android/Kotlin)	0%	50%	50%	0%	100%
Integração API REST	40%	20%	20%	20%	100%
Testes e Validação	25%	25%	25%	25%	100%
Documentação Técnica	30%	20%	20%	30%	100%
Gestão de Projeto e Apresentações	25%	25%	25%	25%	100%

Palavras-chave

Tutoria online, educação digital, *matching* tutor-aluno, agendamento de sessões, Google Meet, aplicação móvel Android, API REST, aprendizagem colaborativa, ensino remoto, plataforma académica.

Abstract

A crescente necessidade de colaboração académica e de acesso facilitado a apoio entre pares evidencia a importância de soluções digitais orientadas para a aprendizagem colaborativa. O projeto Peerly consiste no desenvolvimento de uma aplicação móvel *peerto-peer* que visa aproximar estudantes com interesses e competências complementares, promovendo a troca de conhecimento de forma simples, segura e acessível.

A aplicação permite a criação de perfis personalizados, a identificação de áreas de interesse, o emparelhamento entre utilizadores e o agendamento de sessões de estudo. Para suportar a comunicação síncrona, o Peerly integra funcionalidades de chat em tempo real e videochamadas, recorrendo ao Google Meet, cujo link é gerado automaticamente pelo sistema no momento da marcação da sessão.

A solução proposta assenta numa arquitetura cliente-servidor, composta por uma aplicação Android e um backend baseado em API REST, responsável pela gestão de utilizadores, sessões e comunicação com serviços externos. O projeto tem como principal objetivo fomentar a cooperação académica, valorizar as competências individuais dos estudantes e criar uma rede digital de apoio à aprendizagem, aplicando conceitos fundamentais de engenharia de software e desenvolvimento de sistemas distribuídos.

Objetivos e Motivação

Motivação

A dificuldade em encontrar colegas com competências específicas limita a colaboração e o desenvolvimento académico. O Peerly surge como resposta a este desafio, promovendo uma comunidade digital que valoriza a partilha de conhecimento, incentiva a cooperação e fortalece o crescimento pessoal e coletivo. **Objetivos**

- Aproximar estudantes com interesses e necessidades comuns;
- Facilitar a troca de conhecimento de forma rápida, intuitiva e segura;
- Disponibilizar ferramentas de comunicação integradas (chat);
- Gerar automaticamente links de videochamada através do Google Meet;
- Contribuir para o desenvolvimento de competências individuais e para a criação de uma rede de apoio académico.

Público-Alvo

O público-alvo da Peerly é composto maioritariamente por estudantes do ensino básico, secundário e superior que procuram apoio académico personalizado e acessível, bem como por tutores independentes ou professores que desejam partilhar conhecimento de forma flexível e remunerada. A aplicação destina-se também a utilizadores com competências digitais básicas e acesso a dispositivos móveis, privilegiando uma experiência intuitiva e sem barreiras técnicas. Ao focar-se no contexto académico e na aprendizagem colaborativa, a Peerly visa criar uma comunidade digital inclusiva, onde tanto quem precisa de ajuda como quem pode oferecer apoio encontram um espaço estruturado, seguro e funcional para interagir, agendar sessões e comunicar em tempo real — tudo integrado numa única plataforma móvel.

Pesquisa de Mercado

A análise de mercado revelou a existência de diversas plataformas de tutoria e apoio educativo, cada uma com características distintas: Superprof atua como um marketplace de explicadores, focado principalmente na ligação entre alunos e professores pagos; Preply apresenta-se como uma plataforma internacional de tutoria online, com forte componente comercial e enfoque em aulas individuais; e TutorMe, que se especializa em sessões on-demand, frequentemente integradas em contextos institucionais.

A Peerly diferencia-se destas soluções por ser uma aplicação académica não comercial, centrada na simplicidade e na colaboração peer-to-peer entre estudantes. A sua proposta de valor assenta na integração nativa de ferramentas — como a geração automática de links de Google Meet —, na comunicação em tempo real unificada (chat e vídeo) e numa arquitetura moderna orientada a serviços, pensada para oferecer uma experiência fluida, segura e acessível, sem custos diretos para o utilizador.

Descrição da Solução

A Peerly é uma aplicação móvel Android desenvolvida para potenciar a aprendizagem colaborativa e o apoio académico entre pares. A solução estrutura-se em torno de três pilares fundamentais: perfis personalizados, onde cada utilizador define as suas áreas de conhecimento e interesse; sistema de matching inteligente, que emparelha alunos e tutores com base em competências complementares; e ferramentas de comunicação integradas, que unificam todo o fluxo de interação — desde a marcação até à realização da sessão.

O núcleo da plataforma permite o agendamento flexível de sessões de estudo, com visualização da disponibilidade dos tutores e seleção de horários em tempo real. Durante e após a sessão, os utilizadores dispõem de um chat integrado para troca de mensagens e partilha de materiais, bem como de videochamadas realizadas através do Google Meet. A integração com este serviço é totalmente automatizada: o backend gera e associa um link único de Meet a cada sessão no momento da sua criação, garantindo que tutor e aluno accedem à sala de vídeo de forma imediata e segura, sem necessidade de configurações manuais.

Deste modo, a Peerly centraliza e simplifica todo o processo de aprendizagem colaborativa — desde a descoberta de parceiros de estudo até à realização e follow-up das sessões —, oferecendo uma experiência coesa, moderna e orientada para a comunidade académica.

Guiões de Teste

Caso de Utilização 1 – Agendamento de Sessão (Core)

1. O aluno cria conta ou faz login;
2. Pesquisa um tutor por disciplina;
3. Visualiza o perfil do tutor;
4. Seleciona um horário disponível;
5. Agenda a sessão;
6. Recebe confirmação com link do Google Meet.

Caso de Utilização 2 – Avaliar a sessão

1. O estudante termina a sessão;
2. É redireccionado para o ecrã de avaliação;
3. Submete uma classificação e comentário.;

Caso de Utilização 3 – Comunicação em sessão

1. Aluno entra numa sessão agendada;
2. Acede ao chat da sessão;
3. Troca mensagens com o tutor; 4. Inicia a videochamada via Google Meet;
5. Sessão é concluída.

Requisitos Técnicos

A implementação do Peerly exige o cumprimento dos seguintes requisitos técnicos fundamentais:

Autenticação segura – Implementação de um sistema de autenticação baseado em tokens JWT (JSON Web Tokens), garantindo a segurança das credenciais e o controlo de acesso a recursos protegidos através do método email/password.

Backend RESTful – desenvolvimento de uma API RESTful com endpoints bem definidos, seguindo as convenções HTTP, permitindo a interoperabilidade com a aplicação móvel e futuros clientes;

Base de dados relacional – utilização de um sistema de gestão de bases de dados relacionais (MySQL) para persistência estruturada dos dados de utilizadores, sessões, mensagens e avaliações, assegurando ACID e integridade referencial;

Integração com Google Meet – consumo da Google Calendar API para a criação automática de eventos e geração de links de videochamada, associando cada sessão agendada a uma sala de Google Meet única;

Aplicação Android nativa – desenvolvimento da aplicação cliente em Kotlin, seguindo as boas práticas do Android SDK, com interface responsiva, suporte a diferentes densidades de ecrã e integração profunda com os serviços do sistema.

Arquitetura da Solução

A arquitetura do Peerly segue o modelo cliente-servidor, separando claramente as responsabilidades entre o frontend móvel e o backend, com os seguintes componentes principais:

Frontend (Cliente) – Aplicação Android nativa desenvolvida em Kotlin, responsável pela interface com o utilizador, apresentação dos dados, gestão da navegação e comunicação com a API REST. A aplicação consome os endpoints do backend e apresenta a informação de forma intuitiva e responsiva;

Backend (Servidor) – API REST desenvolvida em Spring Boot, que aloja a lógica de negócios, processa os pedidos dos clientes, gere a autenticação, coordena o agendamento de sessões e interage com a base de dados e serviços externos. É também responsável pela criação automática dos links de Google Meet através da integração com a Google Calendar API;

Base de Dados – Sistema de gestão de bases de dados relacional MySQL, que armazena de forma persistente toda a informação da aplicação: utilizadores, sessões, mensagens, avaliações, disponibilidades e relações entre entidades;

Serviços Externos – Principalmente a Google Calendar API / Google Meet, utilizada para gerar automaticamente links de videochamada no momento do agendamento de cada sessão, assegurando uma experiência de videoconferência imediata e sem configuração manual por parte dos utilizadores.

O backend atua como orquestrador central: recebe os pedidos da aplicação móvel, valida as permissões, executa a lógica de negócio, persiste os dados na base de dados, interage com a API da Google e devolve as respostas adequadas (incluindo o link da sessão Meet) ao cliente Android.

Tecnologias

A implementação do projeto recorre às seguintes tecnologias e ferramentas, selecionadas pela sua maturidade, desempenho e adequação aos objetivos técnicos:

Java / Kotlin – Linguagem principal para o desenvolvimento da aplicação Android (Kotlin) e do backend Spring Boot (Java);

Spring Boot – Framework para a criação rápida e eficiente do backend RESTful, com suporte nativo para segurança, gestão de dependências e integração com bases de dados;

MySQL – Sistema de gestão de bases de dados relacional, utilizado para armazenar toda a informação estruturada da aplicação de forma fiável e performante;

REST API (JSON) – Padrão arquitetural para comunicação entre o cliente e o servidor, utilizando JSON como formato de serialização de dados, garantindo interoperabilidade e leveza nas trocas de informação;

Google Meet API – Conjunto de serviços da Google (integrado via Google Calendar API) que permite a criação programática de eventos e a geração automática de links de videochamada para as sessões de tutoria;

GitHub – Plataforma de controlo de versões utilizada para gestão colaborativa do código fonte, branching, revisão de código e manutenção da documentação do projeto.

Enquadramento nas diversas Unidades Curriculares

O projeto Peerly encontra um enquadramento direto e prático nas unidades curriculares do 3.º semestre, funcionando como um elemento integrador que consolida aprendizagens técnicas e transversais.

Programação de Dispositivos Móveis, desenvolve-se a aplicação Android nativa com Kotlin e Jetpack Compose, responsável pela experiência do utilizador e consumo da API REST.

Programação Orientada a Objetos sustenta a implementação do *backend* em Spring Boot, onde se modela a lógica de negócio, se aplicam princípios SOLID e se disponibiliza a API RESTful.

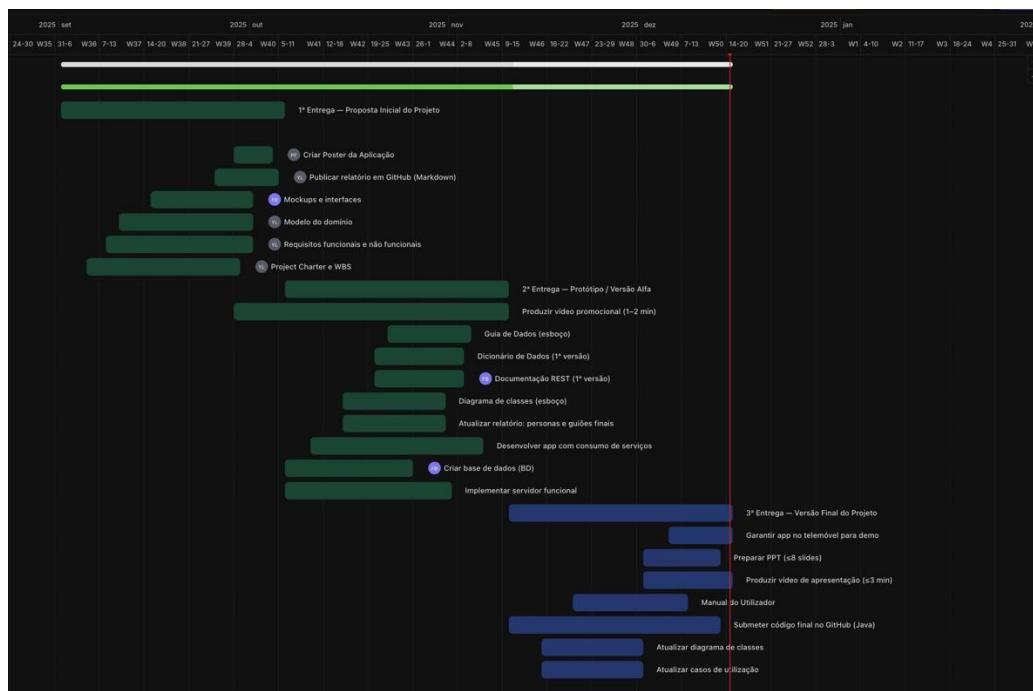
Bases de Dados assegura a modelação e operacionalização do sistema de persistência em MySQL, desde o desenho do modelo entidade-relação até à otimização de consultas e garantia de integridade.

Projeto de Desenvolvimento Móvel coordena a gestão ágil do ciclo de vida do software, utilizando ferramentas como ClickUp e GitHub,

Competências Comunicacionais apoiam a expressão clara e eficaz do projeto através de relatórios, *pitches*, *poster* e vídeo promocional.

Matemática Discreta contribui com conceitos de grafos e análise combinatória para otimizar o *matching* entre tutores e alunos e validar a gestão de horários

Planeamento



Conclusão

O projeto Peerly representa uma resposta concreta e tecnologicamente fundamentada aos desafios de colaboração académica identificados no contexto educativo contemporâneo. Ao desenvolver uma aplicação móvel centrada no conceito de aprendizagem *peer-to-peer*, o projeto visa não só facilitar a conexão entre estudantes e tutores, mas também consolidar competências técnicas essenciais nas áreas de engenharia de software, desenvolvimento móvel, sistemas distribuídos e segurança informática.

Os principais objetivos são entregar uma aplicação funcional e completa, que permita o registo e autenticação segura de utilizadores, a criação e gestão de perfis, o agendamento de sessões de estudo e a comunicação em tempo real via chat e videochamada;

Implementar com sucesso a integração automática com o Google Meet, garantindo que cada sessão agendada dispõe de um link de videochamada único e acessível sem intervenção manual;

Validar a usabilidade e a experiência do utilizador através de testes funcionais e de aceitação, assegurando que a plataforma é intuitiva, acessível e adequada ao público-alvo académico;

Documentar de forma clara e exaustiva todo o processo de desenvolvimento, desde a análise de requisitos até à arquitetura técnica, permitindo a reproduzibilidade, manutenção e eventual evolução do projeto;

Consolidar o trabalho em equipa e a aplicação de metodologias ágeis, gestão de versões com Git e boas práticas de desenvolvimento de software.

Autoavaliação da Implementação do Projeto

Objetivos Alcançados

Conseguimos implementar a funcionalidade core do sistema de forma satisfatória:

- Autenticação segura com JWT e suporte a registo/login;
- Perfis de utilizador com definição de papéis (estudante/tutor) e áreas de conhecimento;
- Sistema de matching básico que permite a estudantes encontrar tutores por disciplina;
- Agendamento de sessões com seleção de horários disponíveis;
- Integração com Google Meet – o link é gerado automaticamente e associado à sessão;
- Chat em tempo real durante as sessões agendadas;
- Backend RESTful com Spring Boot e base de dados MySQL operacional;
- Aplicação Android funcional com interface desenvolvida em Kotlin e Jetpack Compose.

A arquitetura cliente-servidor foi mantida conforme planeado, e a integração entre as diferentes camadas (frontend, backend, BD) foi implementada com sucesso. A documentação técnica (REST API, Dicionário de Dados, Manual de Utilizador) está completa e disponível no repositório.

Alterações em Relação à Proposta Inicial

Simplificação do algoritmo de matching – A versão inicial previa um sistema de recomendação mais complexo, com pesos múltiplos (competências, avaliações, proximidade temporal). Optámos por uma abordagem mais direta: filtragem por disciplina e disponibilidade, com ordenação por rating. A complexidade inicial revelou-se desproporcional para o tempo disponível.

Remoção do sistema de pagamentos integrado – A proposta incluía um módulo de pagamentos (price_per_hour_cents nas tabelas). Decidimos adiar esta funcionalidade para uma versão futura, focando-nos no core gratuito e académico da plataforma.

Chat apenas dentro de sessões agendadas – Inicialmente prevíamos um chat geral entre utilizadores. Limitámos a comunicação ao contexto de sessões, para simplificar a gestão de estados e notificações.

Disponibilidade por horários fixos – Em vez de um calendário interativo completo, os tutores definem blocos de disponibilidade semanal (weekday + time range). Esta simplificação reduziu a complexidade da interface e da lógica de agendamento.

Funcionalidades Não Implementadas

Sessões em grupo (`max_participants > 1`) – A base de dados suporta múltiplos participantes, mas a interface e a lógica de gestão de sessões grupais não foram desenvolvidas.

Sistema de notificações push nativo – As notificações permanecem apenas no backend (tabela notifications), sem integração com Firebase Cloud Messaging.

Integração OAuth com Google/Apple – A autenticação permanece por email/password. A tabela oauth_accounts foi criada, mas não foi implementado o fluxo OAuth.

Avaliações com comentários públicos – O sistema de reviews foi modelado, mas a interface de submissão e visualização não foi totalmente implementada.

Funcionalidade de "favoritos" – A tabela existe, mas não há UI para marcar/gerir tutores favoritos.

Dificuldades Enfrentadas

- Gestão do tempo – A coordenação entre 5 unidades curriculares com entregas paralelas foi desafiadora, especialmente na fase de integração entre backend e frontend.
- Complexidade da integração com Google Calendar API – A configuração de OAuth 2.0 e a gestão de tokens para criação automática de eventos foi mais trabalhosa que o previsto.
- Testes em dispositivos reais – A variedade de versões de Android e tamanhos de ecrã exigiu ajustes contínuos no layout.
- Conciliação de horários na equipa – A disponibilidade assíncrona para reuniões de alinhamento técnico atrasou algumas decisões de arquitetura.

Balanço Final

Apesar dos desvios e simplificações, consideramos que o objetivo principal foi alcançado: desenvolver uma aplicação funcional que resolve o problema de conexão entre estudantes e tutores, com agendamento integrado e videochamadas automáticas via Google Meet. A plataforma está operacional, documentada e demonstra a aplicação prática dos conhecimentos das diversas unidades curriculares.

O projeto Peerly representa, assim, uma base sólida e extensível para futuras evoluções, tendo cumprido os requisitos académicos e proporcionado uma aprendizagem valiosa em desenvolvimento full-stack, gestão de projeto e trabalho colaborativo

Bibliografia

GOOGLE DEVELOPERS. (2024). *Google Identity – Sign-In with Google*. Disponível em: <https://developers.google.com/identity>.

GOOGLE WORKSPACE DEVELOPERS. (2024). *Google Calendar API – Using OAuth 2.0 for Web Server Applications*. Disponível em: <https://developers.google.com/calendar/api/guides/auth>.

SPRING. (2024). *Spring Security Reference Documentation*. Disponível em: <https://docs.spring.io/spring-security/reference/index.html>.

JETBRAINS. (2024). *Kotlin Documentation*. Disponível em: <https://kotlinlang.org/docs/home.html>.

ANDROID DEVELOPERS. (2024). *Jetpack Compose*. Disponível em: <https://developer.android.com/jetpack/compose>.