

UTFPR-UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Bacharelado em Engenharia de Software - 7º Período

DISCIPLINA: *Oficina de Integração 2 - ES47C-ES71*

PROFESSOR: *Antonio Carlos Fernandes da Silva*

Documento de Projeto de Software

Portal ELLP - Módulo de Gestão de Escolas

Álison Christian Rebouças Vidal de Carvalho

João Pedro Correia Leite Moreira

Luccas Philot Gonçalves

Yuri Garcia Yoshida

Cornélio Procópio

2025

Sumário

PLANEJAMENTO.....	2
1 INTRODUÇÃO	2
1.1 CONTEXTO	2
1.2 JUSTIFICATIVA	2
1.3 PROPOSTA	2
1.4 ORGANIZAÇÃO DO DOCUMENTO.....	2
2 DESCRIÇÃO GERAL DO SISTEMA.....	4
2.1 OBJETIVOS (GERAIS E ESPECÍFICOS)	4
2.2 LIMITES E RESTRIÇÕES	4
2.3 DESCRIÇÃO DOS USUÁRIOS DO SISTEMA.....	4
3 DESENVOLVIMENTO DO PROJETO.....	5
3.1 METODOLOGIA DE DESENVOLVIMENTO	5
<i>Papéis no Scrum</i>	<i>5</i>
<i>Adaptação do Scrum no Projeto</i>	<i>5</i>
3.2. ARQUITETURA E TECNOLOGIAS	5
<i>Ambiente e Ferramentas de Apoio</i>	<i>6</i>
<i>Estrutura e Tecnologias do Frontend.....</i>	<i>6</i>
<i>Estrutura e Tecnologias do Backend.....</i>	<i>6</i>
<i>Banco de Dados.....</i>	<i>6</i>
3.3. ESTRATÉGIA DE AUTOMAÇÃO DE TESTES	6
<i>Frontend: Karma e Jasmine</i>	<i>6</i>
<i>Backend: Jest</i>	<i>7</i>
<i>Integração Contínua (CI/CD).....</i>	<i>7</i>
3.3 CRONOGRAMA PREVISTO	7
4 REQUISITOS DO SISTEMA.....	9
4.1 REQUISITOS FUNCIONAIS	9
4.2 DESCRIÇÃO DOS REQUISITOS FUNCIONAIS	9

Planejamento

1 Introdução

1.1 Contexto

O ELLP (Ensino Lúdico de Lógica e Programação) é um projeto de extensão da UTFPR, Campus Cornélio Procopio, coordenado pelo Prof. Antonio Carlos Fernandes da Silva. O principal objetivo do projeto é oferecer a crianças e adolescentes de escolas públicas, ONGs e creches acesso a conhecimentos muitas vezes não disponíveis em suas realidades, como informática básica, lógica, robótica e programação.

O projeto atende um público-alvo de crianças a partir de 10 anos e adolescentes até 16 anos, por meio de oficinas presenciais realizadas aos sábados na própria universidade. A execução das atividades conta com a participação de estudantes voluntários da UTFPR-CP, que atuam na criação de material, monitoria e na condução das oficinas.

1.2 Justificativa

Atualmente, o projeto ELLP gerencia a parceria com diversas instituições de ensino de forma manual, o que pode gerar dificuldades no controle de quais escolas, turmas e alunos estão sendo atendidos. A ausência de um sistema centralizado para gerenciar essas informações dificulta a organização, a emissão de documentos importantes como cartas convite e convênios, e a extração de dados consolidados sobre o alcance do projeto.

A crescente demanda e a necessidade de organizar as informações de forma eficiente tornam essencial o desenvolvimento de uma ferramenta dedicada, que automatize e padronize o controle das escolas participantes.

1.3 Proposta

Com o intuito de apoiar a gestão do projeto de extensão ELLP, será desenvolvido um módulo web para o controle de escolas parceiras. A aplicação permitirá que os coordenadores do projeto cadastrem e gerenciem as escolas, seus representantes, as turmas envolvidas e os alunos participantes. Além disso, o sistema oferecerá funcionalidades para a emissão de documentos, como cartas convite e termos de convênio, e para a visualização de relatórios de alunos atendidos por escola, otimizando a administração e a organização do projeto.

1.4 Organização do Documento

Este documento apresenta, de forma estruturada, a fase de planejamento do desenvolvimento do Módulo de Gestão de Escolas do Portal ELLP. O Capítulo 1 introduz o contexto do projeto de extensão, a motivação e os objetivos da solução proposta. O Capítulo 2 descreve o sistema a ser desenvolvido, incluindo seus objetivos, limitações e o perfil dos usuários. Em seguida, o Capítulo 3 aborda o desenvolvimento do

projeto, detalhando as tecnologias, a metodologia de trabalho e o cronograma. Por fim, o Capítulo 4 é dedicado a apresentar os requisitos funcionais do sistema.

2 Descrição Geral do Sistema

2.1 Objetivos (Gerais e Específicos)

Objetivo Geral: Desenvolver uma aplicação web para otimizar e centralizar o gerenciamento de escolas, turmas e alunos participantes do projeto de extensão ELLP, garantindo maior controle e organização das informações.

Objetivos Específicos:

- Permitir o cadastro, edição, visualização e exclusão de escolas parceiras.
- Gerenciar as informações dos representantes de cada escola.
- Controlar as turmas e os alunos de cada instituição que participam do projeto.
- Facilitar a emissão de documentos padronizados, como cartas convite e convênios.
- Disponibilizar filtros e relatórios para a consulta de alunos atendidos.
- Oferecer uma interface administrativa intuitiva para os coordenadores do projeto.

2.2 Limites e Restrições

A aplicação será desenvolvida exclusivamente para o ambiente web, com foco na compatibilidade com os principais navegadores modernos. Inicialmente, o sistema não contará com um portal público para as escolas; o acesso será restrito aos administradores do projeto ELLP. A aplicação exigirá conexão com a internet para seu funcionamento.

2.3 Descrição dos Usuários do Sistema

O sistema será utilizado por um perfil principal de usuário:

- **Equipe do Projeto (Administradores):** Responsáveis por alimentar o sistema, cadastrar as escolas, turmas, alunos, emitir documentos e manter todas as informações atualizadas.

3 Desenvolvimento do Projeto

3.1 Metodologia de Desenvolvimento

O projeto adotará a metodologia Scrum, um modelo ágil para o desenvolvimento de software. O trabalho será organizado em ciclos curtos, chamados de Sprints, permitindo que o sistema seja construído de forma incremental, com entregas frequentes de funcionalidades que podem ser avaliadas.

Esse modelo torna o processo mais eficiente, adaptável e colaborativo, pois permite que o sistema seja construído de forma incremental. Isso significa que o produto evolui por meio de entregas frequentes, cada uma representando uma parte já funcional e testável. Com isso, é possível receber feedback constante, corrigir problemas com agilidade e garantir que o projeto atenda de forma mais precisa às necessidades do cliente ou usuário final.

Papéis no Scrum

O Scrum define papéis específicos dentro da equipe:

- **Product Owner (PO):** é a pessoa responsável por representar os interesses do cliente, priorizar as tarefas e validar o que foi entregue.
No projeto, o PO é **Álison Christian Rebouças Vidal de Carvalho**.
- **Scrum Master:** atua como facilitador do time, garantindo que o processo do Scrum seja seguido e que não haja obstáculos impedindo o trabalho da equipe.
No grupo, esse papel é desempenhado por **João Pedro Correia Leite Moreira**.
- **Desenvolvedores:** são os membros responsáveis pela implementação prática das funcionalidades.
São eles: **Luccas Philot Gonçalves** e **Yuri Garcia Yoshida**.

Adaptação do Scrum no Projeto

O Scrum tradicional prevê cerimônias e reuniões que foram adaptadas para a realidade do grupo, cujos membros residem em diferentes localidades, tornando a comunicação presencial inviável. Desta forma, toda a interação e o alinhamento da equipe são realizados de forma remota:

- **Comunicação Contínua (via WhatsApp):** Para atualizações rápidas, esclarecimento de dúvidas pontuais e alinhamentos diários, a equipe utiliza o WhatsApp como principal canal de comunicação, garantindo que todos se mantenham sincronizados.
- **Reunião Semanal (domingos, online):** Aos domingos, a equipe realiza uma reunião mais estruturada, que combina as cerimônias de *Sprint Planning*, para definir as metas e tarefas da semana, e *Sprint Review/Retrospective*, para revisar o que foi entregue e discutir pontos de melhoria no processo.

Essa adaptação mantém o foco na entrega contínua e na organização, garantindo o avanço constante no desenvolvimento, mesmo com a dinâmica totalmente remota.

3.2. Arquitetura e Tecnologias

O sistema será desenvolvido com base em uma arquitetura de software moderna, caracterizada pela separação clara entre o Frontend, responsável pela interface do usuário, e o Backend, responsável pelo

processamento de dados e pela lógica de negócio. A comunicação entre essas duas camadas ocorre por meio de uma API RESTful.

Ambiente e Ferramentas de Apoio

- **Ambiente de Desenvolvimento:** A aplicação será desenvolvida no Visual Studio Code (VS Code), um ambiente de desenvolvimento integrado (IDE) gratuito e leve.
- **Prototipação (Figma):** As telas do sistema serão prototipadas no Figma, uma ferramenta online para o design de interfaces que permite colaboração em tempo real.
- **Gerenciamento de Tarefas (Trello):** Para organizar as tarefas e o progresso do projeto, será adotado o Trello, uma ferramenta baseada no método Kanban.
- **Modelagem (Astar UML):** Os diagramas utilizados para representar o sistema (classes, atividades, etc.) serão criados com o Astar UML.
- **Canais de Comunicação:** A comunicação será feita via WhatsApp para atualizações rápidas e Discord para reuniões e sessões de desenvolvimento.

Estrutura e Tecnologias do Frontend

A camada de interface do sistema será implementada utilizando o framework **Angular** em conjunto com **TypeScript**. Esse framework oferece uma estrutura organizada para o desenvolvimento de aplicações web, promovendo a reutilização de componentes. A estrutura será segmentada em diferentes módulos e pastas, incluindo um módulo `admin` para as páginas restritas, `guards` para a segurança da navegação e um diretório `shared` para centralizar elementos reutilizáveis.

Estrutura e Tecnologias do Backend

No lado do servidor, a aplicação será construída com **TypeScript** aliado ao framework **Express**. O Express simplifica a criação de servidores e a definição de rotas, sendo uma solução ideal para o desenvolvimento de APIs RESTful. O backend será estruturado em camadas (rotas, middlewares, controllers, services, repositórios e models) para garantir alto nível de coesão e baixo acoplamento.

Banco de Dados

O armazenamento e a manipulação dos dados serão realizados por meio do **PostgreSQL**, um sistema de gerenciamento de banco de dados relacional gratuito, robusto e reconhecido pela sua confiabilidade.

3.3. Estratégia de Automação de Testes

Para garantir a qualidade e a confiabilidade contínua do software, o projeto adotará uma estratégia de testes automatizados integrada a uma pipeline de CI/CD (Integração Contínua/Entrega Contínua).

Frontend: Karma e Jasmine

No frontend Angular, serão utilizados **Karma** e **Jasmine** para a execução de testes unitários. Essa combinação permite validar componentes e serviços de forma isolada. Será estabelecida uma **cobertura mínima de 80%** para garantir qualidade consistente na camada de apresentação.

Backend: Jest

No backend em Express com TypeScript, será empregado o **Jest**, uma ferramenta de alta performance voltada para testes unitários e de integração. O Jest possibilita executar validações rápidas e precisas sobre a lógica de negócio e os pontos críticos da API. Para esta camada, a **cobertura mínima de testes será de 90%**, reforçando a garantia de qualidade nas funcionalidades críticas.

Integração Contínua (CI/CD)

A estratégia prevê a geração automática de relatórios de cobertura de código, que serão integrados a uma pipeline de CI/CD. Esse fluxo garante que cada atualização no código seja automaticamente validada, permitindo identificar falhas rapidamente e evitando que erros cheguem ao ambiente de produção.

3.4 Cronograma previsto

Tarefa	Entrega	Responsáveis
Planejamento inicial (requisitos funcionais, arquitetura em alto nível, estratégia de testes, tecnologias, cronograma)	13/09	Todos (Álison Carvalho lidera documentação)
Estrutura do repositório (GitHub) e Kanban (Trello)	13/09	Luccas Philot
Setup inicial do projeto (Angular, Express, Postgres)	20/09	Yuri Yoshida e Luccas Philot
Prototipação do Front (layout e telas principais)	24/09	João Pedro
Estrutura inicial do Front (componentes base)	27/09	Yuri Yoshida
Login (Back, autenticação básica)	01/10	João Pedro
Login (Front)	01/10	Luccas Philot
Testes Login (funcionais e automatizados)	04/10	Yuri Yoshida (execução) e Álison Carvalho (documentação)
CRUD Escolas (Back)	08/10	João Pedro
CRUD Escolas (Front)	08/10	Luccas Philot
Testes CRUD Escolas (manuais + unitários Back)	11/10	João Pedro (Back), Yuri Yoshida (execução), Álison Carvalho (documentação)
CRUD Turmas (Back)	15/10	João Pedro

CRUD Turmas (Front)	15/10	Luccas Philot
Integração Front + Back (Escolas e Turmas)	18/10	Yuri Yoshida
Testes Integração (automatizados básicos + manuais)	20/10	Luccas Philot e João Pedro
Revisão Sprint 1 (documentação atualizada)	22/10	Álison Carvalho
Apresentação Sprint 1 (vídeo até 3 min)	25/10	Todos
Ajustes pós-feedback Sprint 1	27/10	Todos
CRUD Alunos (Back)	05/11	Álison Carvalho
CRUD Alunos (Front)	05/11	Luccas Philot
Integração Front + Back (Alunos)	08/11	Yuri Yoshida
Testes CRUD Alunos (unitários + manuais)	10/11	João Pedro (Back), Yuri Yoshida (execução), Álison Carvalho (documentação)
Emissão de Carta Convite/Convênio (Back)	15/11	João Pedro
Emissão de Carta Convite/Convênio (Front)	15/11	Luccas Philot
Testes Carta Convite/Convênio	17/11	Yuri Yoshida e Álison Carvalho
Controle de alunos por escola (relatórios simples)	20/11	João Pedro
Testes Relatórios (funcionais)	22/11	Yuri Yoshida
Ajustes finais no Front e responsividade	27/11	Yuri Yoshida e Luccas Philot
Testes finais de usabilidade + automação básica Front	29/11	João Pedro
Revisão final da Documentação (conclusão e fechamento)	03/12	Álison Carvalho
Ajustes pós-feedback final	05/12	Todos
Vídeo final (execução + funcionalidades)	06/12	Todos

4 Requisitos do Sistema

4.1 Requisitos Funcionais

Código	Descrição	Prioridade
RF01	O administrador deve ser capaz de realizar login e logout na plataforma.	Essencial
RF02	O administrador deve ser capaz de gerenciar escolas (cadastrar, visualizar, editar e excluir).	Essencial
RF03	O sistema deve permitir o cadastro de informações completas da escola (Nome, Endereço, Contato, etc.).	Essencial
RF04	O administrador deve ser capaz de gerenciar representantes (cadastrar, editar, excluir) e associá-los a uma escola.	Essencial
RF05	O administrador deve ser capaz de gerenciar turmas (cadastrar, editar, excluir) e associá-las a uma escola.	Essencial
RF06	O administrador deve ser capaz de gerenciar alunos (cadastrar, editar, excluir) e associá-los a uma turma.	Essencial
RF07	O sistema deve permitir a busca e a aplicação de filtros na listagem de escolas, turmas e alunos.	Importante
RF08	O sistema deve ser capaz de gerar um documento de "Carta Convite" ou "Convênio" em formato PDF com os dados da escola selecionada.	Importante
RF09	O administrador deve ser capaz de visualizar uma lista de todos os alunos atendidos por uma determinada escola.	Importante
RF10	O sistema deve possuir uma interface de navegação clara entre as áreas de gerenciamento (escolas, turmas, etc.).	Essencial
RF11	O sistema deve validar os dados de entrada nos formulários para evitar inconsistências (ex: campos obrigatórios).	Essencial
RF12	O administrador deve ser capaz de visualizar um painel (dashboard) com estatísticas gerais (ex: total de escolas, total de alunos).	Desejável

Tabela 01: Requisitos Funcionais do Sistema

4.2 Descrição dos Requisitos Funcionais

- **RF01 - Login e Logout:** O sistema deve possuir uma tela de autenticação onde o administrador insere suas credenciais (usuário e senha) para acessar a área restrita. Um botão de logout deve estar visível e funcional em todas as páginas administrativas.
- **RF02 - Gerenciamento de Escolas:** Na área administrativa, o sistema deve permitir ao administrador cadastrar novas escolas, visualizar uma lista com todas as já cadastradas, acessar os detalhes de cada uma para edição e, se necessário, removê-las do sistema.
- **RF03 - Cadastro de Informações da Escola:** O formulário de cadastro de escolas deve conter campos para informações essenciais, como nome da instituição, endereço completo, telefone e e-mail para contato.
- **RF04 - Gerenciamento de Representantes:** O sistema deve permitir que o administrador cadastre os dados de um representante (nome, cargo, contato) e o associe a uma escola previamente cadastrada no sistema.
- **RF05 - Gerenciamento de Turmas:** O administrador poderá criar turmas, definindo um nome ou código de identificação, e associá-las a uma escola, permitindo organizar os alunos participantes.
- **RF06 - Gerenciamento de Alunos:** O sistema deve permitir o cadastro de alunos com suas

informações básicas (nome, idade, etc.) e associá-los a uma turma específica de uma escola.

- **RF07 - Busca e Filtros:** As páginas que listam escolas, turmas e alunos deverão conter campos de busca e filtros para que o administrador possa encontrar registros específicos de forma rápida e eficiente.
- **RF08 - Emissão de Documentos:** O sistema deve oferecer uma funcionalidade que, a partir dos dados de uma escola selecionada, gere automaticamente um documento em PDF (Carta Convite ou Convênio) com as informações preenchidas.
- **RF09 - Controle de Alunos por Escola:** O sistema deve permitir que o administrador selecione uma escola e visualize um relatório ou uma lista com todos os alunos vinculados a ela, independentemente da turma.
- **RF10 - Navegação Clara:** A interface da área administrativa deve ser intuitiva, com um menu de navegação fixo ou de fácil acesso para que o administrador possa transitar entre as diferentes seções de gerenciamento.
- **RF11 - Validação de Dados:** Os formulários de cadastro (escola, turma, aluno, etc.) devem possuir validações para garantir que os dados essenciais sejam preenchidos e estejam no formato correto antes de serem salvos.
- **RF12 - Painel de Estatísticas (Dashboard):** A página inicial da área administrativa poderá exibir um painel com dados resumidos, como o número total de escolas parceiras, o total de turmas e a quantidade total de alunos atendidos pelo projeto.