

UTFPR - UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Bacharelado em Engenharia de Software - 7º Período

DISCIPLINA: *Oficina de Integração 2 - ES47C-IF66K*

PROFESSOR: *Antonio Carlos Fernandes da Silva*

Documento de Projeto de Software

Raiane Stefane Campos Correia

Diomenes de Araujo Marangoni

Matheus Shiune Murasaki

Leonardo Conde Feitosa

Cornélio Procópio

2025

1. Introdução

1.1 Contexto:

Este projeto é desenvolvido no âmbito da disciplina extensionista **Oficina de Integração 2**, no qual os alunos são incentivados a atender às necessidades reais do projeto de extensão **ELLP - Ensino Lúdico de Lógica e Programação**. A iniciativa visa aplicar os conhecimentos de engenharia de software na resolução de problemas práticos de gestão e organização de um projeto social e educacional.

1.2 Justificativa:

O controle de oficinas, o cadastro de participantes (professores, tutores e alunos), o acompanhamento de escolas parceiras e a emissão de documentos como convênios e certificados são processos que, se realizados manualmente, geram retrabalho, inconsistência de dados e dificuldade na geração de relatórios de impacto. A ausência de um sistema integrado compromete a eficiência e a escalabilidade do projeto.

1.3 Proposta:

Propõe-se o desenvolvimento de um **Sistema de Gestão Integrado** para automatizar e centralizar a administração do projeto ELLP. A solução será dividida em dois módulos principais:

1. **Módulo 1: Controle de Oficinas:** Focado na gestão acadêmica e logística das atividades, abrangendo o cadastro de oficinas, professores, alunos, controle de inscrições, frequência e a emissão automatizada de certificados.
2. **Módulo 2: Controle de Escolas Participantes:** Voltado para a gestão institucional, permitindo o cadastro de escolas, representantes, turmas, a geração de documentos formais (cartas convite e convênios) e o monitoramento do número de alunos atendidos por instituição.

Este sistema irá otimizar a organização, garantir a integridade das informações e fornecer uma base sólida.

1.4 Ferramentas:

O desenvolvimento do projeto será atualizado no github privado:

<https://github.com/LeoKondii/Projeto-Oficina-de-Integra-o-II>

Atribuição e acompanhamento de tarefas pelo quadro Kanban do Trello:

<https://trello.com/invite/b/68c0be63bea36e3279739379/ATTI884446f96e983c9255ea8602bfa4a0d040AF9B36/kanban-projeto-oi2>

2. Descrição Geral do Sistema

2.1 Objetivos (Geral e Específicos)

- **Objetivo Geral:**

Desenvolver um sistema de software integrado e centralizado para automatizar e otimizar a gestão administrativa e pedagógica do projeto de extensão ELLP - Ensino Lúdico de Lógica e Programação, garantindo a eficiência no controle das oficinas, no relacionamento com as escolas parceiras e na geração de dados confiáveis para o monitoramento das atividades.

- **Objetivos Específicos:**

- Módulo de Controle de Oficinas:

1. **Gerenciar cadastros:** Implementar o cadastro completo de professores e alunos.
2. **Organizar as oficinas:** Permitir a criação e o gerenciamento das oficinas, definindo temas, datas e responsáveis.
3. **Controlar participação:** Automatizar as inscrições e o registro de frequência dos alunos.
4. **Emitir certificados:** Gerar certificados digitais com código de validação para os concluintes.

- Módulo de Controle de Escolas Participantes:

1. **Cadastrar parceiros:** Centralizar o registro de escolas, representantes e turmas.

2. **Monitorar impacto:** Gerar relatórios para visualizar o número de alunos atendidos por escola.

2.2 Descrição das entidades do sistema

Módulo 1 (Entidades e Atributos):

1. Professor

- **id_professor** (Identificador único)
- **nome_professor** (Texto)
- **cpf** (Texto, único)
- **email** (Texto, único)
- **telefone** (Texto)
- **area_conhecimento** (Texto)
- **contrato** (Data)

2. Aluno

- **id_aluno** (Identificador único)
- **nome_aluno** (Texto)
- **ra** (Registro do Aluno, texto)
- **data_nascimento** (Data)
- **id_escola** (Referência à entidade Escola)
- **id_turma** (Referência à entidade Turma)

3. Oficina

- **id_oficina** (Identificador único)
- **tema** (Texto)
- **descricao** (Texto longo)
- **carga_horaria** (Numérico)
- **data_inicio** (Data)
- **data_fim** (Data)
- **status** (Texto - Ex: "Planejada", "Em Andamento", "Concluída")
- **id_professor_responsavel** (Referência à entidade Professor)

4. Inscrição (Entidade que relaciona participantes a uma oficina)

- **id_inscrição** (Identificador único)
- **id_oficina** (Referência à entidade Oficina)

- **id_aluno** (Referência à entidade Aluno)
- **frequencia** (Percentual, numérico)
- **status aprovacao** (Booleano - Aprovado/Reprovado)

5. Certificado

- **id certificado** (Identificador único)
- **id_inscricao** (Referência à entidade Inscrição)
- **codigo_validacao** (Texto, único e gerado automaticamente)
- **data_emissao** (Data)
- **link_pdf** (Texto)

Módulo 2 (Entidades e Atributos):

1. Escola

- **id_escola** (Identificador único)
- **nome_escola** (Texto)
- **cnpj** (Texto, único)
- **cep** (Texto)
- **endereco_completo** (Texto)
- **telefone** (Texto)
- **email_institucional** (Texto)
- **id_representante** (Referência à entidade Representante)

2. Representante (Contato na escola, ex: diretor ou coordenador)

- **id_representante** (Identificador único)
- **nome_representante** (Texto)
- **cargo** (Texto)
- **email** (Texto)
- **telefone_contato** (Texto)

3. Turma

- **id_turma** (Identificador único)
- **nome_turma** (Texto - Ex: "9º Ano A")
- **ano_serie** (Texto/Numérico)
- **periodo** (Texto - "Manhã", "Tarde", "Noite")
- **id_escola** (Referência à entidade Escola)

4. Documento

- **id_documento** (Identificador único)
- **id_escola** (Referência à entidade Escola)
- **tipo_documento** (Texto - "Carta Convite", "Termo de Convênio")
- **data_emissao** (Data)
- **status_documento** (Texto - "Gerado", "Enviado", "Assinado")
- **link_arquivo** (Texto)

2.3 Funcionalidades do sistema

Módulo 1:

- **Gestão de Pessoas:** Cadastrar, editar, consultar e excluir professores, tutores e alunos.
- **Gestão de Oficinas:** Criar novas oficinas, definir seus temas, professores responsáveis, carga horária e datas.
- **Gerenciamento de Inscrições:** Inscrever alunos em oficinas, seja individualmente ou em lote (por turma).
- **Controle de Frequência:** Registrar a presença dos alunos em cada encontro da oficina para calcular a frequência final.
- **Emissão de Certificados:** Gerar automaticamente certificados digitais para os alunos que atingirem os critérios de conclusão.

Módulo 2:

- **Gestão de Escolas:** Cadastrar, editar e consultar informações das escolas parceiras.
- **Gestão de Turmas:** Cadastrar as turmas de cada escola que estão envolvidas no projeto.
- **Geração de Documentos:** Criar documentos padronizados, como cartas convite e termos de convênio, preenchendo automaticamente os dados da escola.
- **Controle de Atendimento:** Gerar relatórios para visualizar o número de alunos atendidos por escola, turma ou período.

3. Requisitos funcionais

RF01	O sistema deve permitir o cadastro de Professores.	Essencial
RF02	O sistema deve permitir o cadastro de Alunos, associando-os a uma Escola e a uma Turma.	Essencial
RF03	O sistema deve permitir a associação entre Alunos e Escola	Essencial
RF04	O sistema deve permitir a associação entre Alunos e Turma	Essencial
RF05	O sistema deve permitir o cadastro de Oficinas.	Essencial
RF06	O sistema deve permitir a associação de Professores a Oficinas	Essencial
RF07	O sistema deve permitir inscrever um ou mais Alunos em uma Oficina.	Essencial
RF08	O sistema deve fornecer uma interface para registrar a frequência dos alunos por oficina.	Desejável
RF09	O sistema deve gerar automaticamente um Certificado em PDF com um código de validação único para alunos aprovados.	Desejável
RF10	O sistema deve permitir a consulta de todos os participantes de uma determinada oficina.	Essencial
RF11	O sistema deve permitir listar todas as oficinas que um aluno participou.	Desejável
RF12	O sistema deve permitir o cadastro completo de Escolas, incluindo seus dados de contato e endereço.	Essencial
RF13	O sistema deve permitir o cadastro de um Representante	Essencial
RF14	O sistema deve permitir associar um Representante a uma Escola	Essencial

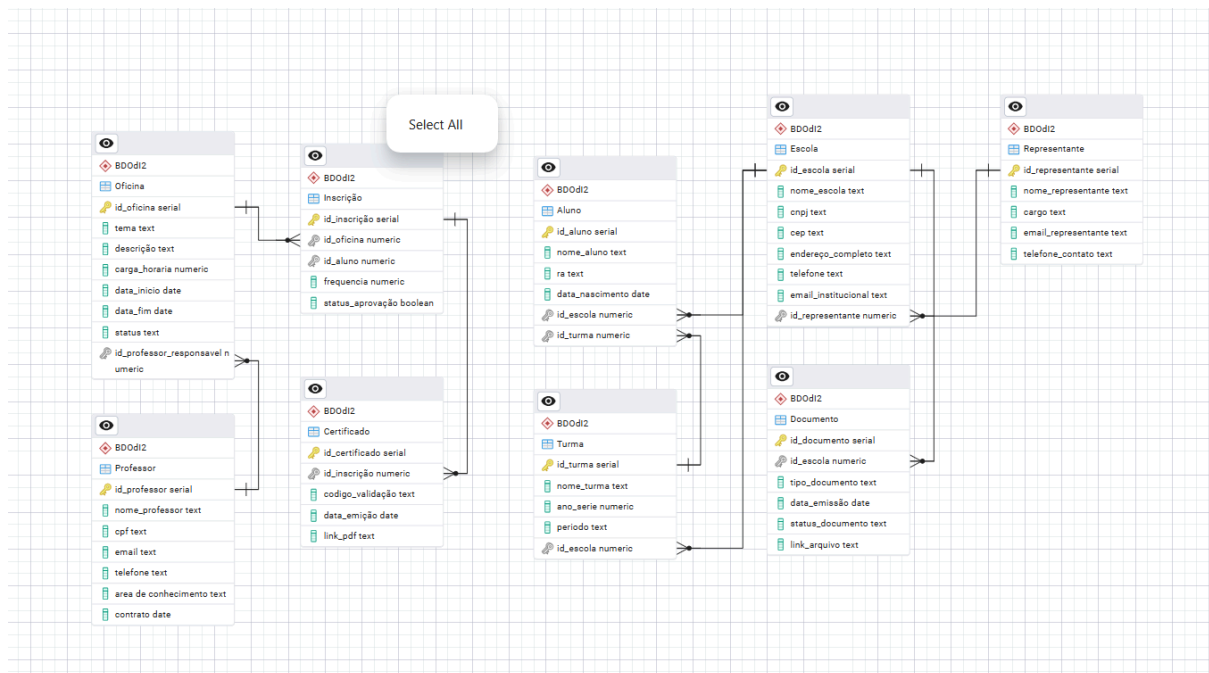
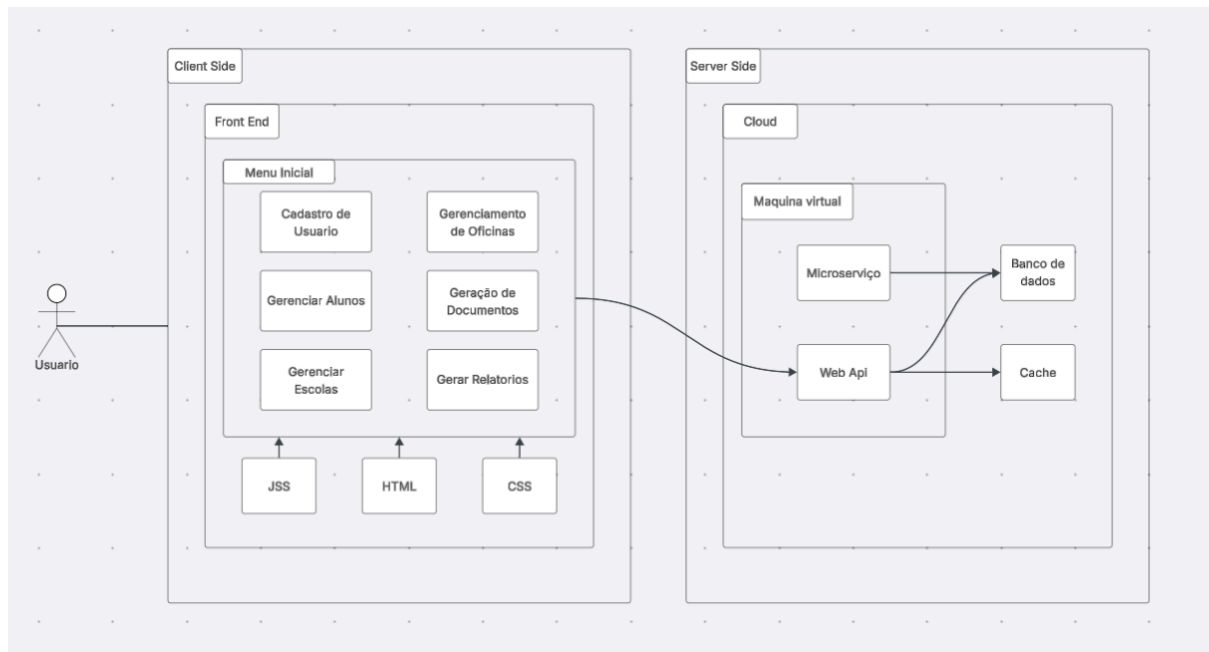
RF15	O sistema deve permitir o cadastro de Turmas	Essencial
RF16	O sistema deve permitir associar um ou mais Turmas a uma Escola	Essencial
RF17	O sistema deve possuir templates para gerar Cartas Convite e Termos de Convênio em formato PDF.	Desejável
RF18	Ao gerar um documento, o sistema deve preencher automaticamente os dados da Escola selecionada.	Desejável
RF19	O sistema deve permitir o controle do status dos documentos	Essencial
RF20	O sistema deve exibir um painel (dashboard) com o número total de alunos atendidos por cada escola.	Desejável

Detalhes dos requisitos funcionais:

- **RF01-Cadastro dos Professores:** O sistema deve permitir que os Professores possam registrar seus dados no site
- **RF02-Cadastro dos Alunos:** O sistema deve permitir que os Alunos possam registrar seus dados no site
- **RF03-Associação de Alunos e Escola:** O sistema deve permitir a associação entre Alunos e Escola
- **RF04-Associação de Alunos e Turmas:** O sistema deve permitir a associação entre Alunos e Turmas
- **RF05-Cadastro de Oficinas:** O sistema deve permitir o cadastro de oficinas dentro do site
- **RF06-Associação de Professores e Oficinas:** O sistema deve permitir a associação de Professores a Oficinas
- **RF07-Inscrição de Alunos em Oficinas:** O sistema deve permitir inscrever um ou mais Alunos a uma Oficina
- **RF08-Interface de frequência de Alunos:** O sistema deve gerar uma interface para registrar as frequências dos Alunos em uma Oficina

- **RF09-Criação de Certificado:** O sistema deve gerar automaticamente um Certificado em PDF com um código de validação único para alunos aprovados (ex: com frequência $\geq 75\%$).
- **RF10-Consulta de Oficina:** O sistema deve permitir a consulta de todos os participantes de uma determinada oficina.
- **RF11-Histórico de Aluno:** O sistema deve registrar o histórico de todas as oficinas que um Aluno participou.
- **RF12-Cadastro de Escolas:** O sistema deve permitir o cadastro de Escolas no site.
- **RF13-Cadastro de Representante:** O sistema deve permitir o cadastro de Representantes no site.
- **RF14-Associação de Representante e Escola:** O sistema deve permitir a associação de um Representante por Escola.
- **RF15-Cadastro de Turmas:** O sistema deve permitir o cadastro de Turmas no site.
- **RF16-Associação de Turmas e Escola:** O sistema deve permitir associar uma ou mais turmas a uma Escola.
- **RF17-Geração de documentos digitais:** O sistema deve permitir a criação de templates para gerar Cartas Convite e Termos de Convênio em formato PDF.
- **RF18-Preenchimento automático:** O sistema deve preencher automaticamente os dados da Escola selecionada ao gerar um novo documento.
- **RF19-Rastreabilidade de documento:** O sistema deve permitir o controle do status dos documentos (ex: se um convênio foi enviado ou assinado).
- **RF20-Rastreabilidade de Alunos por Escola:** O sistema deve exibir um painel (dashboard) com o número total de alunos atendidos por cada escola.

4. Arquitetura em alto nível do sistema



5. Estratégia de automação de testes do sistema

A estratégia ideal segue a pirâmide de testes, que organiza os tipos de testes por profundidade e custo de execução:

Tipo de Teste	Objetivo	Ferramentas
Unitário	Validar métodos e classes isoladamente	xUnit, NUnit
Integração	Testar comunicação entre componentes	Microsoft.AspNetCore.Mvc.Testing, TestServer
End-to-End (E2E)	Simular o fluxo completo do usuário na aplicação	Selenium

1. Testes Unitários

- Foco na camada de Model e Controller
- Cobertura de regras de negócio, validações e lógica de fluxo
- Mocking de dependências com bibliotecas como Moq

2. Testes de Integração

- Validação da interação entre Controllers e Services
- Testes com banco de dados em memória (ex: SQLite in-memory)
- Verificação de rotas, respostas HTTP e status codes

3. Testes E2E

- Automação da interface HTML/CSS com simulação de usuário
- Verificação de elementos visuais, formulários e navegação
- Execução em múltiplos navegadores com Selenium WebDriver

Integração com CI/CD

- Configurar pipelines no GitHub Actions, Azure DevOps ou GitLab CI
- Executar testes automaticamente a cada commit ou pull request
- Gerar relatórios de cobertura e falhas para análise contínua

6. Tecnologias utilizadas no projeto

- **Gerenciamento e Colaboração**

Trello: Plataforma de gerenciamento de projetos e tarefas. Permite organizar fluxos de trabalho, atribuir responsabilidades e acompanhar o progresso em tempo real.

Meet (Google Meet): Ferramenta de videoconferência online da Google, usada para reuniões e chamadas em grupo.

Google Drive: Serviço de armazenamento em nuvem que permite guardar, compartilhar e colaborar em arquivos.

- **Desenvolvimento e Programação**

Visual Studio: Ambiente de desenvolvimento integrado (IDE) completo, especialmente usado para desenvolvimento em C#.

Visual Studio Code (VSCode): Editor de código leve e altamente customizável, para outras linguagens e desenvolvimento visual.

C#: Linguagem de programação moderna e orientada a objetos.

JavaScript: Linguagem de programação voltada para desenvolvimento web.

HTML: Linguagem de marcação usada para estruturar páginas da web.

CSS: Linguagem usada para estilizar elementos HTML.

SQL Server: Sistema de gerenciamento de banco de dados.

- **Controle de Versão**

GitHub: Plataforma de hospedagem de código que usa Git para controle de versão. Permite colaboração em projetos, controle de alterações e integração contínua.

7. Cronograma

Data	Objetivos
15/09 à 20/09	Início da configuração Implementação da arquitetura e execução do projeto na IDE (Visual Studio). Download dos pacotes a serem implementados / Testes.
21/09 à 27/09	Criação e relacionamento de banco de dados / Testes.
28/09 à 04/10	Criação das classes e domínios na IDE.
05/10 à 11/10	Criação da API a ser utilizada, com nomeação das chamadas públicas e autorização dos usuários / Testes.
12/10 à 18/10	Criação das controllers e funções a serem utilizadas no projeto / Testes.
19/10 à 25/10	Review de backlog e correção de problemas.
26/10 à 01/11	Definição da view de Dashboard. Entrega da primeira sprint.
02/11 à 08/11	Review da sprint. Atualização de cronograma. Definição das views de cadastros.
09/11 à 15/11	Implementação das views para gerar o certificado/ Testes.
16/11 à 22/11	Criação dos modelos de relatórios a serem implementados.
23/11 à 29/11	Implementação das views e do modelo de negócio dos relatórios / Testes.
30/11 à 06/12	Elaboração de testes em escala e níveis mais elaborados.
07/12 à 12/12	Review geral do backlog do projeto. Entrega da segunda sprint.