

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

GUSTAVO HENRIQUE SALINA

**ABORDAGEM EM LINHAS DE PRODUTO DE SOFTWARE APLICADA AO
GERENCIAMENTO DE ATIVIDADES ESCOLARES: UMA INSTANCIÇÃO
VOLTADA PARA ALUNOS COM DEFICIÊNCIA INTELECTUAL**

PONTA GROSSA

2022

GUSTAVO HENRIQUE SALINA

**ABORDAGEM EM LINHAS DE PRODUTO DE SOFTWARE APLICADA AO
GERENCIAMENTO DE ATIVIDADES ESCOLARES: UMA INSTANCIÇÃO
VOLTADA PARA ALUNOS COM DEFICIÊNCIA INTELECTUAL**

**A software product line approach applied to school activities management: an
extension aimed at students with intellectual disabilities**

Trabalho de Conclusão de Curso de graduação
apresentado como requisito para obtenção do título
de Bacharel em Ciência da Computação da
Universidade Tecnológica Federal do Paraná
(UTFPR).

Orientadora: Prof. Dra. Simone Nasser Matos

PONTA GROSSA

2022



Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

GUSTAVO HENRIQUE SALINA

**ABORDAGEM EM LINHAS DE PRODUTO DE SOFTWARE APLICADA AO
GERENCIAMENTO DE ATIVIDADES ESCOLARES: UMA INSTANCIÇÃO
VOLTADA PARA ALUNOS COM DEFICIÊNCIA INTELECTUAL**

Trabalho de Conclusão de Curso de graduação
apresentado como requisito para obtenção do título
de Bacharel em Ciência da Computação da
Universidade Tecnológica Federal do Paraná
(UTFPR).

Data de aprovação: 9/junho/2022

Profa. Simone Nasser Matos
Doutora
Universidade Tecnológica Federal do Paraná

Profa. Eliana Cláudia Mayumi Ishikawa
Doutora
Universidade Tecnológica Federal do Paraná

Prof. Vinícius Camargo Andrade
Mestre
Universidade Tecnológica Federal do Paraná

PONTA GROSSA

2022

Dedico este trabalho aos meus pais.

AGRADECIMENTOS

Agradeço aos meus pais pelas lições durante a vida e pela oportunidade de estudar.

Agradeço a minha namorada, pois sem seu apoio tenho certeza que não estaria onde estou hoje em dia.

Agradeço a minha orientadora Profa. Dra. Simone Nasser Matos, pela paciência e ensinamentos valiosos que obtive durante este trabalho.

Agradeço aos meus amigos de faculdade que tornaram a trajetória mais leve e divertida.

Agradeço a mim mesmo pela dedicação e tempo investido no trabalho de conclusão de curso. Hoje em dia vejo que a caminhada valeu a pena.

Por fim, agradeço a você que está lendo meu trabalho. Espero que de alguma forma ele seja útil e enriquecedor.

*Do not go gentle into that good night
Old age should burn and rave at close of
day. Rage, rage against the dying of the
light.*
(THOMAS, 1947)

RESUMO

Este trabalho utilizou a abordagem de linhas de produto de software (LPS) para modelar um sistema de gerenciamento de atividades em que foram analisados três produtos no domínio, sendo dois gratuitos, *Moodle* e *Google Classroom*, e o outro é o funcionamento não informatizado do controle de atividades de uma escola da região, modalidade de educação especial para pessoas com deficiência intelectual. Um sistema de gerenciamento de atividades escolares é fundamental para acompanhar o aprendizado dos alunos, para alunos com DI, é importante adaptar o sistema e adequá-lo com modificações para atendê-los de forma inclusiva. Para o trabalho foram estudadas as principais metodologias de LPS na literatura, além de uma metodologia híbrida que combina características de três outras. Não foram encontradas instanciações para alunos com DI utilizando LPS na literatura, por isso, a metodologia híbrida foi utilizada para o desenvolvimento de um sistema de gerenciamento de atividades para alunos com DI. A LPS criada foi instanciada para o sistema da escola que trabalha com alunos com deficiência intelectual. O sistema gerado para a escola foi implementado para rodar na plataforma *mobile*. Como resultado, obtivemos um modelo robusto para atender implementações futuras de um mesmo domínio.

Palavras-chave: Linhas de produto; gerenciamento de atividades; deficiência intelectual.

ABSTRACT

This work used the software product line approach (SPL) to model an activity management system in which three products in the domain were analyzed, two being free, Moodle and Google Classroom, and the other a non-computerized management of activities in a school of our region, special education modality for people with intellectual disabilities (ID). A school activity management system is fundamental to follow the learning of the students, for students with ID, it is important to adapt the system with modifications to attend them in an inclusive way. For this work the main LPS methodologies in the literature were studied, as well as a hybrid methodology that combines characteristics from three others. No instantiations for students with ID using LPS were found in the literature, so the hybrid methodology was used to develop an activity management system for students with ID. The LPS created was instantiated for the school system that works with students with intellectual disabilities. The system generated for the school was implemented to run on the mobile platform. As a result, we obtained a robust model to complement future implementations of the same domain.

Keywords: Product Line; School Management; Intellectual Disabilities.

LISTA DE ILUSTRAÇÕES

Figura 1 - LPS e seus produtos	18
Figura 2 - Comparativo entre abordagem comum e LPS	20
Figura 3 - Etapas do método FAST	21
Figura 4 - Representação do método PLUS	22
Figura 5 - Fases do método de Delazeri e Wolf	25
Figura 6 - Artefatos produzidos no método adaptado por Delazeri e Wolf	26
Figura 7 - Definição de nome e descrição de atividades no Moodle	30
Figura 8 - Definição de disponibilidade de atividades no Moodle	30
Figura 9 - Tipos de envio e tipos de feedback de atividades no Moodle	31
Figura 10 - Envio de atividades no Moodle	31
Figura 11 - Atividade na linha do tempo do aluno no Moodle	31
Figura 12 - Status do envio de atividades e avaliação do professor no Moodle	32
Figura 13 - Criação de atividades no <i>Classroom</i>	33
Figura 14 - Criação de atividade simples no <i>Classroom</i>	33
Figura 15 - Criação de atividade com quiz no <i>Classroom</i>	34
Figura 16 - Menu superior de navegação e página de atividades do Google <i>Classroom</i>	35
Figura 17 - Página de uma atividade no Google <i>Classroom</i>	36
Figura 18 - Rapininha Flora	37
Figura 19 - Tela de login e tela inicial	39
Figura 20 - Atividades do dia e boletim do aluno	39
Figura 21 - Aulas semanais e contatos da escola	40
Figura 22 - Ranking de turmas e boletim	40
Figura 23 - Tela inicial do professor e criação de atividades	41
Figura 24 - Diagrama de contextos para gerenciamento de atividades	43
Figura 25 - Diagrama de <i>features</i> do núcleo	47
Figura 26 - Diagrama de <i>features</i> opcionais	48
Figura 27 - Diagrama de Caso de Uso das <i>features</i> opcionais e de núcleo	49
Figura 28 - Interfaces do sistema	53
Figura 29 - Diagrama de conceitos de negócio	53
Figura 30 - Diagrama de conceitos de negócio com interfaces	54
Figura 31 - Diagrama de componentes do núcleo	55
Figura 32 - Estrutura principal do projeto	57
Figura 33 - Autenticação de login Flora	57
Figura 34 - Classe Atividade	58
Figura 35 - Função de criação de atividades Flora	58
Figura 36 - Tela de login Flora	59
Figura 37 - Tela inicial Flora	60
Figura 38 - Criação de atividades Flora	61
Figura 39 - Fluxograma de atividades e avaliações da aplicação	63
Figura 40 - Diagrama de caso de uso da aplicação	64
Figura 41 - Diagrama de classes da aplicação	64
Figura 42 - gerarPDF da classe Atividade	65

Quadro 1 - Comparação entre os métodos de LPS.....	23
Quadro 2 - Relação de contextos e características.....	43
Quadro 3 - Contextos e características do Moodle - Perspectiva do Aluno.....	44
Quadro 4 - Contextos e características do Classroom - Perspectiva do Aluno	44
Quadro 5 - Contextos e características do app Flora.....	45
Quadro 6 - Classificações das características dos sistemas.....	46
Quadro 7 - Comparativo de características presentes nas plataformas.....	47
Quadro 8 - Descrição para o caso de uso Realizar Atividades.....	50
Quadro 9 - Descrição para o caso de uso Realizar Login.....	50
Quadro 10 - Descrição para o caso de uso Avaliar Atividades.....	51
Quadro 11 - Descrição para o caso de uso Atribuir Atividades.....	51
Quadro 12 - Descrição para o caso de uso Checar Notas das Atividades.....	52

LISTA DE ABREVIATURAS E SIGLAS

DI	Deficiência Intelectual
DSSA	<i>Domain-Specific Software Architecture</i>
FAST	<i>Family-Oriented Abstraction, Specification and Translation</i>
FODA	<i>Feature Oriented Domain Analysis</i>
GNU	<i>GNU's not Unix</i>
GUI	<i>Graphical User Interface</i>
IDE	<i>Integrated Development Environment</i>
Libras	Linguagem Brasileira de Sinais
LPS	Linhas de Produto de Software
Moodle	<i>Modular Object-Oriented Dynamic Learning Environment</i>
ODM	<i>Organization Domain Modeling</i>
PDF	<i>Portable Document Format</i>
PG	Ponta Grossa
PLUS	<i>Product Line UML-Based Software Engineering</i>
PuLSE	<i>Product Line Software Engineering</i>
UML	<i>Unified Modeling Language</i>
UTFPR	Universidade Tecnológica Federal do Paraná

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Objetivos	14
1.2	Justificativa	15
1.3	Organização do Trabalho	15
2	LINHAS DE PRODUTO DE SOFTWARE (LPS)	17
2.1	Origem das Linhas de Produto de Software	17
2.2	Abordagem de uma LPS	18
2.3	Modelos para Desenvolvimento de uma LPS	19
2.3.1	Método FAST	19
2.3.2	Método FODA	20
2.3.3	Método PLUS	21
2.3.4	Comparação dos Métodos	22
2.3.5	Método Híbrido para Desenvolvimento da LPS	24
2.4	Domínios de aplicação da LPS	26
2.5	Considerações Finais do Capítulo	27
3	SISTEMA DE GERENCIAMENTO DE ATIVIDADES	28
3.1	Objetivo do gerenciamento de atividades	28
3.2	Moodle	29
3.2.1	Visão do professor para criação de atividades no Moodle	30
3.2.2	Visão do aluno da realização de atividades no Moodle	31
3.2.3	Ferramentas de acessibilidade no Moodle	32
3.3	Google Classroom	32
3.3.1	Visão do professor da criação de atividades no Google Classroom	33
3.3.2	Visão do aluno da realização de atividades no Google Classroom	34
3.3.3	Ferramentas de acessibilidade no Google Classroom	36
3.4	Gerenciamento de Atividades de uma Instituição de Educação Especial	36
3.4.1	Funcionamento das atividades	37
3.4.2	Prototipação do sistema	38
3.5	Considerações Finais do Capítulo	41
4	MODELAGEM DO SISTEMA USANDO LPS	42
4.1	Análise do Domínio	42
4.2	Requisitos do Domínio	43

4.2.1	Moodle	43
4.2.2	Google Classroom	44
4.2.3	Flora	45
4.2.4	Classificação das Características Encontradas	45
4.2.5	Modelagem do Domínio	48
4.2.6	Projeto do Domínio	52
4.2.7	Implementação do Domínio	56
4.2.8	Teste do Domínio	62
4.3	Requisitos da Aplicação	63
4.4	Implementação da Aplicação	66
4.5	Testes da Aplicação	66
4.6	Considerações Finais do Capítulo	66
5	CONCLUSÃO	68
5.1	Trabalhos Futuros	69
	REFERÊNCIAS	70

1 INTRODUÇÃO

Em 2020, iniciou-se a maior crise que a população enfrentou nos últimos anos, a pandemia do vírus Covid-19. Vários setores foram afetados pela abrupta situação, como o setor de tecnologia, ensino, pesquisa e trabalho de forma geral (SINGH, 2020). Diversas atividades passaram a ser realizadas de maneira remota para minimizar a propagação do vírus.

As aulas passaram a ser lecionadas na modalidade remota, com raras exceções de ensino presencial. Muitos professores e alunos não tinham o preparo adequado, muitas vezes sem conhecer as ferramentas e plataformas que precisariam utilizar ou até mesmo não possuindo os recursos necessários para ministrar ou assistir aulas. Professores tiveram que assumir os custos de serviços de internet, computadores, câmeras, microfones e reinventar sua forma de lecionar sem nenhum apoio das instituições governamentais (SOUZA, 2020).

Grande parte das instituições de ensino básico e fundamental não possuíam estrutura sistêmica para lidar com o início de trabalho remoto, todo seu funcionamento era muitas vezes apenas no papel (ALVES; XAVIER, 2018). Os professores, na maioria das vezes, ministravam atividades utilizando métodos tradicionais. Então, surgiu um dilema de como manter funcionando uma estrutura essencialmente física em um cenário digitalizado? (VIEIRA, 2021).

A solução foi migrar para sistemas adequados para o dia a dia, no entanto, cada escola tem sua particularidade, suas rotinas, métodos de avaliação e estrutura. Quanto mais um produto é customizado para um cliente, mais investimentos são necessários. Para contornar essa dificuldade, uma alternativa é desenvolver sistemas que possam ser reutilizados para atender o público geral e uma parcela mais prejudicada durante a pandemia, como as instituições que atendem pessoas com deficiência intelectual (DI) (CURY, 2020).

O desenvolvimento de sistemas usando a abordagem de Linhas de Produto de Software (LPS) permite modelar e estender sistemas que possam atender diversos públicos. Como exemplo, tem-se a LPS proposta por Lessa Filho e Domínguez (2018) para o desenvolvimento de jogos educativos.

Seguindo na linha de educação, um sistema escolar pode apresentar muitas semelhanças e essa característica se torna ideal para a aplicação de uma LPS. O

objetivo da LPS é estruturar uma linha de produção para as tarefas repetitivas que serão realizadas pelas mesmas pessoas, como uma fábrica, a meta é agilizar a produção reduzindo a quantidade de erros, gastos, tempo e de pessoal (DURSCKI, 2004). Apesar da LPS ser um método antigo, existem diversas aplicações atuais na literatura como de Coutinho e Bezerra (2019), Saldanha (2019), Bindewald (2020) e Barros (2021).

Na literatura os métodos usados para o desenvolvimento de uma LPS são o FAST (*Family-Oriented Abstraction, Specification and Translation*) (HARSU, 2002), FODA (*Feature Oriented Domain Analysis*) (LOBO; RUBIRA, 2009) e PLUS (*Product Line UML-Based Software Engineering*) (DALMON, 2012). Delazeri e Wolf (2012), estudaram esses métodos e propuseram um método híbrido contendo características relevantes de cada um deles.

A fim de desenvolver um sistema de gerenciamento de atividades que possa atender o público prejudicado durante a pandemia, este trabalho aplicou o método híbrido de Delazeri e Wolf (2012) para analisar as diferenças e semelhanças entre três sistemas de gerenciamento de atividades: *Moodle* (MOODLE, 2001), *Google Classroom* (GOOGLE, 2014) e o sistema da escola onde estudam alunos com deficiência intelectual que ainda não se encontra informatizado.

Um sistema de gerenciamento de atividades deve possibilitar o cadastro de atividades, o acompanhamento das atividades realizadas, a possibilidade de realizar atividades, avaliação de atividades efetuadas e checagem de *feedback* dado pelos professores. Para alunos com algum tipo de DI é necessário adequar as atividades, ou seja, possibilitar que o aluno entenda a proposta, consiga realizar e aprender. Para isso, existem alguns elementos que facilitam o entendimento, como uso de cores, fontes grandes, figuras, sons e vídeos (PLETSCH, 2012).

Como resultado deste trabalho se obteve um modelo que pode ser implementado na plataforma *mobile*, em que se utilizou o núcleo da LPS e as características do sistema para pessoas com deficiência intelectual. O modelo pode também ser usado para estender outros produtos do mesmo domínio.

1.1 Objetivos

O objetivo geral deste projeto é propor um modelo de sistema de gerenciamento de atividades aplicando os conceitos de Linhas de Produto de Software. Os objetivos específicos são:

- Aplicar uma metodologia de LPS no sistema de gerenciamento de atividades;
- Estender o modelo para criação de uma aplicação *mobile* que pode ser usada por educadores e alunos com algum tipo de deficiência intelectual (DI).

1.2 Justificativa

O uso de LPS faz com que o produto final seja mais robusto já que existe a reutilização de um núcleo extensivamente testado e validado em aplicações anteriores e diminui o custo de produção devido a reutilização de código. Facilita também o treinamento dos funcionários, adaptação de novos colaboradores e permite uma maior abrangência no mercado devido sua flexibilidade.

Além dos fatores ditos anteriormente, para o cliente, a utilização de LPS faz com que o sistema seja mais confiável, já que é fácil conferir resultados positivos de outros usuários de sistemas que utilizam essa metodologia.

O acompanhamento das atividades realizadas por um aluno é importante, principalmente para alunos com algum tipo de deficiência intelectual (DI) que durante a pandemia ficaram excluídos, pois muitos não tinham acesso ou conhecimento para executar suas atividades usando a tecnologia.

O acompanhamento das atividades, para a instituição, permite descobrir se a metodologia adotada pelo professor é adequada e para os pais permite ter a segurança de saber que seu filho está amparado pela escola (OLIVEIRA, 2011).

1.3 Organização do Trabalho

O trabalho está organizado em cinco capítulos. O Capítulo 1 apresenta a motivação e problema em questão que se pretende resolver. O Capítulo 2 aborda sobre LPS, seu surgimento, métodos mais comentados na literatura, o método híbrido escolhido para desenvolver este trabalho e algumas aplicações da LPS.

O Capítulo 3 trata do gerenciamento de atividades escolares, traz exemplos de sistemas conhecidos e o funcionamento de um sistema não informatizado voltado a educação na modalidade de educação especial.

O Capítulo 4 aborda a modelagem do sistema utilizando a LPS, suas subfases, os produtos resultantes das etapas e os resultados obtidos após a construção da LPS. Por fim, o Capítulo 5 apresenta as conclusões do trabalho e traz os trabalhos futuros que podem ser realizados a partir deste estudo.

2 LINHAS DE PRODUTO DE SOFTWARE (LPS)

Este capítulo apresenta conceitos sobre Linha de Produto de Software (LPS), sua origem, definições, como funcionam as abordagens de LPS, suas vantagens e desvantagens, alguns dos métodos mais referenciados na literatura e o método escolhido para o desenvolvimento do trabalho. A Seção 2.1 descreve sobre as origens da LPS. A Seção 2.2 relata sobre as abordagens para desenvolvimento de uma LPS. A Seção 2.3 apresenta os modelos de LPS tais como FAST (*Family Oriented Abstraction, Specification, and Translation*), FODA (*Feature Oriented Domain Analysis*), PLUS (*Product Line UML-Based Software Engineering*) e um comparativo entre eles. O método escolhido para o desenvolvimento do trabalho também é descrito nesta seção. A Seção 2.4 apresenta algumas aplicações da LPS. Por fim, a Seção 2.5 traz as considerações finais do capítulo.

2.1 Origem das Linhas de Produto de Software

Pode-se dizer que a origem das Linhas de Produto de Software (LPS) foi inspirada na linha de produção de Henry Ford. Com a popularização dos veículos, foi necessário atender uma grande demanda, a produção de carros aumentou, mas por conta do processo de produção da época não era possível suprir a necessidade de agilidade na produção. A solução foi padronizar o produto, diminuindo os custos, aumentando a escala de produção e barateando o preço final para o cliente (SILVA, 2011).

O problema é que os veículos passaram a ser padronizados. Apesar dos clientes conseguirem o produto em menor tempo, ele não era customizado, ou seja, todos iguais. Para contornar esse problema, foi definido uma base padrão para todos os veículos, mas com possibilidade de modificar a cor, quantidade de portas, tamanho, potência, entre outras características. Com mais tecnologia, pode-se customizar o veículo em maior extensão, diversificando muito o produto final. No desenvolvimento de software também acontece a mesma situação, um produto padronizado não consegue atender um grande número de clientes de forma adequada e satisfatória, é necessário atender as particularidades de cada domínio para que o software seja útil e eficaz (SILVA, 2011).

Devido a esta demanda surgiram as Linhas de Produto de Software (LPS), em que vários produtos distintos são criados seguindo uma mesma base, apenas diversificando onde é necessário, já que cada produto pode ter atores distintos se relacionando de maneiras diferentes e produzindo sequências de eventos variáveis (KISHI; NODA, 2009).

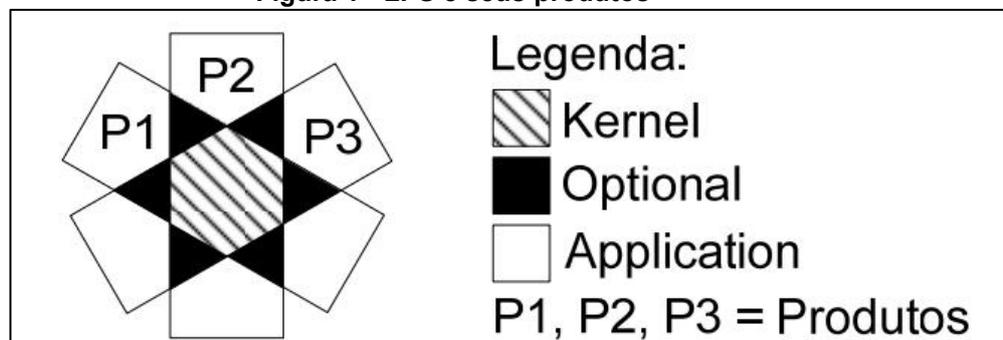
Pode-se dizer que um ponto-chave da LPS é a reusabilidade de elementos para construir sistemas com menor esforço, tempo e recursos. A LPS se torna rentável quando existem pontos em comum exploráveis, ou seja, pontos que podem ser utilizados em diversos produtos (CLEMENTS, 2002).

2.2 Abordagem de uma LPS

A abordagem de uma LPS tem como princípio encontrar diferenças e semelhanças entre produtos e artefatos de software similares. A partir desse levantamento é possível gerar um núcleo base para produzir outros softwares. Durante a criação do novo produto, as diferenças devem ser definidas durante os requisitos (OLIVEIRA, 2009).

A Figura 1 apresenta a organização dos produtos em uma LPS. Para a construção de um novo produto tem-se: as funcionalidades comuns (*Kernel*) e opcionais (*Optional*). Essas fazem parte do *core assert*, o núcleo do produto. Posteriormente é feito o desenvolvimento das funcionalidades específicas do produto (*Application*) (OLIVEIRA, 2009). Na figura 1, o núcleo (representado pelo *kernel*) em conjunto com os elementos opcionais (representado pelos triângulos) e os elementos de aplicação (em branco), resultam nos produtos (P1, P2 e P3).

Figura 1 - LPS e seus produtos



Fonte: Santana et al. (2009).

A LPS possibilita a reutilização de diversos elementos, como requisitos, *design* de arquitetura, componentes, modelos e análises, testes, planejamentos,

processos e pessoas (CLEMENTS, 2002). Como outras vantagens da LPS tem-se (FERGUSON, 2018):

- Menor tempo de produção: devido a reusabilidade o produto ficará pronto em menor tempo.
- Aumento da qualidade do produto: por ser composto de elementos reutilizados, a qualidade dos módulos já foi verificada em produtos anteriores.
- Maior agilidade de mercado: o produto, por ficar pronto mais rápido, pode ser comercializado o quanto antes, atendendo as demandas do mercado com maior velocidade.
- Maior satisfação de clientes: o produto chega mais rápido, mais robusto e customizável.
- Maior presença de mercado: por ter menor tempo de produção, seu produto chega antes no mercado e consegue dominá-lo com maior facilidade.

Como algumas desvantagens da LPS tem-se (FERGUSON, 2018): i) Custo inicial elevado para desenvolver: no início é maior, pois não existe nada para reaproveitar; ii) Necessidade de atualização do núcleo conforme o tempo passa para mantê-lo relevante. Caso isso não seja feito da maneira correta, pode comprometer o desenvolvimento de produtos futuros.

2.3 Modelos para Desenvolvimento de uma LPS

Essa seção aborda os modelos de LPS mais citados na literatura e traz um comparativo dos métodos. Apresenta também o modelo escolhido para o desenvolvimento deste trabalho.

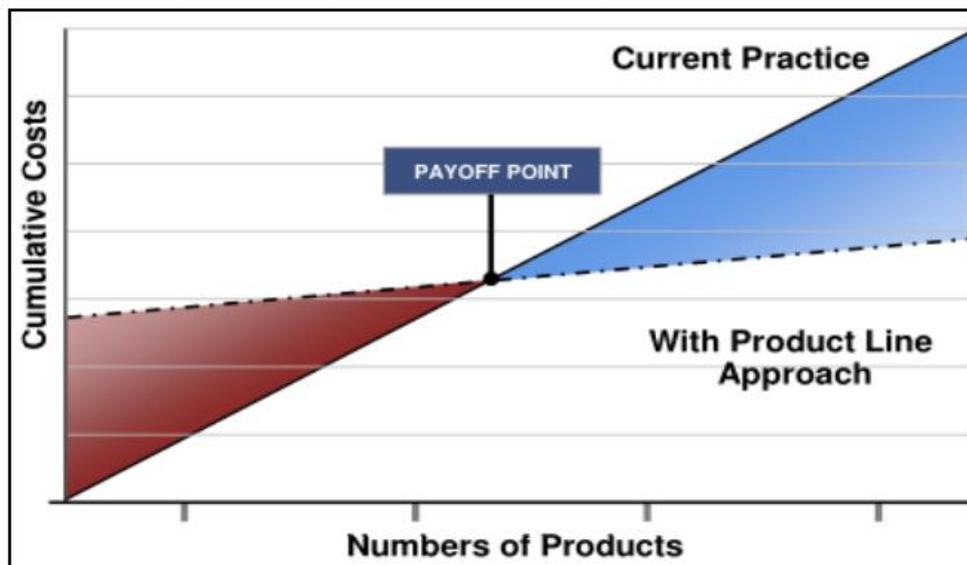
2.3.1 Método FAST

O método FAST é um processo de desenvolvimento dividido em duas partes: criação do núcleo, incluindo o ambiente de desenvolvimento para implementação de cada produto; utilização do núcleo e ambiente na produção de diferentes softwares pertencentes à mesma família (HARSU, 2002). O método utiliza “famílias”, ou seja, conjuntos de software para sua análise.

O FAST pode ser dividido em três subprocessos (HARSU, 2002):

- Qualificação do domínio: busca determinar a viabilidade econômica da implantação no domínio selecionado. Essa viabilidade pode ser verificada por meio de um gráfico de *payoff point*, como pode ser visto na Figura 2.
- Engenharia de domínio: identifica os pontos comuns e diferentes da família. Também desenvolve a base para a LPS.
- Engenharia de aplicação: produz as aplicações da família.

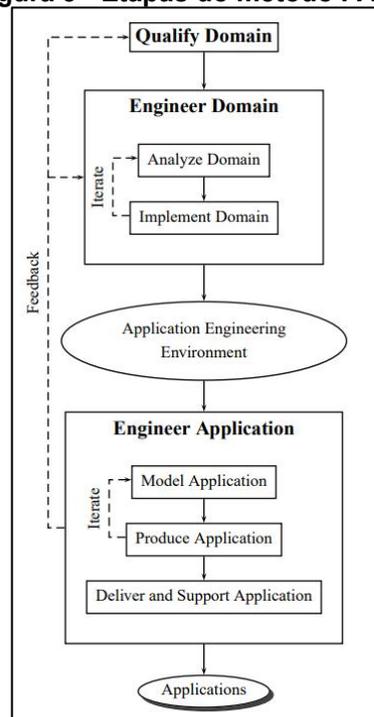
Figura 2 - Comparativo entre abordagem comum e LPS



Fonte: Weiss e Lai (1999).

Na aplicação do método FAST (figura 3), devem existir dois grupos de engenheiros na organização do trabalho: um para a engenharia de domínio e outro para a engenharia de aplicação. Engenheiros de domínio controlam a evolução da família e verificam se ainda é rentável produzir aplicações no domínio atual. Os engenheiros de aplicação produzem os membros da família e mantêm contato constante com o cliente, para assegurar que os requisitos atendem a demanda (HARSU, 2002).

Figura 3 - Etapas do método FAST



Fonte: Harsu (2002).

2.3.2 Método FODA

O método FODA foi um dos métodos mais populares durante a década de 1990 por conta da sua descrição detalhada da análise de domínio (SILVA, 2011). As *features* presentes no método podem ser de um dos seguintes tipos (KANG, 1990):

- Obrigatória: tem de estar presente em todos os membros da LPS;
- Opcional: pode ou não estar presente nos membros da LPS;
- Alternativa: trata da escolha de uma ou mais alternativas de um mesmo grupo, deve ser indicado se é necessário escolher apenas uma ou mais.

O método possui duas atividades essenciais: análise de contexto e a modelagem de domínio. Na análise de contexto, ocorre a definição do escopo do domínio e é realizada a análise das relações entre o domínio e os elementos externos. Os resultados da análise de contexto são documentados em um modelo de contexto, definindo o seu limite e seu escopo.

Na modelagem do domínio é feita a análise de similaridades entre as aplicações do domínio, gerando diversos modelos para diferentes aspectos. Essa atividade é dividida em (KANG, 1990):

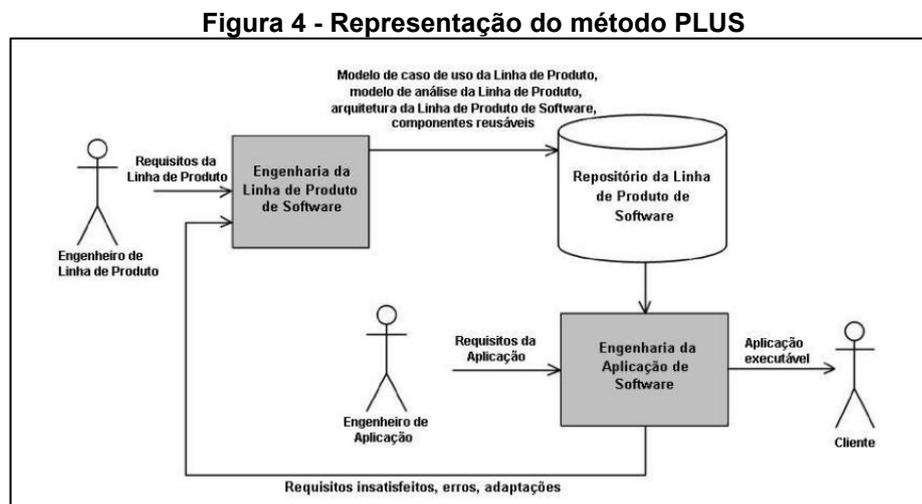
- Análise de características: representação das características similares e variáveis do domínio e seus relacionamentos. Nesta acontece a

definição de quais *features* que são obrigatórias, opcionais ou alternativas.

- Modelagem entidade-relacionamento: definição do conhecimento necessário para a implementação das aplicações no domínio.
- Análise funcional: identifica as diferenças e semelhanças das aplicações do domínio de forma funcional. As funcionalidades comuns são estruturadas em um modelo funcional.

2.3.3 Método PLUS

O método PLUS é baseado na UML (*Unified Modeling Language*). Ele é um modelo executado de forma incremental e iterativa, possuindo como atividades principais a engenharia de linha de produto de software e a engenharia de aplicação (DONEGAN, 2008). A Figura 4 apresenta a execução do método PLUS: a Engenharia da Linha de Produto de Software consiste na construção dos modelos de caso de uso, análise e de arquitetura baseados nos requisitos levantados pelos engenheiros. Além disso, alguns componentes reusáveis são armazenados no repositório da LPS.



Fonte: Donegan (2008).

A Engenharia da Aplicação de Software permite identificar os requisitos de cada aplicação para o desenvolvimento. Os modelos construídos anteriormente são utilizados para derivar os modelos de cada aplicação. A partir da definição desses modelos e componentes do repositório, são obtidas as aplicações executáveis. Todo o processo deve ser executado novamente caso sejam encontrados requisitos

insatisfeitos, erros ou se for necessário adaptar alguma parte da aplicação para atender uma nova demanda.

2.3.4 Comparação dos Métodos

Almeida (2013) realizou um estudo comparativo entre os principais métodos de desenvolvimento de uma LPS, apresentado no Quadro 1.

Quadro 1 - Comparação entre os métodos de LPS

Características	FODA	FAST	PLUS
Abordagem	<i>Features</i>	Família	<i>Features</i>
Foco	Identificação dos processos comuns nas aplicações, na alocação de recursos e funções.	Tornar o software mais eficiente, redução das múltiplas tarefas, custos e tempo de comercialização.	Explicitação das características comuns e variáveis do software.
Diagramas	Caso de Uso, Classe e Modelo de Características.	Caso de Uso, Classe, Estado e Modelo de Características.	Caso de Uso, Sequência e Estado.
Entradas	Definição do domínio a ser modelado, dois ou mais exemplos de aplicações no domínio.	Análise e definição dos produtos, definição do domínio a ser modelado e sua arquitetura, requisitos da aplicação e validação da aplicação.	Definição dos requisitos comuns e variáveis entre os produtos, decomposição do problema, sintetização dos artefatos, arquitetura da parte similar e codificação.
Saídas	Definição do escopo do domínio, identificação de pontos comuns entre os estudos de caso, diagrama de características, modelagem da arquitetura e diagrama de caso de uso.	Definição da aplicação do processo de LPS, definição do escopo do domínio e codificação, lista de requisitos das variabilidades, diagrama de caso de uso, diagrama de classe e aplicação.	Desenvolvimento da modelagem de caso de uso, desenvolvimento da modelagem estática e dinâmica, criação do modelo de projeto de software, codificação, validação dos componentes, armazenamento de modelos, e adaptação de modelos prévios.
Fases	Análise de contexto, modelagem do domínio (análise de características: identificação, modelagem, validação, modelagem entidade - relacionamento; análise funcional: especificação de funções e comportamento).	Qualificação de domínio, engenharia de domínio (análise e implementação), engenharia de aplicação (modelo, produção, entrega e suporte).	Engenharia de Linhas de Produto (modelagem de requisitos, análise, projeto, implementação incremental de componentes, teste) e engenharia de aplicação.

Fonte: Adaptado de Almeida (2013).

Analisando o estudo feito por Almeida (2013) é possível perceber que os métodos FODA e PLUS possuem características semelhantes (como o uso da abordagem de *features* e o foco em encontrar características semelhantes nos produtos analisados), sendo diferentes do método FAST, que possui uma abordagem em famílias de software. Isso mostra a diversificação de abordagens existente nos métodos, mesmo que eles tenham a mesma finalidade, na aplicação de uma LPS.

Delazeri e Wolf (2012) também notaram a presença de características comuns e diferentes nos métodos, mas para capturar as qualidades dos três, propuseram uma adaptação própria para o desenvolvimento da LPS. A adaptação foi usada neste trabalho e é descrita na próxima seção.

2.3.5 Método Híbrido para Desenvolvimento da LPS

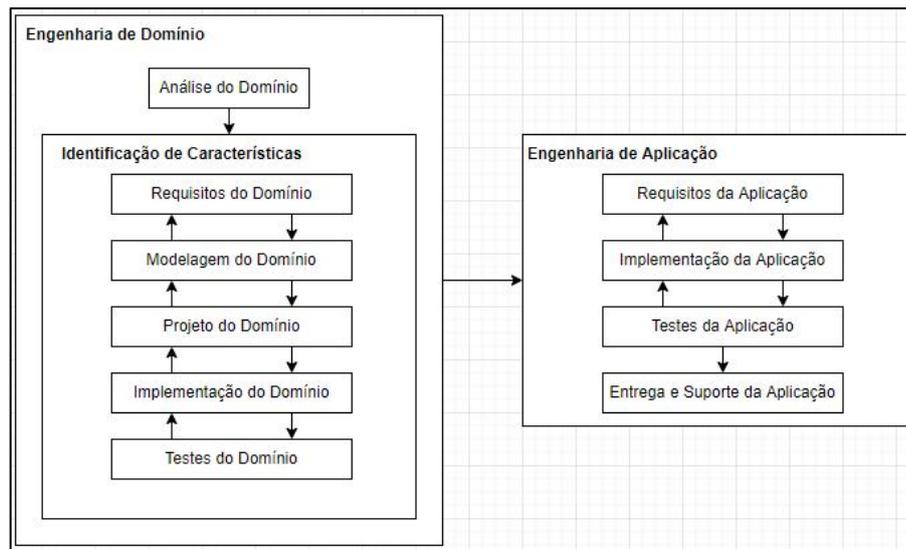
O método elaborado por Delazeri e Wolf (2012) disponibiliza fases, documentações e diagramas específicos para atender as necessidades da linha de produto de software.

Ele possui duas fases principais: uma de engenharia de domínio e uma de engenharia de aplicação (Figura 5). Na fase de engenharia de domínio se tem a construção de artefatos reutilizáveis, além disso, duas subfases estão presentes:

- Análise de domínio: derivada do método FAST, analisa e define o escopo do software. Tem como produto final um modelo de contexto.
- Identificação de características: identificação e modelagem de características comuns e variáveis. Com cinco atividades:
 - Requisitos do domínio: baseada na análise de características dos métodos FODA e PLUS. Levantamento de pontos comuns e variáveis da linha.
 - Modelagem do domínio: baseada nas atividades iniciais da engenharia de LPS do método PLUS. Realiza a modelagem estática por meio do diagrama de classe ou de caso de uso.
 - Projeto de domínio: baseado nas atividades intermediárias da engenharia de LPS do método PLUS. Definição da escolha de arquitetura dos componentes.

- Implementação do domínio: baseada na atividade de implementação incremental de componentes do método PLUS. Programação de componentes reutilizáveis com definição da linguagem de programação do software.
- Teste do domínio: validação dos componentes, visando a integridade e funcionalidade. Originária do método PLUS.

Figura 5 - Fases do método de Delazeri e Wolf



Fonte: Adaptado de Delazeri e Wolf (2012).

Na engenharia de aplicação existem quatro subfases:

- Requisitos da aplicação: originárias dos métodos PLUS e FAST. Modela e identifica os requisitos em diagramas de caso de uso. Responsável pela criação do diagrama de características de variabilidades do domínio em questão.
- Implementação da aplicação: adaptada do método FAST, programação do produto conforme os requisitos.
- Testes de aplicação: validação do produto por meio de testes funcionais.
- Entrega e suporte da aplicação: do método FAST, suporte realizado após entregar o produto ao cliente, tem como objetivo garantir que eventuais problemas sejam corrigidos e que o cliente tenha um devido suporte.

A Figura 6 apresenta os artefatos gerados em cada uma das fases do método adaptado.

Figura 6 - Artefatos produzidos no método adaptado por Delazeri e Wolf

Fases	Subfases	Artefatos de Entrada	Artefatos de Saída	
Engenharia de Domínio	Análise de domínio	Definição do Domínio a ser modelado	Modelo de contexto definindo o escopo do domínio	
	Identificação de Características	Requisitos do domínio	Dois exemplos de aplicações no domínio, no mínimo.	- Descrição narrativa de cada exemplo. Caso não se tenha a descrição narrativa, devem-se utilizar os aplicativos disponíveis. Neste caso, a análise será realizar por meio da execução do software. Identificação dos pontos comuns entre os estudos de caso. - Requisitos identificados - Diagrama de características (contendo os pontos de comuns)
		Modelagem do domínio	Descrição narrativa de cada exemplo. Pontos de comuns	- Diagrama de Caso de Uso - Cenários dos casos de uso - Diagrama de classe
		Projeto do domínio	Diagrama de Caso de Uso	- Arquitetura da parte Similar (baseada em componentes) - Diagramas de classe para a concepção da arquitetura - Especificação das Interfaces do Sistema. - Identificação das Interfaces de Negócio. Identificação dos Componentes.
		Implementação do domínio	Arquitetura da parte Similar	- Codificação dos componentes da arquitetura similar
		Testes do domínio	Codificação	Validação dos componentes
Engenharia de Aplicação	Requisitos da aplicação	Requisitos da aplicação (pontos de variabilidade) oriundos da fase Requisitos do domínio	- Lista de requisitos das variabilidades - Diagrama de caso de uso da aplicação - Diagrama de classe	
	Implementação da aplicação.	Diagrama de classe	- Codificação dos pontos variáveis	
	Testes da Aplicação	Codificação da aplicação	- Plano de testes funcionais e de integração - Produto validado	
	Entrega e Suporte da Aplicação	Aplicação validada	- Aplicação	

Fonte: Delazeri e Wolf (2012).

2.4 Domínios de aplicação da LPS

Durante o estudo sobre a LPS, foram encontrados alguns trabalhos sobre sua aplicação em diversos domínios, os trabalhos foram escolhidos com o objetivo de mostrar a abrangência e a variabilidade da aplicação da LPS. Canteri e Dvulatka (2013) utilizaram a LPS para modelar e implementar um sistema de venda e locação de imóveis. Em seu estudo eles trouxeram outros métodos de LPS, diferentes dos mais populares no meio: Draco, ODM (*Organization Domain Modeling*), FeatuRSEB, DSSA (*Domain-Specific Software Architecture*), PuLSE (*Product Line Software Engineering*) e KobrA.

Zaina, em 2015, propôs uma LPS para a construção de museus virtuais para aprendizagem sobre obras de arte. Seu objetivo foi criar um ambiente virtual que utilizasse obras de arte para ensinar.

Saldanha (2019) utilizou a LPS para desenvolver um sistema de gerenciamento bibliotecário, ele também utilizou o método proposto por Delazeri e Wolf (2012). Martins (2019), desenvolveu um *plugin* para avaliar a qualidade de software, mais especificamente coletar medidas de uma LPS desenvolvida. Ela utilizou a LPS para avaliar a própria LPS.

Coutinho e Bezerra (2019), utilizaram alguns conceitos da LPS para avaliar a variabilidade da modelagem de software de um ambiente virtual de aprendizagem. Eles focaram no modelo de *features* para realizar essa análise.

Bindewald (2020) realizou um estudo sobre a otimização de projetos, com foco na arquitetura, utilizando uma LPS. Seu objetivo foi avaliar formas de melhor atender um arquiteto de software, adaptando a LPS para a demanda.

Barros (2021) propôs o desenvolvimento de uma LPS para construir aplicações para promover o turismo. A necessidade surgiu por cada cidade possuir um tipo diferente de atração, mas todas são atrações turísticas. Cenário ideal para uma LPS por conta das semelhanças que cada sistema teria.

Não foram encontrados trabalhos que utilizaram a LPS para o gerenciamento de atividades escolares para alunos com DI. Devido a isto, este trabalho foi desenvolvido.

2.5 Considerações finais do capítulo

Neste capítulo foi descrito a origem da LPS e a motivação de sua criação. Inspirada na linha de produção das fábricas de veículos, teve sucesso no ambiente digital e na produção de software. A abordagem da LPS foi apresentada, junto com seus pontos positivos e negativos.

Foram apresentadas as abordagens para o desenvolvimento de uma LPS. Na literatura as mais comuns são os métodos FAST, FODA e PLUS. Além disso foi apresentado um quarto modelo, proposto por Delazeri e Wolf em 2012 a qual foi usada neste trabalho. Por fim, foram apresentadas aplicações da LPS encontradas em vários domínios e não se encontrou LPS para gerenciamento de atividades.

3 SISTEMA DE GERENCIAMENTO DE ATIVIDADES

Este capítulo apresenta a concepção de um sistema de gerenciamento de atividades e três aplicações-exemplo (sistemas) que possuem esta finalidade. A Seção 3.1 traz o objetivo do gerenciamento de atividades. A Seção 3.2 apresenta o gerenciamento de atividades da plataforma *Moodle*. A Seção 3.3 relata a ferramenta *Google Classroom* e como as atividades são geridas nesta plataforma. A Seção 3.4 relata como o gerenciamento de atividades é realizado em uma instituição de ensino para alunos com Deficiência Intelectual da região. Por fim, a Seção 3.5 descreve as considerações finais do capítulo.

3.1 Objetivo do gerenciamento de atividades

O gerenciamento de atividades tem como objetivo gerir as atividades que um professor delega para seus alunos. Essas atividades podem ser atribuídas, realizadas e avaliadas. Toda essa dinâmica era, e ainda é feita fisicamente, prejudicando o armazenamento e manuseio de atividades, provas e trabalhos. Com o avanço da tecnologia, o gerenciamento escolar vem sendo realizado digitalmente.

Um sistema de gerenciamento de atividades implementado no modelo virtual, aumenta a quantidade de atividades que podem ser propostas ao aluno. O professor não fica preso ao papel e lápis, pode utilizar animações, sons, vídeos interativos e outras formas de atividades e avaliações (VALENTE, 2018).

Para esse trabalho foram escolhidas duas plataformas digitais e uma das aplicações-exemplo para análise foi descrita fundamentada no gerenciamento de atividades para alunos com deficiência intelectual (DI)

A escolha das duas plataformas foi baseada na quantidade de números de usuários que elas possuem. Segundo dados próprios, o *Moodle* possui cerca de 316 milhões de usuários, em 178 mil sites, espalhados em 241 países (MOODLE, 2022). O *Classroom* possui cerca de 150 milhões de estudantes (GOOGLE, 2021). A escolha da proposta para gerenciamento de atividades escolares para alunos com deficiência intelectual surgiu da inspiração trazida pela orientadora deste trabalho e também pelas informações relatadas em artigos sobre a pandemia do Covid-19 e o público com DI, em que afirmam que estes alunos ficaram com dificuldade de acesso as tecnologias.

De acordo com Courtenay e Perera (2020) pessoas com DI são vulneráveis aos efeitos físicos, mentais e sociais durante a pandemia do Covid-19, pois sua deficiência cognitiva pode dificultar e limitar a compreensão das informações para proteção contra este vírus.

Segundo Cury (2020) a convivência social é inerente ao ser humano e fundamental para o desenvolvimento de competências socioemocionais. O distanciamento por conta da pandemia do Covid-19 que vivenciamos torna-se mais uma barreira para a inclusão das pessoas com deficiência, transtorno do espectro autista e altas habilidades.

3.2 Moodle

O Moodle (*Modular Object-Oriented Dynamic Learning Environment*) é uma plataforma de aprendizado virtual à distância, baseada na filosofia de software livre GNU (*GNU's not Unix*). É acessível pela web ou por meio de aplicativos e o desenvolvimento de novas funcionalidades, manutenções ou traduções para outros idiomas é feito por uma equipe de desenvolvedores voluntários (SABBATINI, 2007).

Por ser um sistema robusto, de fácil implementação e com diversos módulos disponíveis para aplicação, é utilizado em instituições de ensino ao redor do mundo. O *Moodle* é utilizado pela Universidade Tecnológica Federal do Paraná (UTFPR) campus Ponta Grossa como plataforma de aprendizado. Nele é possível realizar o compartilhamento de conteúdo para uma turma de alunos ou professores, realizar atividades, avaliar atividades, se comunicar com alunos e professores, realizar provas, acompanhar evolução de notas e conteúdo, entre outras opções.

O *Moodle* é capaz de atender ao público com DI como relata o trabalho de Oliveira e Nunes (2019) que realizaram o desenvolvimento de uma ferramenta para auxiliar na avaliação da aprendizagem de informática para alunos com deficiência intelectual. Em 2020, Rossi, Amorim e Santos realizaram um estudo sobre a utilização do *Moodle* no ensino da matemática para alunos com deficiência. Nele, foi criado um ambiente virtual na plataforma com diversas atividades matemáticas. Como resultado, eles observaram que a fixação de conteúdo foi maior.

3.2.1 Visão do professor para criação de atividades no Moodle

Na criação de atividade é possível atribuir seu nome, descrição, se essa descrição será exibida na página do curso (Figura 7), especificar a partir de quando será possível enviar a atividade, sua data de entrega, data limite e definir um lembrete para que o professor não se esqueça de avaliar o que os alunos fizeram (Figura 8).

Figura 7 - Definição de nome e descrição de atividades no Moodle

Fonte: Autoria própria (2022).

Figura 8 - Definição de disponibilidade de atividades no Moodle

Fonte: Autoria própria (2022).

É possível definir qual tipo de envio a atividade aceita, seja ele texto *online* ou algum arquivo. A quantidade máxima de arquivos a serem enviados, seu tamanho máximo, tipos de arquivos aceitos (Figura 9). O professor também pode definir o tipo de *feedback* para aquela atividade.

Figura 9 - Tipos de envio e tipos de feedback de atividades no Moodle

The screenshot shows the Moodle activity settings interface. Under the 'Tipos de envio' section, there are options for 'Texto online' (unchecked) and 'Envios de arquivo' (checked). The 'Número máximo de arquivos enviados' is set to 20. The 'Tamanho máximo do envio' is set to 'Limite de upload para Site (128Mb)'. The 'Tipos de arquivo aceitos' field is empty, with an 'Escolher' button and 'Sem seleção' text. Under the 'Tipos de feedback' section, there are options for 'Comentários de feedback' (checked), 'Anotar PDF' (checked), 'Planilha de notas offline' (unchecked), and 'Arquivos de feedback' (unchecked). The 'Comentário na linha' is set to 'Não'.

Fonte: Autoria própria (2022).

3.2.2 Visão do aluno da realização de atividades no Moodle

As atividades criadas pelo professor podem ser acessadas pelos alunos matriculados naquele curso. No curso de Bacharelado de Ciência da Computação da UTFPR-PG, geralmente as atividades envolvem a entrega de arquivos de programação, artigos, apresentações ou atividades escritas. A Figura 10 apresenta a interface de entrega de atividades.

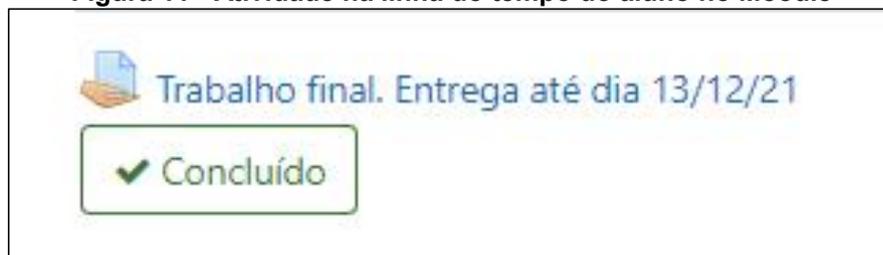
Figura 10 - Envio de atividades no Moodle

The screenshot shows the Moodle file upload interface. At the top, it says 'Envios de arquivo' and 'Tamanho máximo para arquivos: 128Mb, número máximo de anexos: 20'. Below this is a file manager interface with a folder icon and the text 'Arquivos'. A large blue arrow points downwards, and below it, the text reads 'Você pode arrastar e soltar arquivos aqui para adicioná-los.' At the bottom, there are two buttons: 'Salvar mudanças' and 'Cancelar'.

Fonte: Autoria própria (2022).

Uma atividade na linha do tempo do aluno terá um ícone específico, ao entregar, o *status* pode ser atualizado ou não (Figura 11).

Figura 11 - Atividade na linha do tempo do aluno no Moodle



Fonte: Autoria própria (2022).

Após ser entregue, a atividade trará algumas informações para o aluno, como data da última entrega, seus *status* de envio, *status* de avaliação, o que foi

enviado e um campo para comentários (Figura 12). A avaliação do professor aparece exemplificada na Figura 12, com a data de avaliação e a nota recebida.

Figura 12 - Status do envio de atividades e avaliação do professor no Moodle

Status de envio	
Status de envio	Enviado para avaliação
Status da avaliação	Avaliado
Tempo restante	A tarefa foi enviada 2 horas 4 minutos adiantado
Última modificação	segunda, 13 dez 2021, 21:54
Envios de arquivo	TF - Gustavo Salina.rar 13 dezembro 2021, 21:54
Comentários sobre o envio	Comentários (0)
Feedback	
Nota	100,00 / 100,00
Avaliado em	quarta, 15 dez 2021, 14:59
Avaliado por	SAULO JORGE BELTRAO DE QUEIROZ

Fonte: Autoria própria (2022).

3.2.3 Ferramentas de acessibilidade no Moodle

O Moodle possui diversas funcionalidades extras que podem ser incluídas por meio de *plugins* (extensão, traduzido de forma livre). Existem alguns *plugins* de acessibilidade disponíveis na plataforma.

Na Universidade Tecnológica do Paraná é utilizado uma extensão que realiza a transcrição de textos na língua portuguesa para a Língua Brasileira de Sinais (Libras). A extensão VLibras permite que o usuário clique em textos presentes no sistema para que um intérprete virtual realize os sinais correspondentes.

Outra extensão disponível é a *Accessibility* que possibilita o aumento de fontes ou a mudança de cores da página escolhida. Ela pode ser adicionada e utilizada por alunos e professores.

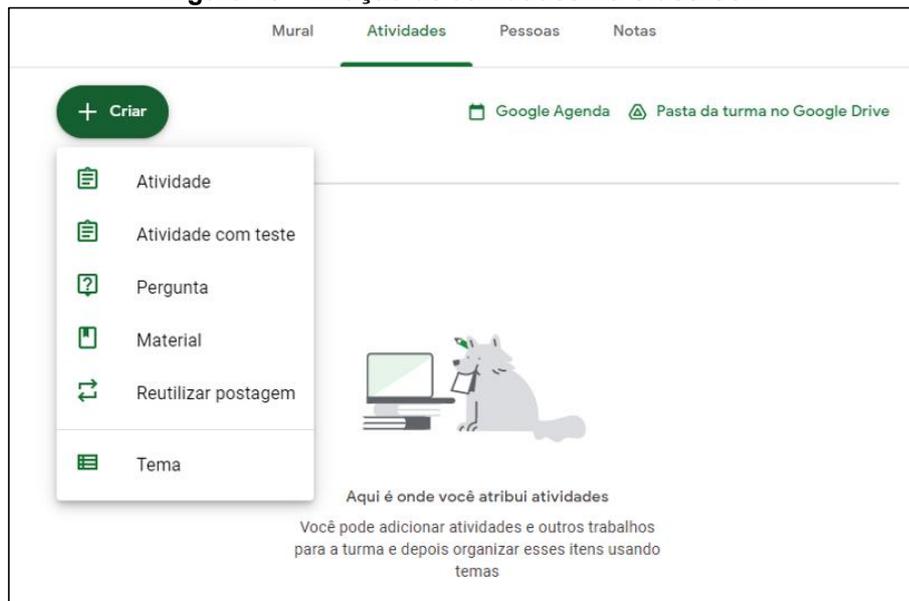
3.3 Google Classroom

O Classroom é uma plataforma de ensino à distância da Google e é acessível por quem possuir computador ou celulares. Facilita o armazenamento e gerenciamento de atividades e avaliações. Por demandar menos tempo para gerenciar uma turma, o professor pode utilizar melhor seu tempo para encurtar a distância da relação aluno - professor (IFTAKHAR, 2016).

3.3.1 Visão do professor da criação de atividades no Google Classroom

Um professor pode criar uma disciplina e compartilhar o acesso para os alunos por meio de um código, enviando um *link* ou convidando por e-mail para a turma e as atividades são disponibilizadas nessa disciplina criada. Elas são diversas, mas as principais são as atividades simples e as com formulário avaliativo (Figura 13).

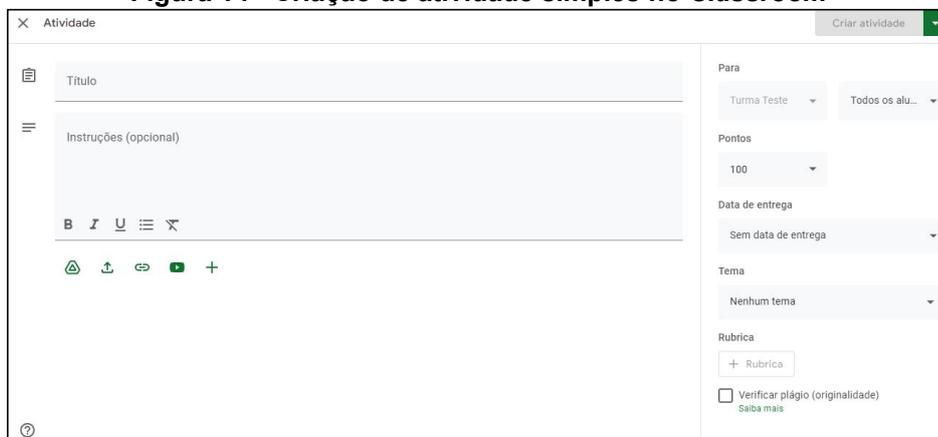
Figura 13 - Criação de atividades no Classroom



Fonte: Autoria própria (2022).

Ao criar uma atividade simples, o professor pode definir seu título, descrição ou instruções, atribuir para uma turma ou para alunos específicos, definir seu valor, data de entrega, tema e ativar a verificação de plágio (Figura 14).

Figura 14 - Criação de atividade simples no Classroom



Fonte: Autoria própria (2022).

A criação de uma atividade com formulário avaliativo possui as mesmas opções da atividade simples, porém, inclui um formulário gerado automaticamente

que pode ser editado pelo professor, além disso, inclui a opção de importar as notas do formulário para o *Classroom*, fazendo com que o aluno só possa enviar uma única resposta para aquele formulário (Figura 15).

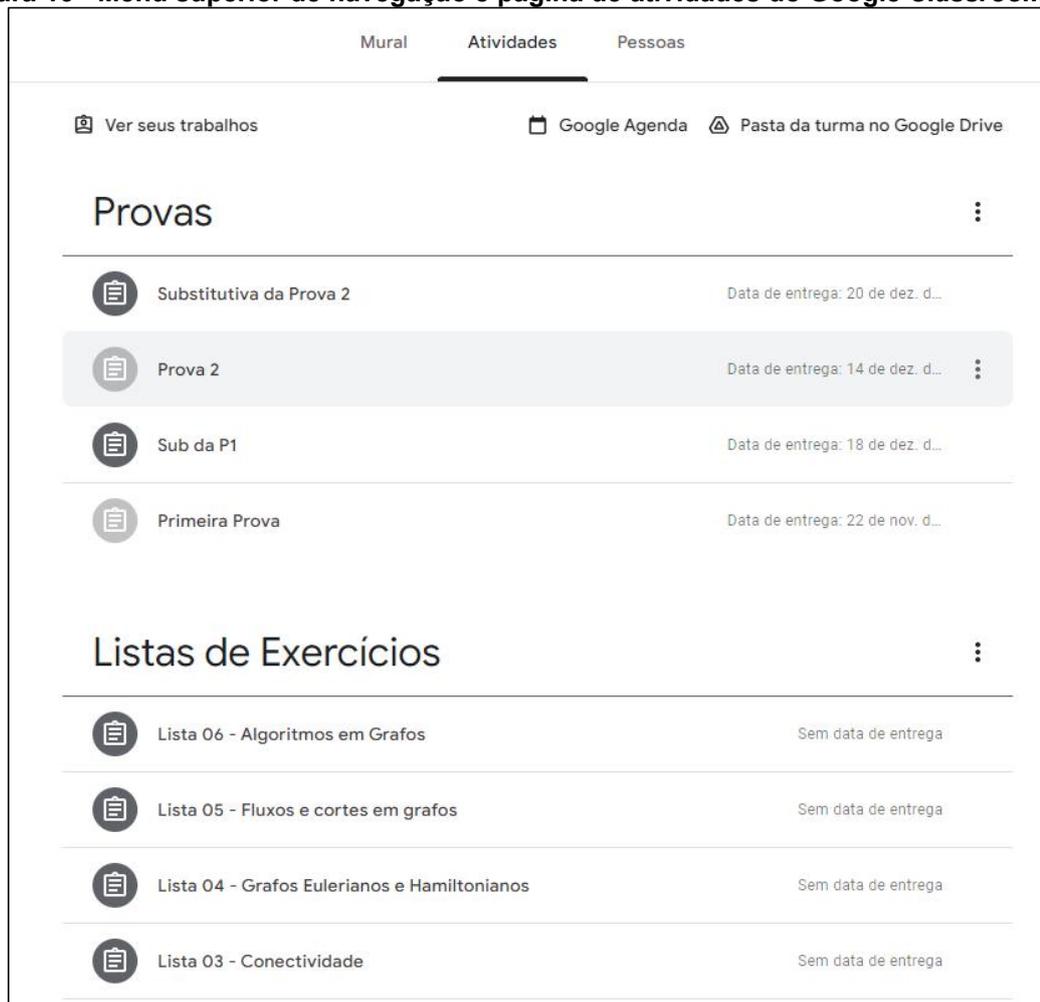
Figura 15 - Criação de atividade com quiz no *Classroom*

Fonte: Autoria própria (2022).

3.3.2 Visão do aluno da realização de atividades no Google Classroom

Semelhante ao *Moodle*, ao acessar a página de uma disciplina, o aluno pode visualizar a linha do tempo do curso, onde o professor consegue disponibilizar atividades para serem realizadas, além disso, é possível acessar uma aba específica de atividades no menu superior (Figura 16).

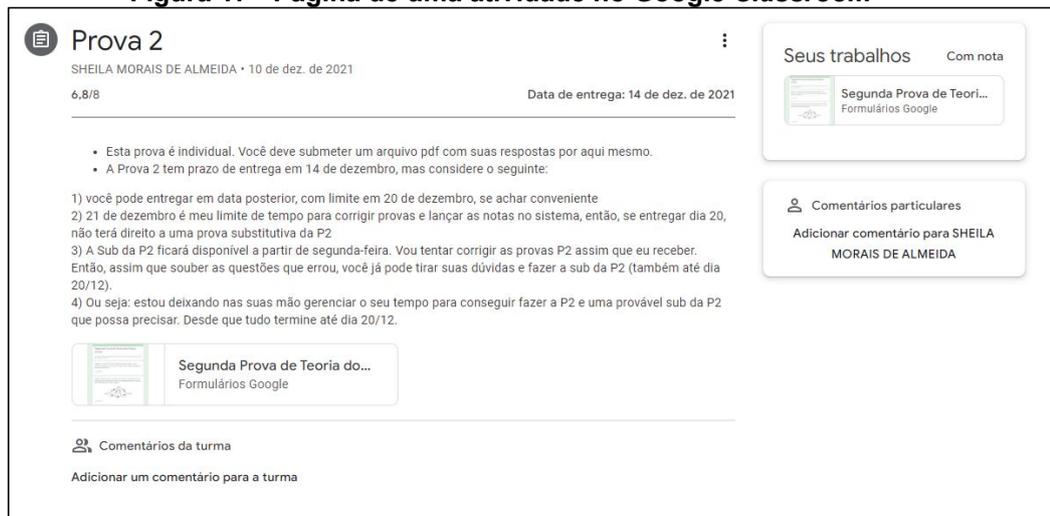
Figura 16 - Menu superior de navegação e página de atividades do Google Classroom



Fonte: Autoria própria (2022).

Uma atividade entregue pode ser avaliada pelo professor e o resultado dessa avaliação fica visível na postagem da atividade. Nessa postagem, é possível ler a descrição da atividade, sua data de entrega e o título. Também é possível enviar um comentário particular para o professor, ou postar um comentário público para a turma inteira. A página de uma atividade é exibida na Figura 17.

Figura 17 - Página de uma atividade no Google Classroom



Fonte: Autoria própria (2022).

3.3.3 Ferramentas de acessibilidade no Google Classroom

O *Classroom* não oferece muito apoio aos alunos com DI, dispondo apenas de uma funcionalidade de leitura de tela. Essa funcionalidade é útil para alunos com dificuldade de leitura ou algum tipo de condição que os impeça de ler. Ela reconhece o texto escrito na tela e transmite via som aquilo que foi reconhecido.

3.4 Gerenciamento de Atividades de uma Instituição de Educação Especial

A origem do sistema avaliado nessa subseção surgiu de um trabalho acadêmico realizado na disciplina de *Análise e Projetos Orientados a Objetos*, em 2021. Nela foi apresentada a necessidade de modelar um sistema de gerenciamento de atividades para uma instituição que atendia alunos com deficiência intelectual. Alunos com algum tipo de deficiência intelectual possuem maior dificuldade no processo de aprendizado. Por isto, é necessário que atividades sejam mais lúdicas, utilizando cores, sons, imagens e abordagens diferentes das tradicionalmente utilizadas (BEZERRA; ARAUJO, 2011). Algumas das características que os sistemas devem possuir para este público são:

- Utilização de cores: podem ser utilizadas para assimilação com sensações ou sentimentos. Além disso, chamam mais atenção do que o tradicional preto e branco utilizados em atividades (AMARAL; GOMEDI, 2004).
- Fontes em caixa alta: chamam mais atenção do que letras minúsculas, quando se tem uma palavra em caixa alta ela se torna destaque.

- Sons: músicas e sons auxiliam no desenvolvimento cognitivo das crianças (PUTKINEN, 2014).
- Vídeos: um vídeo consegue reproduzir a interação entre professor e aluno, diferente de um texto, o aluno consegue se transportar melhor para o cenário desejado pela atividade.

Alunos com DI podem, ou não, serem divididos em turmas por grau de deficiência referente à severidade de sua condição. As atividades também são adequadas seguindo esse grau de deficiência. Alunos com grau leve conseguem realizar atividades com maior quantidade de texto. Alunos com grau mediano conseguem realizar atividades com texto, mas com maior quantidade de elementos visuais. Alunos com grau severo muitas vezes só conseguem realizar atividades visuais (KE; LIU, 2015).

A partir do conhecimento sobre o público com DI e a necessidade de atender a instituição, o pesquisador deste trabalho desenvolveu na disciplina de *Análise e Projetos Orientados a Objetos* a prototipação e modelagem deste sistema. Como o sistema de gerenciamento de atividade na instituição não é informatizado, as características do sistema e seu funcionamento foram obtidos por meio de entrevistas com a orientadora deste trabalho que acompanha os alunos com DI desde 2017. Criou-se o Flora (Sistema de Gerenciamento de Atividades) o qual é detalhado nas próximas seções. O nome Flora foi inspirado por um personagem de um jogo virtual chamado *League of Legends* (RIOT, 2009). O personagem em questão faz parte de um modo de jogo chamado *Teamfight Tactics* (RIOT, 2019). A Figura 18 mostra o personagem chamado Rapininha Flora, que inspirou o nome do sistema.

Figura 18 - Rapininha Flora



Fonte: Riot Games (2019).

3.4.1 Funcionamento das atividades

As atividades são delegadas conforme turmas e grau de deficiência. O professor as elabora considerando a dificuldade dos alunos, usando vídeos,

imagens e cores. Os alunos recebem essas atividades e as realizam. Essas atividades podem ser realizadas com diferentes durações de tempo, as mais simples podem ser realizadas no mesmo dia e as mais complexas possuem prazo para entrega. Além disso, um aluno pode ter mais de uma atividade por matéria ou até mesmo atividades em diferentes matérias ao mesmo tempo.

Após a realização das atividades, os professores podem avaliar o desempenho do aluno com base em uma escala que informa se o aluno conseguiu realizar a atividade por total, parcial, se não conseguiu realizar ou se precisa melhorar em algum ponto. Esse *feedback* pode ser passado por escrito para os responsáveis do aluno, por meio de um arquivo em formato PDF, ou então, mantido no sistema. Os *feedbacks* podem ser entregues por: atividade, semana, mês ou bimestre. Além disso, o professor pode optar por gerar um *feedback* anual, compilando os resultados das atividades realizadas em todo o período.

O professor pode avaliar uma atividade no mesmo dia em que ela foi ministrada, ou então posteriormente, caso seja necessário ou desejado. Uma atividade mais complexa demanda um prazo maior para realização e correção, portanto, não é obrigatório que a correção seja no mesmo dia da elaboração da atividade.

Por fim, as atividades são avaliadas seguindo a escala Likert, de Rensis Likert, que tem como objetivo mensurar atitudes no contexto das ciências comportamentais. A escala de verificação de Likert consiste em tomar um ponto de referência, nesse caso, as atividades escolares, e desenvolver um conjunto de afirmações relacionadas à sua definição, para as quais os respondentes emitirão seu grau de concordância (JÚNIOR; COSTA, 2014). Em uma atividade, é possível utilizar a escala Likert para dizer que um aluno concluiu a atividade, realizou parcialmente ou não realizou.

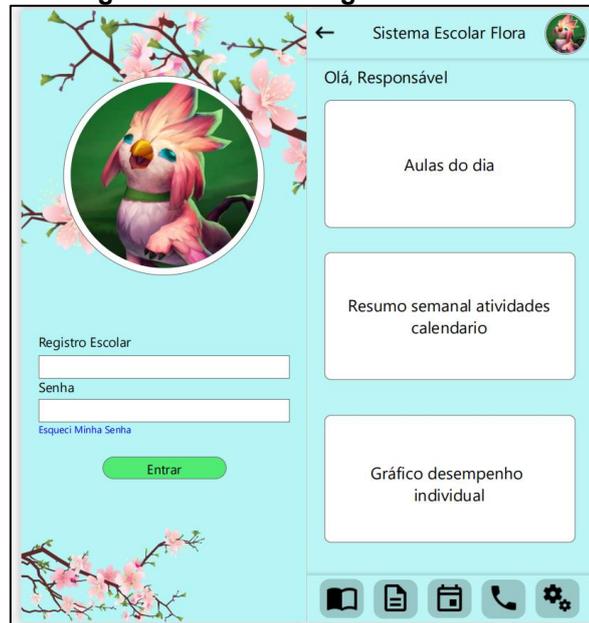
3.4.2 Prototipação do sistema

Para a disciplina cursada, foi solicitada a criação de um protótipo do sistema e das funcionalidades esperadas. O protótipo criado serviu de base para a inspiração do funcionamento do sistema real.

A Figura 19 exibe a tela de login e a tela inicial do protótipo na visão do aluno ou responsável. Na tela de login o usuário acessa o sistema com um registro escolar e com uma senha, ele também pode escolher recuperar a senha caso tenha

se esquecido. A tela inicial possui “cartões” de aulas do dia, resumo das atividades da semana e um gráfico de desempenho do aluno.

Figura 19 - Tela de login e tela inicial



Fonte: Autoria própria (2021).

A figura 20 mostra as atividades do dia do aluno, é possível filtrar por matéria e adicionar novas atividades. Também é possível visualizar a tela de boletim do aluno, cada matéria pode ser expandida para mostrar um detalhamento maior das avaliações.

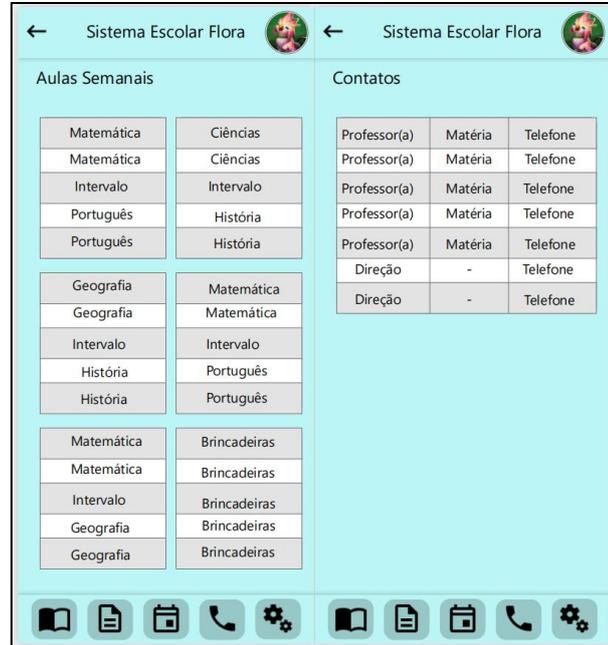
Figura 20 - Atividades do dia e boletim do aluno



Fonte: Autoria própria (2021).

A figura 21 mostra o calendário de aulas semanais e uma lista contatos importantes.

Figura 21 - Aulas semanais e contatos da escola



Fonte: Autoria própria (2021).

Existe também a visualização do professor, ele pode gerenciar atividades e acompanhar o desempenho da turma. Na Figura 22, a visualização de um gráfico de desempenho geral. Na Figura 23 a tela inicial do professor e a criação de atividades.

Figura 22 - Ranking de turmas e boletim



Fonte: Autoria própria (2021).

4 MODELAGEM DO SISTEMA USANDO LPS

O capítulo descreve a modelagem do sistema de gerenciamento de atividades seguindo o método adaptado de Delazeri e Wolf (2012). A Seção 4.1 apresenta a análise do domínio. A Seção 4.2 relata os requisitos do sistema em que se analisou os de núcleo, opcional e de aplicação. A Seção 4.3 traz os requisitos do domínio (núcleo). A Seção 4.4 apresenta o uso da linha para a implementação do Flora. Na Seção 4.5 os testes de aplicação do Flora. Por fim, na Seção 4.6 as considerações finais do capítulo.

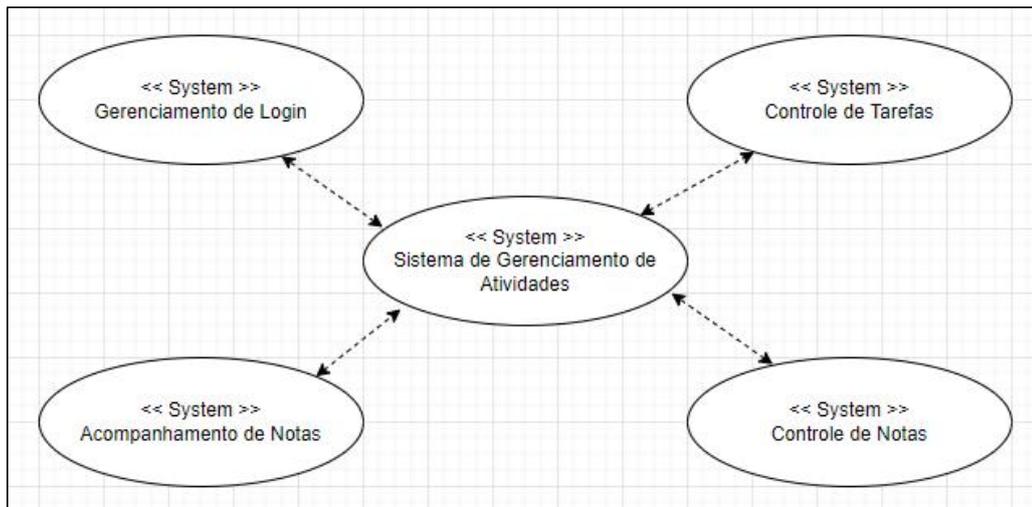
4.1 Análise do Domínio

Com as informações encontradas, foi possível levantar características comuns dos sistemas de gerenciamento de atividades que foram escolhidos para análise. Elas nortearam a modelagem do sistema, guiando a elaboração do diagrama de contexto. São elas:

- Controle de tarefas: um professor pode elaborar uma ou mais atividades para vários alunos diferentes. Essas atividades podem ser realizadas em um tempo determinado e avaliadas após sua entrega.
- Controle de notas: o professor pode avaliar as atividades realizadas pelos alunos e devolver notas para eles.
- Acompanhamento de notas: as notas lançadas pelos professores podem ser acompanhadas por alunos e responsáveis para acompanhamento do desempenho. Além disso, o próprio professor pode verificar como uma turma está se desenvolvendo.
- Gerenciamento de login: é necessário algum tipo de autenticação para validar a identidade de cada aluno, isso é importante do ponto de vista do sistema, para que cada aluno tenha acesso apenas aos seus conteúdos, notas e atividades.

O diagrama de contexto elaborado pode ser visto na Figura 24.

Figura 24 - Diagrama de contextos para gerenciamento de atividades



Fonte: Autoria própria (2022).

O Quadro 2 relaciona as características identificadas para os contextos propostos.

Quadro 2 - Relação de contextos e características

Contexto	Característica
Controle de Tarefas	Criar tarefas, editar tarefas, excluir tarefas, atribuir tarefa, visualizar tarefas.
Controle de Notas	Atribuir notas, editar notas, excluir notas.
Acompanhamento de Notas	Acompanhar notas.
Gerenciamento de Login	Criar cadastros, editar cadastros, excluir cadastros, gerenciar atividades de login.

Fonte: Autoria própria (2022).

4.2 Requisitos do Domínio

As características dos sistemas escolhidos para análise são exploradas nesta subseção. De modo geral, os três sistemas possuem características semelhantes no quesito gerenciamento de atividades.

4.2.1 Moodle

Para acessar a plataforma é necessário que o aluno possua um cadastro. Ao entrar, tem acesso aos cursos em que ele está matriculado. Caso não esteja matriculado em nenhum curso ele pode buscá-lo e se inscrever. Ao acessar a página do curso é possível visualizar as postagens realizadas pelo professor (conteúdos, avisos ou atividades). As atividades disponíveis podem ser realizadas pelo aluno e entregues ao professor que irá avaliá-las e devolver um *feedback*.

O Quadro 3 exibe a relação das características observadas no *Moodle* na perspectiva de um aluno, relacionadas com os contextos da análise de domínio.

Quadro 3 - Contextos e características do Moodle – Perspectiva do Aluno

Contexto	Característica	
	Interface Gráfica	Lógica
Gerenciamento de Login	Recuperar credenciais, login com outras plataformas e entrar como visitante	Manter login
Controle de Tarefas	Consultar Atividades	Manter Atividades, Atribuir Atividades, Avaliar Atividades, notificações de atividades e aviso de atividade atrasada
Controle de Notas	-	Manter notas e atribuir notas
Acompanhamento de notas	Checar notas das atividades	-

Fonte: Autoria própria (2022).

4.2.2 Google Classroom

Para acessar a plataforma é necessário que o aluno possua um cadastro, ao entrar, tem acesso aos cursos em que ele está matriculado. Caso não esteja matriculado em nenhum curso ele pode solicitar para que o professor o inscreva no curso ou então, acessá-lo por um link. Ao acessar a página do curso é possível visualizar as postagens realizadas pelo professor (conteúdos, avisos ou atividades).

As atividades disponíveis podem ser realizadas pelo aluno e entregues ao professor que irá avaliá-las e devolver um *feedback*. No Quadro 4 é possível verificar a relação entre as características encontradas e os contextos propostos na perspectiva de aluno.

Quadro 4 - Contextos e características do Classroom – Perspectiva do Aluno

Contexto	Característica	
	Interface Gráfica	Lógica
Gerenciamento de Login	Recuperar credenciais e Alternar contas	Manter Login
Controle de Tarefas	Consultar atividades	Manter atividades, atribuir atividades, avaliar atividades, gerenciar temas de atividades, acrescentar materiais, vincular com Google Agenda e Google Drive e reutilizar atividades
Controle de Notas	-	Manter notas, atribuir notas, gerar planilha de notas e importar notas automaticamente do Google Forms
Acompanhamento de notas	Checar notas das atividades	-

Fonte: Autoria própria (2022).

4.2.3 Flora

No sistema Flora, para gerenciamento de atividades de uma instituição de educação especial, o aluno irá acessar a plataforma por meio de um *login*, podendo recuperar suas credenciais. A tela inicial mostra seus cursos e ao acessá-los pode visualizar as atividades pendentes, as que já foram realizadas e as atividades futuras.

As atividades pendentes podem ser realizadas e enviadas para avaliação do professor que irá devolver um *feedback* para o aluno. Esses *feedbacks* podem ser visualizados posteriormente pela plataforma. O professor também pode gerar e entregar um *feedback* para o aluno no formato PDF. No Quadro 5 é exibido os contextos para o sistema Flora na perspectiva do aluno.

Quadro 5 - Contextos e características do app Flora

Contexto	Característica	
	Interface Gráfica	Lógica
Gerenciamento de Login	Recuperar credenciais	Manter Login
Controle de Tarefas	Consultar Atividades	Manter Atividades, Atribuir Atividades e Avaliar Atividades
Controle de Notas	-	Manter Notas, Atribuir Notas e Gerar PDF
Acompanhamento de notas	Checar Notas das Atividades	-

Fonte: Autoria própria (2022).

4.2.4 Classificação das características encontradas

Após realizado o levantamento das características dos três sistemas é possível gerar uma tabela para auxiliar a definir o que é parte do núcleo da LPS e o que pode ser classificada como característica opcional. O objetivo é encontrar as similaridades entre as aplicações e também suas variabilidades de domínio.

Durante a análise das plataformas, algumas características específicas das aplicações foram encontradas. O Quadro 6 apresenta as classificações das características. As características lógicas foram levantadas durante a análise dos sistemas, seus funcionamentos foram observados, abstraídos e agrupados conforme presença nos contextos propostos.

Quadro 6 - Classificações das características dos sistemas

Sistema	Contexto	Característica Lógica	Classificação	
Moodle	Controle de Tarefas	Manter atividades	Núcleo	
		Consultar atividades	Núcleo	
		Atribuir atividades	Núcleo	
		Avaliar atividades	Núcleo	
		Notificações de atividades	Opcional	
		Aviso de atividade atrasada	Opcional	
	Controle de Notas	Manter notas	Núcleo	
		Atribuir notas	Núcleo	
	Acompanhamento de Notas	Checar notas das atividades	Núcleo	
	Gerenciamento de Login	Manter login	Núcleo	
		Recuperar credenciais	Opcional	
		Login com outras plataformas	Aplicação	
Entrar como visitante		Aplicação		
Classroom	Controle de Tarefas	Manter atividades	Núcleo	
		Consultar atividades	Núcleo	
		Atribuir atividades	Núcleo	
		Avaliar atividades	Núcleo	
		Gerenciar temas de atividades	Aplicação	
		Acrescentar materiais	Aplicação	
		Vincular com Google Agenda e Google Drive	Aplicação	
		Reutilizar atividades	Aplicação	
	Controle de Notas	Manter notas	Núcleo	
		Atribuir notas	Núcleo	
		Gerar planilha de notas	Aplicação	
		Importar notas automaticamente do Google Forms	Aplicação	
	Acompanhamento de Notas	Checar notas das atividades	Núcleo	
	Gerenciamento de Login	Manter login	Núcleo	
		Recuperar credenciais	Opcional	
		Alternar contas	Aplicação	
	Flora	Controle de Tarefas	Manter atividades	Núcleo
			Consultar atividades	Núcleo
Atribuir atividades			Núcleo	
Avaliar atividades			Núcleo	
Controle de Notas		Manter notas	Núcleo	
		Atribuir notas	Núcleo	
		Gerar PDF	Aplicação	
Acompanhamento de Notas		Checar notas das atividades	Núcleo	
Gerenciamento de Login		Manter login	Núcleo	
		Recuperar credenciais	Opcional	

Fonte: Autoria própria (2022).

No Quadro 7 um comparativo visual de quais características levantadas são núcleo da LPS (verde), quais serão opcionais (amarelo) e quais são aplicações (vermelho) e não farão parte do domínio.

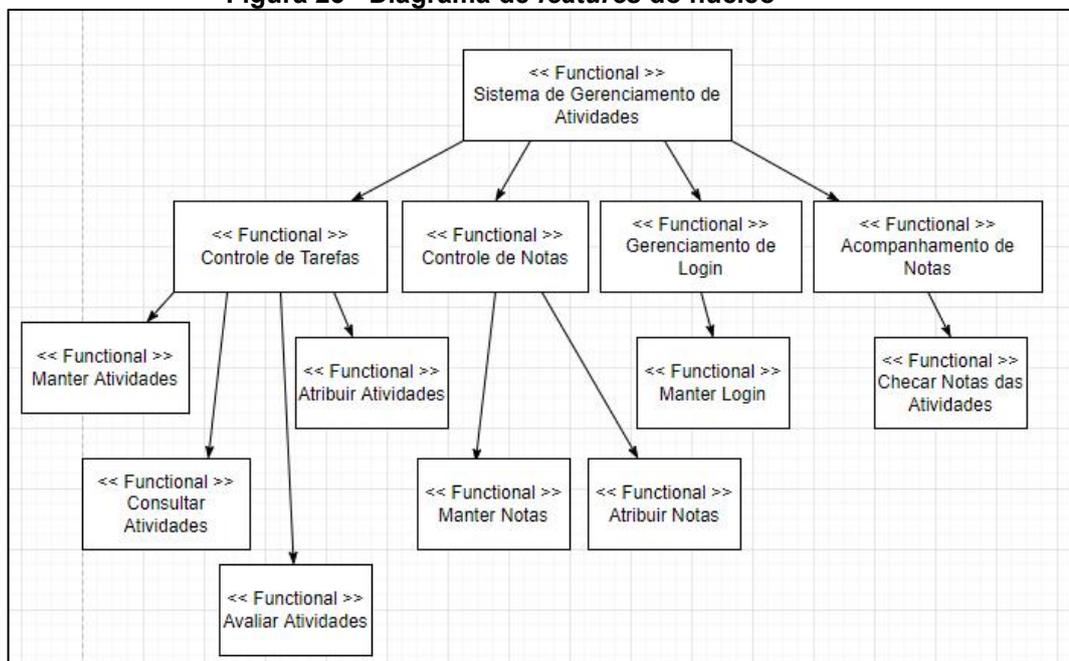
Quadro 7 - Comparativo de características presentes nas plataformas

Característica	Moodle	Classroom	Flora	Classificação
Manter atividades	Sim	Sim	Sim	Núcleo
Consultar atividades	Sim	Sim	Sim	Núcleo
Atribuir atividades	Sim	Sim	Sim	Núcleo
Avaliar atividades	Sim	Sim	Sim	Núcleo
Manter notas	Sim	Sim	Sim	Núcleo
Atribuir notas	Sim	Sim	Sim	Núcleo
Checar notas das atividades	Sim	Sim	Sim	Núcleo
Manter login	Sim	Sim	Sim	Núcleo
Notificações de atividades	Sim	Sim	Não	Opcional
Aviso de atividade atrasada	Sim	Sim	Não	Opcional
Recuperar credenciais	Sim	Sim	Não	Opcional
Login com outras plataformas	Sim	Não	Não	Aplicação
Entrar como visitante	Sim	Não	Não	Aplicação
Gerenciar temas de atividades	Não	Sim	Não	Aplicação
Acrescentar materiais	Não	Sim	Não	Aplicação
Vincular com Google Agenda e Google Drive	Não	Sim	Não	Aplicação
Reutilizar atividades	Não	Sim	Não	Aplicação
Alternar contas	Não	Sim	Não	Aplicação
Gerar PDF	Não	Não	Sim	Aplicação

Fonte: Autoria própria (2022).

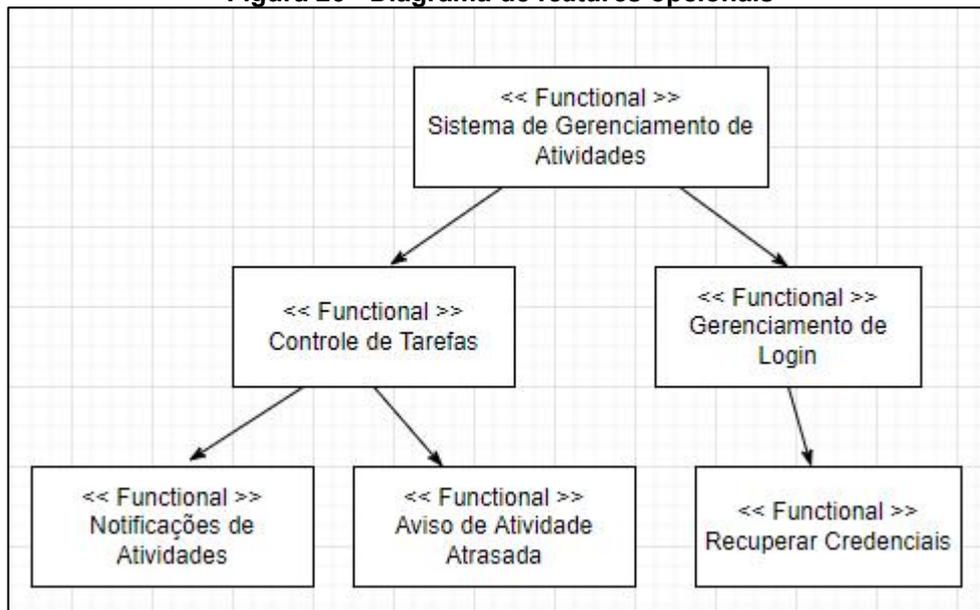
Após definir o que é núcleo e o que é opcional, é possível gerar o diagrama de *features* do núcleo (Figura 25) e o diagrama de *features* das características opcionais (Figura 26).

Figura 25 - Diagrama de *features* do núcleo



Fonte: Autoria própria (2022).

Figura 26 - Diagrama de *features* opcionais



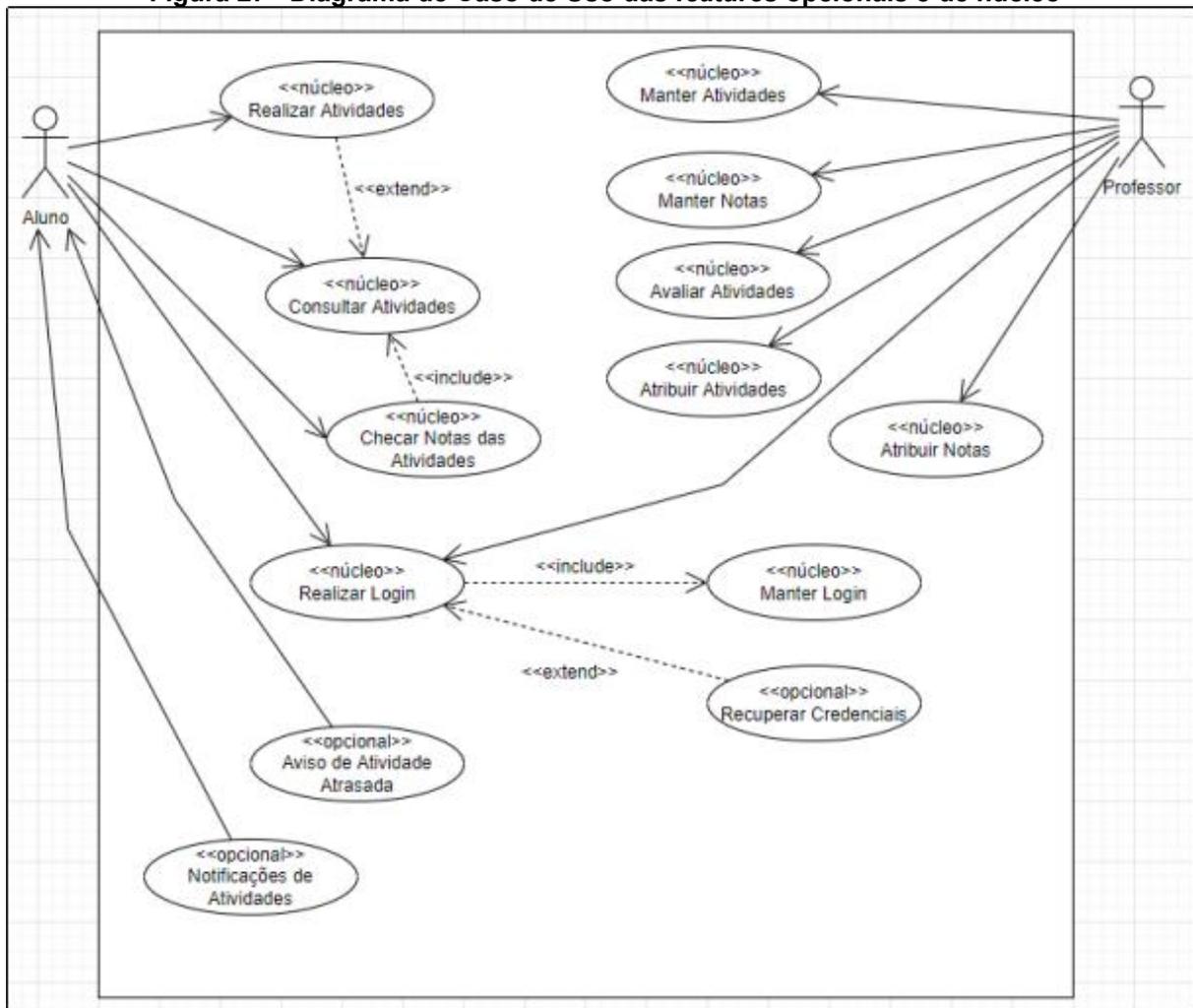
Fonte: Autoria própria (2022).

No diagrama de *features* opcionais foi incluído o *Controle de Tarefas* e o *Gerenciamento de Login*, assim como no diagrama de *features* de núcleo. Para as *features* opcionais, o controle de tarefas possui a notificação de atividades e de atividades atrasadas. Já para o gerenciamento de *login* possui o recuperar credenciais.

4.2.5 Modelagem do Domínio

Nessa etapa é feita uma análise do problema em questão, de como ele é solucionado e como essa solução irá funcionar. O diagrama de caso de uso consegue representar o funcionamento da ferramenta. Além do diagrama de caso de uso, tem-se também a descrição de seus cenários. A Figura 27 mostra o diagrama proposto para as *features* de núcleo e opcionais e posteriormente seus cenários.

Figura 27 - Diagrama de Caso de Uso das *features* opcionais e de núcleo



Fonte: Autoria própria (2022).

Os casos de uso possuem uma rotulação para indicar se são núcleo, opcionais ou aplicação. Os atores encontrados são: aluno e professor. Os casos de uso: *Manter Login*, *Recuperar Credenciais*, *Validar Conta*, *Notificações de Atividades* e *Aviso de Atividade Atrasada* são geridos pelo próprio sistema e não possuem um ator para executá-los. Como a funcionalidade dos casos de uso é semelhante, optou-se por descrever os cenários dos mais importantes para o funcionamento do sistema. O Quadro 8 descreve o cenário para o caso de uso *Realizar Atividades*.

Quadro 8 - Descrição para o caso de uso Realizar Atividades

Caso de Uso	Realizar Atividades.
Descrição	Possibilita que o aluno realize uma atividade cadastrada no sistema.
Pré Condições	Aluno estar logado no sistema, existir uma atividade cadastrada no sistema e existir atividade atribuída para ele.
Fluxo Básico	
Ação do Ator	Resposta do Sistema
1 - O aluno acessa a página de atividades. 3 - O aluno realiza uma atividade. 4 - O aluno envia a atividade para avaliação.	2 - O sistema requisita o caso de uso “Consultar Atividades” e retorna as atividades atribuídas para ele. [A1] . 5 - O sistema envia uma mensagem de que a atividade foi realizada para o aluno. [A2] .
Fluxo Alternativo	
[A1] - Sem atividades para mostrar 2.1 - Não existem atividades cadastradas ou o aluno não possui atividade atribuída. 2.2 - Retorna ao passo 1. [A2] - Não foi possível concluir o envio da atividade 5.1 - O sistema informa que houve um erro e a atividade não pode ser enviada. 5.2 - Retorna ao passo 4.	
Pós Condições	A atividade foi enviada para avaliação.

Fonte: Autoria própria (2022).

O quadro 9 descreve o cenário para o caso de uso *Realizar Login*.

Quadro 9 - Descrição para o caso de uso Realizar Login

Caso de Uso	Realizar Login.
Descrição	Possibilita que o usuário acesse o sistema.
Pré Condições	Usuário possuir cadastro ativo.
Fluxo Básico	
Ação do Ator	Resposta do Sistema
1 - O usuário acessa o sistema. 3 - O usuário insere as credenciais. 6 - O usuário solicita a recuperação de credenciais.	2 - O sistema mostra a tela de login. [A1] . 4 - O sistema requisita o caso de uso “Manter Login” e checa se o cadastro existe. [A2] . 5 - O sistema retorna a tela inicial do sistema. 7 - O sistema retorna a tela de recuperação de credenciais. [A3]
Fluxo Alternativo	
[A1] - O sistema está fora do ar 2.1 - Retorna mensagem avisando que o sistema não está disponível. [A2] - O cadastro do usuário não existe ou as credenciais estão incorretas 4.1 - O sistema informa que o cadastro não existe ou então que as credenciais estão incorretas. 4.2 - Retorna ao passo 3. [A3] - O usuário solicita para recuperar as credenciais. 7.1 - O sistema requisita o caso de uso de “Recuperar Credenciais”. 7.2 - O usuário recupera as credenciais. 7.3 - Retorna ao passo 3.	
Pós Condições	O usuário está logado no sistema.

Fonte: Autoria própria (2022).

O quadro 10 descreve o cenário para o caso de uso *Avaliar Atividades*.

Quadro 10 - Descrição para o caso de uso Avaliar Atividades

Caso de Uso	Avaliar Atividades.
Descrição	Possibilita que o professor avalie as atividades de um ou mais alunos.
Pré Condições	Professor estar logado no sistema, existirem atividades cadastradas no sistema, existirem atividades atribuídas para os alunos e existirem atividades realizadas e entregues pelos alunos.
Fluxo Básico	
Ação do Ator	Resposta do Sistema
1 - O professor acessa a página de atividades. 3 - O professor avalia uma atividade. 4 - O professor atribui notas.	2 - O sistema requisita o caso de uso “Consultar Atividades” e retorna as atividades realizadas pelos alunos. [A1] . 5 - O sistema envia uma mensagem de que a atividade foi avaliada e as notas atribuídas com sucesso. [A2] .
Fluxo Alternativo	
[A1] - Sem atividades para mostrar 2.1 - Não existem atividades cadastradas ou não existem atividades entregues. 2.2 - Retorna ao passo 1. [A2] - Não foi possível concluir a avaliação da atividade 5.1 - O sistema informa que houve um erro e a atividade não pode ser avaliada. 5.2 - Retorna ao passo 4.	
Pós Condições	A atividade foi avaliada.

Fonte: Autoria própria (2022).

O quadro 11 descreve o cenário para o caso de uso *Atribuir Atividades*.

Quadro 11 - Descrição para o caso de uso Atribuir Atividades

Caso de Uso	Atribuir Atividades.
Descrição	Possibilita que o professor atribua atividades existentes para um ou mais alunos.
Pré Condições	Professor estar logado no sistema e existirem atividades cadastradas.
Fluxo Básico	
Ação do Ator	Resposta do Sistema
1 - O professor acessa a página de atividades. 3 - O professor atribui uma atividade.	2 - O sistema requisita o caso de uso “Consultar Atividades” e retorna as atividades disponíveis para serem atribuídas. [A1] . 4 - O sistema envia uma mensagem de que a atividade foi atribuída para o aluno. [A2] .
Fluxo Alternativo	
[A1] - Sem atividades para atribuir 2.1 - Não existem atividades cadastradas disponíveis para serem atribuídas. 2.2 - Retorna ao passo 1. [A2] - Não foi possível concluir a atribuição de uma atividade. 4.1 - O sistema informa que houve um erro e a atividade não pode ser atribuída. 4.2 - Retorna ao passo 3.	
Pós Condições	A atividade foi atribuída e o aluno pode realizá-la.

Fonte: Autoria própria (2022).

O quadro 12 descreve o cenário para o caso de uso *Checar Notas das Atividades*.

Quadro 12 - Descrição para o caso de uso Checar Notas das Atividades

Caso de Uso	Checar Notas das Atividades
Descrição	Possibilita que o usuário verifique as notas de uma ou mais atividades.
Pré Condições	Usuário estar logado no sistema e existir uma atividade avaliada no sistema.
Fluxo Básico	
Ação do Ator	Resposta do Sistema
1 - O usuário acessa a página de atividades. 3 - O usuário escolhe uma atividade para visualizar nota.	2 - O sistema requisita o caso de uso "Consultar Atividades" e retorna as atividades avaliadas. [A1] . 4 - O sistema requisita o caso de uso "Checar Notas das Atividades". 5 - O sistema retorna a nota da atividade. [A2]
Fluxo Alternativo	
[A1] - Permissão de usuário ou professor 2.1 - Se o usuário for um aluno, retorna apenas as próprias atividades. Se o usuário for um professor, retorna todas as atividades que ele vinculou para os alunos (visibilidade geral). 2.2 - Retorna ao passo 3. [A2] - Não foi possível retornar a nota da atividade 5.1 - O sistema informa que houve um erro e a nota não pode ser mostrada. 5.2 - Retorna ao passo 3.	
Pós Condições	A nota foi visualizada pelo usuário.

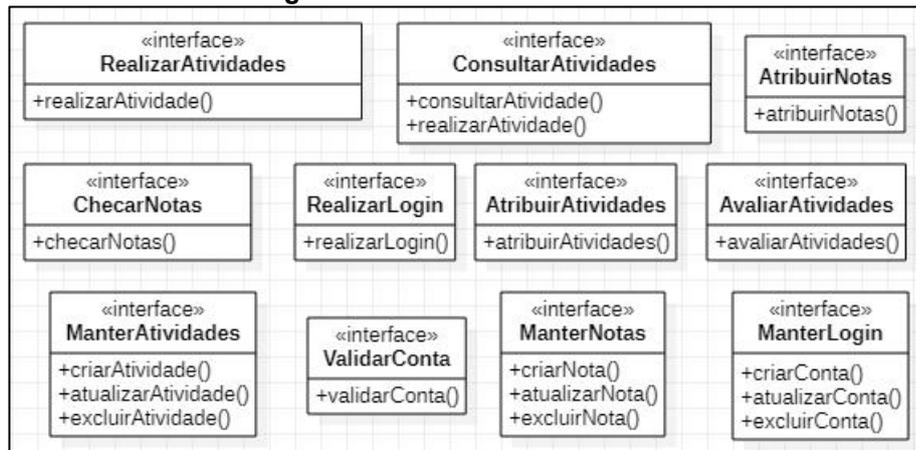
Fonte: Autoria própria (2022).

4.2.6 Projeto do Domínio

Na etapa de projeto de domínio utiliza-se a abordagem de desenvolvimento de software baseado em componentes. Foi escolhido utilizar o processo de *UML Components*, proposto por Cheesman e Daniels (2000). O desenvolvimento baseado em componentes é adotado para trazer uma visão de quais componentes são implementados e como eles se comportam. A maior vantagem dessa abordagem é a facilidade de entendimento que ela traz para o dono do produto, os desenvolvedores, os arquitetos de softwares e os gerentes de projeto. O planejamento de desenvolvimento acaba sendo melhor, já que é possível premeditar como os componentes serão desenvolvidos e quais habilidades serão necessárias para o trabalho (BELL, 2004).

Primeiramente, o diagrama de caso de uso foi utilizado para definir as interfaces do sistema. Cada caso de uso foi transformado em uma interface na Figura 28.

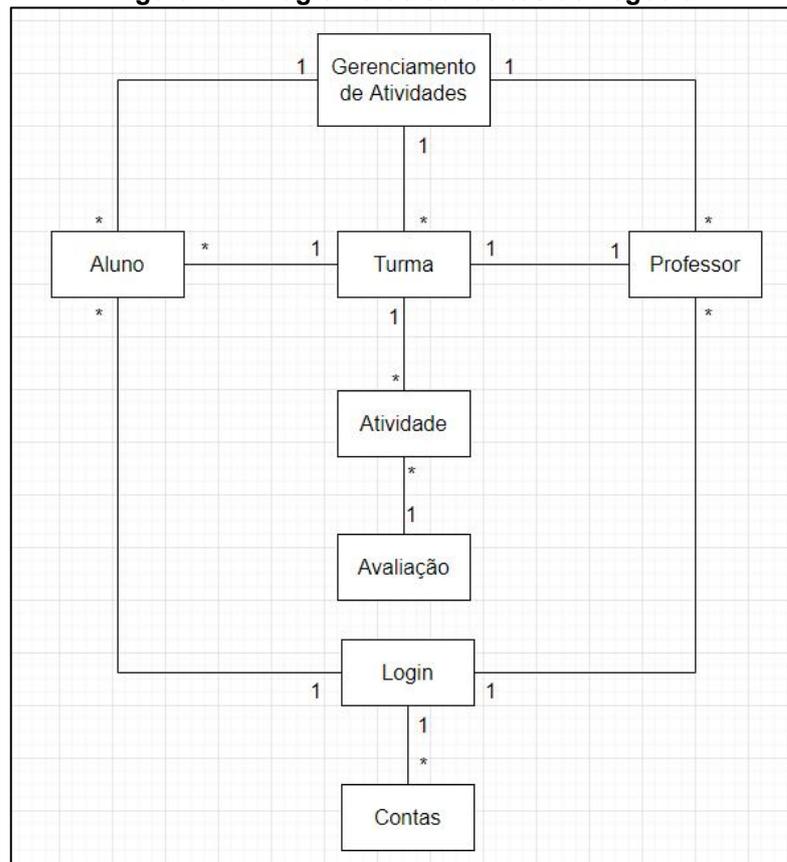
Figura 28 - Interfaces do sistema



Fonte: Autoria própria (2022).

O próximo passo é definir o diagrama de conceitos de negócio. As informações que o sistema deve gerir são abstraídas e representadas no diagrama de conceitos de negócio. A Figura 29 apresenta o diagrama de conceitos de negócio do gerenciamento de atividades.

Figura 29 - Diagrama de conceitos de negócio



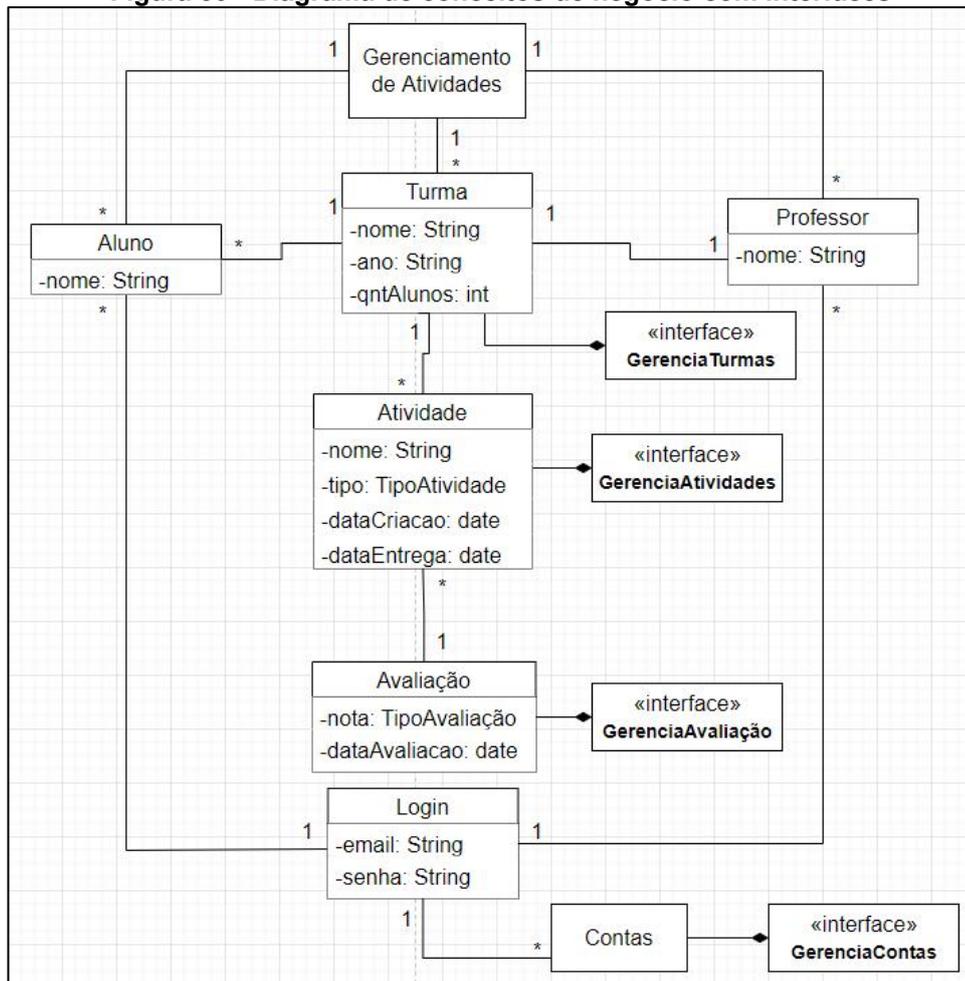
Fonte: Autoria própria (2022).

Os conceitos elaborados tem relação com os casos de uso propostos no diagrama de caso de uso apresentado anteriormente. O conceito *Avaliação*

contempla a realização da avaliação de uma atividade. O conceito *Login* gerencia o login dos usuários, já o conceito *Contas* tem como objetivo armazenar as credenciais dos usuários e validar as contas de quem está solicitando o login para assegurar que a conta é válida.

Após elaborar o diagrama de conceitos de negócio, é possível identificar as interfaces que são parte do negócio e com isso incrementar o diagrama. A Figura 30 ilustra o diagrama de conceitos de negócios com as respectivas interfaces.

Figura 30 - Diagrama de conceitos de negócio com interfaces



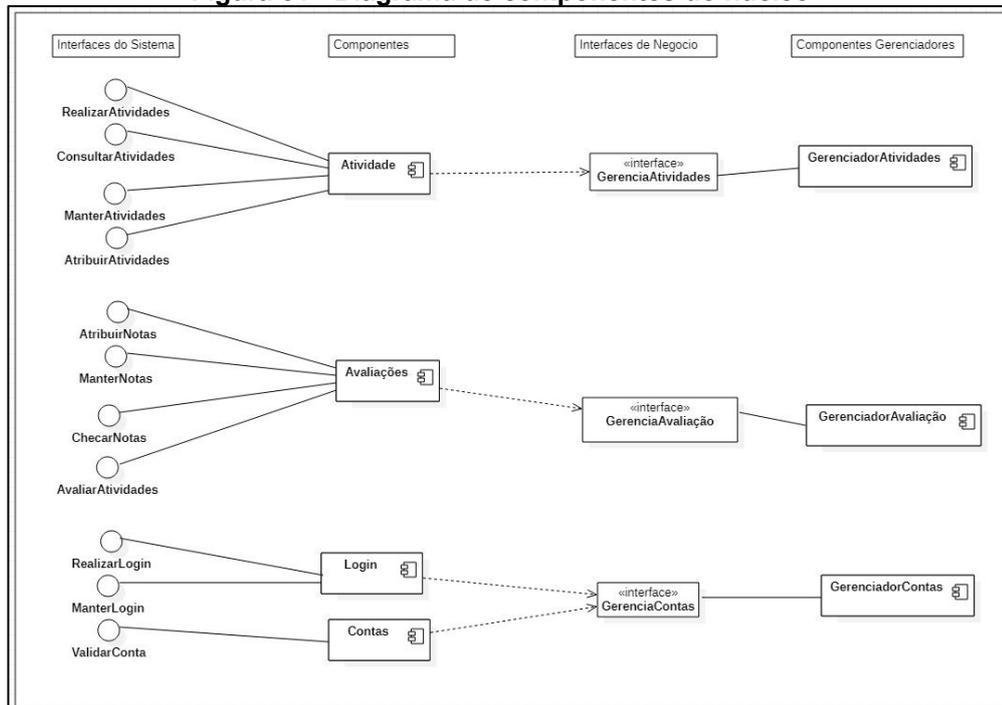
Fonte: Autoria própria (2022).

No conceito *Atividade*, tem-se o atributo *tipo* do tipo *TipoAtividade*, isso pois existem tipos distintos de atividades que podem ser criadas (questões objetivas, múltipla escolha, leitura de texto, assistir vídeos, ouvir algo, etc). O conceito *Avaliação* possui o atributo *nota* do tipo *TipoAvaliacao* pois uma nota pode ser numérica, textual, simbólica ou visual.

Para gerenciar as contas, foi criada uma interface *GerenciaContas*. Para gerenciar as atividades foi criada uma interface *GerenciaAtividades*. Para o gerenciamento das avaliações das atividades foi criada a interface *GerenciaAvaliação*. O gerenciamento das turmas é feito pela *GerenciaTurmas*.

Depois de gerar os diagramas, inicia-se o processo de identificação dos componentes. Um componente pode reunir várias interfaces ao mesmo tempo, como é o caso da realização de atividades que reúne a interface de consultar atividade e realizar atividade. A Figura 31 exibe o diagrama gerado.

Figura 31 - Diagrama de componentes do núcleo



Fonte: Autoria própria (2022).

Os componentes codificam as interfaces do sistema com quem se ligam, esses componentes são instanciados dentro de suas interfaces de negócio. Alguns componentes dependem de outras instanciações de componentes em interfaces diferentes. Por exemplo, a interface *ChecarNotas*, codificada em *Avaliações*, depende da interface *RealizarAtividades* e *AvaliarAtividades*, já que não é possível consultar a avaliação de uma atividade que não foi realizada e avaliada.

Outro exemplo são todas as operações realizadas no sistema, pois elas dependem da interface de *RealizarLogin* e *ValidarConta*, mesmo não possuindo conexão direta com o componente *Login*. Os componentes gerenciadores são reutilizáveis dentro do escopo da LPS.

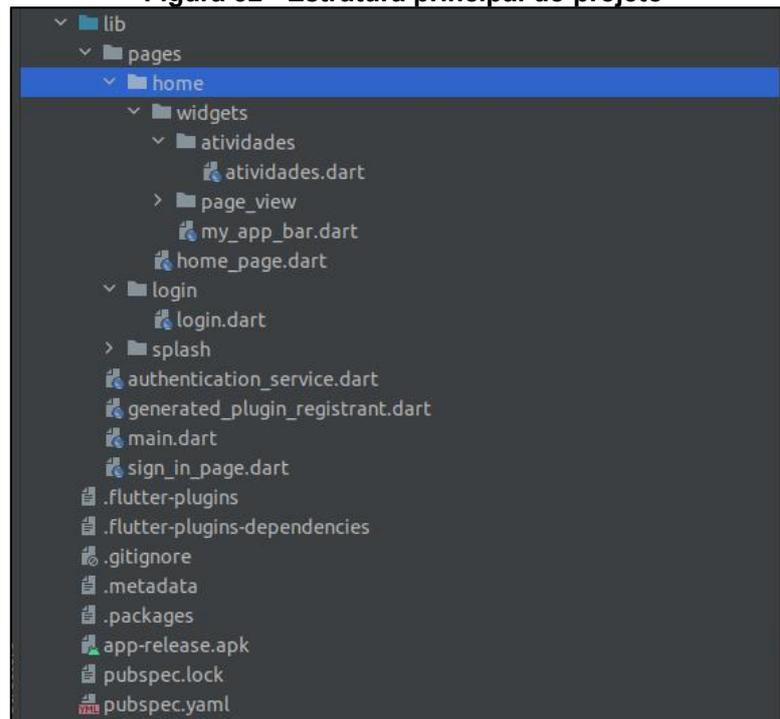
4.2.7 Implementação do Domínio

Nesta subfase foi feita a codificação da linha de produto. O objetivo foi criar um aplicativo *mobile* baseado no *kit* de desenvolvimento Flutter (GOOGLE, 2015), baseado na linguagem de programação Dart (GOOGLE, 2011).

O banco de dados utilizado para o projeto foi o *Firestore Realtime Database*, da plataforma *Firestore*, gerenciada pelo Google (GOOGLE, 2014). Seu ponto forte é facilitar o desenvolvimento do banco de dados para a aplicação, possibilitando que o desenvolvedor foque mais na parte visível do aplicativo.

A aplicação foi desenvolvida na *IDE (Integrated Development Environment) Android Studio* (GOOGLE, 2013). Android Studio é um dos principais ambientes para o desenvolvimento de aplicativos para celular. Com ele é possível customizar qual a versão alvo do sistema operacional, versão mínima para funcionar o aplicativo, facilitação na programação por possuir muitos modelos prontos para desenvolvimento e também a possibilidade de conectar seu próprio aparelho para emular em tempo real aquilo que está sendo desenvolvido. A estrutura principal do projeto pode ser visualizada na Figura 32. As páginas do aplicativo são divididas em pastas para organizar melhor o projeto, o arquivo *authentication_service.dart* é responsável pela autenticação de login. O *login.dart* gerencia o login da aplicação e possui a interface gráfica do mesmo. O projeto inicia pela *main.dart* faz as chamadas necessárias para o funcionamento da aplicação. O *home_page.dart* cria a tela inicial do aplicativo, nesse arquivo existem chamadas para as funções de outros arquivos que iniciam as funcionalidades específicas, isso é feito por meio da importação de arquivos.

Figura 32 - Estrutura principal do projeto



Fonte: Autoria própria (2022).

Para autenticar as credenciais do usuário, é utilizada uma função do *Firebase*, na Figura 33 o trecho de código que autentica o login. Ela recebe o e-mail e a senha como parâmetros e valida com o banco de dados para garantir que aquele usuário existe, caso exista o login é liberado, se não, uma mensagem de erro do próprio banco é enviada.

Figura 33 - Autenticação de login Flora

```
Future<String> signIn({String email, String password}) async {
  try {
    await _firebaseAuth.signInWithEmailAndPassword(
      email: email, password: password);
    return "Logado";
  } on FirebaseAuthException catch (e) {
    return e.message;
  }
}
```

Fonte: Autoria própria (2022).

A classe *ManterAtividades* foi codificada como *Atividade* na Figura 34.

Figura 34 - Classe Atividade

```

class Atividade extends StatefulWidget {

  @override
  Atividades createState() => Atividades();

  String nome;
  String TipoAtividade;
  int id;
  String dataEntrega;
  String dataCriacao;
  bool entregue;

  excluirAtividade(id) {
    return id;
  }

  criarAtividade(nome, tipo) {
    this.nome = nome;
    this.TipoAtividade = TipoAtividade;
    this.dataCriacao = dataCriacao;
    return this;
  }

  atualizarAtividade(nome, tipo) {
    this.nome = nome;
    this.tipo = tipo;
    return this;
  }
}

```

Fonte: Autoria própria (2022).

Para a criação de atividades, a função `_showDialog` da Figura 35 foi utilizada. Ela codifica a interface gráfica do sistema responsável pela criação das atividades, cria um campo para inserir o nome, um botão para salvar e outro para cancelar.

Figura 35 - Função de criação de atividades Flora

```

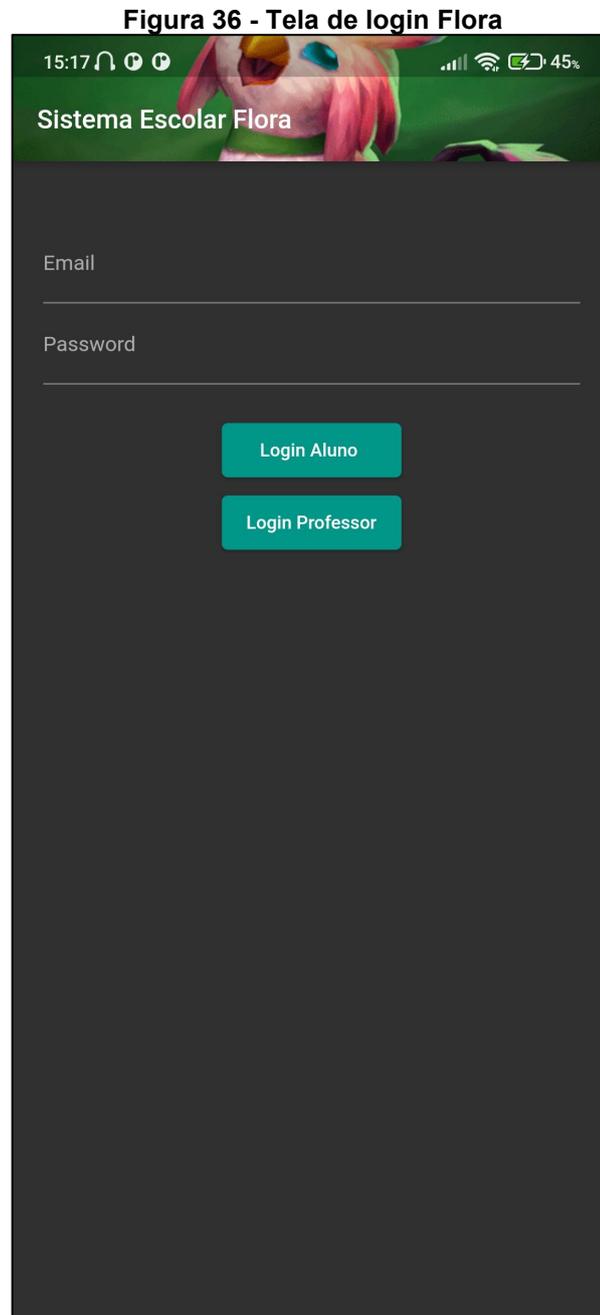
void _showDialog(BuildContext context) {
  showDialog(
    context: context,
    builder: (BuildContext context) {
      return AlertDialog(
        backgroundColor: Colors.teal,
        title: Text('Acrescentar atividade'),
        content: TextField(
          decoration: InputDecoration(hintText: "Nome"),
        ), // TextField

        actions: <Widget>[
          ElevatedButton(
            child: Text("Salvar"),
            style: ElevatedButton.styleFrom(primary: Colors.blue),
            onPressed: () {
              Navigator.of(context).pop();
            }, // ElevatedButton
          ),
          ElevatedButton(
            child: Text("Cancelar"),
            style: ElevatedButton.styleFrom(primary: Colors.red),
            onPressed: () {
              Navigator.of(context).pop();
            }, // ElevatedButton
          ), // <Widget>[]
        ], // <Widget>[]
      ); // AlertDialog
    }
  );
}

```

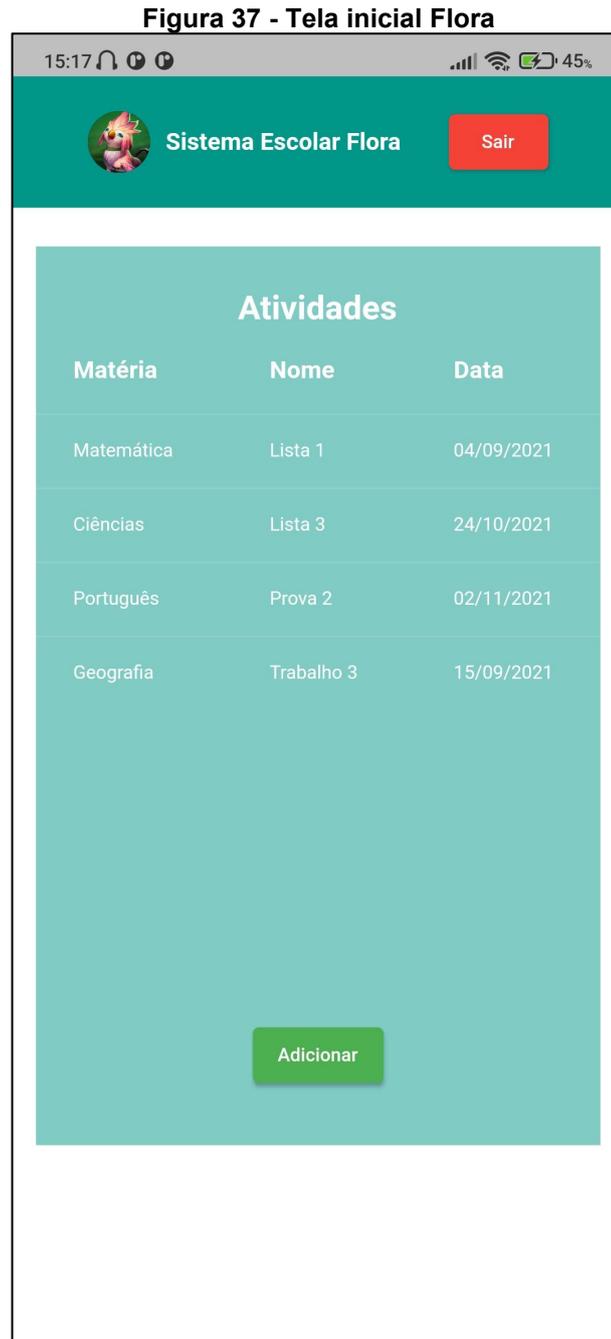
Fonte: Autoria própria (2022).

A primeira parte a ser desenvolvida foi a tela de login (Figura 36), tendo em vista que a autenticação do usuário é uma parte importante da ferramenta, servindo para diferenciar quem é aluno ou professor. A tela de login possui dois formulários, um para inserir o e-mail e outro para inserir a senha.



Fonte: Autoria própria (2022).

Depois, foi desenvolvida a tela inicial do aplicativo (Figura 37), que exibe a visão geral do sistema.



Fonte: Autoria própria (2022).

A seguir, a tela de criação de atividades foi codificada, Figura 38.

Figura 38 - Criação de atividades Flora

Fonte: Autoria própria (2022).

4.2.8 Teste do Domínio

Os testes realizados na aplicação foram os testes de aceitação, ou seja, verificar se o produto final condiz com os requisitos levantados durante todo o desenvolvimento do projeto.

4.3 Requisitos da Aplicação

Durante a subfase de requisitos do domínio, foi elaborado um quadro com as características presentes nos três sistemas analisados. As características de aplicação foram: *Gerar PDF*, *Alternar Contas*, *Reutilizar Atividades*, *Vincular com Google Agenda e Google Drive*, *Gerenciar Temas de Atividades*, *Entrar como Visitante* e *Login com Outras Plataformas*. Essas características não foram implementadas, pois são particularidades dos sistemas.

As características opcionais foram: *Recuperar Credenciais*, *Aviso de Atividade Atrasada* e *Notificações de Atividades*. Todas as três fazem parte do *Moodle* e *Classroom*, mas não fazem parte do escopo definido para o sistema *Flora* e, portanto, não foram implementadas.

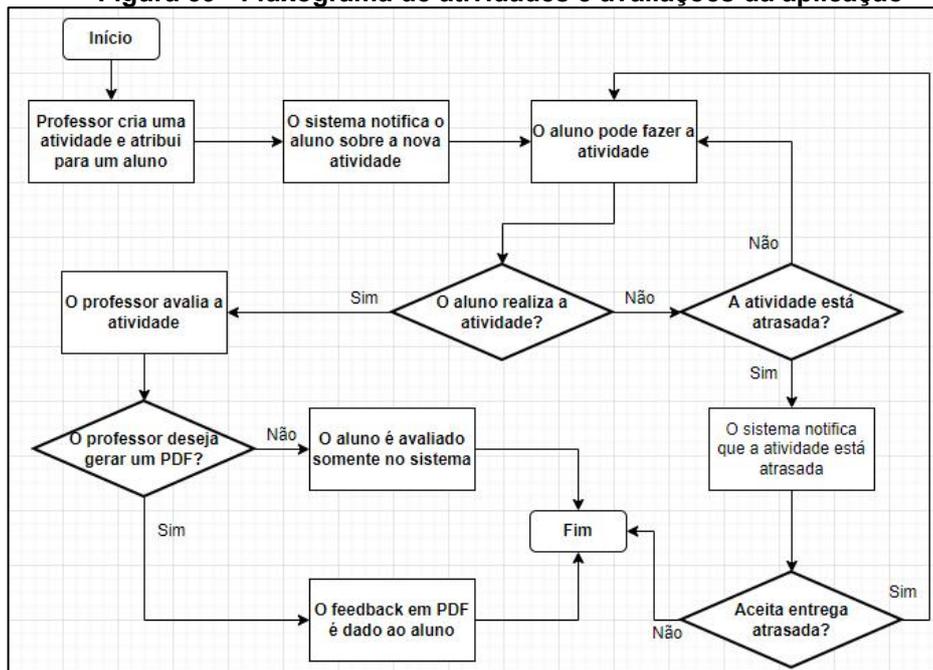
É importante ressaltar que apesar de algumas características estarem presentes em dois sistemas distintos, elas podem funcionar de maneira diferente. É o caso da notificação de atividade atrasada, no *Moodle* é possível ver na própria atividade um aviso de que a atividade está atrasada. No *Classroom* um e-mail é enviado para o aluno. Essas variabilidades teriam que ser consideradas durante a implementação.

Além disso, algumas das características de núcleo também podem apresentar variabilidade dependendo da aplicação. O funcionamento da criação de tarefas é o mesmo, mas no *Classroom* existe a possibilidade de utilizar a característica opcional *Vincular com Google Agenda e Google Drive* para integração com a atividade.

Em relação ao login, na plataforma *Moodle* o login pode ser realizado utilizando uma conta Google, por isso, foi proposto o caso de uso *Validar Conta* já que é necessário realizar uma validação dessa conta utilizada para acessar o sistema.

Para esta etapa foram escolhidas algumas *features* que possuem relacionamento entre si, isso foi feito para construir modelos mais completos com várias *features*, tal como a etapa de requisitos do domínio. A Figura 39 mostra um fluxograma do funcionamento da notificação de atividades, notificação de atividades atrasadas e a geração de PDF de uma avaliação realizada pelo professor.

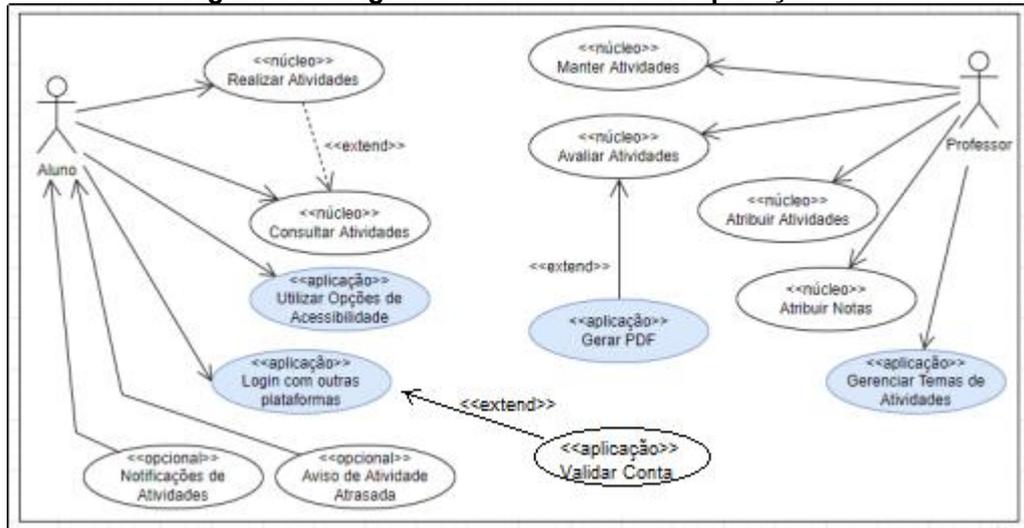
Figura 39 - Fluxograma de atividades e avaliações da aplicação



Fonte: Autoria própria (2022).

Após a elaboração do fluxograma, foi criado um diagrama de caso de uso (Figura 40) para representar as características observadas. Ele contempla características de núcleo, opcionais e de aplicação. O caso de uso *Gerar PDF* é utilizado quando o professor deseja gerar um PDF de uma avaliação. O caso de uso *Utilizar Opções de Acessibilidade* contempla as opções de acessibilidade observadas na plataforma *Moodle* que foram apresentadas anteriormente (3.2.3). O caso de uso *Login com outras plataformas* trata da opção da plataforma *Moodle* de acessar o sistema com uma conta *Google*. O caso de uso *Gerenciar Temas de Atividades* contempla a opção do *Google Classroom* em que um professor pode determinar um tema para uma atividade que está sendo criada.

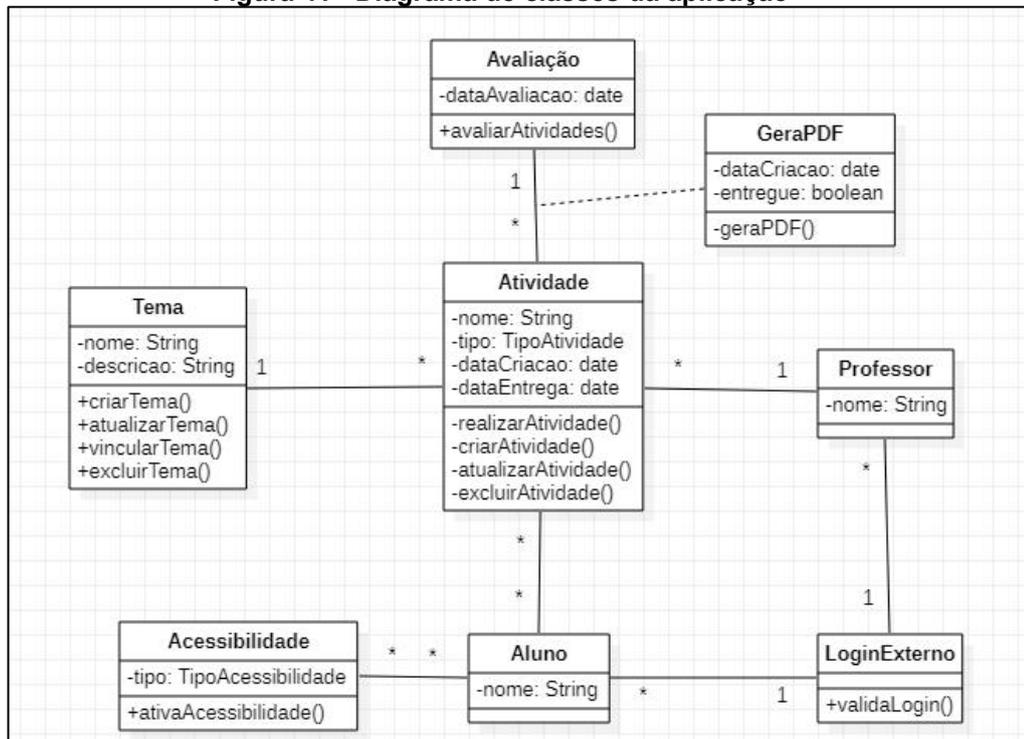
Figura 40 - Diagrama de caso de uso da aplicação



Fonte: Autoria própria (2022).

Por fim, define-se o diagrama de classes da aplicação. Na Figura 41 é possível visualizar o diagrama proposto para a etapa. A classe *GeraPDF* da aplicação foi definida para que o professor possa gerar um PDF da avaliação de uma atividade. A classe *Tema* gerencia os temas das atividades. A classe *Acessibilidade* possibilita a utilização de ferramentas de acessibilidade pelo aluno. A classe *LoginExterno* valida o login com conta externa para o professor e aluno.

Figura 41 - Diagrama de classes da aplicação



Fonte: Autoria própria (2022).

4.4 Implementação da Aplicação

Para a implementação da aplicação foi decidido incluir a funcionalidade de *GerarPDF*, seu método foi incluído na classe *Atividade* apresentada na implementação do domínio e seu código é exibido na Figura 42.

Figura 42 - gerarPDF da classe Atividade

```
gerarPDF(){  
    this.dataCriacao = dataCriacao;  
    this.entregue = entregue;  
}
```

Fonte: Autoria própria (2022).

4.5 Testes da Aplicação

Para validar a aplicação foram realizados testes de aceitação novamente, foi verificado se as funcionalidades de login, funcionamentos de botões e navegação desenvolvidas estavam funcionando corretamente. As interfaces gráficas foram validadas em um aparelho celular para garantir que o *layout* estaria de acordo com o desejado.

4.6 Considerações Finais do Capítulo

Neste capítulo foi apresentada a modelagem do sistema para gerenciamento de atividades utilizando a LPS. Primeiramente foi realizada a análise do domínio para o levantamento de características encontradas nos sistemas analisados. As características levantadas foram importantes para determinar o que faz parte do núcleo, ou são opcionais ou específicos de cada aplicação.

A modelagem foi feita com base neste conjunto de características, por meio da análise do problema, tentando solucioná-lo. O domínio foi projetado e implementado, visando se adequar ao que foi levantado na fase da modelagem. O modelo de domínio proposto é expansível repetindo os passos da LPS para adicionar novas características ao modelo. Ele pode ser incrementado com características de outros sistemas de gerenciamento de atividades, isso é observado pelas etapas da aplicação que acrescentaram mais funcionalidades nos modelos sem que eles ficassem incorretos para a LPS.

Posteriormente, foi realizada a modelagem e implementação da aplicação encontrada para o sistema de gerenciamento para alunos com DI. A implementação

realizada não contemplou todas as *features* pois o objetivo era manter os modelos focados nas atividades. O instanciamento foi realizado para a aplicação-exemplo de uma instituição de ensino específica para alunos com DI, visto que na proposta utilizada, os alunos e nem a instituição possuem um sistema digitalizado para utilizar.

O instanciamento também foi realizado para elementos do *Moodle* (*Login com outras plataformas, Utilizar Opções de Acessibilidade*) e do *Google Classroom* (*Gerenciar Temas de Atividades*). Isso porque eram elementos que se encaixam no modelo atual e poderiam expandir o escopo daquilo que já havia sido proposto na modelagem do domínio do núcleo.

5 CONCLUSÃO

Durante a elaboração do projeto, foram estudadas quatro metodologias para o desenvolvimento de uma LPS. A escolhida foi a de Delazeri e Wolf (2012) por englobar em seu processo um conjunto de aspectos positivos das outras metodologias (FAST, FODA e PLUS).

A LPS permite que a construção de um componente seja reutilizado em outras instâncias de aplicações. Um componente de gerenciamento de login pode ser reutilizado para várias aplicações, já que seu funcionamento é o mesmo, compartilhado entre os domínios.

Na engenharia de domínio foram analisadas duas plataformas existentes, *Moodle* e *Classroom*, e a proposta de um sistema de gerenciamento de atividades escolares para alunos com algum tipo de DI, chamado de Flora. Essas plataformas possuem similaridades e diferenças, tais características foram classificadas como núcleo, opcional e aplicação. Quando uma característica faz parte de todos os produtos, é denominada *núcleo*. Quando uma característica não está presente em um dos produtos, é denominada *opcional*. Finalmente, quando uma característica faz parte apenas de um único produto, é denominada *aplicação*. No levantamento realizado, realizar login foi considerado como *núcleo*, recuperação de credenciais como *opcional* e a geração de PDF de uma avaliação realizada pelo professor foi considerada de *aplicação*.

A modelagem do sistema de gerenciamento de atividades utilizando Linhas de Produto de Software permitiu a construção de um modelo que atende as necessidades básicas de alunos com deficiência intelectual, além de contemplar as funcionalidades de aplicações-exemplo bem conhecidas como o *Google Classroom* e *Moodle*.

A instanciação foi implementada para mobile visando a facilitação do uso do sistema, já que um celular é portátil. Além disso, as opções de acessibilidade nativas do celular podem ser um facilitador para a utilização da ferramenta por alunos com DI.

5.1 Trabalhos Futuros

A partir do trabalho realizado, é possível realizar os seguintes trabalhos:

- Refatoração da linha: o produto final não está otimizado e não possui todas as funcionalidades necessárias, como o gerenciamento de turmas e usuários. Existe espaço para expandir e melhorar o produto final;
- Implementação de outras aplicações-exemplos: existem aplicações que não foram codificadas, pois para este trabalho apenas a implementação da instância para alunos com DI, faltando a implementação das características do *Moodle* e *Google Classroom*;
- Estudo da inclusão de um arquivo de ambiente: é possível estudar a possibilidade de incluir um arquivo de configuração do ambiente para determinar características específicas de sistemas, por exemplo o nome de uma escola, cor dos componentes, funcionalidades presentes, entre outras configurações.
- Testes mais rigorosos: foram realizados apenas testes de aceitação, idealmente seria necessário realizar testes para verificar a integridade do sistema e como ele se comporta em diferentes cenários.

REFERÊNCIAS

- ALMEIDA, T. B; DOS SANTOS, B. C; MATOS, S. N. Análise Qualitativa dos Métodos de LPS para a Modelagem de um Sistema de FAQ. **Anais SULCOMP**, v. 6, 2013.
- ALVES, M. T. G; XAVIER, F. P. Indicadores multidimensionais para avaliação da infraestrutura escolar: o ensino fundamental. **Cadernos de Pesquisa**, v. 48, p. 708-746, 2018.
- AMARAL, M. A; GOMEDI, G. Desenvolvimento de software educacional para crianças Portadoras de síndrome de Down. In: **Anais do IV Congresso Brasileiro de Computação–CBComp, pag.** 2004. p. 238-247.
- BARROS, K. O. Uma linha de produto de software para o turismo em Sergipe. 2022.
- BELL, D. Uml basics: The component diagram. **IBM Global Services**, 2004.
- BEZERRA, G. F; ARAUJO, D. A. C. Atividades linguísticas e cognitivas para alunos com deficiência intelectual: proposta metodológica na inclusão escolar. **Revista Educação Especial**, v. 24, n. 40, p. 233-244, 2011.
- BINDEWALD, C. V. Otimização interativa de projeto de Arquitetura de Linha de Produto de Software. 2020.
- CANTERI, C; DVULATKA, S. D. **Modelagem e implementação de um sistema de venda e de locação de imóveis baseado em linhas de produto.** 2013. Trabalho de Conclusão de Curso. Universidade Tecnológica Federal do Paraná.
- CHEESMAN, J; DANIELS, J. **UML components: a simple process for specifying component-based software.** Addison-Wesley Longman Publishing Co., Inc., 2000.
- CLEMENTS, P; NORTHROP, L. **Software product lines.** Boston: Addison-Wesley, 2002.
- COURTENAY, K; PERERA, B. COVID-19 and people with intellectual disability: impacts of a pandemic. **Irish Journal of Psychological Medicine**, v. 37, n. 3, p. 231-236, 2020.
- COUTINHO, E. F; BEZERRA, C. I. M. Um estudo sobre a variabilidade de aspectos dinâmicos no ecossistema de software educacional solar. In: **Anais do I Workshop em Modelagem e Simulação de Sistemas Intensivos em Software.** SBC, 2019. p. 59-68.
- CURY, C. R. J. et al. O Aluno com Deficiência e a Pandemia. **Instituto Fabris Ferreira**, p. 2020-07, 2020.
- DALMON, D. L; BRANDÃO, L. O. Uma linha de produto de software para módulos de aprendizagem interativa. In: **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE).** 2012.

DE AZEVEDO VIEIRA, E. et al. Contrastes da implementação da infraestrutura tecnológica em 2020 para aulas remotas das escolas públicas brasileiras no período da pandemia da Covid-19. 2021.

DELAZERI, Bruna Rossetto; WOLF, Ellen Cristina. **Modelagem de um sistema organizador baseado em linhas de produto**. 2012. Trabalho de Conclusão de Curso. Universidade Tecnológica Federal do Paraná.

DONEGAN, P. M. **Geração de famílias de produtos de software com arquitetura baseada em componentes**. 2008. Tese de Doutorado. Universidade de São Paulo.

DURSCKI, R. C. et al. Linhas de Produto de Software: riscos e vantagens de sua implantação. **VI Simpósio Internacional de Melhoria de Processos de Software**, p. 155-166, 2004.

FERGUSON, R. Decisions For Sustaining A Software Product Line.[online] Insights. sei. cmu. edu. 2018.

GOOGLE. Flutter. 2015.

GOOGLE. Google Classroom. 2014.

IFTAKHAR, S. Google classroom: what works and how. **Journal of Education and Social Sciences**, v. 3, n. 1, p. 12-18, 2016.

JÚNIOR, S. D. S; COSTA, F. J. Mensuração e escalas de verificação: uma análise comparativa das escalas de Likert e Phrase Completion. **PMKT–Revista Brasileira de Pesquisas de Marketing, Opinião e Mídia**, v. 15, n. 1-16, p. 61, 2014.

KANG, K. C. et al. **Feature-oriented domain analysis (FODA) feasibility study**. Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst, 1990.

KE, X; LIU, J. Deficiência intelectual. **Tratado de Saúde Mental da Infância e Adolescência da IACAPAP (International Association for Child and Adolescent Psychiatry and Allied Professions)**, 2015.

KISHI, T; NODA, N. Formal verification and software product lines. **In: Applied Software Product Line Engineering**. Auerbach Publications, 2009. p. 253-268..

LESSA FILHO, C. A. C; DOMÍNGUEZ, A. H. Uma linha de produto de software para desenvolvimento de jogos educativos. **Revista Brasileira de Informática na Educação**, v. 26, n. 01, p. 1, 2018.

LOBO, A. E. C; RUBIRA, C. M. F. Um Estudo para Implantação de Linha de Produto de Software Baseada em Componentes. **Campinas, Universidade Estadual de Campinas**, 2009.

MARTINS, L. et al. SPLiME: A Tool for Software Product Lines Maintainability Evaluation. **In: Anais Estendidos do X Congresso Brasileiro de Software: Teoria e Prática**. SBC, 2019. p. 114-119.

MOODLE. Modular Object-Oriented Dynamic Learning Environment. 2001

MUXFELDT, P. (2020) Sistemas operacionais para celulares e dispositivos móveis. Disponível em: < <https://br.ccm.net/faq/11106-sistemas-operacionais-para-celulares-e-dispositivos-moveis>>. Acesso em: 25 de Out. 2021.

DE OLIVEIRA, A. A. S. Aprendizagem escolar e Deficiência Intelectual: a questão da avaliação curricular. **EDUCACAO ESPECIAL**, 2011.

OLIVEIRA, B. H. M; NUNES, V. R. **Cognita: objeto virtual de aprendizagem para auxílio do ensino e avaliação da informática à pessoas com deficiência intelectual**. 2019. Trabalho de Conclusão de Curso. Universidade Tecnológica Federal do Paraná.

OLIVEIRA, R. P. UbiComSPL: desenvolvimento baseado em MDA, de linha de produto de software no domínio de aplicações ubíquas. 2009.

PLETSCH, M. D; GLAT, R. A escolarização de alunos com deficiência intelectual: uma análise da aplicação do Plano de Desenvolvimento Educacional Individualizado. **Linhas Críticas**, v. 18, n. 35, p. 193-208, 2012.

PUTKINEN, V. et al. Musical activities and the development of neural sound discrimination. 2014.

RIOT. Teamfight Tactics. 2019.

ROSSI, C. M. S; AMORIM, D. B; SANTOS, S. A Utilização do Moodle no Ensino Da Matemática Para Alunos Com Deficiência. **REAMEC-Rede Amazônica de Educação em Ciências e Matemática**, v. 8, n. 2, p. 88-101, 2020.

SABBATINI, R. ME. Ambiente de ensino e aprendizagem via Internet: a Plataforma Moodle. **Instituto EduMed**, v. 7, 2007.

SANTANA, E. F. Z. et al. Modelagem Específica de Domínio em Linhas de Produto de Software na computação ubíqua. In: **SBCARS**. 2009. p. 209-222.

SALDANHA, H. L. **Desenvolvimento de um sistema de gestão bibliotecária baseado em linha de produto de software**. 2019. Trabalho de Conclusão de Curso. Universidade Tecnológica Federal do Paraná.

SILVA, F. A. P; NETO, P. A. M. S; GARCIA, V. C; MUNIZ, P. F. Linhas de Produto de Software: Uma tendência da indústria In: SANTANA, A. M.; NETO, P. de A. dos S.; MOURA, R. S. V Escola Regional de Informática Ceará, Maranhão, Piauí: Livro texto dos minicursos. Teresina: Sociedade Brasileira de Computação, p. 7-31. 2011.

SINGH, J; SINGH, J. COVID-19 and its impact on society. **Electronic Research Journal of Social Sciences and Humanities**, v. 2, 2020.

SOUZA, K. R. et al. Trabalho remoto, saúde docente e greve virtual em cenário de pandemia. **Trabalho, Educação e Saúde**, v. 19, 2020.

THOMAS, D. **Do not go gentle into that good night**. 1947.

VALENTE, J. A. A sala de aula invertida e a possibilidade do ensino personalizado: uma experiência com a graduação em midialogia. **Metodologias ativas para uma**

educação inovadora: uma abordagem teórico-prática. Porto Alegre: Penso, p. 26-44, 2018.

WEISS, D. M; LAI, C. T. R. Software Product-Line Engineering: A Family Based Software Development Process. 1. ed. Boston: Addison-Wesley, 1999.

ZAINA, L. et al. Uma linha de produto de software para construção de museus virtuais para aprendizagem. In: **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)**. 2015. p. 51.