

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
GOIÁS - CÂMPUS JATAÍ
DEPARTAMENTO DE ÁREAS ACADÊMICAS
CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS**

**Muriel Ferreira Vieira
20151020230249**

***Aplicação Mobile para consumo de material didático digital estruturado em IMS
Learning Design***

Jataí
2018

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
GOIÁS - CÂMPUS JATAÍ
DEPARTAMENTO DE ÁREAS ACADÊMICAS
CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS**

Muriel Ferreira Vieira

RELATÓRIO TÉCNICO

***Aplicação Mobile para consumo de material didático digital estruturado em IMS
Learning Design***

Relatório apresentado como requisito para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas pelo Instituto Federal de Educação, Ciência e Tecnologia de Goiás.

Orientador: Dr. Aladir Ferreira da Silva Júnior.
Co-Orientador: Me. Leizer Fernandes Moraes

Jataí
2018

Dados Internacionais de Catalogação na Publicação na (CIP)

VIE/apl	Vieira, Muriel Ferreira. Aplicação <i>Mobile</i> para consumo de material digital estruturado em IMS <i>Learning Design</i>/ Muriel Ferreira Vieira. -- Jataí: IFG, Coordenação dos cursos de Informática – Tecnologia em Análise e Desenvolvimento de Sistemas, 2018. 43 f. Orientador: Prof. Dr. Aladir Ferreira da Silva Júnior. Co-orientador: Prof. Me. Leizer Fernandes Moraes. Bibliografias. 1. Software. 2. <i>Learning Design</i>. 3. Educação. 4. IMS-LD. 5. <i>Mobile</i>. I. Silva Júnior, Aladir Ferreira da. II. Moraes, Leizer Fernandes. III. IFG, Câmpus Jataí. IV. Título. CDD 005.133
---------	---

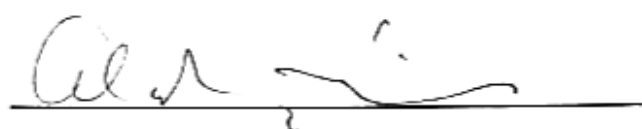
Muriel Ferreira Vieira

**Aplicação Mobile para consumo de material didático digital
estruturado em IMS Learning Design.**

Este Relatório Técnico foi julgado adequado para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas e aprovado em sua forma final pela Banca Examinadora.

Trabalho aprovado em 08 de agosto de 2018.

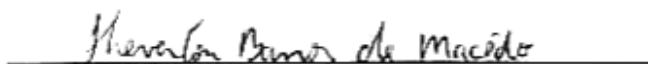
BANCA EXAMINADORA



Prof. Dr. Aladir Ferreira da Silva Júnior
(Presidente da Banca/Orientador)
Instituto Federal de Educação, Ciência e Tecnologia de Goiás



Prof. Me. Eliane Raimann
(Membro Interno)
Instituto Federal de Educação, Ciência e Tecnologia de Goiás



Prof. Dr. Heverton Barros de Macedo
(Membro Interno)
Instituto Federal de Educação, Ciência e Tecnologia de Goiás

Jataí, agosto de 2018.

AGRADECIMENTOS

Agradeço primeiramente ao G·A·D·U· por sempre me iluminar, e me guiar nas decisões e desafios mais tenebrosos.

Aos meus pais Maria Aparecida Ferreira Vieira e Ariel Oliveira Vieira que não pouparam esforços para que esse sonho fosse realizado. Dedico a vocês esse trabalho. A minha luta sempre foi a de vocês. A minha vitória, será eternamente nossa.

Dedico também à minha colega de infância, colega de curso e parceira na vida, Victoria Ferreira Souza, que sempre me acompanhou, apoiou e esteve comigo nos dias bons e ruins.

Agradeço ao meu colega de curso Douglas Alves de Campos por me ceder a conta paga da *playstore*, possibilitando a publicação da aplicação.

Regracio ao Instituto Federal de Goiás, por me proporcionar educação de qualidade e gratuita, desde do meu ensino médio, através do curso de Técnico de Informática, até o meu ensino superior, com o presente curso de Tecnologia em Análise e Desenvolvimento de Sistemas, totalizando quase 8 anos de relacionamento, oferecendo aporte e incentivo para que sempre pudesse prosseguir em meus estudos.

Finalmente, estendo meus agradecimentos a todos os professores do núcleo docente estruturante do curso, em especial aos professores Dr. Aladir Ferreira da Silva Júnior e Me. Leizer Fernandes Moraes que se dedicaram na orientação desse projeto, sempre me incentivando com orientações profissionais e pessoais, que levarei por toda a vida.

Todos vocês foram fundamentais para a realização desse momento.

Muito Obrigado.

RESUMO

Há uma dificuldade considerável em se produzir e utilizar materiais didáticos digitais (MDD) alinhados a técnicas pedagógicas reconhecidas, como WebQuest, JigSaw, entre outras, levando em consideração o uso de padrões de estruturação de conteúdo, tais como o IMS *Learning Design* (IMS-LD), ADL SCORM (SCORM) entre outros. Para produzir MDD se tem atualmente os editores que minimizam as dificuldades do professor ao estruturar seu material didático seguindo um padrão, no entanto se faz necessária a utilização desse material pelos alunos. Identificou-se que há poucas opções de softwares executores do material produzido no padrão SCORM e de modo mais acentuado há uma escassez de executores de material didático produzido no padrão IMSLD. Nesta pesquisa ainda se verificou a inexistência de um executor IMS-LD voltado a plataformas móveis e essa foi a motivação para o desenvolvimento dessa pesquisa aplicada. A partir da metodologia ágil de desenvolvimento de software chamada *Mobile Application Software Development Based on Agile Methodology* (MASAM), desenvolveu-se um aplicativo para plataforma móvel que executa MDDs estruturados em IMS-LD, objetivo principal dessa pesquisa. Neste relatório técnico se apresenta uma descrição dos passos realizados e consolida a pesquisa realizada mostrando o aplicativo desenvolvido.

Palavras-chave: *Software; Educação; Learning Design; IMS-LD; Mobile.*

LISTA DE ABREVIATURAS E SIGLAS

API - *Application Programming Interface*
EAA - Executores de Atividades de Aprendizagem
EML - *Educational Modelling Language*
HME - *Hybrid Method Engineering*
IMS-LD – *IMS Learning Design*
LDP - *Learning Design Player*
MASAM - *Mobile application agile development process*
MDD – *Materiais Didáticos Digitais*
OUNL - *Open University of the Netherlands*
SCORM - *Sharable Content Object Reference Model*
SDK - *Software Development Kit*
TDICs - *Tecnologias Digitais de Informação e Comunicação*
UoLs - *Units of Learning*
USB – *Universal Serial Bus*
RUP - *Rational Unified Process*
XP - *Extreme Programming*
LTSC - *Learning Technology Standards Committee*

SUMÁRIO

1. INTRODUÇÃO	5
2. DESENVOLVIMENTO	10
2.1. METODOLOGIA DE DESENVOLVIMENTO	10
2.2. DESENVOLVIMENTO COM A METODOLOGIA MASAM (MOBILE APPLICATION SOFTWARE DEVELOPMENT BASED ON AGILE METHODOLOGY).....	12
2.2.1. PREPARAÇÃO	13
2.2.2. PERSONALIZAÇÃO.....	15
2.2.3. DESENVOLVIMENTO	16
2.2.3.1. CRIAÇÃO E CONFIGURAÇÃO DE UM PROJETO REACT NATIVE	18
2.2.3.2. INSTALANDO O PROJETO MOBILE LD-PLAYER.....	20
2.2.3.3. COMERCIALIZAÇÃO	32
3. CONSIDERAÇÕES FINAIS	34
REFERÊNCIAS.....	35

1. INTRODUÇÃO

As inovações tecnológicas e seus respectivos paradigmas desencadeiam a premência de se incorporar o uso de ferramentas tecnológicas no processo de formação acadêmica e profissional do cidadão.

No entanto, essa onda de inovação tecnológica aparenta ainda não ter atingido de modo considerável o ambiente escolar, e por consequência, o pouco uso de novas tecnologias acaba atingindo o aluno. Se por um lado, o aluno moderno possui um perfil nativo digital, possuindo facilidade para lidar com tecnologia (RODRIGUES JUNIOR, 2014), pelo outro, há falta de domínio no uso das Tecnologias Digitais de Informação e Comunicação (TDICs) por considerável parte dos professores (FISCARELLI, 2007; VALENTE e ALMEIDA, 1997).

Muitos professores tentam atenuar a insuficiência de domínio tecnológico, mas não sabem como fazê-lo, não sentindo segurança em experimentar novas tecnologias (MORAN; MASETTO; BEHRENS, 2006), com isso, reflete-se na produção, utilização e divulgação dos materiais digitais produzidos. Atualmente, já existem *softwares* que auxiliam na produção de material digital seguindo padrões de estruturação de conteúdo, porém não basta apenas produzir tais materiais didáticos, é necessário que o público-alvo tenha meios para utilizar o material produzido.

Ao se utilizar de tecnologias que promovam a aproximação entre a produção da informação e o destinatário final, é notório que o acesso ao conhecimento se torna mais fácil (RODRIGUES JUNIOR, 2014). Para execução e consumo do conteúdo gerado pelos editores de Materiais Didáticos Digitais (MDD), tais como: EduBox, WQE editor, ReCourse se faz necessário o uso de Executores de Atividades de Aprendizagem (EAA) (ASSIS; ALMEIDA, 2017).

Executores e Produtores de MDD podem utilizar um padrão de estruturação de arquivos. O uso desses padrões, gera benefícios aos seus usuários, podendo destacar a forma padronizada de estruturação e empacotamento de conteúdo, facilitando a disseminação de dados e conteúdos, pois ao criar o conteúdo seguindo um padrão, já existirá diversos leitores que poderão interpretar. Para Dutra e Tarouco (2006), a adoção de padrões e modelos para o desenvolvimento e Objetos de Aprendizagem é importante, visto que o avanço da tecnologia acaba levando à substituição de Ambientes Informatizados de Aprendizagem. Sendo assim, o uso de Objetos de Aprendizagem adequado à um modelo ou padrão, prolonga e expande o uso deste. O *IMS Learning Design*, aludido

aqui, como IMS-LD, é um padrão que contém uma metalinguagem para construção de unidades de aprendizagem (Units of Learning – UoLs) que não depende de modelos pedagógicos (LD, 2003).

A Figura 1, mostra uma da estrutura de aprendizagem do IMS-LD, e a instanciação concreta das atividades na estrutura.

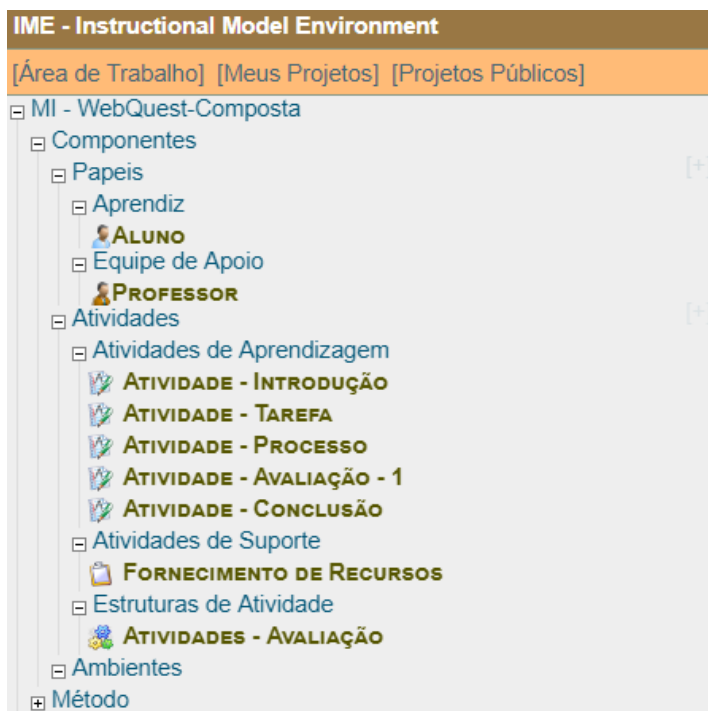


Figura 1: Exemplo de Estruturação de Arquivos IMS-LD

Fonte: (JÚNIOR; SILVA; FERNANDES, 2005).

O IMS-LD surgiu em 1998, na *Open University of the Netherlands* (OUNL – Universidade Aberta da Holanda) em um projeto de pesquisa criado com o objetivo de construir uma notação semântica para representar unidades de estudo a serem utilizadas em e-Learning (Koper, 2001).

O projeto de pesquisa inicialmente concebeu o *Educational Modelling Language* (EML), um conjunto de elementos que podem ser utilizados para descrever unidades de estudo. O EML foi enviado para análise do *IMS Global Learning Consortium, Inc.*, instituição provedora de padrões abertos e livres. O EML sendo amplamente discutido, retrabalhado e declarado um padrão oficial do IMS em 2003.

A conjectura que abrange a especificação IMS-LD aponta que convicções e fundamentos que norteiam práticas educacionais, podem ser representados por meio de um *design*, que, por sua vez, pode ser ilustrado por intermédio de elementos de representação. Além disso, o *design* de uma unidade de estudo é guiado por modelos pedagógicos, que podem ser definidos como um conjunto

de regras que prescrevem como aprendizes podem atingir objetivos educacionais, em determinado contexto ou domínio de conhecimento, da forma mais eficaz (KOPER; OLIVER, 2004).

Destarte, o IMS-LD é aplicado para retratar o processo de ensino-aprendizagem, definindo sob que circunstâncias determinadas atividades devem ser realizadas por alunos e professores com a finalidade de que os objetivos educacionais estipulados sejam atingidos (KOPER; OLIVER, 2004).

Em busca de literatura acerca de trabalhos correlatos, verificou-se que não existem muitas opções de Executores de Atividades de Aprendizagem (EAA), disponíveis para discentes e docentes. Entretanto, dentre os poucos disponíveis, destacam-se o *ScormPlayer*: Leitor de Atividades de Aprendizagem do Padrão SCORM, disponível para a plataforma *Mobile* Android e o *Reload Learning Design Player*: EAA do padrão IMS *Learning Design* disponível para *Desktops*.

O *Reload - Learning Design Player* (LDP), foi desenvolvido pela *University of Bolton*. Esse EAA permite uma fácil importação de pacotes *Learning Design* sem o uso de linhas de comando. Através de uma inteligível interface gráfica, o usuário seleciona o arquivo no referido padrão, e o *software* automaticamente o lê e interpreta, apresentando visualmente seu conteúdo. O *Reload* LDP utiliza o motor de tempo de execução *CopperCore* para fornecer essa interface de usuário gráfica simples (MARTENS; VOGTEN, 2005).

O *ScormPlayer* é um *software* desenvolvido para plataformas *mobile*, pela empresa húngara InfoTec Kft. Esse aplicativo reproduz conteúdo estruturado no padrão internacional *Sharable Content Object Reference Model* (SCORM). O SCORM é um modelo de referência ou conjunto de especificações destinados a disponibilização de conteúdos e serviços *e-learning*. Esse conjunto de especificações determina um modelo de agregação de conteúdo e um ambiente de execução de objetos de aprendizagem baseados na *Web*. (ADL, 2004).

Segundo Dutra e Tarouco (2006), mesmo que tenham os Objetos de Aprendizagem como seu foco principal, o SCORM e o IMS-LD se diferem bastante em sua estrutura. Enquanto o SCORM preocupa-se essencialmente com o conteúdo, de que forma é estruturado e sequenciado, na maneira de expor e rastrear as ações do usuário nas interações com o conteúdo, o IMS-LD ressalta que existem diversas relações no processo de ensino de aprendizagem, tais como as relações: aluno e conteúdo, aluno e colegas, aluno e professor e aluno e recursos de aprendizagem. As duas especificações possuem características similares, com ambas tendo consórcios que produzem e mantem os referidos padrões, como o IMS *Global Consortium* e IEEE *Learning Technology Standards Committee* (LTSC).

Tanto o IMS-LD, quanto o SCORM utilizam a linguagem XML para múltiplas finalidades, como proceder com o empacotamento das unidades de aprendizagem e uso dos metadados. Porém, mesmo com as semelhanças, o SCORM segue uma linha behaviorista, com conteúdos pré-definidos, impossibilitando o uso de outras abordagens educacionais, já o IMS-LD fornece suporte ao uso de diversas abordagens educacionais, como: behavioristas, cognitivistas e construtivistas.

O IMS-LD fornece diversas possibilidades de uso, mas como implementar efetivamente essa especificação? Para responder essa questão, é importante entender o ciclo de vida da especificação. A Figura 2 exibe o sequenciamento do ciclo de vida, os produtos de cada etapa, e os tipos de ferramentas necessárias (PAQUETTE et al., 2005).

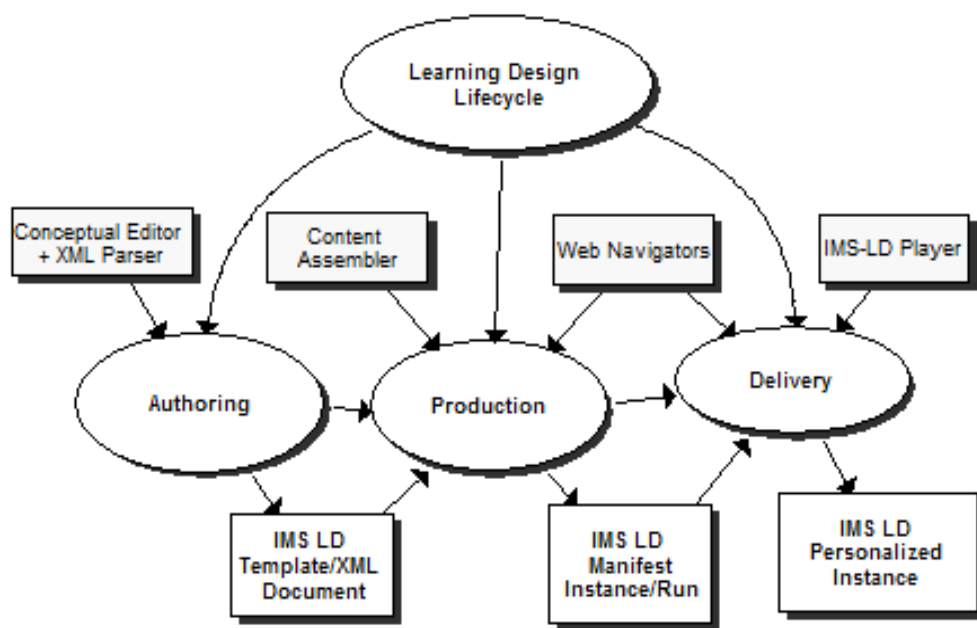


Figura 2: Ciclo de Vida do *Learning Design*

Fonte: PAQUETTE et al. (2005).

No ciclo de vida do *Learning Design*, a autoria é a primeira etapa, que corresponde a criação do MDD. Aqui o professor, ou usuário, estrutura o seu conteúdo, e tem como produto desse processo de autoria um documento XML, compatível com o padrão IMS-LD.

Com o conteúdo estruturado, parte-se para etapa de produção, onde é feita a instanciação do ambiente de aprendizagem. Para isso, é necessário definir a descrição, tempo de início e término da atividade, escolha dos usuários (alunos) para qual a atividade é destinada.

A etapa da entrega é a execução do MDD produzido. Nessa etapa é necessária uma plataforma, um EAA, que reproduza, interprete e entregue ao usuário, todo o conteúdo que foi estruturado anteriormente.

O IMS-LD comporta três níveis de representação: A, B e C. Boaretto, Nunes e Filatro (2006) detalham a representação dos níveis do IMS-LD.

No Nível A é possível representar um sequenciamento simples, pré-determinado, no qual as atividades são apresentadas da mesma maneira para todos os usuários atribuídos a um papel. O Nível B permite maior personalização da aprendizagem e um sequenciamento mais complexo das atividades pela adição de propriedades e condições, através das quais se pode definir, por exemplo, que determinada atividade deve ser feita por determinado papel somente se determinada condição for satisfeita. Além disso, as propriedades permitem armazenar dados de cada usuário (aluno ou professor), facilitando o rastreamento e a análise processual da aprendizagem. O Nível C agrega um recurso de envio de notificações, adicionando uma flexibilidade ainda maior com relação à personalização da aprendizagem.

Verificou-se então a inexistência de um EAA desenvolvido para dispositivos móveis, que trabalhe com o padrão IMS-LD. Deste modo, justifica-se a viabilidade e necessidade de desenvolvimento de uma ferramenta de leitura e consumo do material didático produzido, que siga os padrões de técnicas educacionais consolidadas, destinado a dispositivos móveis embutidos com o sistema operacional *Android*. A ferramenta interpreta o arquivo IMS-LD, estruturado em nível A, e disponibiliza ao usuário, em uma interface amigável, todo o conteúdo inserido no arquivo.

Este leitor poderá ser de grande valia para professores, instrutores e outros responsáveis pela docência em sala de aula, presencial ou não, assim como, para discentes, que poderão usufruir do material didático digital produzido pelo docente, mesmo fora da sala de aula. Espera-se que o EAA desenvolvido também contribua para a disseminação do padrão Learning Design, assim, expandindo o seu uso e aumentando a literatura para o padrão.

Neste relatório se apresenta a descrição do processo de desenvolvimento do software executor de material didático estruturado em IMS-LD, o que será feito em detalhes a partir da próxima seção.

2. DESENVOLVIMENTO

A aplicação, cujo desenvolvimento é descrito neste relatório técnico é chamado de Mobile LD-PLayer. Trata-se de uma plataforma *mobile*, desenvolvida para o sistema operacional Android, que possibilita a leitura e consumo de materiais desenvolvidos no padrão IMS-LD, advindos de qualquer Sistema Operacional, plataformas *Web*, entre outros, desde que seguindo o referido padrão e que esteja no nível A da especificação. O usuário deve selecionar um arquivo local no padrão IMS-LD e a partir desta importação, a aplicação interpretará o arquivo e apresentará o conteúdo na tela. Com o conteúdo na tela, o usuário pode navegar em todo material ali presente, da maneira como foi organizada pelo criador do conteúdo.

Durante o desenvolvimento da aplicação foram utilizados métodos e procedimentos descritos a seguir.

2.1. METODOLOGIA DE DESENVOLVIMENTO

Recentemente, no ano de 2017, o Sistema Operacional *Mobile* Android tornou-se o mais utilizado no mundo dentre todas plataformas computacionais operadas, ultrapassando o *Microsoft Windows* (SIMPSON, 2017).

Em consequência desta grande demanda de usuários e devido ao avanço dos smartphones, que se tornaram as tecnologias mais usadas para acessar a internet, o desenvolvimento de aplicativos móveis tem sido muito tencionado, exigindo soluções computacionais cada vez mais rápidas, otimizadas e com o menor custo possível. Nessa conjectura, a adoção de metodologias ágeis de desenvolvimento torna-se fortemente recomendada.

O desenvolvimento de *software* destinado para plataformas móveis possui características que tornam o processo e sua execução discernida em relação ao desenvolvimento de aplicações *web* e *desktop*, modalidades estas, tidas como tradicionais (RAHIMIAN; RAMSIN, 2008). As divergências que podem ser destacadas estão no ambiente de desenvolvimento e nos procedimentos técnicos envolvidos no processo, uma vez que os sistemas operacionais dos dispositivos móveis dispõem de variadas versões, capacidades de *hardware* e tamanho de tela (HAMMERSHOJ, 2010). Juntamente com as limitações técnicas citadas, as aplicações móveis fazem parte de um modelo de negócio particular, destoando dos modelos aplicados em *softwares* tradicionais.

Dentre as metodologias ágeis mais aceitas e utilizadas, no âmbito dos dispositivos móveis, destaca-se a *Mobile-D*. Foi proposta por Abrahamsson et al. (2004), e propõe a separação das fases de desenvolvimento de um projeto em cinco iterações: *Setup*, *Core*, *Core2*, Estabilizar e Empacotar. Essa metodologia é alicerçada no ciclo de vida do *Rational Unified Process* (RUP), nas práticas de desenvolvimento do *Extreme Programming* (XP) e da escalabilidade da metodologia *CRYSTAL*. Todavia, apesar da *Mobile-D* parecer auspiciosa, sua documentação é rasa e incompleta (RAHIMIAN; RAMSIN, 2008).

Também há propostas de métodos ágeis móveis que se contrapõem ou buscam complementar e melhorar a *Mobile-D*, como as metodologias: *Hybrid Method Engineering* (HME) e *Mobile Application Software Development Based on Agile Methodology* (MASAM). A primeira propositura foi concebida por Rahimian e Ramsin (RAHIMIAN; RAMSIN, 2008), que engloba numerosas fases que vão desde o processo de concepção até a comercialização do produto, tornando-se bem detalhada e apinhada, podendo até ser de utilização complexa, pois apesar dos autores dessa metodologia criticarem a *Mobile-D* por possuir uma documentação superficial, a HME sofre do mesmo problema, tendo o agravante de não apresentar nenhum estudo de caso.

Jeong et al. (JEONG et al., 2008) apresentam a abordagem ágil MASAM, que é alicerçada em arquiteturas e componentes reutilizáveis, propondo um ciclo de desenvolvimento composto de quatro fases: Preparação, Personalização, Desenvolvimento e Comercialização.

Essa metodologia, assim como a *Mobile-D*, é baseada no método de desenvolvimento do XP e no ciclo de vida do RUP. O XP tem como principal característica as entregas frequentes em pequenos ciclos de desenvolvimento, cujo objetivo é melhorar a qualidade do software e a capacidade de resposta à evolução das necessidades do cliente. No ciclo de vida do RUP, são definidas práticas a serem aplicadas pela equipe de desenvolvimento, customizadas de acordo com o cenário do projeto.

Apesar da MASAM apresentar uma forte similaridade com a *Mobile-D*, há variações de conteúdo, como gerenciamento de projetos e ferramentas de acompanhamento (FLORA; CHANDE, 2013).

Ao utilizar dentro do processo de desenvolvimento, etapas de metodologias bem consolidadas no âmbito do desenvolvimento de software, como XP e RUP, possuindo um ciclo de vida inteligível e etapas bem definidas e intuitivas, mesmo com o pesar de não possuir estudos de caso, para facilitar a compreensão da aplicabilidade da metodologia, e finalmente, analisando os

pontos positivos e negativos das principais metodologias de desenvolvimento para plataformas móveis, optou-se neste projeto, utilizar a abordagem ágil MASAM.

2.2. Desenvolvimento com a metodologia MASAM (*Mobile Application Software Development Based On Agile Methodology*).

A tabela abaixo demonstra discriminadamente a distribuição das atividades de desenvolvimento, descrevendo as tarefas e atividades de cada fase da implementação, que foram seguidas para construção da aplicação *Mobile LD-Player*.

Tabela 01: Fases do ciclo da Metodologia MASAM.

Fase	Atividade	Tarefa
Preparação	Compreensão do produto	Sumário do produto e pré-planejamento
	Compartilhamento do conceito do produto	Definição do usuário e análise inicial do produto
	Setup do projeto	Setup (configuração) dos processos de desenvolvimento e dos recursos do projeto, realização de estudo inicial
Personalização	Entendimento das necessidades do usuário	Story card workshop e design da interface de usuário
	Definição e elaboração da arquitetura	Análise dos requisitos não funcionais, definição da arquitetura e identificação dos padrões de software a serem utilizados
Desenvolvimento	Preparação e Implementação	Setup do ambiente de desenvolvimento e planejamento do desenvolvimento
	Ciclos de lançamentos	Planejamento dos lançamentos, dos ciclos ou iterações e a realização propriamente dita desses lançamentos
Comercialização	Testes do sistema	Testes de aceitação e testes pelo usuário
	Comercialização do produto	Testes de lançamento e lançamento do produto

Fonte: (JEONG et al., 2008).

A partir desse ponto, é descrito o funcionamento da metodologia, compreendendo suas atividades e tarefas, e detalhando como cada ponto da metodologia foi usado durante todo o processo de desenvolvimento do produto.

2.2.1. Preparação



Figura 3: Fluxograma da fase de Preparação

Fonte: Autoria Própria

O ciclo de desenvolvimento inicia-se com a fase de preparação (Figura 3), partindo para a compreensão do produto, obtendo um resumo sobre o mesmo. Logo após, com o alicerce do entendimento inicial, é elaborado um pré-planejamento do projeto.

Para a compreensão inicial dos conceitos do *Mobile LD-Player*, foi feito um levantamento de informações (Quadro 01), visando responder questionamentos iniciais, como: “O que a aplicação irá fazer? Qual o público alvo da aplicação? Qual o escopo do projeto?”.

Quadro 01: Sumário do Produto, *Mobile LDPlayer*

Sumário do Produto

O *Mobile LD-Player* é uma plataforma *mobile*, desenvolvida para o sistema operacional *Mobile Android*. Essa plataforma, como principal funcionalidade, possibilitará o consumo de materiais desenvolvidos no padrão *IMS Learning Design*. Os arquivos lidos pelo Leitor poderão ser advindos de qualquer Sistema Operacional, plataforma web, desde que venha no padrão estipulado. [...]

Fonte: Autoria Própria.

Com o levantamento introdutório concluído, partiu-se para a tarefa de pré-planejamento e compartilhamento do conceito do produto, onde os conceitos da aplicação são compartilhados com os envolvidos no projeto, para uma análise mais aprofundada destes. Como neste projeto, há somente um envolvido, nesta etapa foi feito um estudo dos conceitos já levantados, fortificando o conhecimento sobre o projeto até aqui adquirido, destacando-se aqui, os estudos sobre o padrão IMS-LD. Destarte, acarreta-se o processo de definição de usuários (Quadro 02).

Quadro 02: Definição de Usuários

Definição de usuários

Os usuários principais da aplicação *Mobile LD-Player* constituem-se de professores e alunos, com perfis abrangentes.

Alunos: De qualquer nível educacional, sem restrições de faixa-etária, desde que tenha capacidade de utilizar um *smartphone*, podendo pertencer a esfera pública ou privada de ensino. Sendo o principal usuário do *software*, este terá a função de consumir os materiais didáticos digitais disponibilizados pelo seu professor, ou disponibilizado por qualquer repositório de material didático digital do formato IMS-LD. [..]

Fonte: Autoria Própria.

A fase de preparação finaliza com o *setup* do projeto, que incorpora o planejamento dos recursos necessários, tais como o estabelecimento de ambiente, disponibilidade de computadores e ferramentas de desenvolvimento, concluindo com uma análise inicial do produto, levando em conta os artefatos e os conhecimentos já adquiridos do projeto. Para cumprir esta etapa, foi definido o planejamento de recursos necessários para o desenvolvimento do *Mobile LD-Player* (Quadro 03), definindo o editor de código fonte, configuração do computador necessária e o *smartphone* para testes.

Quadro 03: Definição de Recursos e Ferramentas.

Definição de Recursos e Ferramentas

Computador (para o desenvolvimento da aplicação): Notebook ASUS PU401LA – Processador Intel Core i5 4200U, 6GB RAM, 500HD

Editor de Código: Microsoft Studio Code.

Smartphone usado para testes da aplicação: ASUS ZenFone 3 Zoom.

Fonte: Autoria Própria.

2.2.2. Personalização



Figura 4: Fluxograma da fase de Personalização

Fonte: Autoria Própria

A fase de personalização compreende as atividades de entendimento das necessidades do usuário, sugerido por meio de *workshops* de requisitos e com a construção de protótipos de interface e da arquitetura da aplicação.

Analizando a conjectura dos artefatos, dados e estudos levantados na fase e nas atividades anteriores, com o escopo do projeto e público alvo definidos, esta atividade foi focada no entendimento das necessidades do uso de alunos e professores, que são os potenciais usuários da aplicação *Mobile LD-Player*.

Por ser um projeto enxuto, com um desenvolvedor, tendo informações e requisitos anteriormente fundamentados, não se aplicou a necessidade em fazer um *workshop* de requisitos, partindo então, para a criação da interface e arquitetura.

A interface foi feita usando o *framework React Native*, que contém os componentes gráficos (Exemplos: *Button*, *ListView*, *CheckBox*) nativos do sistema operacional Android, facilitando assim, desenvolver a aplicação dentro dos padrões e convenções deste sistema operacional.

A aplicação, contará principalmente com alunos e professores de diversas faixas-etárias, com isso, é necessário que a interface seja intuitiva e uniforme, para facilitar o uso por um público heterogêneo. Inicialmente foi estabelecido o esquema de cores da aplicação, definindo a paleta de cores: Azul Marinho e Branco.

Com as prototipações das interfaces, caminhou-se para a estruturação da arquitetura da aplicação, com o intuito de realizar as junções das interfaces, definindo a configuração de navegabilidade e estruturação lógica e sequencial das telas (Figura 5).

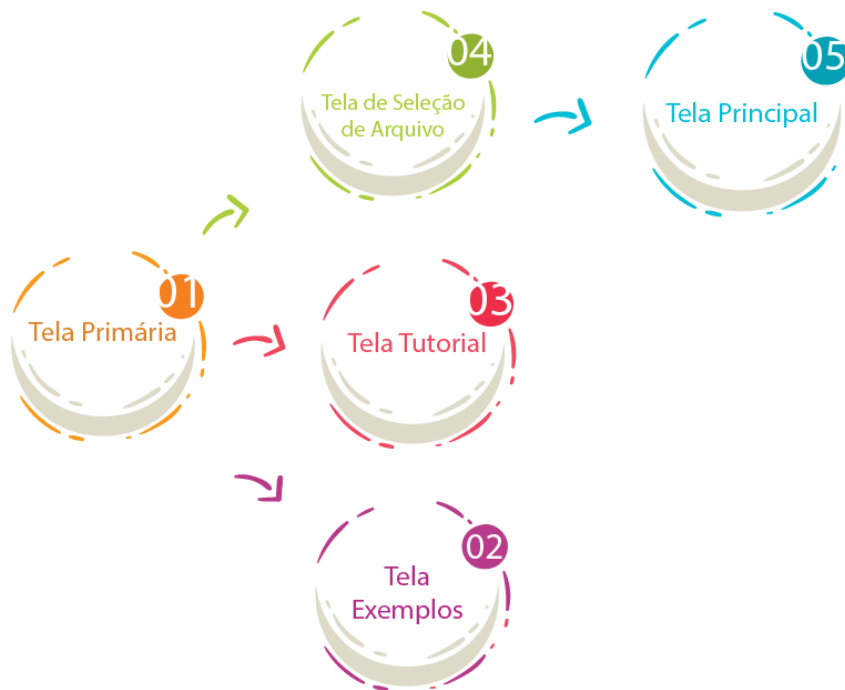


Figura 5: Fluxograma das telas

Fonte: Autoria Própria

2.2.3. Desenvolvimento

Com a arquitetura feita, foi iniciada a fase de desenvolvimento, com a preparação do ambiente de desenvolvimento. O editor de código fonte, que servirá como principal ambiente de codificação da aplicação é o Microsoft *Visual Studio Code*. O *framework* utilizado para permitir o desenvolvimento da aplicação é o *React Native*. O *React Native* é um *framework* baseado no *React*, desenvolvido pela equipe do Facebook, que possibilita o desenvolvimento de aplicações *mobile* nativas para Android e iOS, com o uso da já conhecida linguagem JavaScript (WINTERFELDT, 2017).

React, que é uma biblioteca JavaScript que é baseada no desenvolvimento de interfaces JSX, uma extensão JavaScript que viabiliza escrever sintaxe XML, possibilitando codificar a lógica de programação juntamente com a interface.

As aplicações *React Native* são como aplicações *React* em todo o seu funcionamento, com exceção em que no *React Native* são invocadas a *Application Programming Interface* (APIs) da linguagem nativa do dispositivo, renderizando os componentes, de acordo com o dispositivo em que está instalado. Tudo isso é possível a partir da conexão que o *React Native* estabelece com a API dos elementos nativos. (FACEBOOK, 2018).

Primeiramente, é necessário fazer o *download* e instalação de um editor de código, como, por exemplo, o *Visual Studio Code*, utilizado nesse projeto. Parte-se então, para a instalação do pacote Node JS, um interpretador de código JavaScript, necessário para o desenvolvimento utilizando *React Native*. O Node JS pode ser encontrado no site: <https://nodejs.org>, ao acessar o endereço, é necessário realizar o *download* da versão estável do pacote. Para o desenvolvimento do projeto descrito nesse relatório técnico, foi usada a versão 8.11.3 LTS do pacote Node JS.

Em seguida é necessário instalar o git, um sistema de controle de versão de arquivos, que possibilitou durante o desenvolvimento desse projeto, um maior controle e organização da criação das funcionalidades, e principalmente incrementou uma maior segurança no projeto, oferecendo integridade e *backups* constantes do código fonte. Para o *download*, acesse: <https://git-scm.com/downloads>.

Para o correto funcionamento do *React-Native*, é necessário baixar, instalar e configurar o *Software Development Kit* (SDK), já que o mesmo, utiliza os recursos nativos do *Android*, necessitando do conjunto de ferramentas de desenvolvimento padrão da plataforma. O SDK pode ser encontrado no site <https://developer.android.com/studio/>. Dentro da página, é possível instalar o SDK separadamente, ou baixar em conjunto com o *Android Studio*.

Após a instalação do SDK, é importante configurar a variável de ambiente ANDROID_HOME. No Painel de Controle do Windows, abrir “Sistema e Segurança” ou “Sistema”, clicar em “Configurações avançadas do sistema”, selecionar “Variáveis de ambiente” e clique no botão “Nova variável de ambiente”, insira o nome da variável como ANDROID_HOME, coloque o caminho do local onde foi instalado o SDK (Figura 06).

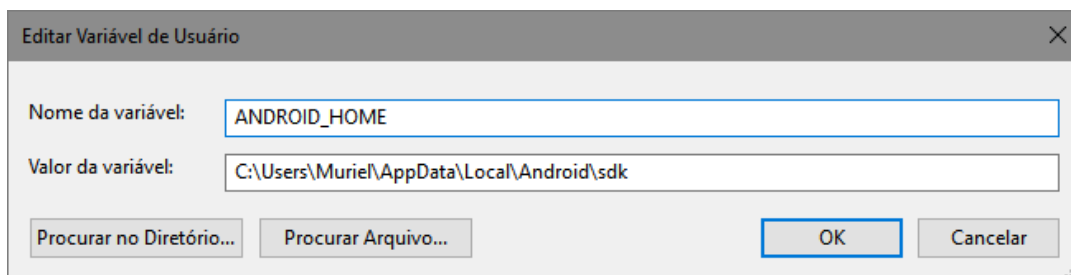


Figura 6: Configuração da Variável de Ambiente ANDROID_HOME

Fonte: Autoria Própria

Terminada as configurações da variável de ambiente, é necessário instalar o *framework React Native*, cujo o processo de instalação é realizado através do gerenciador de pacote do node. O Quadro 04 demonstra o comando de *download* e instalação do *React Native*.

Quadro 04: Comando de instalação do *React Native*.

```
npm install -g create-react-native-app
```

Fonte: Autoria Própria

A partir de então, o *React Native* está instalado e pronto para ser utilizado.

2.2.3.1. Criação e configuração de um projeto *React Native*

O processo de criação de um projeto *React Native* é feito via *prompt* de comando. Conforme demonstrado no Quadro 05.

Quadro 05: Comando de criação de um projeto *React Native*.

```
create-react-native-app NomeDoProjeto
```

Fonte: Autoria Própria

Com o projeto criado, é chegada a hora de enviar o projeto para o git. Utiliza-se nesse projeto o gerenciador de repositórios git chamado GitLab, que pode ser acessado no endereço eletrônico: https://gitlab.com/users/sign_in. Após acessar, é dado as opções de registrar uma conta ou acessar a plataforma a partir de uma conta já criada. Após entrar na plataforma, é redirecionado para a tela de projetos do usuário, onde mostra os projetos já criados pelo usuário, e a opção de novo projeto. Para criar um projeto, clicar em “New Project” (Figura 7).

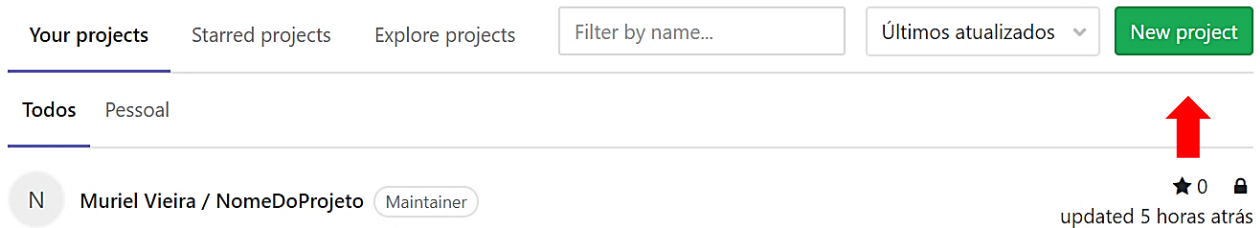


Figura 7: Tela de gerenciamento de projetos do GitLab

Fonte: Autoria Própria

Ao clicar em “*New Project*”, é redirecionado para a criação e configurações do novo projeto. O usuário pode optar por deixar seu projeto privado. Dessa forma, poderá conceder acesso aos outros usuários, cadastrando-os no projeto posteriormente. Também há a opção “interno”, que concede acesso para qualquer usuário registrado e autenticado no GitLab. A opção “público” libera o projeto para ser acessado, independentemente se está registrado e autenticado na plataforma GitLab (Figura 8).

Project path

https://gitlab.com/ mfv

Project name

my-awesome-project

Want to house several dependent projects under the same namespace? Create a group

Project description (optional)

Description format

Visibility Level

☒ Privado
Acesso ao projeto deve ser concedido explicitamente para cada usuário.

☐ Interno
O projeto pode ser acessado por qualquer usuário autenticado.

☐ Público
O projeto pode ser acessado sem a necessidade de autenticação.

☐ Initialize repository with a README
Allows you to immediately clone this project's repository. Skip this if you plan to push up an existing repository.

Create project Cancel

Figura 8: Tela de criação de projetos do GitLab

Fonte: Autoria Própria

Quando o projeto é criado, o repositório git inicia-se vazio, por padrão. Para adicionar o projeto *React Native* a esse repositório é necessário realizar o *upload* dos arquivos do projeto, processo feito através do *prompt* de comando do sistema operacional, onde é necessário navegar até a pasta onde está localizado o projeto *React Native*, e inserir e executar os comandos próprios da ferramenta (Quadro 06).

Quadro 06: Comandos de *upload* para um repositório git.

```
cd pasta_do_projeto
git init
git remote add origin https://gitlab.com/mfv/NomeDoProjeto.git
git add .
git commit -m "comentários das alterações feitas ex: subindo projeto para o git"
git push -u origin master
```

Fonte: Autoria Própria

Concluindo essa etapa, finaliza-se a configuração do ambiente de desenvolvimento utilizado na criação do projeto *Mobile LD-Player*.

2.2.3.2. Instalando o projeto *Mobile Ld-Player*

Para baixar e instalar o projeto, é necessário abrir o *prompt* de comando do sistema operacional, e executar o comando git que faz o download do projeto. O Quadro 07 contém o comando com o link do projeto, que está alocado de forma privada.

Quadro 07: Comando de *download* do projeto *MOBILE LD-PLAYER*

```
git clone https://gitlab.com/mfv/mobile-ld-player.git
```

Fonte: Autoria Própria

Após baixar o projeto advindo do repositório git, é imprescindível realizar as instalações das dependências dentro da pasta do projeto, para que esse possa ser executado (Quadro 08).

Quadro 08: Comando de instalação das dependências do projeto *MOBILE LD-PLAYER*

```
npm install
```

Fonte: Autoria Própria

Durante o desenvolvimento desse projeto foi usado como ambiente de teste um dispositivo Android. Ocorreu problemas ao usar emuladores para executar os códigos, devido à natureza desse projeto, que trabalha com arquivos e XML. Por isso recomenda-se fortemente usar um dispositivo Android conectado ao computador via USB. Para verificar se o dispositivo está conectando corretamente ao *Android Debug Bridge* (ADB), é necessário ir via *prompt* de comando à pasta onde o sdk está instalado, entrar em *platform-tools* e executar o comando que se encontra no Quadro 09.

Quadro 09: Comando de verificação de dispositivos *Android* conectados.

```
adb devices
```

Fonte: Autoria Própria

Finalmente, é chegada a hora de executar o projeto. Para isso é primordial estar com o *smarthphone* conectado e estar na pasta do mesmo, via *prompt* de comando, e executar a instrução como consta no quadro 10.

Quadro 10: Comando de execução do projeto

```
react-native run-android
```

Fonte: Autoria Própria

Para realizar os testes da aplicação, o código foi compilado e executado em um *smartphone Android* real, com exceção nos testes de responsividade da aplicação, verificando o comportamento da aplicação em telas maiores, como em *tablets*. Para esses testes foi usado o emulador *GenyMotion*.

A maioria dos dispositivos Android só pode instalar e executar aplicativos baixados do Google Play, por padrão. Entretanto, existe um processo a ser feito no aparelho *Android* para que seja possível instalar o aplicativo durante o desenvolvimento e usar o próprio aparelho para execução: ativar a depuração USB.

Para ativar a depuração USB no dispositivo, primeiro é necessário ativar o "Opções do desenvolvedor". Para ativar esse menu "secreto", é necessário acessar as Configurações, selecionar a opção Sobre o telefone e tocar sobre Número da Versão sete vezes seguidas (Figura 6). Será então exibida a mensagem "Você agora é um desenvolvedor" e o novo menu ficará disponível nas Configurações.

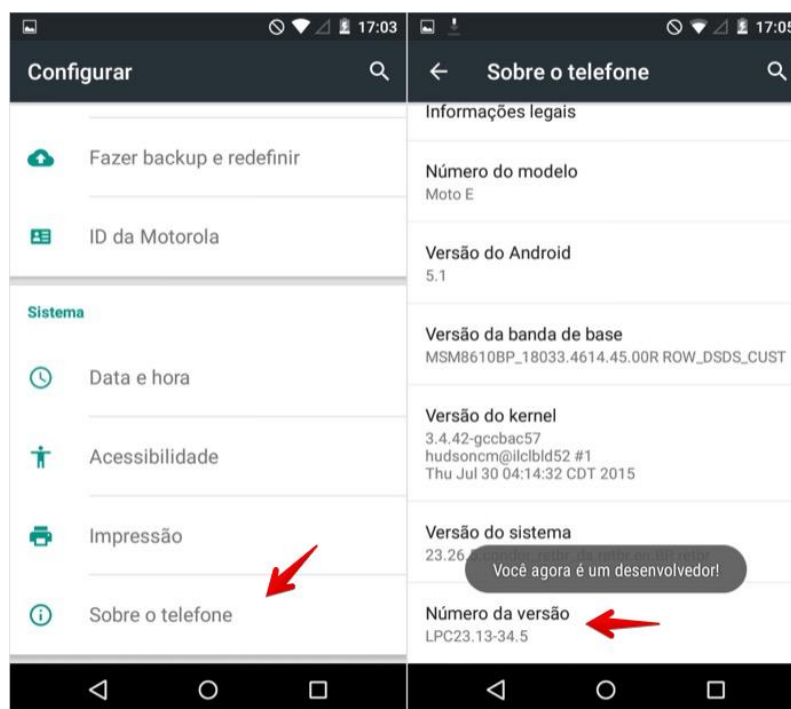


Figura 9: Ativação do modo desenvolvedor do *Android*

Fonte: BIJORA (2015)

Logo após, é necessário acessar esse novo menu, e realizar a ativação da opção "depuração USB" (conf. Figura 10).

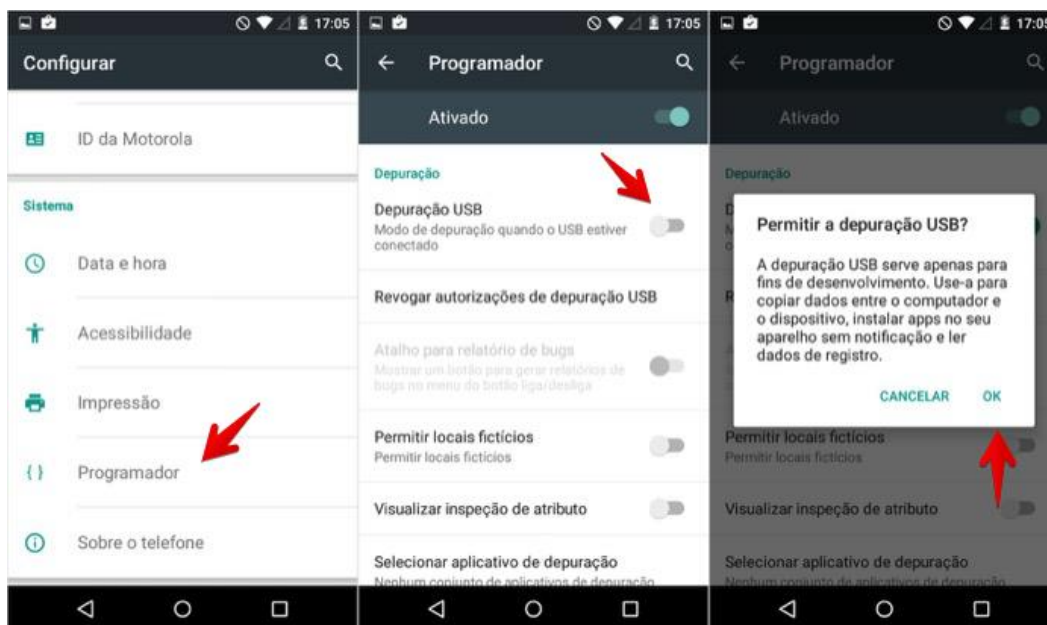


Figura 10: Ativação da Depuração USB Android

Fonte: BIJORA (2015)

Seguindo a metodologia, após o *setup* do ambiente de desenvolvimento, segue-se para o planejamento do desenvolvimento, definindo o ciclo de execução de desenvolvimento (Quadro 11).

Quadro 11: Ciclo de execução de desenvolvimento.

Definição do ciclo de execução de desenvolvimento.

- 1** – Desenvolvimento da tela de tutorial, contendo as instruções para o uso do aplicativo.
- 2** – Desenvolvimento da importação de arquivo *IMS* – LD.
- 3** – Gerar conteúdo do arquivo importado na tela.
- [..]

Fonte: Autoria Própria.

A tela primária (Figura 11) é a tela que abrirá no momento da execução do aplicativo. Essa interface direcionará o usuário para o tutorial da aplicação, O usuário também pode partir direto para o uso, a partir de importação do arquivo IMS-LD. Por fim, há um pequeno repositório de conteúdos de IMS-LD, que servem como exemplos de usabilidade para os usuários da aplicação.



LD-Player

Escolha um arquivo no formato
IMS Learning Design ao começar



Figura 11: Tela Principal – *Mobile LD-Player*

Fonte: Autoria Própria

Caso o usuário seja inexperiente na plataforma, ele poderá utilizar o tutorial prescrito na aplicação, onde terá informações relevantes sobre o IMS-LD, tais como: o que é o IMS-LD, qual a sua finalidade, como utilizá-lo, e por fim, contém instruções sobre a usabilidade e aplicabilidade do aplicativo (Figura 12).



Figura 12: Tela Tutorial – Mobile LD-Player

Fonte: Autoria Própria

Na Tela Primária, também há o botão “começar”. Ao clicar no botão é aberto uma tela do sistema operacional, indicando visualmente que o usuário deverá selecionar um arquivo no formato IMS-LD.

Na Figura 13, mostra-se a seleção de arquivos, onde o usuário deve escolher o arquivo que deseja abrir. O aplicativo filtra os arquivos que não estão no formato .zip, facilitando o uso para o usuário. Para escolher o arquivo basta clicar sobre o arquivo que o mesmo será importado. Se o arquivo escolhido não estiver no padrão IMS-LD, abrirá uma mensagem de erro informando ao usuário que o arquivo selecionado não está adequado ao aludido padrão.

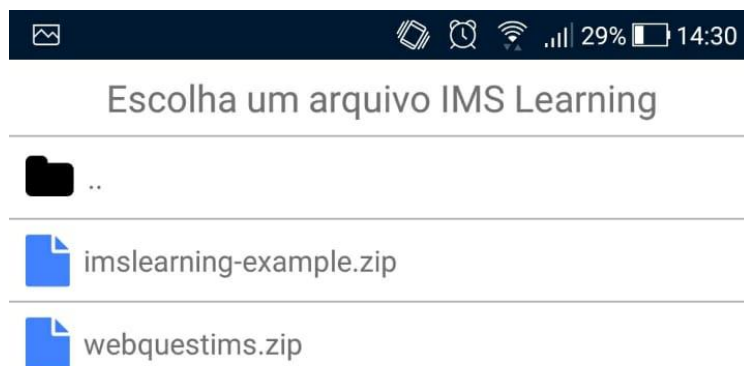


Figura 13: Selecionando o Arquivo – *Mobile LD-Player*

Fonte: Autoria Própria

A Figura 14, representa a arquitetura da funcionalidade de seleção de arquivos IMS-LD, ilustrando que após a seleção e validação do arquivo, seguirá o chamamento da funcionalidade que estrutura o arquivo IMS-LD na tela e permite a navegabilidade do conteúdo.

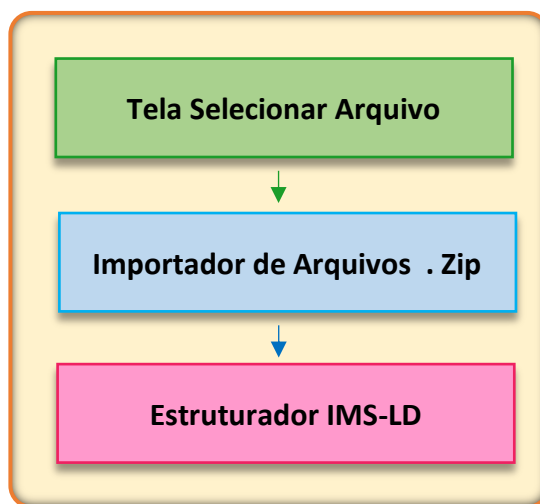


Figura 14: Representação da arquitetura da funcionalidade seleção de arquivos – *Mobile LD-Player*

Fonte: Autoria Própria

O Quadro 12, mostra a parte técnica da seleção do arquivo. No código, verifica-se se existe diretórios, caso exista, retornará função que possibilita a navegação nas pastas do dispositivo móvel. Verifica-se a extensão do arquivo corresponde ao formato “.zip”, para que mostre para o usuário, somente arquivos com esse tipo de extensão.

Quadro 12: Código da exibição e validação do arquivo IMS-LD

```
1 itemList(item) {
2
3   if (item.isDirectory() == true) {
4     return (
5       <TouchableOpacity
6         key={item.path}
7         style={styles.itemList}
8         onPress={() => this.navigateFoward(item.path)}
9       >
10        <Icon style={styles.icon} name="folder" size={25} color="#edbf42"
11      />
12        <Text>{item.name}</Text>
13      </TouchableOpacity>
14    )
15
16  } else {
17    let fileName = item.name;
18    var indexDot = fileName.lastIndexOf(".");
19    let extension = fileName.substring(indexDot);
20
21    if(extension == '.zip' || extension == '.ZIP')
22      return (
23        <TouchableOpacity
24          key={item.path}
25          style={styles.itemList}
26          onPress={() => this.selectFile(item.path)}
27        >
28          <Icon style={styles.icon} name="file" size={25}
29            color="#4281ff" />
30          <Text>{item.name}</Text>
31        </TouchableOpacity>
32      )
33    }
34  }
35
36 }
```

Fonte: Autoria Própria

Após a codificação que faz com que, apenas arquivos “.zip” sejam mostrados ao usuário, a Quadro 13 exibe a parte da seleção e leitura do arquivo escolhido.

Quadro 13: Código de seleção e leitura do arquivo IMS-LD

```
1 chooseFile() {
2
3   // Obtém uma lista de arquivos e diretórios no pacote principal
4   RNFS.readdir(this.getPath())
5
6   .then((result) => {
7     this.setState({ filesToShow: result, readyToShow: true });
8     // Aponta para o primeiro arquivo
9     return Promise.all([RNFS.stat(result[0].path), result[0].path]);
10  })
11
12  .then((statResult) => {
13    if (statResult[0].isFile()) {
14      // Se tiver um arquivo, leia
15      return RNFS.readFile(statResult[1], 'utf8');
16    }
17    return 'no file';
18  })
19
20  .then((contents) => {
21    // log the file contents
22    console.log(contents);
23  })
24
25  .catch((err) => {
26    console.log(err.message, err.code);
27  });
28 }
```

Fonte: Autoria Própria

A partir de então, é necessário descompactar o arquivo, para possibilitar a leitura do IMS-LD contido ali. O Quadro 14, exibe o código de descompactação do arquivo “.zip”.

Quadro 14: Descompactando o arquivo “.zip”

```
1 unzipFile(originZip){
2   let folder = '1'; //new Date().toLocaleString();
3   let directoryIMS = RNFS.ExternalStorageDirectoryPath + '/IMS/'+folder;
4
5   if(RNFS.exists( directoryIMS )){
6
7     RNFS.unlink(directoryIMS)
8     .then(() => {
9       console.log('FOLDER/FILE DELETED');
10    })
11    // irá lançar um erro, se o item para 'unlink' não existir
12    .catch((err) => {
13      console.log('CANT DELETE',err.message);
14      this.setState({showError:true})
15    });
16
17    RNFS.mkdir(directoryIMS)
18  }
19
20  let targetPath = directoryIMS;
21
22  unzip(originZip, targetPath)
23  .then((path) => {
24    console.log(`unzip completed at ${path}`)
25    this.openTree(path);
26  })
27  .catch((error) => {
28    console.log(error)
29    this.setState({showError:true})
30  })
31 }
```

Fonte: Autoria Própria

Com o arquivo descompactado, parte-se para a etapa de estruturar o IMS-LD contido no arquivo, de forma que já apresente para o usuário, todo o conteúdo lido no arquivo. O Quadro 15

mostra um exemplo, da parte inicial da estruturação do IMS-LD, onde é definido a “leitura” das “tags” do referido padrão. Ou seja, é feita a interpretação da sintaxe do IMS-LD.

Quadro 15: Estruturando a interpretação da sintaxe do IMS-LD

```
1 printAct(act){
2     let title = act[prefix+'ld:title'];
3     let identifier = act['$'].identifier;
4
5     return(
6         <TextTree key={typeof variable === 'undefined'?identifier:title}
7         onPress={() => this.viewContent(title, act)} >{title}</TextTree>
8     )
9 }
10
11 printPlay(play){
12     let title = play[prefix+'ld:title'];
13     let identifier = play['$'].identifier;
14     let acts = play[prefix+'ld:act'];
15     let retActs = [];
16
17     if(acts)
18     for (let i = 0; i < acts.length; i++) {
19         const act = acts[i];
20         retActs.push( this.printAct(act) );
21     }
22
23     return(
24         <ViewTree title={'Play - '+title}>
25             {retActs}
26         </ViewTree>
27     )
28 }
```

Fonte: Autoria Própria

Com a importação realizada com sucesso, a aplicação gera o conteúdo do arquivo na tela, possibilitando que o usuário navegue livremente. A figura 15 mostra a árvore de arquivos, por onde o usuário navega, acessando os conteúdos ali contidos.

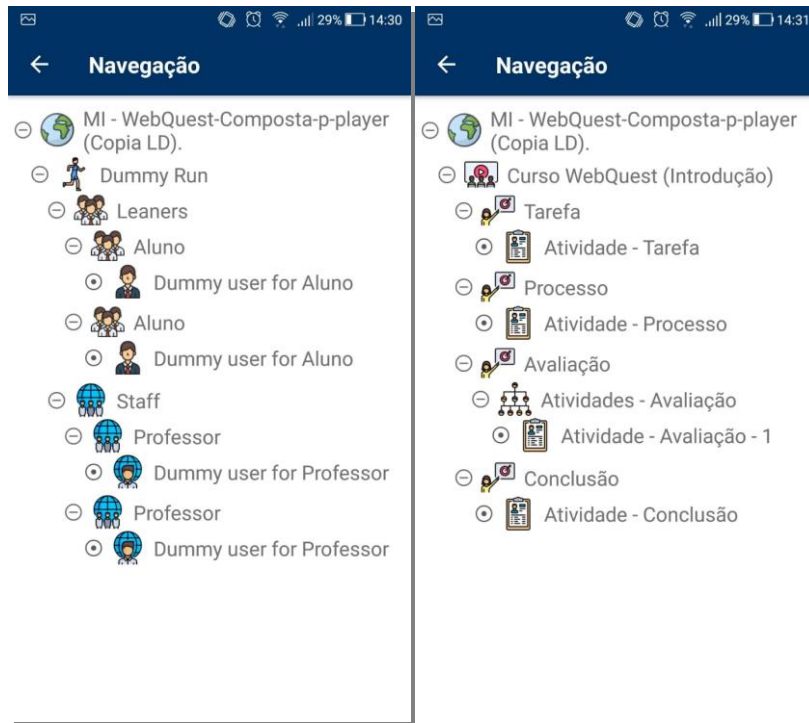


Figura 15: Exibindo Conteúdo Importado– *Mobile LD-Player*

Fonte: Autoria Própria.

A Figura 16 exibe os exemplos embutidos na aplicação. Esses exemplos ilustram para o usuário leigo o funcionamento do aplicativo. Os arquivos com os conteúdos de exemplo presentes, estão em um servidor de hospedagem *Hostgator*. Para ter acesso a esses conteúdos presentes na aplicação, é necessário que o usuário tenha conexão com a internet, para fazer o *download* dos arquivos.

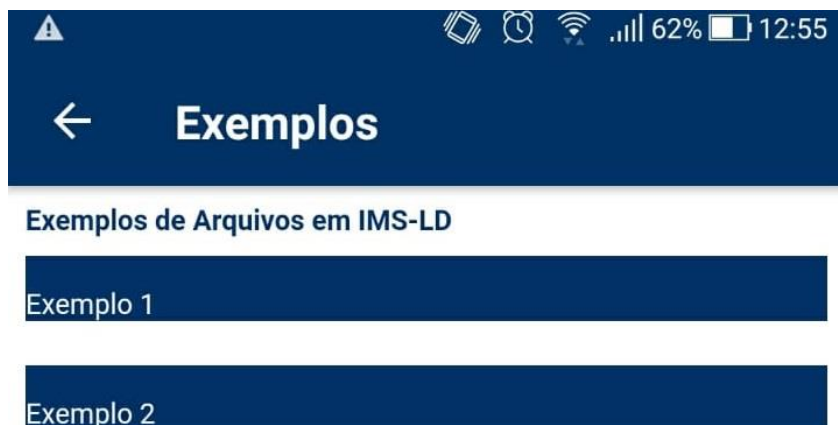


Figura 16: Tela de exemplos de conteúdos IMS-LD - *Mobile LD-Player*

Fonte: Autoria Própria.

2.2.3.3. Comercialização

Finalizando a fase de desenvolvimento, parte-se para a última fase da metodologia de desenvolvimento. Aqui consiste em realizar os testes de aceitação e testes pelo usuário, partindo para fazer os testes de lançamento, para finalmente realizar o lançamento do produto para comercialização.

O cronograma e escopo deste projeto não abrangeu testes de aceitação e testes de usuários. Portanto, deu-se seguimento para os testes de lançamento. Os testes de lançamento da aplicação envolveram testes funcionais (Quadro 16), que possuem o propósito de testar as funcionalidades, requisições e regras de negócios presentes na documentação do projeto.

Os testes funcionais são separados em introdução, itens a testar, critérios de sucesso, especificação dos testes e resumo dos resultados. A introdução vai explicar qual escopo do projeto à ser testado e os objetivos à serem alcançados; itens a testar cita quais são as funcionalidades à serem testadas; os critérios de sucesso descreve quais as ações esperadas para a funcionalidade; as especificações dos testes descrevem detalhadamente cada passo feito para ser realizado o teste da funcionalidade; em resumo dos resultados são registrados, excepcionalmente, os eventos ocorridos durante a execução dos testes, informando os resultados obtidos e possíveis anomalias encontradas.

Quadro 16: Testes Funcionais

Testes Funcionais do Sistema:

Introdução

Suíte de testes funcionais, com passos definidos, sendo eles: Introdução, Itens a testar, Critérios de Sucesso, Especificação dos testes e Resumo dos resultados. Os testes buscam validar e testar todas as funcionalidades desenvolvidas no projeto *Mobile LD-Player*. [...]

Fonte: Autoria Própria

Após os testes para o lançamento, é chegada a hora de realizar o lançamento da aplicação. O lançamento do *Mobile LD-Player* ocorreu via plataforma *PlayStore*. Ambiente de disponibilização de aplicativos oficial do sistema operacional Android (Figura 17).



Figura 17: Aplicativo disponível na *Play Store* – *Mobile LD-Player*

Fonte: Autoria Própria

3. CONSIDERAÇÕES FINAIS

Este relatório apresentou o processo de desenvolvimento do *Mobile LD-Player*, um aplicativo mobile que realiza a interpretação de materiais didáticos e conteúdos no padrão IMS – LD. Há uma quantidade significativa de editores que produzem material no formato IMS-LD, porém há poucas aplicações que fazem o consumo do conteúdo gerado nesses editores, principalmente no âmbito dos dispositivos móveis. Para diminuir essa disparidade, o *Mobile LD-Player* é indicado para ser utilizado por professores e principalmente alunos, via dispositivos móveis com o sistema operacional Android.

Durante o desenvolvimento da aplicação, foram compreendidas, analisadas e utilizadas diversas tecnologias relativamente novas, alicerçadas por uma metodologia ágil inerente para o desenvolvimento de dispositivos móveis, que teve como fruto uma ferramenta especializada. A disponibilização do *Mobile LD-Player* para o público em geral, por meio da loja virtual do Android, resulta na única ferramenta encontrada específica para interpretação de conteúdo IMS-LD na loja oficial da plataforma, tendo uma massa de usuários em potencial. A ferramenta também busca contribuir para a disseminação do padrão IMS-LD, fomentando de forma indireta o uso de editores do dito padrão, já que está disponível mais uma ferramenta acessível que permite o uso do conteúdo gerado por tais ferramentas editoras.

O produto, apesar do referencial teórico explanado neste relatório, que justificou o seu desenvolvimento, até o momento da escrita desse relatório não passou por processo de validação e estudos de casos com usuários. Um levantamento de dados com os *feedbacks* dos usuários é importante para a continuação do projeto, pois a partir disto, pode-se melhorar pontos fracos e possíveis falhas encontradas no projeto e otimizar os pontos positivos, sempre pensando na experiência do usuário e na aplicabilidade do *software*.

Aproveitando da estrutura já construída neste projeto, para as próximas versões do *Mobile LD-Player* é sugerido como trabalho futuro que cada usuário da aplicação armazene os conteúdos importados em uma área específica, alocado em um banco de dados, para que fique disponível sempre que o usuário desejar.

REFERÊNCIAS

ABRAHAMSSON, P.; HANHINEVA, A.; HULKKO, H.; IHME, T.; JÄÄLINOJA, J.; KORKALA, M.; KOSKELA, J.; KYLLONEN, P.; SALO, O. Mobile-d: An agile approach for mobile application development. Proc of the OOPSLA'04 Conference, 2004.

ADVANCED DISTRIBUTED LEARNING. Sharable Content Object Reference Model – SCORM 2004 2º Edition – Overview, 2004.

ASSIS, M. P.; ALMEIDA, M. E. B. Learning design e tecnologias: criação de ambientes colaborativos para a aprendizagem. Psicologia da Educação, 2017.

BIJORA, H. Como usar o Vysor e espelhar o Android no seu computador. 2015. Disponível em: <<http://www.techtudo.com.br/dicas-e-tutoriais/noticia/2015/09/como-usar-o-vysor-e-espelhar-o-android-no-seu-computador.html>>. Acesso em: 19 jul. 2018.

BOARETTO, R.; NUNES, C. A. A.; FILATRO, A. Representação de uma ação de aprendizagem através do IMS-Learning Design e implicações para o desenvolvimento de LMSs. Anais do 4º Seminário Nacional ABED de EAD, 2006.

DUTRA, R. L. S.; TAROUÇO, L. M. R. Objetos de Aprendizagem: uma comparação entre SCORM e IMS Learning Design. RENOTE: revista novas tecnologias na educação [recurso eletrônico]. Porto Alegre, RS, 2006.

FACEBOOK. React Native Docs. 2018. Disponível em: <<https://facebook.github.io/react-native/docs/>>. Acesso em: 13 jul. 2018.

FISCARELLI, R. B. O. Material didático e prática docente. Revista Ibero Americana em Estudos de Educação. V. 2, n.1, 2007, p. 01-09.

FLORA, H. K.; CHANDE, S. V. A REVIEW AND ANALYSIS ON MOBILE APPLICATION DEVELOPMENT PROCESSES USING AGILE METHODOLOGIES. International Journal of Research in Computer Science, 2013.

HAMMERSHOJ, A.; SAPUPPO, A.; TADAYONI, R. Challenges for Mobile Application Development. In: International Conference on Intelligence in Next Generation Networks. Berlin, 2010.

JEONG, Y.; LEE, J.; SHIN, G.. Development process of mobile application SW based on agile methodology. In: Advanced Communication Technology, 2008. ICACT 2008. 10th International Conference on. IEEE, 2008. p. 362-366.

JÚNIOR, A. F. S.; DA SILVA, L. R.; FERNANDES, C. T. IME: Um Modelo e Gerador de Editores de Atividades de Aprendizagem em IMS Learning Design. In: Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE).

KOPER, R. (2001). “Modeling Units of Study from a Pedagogical Perspective: the pedagogical meta-model behind EML”, 2018.

KOPER, R.; OLIVIER, B. (2004). “Representing the Learning Design of Units of Learning”. Educational Technology & Society, vol. 7, n. 3, pp. 97-111.

MARTENS, H.; VOGTEN, H. (2005). A Reference Implementation of a Learning Design Engine. In: Koper, R and Tattersall, C, (eds) Learning Design, A Handbook on Modelling and Delivering Networked Education and Training. Springer-Verlag, Berlin Heidelberg, pp 91-108

MORAN, J. M.; MASETTO, M. T.; BEHRENS, M. A. Novas tecnologias e mediação pedagógica. 12. ed. Campinas -SP, 2006.

PAQUETTE, G. et al. Implementation and deployment of the IMS learning design specification. Canadian Journal of Learning and Technology/La revue canadienne de l'apprentissage et de la technologie,, 2005.

RAHIMIAN, V.; RAMSIN, R. Designing an agile methodology for mobile software development: A hybrid method engineering approach. In: Research Challenges in Information Science, 2008.

RODRIGUES JUNIOR, E.; FERNANDES, F. J. Os Desafios da Educação Frente as Novas Tecnologias. In: Seminário Internacional de Educação Superior, 2014, Sorocaba - SP. ANAIS do IV Seminário de Extensão, 2014.

SIMPSON, R. Android overtakes Windows for first time. Disponível em: <<http://gs.statcounter.com/press/android-overtakes-windows-for-first-time>>. Acesso em jun. 2018.

VALENTE, J. A.; ALMEIDA, F. J. Visão Analítica da Informática na Educação: a questão da formação do professor. Revista Brasileira de Informática na Educação, Sociedade Brasileira de Informática na Educação, 1997.

WINTERFELDT, J. Context-Aware Mobile Crowd Sensing using Mobile Hybrid Application Frameworks. 2017.

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
GOIÁS - CÂMPUS JATAÍ
DEPARTAMENTO DE ÁREAS ACADÊMICAS
CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS**

**Muriel Ferreira Vieira
20151020230249**

Documentação do Software

Jataí
2018

Tabela 01: Fases do ciclo da Metodologia MASAM.

Fase	Atividade	Tarefa
Preparação	Compreensão do produto	Sumário do produto e pré-planejamento
	Compartilhamento do conceito do produto	Definição do usuário e análise inicial do produto
	Setup do projeto	Setup dos processos de desenvolvimento e dos recursos do projeto, realização de estudo inicial
Personalização	Entendimento das necessidades do usuário	Story card workshop e design da interface de usuário
	Definição e elaboração da arquitetura	Análise dos requisitos não funcionais, definição da arquitetura e identificação dos padrões de <i>software</i> a serem utilizados
Desenvolvimento	Preparação e Implementação	Setup do ambiente de desenvolvimento e planejamento do desenvolvimento
	Ciclos de lançamentos	Planejamento dos lançamentos, dos ciclos ou iterações e a realização propriamente dita desses lançamentos
Comercialização	Testes do sistema	Testes de aceitação e testes pelo usuário
	Comercialização do produto	Testes de lançamento e lançamento do produto

Quadro 01: Sumário - LDPlayer**Sumário do Produto**

O *Mobile LD-Player* é uma plataforma mobile, desenvolvida para o sistema operacional *Mobile Android*. Essa plataforma, como principal funcionalidade, possibilitará a leitura e consumo de materiais desenvolvidos no padrão IMS – *Learning Design*. Os arquivos lidos pelo Leitor poderão ser advindos de qualquer Sistema Operacional, plataforma web, desde que venha no padrão estipulado. A aplicação será destinada, principalmente para alunos e professores de diversos níveis educacionais, independente de faixa etária. A aplicação contará com criação de usuários, para obter um maior controle de acesso do seu público alvo. O usuário poderá cadastrar, informando seu nome, e-mail e senha. O usuário poderá recuperar a senha via requisição para o e-mail, e poderá logar autenticado no sistema. O usuário, então autenticado, poderá importar o arquivo *Learning Design*. A Partir da importação, a aplicação

interpretará o arquivo e gerará o conteúdo na tela. Com o conteúdo na tela, o usuário poderá navegar em todo material ali presente, da maneira que foi organizada pelo criador do conteúdo.

Quadro 02: Definição de Usuários

Definição de usuários

Os usuários principais da aplicação *Mobile LD-Player* constituem-se de professores e alunos, com perfis abrangentes.

Alunos: De qualquer nível educacional, sem restrições de faixa-etária, desde que tenha capacidade de utilizar um *smartphone*, podendo pertencer a esfera pública ou privada de ensino. Sendo o principal utilizador do *software*, este terá a função de consumir os materiais didáticos digitais disponibilizados pelo seu professor, ou disponibilizado por qualquer repositório de material didático digital do formato IMS-LD.

Professores: De qualquer nível educacional, sem restrições de faixa-etária, podendo pertencer a esfera pública e privada de ensino, sendo o principal papel deste usuário, testar o material didático advindo de editores de *Learning Design*, e divulgar o *LD-Player* entre seus alunos, para que utilizem o material didático digital.

Quadro 03: Definição de Recursos e Ferramentas.

LD-Player

Definição de Recursos e Ferramentas

Computador (para o desenvolvimento da aplicação): Notebook ASUS PU401LA – Processador Intel Core i5 4200U, 6GB RAM, 500HD

Editor de Código: Microsoft Studio Code.

Smartphone (para testes da aplicação): ASUS ZenFone 3 Zoom.

Quadro 11: Ciclo de execução de desenvolvimento

LD-Player

Definição do ciclo de execução de desenvolvimento.

- 1 – Desenvolvimento da tela de tutorial, contendo as instruções para o uso do aplicativo.
- 2 – Desenvolvimento da importação de arquivo *IMS* – LD.
- 3 – Gerar conteúdo do arquivo importado na tela.
- 4 - Implementar a navegabilidade do conteúdo gerado.

Quadro 16: Testes funcionais.

Teste Funcional do Sistema:

Introdução

Suíte de testes funcionais, com passos definidos, sendo eles: Introdução, Itens a testar, Critérios de Sucesso, Especificação dos testes e Resumo dos resultados. Os testes buscam validar e testar todas as funcionalidades desenvolvidas no projeto *LD-Player*.

Itens a testar

Número de ordem do item	Nome do item	Descrição da funcionalidade
1	Transição entre telas	Verificar se as transições de tela ocorrem de acordo com o esperado.
2	Validar tipo de arquivo a ser importado	Verificar se os arquivos apresentados correspondem ao formato padrão necessário para a importação.
3	Validar Importação	Verificar a funcionalidade de importação de arquivo <i>IMS-Learning Design</i> . Verificar se o conteúdo foi gerado na tela.
4	Verificar navegabilidade do conteúdo importado	Validar navegabilidade no conteúdo gerado pela importação

Critérios de sucesso

Número de ordem do item	Critérios
1	Testar interface de usuário atentando-se ao comportamento da aplicação em várias resoluções. Se terá uma quebra de elementos, ou despixelização dos ícones inseridos.
2	No momento de apresentar os arquivos, a aplicação deverá mostrar somente os arquivos no formato .zip
3	Ao selecionar o arquivo, a aplicação verifica se o arquivo contém conteúdos no formato IMS-LD. Caso o arquivo corresponda ao formato esperado, o conteúdo do mesmo será gerado na tela.
4	Navegabilidade no conteúdo gerado pela importação.

Especificação dos testes

Procedimentos de Testes:

Transição entre interfaces

Abrir a interface: Tela Primária.

Abrir a interface: Tela de Tutorial.

Transitar os slides até o final.

Clicar no botão Voltar.

Abrir a interface: Tela Primária.

Clicar no botão Cancelar.

Abrir a interface: Tela de *Login*.

Clicar no botão Cancelar.

Abrir a interface: Tela de *Login*.

Inserir e-mail e senha.

Clicar no botão Entrar.

Abrir: Tela Principal.

Clicar no botão Sair.

Validar Importação

Clicar no Botão Iniciar.

Selecionar o arquivo no formato *IMS-Learning Design*.

Verificar a navegabilidade do conteúdo gerado na tela.

Verificar Navegabilidade

Interface: Tela Principal.

clicar nas Guias de interação na árvore de conteúdos.

Resumo dos Resultados:

Item 1: O aplicativo foi testado no Smartphone Zenfone 3 Zoom, e por final no emulador GenyMotion simulando vários tamanhos de telas, variando entre 4,0” à 10” polegadas, a fim de verificar a disposição dos componentes na tela, e possíveis distorções recorrentes das mudanças de tamanho e resolução do ambiente.

Item 2: A partir da Tela Primária, foi clicado o botão iniciar, abrindo a tela de seleção de arquivo. Ao abrir a tela de seleção, é feita a navegação entre as pastas, onde verifica-se que somente é exibido os arquivos .zip.

Item 3: A partir da Tela Principal, clica-se no menu superior, na opção *Começar*. É selecionado o arquivo no formato *IMS-Learning Design*, e confirma para importar. O arquivo é interpretado pela aplicação, que disponibiliza o conteúdo gerado do arquivo na tela do *smartphone*.

Item 4: Após a importação com sucesso, aqui é verificado a navegabilidade do conteúdo, através da árvore de arquivo, navegou normalmente entre todo conteúdo disponibilizado.