



Universidade de Brasília

Instituto de Ciências Exatas  
Departamento de Ciência da Computação

# Guia de planejamento e criação de cursos utilizando EMLs: estudo de caso com um curso introdutório de programação

Lucas N. C. Cardoso

Monografia apresentada como requisito parcial  
para conclusão do Curso de Computação — Licenciatura

Orientadora  
Prof.a Dr.a Germana Menezes da Nóbrega

Brasília  
2021

Ficha catalográfica elaborada automaticamente,  
com os dados fornecidos pelo(a) autor(a)

CC268g Cardoso, Lucas N. C.  
Guia de planejamento e criação de cursos utilizando EMLs:  
estudo de caso com um curso introdutório de programação /  
Lucas N. C. Cardoso; orientador Germana Menezes da Nóbrega.  
- Brasília, 2021.  
104 p.

Monografia (Graduação - Computação) -- Universidade de  
Brasília, 2021.

1. IMS Learning Design. 2. Ensino de Programação. 3. EML.  
4. Linguagem de Modelagem Educacional. 5. Design  
Educacional. I. Nóbrega, Germana Menezes da, orient. II.  
Título.



# Dedicatória

Dedico este trabalho a todos que se aventuram ao ato de ensinar; nosso compromisso maior é e sempre será com nossos alunos: Que nossas aulas sejam motivadoras, que sejam propícias à construção de novos conhecimentos, e que sejam prazerosas para os alunos e os professores que as ministram.

# Agradecimentos

Agradeço aos meus pais Marcio e Flor, pelo apoio incondicional, incentivo constante e por terem proporcionado a mim e meus irmãos uma educação sem igual. Aos meus avós Cléa, Claudionor, Aricléa e Mozart por todo o afeto. À minha esposa Júlia, por todo amor, companheirismo e por me estimular a ser o meu melhor. Aos meus irmãos, Bruno, Thiago e Izabela, por serem minhas referências e por participarem de toda a minha jornada. À Carol e Flávia, por todo suporte e carinho. Ao Evandro e Wanessa, por todo acolhimento, preocupação e cuidado. À Paula e Luiz Felipe por terem me recebido e aceito como parte da família. Ao Gustavo e Deborah, por sempre estarem presentes, independente da distância. Ao corpo docente da Universidade de Brasília, em especial à Profa. Leticia, Profa. Maria de Fatima e Prof. Zaghetto, por terem despertado meu interesse pela docência durante as disciplinas de teoria e prática pedagógica em informática, e à Profa. Germana, pela calorosa recepção durante um período muito necessário. Agradeço também a todos que fizeram parte de minha jornada acadêmica, e que contribuíram de forma direta ou indireta com meu processo de aprendizagem.

# Resumo

*Learning Design* é o estudo de meios de modelagem de experiências de ensino, podendo esses serem cursos, aulas ou atividades. Este trabalho objetiva ilustrar motivações para aplicação da padronização de artefatos de ensino em instituições, bem como preparar o corpo docente para desempenho da atividade de modelagem em Linguagem de Modelagem Educacional (EML) em padrão *IMS Learning Design* (IMS LD). Propõe-se então um guia com instruções para modelagem de cursos em IMS LD, exemplificando esse processo com um curso introdutório de programação em linguagem Java de uso direto para instituições de ensino fundamental e médio da rede pública. Esse curso possui atividades de programação por meio visual usando a biblioteca JKarel, um exemplo de atividade desplugada, e uma atividade inspirada na Aprendizagem Baseada em Projetos.

**Palavras-chave:** EML, Linguagem de Modelagem Educacional, Design Educacional, *IMS Learning Design*, Ensino de Programação

# Abstract

Learning Design is the study of ways of modelling learning experiences, those being activities, lessons or entire courses. This paper outlines motives for educational institutions to invest in the formalization of teaching efforts, as well as to prepare the teaching staff to use the Educational Modelling Language (EML) described in the IMS Learning Design (IMS LD) standard. A guide with instructions on the whole process is proposed in this paper, and also an introductory course in Java programming language, used as an example of the application of the guide. The course created for this paper has activities using Visual Based Programming through the JKarel library, an example of an unplugged activity, and one activity inspired in Project-Based Learning.

**Keywords:** EML, Education Modelling Language, Learning Design, IMS Learning Design, CS Teaching

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Contextualização . . . . .	1
1.2	Motivação e Justificativa . . . . .	1
1.3	Problema de Pesquisa . . . . .	5
1.4	Objetivos . . . . .	5
1.4.1	Objetivo Geral . . . . .	5
1.4.2	Objetivos Específicos . . . . .	6
1.5	Metodologia . . . . .	7
1.6	Estrutura deste Documento . . . . .	7
<b>2</b>	<b>Revisão de Literatura</b>	<b>9</b>
2.1	Iniciativas recentes em produção de material didático e ensino de programação	9
2.2	Suporte computacional ao planejamento docente . . . . .	12
<b>3</b>	<b>Referencial Teórico</b>	<b>18</b>
3.1	Das escolhas para o curso proposto . . . . .	18
3.2	Movimento <i>Maker</i> . . . . .	20
3.2.1	Movimento <i>Maker</i> no Brasil . . . . .	21
3.2.2	Produtos e Tecnologias <i>Maker</i> no Mercado . . . . .	22
3.2.3	Movimento <i>Maker</i> na Educação . . . . .	23
3.3	<i>Learning Design</i> e Linguagens de Modelagem Educacional . . . . .	25
3.4	Padrão IMS LD . . . . .	26
3.5	Ferramenta adotada . . . . .	29
<b>4</b>	<b>Proposta</b>	<b>30</b>
4.1	Ao Tutor: Um curso de Introdução à Programação . . . . .	30
4.2	Ao <i>Learning Designer</i> . . . . .	34
4.2.1	Visão do plano do curso em Linguagem de Modelagem Educacional . . . . .	35
4.2.2	Guia para planejamento de cursos utilizando o padrão IMS LD . . . . .	37

<b>5 Conclusão</b>	<b>43</b>
5.1 Contribuições . . . . .	43
5.2 Trabalhos Futuros . . . . .	44
<b>Referências</b>	<b>45</b>
<b>Apêndice</b>	<b>48</b>
<b>A <i>Coding 101</i>: Plano de Curso</b>	<b>49</b>
<b>B Guia para Modelagem de Curso em Padrão IMS LD Utilizando o Software <i>Reload Editor</i></b>	<b>60</b>

# Lista de Figuras

1.1	Dificuldades para o uso de tecnologias em atividades pedagógicas, em % de professores de escolas urbanas do Brasil. Fonte: CGI.BR/NIC.BR, Centro Regional de Estudos para o Desenvolvimento da Sociedade da Informação (Cetic.br) - 2021 . . . . .	4
2.1	Linha do tempo de iniciativas relevantes ao amparo do corpo docente com uso de tecnologia. . . . .	14
2.2	Resumo comparativo dos editores LD. Fonte: JÚNIOR, Aladir F. Silva; SILVA, Leandro Roberto da; FERNANDES, Clovis Torres. Panorama dos Editores de Atividades de Aprendizagem em IMS Learning Design. XXIV Simpósio Brasileiro de Informática na Educação (SBIE 2013). (336-345).. . . . .	16
3.1	<i>LiteMaker</i> : Laboratório <i>Maker</i> móvel para uso em instituições de ensino. Fonte: <a href="http://lite.acad.univali.br/pt/lite-maker/">http://lite.acad.univali.br/pt/lite-maker/</a> Acessado em: 28/05/2021 . . . . .	22
3.2	Modelo conceitual da estrutura do <i>Learning Design</i> por nível. . . . .	27
3.3	Diagrama UML do IMS LD em Nível C; Classes principais acentuadas por gradiente cinza. Fonte: Adaptado de <i>IMS Learning Consortium</i> (2003). Traduzido pelo autor. . . . .	28
4.1	Biblioteca JKarel de aprendizagem por meio visual. . . . .	32
4.2	Atividade Desplugada - <i>Code Bot</i> . . . . .	32
4.3	Exemplo de Modelo Narrativo de Ficção Interativa. Fonte: FREIRE, George André Alve; MADEIRA, Charles Andryê Galvão. Modelo conceitual para criação de jogos voltados para a tomada de decisão em gerenciamento de projetos. Anais dos Workshops do IV Congresso Brasileiro de Informática na Educação (CBIE 2015). . . . .	33

4.4	Exemplo de símbolos a serem usados para confecção visual de algoritmos segundo Ascencio(2007). Fonte: ASCENCIO, Ana Fernanda Gomes; CAMPOS, Edilene Aparecida Veneruchi. Fundamentos da Programação de Computadores: Algoritmos, Pascal, C, C++ e Java. Pearson Prentice Hall, 2007 . . . . .	34
4.5	Modelo estrutural do curso <i>Coding 101</i> . . . . .	36
4.6	Descrição geral do <i>Coding 101</i> no <i>Reload Editor</i> . . . . .	37
4.7	Exibição em árvore das atividades do curso <i>Coding 101</i> no <i>Reload Editor</i> . . . . .	38
4.8	Papéis do curso <i>Coding 101</i> no <i>Reload Editor</i> . . . . .	38
4.9	Recursos do curso <i>Coding 101</i> no <i>Reload Editor</i> . . . . .	39
4.10	Peça representando o curso <i>Coding 101</i> no <i>Reload Editor</i> . . . . .	40
4.11	Exibição em árvore do pacote de conteúdo <i>IMS</i> no <i>Reload Editor</i> . . . . .	41

# Lista de Tabelas

1.1 Quantidade de Matrículas e Número de Concluintes por Curso de Tecnologia da Informação Presenciais do Ensino Superior (2019) . . . . .	2
--	---

# Lista de Abreviaturas e Siglas

**DIY** Do It Yourself.

**EML** Educational Modelling Language.

**FabLab** Fabrication Laboratory.

**IBL** Inquiry-Based Learning.

**IMS LD** IMS Learning Design.

**INEP** Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira.

**LD** Learning Design.

**MEC** Ministério da Educação.

**MIT** Massachusetts Institute of Technology.

**PBL** Problem-Based Learning.

**PJBL** Project-Based Learning.

**STEAM** Science, Technology, Engineering, Art and Mathematics.

**TI** Tecnologias da Informação.

**UoL** Units of Learning.

**XML** Extensible Markup Language.

# Capítulo 1

## Introdução

### 1.1 Contextualização

O ensino de habilidades da área da Ciência da Computação, como a implementação e o desenvolvimento de algoritmos computacionais, tem se mostrado um grande desafio; Ora pelo caráter abstrato de seus conceitos fundamentais, ora pela falta de contato prévio do aluno ao adentrar um curso técnico ou de ensino superior na área, muitas vezes por falta de incentivo ou investimentos pelas instituições de ensino tanto da iniciativa pública quanto particular. Em [1] o autor, dada a presença de computadores em nosso cotidiano, classifica o estudo de algoritmos e programação como uma atividade essencial para o desenvolvimento do raciocínio necessário para a solução de problemas computacionais. Por se tratar de uma atividade orientada à resolução de problemas práticos e exercício de pensamento lógico, é uma habilidade importante a ser desenvolvida até por indivíduos que não irão necessariamente buscar uma profissão na área de Tecnologias da Informação (TI).

Neste estudo, identificamos desafios do ensino de programação, e também analisamos a dificuldade de instituições de ensino relacionada ao registro e reuso de iniciativas de aprendizagem desenvolvidas pelo corpo docente. Esse estudo visa discutir uma proposta de formalização de artefatos de ensino, e mostrar sua aplicação em um curso de programação para leigos.

### 1.2 Motivação e Justificativa

Por se tratar de uma disciplina com grande barreira de entrada para iniciantes na prática de codificação de *scripts* em linguagens de programação e entendimento de seus conceitos fundamentais, não é surpresa que os cursos do Ensino Superior da área computacional apresentem números alarmantes de alunos egressos.

Segundo censo disponibilizado em pesquisa conjunta pelo Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP) e pelo Ministério da Educação (MEC), pode-se analisar os seguintes dados referentes à quantidade de alunos ingressantes em cursos da área de Ciência da Computação e número de alunos concluintes de seus respectivos cursos.

Tabela 1.1: Quantidade de Matrículas e Número de Concluintes por Curso de Tecnologia da Informação Presenciais do Ensino Superior (2019)

Curso	Matrículas	Concluintes	Concluintes (%)
Computação formação de professor	5,908	518	8.77 %
Ciência da computação	65,053	6,426	9.88 %
Engenharia de computação (DCN Computação)	12,993	1,337	10.29 %
Sistemas embarcados	44	5	11.36 %
Segurança da informação	2,203	330	14.98 %
Sistemas de informação	121,762	16,770	13.77 %
Sistemas para internet	6,564	938	14.29 %
Banco de dados	1,182	274	23.18 %
Defesa cibernética	81	19	23.46 %
Gestão da tecnologia da informação	13,053	2,675	20.49 %
Redes de computadores	12,021	2,517	20.94 %
Engenharia de software	6,814	405	5.94 %
Jogos digitais	6,684	985	14.74 %
Programas interdisciplinares abrangendo computação e TIC	3,109	254	8.17 %
Total	257,471	33,453	12.99 %

Fonte: INSTITUTO NACIONAL DE ESTUDOS E PESQUISAS EDUCACIONAIS ANÍSIO TEIXEIRA. Sinopse Estatística da Educação Superior 2019. Brasília: Inep, 2020. Disponível em: <<http://portal.inep.gov.br/basica-censo-escolar-sinopse-sinopse>>. Acesso em: 18/04/2021

Segundo o censo, temos que os cursos da grande área da Computação possuem um número muito baixo de concluintes, se comparado ao número de alunos matriculados. O grande número de alunos desistentes desses cursos destaca um problema alarmante, desperta a necessidade de buscar as causas dessas desistências, e discorrer sobre possíveis soluções para este desafio.

Em estudo realizado analisando o curso de Licenciatura em Computação da Universidade do Estado do Amazonas, foi identificado que as disciplinas de Introdução a Programação de Computadores, Programação de Computadores e Algoritmos, Algoritmos e Estrutura de Dados I possuem o maior índice de reprovação e trancamento dos alunos durante os anos iniciais do curso.[2]

Em [3] o autor, ao investigar dados referentes aos índices de reprovação em disciplinas nos cursos de Computação na Universidade de Brasília, identificou realidade semelhante, com as disciplinas de Computação Básica, Estrutura de Dados, Organização de Arquitetura de Computadores, Software Básico e Organização de Arquivos dentre as disciplinas com maior grau de reprovação. Ambos estudos ilustram o fato de disciplinas introdutórias

de programação em cursos de ensino superior agirem como obstáculos significativos no percurso do aluno para sua graduação.

Esses dados despertam o questionamento das possíveis causas para o número elevado de desistentes nos cursos de Computação, e se existe algum contato prévio ou treinamento na prática de programação para alunos do ensino fundamental e ensino médio. Caso um aluno tenha sua primeira experiência com codificação em um ambiente universitário, a alta cobrança juntamente com a falta de conhecimento prévio podem servir como desestimulantes para que esse aluno siga seus estudos e conclua seu curso.

Avaliando as Diretrizes Curriculares Nacionais emitidas pelo MEC para o Ensino Médio e para o Ensino Fundamental de Nove Anos, não existe nenhuma menção sobre a necessidade de serem aplicados conteúdos relacionados à programação e utilização de computadores. A BNCC <sup>1</sup> definida pelo MEC cita os estudos de tópicos computacionais e de informática de forma vaga, definindo apenas como habilidades a serem desenvolvidas a capacidade do aluno de analisar e discorrer sobre uso e impacto da tecnologia em uma sociedade. Já na BNCC-EM <sup>2</sup> temos a presença da habilidade pensamento computacional, porém citada de forma breve e a ser desenvolvida durante as aulas da disciplina matemática, com uso de modelos e teoria matemáticas. Com isso, vemos que não temos como metas curriculares do Ensino Fundamental ou do Ensino Médio atividades que preparem o aluno para uso de computadores por meio de aulas de informática básica, nem ao menos que prepare o aluno para um curso de graduação ou técnico que tenha em seu currículo disciplinas de programação de computadores. Isso se torna um obstáculo para que o aluno tenha sua primeira vivência em atividades da área.

Se por sua vez não existe uma exigência definida do MEC para conteúdos previstos para a grade curricular que desenvolva habilidades dos alunos para uso de tecnologias essenciais, um grande número de instituições de ensino são equipadas com laboratórios de informática. Em censo escolar da Educação Superior referente ao ano 2020, divulgado pelo INEP<sup>3</sup> em 2021, identificamos alguns dados importantes para este estudo: 91,3% das instituições de ensino fundamental federais possuem computador de mesa disponíveis para uso dos alunos, enquanto esse número cai para 76,7% em instituições estaduais, 65,5% em instituições do setor privado e para 38,3% de instituições municipais. Esses dados são importantes para nos informar do grande número de instituições de ensino públicas que já passaram por todo o processo de licitação, compra e instalação de laboratórios de informática. Porém, somente a presença dos computadores não é o bastante, pois é essencial

---

<sup>1</sup>Portaria nº 1.570, publicada no D.O.U. de 21/12/2017, Seção 1, Pág. 146

<sup>2</sup>Portaria nº 1.348, publicada no D.O.U. de 17/12/2018, Seção 1, Pág. 33.

<sup>3</sup>FONTE: INEP. Microdados da Educação Superior 2019. Acessado em 14/05/2021 - <https://www.gov.br/inep/pt-br/areas-de-atuacao/pesquisas-estatisticas-e-indicadores/censo-da-educacao-superior/resultados>

**Dificuldades para o uso de tecnologias em atividades pedagógicas, em % de professores**

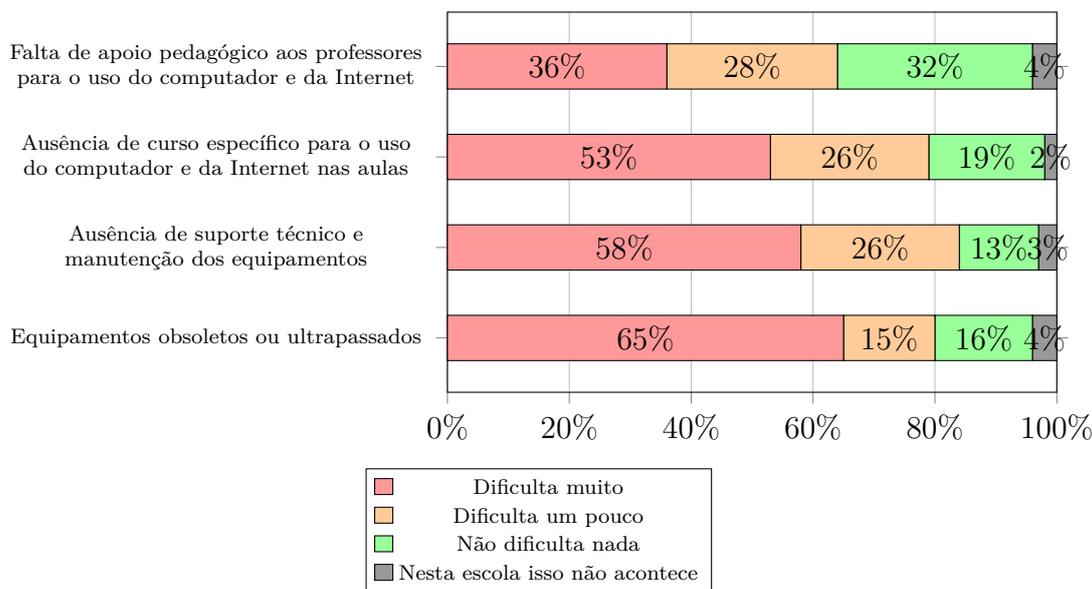


Figura 1.1: Dificuldades para o uso de tecnologias em atividades pedagógicas, em % de professores de escolas urbanas do Brasil. Fonte: CGI.BR/NIC.BR, Centro Regional de Estudos para o Desenvolvimento da Sociedade da Informação (Cetic.br) - 2021

que para um uso produtivo destes equipamentos, uma instituição de ensino tenha também investimentos para capacitação da equipe docente e manutenção de equipamentos.

A Figura 1.1 representa uma pesquisa publicada pelo Cetic.br em 2020, com dados referentes ao ano de 2019, e gera um alerta para a realidade dessas instituições de ensino, que apesar de terem tido investimento inicial para aquisição de equipamentos, não aplicam políticas de aproveitamento a longo prazo para as máquinas adquiridas. Entre os principais pontos levantados pelos professores de escolas urbanas entrevistados para a base de dados do estudo do Cetic.br, estão a falta de treinamento do corpo docente para uso dos equipamentos, e falta de apoio pedagógico para uso dos computadores durante os cursos ministrados, levantando o questionamento do real uso desses equipamentos, que deveriam ser instrumentos imprescindíveis para a familiarização dos alunos com uso de tecnologias, e treinamento de habilidades essenciais para o mercado de trabalho moderno.

Instituições de ensino públicas possuem o objetivo de disponibilizar um ensino de qualidade para as famílias de sua comunidade, independente de sua situação financeira. Tendo em vista a presença cada vez mais constante da tecnologia no cotidiano da sociedade moderna, poderia ser parte desse ensino de qualidade o contato e a prática do uso destes equipamentos, especialmente por um número significativo dessas instituições já possuírem uma infraestrutura funcional de equipamentos que possuem um alto custo de investimento, muitas vezes fora do alcance financeiro de famílias com menor poder aquisitivo.

Consolidado o fato da necessidade de melhor preparo do corpo docente para uso de tecnologias em sala de aula, e de apoio pedagógico para tal, se tem como desafio concreto algum tipo que ação que facilite esse preparo da equipe docente, assim como estimular o uso de equipamentos tecnológicos já presentes nas instituições de ensino brasileiras. Para tal, este estudo propõe uma forma de organização para desenvolvimento de atividades a serem realizadas com uso de computadores em escolas do ensino fundamental e médio. Este estudo objetiva expor uma forma de padronizar a concepção e aplicação de cursos, programas e aplicações utilizadas para o mesmo, exemplificando com uma proposta de curso de baixo custo de implementação para familiarização inicial de alunos do ensino fundamental e médio para programação de computadores utilizando linguagem de programação Java. Essa formalização sistemática do conteúdo a ser lecionado será peça-chave para a proposta de disseminação de informação entre professores, que possibilitará a alteração, reuso e reprodução de conteúdos de ensino criados seguindo o guia proposto.

## **1.3 Problema de Pesquisa**

Este estudo busca mostrar os benefícios para o docente e sua instituição do investimento na padronização formal de objetos de ensino em formato que permita interpretação por equipamentos computacionais. Esta iniciativa, permite benefícios para o corpo docente, permitindo registro e armazenamento de seus objetos desenvolvidos por seus esforços na instituição, evitando retrabalho por permitir o compartilhamento, modificação e reuso desses objetos pelo corpo docente, e novos funcionários que venham fazer parte da equipe de professores. Para a instituição, os benefícios vêm na forma da capacidade de se criar um conteúdo legado de cursos, aulas e atividades de ensino, podendo estes serem analisados para relatórios buscando a melhoria da qualidade do ensino aplicado e a maior eficiência do processo de aprendizagem dos alunos.

## **1.4 Objetivos**

### **1.4.1 Objetivo Geral**

O presente trabalho tem como objetivo geral contribuir para a prática de compartilhamento e reutilização de material didático que venha a ser desenvolvido em uma instituição de ensino pelo seu corpo docente, visando o impacto positivo na qualidade do ensino e da sua consequente aprendizagem.

## 1.4.2 Objetivos Específicos

### **Objetivo Específico 1: Prover um Guia de Modelagem de Curso em Padrão IMS *Learning Design***

Esse guia objetiva preparar instrutores para o registro, na forma de Unidades de Aprendizagem, de esforços e iniciativas educacionais aplicados em cursos, aulas e atividades pelo corpo docente de uma instituição de ensino, possibilitando também o compartilhamento, reprodução e modificação destas Unidades de Aprendizagem.

Espera-se com este estudo, que o guia sirva como motivador para que instituições de ensino formalizem o conteúdo ministrado por seus professores para identificação de boas práticas, atividades e metodologias de sucesso aplicadas a alunos, para que estas sejam reproduzidas e reaproveitadas, além de possibilitar desenvolvimento de cursos dinâmicos, onde caso ocorra de um aluno não apresentar rendimento desejável, ou tenha dificuldade em conteúdos específicos ao longo do curso ministrado, este curso possa se adaptar, e sugerir novas atividades, ou alterar o contexto da atividade ou metodologia utilizada, para melhor desempenho do aluno em questão.

### **Objetivo Específico 2: Prover uma Proposta de Curso Introdutório de Programação**

Este estudo objetiva apresentar um material didático e curso que possa ser aplicado em escolas dos mais variados perfis, com equipamentos de baixo desempenho de processamento, para estimular as escolas a iniciarem a prática de programação em seus alunos, para que estes tenham familiarização com tecnologia e atividades computacionais voltadas ao estímulo do raciocínio lógico e a resolução de problemas.

O curso tem como propósito também possibilitar aos alunos de ensino fundamental e médio um primeiro contato com a atividade de programação, antes que adentrem um curso do ensino superior, visando familiarizar o aluno com práticas e conceitos estudados nas disciplinas introdutórias de algoritmos e introdução a programação. Espera-se que esse contato com programação reduza a abstração e a curva de aprendizagem inicial dos cursos de TI.

Além de ser um curso preparado para uso direto em instituições de ensino, o curso também serve como modo de ilustrar a utilização do guia de modelagem em *IMS Learning Design (IMS LD)*.

## 1.5 Metodologia

Por este estudo ter a forma de proposta de modo de formalização de atividades educacionais com exemplo de curso aplicável, buscando benefícios ao docente e sua instituição, caracteriza-se como pesquisa aplicada. Sua abordagem é qualitativa, por não necessitar de métodos estatísticos para sua interpretação ou conclusão e meio investigativo bibliográfico, por se basear em publicações, obras e anais de eventos especializados no tema abordado[4].

## 1.6 Estrutura deste Documento

Este trabalho tem sua estrutura composta pelos seguintes elementos:

Capítulo 2, referente à Revisão de Literatura, onde serão apresentadas iniciativas voltadas para o ensino específico de programação, seja por metodologias propostas ou por tecnologias a serem utilizadas para este fim, por exemplo: Métodos de ensino como Gamificação, formas de quebra de abstração e linguagens de programação simplificadas voltadas para objetivos educativos. Serão tratadas também tecnologias propostas para auxílio do corpo docente como um todo, independente de disciplina; ferramentas utilizadas em ambiente escolar, como Moodle, assim como ferramentas usadas para planejamento de cursos.

No Capítulo 3, será tratado o referencial teórico, com os principais conceitos necessários para entendimento e desenvolvimento da proposta sugerida por este estudo. Justificativas acerca das decisões tomadas para os artefatos desenvolvidos nesta análise, além dos conceitos de Movimento Maker, sua aplicação na educação e como motivador de uma das atividades propostas neste estudo; os conceitos de *Learning Design*, Unidades de Aprendizagem, e o padrão IMS LD, além de ferramentas existentes para a modelagem educacional de cursos.

O Capítulo 4, contendo a proposta deste estudo, constituída por duas sugestões, motivadas pelos problemas apresentados previamente: Um guia, voltado para o indivíduo interessado em desenvolver um curso formalizado nos moldes do *Learning Design*, independente de disciplina a ser ministrada; E um exemplo de um curso introdutório de programação formalizado em *Learning Design*, direcionado para facilitar o instrutor que queira implementar um curso de programação em uma instituição de ensino fundamental ou médio da rede pública.<sup>4</sup>

---

<sup>4</sup>Todos os artefatos gerados por este estudo, se encontram no endereço <https://drive.google.com/drive/folders/1qtd8k0UW0uen1Kc90uM8DZbImQL1zwdt?usp=share> e também no repositório Lourdes Brasil da Universidade de Brasília em <http://quizwiki.cic.unb.br/lmb> acessados em 28/05/2021. O plano de curso desenvolvido, juntamente com o guia para padronização em IMS LD se encontram na seção de apêndices ao final do documento.

Por fim, no Capítulo 5, temos a conclusão deste trabalho, juntamente com considerações finais e sugestões de trabalhos futuros para continuidade do estudo proposto neste documento.

# Capítulo 2

## Revisão de Literatura

Para o exemplo de atividades previsto neste estudo, foi escolhido como tema a ser lecionado aulas introdutórias de programação. Este tema foi escolhido por três motivos: Para servir como um ponto inicial para motivar instrutores que tenham interesse em iniciar atividades voltadas para programação de computadores; para possibilitar que alunos de instituições de ensino tenham um primeiro contato com a prática de codificação antes que ingressem em um curso técnico ou de ensino superior; e para estimular as instituições de ensino a fazer de seus equipamentos de informática.

Apesar da proposta de aulas sobre programação, o método utilizado para a formalização das atividades propostas não é restrito a esse tema, tal qual métodos pedagógicos, pode ser aplicado independente da disciplina ministrada. É uma forma de registrar atividades ministradas, e facilitar sua reutilização, modificação e compartilhamento entre professores. Como a área de ensino de programação é um campo de estudo e ensino mais recente que as disciplinas tradicionais lecionadas em escolas (matemática, português, história, dentre outras), não usufrui de extensa literatura e grande quantidade de métodos de ensino específicos, senão algumas iniciativas e a aplicação de conceitos pedagógicos independentes de conteúdo. Para melhor análise da proposta deste estudo, é essencial conhecermos algumas iniciativas na área de ensino de programação, que servirão também de inspiração para as atividades que irão compor o curso-exemplo.

### **2.1 Iniciativas recentes em produção de material didático e ensino de programação**

Material didático remete a todo livro e artefato desenvolvido para ser usado por professores durante o processo de ensino. No tópico do ensino específico de programação, existem algumas iniciativas acerca de métodos ou tecnologias a serem aplicadas visando melhor

qualidade da aprendizagem e otimização do processo de construção de conhecimentos computacionais pelos alunos.

Em [5], o autor cita algumas inovações neste campo sugeridas por Edsger W. Dijkstra, Tony Hoare, e Niklaus Wirth, datadas no fim dos anos 60 e início dos anos 70, onde já propunham princípios de uso de linguagens de programação simplificadas e elegantes, uso de metodologias para focar o ensino do aluno no entendimento de programação, e não apenas replicação de memorização de códigos, e a ênfase no aprendizado da motivação e raciocínio usado na criação de algoritmos. Sobre a replicação de códigos, essa é uma abordagem muito usada em livros didáticos para iniciantes, onde exibem códigos prontos para o leitor, e espera-se que este seja capaz de criar códigos semelhantes, sem a preocupação no entendimento do processo e das decisões tomadas para construção daquela solução.

Algumas propostas e metodologias de ensino que se apoiam em bases pedagógicas opostas serão comentadas a seguir, classificadas neste estudo da seguinte forma:

- Linguagens de Programação, Bibliotecas de Classes e Métodos, ou *Software* Educacional;
- Metodologias de ensino.

Atualmente, não existe um consenso sobre qual a forma mais eficiente para aprendizagem de programação para leigos, qual o melhor software ou a linguagem de programação mais indicada para o primeiro contato de pessoas que iniciam os estudos em programação[6][7]. No campo de Linguagens de Programação, Bibliotecas de Classes e Métodos, ou *Software* Educacional, temos alguns destaques de iniciativas relevantes e/ou aplicadas atualmente:

- Logo<sup>1</sup>: Uma das linguagens de programação precursoras para ensino de programação por meio de codificação de gráficos vetoriais. Desenvolvida em 1967, deu origem a várias outras linguagens de programação educacionais derivadas, muitas vezes referidas como *turtle* ou *turtle graphics*, por usarem um personagem na tela de uma tartaruga como ponteiro para o marcador que grava as imagens na tela durante a execução dos códigos;
- Karel<sup>2</sup> e JKarel<sup>3</sup>: A linguagem de programação Karel the Robot, desenvolvida em 1981, propunha ser uma linguagem introdutória de ensino de programação utilizando a programação de robôs existentes em plano cartesiano de coordenadas. O Jkarel,

---

<sup>1</sup>LOGO Foundation - Acessado em 14/05/2021 - [https://el.media.mit.edu/logo-foundation/what\\_is\\_logo/logo\\_programming.html](https://el.media.mit.edu/logo-foundation/what_is_logo/logo_programming.html)

<sup>2</sup>Karel IDE - Acessado em 14/05/2021 - <https://stanford.edu/~cpiech/karel/ide.html>

<sup>3</sup>JKarel - Acessado em 14/05/2021 - <http://csis.pace.edu/~bergin/KarelJava2ed/karelexperimental.html>

uma reimplementação do Karel em linguagem Java, foi a ferramenta escolhida para ser utilizada neste estudo;

- Linguagens de Programação Baseada em Blocos: Blockly (Biblioteca em Javascript para editores de programação visual de autoria da Google)<sup>4</sup>, Scratch<sup>5</sup>, são linguagens que utilizam instruções na forma de blocos codificados por cores e de diferentes tamanhos, arrastáveis pelo usuário para criação de algoritmos executáveis. O Code.org<sup>6</sup>, Microsoft MakeCode<sup>7</sup> e AppInventor<sup>8</sup> são ambientes de programação online baseados em Blockly;
- Software de Desenvolvimento de Jogos Eletrônicos: Softwares como o Construct<sup>9</sup>, desenvolvida pela Scirra, ou como o Clickteam Fusion<sup>10</sup>, possuem programação simplificada de forma visual e orientada a eventos, sem necessidade de digitação de código por parte do usuário, buscam proporcionar aos usuários resultados mais rápidos e baixa curva de aprendizagem inicial.

O mesmo vale para metodologias. Existem várias propostas para facilitar o entendimento de novos aprendizes de programação, sendo algumas das mais relevantes atualmente:

- Atividades Desplugadas: Atividades práticas de fácil compartilhamento que exercitam conceitos relacionados à solução de problemas computacionais, sem o uso de computadores, permitindo que alunos realizem atividades envolvendo algoritmos sem a necessidade de conhecimentos prévios de programação[8];
- Metodologia visual: Propõe o uso de artefatos e mídias para a reduzir a abstração dos conceitos fundamentais de programação[9];
- Gamificação: Uso de características inerentes ao design de jogos jogos, como estética, organização das aulas e atividades por níveis, monetização de atividades por pontos ou moeda sem valor comercial, desbloqueio de atividades e regalias pelo aluno para estimular seu interesse durante as aulas[10];
- Sala de Aula Invertida: Metodologia de ensino ativa, baseada no protagonismo do aluno no processo de aprendizagem, orientado e avaliado por tutor. Apresentado um

---

<sup>4</sup>Blockly - Acessado em 14/05/2021 - <https://developers.google.com/blockly>

<sup>5</sup>Scratch - Acessado em 14/05/2021 - <https://scratch.mit.edu/>

<sup>6</sup>Code.org - Acessado em 14/05/2021 - <https://code.org/>

<sup>7</sup>Microsoft MakeCode - Acessado em 14/05/2021 - <https://www.microsoft.com/pt-br/makecode?rtc=1>

<sup>8</sup>AppInventor - Acessado em 14/05/2021 - <https://appinventor.mit.edu/>

<sup>9</sup>Construct - Acessado em 14/05/2021 - <https://www.construct.net/>

<sup>10</sup>Clickteam Fusion - Acessado em 14/05/2021 - <https://www.clickteam.com/clickteam-fusion-2-5>

tema pelo tutor, os alunos se organizam em grupos e buscam em diferentes fontes de informação por conhecimentos sobre o assunto de forma colaborativa[11];

- *Science, Technology, Engineering, Art and Mathematics (STEAM)*: Metodologia de ensino através de atividades multidisciplinares, envolvendo conceitos de ciências, tecnologia, engenharia, artes e matemática para estimular habilidades de investigação, solução de problemas, estímulo da criatividade e de habilidades ligadas ao preparo dos alunos para situações reais de ambiente de trabalho[12];
- *Problem-Based Learning (PBL)*: Aprendizagem Baseada em Problemas. Em [13] o autor a define como forma de ensino guiado por tutor que tem como característica o empoderamento do aluno, capaz de realizar pesquisas em fontes de informação que não o tutor, e decidir a melhor forma de resolução dos problemas sugeridos, em geral de caráter multidisciplinar. Propõe estimular a curiosidade do aluno como motivador do processo de aprendizagem;
- *Inquiry-Based Learning (IBL)*: Aprendizagem Baseada em Investigação. Apesar de semelhante ao PBL por apresentar um problema a ser solucionado aos alunos, sua principal diferença se encontra no papel do tutor, que também atua como fonte de informação, enquanto no PBL essa responsabilidade é dos alunos[13]. Outro diferencial é por estimular o uso do método científico pelos alunos durante a resolução do problema[14];
- *Project-Based Learning (PJBL)*: Aprendizagem Baseada em Projetos. Tem como objetivo o desenvolvimento de produtos pelos alunos como proposta de solução a problemas propostos. Simula um processo real de atuação no mercado de trabalho e cria um senso de progresso e realização nos alunos durante o ensino[15].

## 2.2 Suporte computacional ao planejamento docente

Em [16], a autora cita algumas iniciativas para amparo do corpo docente, que por meio da tecnologia, visam reforçar a rede de informações do corpo docente, muitas vezes subestimada pela instituição de ensino, e realizada muitas vezes informalmente por iniciativa do próprio corpo docente, mas que é essencial para a qualidade do ensino, uma vez que a experiência individual do professor tem muito a contribuir para a melhoria dos cursos ministrados. São essas iniciativas:

- Rede Internacional Virtual de Educação (RIVED): Repositório de materiais educacionais objetivando o aperfeiçoamento do ensino de Ciências e Matemática no ensino médio;

- Laboratório Didático Virtual (LABVIRT): Repositório de materiais educacionais para as áreas de Física e Química para fortalecimento da troca de conhecimentos entre instituições de ensino;
- Biblioteca digital de Buseti e equipe: Biblioteca digital contendo conhecimento pedagógico, materiais instrucionais e experiências docentes em forma de Objetos de Aprendizagem;
- Ambiente inteligente QSabe: Ferramenta de perguntas e respostas para uso com fins educacionais por docentes;
- Ferramenta Debyte: Ferramenta para comunicação de professores por meio de bate papo e listas de discussão virtuais;
- Computer-integrated classroom (CiC): Ferramenta de quadro virtual, exposição de conteúdos e comunicação a ser usado durante as aulas por alunos e professores;
- Courses as seeds: Ferramenta de comunicação envolvendo alunos e professores para a construção continuada e colaborativa de registros para troca de experiências pelos envolvidos no processo de aprendizagem;
- European Quality Observatory (EQO): Ferramenta para avaliação e recomendação para curadoria de cursos de Ensino Virtual (*e-learning*).
- Site do Núcleo de Suporte Pedagógico para Professores de Língua Estrangeira (NUSP-PLE): Base de dados para artigos, discussões e materiais didáticos para ensino de língua estrangeira.

Grande parte dessas iniciativas buscam armazenar materiais de ensino, e realizar a curadoria destes materiais para facilitar o planejamento docente, porém, é de suma importância não só disponibilizar materiais prontos de ensino, mas também estimular que os professores tenham autonomia para criação de novos conteúdos ou modificação de conteúdos pré-existentes. Para isso, é essencial que o registro de artefatos de ensino e a troca de informações ocorra de forma padronizada, para facilitar o reuso e modificação pelo corpo docente, mas também o processamento computacional destes artefatos educacionais pela instituição de ensino.

Em análise realizada em [17], foram investigadas 9 ferramentas de apoio ao planejamento docente (dentre elas, AulaNet, SimEduc, EnsinarNet, DoceNet, AMADIS e TeleEduc). Essas iniciativas se preocupam em facilitar processos de desenvolvimento de planos de aula, atividades, e processos administrativos como definição de turmas; porém, não agem como um meio de troca destes planos de aula e artefatos de ensino entre os professores da instituição.

A seguir, serão discutidos alguns estudos direcionados para a categorização, listagem ou produção de programas de modelagem de design de aprendizagem, em ordem cronológica do ano de 2002 até 2020, a fim de ilustrar uma linha do tempo de avanços e iniciativas inovadoras acerca do design de aprendizagem:

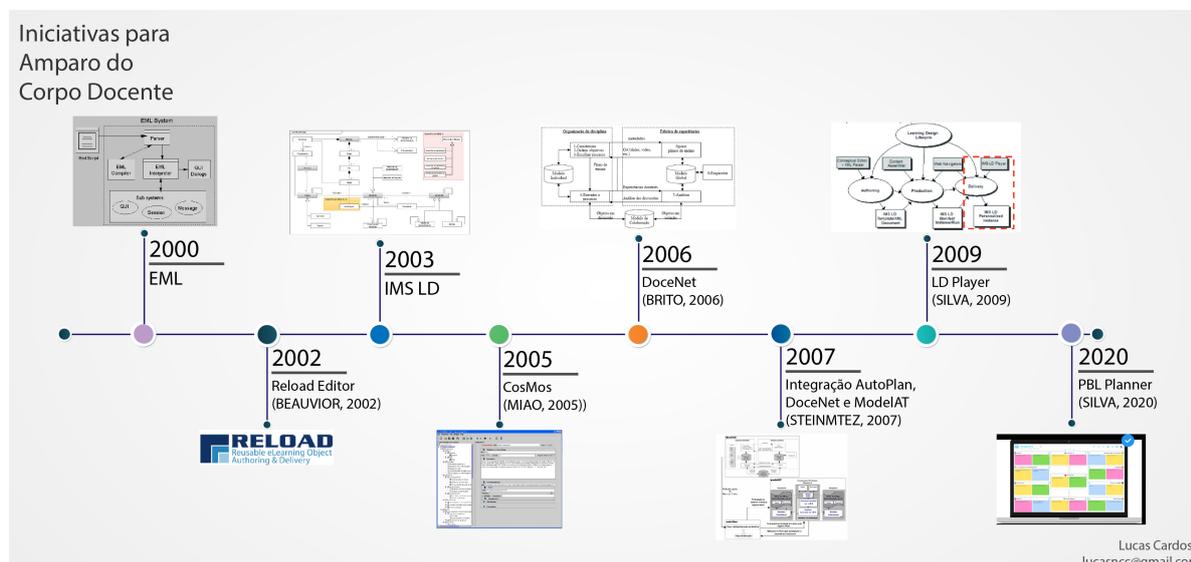


Figura 2.1: Linha do tempo de iniciativas relevantes ao amparo do corpo docente com uso de tecnologia.

Em obra publicada em 2002[18], uma EML é definida como um modelo semântico informacional para descrição de conteúdos e processos que compõem uma unidade de aprendizagem, do ponto de vista pedagógico visando reuso. Ainda nesta obra, o autor em sua análise realiza um levantamento de seis principais linguagens de modelagem educacional: CDF, EML, LMML, PALO, Targeteam e TML/Netquest. Destas linguagens, o EML foi usado como base para o desenvolvimento do padrão IMS LD, desenvolvido em 2003 pelo consórcio IMS.

Em publicação de 2005[19], o autor propõe a plataforma CoSMos[19] para desenvolvimento de Unidades de Aprendizagem - *Units of Learning (UoL)* em padrão IMS LD. Nesse trabalho, o autor classifica as ferramentas de modelagem em categorias relacionadas a forma como exibem visualmente ao usuário o modelo convertido do arquivo de manifesto XML de seu *Learning Design*. Essas ferramentas são classificadas como:

- Ferramentas baseadas em visualização por árvore: Exibem o *LD* como uma árvore relacional dos componentes do *LD* e suas relações, citando como exemplo os editores ALFANET, CopperAuthor, e o editor RELOAD, citado por Miao como referência de implementação de editor de *LD*[19];

- Ferramentas baseadas em visualização por diagrama: Exibem o *LD* em forma de fluxograma utilizando setas e nodos. Exemplos dessa abordagem são MOT+ e ASK-LDT;
- Ferramentas que não seguem o padrão IMS LD, mas que possuem semelhanças em especificação e estrutura. Exemplos citados nessa categoria são o COMPILE e LAMS.

Exceto pelo Reload, que possibilita a modelagem nos níveis A, B e C do IMS LD, a maioria dos editores citados na pesquisa de Miao focam na modelagem em nível A. O *CoSMoS (Collaboration Script Modelling System)* oferece como contribuição a modelagem nos 3 níveis, e melhorias em relação ao *Reload* na forma de exibição e modelagem em nível B.

Em [20], publicado em 2007, o autor propôs a modelagem de atividades de aprendizagem com o uso de jogos. Um detalhe importante desse trabalho foi a escolha do IMS LD ao invés do padrão *SCORM (Sharable Courseware Object Reference Model)*, devido ao IMS LD permitir a modelagem de atividades de aprendizagem com especificando diferentes relações aluno-aluno, aluno-professor e professor-professor, possibilitando dinâmicas diferenciadas.

Em [21], também do ano 2007, é proposta uma classificação de EMLs em pesquisa, determinando tipo de linguagem de acordo com os seguintes atributos:

- *Evaluation Languages* - Linguagens usadas para descrever estágios de aprendizagem que envolvam de modo abstrato resolução de problemas e atividades avaliativas;
- *Content Structuring Languages* - Linguagens de sequenciamento de recursos de aprendizagem, considerando a performance do aluno e suas necessidades para melhoria da experiência de aprendizagem;
- *Activity Languages* - Linguagens focadas em atividades gerais no processo de aprendizagem; É a categoria onde se encontra o IMS LD.

Em publicação de 2008[22], é realizado o desenvolvimento do *LD Player*, ferramenta para modelagem e reprodução de Unidades de Aprendizagem em padrão IMS LD. Essa ferramenta teve como característica inovadora permitir a interação de arquivos IMS LD em ambiente Moodle, já que esse ambiente não possui suporte nativo para o padrão IMS LD, apenas para o padrão SCORM.

Em [23], publicação de 2013, é produzido o CourseEditor, ferramenta de planejamento de curso compatível com IMS LD. Enquanto ferramentas baseadas em formulário e visualização por árvore necessitem que o usuário tenha um entendimento prévio sobre o padrão

IMS LD, o CourseEditor tem como objetivo a acessibilidade por meio visual, facilitando a modelagem por professores sem conhecimentos técnicos.

Em análise exploratória de 2013[24], o autor classificou os editores autorais em função de suas funcionalidades conforme Figura 2.2, e propôs como contribuição para evolução do estado da arte o ambiente *Instructional Model Environment (IME)* para construção, edição, execução e distribuição de pacotes de conteúdo em padrão IMS.

**Quadro 1 - Resumo comparativo dos editores LD.**

Editor IMSLD	GUI	Apresentação	DAD	Nível	Licença	Nível IMSLD	Importa IMSLD	Exporta IMSLD
CADMOS	✓	Diagrama	✓	1	Freeware	A, B	Não	✓
CopperAuthor	✓	Árvore / Tabela	Não	1	Open Source	A	Não	✓
Editor LD LAI	✓	Formulário / Diagrama	✓	1	Open Source	A	Não	✓
LD Authoring Tool	✓	Árvore / Botões / Formulário	✓	1	Open Source	A	Não	✓
MOCOLADE	✓	Diagrama	✓	1	Proprietária	A	Não	✓
MOT+LD	✓	Diagrama	✓	1	Proprietária	A	Não	✓
Pathway ASK-LDT	✓	Diagrama	✓	1	Freeware	A, B	Não	✓
RELOAD	✓	Formulário	✓	1	Open Source	A, B, C	Não	✓
WQE Editor	✓	Botões / Formulário	✓	2	Open Source	A	Não	✓
Recourse	✓	Formulário	✓	3	Open Source	A, B, C	✓	✓
Collage	✓	Diagrama	✓	4	Open Source	A	Não	✓
LAMS	✓	Diagrama	✓	4	Open Source	A	✓	✓
Open GLM Prolix	✓	Diagrama	✓	4	Open Source	A, B	Não	✓
IME	✓	Árvore / Formulário	✓	5	Open Source	A	✓	✓

Figura 2.2: Resumo comparativo dos editores LD. Fonte: JÚNIOR, Aladir F. Silva; SILVA, Leandro Roberto da; FERNANDES, Clovis Torres. Panorama dos Editores de Atividades de Aprendizagem em IMS Learning Design. XXIV Simpósio Brasileiro de Informática na Educação (SBIE 2013). (336-345)..

Em revisão sistemática publicada em 2015[25], o autor concluiu que, apesar de existir quantidade considerável de ferramentas para autoria de modelos IMS LD, a falta de interoperabilidade entre tais ferramentas, mesmo tendo elas em comum o padrão IMS, a complexidade da codificação do processo de modelagem, e problemas de usabilidade se apresentam como obstáculos para o uso em larga escala do IMS LD, mostrando que ainda existe a necessidade para melhorias, inovações e meios de capacitação dos interessados em realizar a modelagem por IMS LD.

Iniciativas mais modernas como o PBL Planner[26] desenvolvido em 2020, propõem a formalização em modelo autoral de iniciativas de ensino especializadas para a metodologia de ensino baseada em problemas.

O grande diferencial do padrão IMS LD sobre padrões como SCORM, utilizado em ambientes Moodle, e outras das soluções discutidas, é a adaptabilidade do IMS LD para qualquer metodologia de ensino desejado, além de comportar as mais variadas situações de ensino, como ensino a distância, presencial e híbrido, além de permitir atividades em grupos entre alunos, e que alunos desempenhem diferentes papéis nas atividades propostas.

Vale ressaltar também, que o consórcio IMS tem como proposta diversos padrões para as mais variadas necessidades, sendo o IMS LD e o IMS *Content Package* duas dos muitos padrões sugeridos. Existem padrões desenvolvidos como o IMS LTI, atualizado em 2019, que evidenciam que o consórcio IMS continua relevante e em constante atualização. O consórcio IMS pretende viabilizar a criação de sistemas interoperáveis através de seus diversos padrões, comportando os mais variados artefatos de modo que seja possível a comunicação entre eles.

# Capítulo 3

## Referencial Teórico

Este capítulo visa apresentar alguns conceitos fundamentais utilizados neste estudo. Serão discutidos o Movimento Maker, utilizado como inspiração para a proposta desse estudo, e também serão discutidos os conceitos de *Learning Design*, linguagens de modelagem educacional e o padrão IMS LD do consórcio educacional *IMS Global*, além das escolhas selecionadas para desenvolvimento da proposta.

### 3.1 Das escolhas para o curso proposto

Para o curso proposto neste estudo, foi selecionada a linguagem de programação Java, por sua característica de ser executada em máquina virtual, facilitando sua portabilidade, programação, compilação e execução em computadores das mais variadas especificações, além de ser uma linguagem de programação largamente usada em ambientes acadêmicos e pela indústria de TI[27]. Outro motivo responsável pela escolha do Java como primeiro contato dos alunos com a atividade de programar foi ser uma linguagem de programação tipada, isto é, que exige no ato de codificação que se explicita o tipo de cada variável declarada. Isso é importante para reforçar nos alunos os conceitos fundamentais de variáveis, e através da prática consolidar boas práticas de programação. Serão aplicados alguns métodos para redução da abstração de conceitos de programação, como atividades desplugadas por seu caráter tátil, e uso de biblioteca de programação visual para renderização visual do código produzido durante as atividades iniciais. Foram usados como referência bibliográfica das atividades sugeridas, os seguintes livros:

- [28] DEITEL, Paul; DEITEL, Harvey: Java: como programar. Pearson Prentice Hall, 2010
- [1] BERGIN, Joseph et al.: Karel J Robot: a gentle introduction to the art of object-oriented programming in java. Dream Songs Press 2005

- [29] SIERRA, Kathy; BERT, Bates: Use a cabeça! Java. Alta Books, 2009
- [30] ASCENCIO, Ana Fernanda Gomes, CAMPOS, Edilene Aparecida Veneruchi de: Fundamentos da programação de computadores. Pearson Prentice Hall, 2007

As obras [28], [29] e [30] são obras consolidadas no ensino de programação em cursos de ensino superior, enquanto a obra [1], foca na utilização da biblioteca de JKarel, um conjunto de métodos e funções que simulam na tela do usuário, ao executar seu código compilado, um plano cartesiano e robôs programáveis. Utilizando o JKarel, alunos podem programar o comportamento de um ou mais robôs, simulados virtualmente pelo computador, que respondem a sequências de comandos simples implementados por códigos em Java. A escolha da obra [1] é essencial para que seja possível apresentar de forma visual a execução sequencial de instruções para o aluno, visando a citada quebra de abstração, uma vez que o estímulo visual do robô executando cada instrução programada sequencialmente, irá facilitar aos alunos a identificação de erros em seus algoritmos, e também a prática e consolidação dos conceitos fundamentais de variáveis, laços de repetição, métodos/funções e estruturas condicionais de código.

Um outro artefato que foi utilizado nesse curso, é a atividade desplugada, motivada pelas características do Movimento Maker, na forma de um jogo de tabuleiro que irá solidificar a prática realizada em computador pelos alunos, agora em um contexto de brincadeira competitiva. A atividade desplugada também serve de exemplo ao tutor de que podemos exercitar a programação mesmo que aconteça imprevistos relacionados à disponibilidade de computadores para uso dos alunos. O caráter tátil e visual da atividade desplugada disponibiliza mais um meio de aprendizagem para os alunos. Outra decisão tomada para este curso, foi a de inserir uma atividade final, inspirada no *Project-Based Learning (PJBL)*, onde os alunos deverão desenvolver um produto na forma de um jogo de narrativa interativa, com o propósito de unir em uma atividade todo o conteúdo estudado nas aulas anteriores, além de familiarizar os alunos com a modelagem de fluxograma para algoritmos e apresentações curtas de produtos simulando situação de venda (*Sales Pitch*).

Dado o interesse em aumentar o contato de alunos na atividade de programação, antes que estes iniciem os estudos em cursos do ensino superior, o exemplo de curso proposto têm como público-alvo alunos do ensino fundamental e médio de escolas públicas do território brasileiro, podendo ser aplicado também em escolas particulares e ou no formato de curso livre. Como pré-requisito para participar do curso, o aluno deve estar alfabetizado, com fluência em leitura e escrita em português; São habilidades desejáveis para o aluno que participe do curso proposto: Que o aluno tenha entendimento e familiarização com a língua inglesa; Que ele tenha tido contato prévio com computadores, mesmo que básico.

É importante ressaltar também que, mesmo possuindo uma atividade desplugada, essa agem como complemento das atividades de programação a serem realizadas em com-

putadores, sendo necessário um laboratório de informática, preferencialmente com um computador disponível para cada aluno participante do curso. Por objetivar escolas do segmento público, que muitas vezes possuem laboratórios de informática funcionais, mas com pouco uso, e com máquinas de baixa capacidade de processamento, todos os programas e atividades foram selecionadas para não necessitarem de computadores de alto desempenho, e em diferentes sistemas operacionais.

## 3.2 Movimento *Maker*

Movimento *Maker* é o termo utilizado para referenciar de forma abrangente grupos de indivíduos que constroem produtos (físicos ou digitais) de forma colaborativa, realizando troca de conhecimentos de forma presencial ou através de meios virtuais, como redes sociais ou fóruns[31]. O Movimento *Maker* é baseado no *Do It Yourself (DIY)*, forma de produção artística caseira e de baixo custo, com o uso de materiais e ferramentas de fácil acesso ao cidadão comum. O *DIY* teve seu início no fim da década de 70 como subcultura do movimento musical *punk*[32], que também enfatizava o caráter único da criação artesanal. O principal diferencial do Movimento *Maker* se comparado ao *DIY* é a forte presença da tecnologia durante o processo de construção, podendo ser considerado como um avanço lógico do *DIY*. Isso foi possível pelo crescimento da acessibilidade da tecnologia como parte fundamental no cotidiano moderno, dado pela produção em larga escala de dispositivos tecnológicos e redução nos custos de produção gerando uma baixa no preço de compra para o cliente.

No Manifesto *Maker*[33], o autor cita nove princípios fundamentais do Movimento *Maker*:

1. Construir (*Make*): O ato de criar age como forma de expressão humana fundamental;
2. Compartilhar (*Share*): Divulgar o produto criado e o conhecimento atrelado à sua criação;
3. Doar (*Give*): O ato de dar a outra pessoa um produto criado estimula a prática do desapego e sentimento de sociedade;
4. Aprender (*Learn*): Deve-se sempre buscar conhecimentos novos, relacionados ao ato de construir;
5. Equipar (*Tool Up*): É importante ter acesso às ferramentas necessárias para a construção do produto visado;
6. Brincar (*Play*): Um indivíduo deve se divertir durante o processo de criação e celebrar seu aprendizado;

7. Participar (*Participate*): Comunique com a comunidade e faça parte de grupos de atividades *maker*;
8. Apoiar (*Support*): O crescimento do movimento depende do apoio mútuo de seus membros em diversas áreas;
9. Mudar (*Change*): Seja receptivo à mudanças.

Observando estes princípios, é possível perceber a influência do *Do It Yourself* na cultura *Maker*, e o autor em [33] é ainda mais incisivo nesse aspecto ao descrever em seu Manifesto *Maker* que até mesmo o texto escrito de sua publicação é passível de ser modificado e utilizado por outras pessoas. Como característica primordial do Movimento *Maker* podemos destacar o estímulo ao ato de compartilhar entre seus participantes, estimulando o trabalho comunitário e o sentimento de um indivíduo em pertencer a um grupo com um objetivo comum, visando solucionar problemas que fazem parte da sociedade moderna. Desenvolver produtos para uso próprio não é o único objetivo de projetos *Maker*; Muitas vezes a solução de problemas relevantes à sociedade moderna surge como meta em projetos *Maker*, como reciclagem, sustentabilidade e preservação do meio ambiente. Todo este pensamento de criação acessível e colaborativa fez surgir diferentes formas de atuação *Maker*, como os *Makerspaces* e os *FabLabs*.

*Makerspaces* são caracterizados como comunidades para práticas relacionadas ao Movimento *Maker* em um espaço físico, mas não limitados à prática de criação, podendo o espaço ser utilizado para feiras e eventos[31].

*Fabrication Laboratory (FabLab)* é um espaço físico de pequeno porte que reúne ferramentas que possibilitam práticas *Maker*. Inicialmente desenvolvido por uma iniciativa do *Massachusetts Institute of Technology (MIT)*, foi criado de acordo com o princípio de que, caso tenha acesso às ferramentas corretas, um homem comum pode criar produtos extraordinários.<sup>1</sup>. Existem também *FabLabs* móveis, que possibilitam a mobilidade do laboratório de fabricação e sua integração a outros espaços físicos.

### 3.2.1 Movimento *Maker* no Brasil

Podemos observar uma atuação crescente do Movimento *Maker* no Brasil, principalmente na forma de projetos de *FabLabs* para escolas, por seu baixo investimento financeiro inicial se comparado ao *Makerspace*, com objetivo de estimular a participação de jovens estudantes na prática de atividades *Maker*.

Uma iniciativa nacional de *FabLab* é o *Lite Maker*[34], *FabLab* nacional móvel e de baixo custo para uso em salas de aula desenvolvido pela Universidade do Vale de Itajaí.

---

<sup>1</sup><https://www.media.mit.edu/projects/fab-labs/overview/> (Acessado em: 09/09/2020)

O *Lite Maker* propõe tornar acessível as ferramentas tecnológicas para uso de alunos em instituições de ensino da iniciativa pública e privada, na forma de um armário móvel que quando aberto se torna uma estação de trabalho *Maker*, completo com bancada, ferramentas, computador e impressora 3d. Um grande atrativo de um *FabLab* móvel é a possibilidade de tornar qualquer espaço, incluindo uma sala de aula regular, em um espaço de atividades *Maker*.

Figura 3.1: *LiteMaker*: Laboratório *Maker* móvel para uso em instituições de ensino. Fonte: <http://lite.acad.univali.br/pt/lite-maker/> Acessado em: 28/05/2021

(a) *LiteMaker* fechado.



(b) *LiteMaker* aberto.



### 3.2.2 Produtos e Tecnologias *Maker* no Mercado

Com a popularização dos Movimentos *Maker* e *DIY*, algumas empresas surgiram oferecendo produtos especializados em atividades *Maker*, buscando a inovação através da simplificação de uso de tecnologias já existentes, desenvolvimento de produtos para estimular a produção *Maker* em crianças e adolescentes, ou unindo diferentes tecnologias para maior comodidade do usuário.

Alguns produtos que podemos identificar atualmente como expoentes na categoria *Maker* são:

- Arduino<sup>2</sup> - Microcontrolador de baixo custo utilizado para prototipação e desenvolvimento de circuitos elétricos envolvendo sensores, *displays* e *LEDs*. Muito utilizados em projetos de automação em casas inteligentes, para controle de equipamentos elétricos, consumo de energia e água.

<sup>2</sup>Arduino - Acessado em 14/05/2021 - <https://www.arduino.cc/>

- *Raspberry Pi*<sup>3</sup> - Microcomputador utilizado para criação de dispositivos eletrônicos e servidores por um preço acessível .
- *Mindstorms*<sup>4</sup> - Criado pela Lego, empresa de renome no ramo de brinquedos para crianças e jovens, é um kit para criação de robôs e automóveis em pequena escala, programáveis por linguagem de programação visual baseada em blocos em *Software* próprio.
- *LittleBits*<sup>5</sup> - Kits simplificados de criação de circuitos elétricos para crianças, através de placas com circuitos e componentes elétricos conectados por ímãs, sem necessidade de fios ou perigo do usuário sofrer choques por descarga elétrica. Possibilita a criação de produtos por meio do uso de baterias, motores, sensores e *LEDs* de forma acessível para crianças.
- *SAM Labs*<sup>6</sup> - Bloco de componentes eletrônicos formados por sensores, motores, *LEDs* e *buzzers*, conectados por *Bluetooth* em computador ou *tablet* para programação de comportamentos por linguagem de programação visual simplificada.

Dos produtos listados, o *Mindstorms*, *LittleBits* e *SAM Labs* possuem como público alvo crianças e adolescentes, sendo produtos de entrada para facilitar o primeiro contato de jovens com atividades *Maker*. Já o Arduino e *Raspberry Pi* não focam em simplificar o uso, mas sim em tornar mais acessível a tecnologia.

### 3.2.3 Movimento *Maker* na Educação

É de suma importância, após termos analisado o que é o Movimento *Maker* e seus princípios, que essa filosofia sirva para impulsionar o profissional que deseja desenvolver um material didático ou curso a ser ministrado em sala de aula. Dado o desafio do ensino de desenvolver os alunos de acordo com suas individualidades, ao mesmo tempo que o processo de aprendizagem ocorrem na maioria de suas vezes em círculos e espaços sociais, é essencial que este profissional visualize fontes de informações, livros didáticos e materiais de ensino não como algo finalizado e imutável, capaz de ser aplicado em qualquer aluno ou turma que seja submetida à aula ministrada. Devem ser consideradas sim, como ferramentas, que possibilitam ao profissional fazer o uso dos mais diversos materiais e metodologias, sempre adaptando seu conteúdo e aulas ao seu público alvo: seus alunos.

Baseando-se na proposta do Movimento *Maker*, toda ferramenta pode ser usada, manipulada, transformada ou aproveitada, em parte ou em seu todo, para uso em sala de

<sup>3</sup>*Raspberry Pi* - Acessado em 14/05/2021 - <https://www.raspberrypi.org/>

<sup>4</sup>*Mindstorms* - Acessado em 14/05/2021 - <https://www.lego.com/pt-br/themes/mindstorms>

<sup>5</sup>*LittleBits* - Acessado em 14/05/2021 - [https://sphero.com/collections/all/family\\_littlebits](https://sphero.com/collections/all/family_littlebits)

<sup>6</sup>*SAM Labs* - Acessado em 14/05/2021 - <https://samlabs.com/us/>

aula. Não que isso impossibilite um profissional de desenvolver cursos e materiais, uma vez que eles serão obsoletos uma vez que mude de alunos; muito pelo contrário. O material tem que ser visto não como algo finalizado, mas sim como algo em constante desenvolvimento, sempre aberto para modificações, atualizações e experimentos. O material por si é uma base, um fundamento onde o curso a ser ministrado irá se basear. Porém, é necessário que o profissional que ministra as aulas considere também seu público, seus alunos. A opinião dos alunos é fundamental para o direcionamento das aulas e para a melhor aplicação de um curso proposto. As atividades e conteúdos têm de fazer sentido para os alunos, e muitas vezes bastam pequenas alterações nas atividades propostas ou no modo de apresentação de conteúdos para que o curso se adapte ao contexto de uma turma nova de alunos.

Este estudo propõe a aplicação dos nove princípios fundamentais do Movimento *Maker* em um contexto educacional de seguinte modo:

1. Construir (*Make*): Preparar o profissional que se interesse na área de ensino de programação a criar seu próprio curso ou atividades a serem ministradas em sala de aula;
2. Compartilhar (*Share*): Estimular a divulgação de materiais desenvolvidos, visando solidificar uma rede de conhecimento e comunicação entre educadores da área de Tecnologia da Informação;
3. Doar (*Give*): Iremos considerar aqui a aplicação do curso, onde os alunos irão receber em contexto de aula o curso desenvolvido;
4. Aprender (*Learn*): Lecionar é um constante aprendizado, especialmente em uma área como a da tecnologia, onde avanços e novas ferramentas são constantes. O educador deve se colocar constantemente em posição de aprendizagem, e sempre considerar as respostas e reações de seus alunos acerca de seu método de ensino, e das atividades propostas;
5. Equipar (*Tool Up*): Iremos considerar como ferramentas: propostas de atividades, livros didáticos, metodologias de ensino e tecnologias utilizadas em aula. As ferramentas serão essenciais para moldar o curso proposto para melhor se adequar à turma que será ministrado;
6. Brincar (*Play*): O processo de aprendizado deve ser prazeroso não só para os alunos, mas é importante que o seja também para o educador;
7. Participar (*Participate*): A troca de atividades, experiências e informações é essencial para profissionais do ensino, não seria diferente para contexto de tecnologias;

8. Apoiar (*Support*): Aqui, dado o ensino de programação e tecnologias emergente em nossa sociedade, só será solidificado como um movimento de ensino de qualidade, se os educadores se considerarem parte de um todo, de um movimento de ensino de Tecnologia da Informação, dispostos a trabalhar em equipe, e em conjunto com outras áreas, como a Pedagogia;
9. Mudar (*Change*): Assim como o curso ministrado deve estar sujeito a mudanças, a metodologia utilizada, e forma de lecionar também devem estar. Se os alunos não estiverem engajados, é preciso que haja experimentação de novas abordagens, metodologias e atividades. Para isso, o educador deve estar disposto a ouvir seus alunos, e não ter receio de aceitar sugestões. Um educador experiente tende a desenvolver um estilo de docência, munido de uma metodologia de ensino que melhor se adapte ao seu modo de ensino. Este não é o problema; O erro está em considerar sua aula como infalível, sem espaço para melhorias. O professor pode e deve aplicar a aula de acordo com suas individualidades e preferências, porém, uma vez que essa aula não se mostre eficiente se aplicada em uma turma, esse educador deve estar preparado para se desafiar e modificar suas aulas para melhor servir tal turma.

### **3.3 *Learning Design* e Linguagens de Modelagem Educacional**

*Learning Design (LD)* é o termo utilizado para o processo de desenvolvimento de artefatos de aprendizagem, e também o próprio produto resultante deste processo[35]. Linguagem de Modelagem Educacional, ou *Educational Modelling Language (EML)*, é um termo com dupla conotação: Representa tanto um termo genérico para todas as linguagens desenvolvida para a criação e formalização de unidades de ensino, como é o nome de umas das *EMLs* precursoras. A primeira conotação, mais abrangente, será utilizada com maior frequência nesta análise, mas também será observada a linguagem de modelagem educacional homônima. Em [18], o autor define uma *EML* também pelo objetivo de facilitar a reaproveitamento e suporte não só de unidades de ensino, mas de seus componentes, como artefatos digitais, metodologias de ensino e dinâmicas desenvolvidas. Essa facilidade de manutenção, distribuição e reaproveitamento de unidades de aprendizagem é essencial para a proposta deste estudo, de formalização e preservação dos esforços que os tutores exercem ao criar seus planejamentos de curso.

A linguagem de modelagem educacional *EML*, publicada pela Universidade Aberta dos Países Baixos (*Open University of the Netherlands*) em Dezembro do ano 2000, teve seu desenvolvimento focado principalmente na criação de conteúdos para o modelo remoto

de ensino (*e-learning*). Visava construir cursos, que uma vez formalizados em formato *Extensible Markup Language (XML)*, escolhida para exercer um comportamento próximo ao de arquivos *HTML* e navegadores de internet: um editor cria arquivos no formato da linguagem *EML*, que por sua vez podem ser interpretados por programas leitores de *EML*, possibilitando a execução de unidades de ensino de modo automatizado para cursos de ensino remoto[36].

### 3.4 Padrão IMS LD

Desenvolvido à partir da linguagem de modelagem educacional *EML* pelo consórcio IMS em 2003, o padrão IMS LD é um conjunto de normas para modelagem em formato computacional de unidades de ensino, podendo estas serem atividades, aulas ou cursos, na forma de atos realizados por indivíduos que podem exercer diferentes papéis durante o processo de aprendizagem, suportando diferentes abordagens pedagógicas, independente de disciplina ou área de ensino, e recursos computacionais diversos[37].

O IMS LD é composto por quatro documentos: Modelo conceitual com seus elementos básicos, Modelo informacional com as definições formais de seus elementos e atributos, Modelo representativo dos seus elementos XML, e Guia de Exercícios e boas práticas de implementação com exemplos[23].

O padrão IMS LD têm em sua terminologia, usada para representar experiências de ensino, analogias ao contexto de uma peça de teatro, onde um curso é representado por uma peça, dividida em atos compostos por atividades, onde os participantes do processo de aprendizagem exercem papéis, podendo exercer diferentes papéis em atividades variadas.

O resultado da modelagem em IMS LD é um arquivo *Extensible Markup Language (XML)* contendo todos os componentes de uma unidade de ensino classificados por *tags*, com chaves de identificação únicas, em formato interpretável por software computacional. Diferente de outros padrões de modelagem educacional, o IMS LD permite atividades em grupos, comportando qualquer metodologia de ensino desejada, uma vez que as descrições dos componentes do *LD* são digitadas durante o processo de modelagem, e não captadas de listas pré existentes. Outra característica importante do IMS LD, é que permite situações de ensino híbrido, presenciais ou de ensino virtual.

Um *Learning Design* em padrão IMS possui 3 níveis para implementação(Figura 3.2)[38]:

- Nível A, composto por todos os componentes básicos de um contexto de ensino, como atividades, papéis a serem exercidos pelos participantes, metodologia de ensino, ambiente de aprendizagem, sequenciamento de tarefas e recursos;

- Nível B, composto pelos elementos do nível A, além de Propriedades, necessárias para armazenar informações sobre papéis durante a execução de um *LD* e Condições, para registro da evolução do cenário didático;
- Nível C, composto pelos elementos dos níveis A e B, além de Notificações, que são mecanismos disparados para iniciar novas atividades baseadas em eventos.

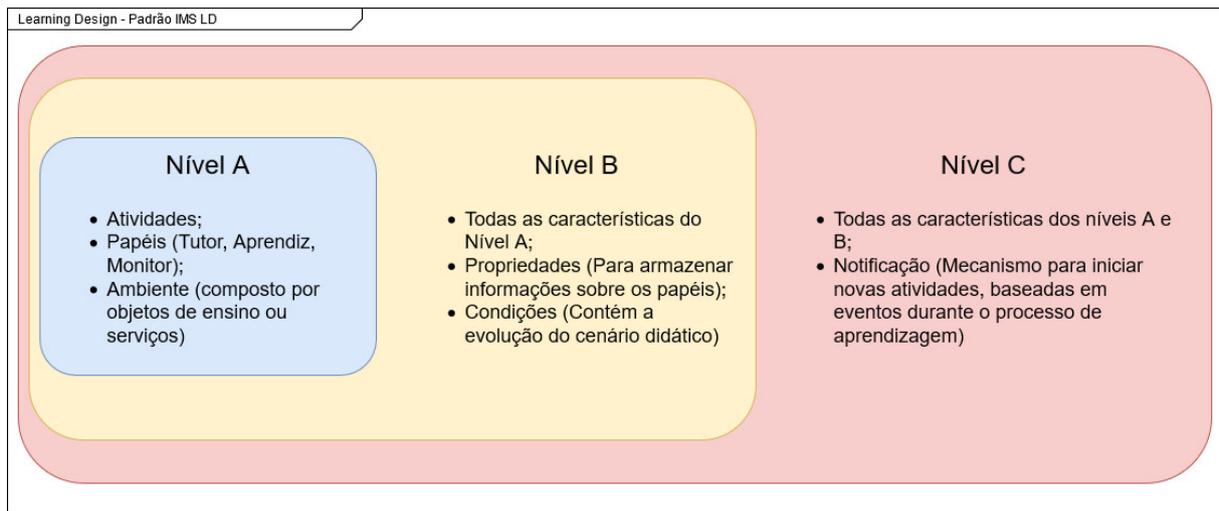


Figura 3.2: Modelo conceitual da estrutura do *Learning Design* por nível.

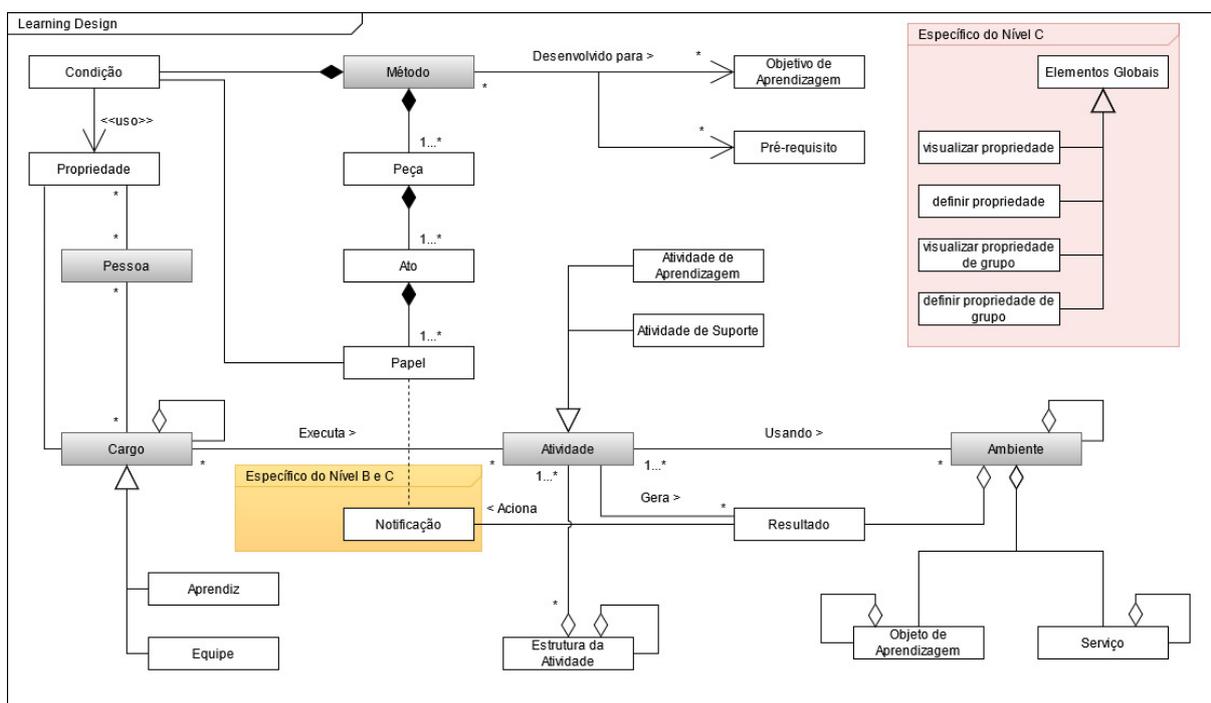


Figura 3.3: Diagrama UML do IMS LD em Nível C; Classes principais acentuadas por gradiente cinza. Fonte: Adaptado de *IMS Learning Consortium* (2003). Traduzido pelo autor.

A Figura 3.3 ilustra os componentes que compõem o padrão IMS LD, demarcados por nível. Os elementos em nível A são essenciais para se criar um arquivo em formato IMS LD, como demonstrado no guia desenvolvido neste estudo

### 3.5 Ferramenta adotada

Para este estudo, foi escolhido o *RELOAD Editor*: editor em forma de formulário e com visualização da estrutura em árvore hierárquica, que suporta modelagem de arquivos *IMS Learning Design* nos níveis A, B e C, além de comportar modelagem de Pacotes de Conteúdo IMS e arquivos em formato SCORM. Durante a fase de escolha da ferramenta para uso neste estudo, foram encontradas algumas limitações e obstáculos relacionados a alguns editores: os editores *Prolix* e *Open GLM*, desenvolvidos para Java versão 16 usando *Framework Eclipse*, se mostraram incompatíveis com versões mais novas do Java, e com máquinas modernas com sistema operacional *Windows 10* e Linux baseado em distribuição Ubuntu. Quanto ao *LAMS*, não foi escolhido por necessitar de instalação em servidor em rede local ou online; como o objetivo deste estudo é apresentar uma proposta acessível, o *RELOAD Editor* se mostrou mais relevante, de fácil instalação e por poder ser utilizado mesmo com a falta de sinal de internet, uma vez instalado em uma máquina.

# Capítulo 4

## Proposta

Este estudo pretende trazer duas análises sobre a criação de Unidades de Aprendizagem: Uma direcionada ao tutor, com um exemplo de Unidades de Aprendizagem em um contexto de aulas introdutórias para ensino de programação para leigos, idealizada para alunos do ensino médio e do ensino fundamental de escolas públicas. Essa Unidade de Aprendizagem não apenas serve de modelo, mas também pode ser aplicada caso o tutor tenha interesse em iniciar um curso do tema, e em seguida praticar o desenvolvimento de novas Unidades de Aprendizagem para dar sequência ao conteúdo sugerido. Já a outra subseção deste capítulo é direcionada ao *Learning Designer*, indivíduo responsável pelo planejamento e criação de aulas ou cursos a serem ministrados, detalhando as especificidades necessárias para a formalização destas aulas em Unidades de Aprendizagem. Para isso, serão discutidas ferramentas utilizadas, modo de formalização semântica e adequação à linguagem de modelagem selecionada. Os anexos desenvolvidos neste estudo se encontram em páginas em servidor *Google Drive* e em repositório Lourdes Brasil, da Universidade de Brasília.<sup>1</sup>

### 4.1 Ao Tutor: Um curso de Introdução à Programação

Para este estudo foi desenvolvido um curso introdutório de programação, na forma de 12 aulas com duração de 1 hora e 30 minutos cada encontro. O formato selecionado para as aulas são aulas em sua grande maioria com dois principais momentos, exposição de conceitos e em seguida atividades práticas de implementação destes conceitos. Este curso

---

<sup>1</sup>Todos os artefatos gerados por este estudo, se encontram no endereço <https://drive.google.com/drive/folders/1qtd8k0UW0uen1Kc90uM8DZbImQL1zwdt?usp=sharing> e também no repositório Lourdes Brasil da Universidade de Brasília em <http://quizwiki.cic.unb.br/lmb> acessados em 28/05/2021. O plano de curso desenvolvido, juntamente com o guia para padronização em IMS LD se encontram na seção de apêndices ao final do documento.

apresenta algumas aulas diferenciadas, como uma aula específica para atividade avaliativa e atividade desplugada, uma aula opcional para revisão de conteúdos e o desenvolvimento de um projeto pelos alunos de um jogo narrativo de ficção interativa com duração previsto para 4 aulas.

O curso *Coding 101*, é composto pelos seguintes artefatos:

- Plano de Curso: Apresenta descritivo das características do curso, uma visão geral dos conteúdos previstos e descrição de aulas;
- Material Didático: Contém um módulo demonstrando a instalação dos programas e arquivos necessários para os computadores a serem usados durante o curso. Possui também os enunciados de cada atividade proposta, assim como detalhamento das soluções das atividades a serem aplicadas nos planos de aula;
- Slides de apoio para cada aula, contendo os principais conceitos a serem estudados em sala;
- Plano de Aula: Planejamento de cada encontro previsto para o curso, explicações e tempo previsto para cada atividade e dinâmica planejada;
- Arquivos necessários para os exercícios de programação em computadores, assim como arquivos de suas soluções;
- Atividade Desplugada *Code Bot*: Jogo de tabuleiro que permite a prática dos conceitos vistos durante as aulas de programação em um contexto competitivo e descontraindo.
- Arquivos de *Learning Design* em padrão IMS LD do curso, desenvolvidos de acordo com o guia para modelagem de curso em padrão *IMS LD* utilizando o *software RELOAD Editor*.

Os primeiros encontros do curso apresentam aos alunos os seguintes conceitos introdutórios de programação, de forma gradual e com aumento de complexidade: Algoritmos, linguagens de programação, métodos, estruturas condicionais, laços de repetição e variáveis. O projeto realizado nos últimos encontros serve como uma aplicação prática de todos os conceitos vistos durante o curso.

Para este curso, foram selecionadas a linguagem de programação Java por ter execução baseada em máquina virtual, o que permite que execute em máquinas com as mais variadas especificações de componentes e sistemas operacionais. Algumas necessidades consideradas foram também que o curso não dependa de sinal de internet, e que utilize apenas programas com baixa exigência de processamento, daí a escolha por utilizar a biblioteca de programação visual JKarel (Figura 4.1) e ao ambiente de desenvolvimento

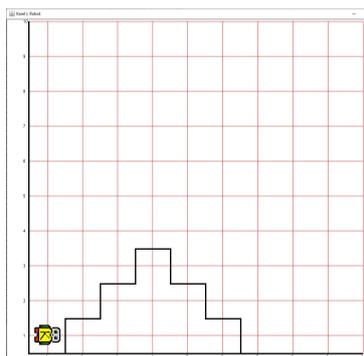


Figura 4.1: Biblioteca JKarel de aprendizagem por meio visual.

integrado (IDE) *JGrasp*. As ferramentas escolhidas para uso neste curso possuem versões para os sistemas operacionais *Windows*, *Mac* e *Linux*, o que abrange ainda mais seu alcance de aplicação. Como não existe padronização de máquinas utilizadas em instituições de ensino brasileiras, é de suma importância que este curso possa ser aplicado nas mais diversas configurações de equipamento e estrutura.

Figura 4.2: Atividade Desplugada - *Code Bot*

(a) Componentes do *Code Bot*.



(b) Representação de uma partida competitiva com 4 jogadores.



A atividade desplugada desenvolvida para este curso, diretamente influenciada pelo Movimento *Maker*, possui bem claras suas características *Do It Yourself* em seus componentes impressos, que devem ser recortados, montados e colados. O fato de todos os componentes serem impressos é essencial para facilitar o acesso à atividade; não há necessidade de nenhum outro objeto além dos presentes nas folhas a serem impressas. Quanto ao livro de regras, possui a descrição de todos os componentes, seus usos e instruções do jogo. Pontos considerados em sua confecção foi a facilidade de ser impresso, possuindo apenas imagens essenciais, possui tamanho padrão A4, e todas as cartas e componentes do jogo possuem indicadores com linhas de corte e, no caso dos dados, instruções de montagem e colagem. O propósito da atividade desplugada é exercitar conceitos vistos durante

as aulas em um contexto descontraído de um jogo de tabuleiro competitivo. Como o jogo possui ações semelhantes às vistas durante as aulas de programação do JKarel, disponibiliza aos alunos uma nova forma de experienciar o conteúdo estudado, através do estímulo tátil de jogos de tabuleiro, e seu aspecto visual intuitivo. Por se assemelhar à estrutura das atividades de programação com JKarel, a atividade desplugada foi criada com o propósito também de servir como uma alternativa ao instrutor para imprevistos que podem ocorrer em sua instituição de ensino, como falta de energia ou impossibilidade de uso dos computadores para a aula planejada.

As últimas aulas do curso tem como sugestão uma atividade de revisão inspirada na Aprendizagem Baseada em Problemas: os alunos deverão criar um produto na forma de narrativa interativa, aplicando todos os conceitos aprendidos nas aulas anteriores.

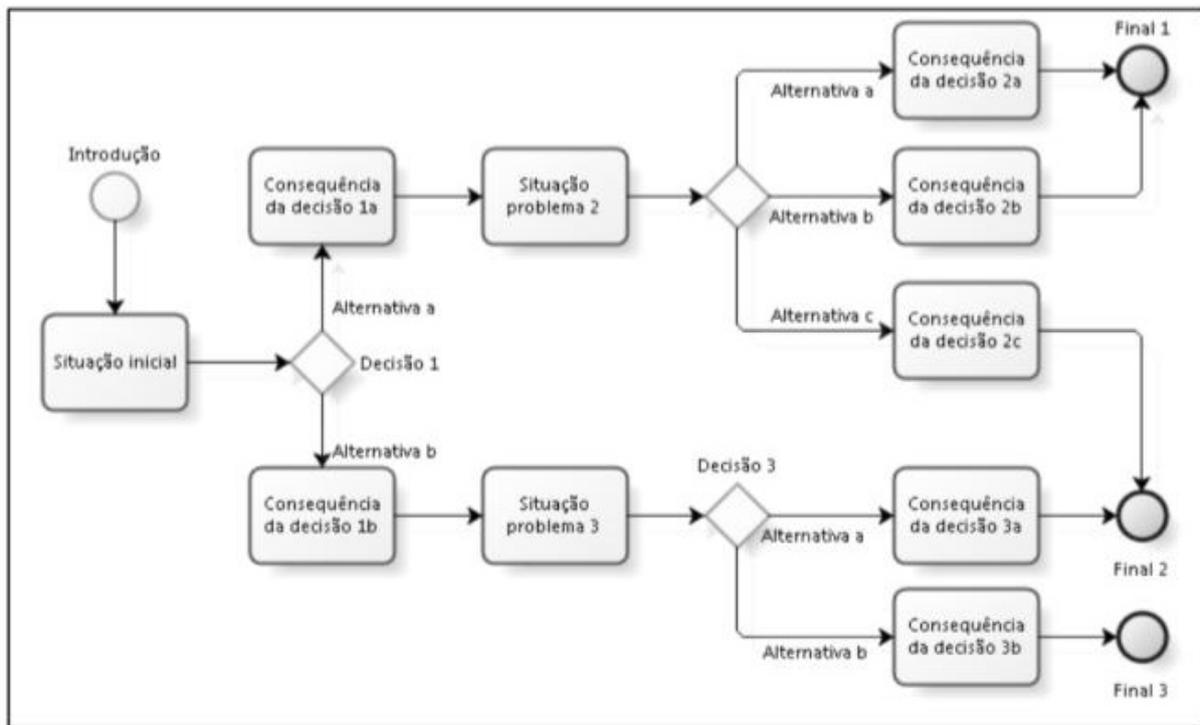


Figura 4.3: Exemplo de Modelo Narrativo de Ficção Interativa. Fonte: FREIRE, George André Alve; MADEIRA, Charles Andryê Galvão. Modelo conceitual para criação de jogos voltados para a tomada de decisão em gerenciamento de projetos. Anais dos Workshops do IV Congresso Brasileiro de Informática na Educação (CBIE 2015).

Essa atividade na forma de projeto, tem como objetivo além da revisão de conceitos estudados, visa também introduzir o aluno ao desenvolvimento de algoritmos utilizando simbologia padronizada em [30], conforme Figura 4.4.

Espera-se que o desenvolvimento de um produto pelo aluno, que poderá ser compartilhado e executado por amigos, colegas e parentes, sirva para estimular o engajamento

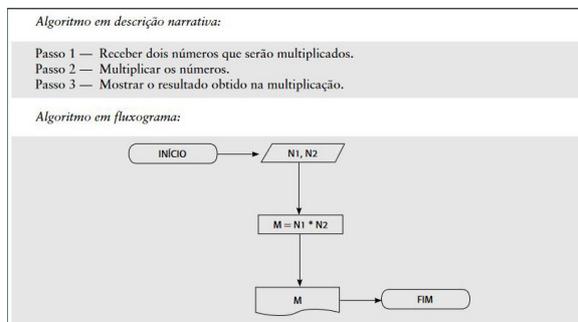
Figura 4.4: Exemplo de símbolos a serem usados para confecção visual de algoritmos segundo Ascencio(2007). Fonte: ASCENCIO, Ana Fernanda Gomes; CAMPOS, Edilene Aparecida Veneruchi. Fundamentos da Programação de Computadores: Algoritmos, Pascal, C, C++ e Java. Pearson Prentice Hall, 2007

(a) Descrição dos símbolos e seus significados

Tabela 1.1 Conjunto de símbolos utilizados no fluxograma.

	Símbolo utilizado para indicar o início e o fim do algoritmo.
	Símbolo que permite indicar o sentido do fluxo de dados. Serve exclusivamente para conectar os símbolos ou blocos existentes.
	Símbolo utilizado para indicar cálculos e atribuições de valores.
	Símbolo utilizado para representar a entrada de dados.
	Símbolo utilizado para representar a saída de dados.
	Símbolo utilizado para indicar que deve ser tomada uma decisão, apontando a possibilidade de desvios.

(b) Exemplo de algoritmo segundo padrão sugerido



do aluno durante seu processo de aprendizagem. Ao término do desenvolvimento do projeto, é recomendado que os alunos apresentem seus projetos em forma de *Sales Pitch*, com duração entre 5 e 15 minutos, onde os alunos deverão simular uma apresentação de produto objetivando sua venda para compradores ou investidores fictícios. Essa atividade é importante para estimular habilidades nos alunos como oratória, exposição de ideias e exercitar também no aluno a habilidade de valorização do trabalho desenvolvido.

As atividades sugeridas neste curso visam trazer meios e abordagens diferentes para contribuir para a qualidade do processo de aprendizagem do aluno, sempre no contexto de programação e uso de tecnologias, buscando a familiarização do aluno na prática de seus conceitos.

## 4.2 Ao Learning Designer

Nesta seção, serão discutidas a formalização de cursos no formato *IMS LD*. O curso *Coding 101*, desenvolvido neste estudo, foi utilizado como exemplo para esta formalização, mas é importante destacar que a padronização de cursos em *Learning Design* independe de disciplina ou área de estudo; pode ser aplicado em qualquer contexto educacional. Neste estudo, serão aplicados dois padrões desenvolvidos pelo consórcio *IMS*: *IMS LD* para arquivos de *Learning Design* e *IMS Content Package* para pacotes de conteúdo. O primeiro se refere à modelagem e estruturação de cursos, enquanto os pacotes de conteúdo possuem função orientada à reprodução automatizada de seus conteúdos em modelo de ensino *e-learning*. Neste estudo, a modelagem foi realizada utilizando o software *Reload Editor*.

Esta seção é voltada principalmente para a instituição de ensino, uma vez que a iniciativa de padronização de cursos, aulas e atividades, para um resultado frutífero, deve ser implementada por todo o corpo docente, após treinamento pela instituição. Uma vez formalizadas as atividades educacionais, os benefícios de armazenamento e reutilização já mencionados possibilitam um acolhimento mais completo aos novos instrutores que ingressam ao corpo docente institucional, uma vez que terá um acervo das atividades educacionais realizadas na instituição.

#### **4.2.1 Visão do plano do curso em Linguagem de Modelagem Educacional**

Para início da modelagem do curso, foi feito inicialmente um modelo em forma de fluxograma utilizando um editor de imagem (Figura 4.5) para facilitar a visualização e registro dos componentes do curso no *software* editor de *Learning Design*. Uma vez clara a ordem e os componentes do curso, criou-se o arquivo no formato *IMS LD* no *Reload Editor*. Um detalhe importante é que ao criar um modelo de *IMS LD* ou um pacote de conteúdo *IMS*, será gerado um arquivo na pasta destino chamado *imsmanifest.xml*, representando o arquivo do manifesto criado. O arquivo *XML* do manifesto é o arquivo mais importante de todo modelo criado; será referenciado pelos outros arquivos que compõem o modelo, e possui a definição e descrição de todas as atividades, objetivos de aprendizagem, estrutura de atividades, todos os recursos que compõem o modelo, os papéis a serem desempenhados pelos participantes do processo de ensino, ambientes utilizados para o ensino e métodos de ensino aplicados. Todos estes componentes estão descritos no manifesto com identificadores em forma de *tags*, semelhante à estrutura de arquivos *HTML*, para serem analisados e reproduzidos por programas reprodutores de arquivos padrão *IMS*.

## Estrutura de Curso: Coding 101

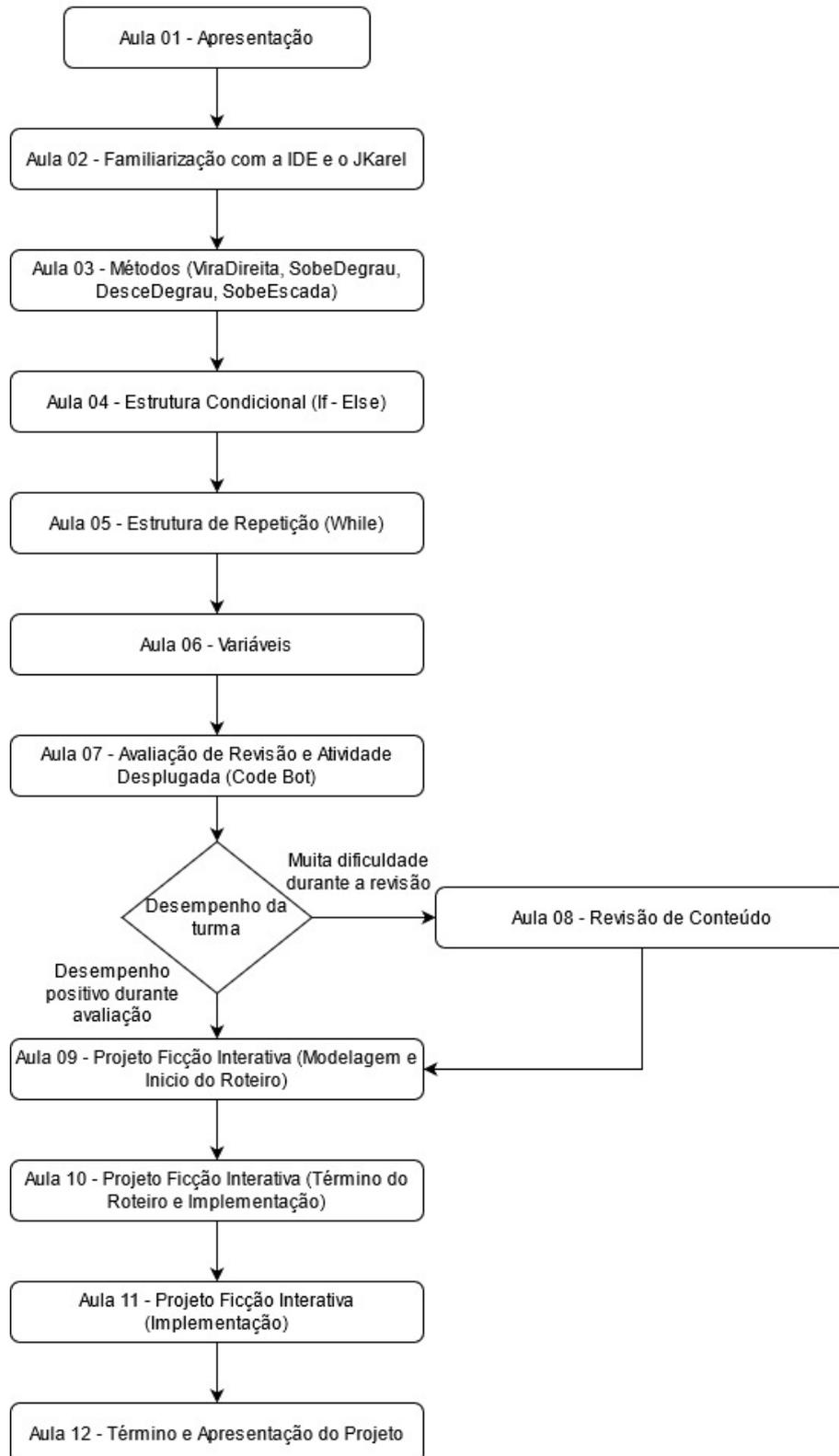


Figura 4.5: Modelo estrutural do curso *Coding 101*.

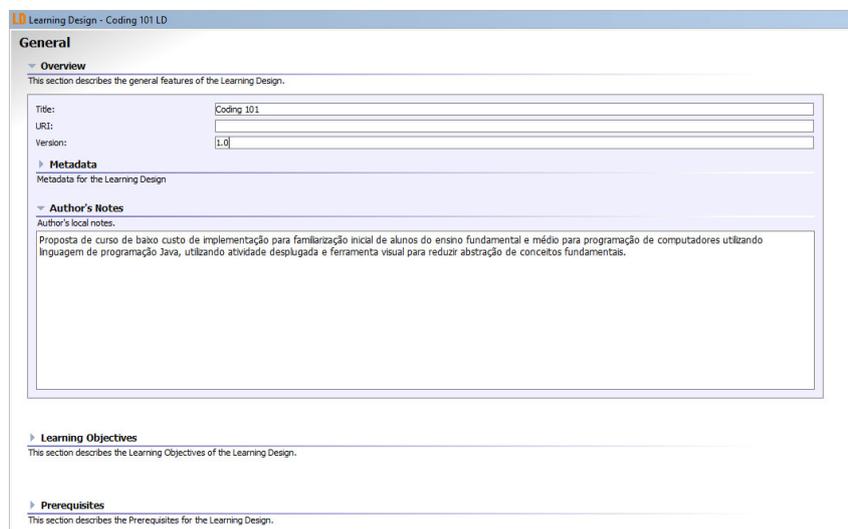
De acordo com a Figura 4.5, podemos identificar a estrutura do curso formada por 12 aulas, sendo uma delas uma atividade de suporte opcional (Aula 08 - Revisão de Conteúdo), a ser aplicada somente caso o desempenho da turma durante a atividade avaliatória da aula anterior tenha sido indesejável, com grande dificuldade dos alunos durante as questões realizadas.

Um outro ponto a ser considerado é que todas as aulas possuem recursos na forma de slides de apoio, e as aulas possuem uma estrutura de atividade na forma de aulas expositivas com aplicação prática de conceitos.

## 4.2.2 Guia para planejamento de cursos utilizando o padrão IMS LD

Neste estudo foi desenvolvido um guia de instruções em formato passo a passo para formalização de cursos em *IMS Learning Design* em Nível A: Modelagem necessária para instrutores de ensino praticarem para armazenamento de suas atividades de ensino e início de alimentação de um servidor ou banco de dados institucional. Apesar de desenvolvido em software *Reload Editor*, os conceitos mostrados são comuns em qualquer outro editor de *Learning Design* que possa ser usado.

Criado o modelo do *LD*, realizamos a descrição geral do curso.



The screenshot displays the 'General' section of the Reload Editor interface for a course titled 'Coding 101 LD'. The interface is organized into several expandable sections:

- Overview:** This section describes the general features of the Learning Design. It contains three input fields: 'Title' (filled with 'Coding 101'), 'URL', and 'Version' (filled with '1.0').
- Metadata:** A section for 'Metadata for the Learning Design'.
- Author's Notes:** A section for 'Author's local notes'. It contains a text area with the following content: 'Proposta de curso de baixo custo de implementação para familiarização inicial de alunos do ensino fundamental e médio para programação de computadores utilizando linguagem de programação Java, utilizando atividade desplugada e ferramenta visual para reduzir abstração de conceitos fundamentais.'
- Learning Objectives:** A section for 'Learning Objectives of the Learning Design'.
- Prerequisites:** A section for 'Prerequisites for the Learning Design'.

Figura 4.6: Descrição geral do *Coding 101* no *Reload Editor*.

Ao criar atividades, o *Reload Editor* realiza toda a sua modelagem em forma de árvore, conforme a Figura 4.7, enquanto existem outros editores com outras propostas, como exibição em fluxograma. Durante a criação das atividades, foram atribuídas as informações de descrição e objetivos de aprendizagem a cada atividade, além da criação de atividade

de suporte, e de estrutura de atividades, onde foram atribuídas as atividades criadas para classificação da estrutura de aula usada em cada atividade.

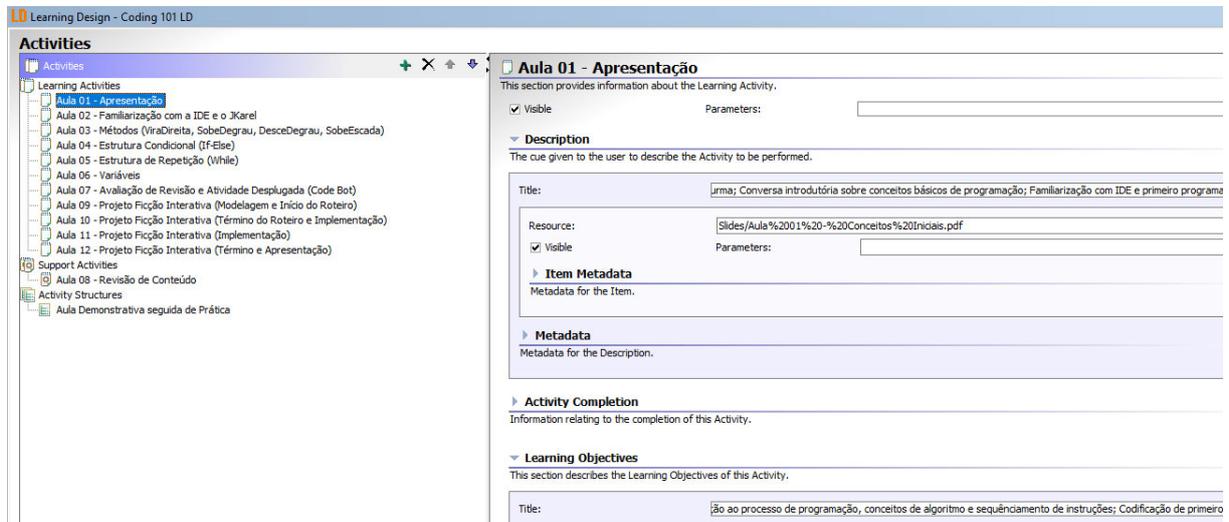


Figura 4.7: Exibição em árvore das atividades do curso *Coding 101* no *Reload Editor*.

Uma vez criadas as atividades(Figura 4.7), foram criadas os papéis do processo de aprendizagem (Figura 4.8), e foram introduzidos ao *LD* os recursos que compõem o curso (Figura 4.9). Após adicionados os recursos, estes devem ser relacionados com suas respectivas atividades.

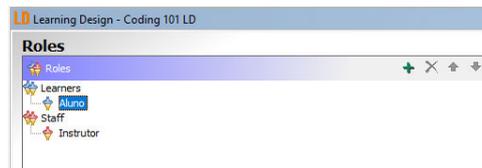


Figura 4.8: Papéis do curso *Coding 101* no *Reload Editor*.

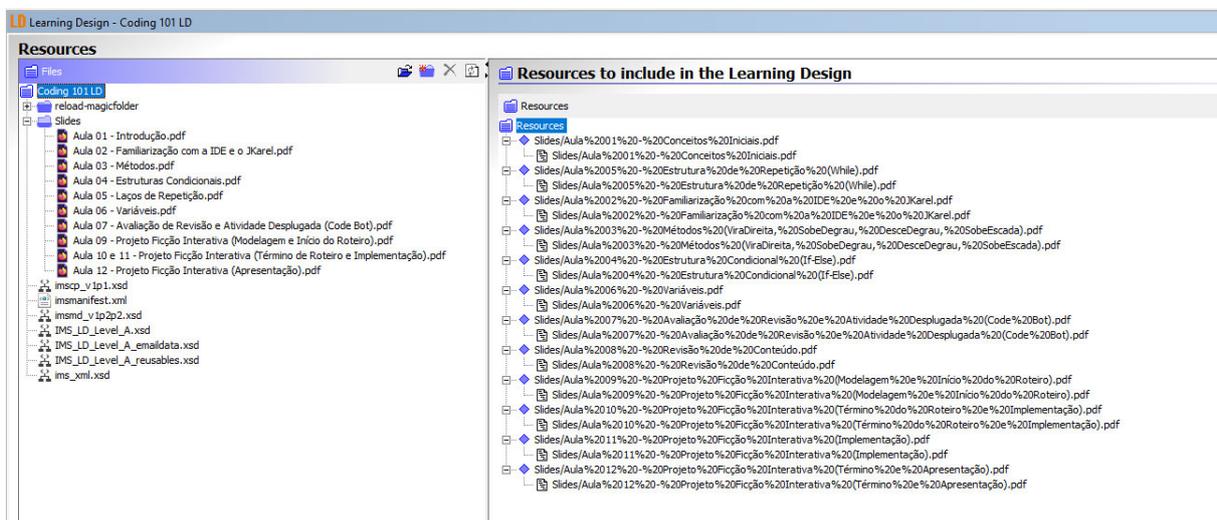


Figura 4.9: Recursos do curso *Coding 101* no *Reload Editor*.

Criados todos os papéis, atividades com estrutura e recursos atribuídos, deve ser feito o relacionamento das atividades e papéis através dos atos, que por sua vez compõem a peça que é nosso curso, conforme mostrado na Figura 4.10.

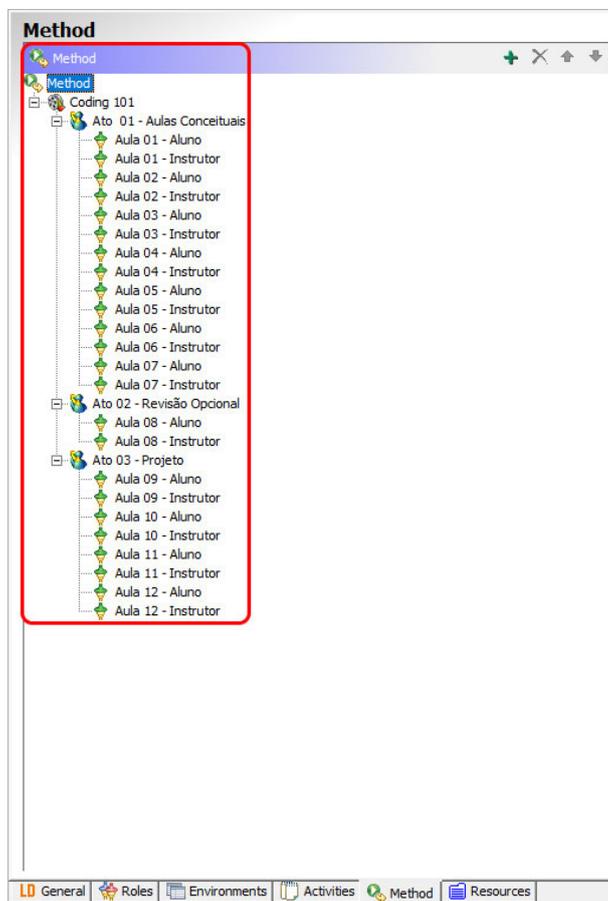


Figura 4.10: Peça representando o curso *Coding 101* no *Reload Editor*.

Uma vez salvo o arquivo após estas alterações, fica registrado no XML do manifesto os seguintes dados, organizados por *tags* identificadoras, suas relações e códigos de identificação únicos para cada componente do curso:

- Cabeçalho com informações e metadados do padrão *IMS LD* utilizado e sua versão;
- Nome do curso com sua descrição como Organização;
- Papéis, e no caso do papel de Aluno, valor mínimo e máximo de alunos para as atividades
- Atividades ordenadas com descrição, objetivos de aprendizagem e recursos relacionados;
- Métodos compostos por atos, e o relacionamento 1 a 1 feito para cada papel a ser exercido em cada atividade criada e ambiente externo usado;
- Listagem de todos recursos adicionados ao *LD*

Neste guia, foi demonstrada não só a modelagem do *Learning Design* do curso *Coding 101*, mas também de um pacote de conteúdo *IMS*. Pacotes de conteúdo são essenciais para os interessados em desenvolvimento de unidades de ensino para o formato *e-learning*, para criação de cursos automatizados para serem reproduzidos a distância. Para esse processo, os slides de apoio para a Aula 01 foram convertidos em formato HTML com sequência de imagens, para exemplificar a criação de um pacote de conteúdo.

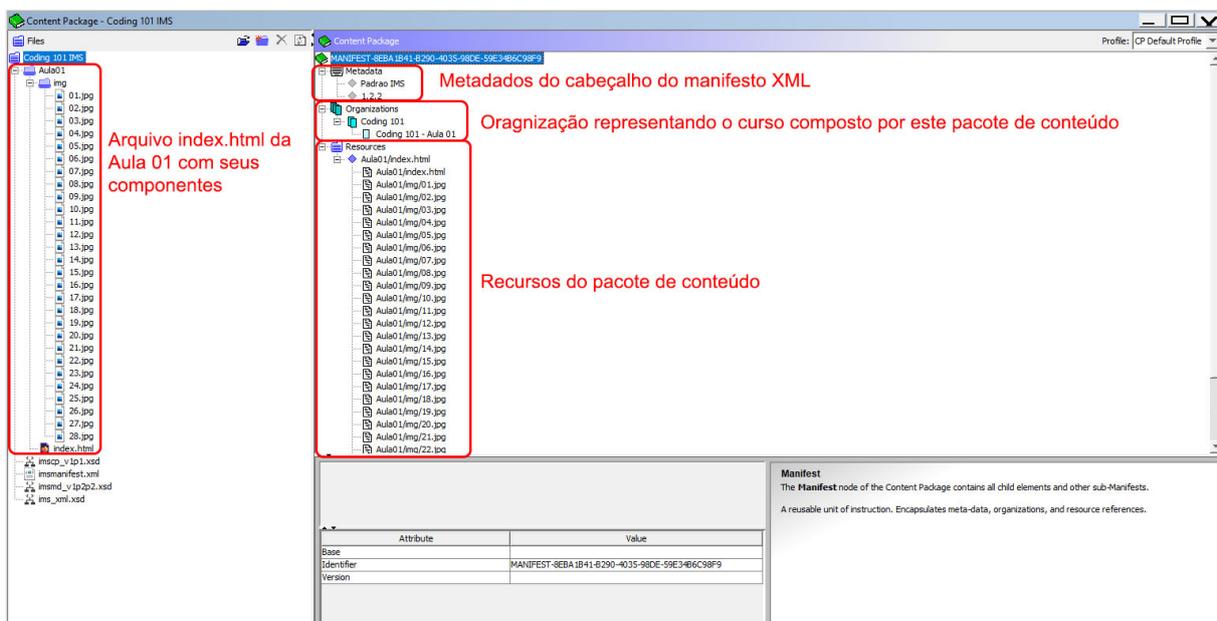


Figura 4.11: Exibição em árvore do pacote de conteúdo *IMS* no *Reload Editor*.

Espera-se que os apêndices desenvolvidos nesse estudo sirvam para ilustrar e motivar a formalização em padrão IMS LD pelo corpo docente em instituições de ensino, e que o curso desenvolvido como exemplo seja utilizado para início do ensino de programação em escolas, e que a aplicação destas aulas não só atuem na aprendizagem dos alunos, mas também preparem o tutor para que desenvolva suas próprias aulas, dando continuidade às atividades propostas.

# Capítulo 5

## Conclusão

### 5.1 Contribuições

É desejável que este estudo motive as instituições de ensino a aplicar a modelagem e formalização das ações de ensino praticadas pelo seu corpo docente, visando o registro, armazenamento, reutilização e modificação de suas aulas e cursos. Uma vez padronizados em arquivos computacionais, estas iniciativas de ensino podem ser analisadas para criação de relatórios e análises visando melhoria da qualidade do ensino e maior eficiência durante o processo de aprendizagem dos alunos.

A discussão da formalização e compartilhamento de esforços de ensino é de suma importância em um contexto como o atual, onde a tecnologia se faz tão presente na sociedade. É essencial que sejam aplicadas novas formas de uso dessas tecnologias para beneficiar alunos, docentes e instituições. A pandemia do coronavírus (Covid-19) ocorrida durante o desenvolvimento desse estudo, demonstrou como é essencial essa discussão, com a necessidade imediata de ajuste das instituições de ensino para modelos de ensino a distância e ensino híbrido.

Espera-se que o curso *Coding 101* proposto neste estudo sirva como motivador da implementação em larga escala para o ensino de programação nas instituições de ensino, principalmente da rede pública. É desejado que, ao aplicar as aulas previstas neste curso, o instrutor tenha familiaridade com a prática de ensino de programação, planejamento e formalização de atividades, e que tenha autonomia para dar continuidade ao curso, desenvolvendo novas aulas e atividades para construção da aprendizagem de novos conceitos de programação pelos alunos.

## 5.2 Trabalhos Futuros

Previsto para trabalhos futuros, estão o treinamento de campo para professores de variadas disciplinas em uma instituição de ensino para a formalização de atividades de ensino, a aplicação das atividades propostas como exemplo para serem replicadas visando iniciar o ensino de programação de instituições da rede pública de ensino, e a concepção de uma ferramenta de planejamento pela equipe, no âmbito do projeto *smartUnB.ECOS*[39].

# Referências

- [1] BERGIN, Joseph et al.: *Karel J Robot: a gentle introduction to the art of object-oriented programming in java*. Dream Songs Press, 2005. 1, 18, 19
- [2] JESUS, Hiago Oliveira de; RODRIGUEZ, Luis Cuevas; JUNIOR Almir de Oliveira Costa: *Predição de evasão escolar na licenciatura em computação*. Revista Brasileira de Informática na Educação – RBIE, (29):255–272, 2021. 2
- [3] HOLANDA, Maristela et al.: *Perfil das alunas no departamento de computação da universidade de Brasília*. WOMEN IN INFORMATION TECHNOLOGY (WIT), 11, 2017. 2
- [4] GOMES, Alex Sandro; GOMES, Claudia Roberta Araújo: *Classificação dos tipos de pesquisa em informática na educação*. in: Jaques, Patrícia Augustin; Pimentel, Mariano; Siqueira, Sean; Bittencourt, Ig. (org.) *metodologia de pesquisa científica em informática na educação: Concepção de pesquisa*. Porto Alegre: Sbc, 2020. (série metodologia de pesquisa em informática na educação, v. 1). <https://metodologia.ceie-br.org/livro-1/>, acesso em 2020-05-14. 7
- [5] GRIES, David: *What have we not learned about teaching programming?* IEEE Xplore, 2006. 10
- [6] ROBERTS, Eric: *The dream of a common language: The search for simplicity and stability in computer science education*. In Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education (SIGCSE '04). ACM, New York, NY, USA, página 115–119, 2004. 10
- [7] BECKER, Brett A.; QUILLE, Keith: *50 years of cs1 at sigcse: A review of the evolution of introductory programming education research*. In Proceedings of the 50th ACM Technical Cruz et al. RBIE v.29–2021180 symposium on computer science education, páginas 338–344, 2019. 10
- [8] BELL, Tim et al.: *Computer science unplugged: school students doing real computing without computers*. The New Zealand Journal of Applied Computing and Information Technology, 13(1):20–29, 2009. 11
- [9] NAPS, Thomas L. et al.: *Exploring the role of visualization and engagement in computer science education*. ACM SIGCSE Bulletin, 2003. 11
- [10] PIMENTEL, Fernando Silvio Cavalcante; FERREIRA, Adilson Rocha; FREITAS Raphael de Oliveira: *Gamificação como estratégia pedagógica no combate à evasão*.

- Potencialidades da implementação no ensino superior.* Congresso Internacional de Educação e Tecnologias - Encontro de Pesquisadores em Educação a Distância (CIET - EnPED), 2020. 11
- [11] GIMENES, R.: *A sala de aula invertida na disciplina literatura no ensino a distância.* Congresso Internacional ABED de Educação a Distância, 2018. 12
- [12] CONNOR, A.M.; KARMOKAR, S.; WHITTINGTON C.: *From stem to steam: Strategies for enhancing engineering technology education.* International Journal of Engineering Pedagogies, 5(2):37–47, 2015. 12
- [13] SAVERY, John R.: *Overview of problem-based learning: Definitions and distinctions.* Interdisciplinary Journal of Problem-Based Learning, 1(1):9–20, 2013. 12
- [14] KAÇAR, Turan et al.: *The effect of inquiry-based learning on academic success: A meta-analysis study.* International Journal of Education Literacy Studies, 9(2):15–23, 2021. 12
- [15] BADRIYAH, Nur Lailatul; ANEKAWATI, Anik; AZIZAH Lutfiana Fazat: *Application of pjbl with brain-based steam approach to improve learning achievement of students.* Jurnal Inovasi Pendidikan IPA, 6(1):88–100, 2020. 12
- [16] BRITO, Mirian Cristiane Alves: *Integrando material instrucional e experiências em um ambiente colaborativo de suporte à docência no âmbito de áreas temáticas em uma instituição de ensino superior.* Dissertação (mestrado) – Universidade Católica de Brasília, página 115–119, 2006. 12
- [17] STEINMETZ, Ernesto Henrique Radis: *Concepção de um ambiente de apoio ao auto-planejamento discente no seu processo de aprendizagem.* Dissertação (mestrado) – Universidade Católica de Brasília, 2007. 13
- [18] RAWLINGS, Adrian et al.: *Survey of educational modelling languages (emls).* CEN/ISSS Learning Technology Workshop Report, 2002. 14, 25
- [19] MIAO, Yongwu: *Cosmos: Facilitating learning designers to author units of learning using ims ld.* Proceedings of the 2005 conference on Towards Sustainable and Scalable Educational Innovations Informed by the Learning Sciences: Sharing Good Practices of Research, Experimentation and Innovation, página 275–282, 2005. 14
- [20] SÁ, Eveline de J. V.; TEIXEIRA, Jeane S. F.; FERNANDES Clovis T.: *Design de atividades de aprendizagem que usam jogos como princípio para cooperação.* XVIII Simpósio Brasileiro de Informática na Educação - SBIE, página 539–549, 2007. 15
- [21] MARTINEZ-ORTIZ, I. et al.: *Educational modeling languages. a conceptual introduction and a high-level classification.* Computers and Education. E-Learning, From Theory to Practice, Springer Netherlands, página 27–40, 2007. 15
- [22] SILVA, Luiz Augusto Matos da; BARRETO, Luciano Porto: *Interoperabilidade de unidades de aprendizagem do ims learning design em ambientes virtuais de aprendizagem.* XIX Simpósio Brasileiro de Informática na Educação (SBIE 2008), página 390–399, 2008. 15

- [23] CUEVAS, Patricia López et al.: *Courseeditor: A course planning tool compatible with ims-ld*. Computer Applications in Engineering Education, 21(3):421–431, 2013. 15, 26
- [24] JÚNIOR, Aladir F. Silva; SILVA, Leandro Roberto da; FERNANDES Clovis Torres: *Panorama dos editores de atividades de aprendizagem em ims learning design*. XXIV Simpósio Brasileiro de Informática na Educação (SBIE 2013), páginas 336–345, 2013. 16
- [25] OLIVEIRA, Francisco Kelsen de; GOMES, Alex Sandro: *Uma revisão sistemática da literatura sobre ferramentas de autoria de ims-ld*. Anais do XXVI Simpósio Brasileiro de Informática na Educação (SBIE 2015), páginas 376–385, 2015. 16
- [26] DA SILVA, Jesiel Viana; ALEXANDRE, Gustavo Henrique da Silva: *Um ambiente virtual colaborativo para o planejamento de ensino pbl em cursos de computação*. WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO (WEI), 28:41–45, 2020. 17
- [27] JANICIC, Milena Vujosevic; TOSIC, Dusan: *The role of programming paradigms in the first programming courses*. The teaching of mathematics, XI(2):63–83, 2008. 18
- [28] DEITEL, Paul; DEITEL, Harvey: *Java: como programar*. Pearson Prentice Hall, 2010. 18, 19
- [29] SIERRA, Kathy; BERT, Bates: *Java: como programar*. Alta Books, 2009. 19
- [30] ASCENCIO, Ana Fernanda Gomes, CAMPOS Edilene Aparecida Veneruchi de: *Fundamentos da programação de computadores*. Pearson Prentice Hall, 2007. 19, 33
- [31] HALVERSON, Erica Rosenfeld; SHERIDAN, Kimberly M.: *The maker movement in education*. Harvard Educational Review, 84(4):495–505, 2014. 20, 21
- [32] TRIGGS, Teal: *Scissors and glue:: Punk fanzines and the creation of a diy aesthetic*. Journal of Design History, 19(1):69–83, 2006. 20
- [33] HATCH, Mark: *The Maker Movement Manifesto: Rules for Innovation in the New World of Crafters, Hackers, and Tinkerers*. McGraw-Hill Education, 2013. 20, 21
- [34] SANTANA, André L. M. et al.: *Lite maker: Um fab lab móvel para aplicação de atividades mão na massa com estudantes do ensino básico*. V Congresso Brasileiro de Informática na Educação, Anais do XXII Workshop de Informática na Escola, páginas 211–220, 2016. 21
- [35] LOCKYER, Lori et al.: *Handbook of research on learning design and learning objects: Issues, applications, and technologies*. IGI Global, 2009. 25
- [36] KOPER, Rob: *Modeling units of study from a pedagogical perspective the pedagogical meta-model behind eml*. Educational Technology Expertise Centre - Open University of the Netherlands, 2001. 26

- [37] TATTERSALL, Colin: *Using the ims learning design notation for the modelling and delivery of education*. The Open University of the Netherlands, 2007. 26
- [38] IMS Global Learning Consortium, Inc.: *Ims learning design best practice and implementation guide*. [https://www.imsglobal.org/learningdesign/ldv1p0/imsld\\_bestv1p0.html#1503778](https://www.imsglobal.org/learningdesign/ldv1p0/imsld_bestv1p0.html#1503778), acesso em 2020-05-14. 26
- [39] NÓBREGA, Germana M. da; CRUZ, Fernando W.: *Rumo a um ecossistema educacional apoiado por computador e socialização em rede descentralizada*. Anais do Simpósio Brasileiro de Sistemas Colaborativos - SBSC 2021 (to appear), 2021. 44

# Apêndice A

## *Coding 101*: Plano de Curso



**UnB**

**Universidade de Brasília  
Departamento de Ciência da Computação**

# **Projeto de Licenciatura Plano de Curso: Coding 101**

**Professora Orientadora:  
Prof.a Dr.a Germana Menezes da Nóbrega**

**Aluno:  
Lucas Nunes Coelho Cardoso - 14/0062998**

**1/2021**

# Sumário

<b>1. Introdução</b>	<b>2</b>
<b>2. Público Alvo</b>	<b>3</b>
<b>3. Objetivos</b>	<b>4</b>
<b>4. Conteúdo Previsto</b>	<b>5</b>
4.1. Habilidades	5
4.2. Bases Tecnológicas	5
<b>5. Estrutura do Curso</b>	<b>6</b>
<b>6. Descritivo de Aulas</b>	<b>7</b>
<b>Segue um descritivo de cada aula prevista para o curso Coding 101:</b>	<b>7</b>
<b>7. Referência Bibliográfica do Curso</b>	<b>9</b>

# 1. Introdução

Este documento descreve as atividades práticas que compõem o curso introdutório de programação intitulado “Coding 101”, desenvolvido como parte da proposta para a disciplina de atividade de conclusão do curso de Licenciatura em Computação na Universidade de Brasília. O projeto propõe o desenvolvimento de material didático acessível e de baixo custo, para ser disponibilizado para professores da rede pública de ensino, podendo ser aplicado com a utilização de laboratório de informática, e possui materiais a serem usados mesmo se a instituição de ensino não dispuser de computadores para uso dos alunos e professor na forma de um jogo de tabuleiro onde todos os componentes podem ser impressos.

O conteúdo previsto para o curso é formado pela introdução de conceitos fundamentais de programação em Java, utilizando a classe JKarel como ferramenta lúdica para ensino visual de conceitos abstratos.

Tabela 1 – Lista de Contatos

Nome	Vínculo Institucional	Contato
Prof.a Dr.a Germana Menezes da Nóbrega	Orientadora da disciplina Projeto de Licenciatura na Universidade de Brasília	gmnobrega@unb.br
Lucas Nunes Coelho Cardoso	Aluno da disciplina Projeto de Licenciatura na Universidade de Brasília	lucasncc@gmail.com

## 2. Público Alvo

O público alvo definido são alunos do Ensino Fundamental e Ensino Médio, e possui como único pré-requisito que o aluno esteja alfabetizado. A metodologia empregada será, principalmente, a utilização de ferramenta visual como facilitador para exposição de conceitos abstratos (Classe JKarel), além da aplicação de princípios do socioconstrutivismo em sala de aula.

O curso possui carga horária prevista para 12 aulas, sendo um encontro opcional caso a turma necessite de revisão após avaliação, com duração de 1 hora e 30 minutos cada encontro, totalizando 18 horas de curso.

### 3. Objetivos

O objetivo geral deste projeto é possibilitar que professores da rede pública tenham acesso a um material que os permita lecionar um curso de introdução à informática e programação.

Como objetivos específicos, este propõe a exposição, discussão e fixação dos conceitos fundamentais da computação, e a contextualização destes conceitos aplicados à realidade do aluno.

Este curso é composto por:

1. Plano de Curso: Apresenta descritivo das características do curso, uma visão geral dos conteúdos previstos e descrição de aulas;
2. Material Didático: Contém um módulo demonstrando a instalação dos programas e arquivos necessários para os computadores a serem usados durante o curso. Possui também os enunciados de cada atividade proposta, assim como detalhamento das soluções das atividades;
3. Slides de apoio para cada aula, contendo os principais conceitos a serem estudados em sala;
4. Plano de Aula: Planejamento de cada encontro previsto para o curso, explicações e tempo previsto para cada atividade e dinâmica planejada;
5. Arquivos necessários para os exercícios de programação em computadores, assim como arquivos de suas soluções;
6. Atividade Desplugada `\emph{Code Bot}`: Jogo de tabuleiro que permite a prática dos conceitos vistos durante as aulas de programação em um contexto competitivo e descontraído.
7. Arquivos de Learning Design em padrão *IMS LD* do curso.

## 4. Conteúdo Previsto

Serão trabalhados as seguintes habilidades e conceitos fundamentais de programação:

### 4.1. Habilidades

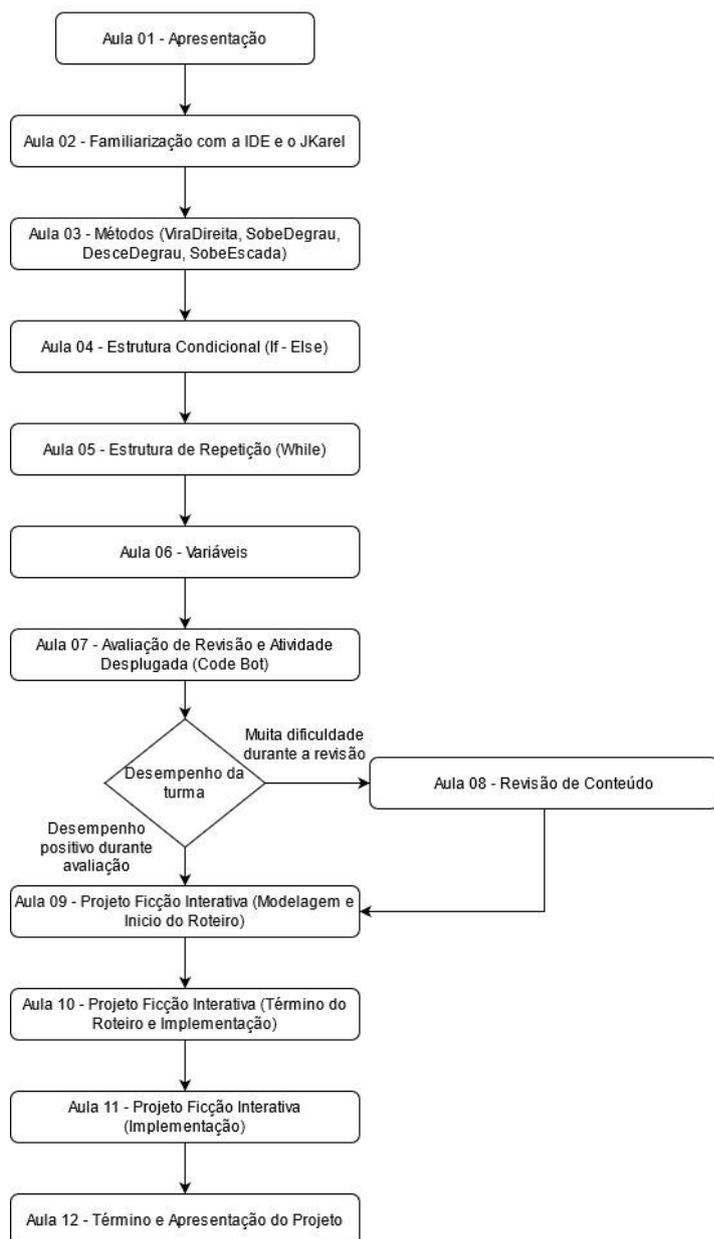
- Ter domínio dos conceitos fundamentais de programação;
- Situar-se na Lógica de Programação;
- Raciocínio lógico;
- Abstração.

### 4.2. Bases Tecnológicas

- Conceitos fundamentais de lógica de programação;
- Algoritmos e fluxogramas;
- Programação estruturada;
- Elementos de um algoritmo;
- Diversos tipos de dados;
- Tipos de variáveis;
- Declaração de variáveis;
- Estrutura sequencial;
- Constantes;
- Comentários;
- Expressões, operadores e funções;
- Comandos básicos;
- Estruturas condicionais;
- Estruturas de repetição.

## 5. Estrutura do Curso

### Estrutura de Curso: Coding 101



## 6. Descritivo de Aulas

Segue um descritivo de cada aula prevista para o curso Coding 101:

- Aula 01 - Apresentação
  - Descrição: Apresentação do Instrutor e Alunos da turma, conversa introdutória sobre conceitos básicos de programação e familiarização com IDE e codificação do primeiro programa para realizar impressão de mensagem em tela.
  - Objetivos de Aprendizagem: Iniciar o desenvolvimento de um vínculo afetivo entre os alunos e o instrutor; Apresentação ao processo de programação, conceitos de algoritmo e sequenciamento de instruções; Codificação de primeiro programa simples pelos alunos.
- Aula 02 - Familiarização com a IDE e o JKarel
  - Descrição: Familiarização com a IDE e a biblioteca de classes JKarel; Realizar sequência de atividades utilizando o JKarel.
  - Objetivos de Aprendizagem: Aprender as instruções básicas do JKarel: Mover, Desligar, Virar para a Direita, Pegar e Depositar Objeto.
- Aula 03 - Métodos (ViraDireita, SobeDegrau, DesceDegrau, SobeEscada)
  - Descrição: Prática de programação usando JKarel. Introdução ao conceito de criação de novos métodos para realização de tarefas repetitivas.
  - Objetivos de Aprendizagem: Encapsulamento de conjunto de instruções em novos métodos para tarefas repetitivas.
- Aula 04 - Estrutura Condicional (If-Else)
  - Descrição: Apresentação e prática do conceito de estrutura condicional (If / Else) utilizando JKarel.
  - Objetivos de Aprendizagem: Utilização de estruturas condicionais para desenvolvimento de algoritmos adaptáveis.
- Aula 05 - Estrutura de Repetição (While)
  - Descrição: Apresentação e prática do conceito de estrutura de repetição (While) utilizando JKarel.
  - Objetivos de Aprendizagem: Utilização de laços de repetição para resolução de tarefas que contenham redundância de instruções.
- Aula 06 - Variáveis
  - Descrição: Apresentação e prática do conceito de variáveis e seus diferentes tipos utilizando JKarel.
  - Objetivos de Aprendizagem: Utilização de variáveis para realização de tarefas que precisem de armazenamento de dados.
- Aula 07 - Avaliação de Revisão e Atividade Desplugada (Code Bot)
  - Descrição: Realizar atividade avaliativa em turma na forma de questões de múltipla escolha para detecção do ritmo de aprendizagem da turma, identificar

- dúvidas e dificuldades que necessitem de revisão; Atividade desplugada para prática dos conceitos estudados em método tátil sem uso de computadores.
- Objetivos de Aprendizagem: Avaliação de aprendizagem dos alunos; Definir a necessidade de aula de revisão de conceitos antes do início do projeto.
  - Aula 08 - Revisão de Conteúdo (Atividade de Suporte)
    - Descrição: Atividade opcional para revisão de conteúdos, identificados por dúvidas e dificuldades apresentadas por alunos durante a atividade avaliativa da Aula 07.
    - Objetivos de Aprendizagem: Reforçar conceitos prévios que possam apresentar pontos de dúvidas ou dificuldades.
  - Aula 09 - Projeto Ficção Interativa (Modelagem e Início do Roteiro)
    - Descrição: Iniciar a modelagem e escrita de roteiro de um jogo de ficção interativa, utilizando apenas texto para criar um livro-jogo eletrônico.
    - Objetivos de Aprendizagem: Prática de atividade de modelagem de árvore de decisões que compõem o jogo.
  - Aula 10 - Projeto Ficção Interativa (Término do Roteiro e Implementação)
    - Descrição: Finalizar a escrita do roteiro e iniciar a programação do jogo de ficção interativa.
    - Objetivos de Aprendizagem: Praticar aplicação dos conceitos vistos em aula, principalmente estrutura condicional; apresentação e aplicação do conceito de entrada de dados pelo usuário.
  - Aula 11 - Projeto Ficção Interativa (Implementação)
    - Descrição: Dar continuidade à programação do jogo de ficção interativa.
    - Objetivos de Aprendizagem: Praticar aplicação dos conceitos vistos em aula, principalmente estrutura condicional.
  - Aula 12 - Projeto Ficção Interativa (Término e Apresentação)
    - Descrição: Terminar a implementação do projeto; Alunos realizarem apresentação curta, em formato de discurso de vendas do projeto desenvolvido; Possibilitar que os alunos joguem os jogos de seus colegas de sala para troca de experiências.
    - Objetivos de Aprendizagem: Praticar aplicação dos conceitos vistos em aula; prática da habilidade de apresentação de trabalhos em sala de aula.

## 7. Referência Bibliográfica do Curso

ASCENCIO, Ana Fernanda Gomes; CAMPOS, Edilene Aparecida Veneruchi. Fundamentos da Programação de Computadores: Algoritmos, Pascal, C, C++ e Java. Pearson Prentice Hall, 2007

BERGIN, Joseph; STEHLIK, Mark; ROBERTS, Jim; PATTIS, Richard. Karel J Robot: A Gentle Introduction to the Art of Object-Oriented Programming in Java. Dreamsongs Press, 2013

DEITEL, Paul; DEITEL, Harvey. Java: Como Programar. Pearson Prentice Hall, 2010

SIERRA, Kathy; BATES, Bert. Use a cabeça! Java. Ed Alta Books, 2009

## Apêndice B

# Guia para Modelagem de Curso em Padrão IMS LD Utilizando o Software *Reload Editor*



**UnB**

**Universidade de Brasília  
Departamento de Ciência da Computação**

# **Guia para Modelagem de Curso em Padrão IMS LD Utilizando o Software Reload Editor**

**Professora Orientadora:  
Prof.a Dr.a Germana Menezes da Nóbrega**

**Aluno:  
Lucas Nunes Coelho Cardoso - 14/0062998**

**1/2021**

# Sumário

<b>1. Introdução</b>	<b>2</b>
<b>2. Modelagem de Arquivo IMS Learning Design (LD)</b>	<b>4</b>
<b>3. Modelagem de Pacote de Conteúdo em Padrão IMS</b>	<b>22</b>
<b>4. Referências Bibliográficas</b>	<b>30</b>

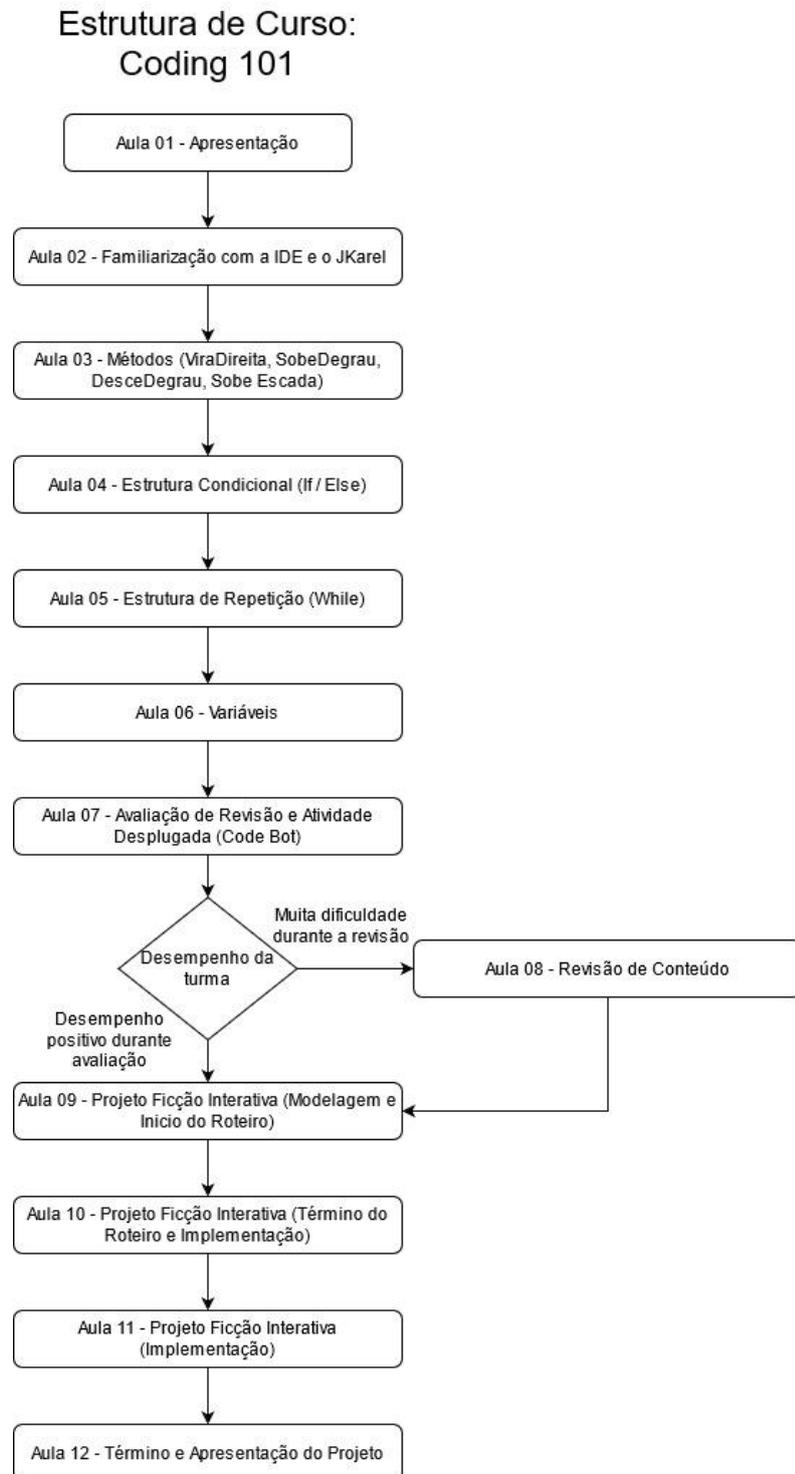
# 1. Introdução

Este documento descreve a modelagem em padrão *IMS Learning Design* de um curso introdutório de programação intitulado “*Coding 101*”, em formato de guia, desenvolvido como parte da proposta para a disciplina de atividade de conclusão do curso de Licenciatura em Computação na Universidade de Brasília.

Tabela 1 – Lista de Contatos

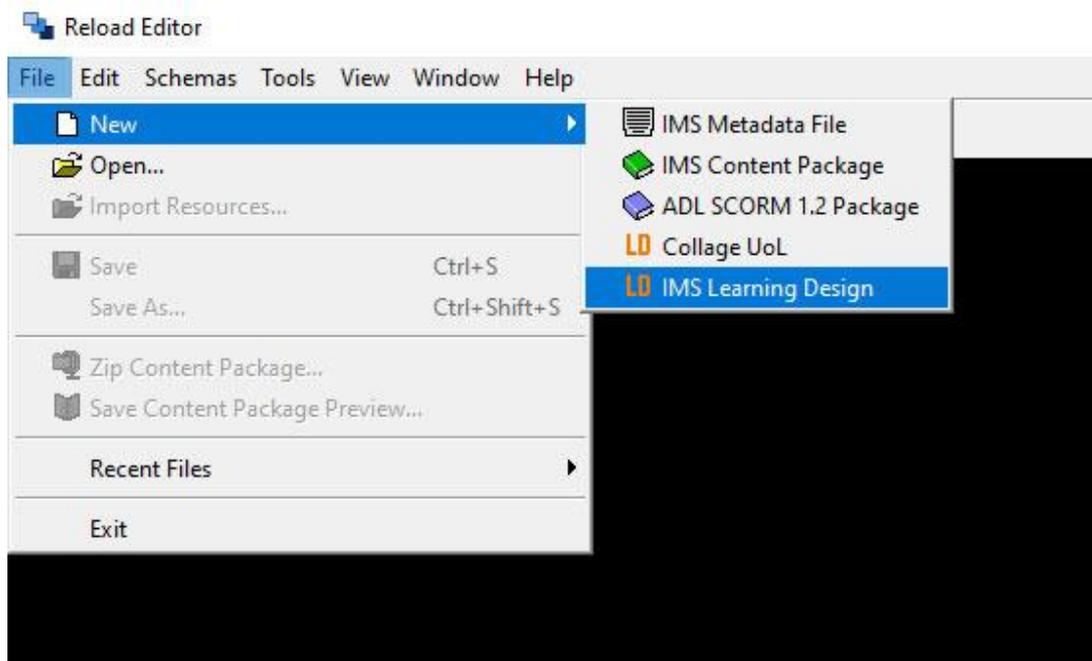
Nome	Vínculo Institucional	Contato
Prof.a Dr.a Germana Menezes da Nóbrega	Orientadora da disciplina Projeto de Licenciatura na Universidade de Brasília	gmnobrega@unb.br
Lucas Nunes Coelho Cardoso	Aluno da disciplina Projeto de Licenciatura na Universidade de Brasília	lucasncc@gmail.com

**Figura: Estrutura Conceitual do Curso “Coding 101”**



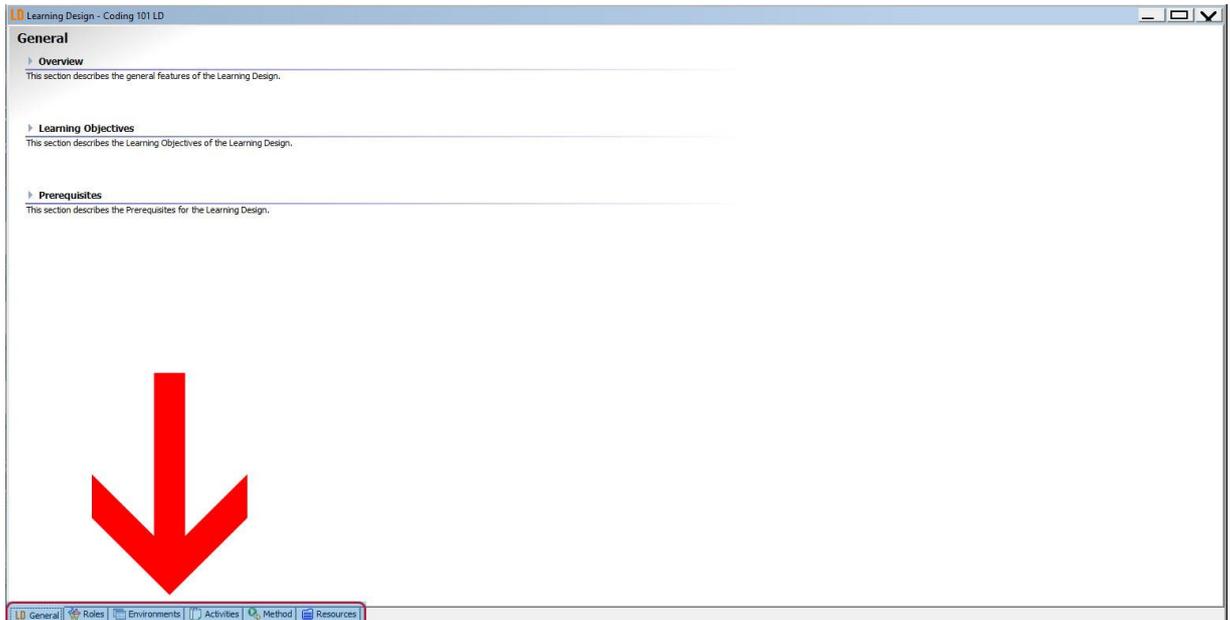
## 2. Modelagem de Arquivo *IMS Learning Design* (LD)

Vamos criar um arquivo de *Learning Design* no padrão *IMS-LD* (também referido como somente LD neste documento), selecionando "File", "New", "IMS Learning Design".

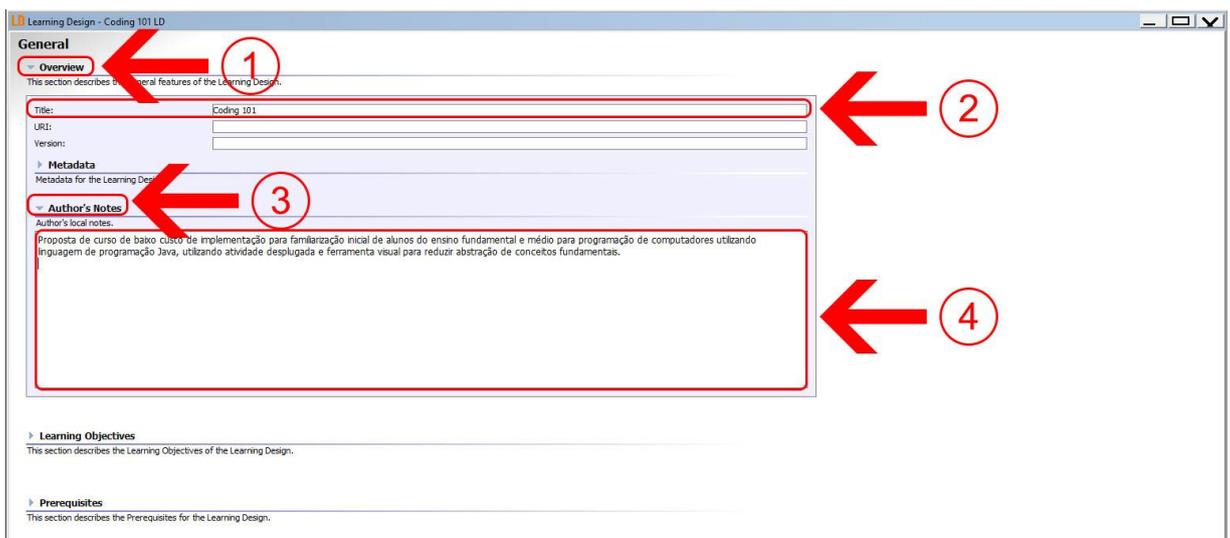


Após selecionada a pasta para registro dos arquivos, será exibida uma nova janela, mostrando o nome do pacote, definido pelo nome da pasta destino selecionada. Ao criar um pacote de conteúdo, o editor Reload imediatamente cria 7 arquivos na pasta destino com nomes padronizados, sendo o mais importante deles o "*imsmanifest.xml*", principal arquivo do pacote de conteúdo, que registra todas as informações do pacote e é referenciado pelos outros arquivos.

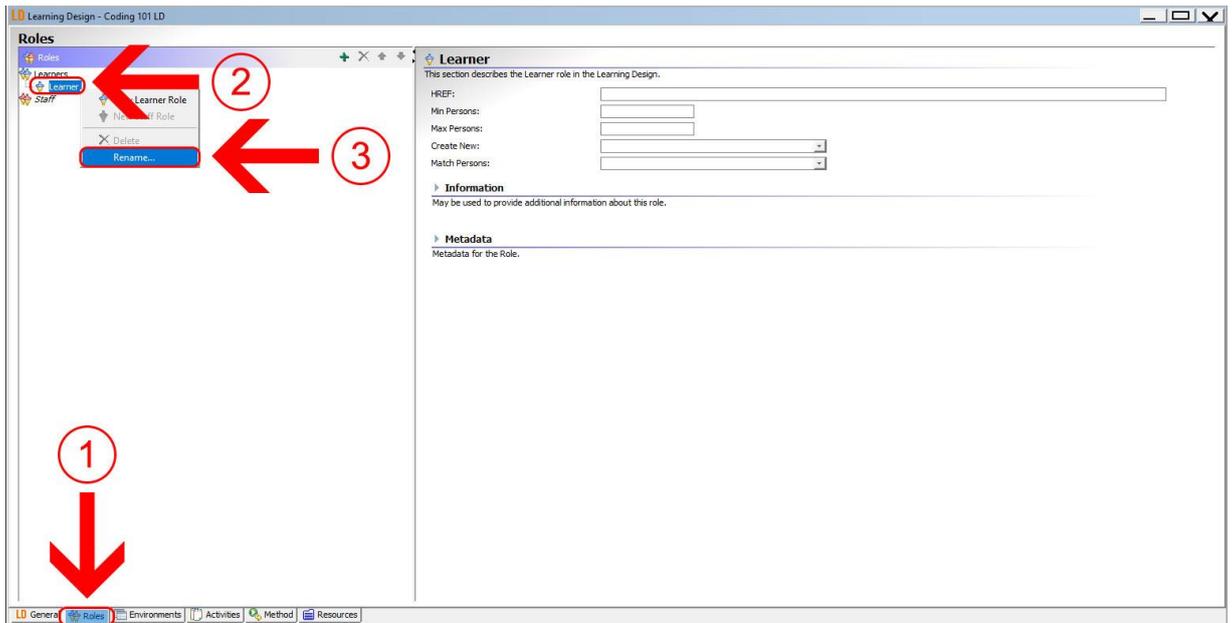
Na tela inicial do editor, as principais áreas utilizadas durante este processo serão as abas localizadas na canto inferior esquerdo da tela, contendo as opções "General" (Geral), "Roles" (Papéis), "Environments" (Ambientes), "Activities" (Atividades), "Method" (Método) e "Resources" (Recursos).



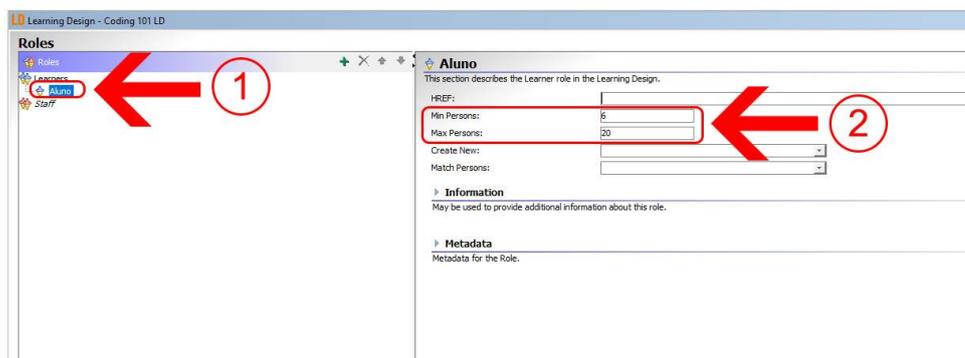
Criado o arquivo de *IMS Learning Design*, a primeira ação tomada será alterar o nome de nosso LD, para registro no cabeçalho do arquivo XML. Para isso, dentro da aba "General", iremos clicar em "Overview", e preencher os campos "Title" com o nome de nosso curso (*Coding 101*) e clicar na campo "Author's Note", para digitar um descritivo do curso proposto neste arquivo. Os campos de Metadados não serão preenchidos neste roteiro, pois tratam de normas definidas no padrão *IMS LRM* para padronização de metadados, fora do escopo deste documento.



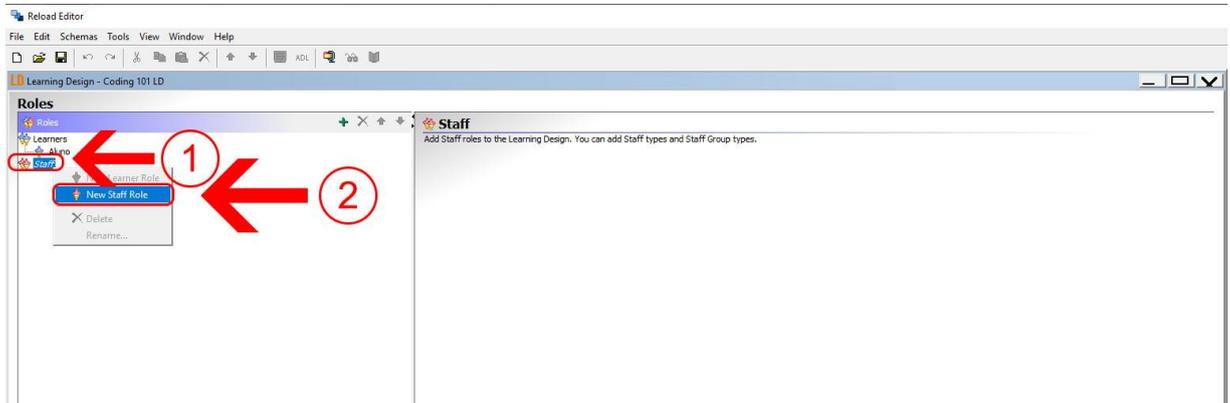
Clique agora na aba "Roles", para definir os papéis possíveis de serem assumidos pelos participantes durante o processo de aprendizagem de nosso curso. Podemos visualizar que existe um papel de aprendiz com o nome "Learner", e nenhum papel existe na hierarquia de equipe da instituição de ensino ("Staff"). Primeiramente, vamos renomear o papel com o nome "Learner" para "Aluno", clicando sobre ele com o botão direito do mouse e depois em "Rename".



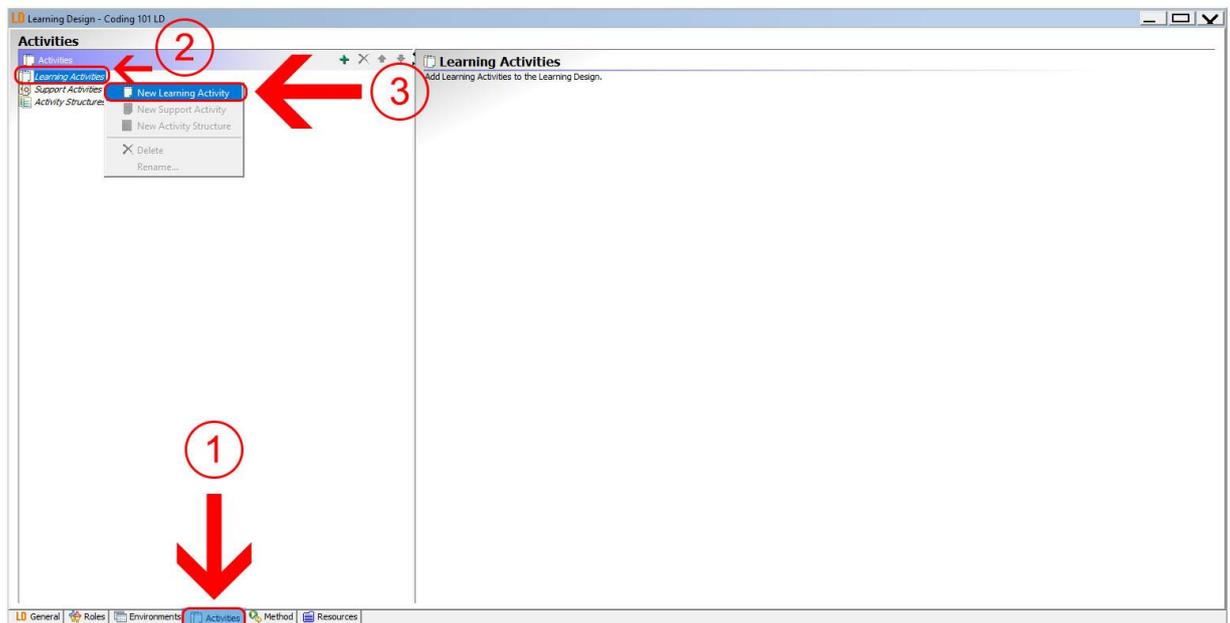
Para definir também o número mínimo e máximo de alunos para nossas atividades, selecionando o papel "Aluno", alterando os valores de "Min Persons" e "Max Persons"



Vamos agora criar o primeiro papel da equipe da instituição de ensino, o papel de "Instrutor", que será responsável por ministrar as aulas e avaliar o desempenho dos alunos. Clique com o botão direito do mouse em "Staff", e em seguida em "New Staff Role", em seguida, escolha o nome do papel desejado.

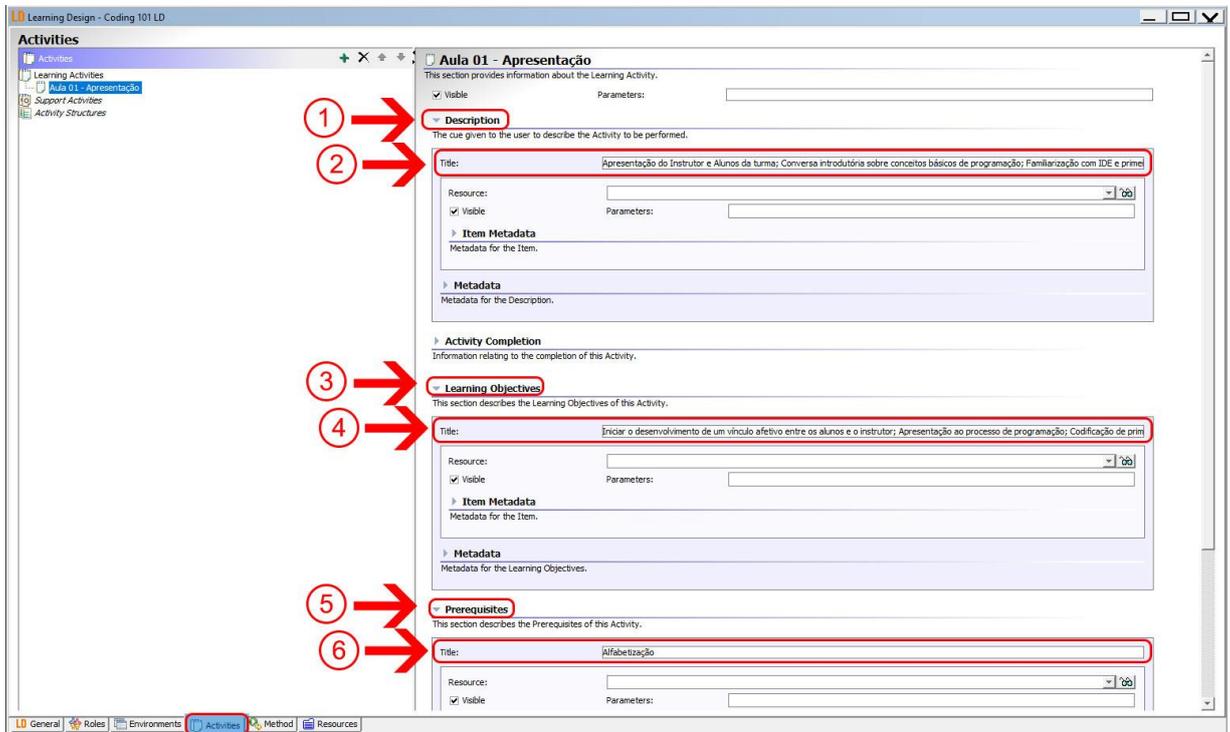


Para criar a nossa primeira atividade, iremos para a aba "Activities". Aqui iremos criar nossa primeira atividade de aprendizagem clicando com o botão direito do mouse em "Learning Activities", e em seguida em "New Learning Activity"

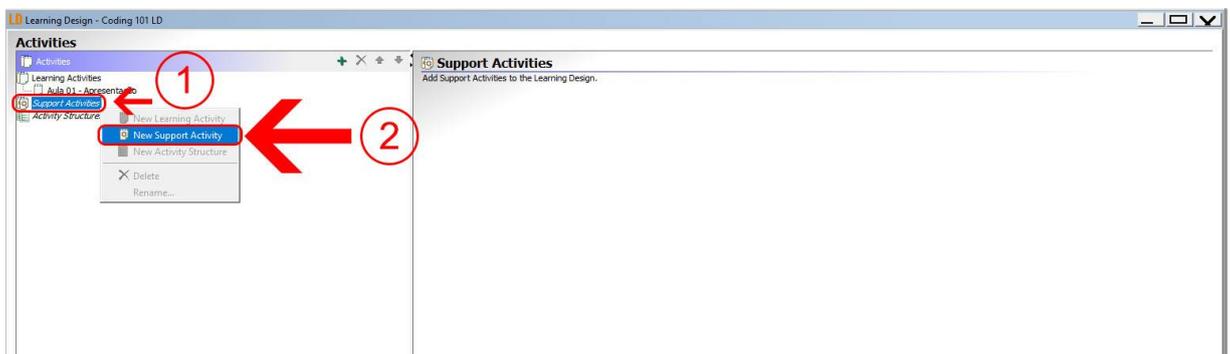


Selecionando a atividade criada, vamos agora atribuir alguns valores para ela, nos campos "Description" com a descrição da atividade, "Learning Objectives" com os

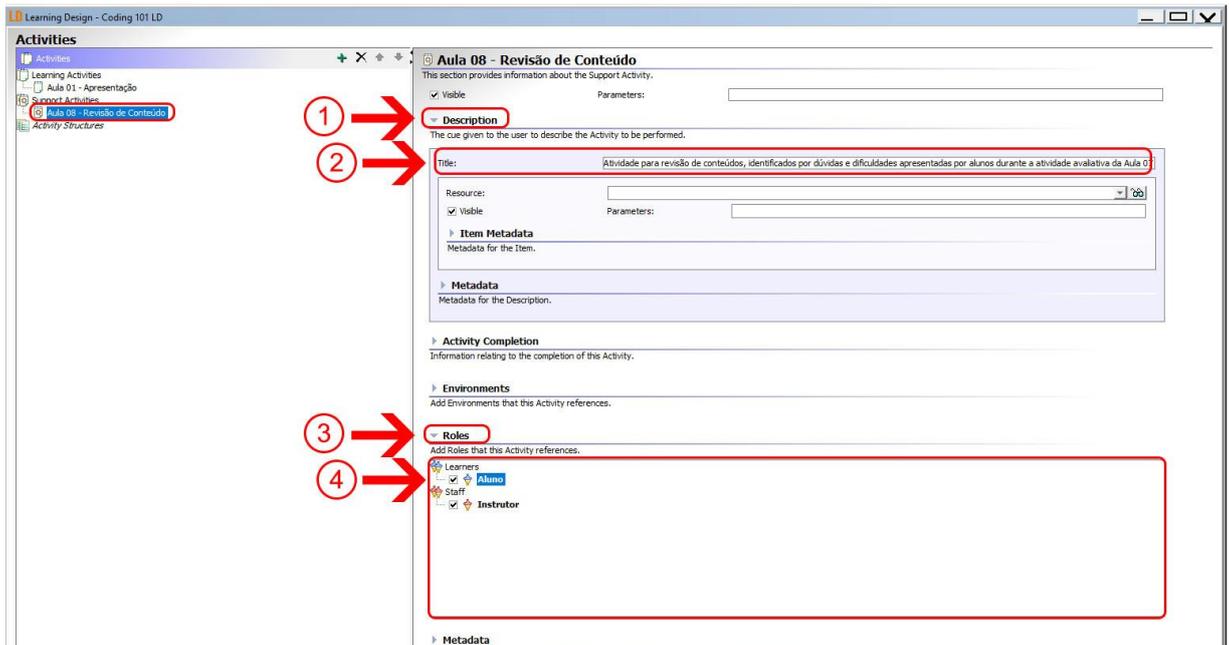
objetivos de aprendizagem da tarefa, e "Prerequisites", com os pré requisitos para participação da atividade



Iremos repetir o processo para todas as atividades do curso. Uma das atividades, a Aula 08 - Revisão de Conteúdo, é uma atividade de suporte. Só irá acontecer caso o resultado de entendimento dos conceitos estudados pela turma durante a avaliação não for positiva. Vamos então criar uma Atividade de Suporte, ainda na aba "Activities". O processo é semelhante à criação de atividades de aprendizagem feitas anteriormente. Basta clicar com o botão direito do mouse em "Support Activities", e em seguida em "New Support Activity"



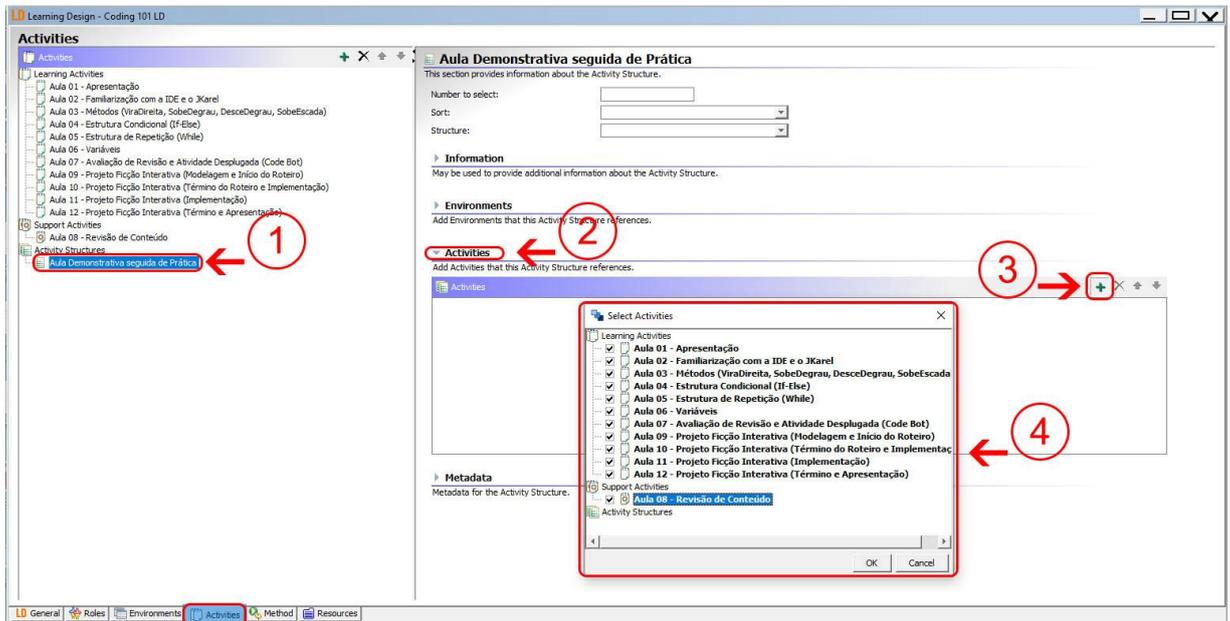
Novamente, como feito com a atividade de aprendizagem, iremos definir atributos para esta atividade. Atividades de suporte solicitam definição dos participantes por papéis, vamos marcar o Aluno e o Instrutor para esta atividade. Atividades de suporte são boas categorias para atividades como debates, onde os alunos são separados em grupos ou interpretando diferentes papéis, como por exemplo representando países em uma simulação de debate entre nações, podendo ser criados papéis diferentes para cada país/grupo.



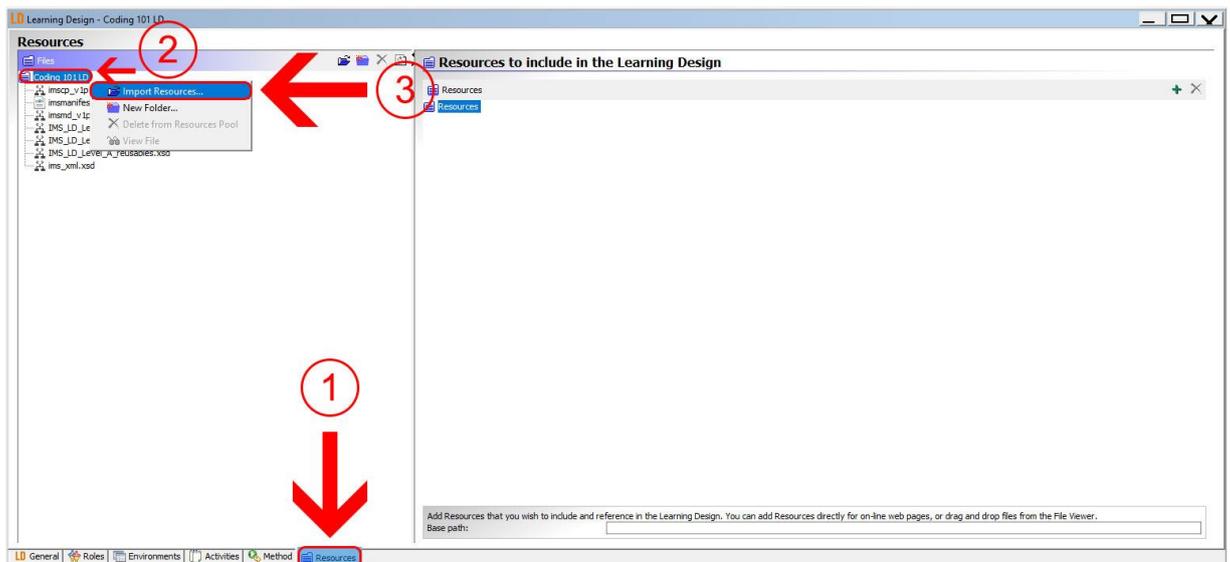
Criaremos agora uma estrutura para relacionar as nossas atividades. Este curso utilizará a estrutura de Aula Expositiva seguida de Prática. Essa estrutura será atribuída a todas as atividades previstas para o curso. Clique em "Activity Structures" com o botão direito do mouse, e em seguida em "Create"



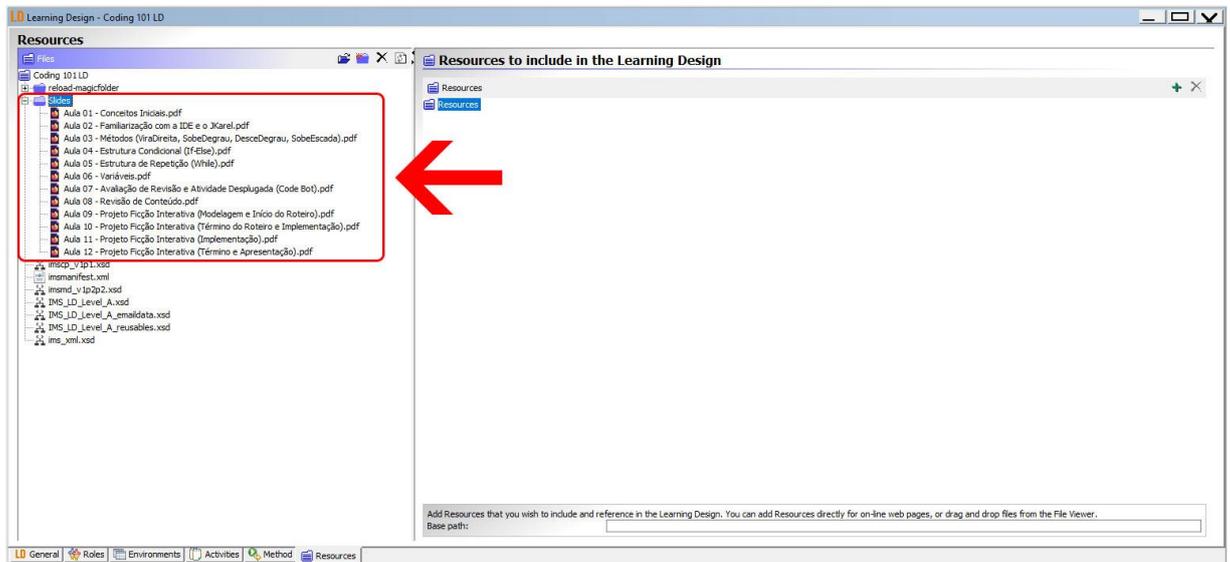
Vamos agora atribuir esta estrutura para todas as atividades. Basta selecionar a estrutura criada, e no campo "Activities", clicar no símbolo "+" para atribuir às atividades.



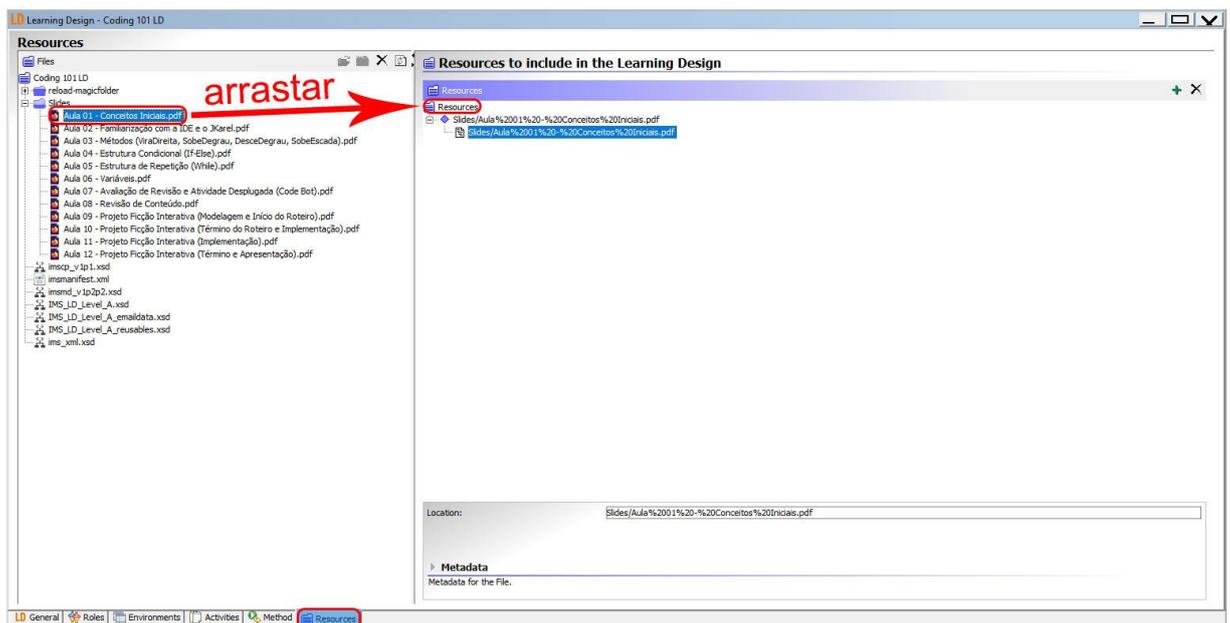
Vamos agora importar recursos para atribuir às atividades criadas. Na aba "Resources", clique com o botão direito do mouse no nome do LD, "Coding 101" em nosso exemplo, e clique em "Import Resources". Em seguida, escolha a pasta que contém os arquivos desejados. No exemplo, serão importados os arquivos no formato PDF referentes aos slides usados para as atividades criadas.



Repare que todos os *slides* foram importados com sucesso.

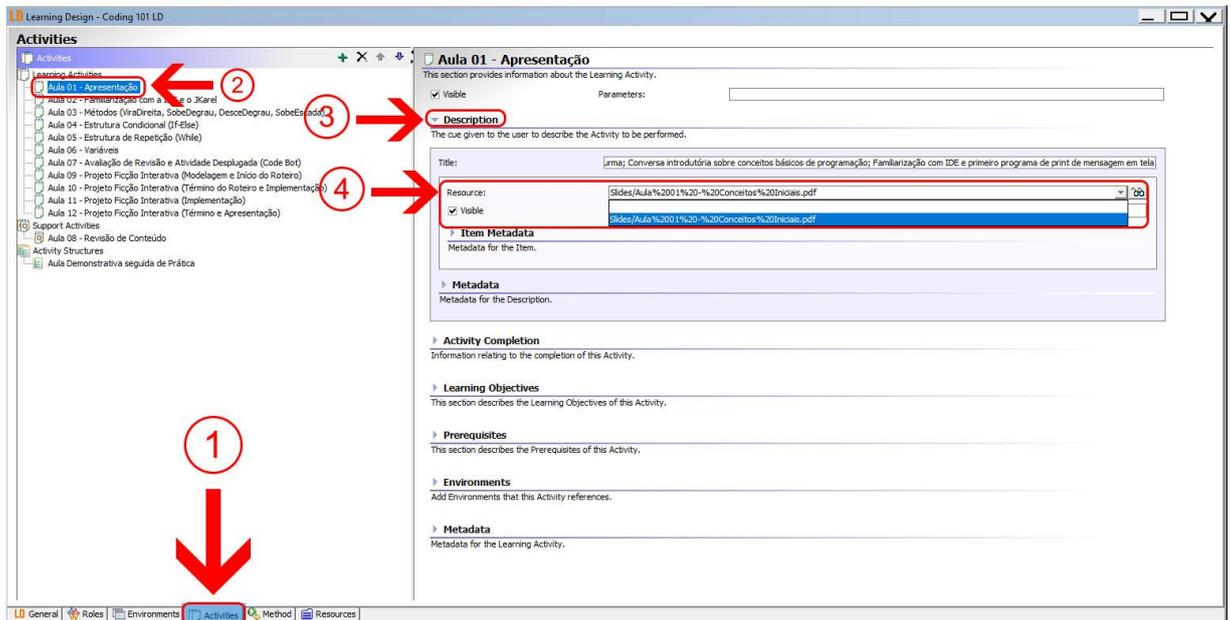


Antes de alocar Recursos a Atividades, precisamos incluir estes recursos em nosso LD. Para isso, basta arrastar o recurso desejado para o LD conforme imagem.

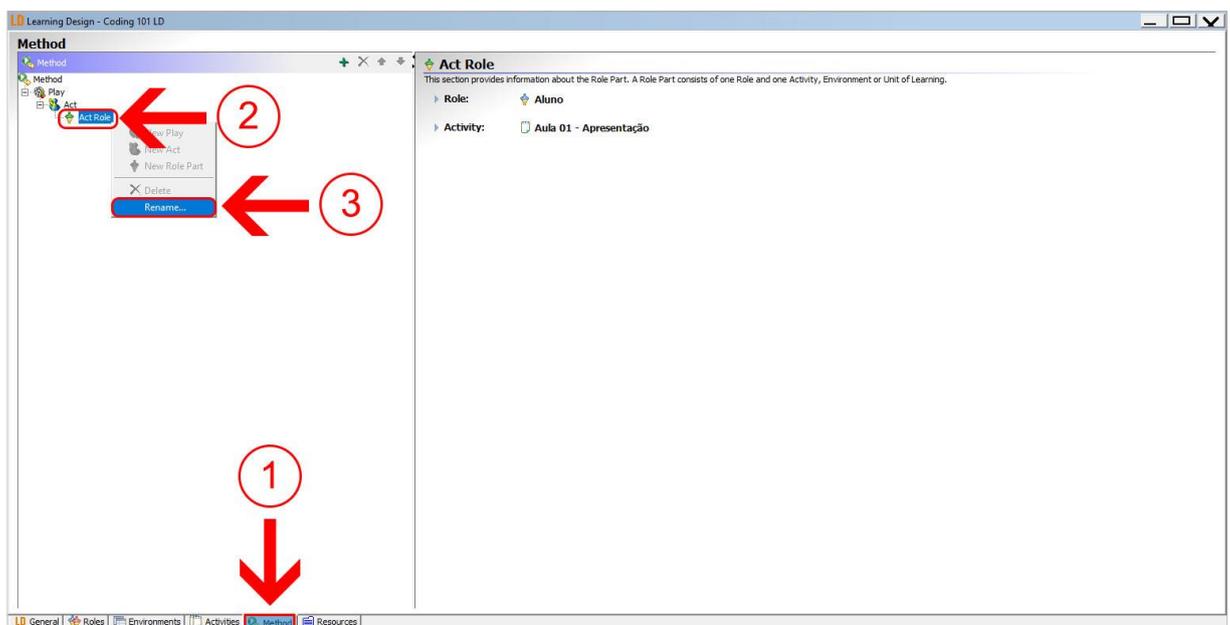


Vamos agora atribuir o arquivo de slides referente à Aula 01. Retorne à aba "Activities", selecione a atividade que deseja atribuir o recurso importado. Em seguida,

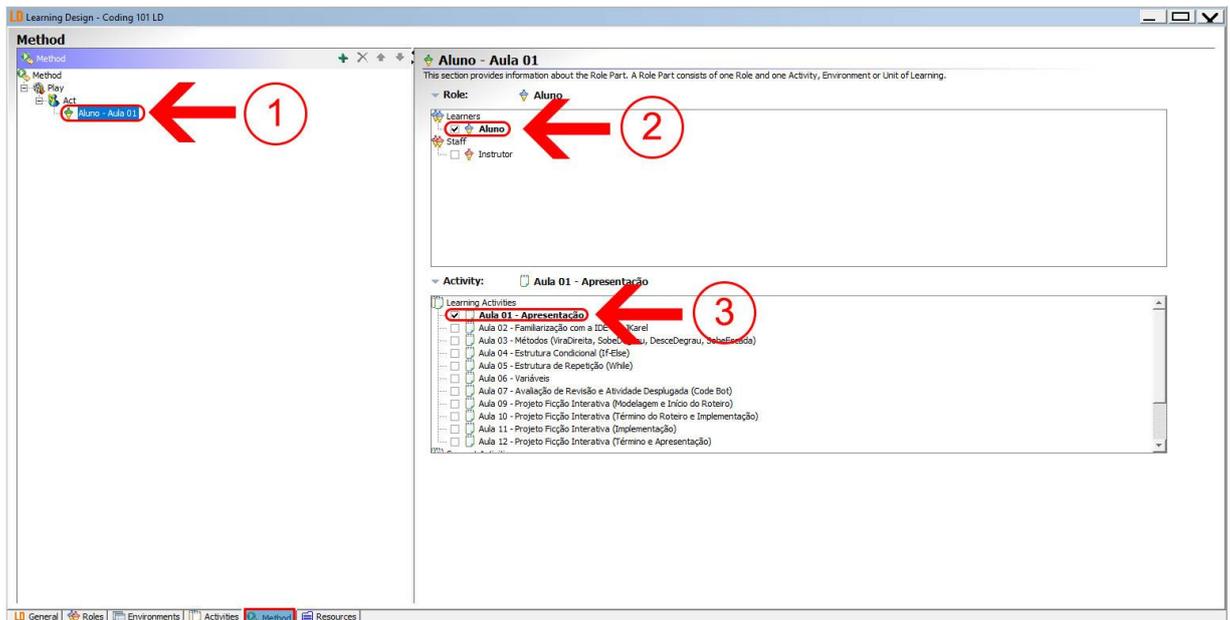
em "Description", selecione o recurso desejado. Vamos repetir esse processo com todas as atividades.



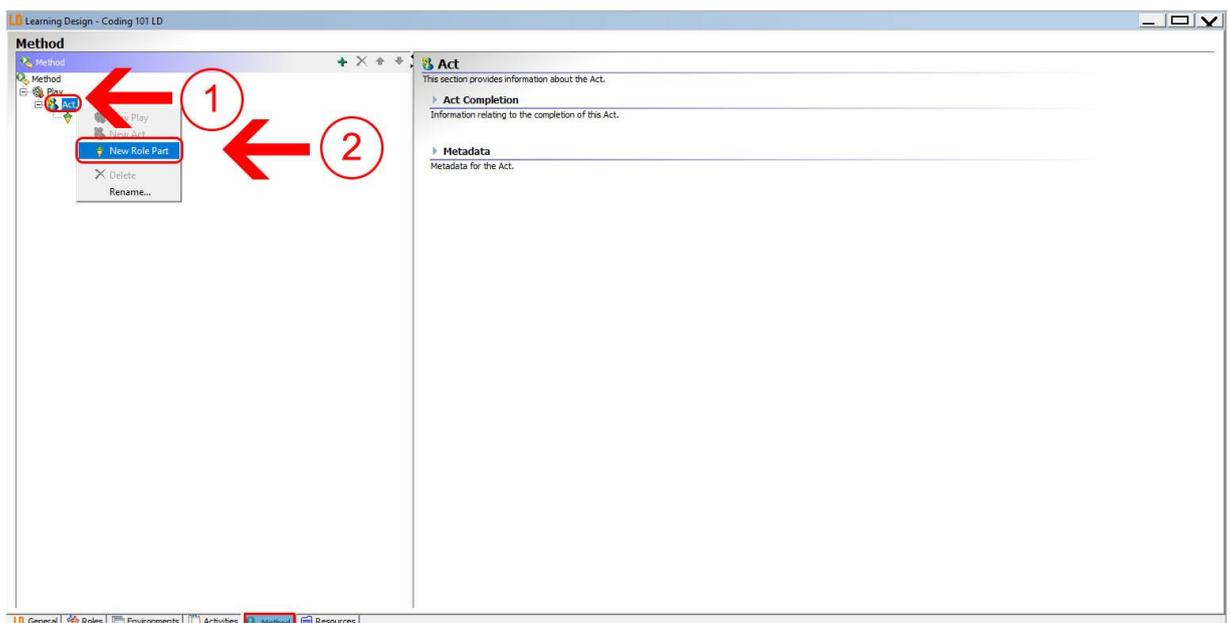
Agora, na aba "Method", iremos relacionar os papéis e atividades em relação 1 para 1, de acordo com atos que iremos definir. Para isso, vamos começar renomeando a relação já existente Ato x Papel. Clique com o botão direito em "Act Role", em seguida em "Rename", e altere o nome para "Aula 01 - Aluno".



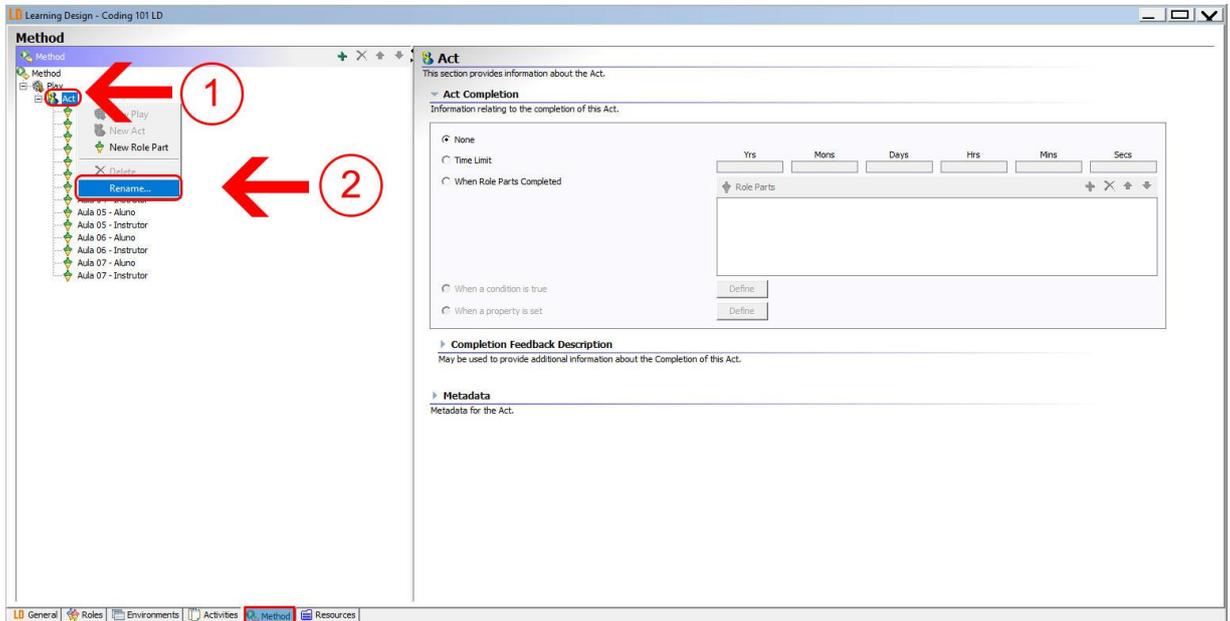
Vamos agora criar a relação entre o papel Aluno e a atividade Aula 01. Selecione “Aula 01 - Aluno”, e em seguida no campo à direita, selecione as caixas com o papel “Aluno” e a atividade “Aula 01 - Introdução”.



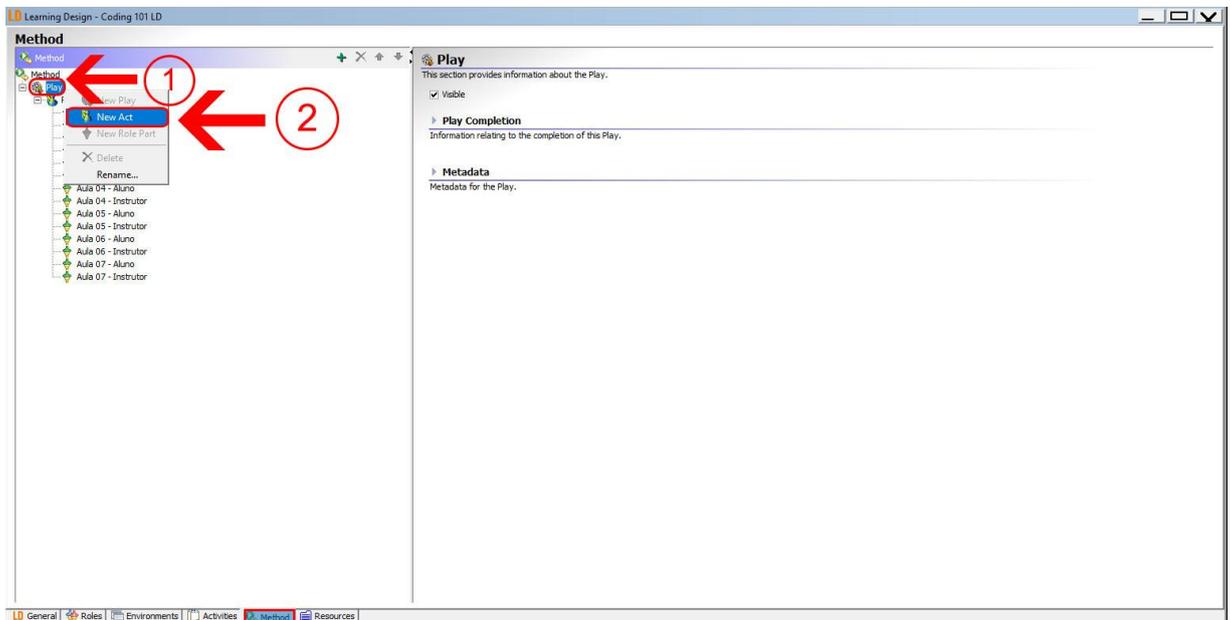
Vamos agora criar novas relações entre as atividades Aula 02 até Aula 07 com o papel “Aluno”, e as relações dessas atividades com o papel “Instrutor”. Clique com o botão direito em “Act”, e em seguida “New Role Part”.



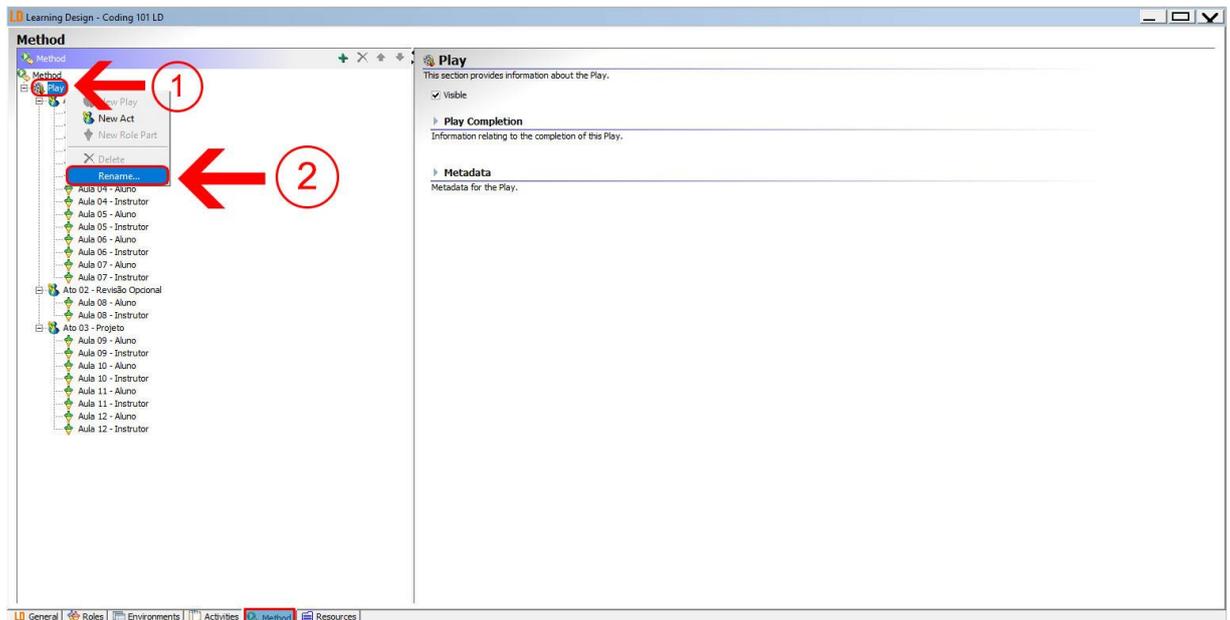
Vamos agora renomear o ato de “Act” para “Ato 01 - Aulas Conceituais”. Clique com o botão direito do mouse em “Act”, e em seguida “Rename”.



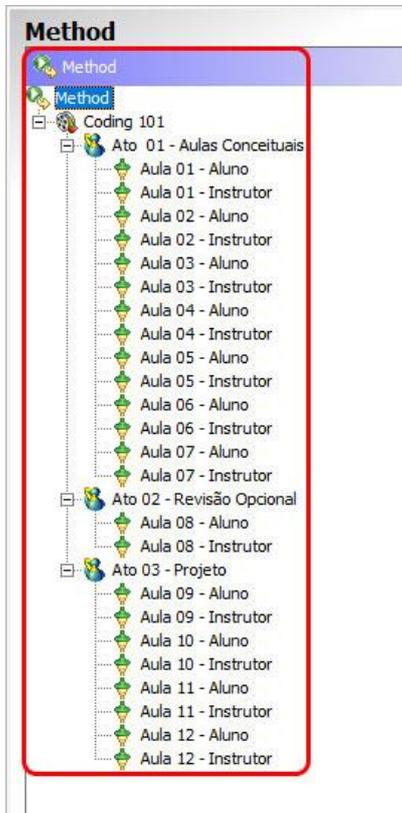
Vamos criar os outros atos que compõem nosso curso: “Ato 02 - Revisão Opcional” e “Ato 03 - Projeto”, clicando com o botão direito em “Play”, e em “New Act”.



Após adicionar todas as relações dos papéis “Aluno” e “Instrutor” com as aulas restantes, vamos renomear a peça “Play” clicando com o botão direito sobre ela, e selecionando “Rename”.



A estrutura final do curso em *IMS LD* ficará como a figura abaixo.



Este curso não fará uso de método ou ambiente externo de serviços, mas o processo de criação destes é semelhante ao processo de criação de atividades e papéis, caso o leitor queira utilizar estes recursos. Salve o *LD* para que atualize o arquivo "*imsmanifest.xml*" na pasta raiz do *LD*, e podemos observar que nosso manifesto já inclui toda a estrutura do curso, com papéis, estrutura de atividades, juntamente com suas descrições e recursos.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!--This is a Reload version 2.0 Learning Design document-->
3 <!--Spawned from the Reload Learning Design Generator - http://www.reload.ac.uk-->
4 <manifest xmlns="http://www.imsglobal.org/xsd/imsdp_v1p1" xmlns:imsmd=
  "http://www.imsglobal.org/xsd/imsmd_v1p2" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:imsld="http://www.imsglobal.org/xsd/imsld_v1p0" identifier=
  "MANIFEST-352DB8B2-1B2B-9D1B-CA02-1A217666452D" xsi:schemaLocation=
  "http://www.imsglobal.org/xsd/imsdp_v1p1 imscp_v1p1.xsd http://www.imsglobal.org/xsd/imsmd_v1p2
  imsmd_v1p2p2.xsd http://www.imsglobal.org/xsd/imsld_v1p0 IMS_LD_Level_A.xsd">
5 <organizations>
6 <imsld:learning-design identifier="LD-D6427229-D1F5-5660-A797-5A060082B273" level="A" uri=""
  version="1.0">
7 <imsld:title>Coding 101</imsld:title>
8 <imsld:learning-objectives>
9 <imsld:item />
10 <imsld:metadata />
11 </imsld:learning-objectives>
12 <imsld:components>
13 <imsld:roles identifier="LD-303E19DA-FB70-81F0-45D9-AA521196A2C2">
14 <imsld:learner identifier="LD-5F01C274-26F6-191A-0A56-80D2911828C4" max-persons="20"
  min-persons="6">
15 <imsld:title>Aluno</imsld:title>
16 </imsld:learner>
17 <imsld:staff identifier="LD-009A559B-61C8-240C-B00A-5E2CF81646D6">
18 <imsld:title>Instrutor</imsld:title>
19 </imsld:staff>
20 </imsld:roles>
21 <imsld:activities>
22 <imsld:learning-activity identifier="LD-428ED59D-676E-F5B6-99B7-696439304B7C">
23 <imsld:title>Aula 01 - Apresentação</imsld:title>
24 <imsld:learning-objectives>
25 <imsld:title>Iniciar o desenvolvimento de um vínculo afetivo entre os alunos e o
  instrutor; Apresentação ao processo de programação, conceitos de algoritmo e
  seqüenciamento de instruções; Codificação de primeiro programa simples pelos alunos
  </imsld:title>
26 <imsld:item />
27 </imsld:learning-objectives>
28 <imsld:prerequisites>
29 <imsld:title>Aluno alfabetizado, com facilidade em leitura e escrita</imsld:title>
30 <imsld:item />
31 </imsld:prerequisites>
32 <imsld:activity-description>
33 <imsld:title>Apresentação do Instrutor e Alunos da turma; Conversa introdutória sobre
  conceitos básicos de programação; Familiarização com IDE e primeiro programa de print de
  mensagem em tela</imsld:title>
34 <imsld:item identifierref="RES-89F46E1B-757F-908C-E161-1EC411C2E90A" />
35 </imsld:activity-description>
36 </imsld:learning-activity>
37 <imsld:learning-activity identifier="LD-7DB657BC-5F37-76F6-718C-B81F876B0921">
38 <imsld:title>Aula 02 - Familiarização com a IDE e o JKarel</imsld:title>
39 <imsld:learning-objectives>
40 <imsld:title>Aprender as instruções básicas do JKarel: Mover, Desligar, Virar para a
  Direita, Pegar e Depositar Objeto</imsld:title>
41 <imsld:item />
42 </imsld:learning-objectives>
43 <imsld:prerequisites>
44 <imsld:title />
45 <imsld:item />
46 </imsld:prerequisites>
47 <imsld:activity-description>
48 <imsld:title>Familiarização com a IDE e a biblioteca de classes JKarel; Relizar
  seqüência de atividades utilizando o JKarel</imsld:title>
49 <imsld:item identifierref="RES-2627A423-189C-B267-F152-EE383799B383" />
50 </imsld:activity-description>
51 </imsld:learning-activity>
52 <imsld:learning-activity identifier="LD-90934B6F-4CDB-319A-CD3C-E807EA6D3A21">
53 <imsld:title>Aula 03 - Métodos (ViraDireita, SobeDegrau, DesceDegrau, SobeEscada)
  </imsld:title>
54 <imsld:learning-objectives>
55 <imsld:title>Encapsulamento de conjunto de instruções em novos métodos para tarefas
  repetitivas</imsld:title>
56 <imsld:item />
57 </imsld:learning-objectives>
58 <imsld:activity-description>
59 <imsld:title>Prática de programação usando JKarel. Introdução ao conceito de criação de
  novos métodos para realização de tarefas repetitivas</imsld:title>
60 <imsld:item identifierref="RES-ADE28659-9276-FD3E-F188-225F0C4B454C" />
61 </imsld:activity-description>
62 </imsld:learning-activity>
63 <imsld:learning-activity identifier="LD-0969FE91-63B6-DA9E-EE0D-FB4C11F7C56E">
64 <imsld:title>Aula 04 - Estrutura Condicional (If-Else)</imsld:title>
65 <imsld:learning-objectives>
66 <imsld:title>Utilização de estruturas condicionais para desenvolvimento de algoritmos
  adaptáveis</imsld:title>
67 <imsld:item />
68 </imsld:learning-objectives>
69 <imsld:activity-description>
70 <imsld:title>Apresentação e prática do conceito de estrutura condicional (If / Else)
  utilizando JKarel</imsld:title>
71 <imsld:item identifierref="RES-19AD44B1-A70D-95CB-5FE0-D04250E853E7" />

```

```

72 </imsld:activity-description>
73 </imsld:learning-activity>
74 <imsld:learning-activity identifier="LD-FCB67B9F-EC19-D4A1-BDD8-CBDEA59564CB">
75 <imsld:title>Aula 05 - Estrutura de Repetição (While)</imsld:title>
76 <imsld:learning-objectives>
77 <imsld:title>Utilização de laços de repetição para resolução de tarefas que contenham
redundância de instruções</imsld:title>
<imsld:item />
78 </imsld:learning-objectives>
79 <imsld:activity-description>
80 <imsld:title>Apresentação e prática do conceito de estrutura de repetição (While)
utilizando JKarel</imsld:title>
81 <imsld:item identifierref="RES-7B742F33-3618-63C3-B995-8DC7DEAC9EE9" />
82 </imsld:activity-description>
83 </imsld:learning-activity>
84 <imsld:learning-activity identifier="LD-455ACD0B-EEE2-0686-9DCE-EDF09ECD2C58">
85 <imsld:title>Aula 06 - Variáveis</imsld:title>
86 <imsld:learning-objectives>
87 <imsld:title>Utilização de variáveis para realização de tarefas que precisem de
armazenamento de dados</imsld:title>
88 <imsld:item />
89 </imsld:learning-objectives>
90 <imsld:activity-description>
91 <imsld:title>Apresentação e prática do conceito de variáveis e seus diferentes tipos
utilizando JKarel</imsld:title>
92 <imsld:item identifierref="RES-53D25989-8D39-CD58-86FC-A4957E75595E" />
93 </imsld:activity-description>
94 </imsld:learning-activity>
95 <imsld:learning-activity identifier="LD-22B8FB74-99A0-544E-CEB9-9D65ADA7DB41">
96 <imsld:title>Aula 07 - Avaliação de Revisão e Atividade Desplugada (Code Bot)</imsld:title>
97 <imsld:learning-objectives>
98 <imsld:title>Avaliação de aprendizagem dos alunos; Definir a necessidade de aula de
revisão de conceitos antes do início do projeto</imsld:title>
99 <imsld:item />
100 </imsld:learning-objectives>
101 <imsld:activity-description>
102 <imsld:title>Realizar atividade avaliativa em turma na forma de questões de múltipla
escolha para detecção do ritmo de aprendizagem da turma, identificar dúvidas e
dificuldades que necessitem de revisão; Atividade desplugada para prática dos conceitos
estudados em método tático sem uso de computadores</imsld:title>
103 <imsld:item identifierref="RES-336E84ED-2886-9986-5627-C86522803830" />
104 </imsld:activity-description>
105 </imsld:learning-activity>
106 <imsld:learning-activity identifier="LD-14812750-C055-E2E1-44E8-236A9ED56374">
107 <imsld:title>Aula 09 - Projeto Ficção Interativa (Modelagem e Início do Roteiro)
108 <imsld:learning-objectives>
109 <imsld:title>Prática de atividade de modelagem de árvore de decisões que compõem o jogo
</imsld:title>
<imsld:item />
110 </imsld:learning-objectives>
111 <imsld:activity-description>
112 <imsld:title>Iniciar a modelagem e escrita de roteiro de um jogo de ficção interativa,
utilizando apenas texto para criar um livro-jogo eletrônico</imsld:title>
113 <imsld:item identifierref="RES-FDD6C472-38CC-B0A7-DB33-867DF2687E91" />
114 </imsld:activity-description>
115 </imsld:learning-activity>
116 <imsld:learning-activity identifier="LD-24A30AD2-697E-BC86-08C1-CEE084D380E5">
117 <imsld:title>Aula 10 - Projeto Ficção Interativa (Término do Roteiro e Implementação)
118 <imsld:learning-objectives>
119 <imsld:title>Praticar aplicação dos conceitos vistos em aula, principalmente estrutura
condicional; apresentação e aplicação do conceito de entrada de dados pelo usuário
</imsld:title>
<imsld:item />
120 </imsld:learning-objectives>
121 <imsld:activity-description>
122 <imsld:title>Finalizar a escrita do roteiro e iniciar a programação do jogo de ficção
interativa</imsld:title>
123 <imsld:item identifierref="RES-0CDBF08A-F531-C3EF-31FC-048439125006" />
124 </imsld:activity-description>
125 </imsld:learning-activity>
126 <imsld:learning-activity identifier="LD-74E37B10-FEEA-EEE2-1A5C-E237F76201D4">
127 <imsld:title>Aula 11 - Projeto Ficção Interativa (Implementação)</imsld:title>
128 <imsld:learning-objectives>
129 <imsld:title>Praticar aplicação dos conceitos vistos em aula, principalmente estrutura
condicional</imsld:title>
<imsld:item />
130 </imsld:learning-objectives>
131 <imsld:activity-description>
132 <imsld:title>Dar continuidade à programação do jogo de ficção interativa</imsld:title>
133 <imsld:item identifierref="RES-E95F4EBC-D6C7-E617-A961-E71133CDC748" />
134 </imsld:activity-description>
135 </imsld:learning-activity>
136 <imsld:learning-activity identifier="LD-F7E0DFD9-380E-026D-8D16-E819A99E6921">
137 <imsld:title>Aula 12 - Projeto Ficção Interativa (Término e Apresentação)</imsld:title>
138 <imsld:learning-objectives>
139 <imsld:title>Praticar aplicação dos conceitos vistos em aula; prática da habilidade de
apresentação de trabalhos em sala de aula</imsld:title>
140 <imsld:item />
141 </imsld:learning-objectives>
142 </imsld:learning-activity>

```

```

146 <imsld:activity-description>
147 <imsld:title>Terminar a implementação do projeto; Alunos realizarem apresentação curta,
em formato de discurso de vendas do projeto desenvolvido; Possibilitar que os alunos
joguem os jogos de seus colegas de sala para troca de experiências</imsld:title>
148 <imsld:item identifierrref="RES-E450CCD6-E869-1C32-D67A-3F30FE074567" />
149 </imsld:activity-description>
150 </imsld:learning-activity>
151 <imsld:support-activity identifier="LD-4C32C8C2-7AA9-D5F2-4955-12BE401D3164">
152 <imsld:title>Aula 08 - Revisão de Conteúdo</imsld:title>
153 <imsld:role-ref ref="LD-009A559B-61C8-240C-B00A-5E2CF81646D6" />
154 <imsld:role-ref ref="LD-5F01C274-26F6-191A-0A56-80D2911828C4" />
155 <imsld:activity-description>
156 <imsld:title>Atividade para revisão de conteúdos, identificados por dúvidas e
dificuldades apresentadas por alunos durante a atividade avaliativa da Aula 07
</imsld:title>
<imsld:item identifierrref="RES-DD7EB90E-9910-1E55-C8AE-29DE061591A1" />
</imsld:activity-description>
</imsld:support-activity>
159 <imsld:activity-structure identifier="LD-C74D2DB4-1EC3-DB92-F3C3-C07271106997">
160 <imsld:title>Aula Demonstrativa seguida de Prática</imsld:title>
161 <imsld:learning-activity-ref ref="LD-428ED59D-676E-F5B6-99B7-696439304B7C" />
162 <imsld:learning-activity-ref ref="LD-7DB657BC-5F37-76F6-718C-B81F876B0921" />
163 <imsld:learning-activity-ref ref="LD-90934B6F-4CDB-319A-CD3C-E807EA6D3A21" />
164 <imsld:learning-activity-ref ref="LD-0969FE91-63B6-DA9E-EE0D-FB4C11F7C56E" />
165 <imsld:learning-activity-ref ref="LD-FCB67B9F-EC19-D4A1-BDD8-CBDEA59564CB" />
166 <imsld:learning-activity-ref ref="LD-455ACD0B-EEE2-0686-9DCE-EDF09ECD2C58" />
167 <imsld:learning-activity-ref ref="LD-22B8F874-99A0-544E-CEB9-9D65ADA7DB41" />
168 <imsld:learning-activity-ref ref="LD-14812750-C055-E2E1-44E8-236A9ED56374" />
169 <imsld:learning-activity-ref ref="LD-2A30AD2-697E-BC86-08C1-CEE084D380E5" />
170 <imsld:learning-activity-ref ref="LD-74E37B10-FEEA-EEE2-1A5C-E237F76201D4" />
171 <imsld:learning-activity-ref ref="LD-F7E0DFD9-380E-026D-8D16-E819A996921" />
172 <imsld:support-activity-ref ref="LD-4C32C8C2-7AA9-D5F2-4955-12BE401D3164" />
173 </imsld:activity-structure>
174 </imsld:activities>
175 </imsld:components>
176 </imsld:method>
177 <imsld:play identifier="LD-7E533C4C-2F70-B622-836D-055F64FF377C">
178 <imsld:title>Coding 101</imsld:title>
179 <imsld:act identifier="LD-FD4F9CC9-06ED-BE43-027B-BAC271FF367F">
180 <imsld:title>Ato 01 - Aulas Conceituais</imsld:title>
181 <imsld:role-part identifier="LD-95A539C9-C942-345C-0271-04F89AD9C13E">
182 <imsld:title>Aula 01 - Aluno</imsld:title>
183 <imsld:role-ref ref="LD-5F01C274-26F6-191A-0A56-80D2911828C4" />
184 <imsld:learning-activity-ref ref="LD-428ED59D-676E-F5B6-99B7-696439304B7C" />
185 </imsld:role-part>
186
187 <imsld:role-part identifier="LD-DEE1E196-C34B-091F-5AE9-0FCACEB2BC6C">
188 <imsld:title>Aula 01 - Instrutor</imsld:title>
189 <imsld:role-ref ref="LD-009A559B-61C8-240C-B00A-5E2CF81646D6" />
190 <imsld:learning-activity-ref ref="LD-428ED59D-676E-F5B6-99B7-696439304B7C" />
191 </imsld:role-part>
192 <imsld:role-part identifier="LD-FB99097B-2757-5B69-859C-31E1D3DF09B9">
193 <imsld:title>Aula 02 - Aluno</imsld:title>
194 <imsld:role-ref ref="LD-5F01C274-26F6-191A-0A56-80D2911828C4" />
195 <imsld:learning-activity-ref ref="LD-7DB657BC-5F37-76F6-718C-B81F876B0921" />
196 </imsld:role-part>
197 <imsld:role-part identifier="LD-AE8DEA8E-9970-4AA3-1C62-3D8CFFFD498E">
198 <imsld:title>Aula 02 - Instrutor</imsld:title>
199 <imsld:role-ref ref="LD-009A559B-61C8-240C-B00A-5E2CF81646D6" />
200 <imsld:learning-activity-ref ref="LD-7DB657BC-5F37-76F6-718C-B81F876B0921" />
201 </imsld:role-part>
202 <imsld:role-part identifier="LD-8F55DD24-ECEF-DFD0-1B16-61263E62626E">
203 <imsld:title>Aula 03 - Aluno</imsld:title>
204 <imsld:role-ref ref="LD-5F01C274-26F6-191A-0A56-80D2911828C4" />
205 <imsld:learning-activity-ref ref="LD-90934B6F-4CDB-319A-CD3C-E807EA6D3A21" />
206 </imsld:role-part>
207 <imsld:role-part identifier="LD-B3DD3A81-B5EC-9166-8DC3-6F5B648A5F6D">
208 <imsld:title>Aula 03 - Instrutor</imsld:title>
209 <imsld:role-ref ref="LD-009A559B-61C8-240C-B00A-5E2CF81646D6" />
210 <imsld:learning-activity-ref ref="LD-90934B6F-4CDB-319A-CD3C-E807EA6D3A21" />
211 </imsld:role-part>
212 <imsld:role-part identifier="LD-44639397-87D1-9C94-D7EA-B6809458F171">
213 <imsld:title>Aula 04 - Aluno</imsld:title>
214 <imsld:role-ref ref="LD-5F01C274-26F6-191A-0A56-80D2911828C4" />
215 <imsld:learning-activity-ref ref="LD-0969FE91-63B6-DA9E-EE0D-FB4C11F7C56E" />
216 </imsld:role-part>
217 <imsld:role-part identifier="LD-F4C04D30-DB74-E807-A5E1-225FBD15D2C5">
218 <imsld:title>Aula 04 - Instrutor</imsld:title>
219 <imsld:role-ref ref="LD-009A559B-61C8-240C-B00A-5E2CF81646D6" />
220 <imsld:learning-activity-ref ref="LD-0969FE91-63B6-DA9E-EE0D-FB4C11F7C56E" />
221 </imsld:role-part>
222 <imsld:role-part identifier="LD-CC2BCA93-1F35-50E7-4AC3-DD9026D65935">
223 <imsld:title>Aula 05 - Aluno</imsld:title>
224 <imsld:role-ref ref="LD-5F01C274-26F6-191A-0A56-80D2911828C4" />
225 <imsld:learning-activity-ref ref="LD-FCB67B9F-EC19-D4A1-BDD8-CBDEA59564CB" />
226 </imsld:role-part>
227 <imsld:role-part identifier="LD-692675C6-CB75-2BB6-D1D9-4C62876D3379">
228 <imsld:title>Aula 05 - Instrutor</imsld:title>
229 <imsld:role-ref ref="LD-009A559B-61C8-240C-B00A-5E2CF81646D6" />
230 <imsld:learning-activity-ref ref="LD-FCB67B9F-EC19-D4A1-BDD8-CBDEA59564CB" />
231 </imsld:role-part>

```

```

232 <imsld:role-part identifier="LD-7789D078-DEA5-64D0-0450-19E231BFF75A">
233 <imsld:title>Aula 06 - Aluno</imsld:title>
234 <imsld:role-ref ref="LD-5F01C274-26F6-191A-0A56-80D2911828C4" />
235 <imsld:learning-activity-ref ref="LD-455ACD0B-EEE2-0686-9DCE-EDF09ECD2C58" />
236 </imsld:role-part>
237 <imsld:role-part identifier="LD-10B08521-7E41-18B1-DD2A-AD232F09DC2F">
238 <imsld:title>Aula 06 - Instrutor</imsld:title>
239 <imsld:role-ref ref="LD-009A559B-61C8-240C-B00A-5E2CF81646D6" />
240 <imsld:learning-activity-ref ref="LD-455ACD0B-EEE2-0686-9DCE-EDF09ECD2C58" />
241 </imsld:role-part>
242 <imsld:role-part identifier="LD-ACA38641-4DE1-C17A-B600-641F1D94DC">
243 <imsld:title>Aula 07 - Aluno</imsld:title>
244 <imsld:role-ref ref="LD-5F01C274-26F6-191A-0A56-80D2911828C4" />
245 <imsld:learning-activity-ref ref="LD-22B8F874-99A0-544E-CEB9-9D65ADA7DB41" />
246 </imsld:role-part>
247 <imsld:role-part identifier="LD-C55BABC4-36F6-3EAE-0E19-51CA835660E3">
248 <imsld:title>Aula 07 - Instrutor</imsld:title>
249 <imsld:role-ref ref="LD-009A559B-61C8-240C-B00A-5E2CF81646D6" />
250 <imsld:learning-activity-ref ref="LD-22B8F874-99A0-544E-CEB9-9D65ADA7DB41" />
251 </imsld:role-part>
252 </imsld:act>
253 <imsld:act identifier="LD-7FA99957-089B-6348-21D2-7B597B6EA6C7">
254 <imsld:title>Ato 02 - Revisão Opcional</imsld:title>
255 <imsld:role-part identifier="LD-B88AFPE9-EDFF-D114-321E-54568402CC78">
256 <imsld:title>Aula 08 - Aluno</imsld:title>
257 <imsld:role-ref ref="LD-5F01C274-26F6-191A-0A56-80D2911828C4" />
258 <imsld:support-activity-ref ref="LD-4C32C8C2-7AA9-D5F2-4955-12BE401D3164" />
259 </imsld:role-part>
260 <imsld:role-part identifier="LD-A570FOCF-1791-D3FE-D256-F96F257462EE">
261 <imsld:title>Aula 08 - Instrutor</imsld:title>
262 <imsld:role-ref ref="LD-009A559B-61C8-240C-B00A-5E2CF81646D6" />
263 <imsld:support-activity-ref ref="LD-4C32C8C2-7AA9-D5F2-4955-12BE401D3164" />
264 </imsld:role-part>
265 </imsld:act>
266 <imsld:act identifier="LD-C08CB26A-E546-0A37-D766-DC03E33AC1C1">
267 <imsld:title>Ato 03 - Projeto</imsld:title>
268 <imsld:role-part identifier="LD-C7C7E4DB-0E39-0C41-178D-4B205F28A246">
269 <imsld:title>Aula 09 - Aluno</imsld:title>
270 <imsld:role-ref ref="LD-5F01C274-26F6-191A-0A56-80D2911828C4" />
271 <imsld:learning-activity-ref ref="LD-14812750-C055-E2E1-44E8-236A9ED56374" />
272 </imsld:role-part>
273 <imsld:role-part identifier="LD-84B99B95-2F3C-10E3-8648-3A1C5A73DDDC">
274 <imsld:title>Aula 09 - Instrutor</imsld:title>
275 <imsld:role-ref ref="LD-009A559B-61C8-240C-B00A-5E2CF81646D6" />
276 <imsld:learning-activity-ref ref="LD-14812750-C055-E2E1-44E8-236A9ED56374" />
277 </imsld:role-part>
278 <imsld:role-part identifier="LD-5AAA92CD-E6F7-83E6-067D-247295276483">
279 <imsld:title>Aula 10 - Aluno</imsld:title>
280 <imsld:role-ref ref="LD-5F01C274-26F6-191A-0A56-80D2911828C4" />
281 <imsld:learning-activity-ref ref="LD-24A30AD2-697E-BC86-08C1-CEE084D380E5" />
282 </imsld:role-part>
283 <imsld:role-part identifier="LD-5828D815-7B51-D906-EDAA-0DB41AF8E1C7">
284 <imsld:title>Aula 10 - Instrutor</imsld:title>
285 <imsld:role-ref ref="LD-009A559B-61C8-240C-B00A-5E2CF81646D6" />
286 <imsld:learning-activity-ref ref="LD-24A30AD2-697E-BC86-08C1-CEE084D380E5" />
287 </imsld:role-part>
288 <imsld:role-part identifier="LD-A8A24A49-5BD7-D2C2-E207-07E06D0C34AF">
289 <imsld:title>Aula 11 - Aluno</imsld:title>
290 <imsld:role-ref ref="LD-5F01C274-26F6-191A-0A56-80D2911828C4" />
291 <imsld:learning-activity-ref ref="LD-74E37B10-FEEA-EEE2-1A5C-E237F76201D4" />
292 </imsld:role-part>
293 <imsld:role-part identifier="LD-2E5CE7AE-D6B6-01D3-C7B4-9A955FA5B6BD">
294 <imsld:title>Aula 11 - Instrutor</imsld:title>
295 <imsld:role-ref ref="LD-009A559B-61C8-240C-B00A-5E2CF81646D6" />
296 <imsld:learning-activity-ref ref="LD-74E37B10-FEEA-EEE2-1A5C-E237F76201D4" />
297 </imsld:role-part>
298 <imsld:role-part identifier="LD-DFA44409-3045-7961-7944-66B33964CACE">
299 <imsld:title>Aula 12 - Aluno</imsld:title>
300 <imsld:role-ref ref="LD-5F01C274-26F6-191A-0A56-80D2911828C4" />
301 <imsld:learning-activity-ref ref="LD-F7E0DFD9-380E-026D-8D16-E819A99E6921" />
302 </imsld:role-part>
303 <imsld:role-part identifier="LD-1778F83A-A0C2-96F1-6AAD-84903843AC2D">
304 <imsld:title>Aula 12 - Instrutor</imsld:title>
305 <imsld:role-ref ref="LD-009A559B-61C8-240C-B00A-5E2CF81646D6" />
306 <imsld:learning-activity-ref ref="LD-F7E0DFD9-380E-026D-8D16-E819A99E6921" />
307 </imsld:role-part>
308 </imsld:act>
309 </imsld:play>
310 </imsld:method>
311 <imsld:metadata />
312 </imsld:learning-design>
313 </organizations>
314 <resources>
315 <resource identifier="RES-89F46E1B-757F-908C-E161-1EC411C2E90A" type="webcontent" href=
316 "Slides/Aula%2001%20-%20Conceitos%20Iniciais.pdf">
317 <file href="Slides/Aula%2001%20-%20Conceitos%20Iniciais.pdf" />
318 </resource>
319 <resource identifier="RES-7B742F33-3618-63C3-B995-8DC7DEAC9EE9" type="webcontent" href=
320 "Slides/Aula%2005%20-%20Estrutura%20de%20Repetição%20(While).pdf">
321 <file href="Slides/Aula%2005%20-%20Estrutura%20de%20Repetição%20(While).pdf" />

```

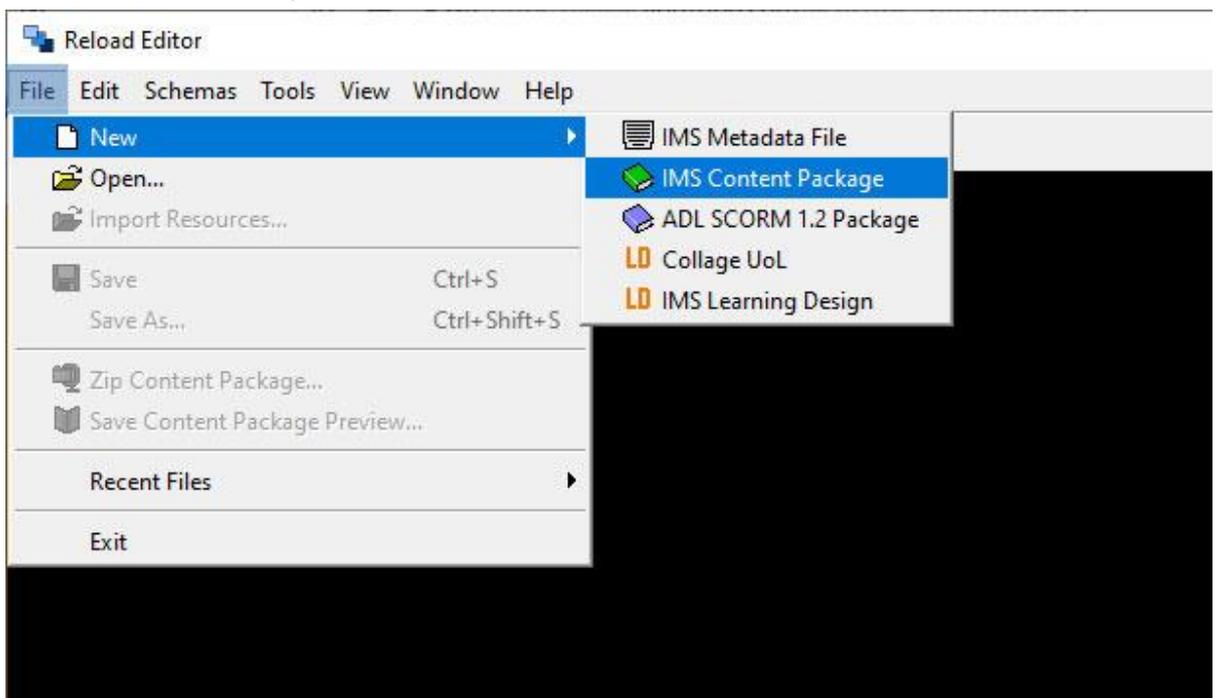
```

320 </resource>
321 <resource identifier="RES-2627A423-189C-B267-F152-EE383799B383" type="webcontent" href=
"Slides/Aula%2002%20-%20Familiarização%20com%20a%20IDE%20e%20o%20JKarel.pdf">
322 <file href="Slides/Aula%2002%20-%20Familiarização%20com%20a%20IDE%20e%20o%20JKarel.pdf" />
323 </resource>
324 <resource identifier="RES-ADE28659-9276-FD3E-F188-225F0C4B454C" type="webcontent" href=
"Slides/Aula%2003%20-%20Métodos%20(ViraDireita,%20SobeDegrau,%20DesceDegrau,%20SobeEscada).pdf">
325 <file href=
"Slides/Aula%2003%20-%20Métodos%20(ViraDireita,%20SobeDegrau,%20DesceDegrau,%20SobeEscada).pdf" />
326 </resource>
327 <resource identifier="RES-19AD44B1-A70D-95CB-5FE0-D04250E853E7" type="webcontent" href=
"Slides/Aula%2004%20-%20Estrutura%20Condicional%20(If-Else).pdf">
328 <file href="Slides/Aula%2004%20-%20Estrutura%20Condicional%20(If-Else).pdf" />
329 </resource>
330 <resource identifier="RES-53D25989-8D39-CD58-86FC-A4957E75595E" type="webcontent" href=
"Slides/Aula%2006%20-%20Variáveis.pdf">
331 <file href="Slides/Aula%2006%20-%20Variáveis.pdf" />
332 </resource>
333 <resource identifier="RES-336E84ED-2886-9986-5627-C86522803830" type="webcontent" href=
"Slides/Aula%2007%20-%20Avaliação%20de%20Revisão%20e%20Atividade%20Desplugada%20(Code%20Bot).pdf">
334 <file href=
"Slides/Aula%2007%20-%20Avaliação%20de%20Revisão%20e%20Atividade%20Desplugada%20(Code%20Bot).pdf"
/>
335 </resource>
336 <resource identifier="RES-DD7EB90E-9910-1E55-C8AE-29DE061591A1" type="webcontent" href=
"Slides/Aula%2008%20-%20Revisão%20de%20Conteúdo.pdf">
337 <file href="Slides/Aula%2008%20-%20Revisão%20de%20Conteúdo.pdf" />
338 </resource>
339 <resource identifier="RES-FDD6C472-38CC-B0A7-DB33-867DF2687E91" type="webcontent" href=
"Slides/Aula%2009%20-%20Projeto%20Ficção%20Interativa%20(Modelagem%20e%20Início%20do%20Roteiro).pdf"
>
340 <file href=
"Slides/Aula%2009%20-%20Projeto%20Ficção%20Interativa%20(Modelagem%20e%20Início%20do%20Roteiro).pd
f" />
341 </resource>
342 <resource identifier="RES-0CDBF08A-F531-C3EF-31FC-048439125006" type="webcontent" href=
"Slides/Aula%2010%20-%20Projeto%20Ficção%20Interativa%20(Término%20do%20Roteiro%20e%20Implementação)
.pdf">
343 <file href=
"Slides/Aula%2010%20-%20Projeto%20Ficção%20Interativa%20(Término%20do%20Roteiro%20e%20Implementaçã
o).pdf" />
344 </resource>
345 <resource identifier="RES-E95F4EBC-D6C7-E617-A961-E71133CDC748" type="webcontent" href=
"Slides/Aula%2011%20-%20Projeto%20Ficção%20Interativa%20(Implementação).pdf">
346 <file href="Slides/Aula%2011%20-%20Projeto%20Ficção%20Interativa%20(Implementação).pdf" />
347 </resource>
348 <resource identifier="RES-E450CCD6-E869-1C32-D67A-3F30FE074567" type="webcontent" href=
"Slides/Aula%2012%20-%20Projeto%20Ficção%20Interativa%20(Término%20e%20Apresentação).pdf">
349 <file href=
"Slides/Aula%2012%20-%20Projeto%20Ficção%20Interativa%20(Término%20e%20Apresentação).pdf" />
350 </resource>
351 </resources>
352 </manifest>

```

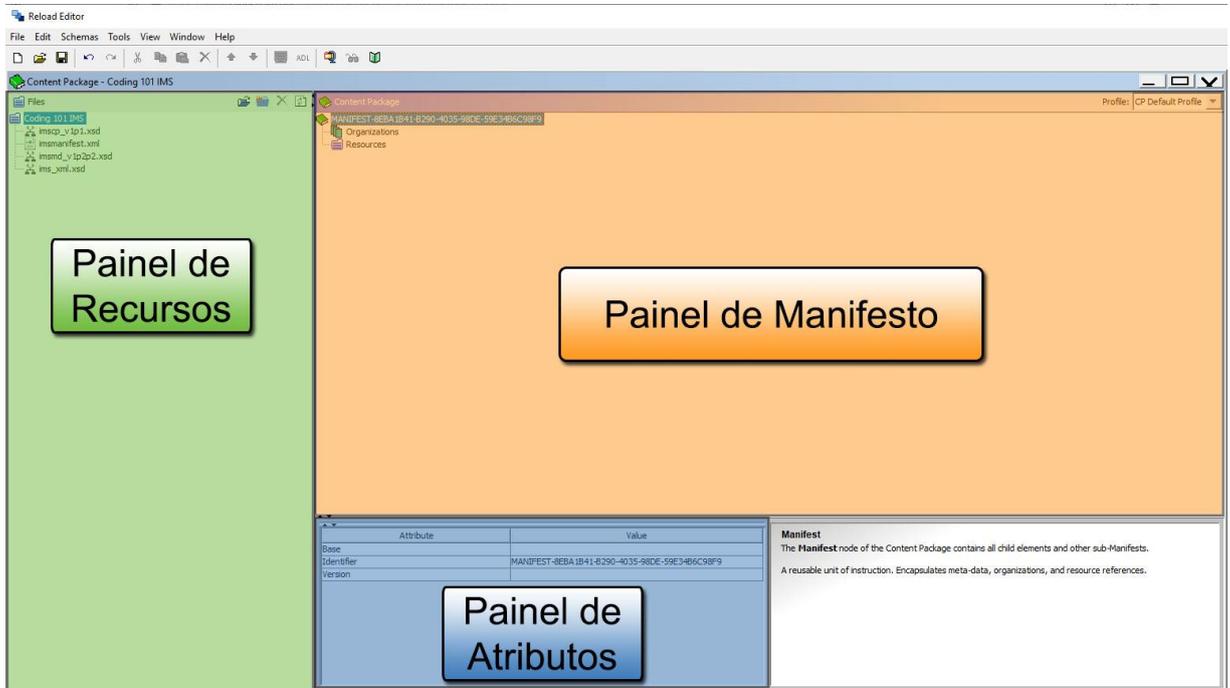
### 3. Modelagem de Pacote de Conteúdo em Padrão IMS

Vamos agora criar um Pacote de Conteúdo no formato *IMS*, selecionando "*File*", "*New*", "*IMS Content Package*". Pacotes de Conteúdo podem ser usados em conjunto com arquivos de *Learning Design IMS-LD*, formando Unidade de Ensino, e também podem ser executados de forma automatizada por sistemas de gerência de aprendizagem *LMS (Learning Management System)*, para uso em cursos online na modalidade *e-learning* para ensino online a distancia, caso estruturados utilizando arquivos *HTML* para cada atividade. Como este o curso deste estudo foi visualizado para o modelo presencial de aula, somente os slides da Aula 01 foram convertidos para arquivo *HTML* a título de exemplo caso o leitor tenha interesse em desenvolver cursos automatizados de *e-learning*.

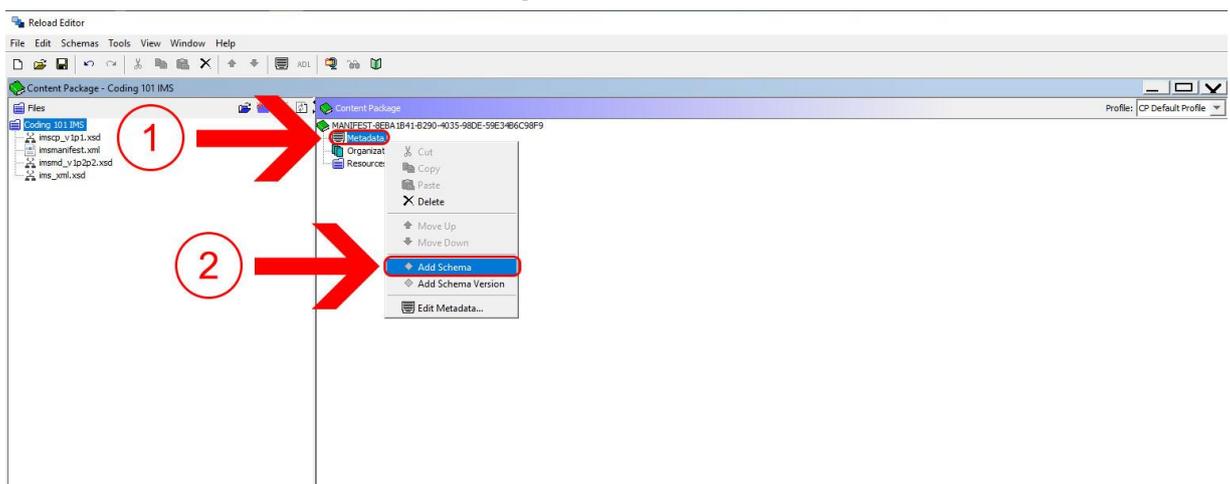


Após selecionada a pasta para registro dos arquivos, será exibida uma nova janela, mostrando o nome do pacote, definido pelo nome da pasta destino selecionada. Ao criar um pacote de conteúdo, o editor *Reload* imediatamente cria 4 arquivos na pasta destino com nomes padronizados, sendo o mais importante deles o "*imsmanifest.xml*", principal arquivo do pacote de conteúdo, que registra todas as informações do pacote e é referenciado pelos outros arquivos.

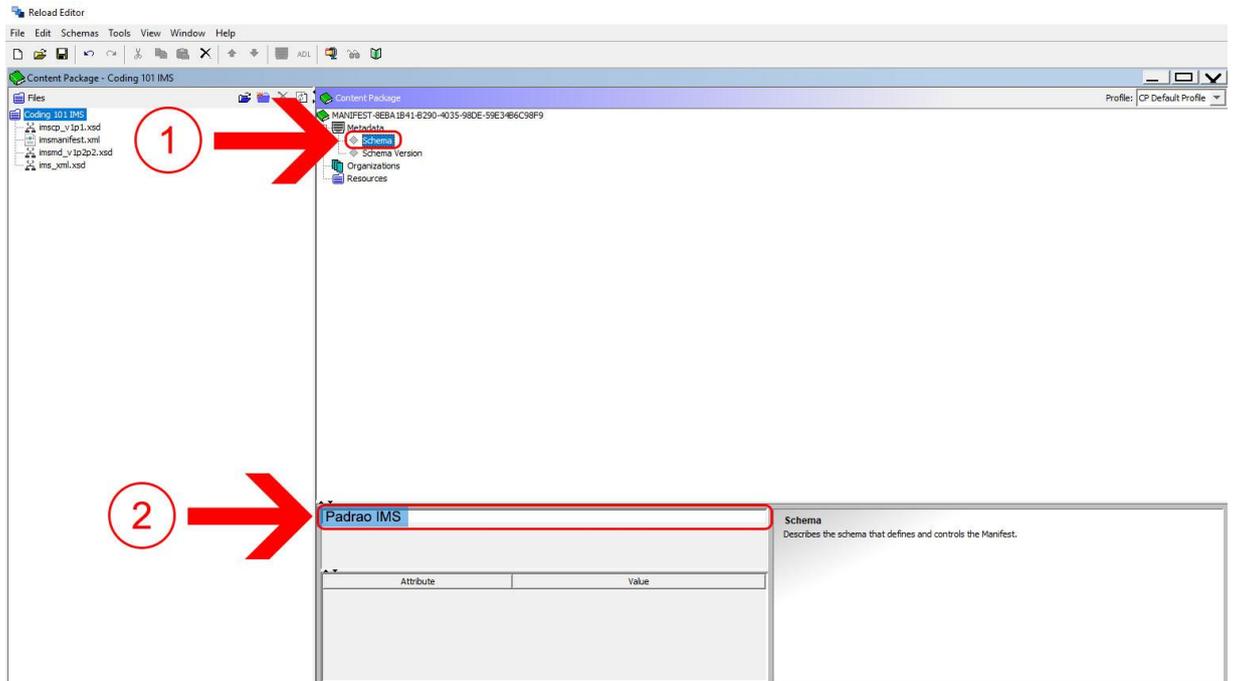
Na tela inicial do editor, as principais áreas utilizadas durante este processo serão os painéis de Manifesto, Recursos e Atributos, conforme destacados abaixo.



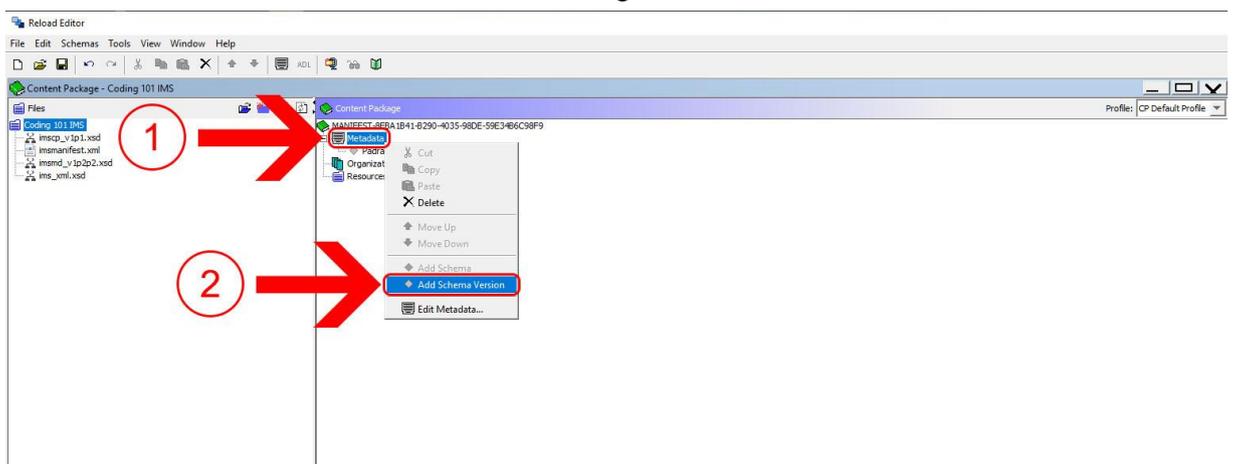
Criado o pacote de conteúdo no padrão *IMS*, a primeira ação tomada será adicionar uma referência de metadados, para identificar no cabeçalho do arquivo *XML* que nosso pacote está formalizado no padrão *IMS* junto a sua versão. Para isso, clique com o botão direito em "*Metadata*", e em seguida em "*Add Schema*".



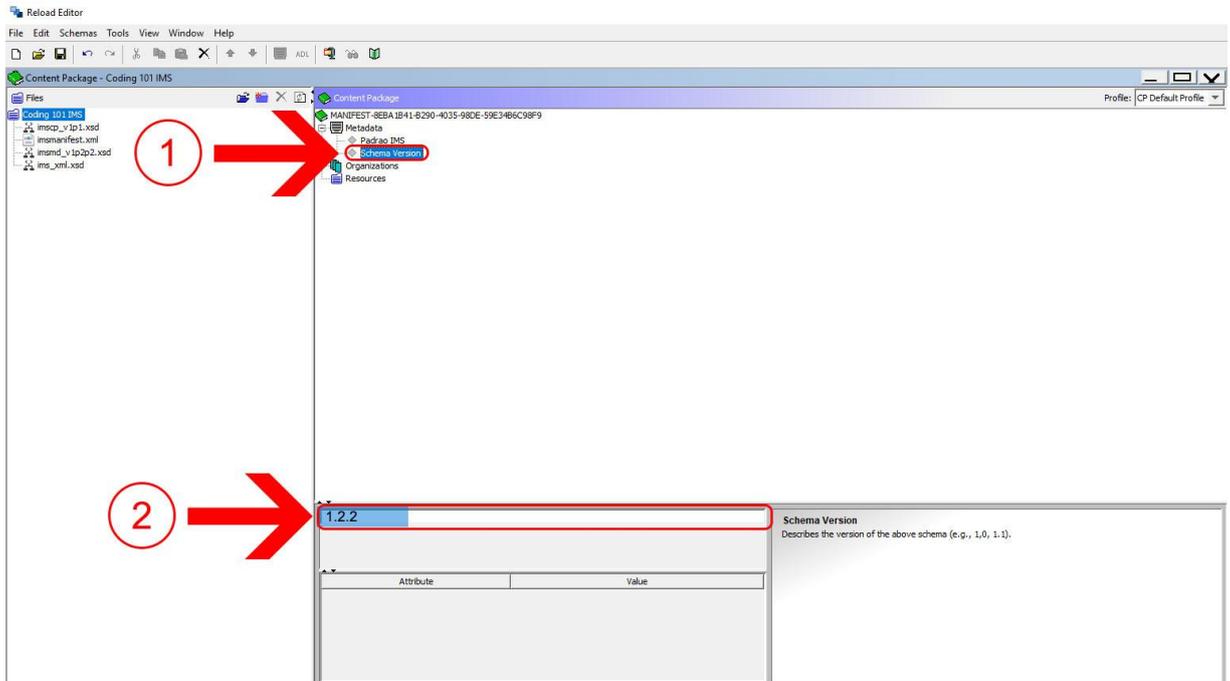
Após selecionar o "Schema" criado, podemos mudar sua identificação no Painel de Atributos, selecionando ele com o botão esquerdo do mouse e mudando seu valor para "Padrão IMS"



Vamos adicionar agora o metadado referente ao versionamento do *IMS LD*. Clique com o botão direito em "Metadata", e em seguida em "Add Schema Version".



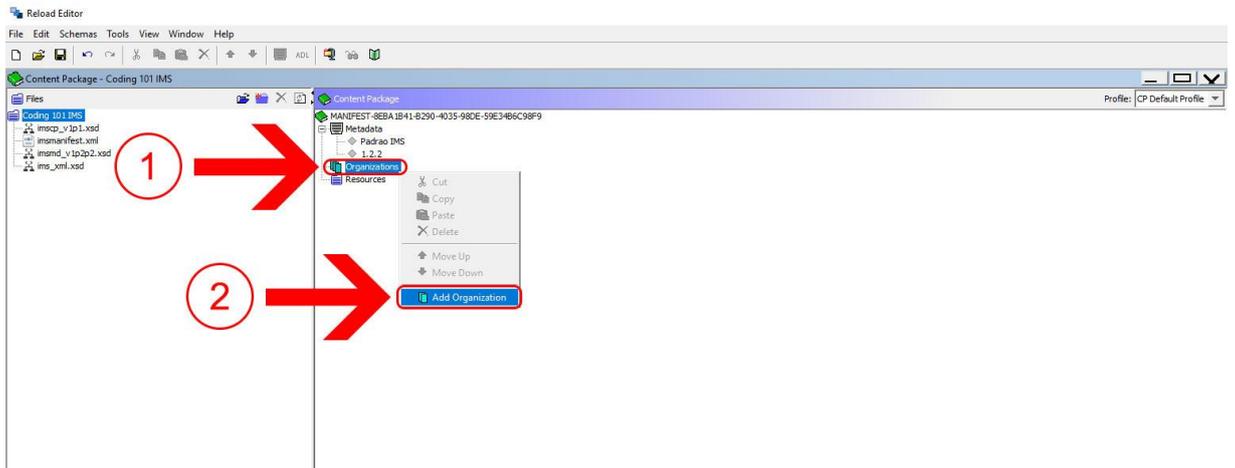
Assim como fizemos com o "Schema", vamos alterar o valor da versão de "Schema" criado, selecionando ele com o botão esquerdo do mouse e mudando seu valor para "1.2.2" no Painel de Atributos.



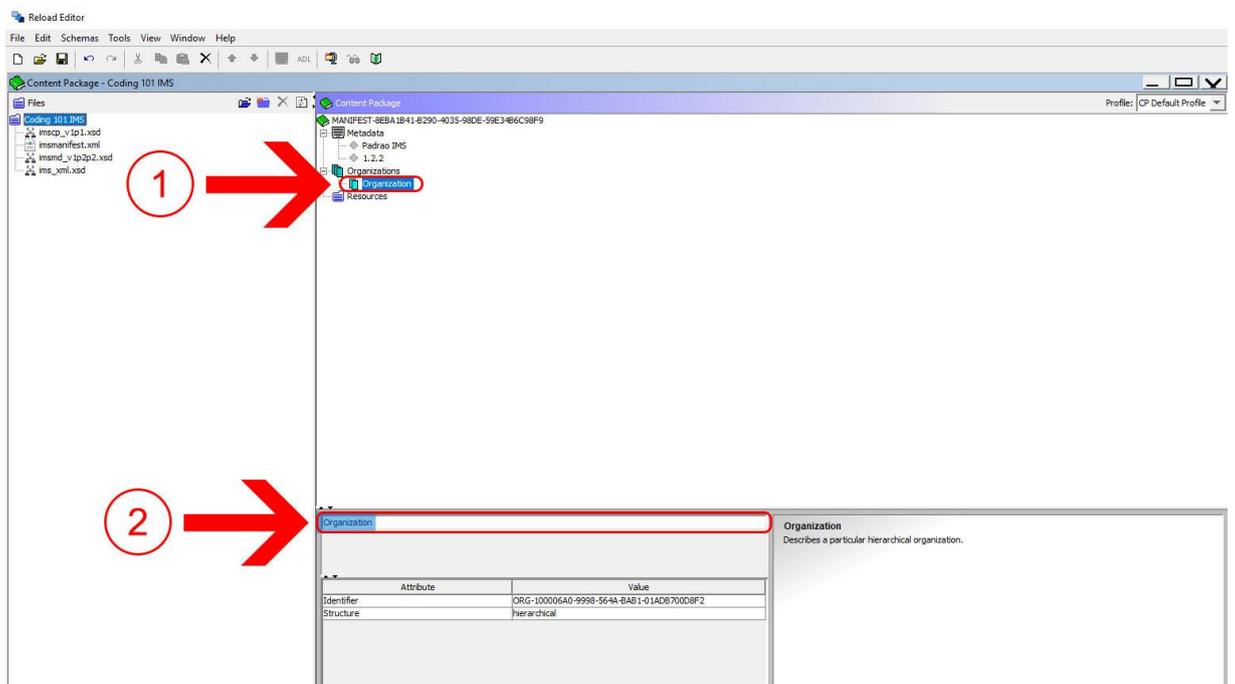
Para importar a primeira lição de nosso pacote de conteúdos, clique com o botão direito no nome do Pacote de Conteúdo no Painel de Recursos, e depois em "Import Resources". Neste exemplo, foi selecionada a pasta com os arquivos "index.html" referente à Aula 01, e seus respectivos arquivos de imagens.



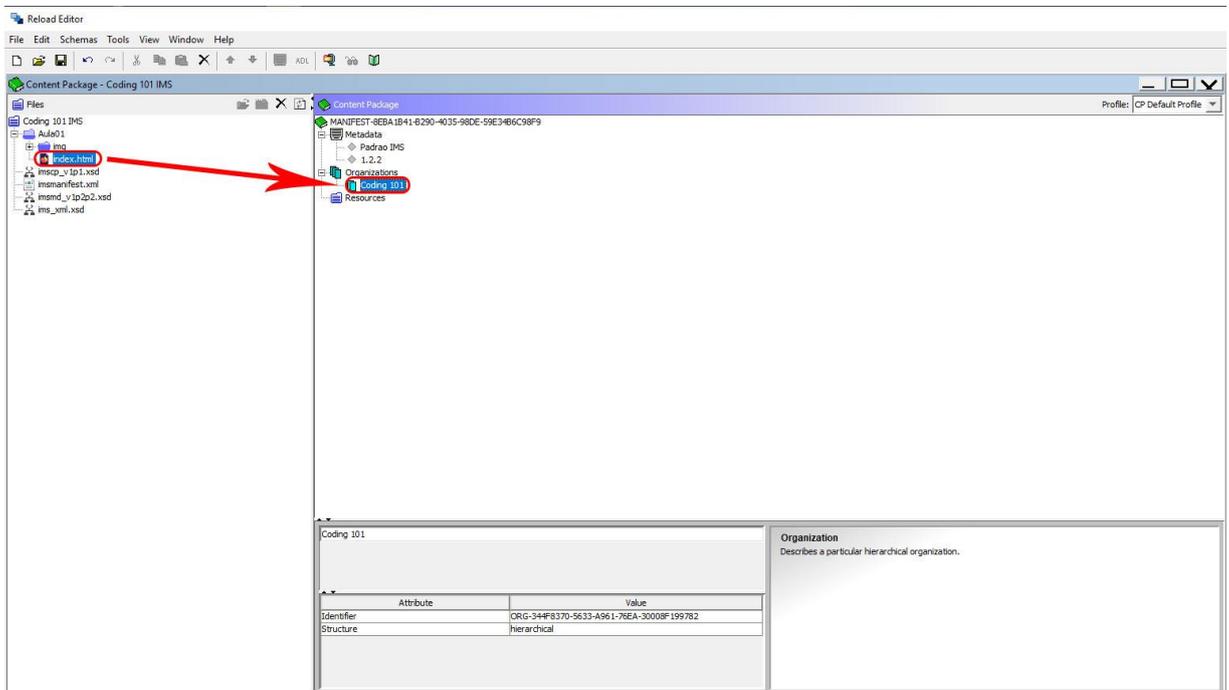
Precisamos agora criar uma nova organização, que neste contexto irá representar o curso que pretendemos formalizar no padrão *IMS LD*, onde serão alocadas as lições importadas. Para isso, basta clicar com o botão direito em "Organizations", e em seguida "Add Organization".



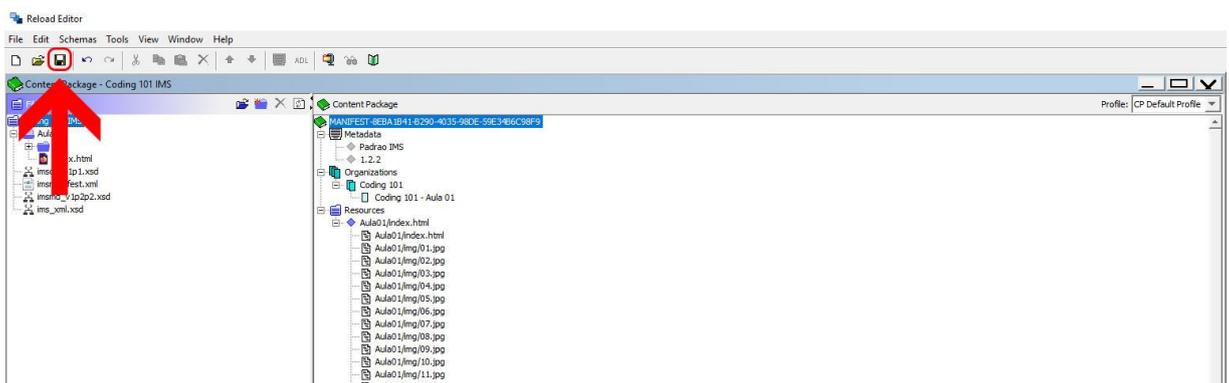
Renomeie a organização criada com o nome de nosso curso: "Coding 101". Para isso, basta selecionar a organização criada clicando uma vez com o botão esquerdo e, em seguida, no Painel de Atributos, modificar o nome da organização para o nome do curso desejado.



Vamos atribuir agora nosso arquivo "*index.html*", encontrado dentro da Aula01 importada anteriormente, para nossa recém criada organização. Para isso, basta arrastar o arquivo "*index.html*" para a organização criada ("Coding 101" em nosso caso).



Repare que automaticamente o campo de Recursos, logo abaixo das organizações, já foi atualizado com todos os arquivos que compõem o *HTML* da aula 01. Agora basta salvar o Pacote de Conteúdo *IMS* clicando no ícone indicado.



Agora basta abrir o arquivo do manifesto ("*imsmanifest.xml*"), presente na pasta onde se encontra o Pacote de Conteúdo IMS criado, para verificar o XML do mesmo.

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <!--This is a Reload version 2.0 Content Package document-->
3  <!--Spawned from the Reload Content Package Generator - http://www.reload.ac.uk-->
4  <manifest xmlns="http://www.imsglobal.org/xsd/imsdp_v1p1" xmlns:imsmd=
   "http://www.imsglobal.org/xsd/imsmd_v1p2" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   identifier="MANIFEST-8EBA1B41-B290-4035-98DE-59E34B6C98F9" xsi:schemaLocation=
   "http://www.imsglobal.org/xsd/imsdp_v1p1 imsdp_v1p1.xsd http://www.imsglobal.org/xsd/imsmd_v1p2
   imsmd_v1p2p2.xsd">
5  <metadata>
6  <schema>Padrao IMS</schema>
7  <schemaversion>1.2.2</schemaversion>
8  </metadata>
9  <organizations default="ORG-344F8370-5633-A961-76EA-30008F199782">
10 <organization identifier="ORG-344F8370-5633-A961-76EA-30008F199782" structure="hierarchical">
11 <title>Coding 101</title>
12 <item identifier="ITEM-F9013DE9-8928-6480-1BF8-5F4FE9BA1BEB" isvisible="true" identifierref=
   "RES-EF8651DC-8E99-8635-E3CF-D71160A86833">
13 <title>Coding 101 - Aula 01</title>
14 </item>
15 </organization>
16 </organizations>
17 <resources>
18 <resource identifier="RES-EF8651DC-8E99-8635-E3CF-D71160A86833" type="webcontent" href=
   "Aula01/index.html">
19 <file href="Aula01/index.html" />
20 <file href="Aula01/img/01.jpg" />
21 <file href="Aula01/img/02.jpg" />
22 <file href="Aula01/img/03.jpg" />
23 <file href="Aula01/img/04.jpg" />
24 <file href="Aula01/img/05.jpg" />
25 <file href="Aula01/img/06.jpg" />
26 <file href="Aula01/img/07.jpg" />
27 <file href="Aula01/img/08.jpg" />
28 <file href="Aula01/img/09.jpg" />
29 <file href="Aula01/img/10.jpg" />
30 <file href="Aula01/img/11.jpg" />
31 <file href="Aula01/img/12.jpg" />
32 <file href="Aula01/img/13.jpg" />
33 <file href="Aula01/img/14.jpg" />
34 <file href="Aula01/img/15.jpg" />
35 <file href="Aula01/img/16.jpg" />
36 <file href="Aula01/img/17.jpg" />
37 <file href="Aula01/img/18.jpg" />
38 <file href="Aula01/img/19.jpg" />
39 <file href="Aula01/img/20.jpg" />
40 <file href="Aula01/img/21.jpg" />
41 <file href="Aula01/img/22.jpg" />
42 <file href="Aula01/img/23.jpg" />
43 <file href="Aula01/img/24.jpg" />
44 <file href="Aula01/img/25.jpg" />
45 <file href="Aula01/img/26.jpg" />
46 <file href="Aula01/img/27.jpg" />
47 <file href="Aula01/img/28.jpg" />
48 </resource>
49 </resources>
50 </manifest>

```

Uma observação relevante a ser feita: Para criar Pacotes de Conteúdo no padrão de formalização **SCORM** (*Sharable Content Object Reference Model*), usado em plataformas como Moodle, segue o mesmo processo descrito acima, adicionando apenas 5 atributos opcionais para as atividades adicionadas na organização, sendo eles:

- \* **prerequisites** - Pré-requisitos necessários para o aprendiz participar da atividade sugerida
- \* **maxtimeallowed** - Tempo máximo permitido para realização da atividade
- \* **timelimitaction** - Ação a ser tomada caso o tempo da atividade chegue ao fim, definido em **maxtimeallowed**
- \* **datafromlms** - Data de inicialização a ser enviada ao sistema de gerencia de aprendizagem *LMS* (*Learning Management System*) na iniciação da atividade
- \* **masteryscore** - Valor referente à nota desejada como satisfatória ao término da atividade, entre 0 e 100

## 4. Referências Bibliográficas

IMS Global Learning Consortium. IMS Learning Design Best Practice and Implementation Guide. IMS Global Learning Consortium Inc, 2003. Link: [http://www.imsglobal.org/learningdesign/ldv1p0/imslid\\_bestv1p0.html](http://www.imsglobal.org/learningdesign/ldv1p0/imslid_bestv1p0.html) (Acessado em: 10/05/2021)

IMS Global Learning Consortium. IMS Learning Design Information Model. IMS Global Learning Consortium Inc, 2003. Link: [http://www.imsglobal.org/learningdesign/ldv1p0/imslid\\_infov1p0.html](http://www.imsglobal.org/learningdesign/ldv1p0/imslid_infov1p0.html) (Acessado em: 10/05/2021)