

UNIVERSIDADE DE SÃO PAULO  
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA  
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

**Facilitando o ensino da  
computação a distância através  
do uso de Livros Interativos**

Jiang Zhi  
Larissa Goto Sala

MONOGRAFIA FINAL

MAC 0499 — TRABALHO DE  
FORMATURA SUPERVISIONADO

Supervisor: Prof. Dr. Carlos Hitoshi Morimoto  
Supervisor: Prof. Dr. Jose Coelho de Pina Junior

São Paulo  
30 de Dezembro de 2020



# Agradecimentos

Aos nossos supervisores Carlos Hitoshi Morimoto e Jose Coelho de Pina Junior pela paciência e suporte durante este projeto.

Às nossas famílias e amigos que nos apoiaram durante toda a graduação e tornaram isso possível.



# Resumo

Jiang Zhi, Larissa Goto Sala. **Facilitando o ensino da computação a distância através do uso de Livros Interativos**. Monografia (Bacharelado). Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2020.

Este trabalho tem como objetivo utilizar ferramentas tecnológicas para democratizar e promover o ensino e o aprendizado a distância da Ciência da Computação e áreas relacionadas. Para isso, pretende-se acessibilizar as ferramentas da plataforma de livros interativos Runestone Interactive para uso em cursos na língua portuguesa. As funcionalidades interativas como o ambiente de desenvolvimento integrado, visualização da execução de programas passo a passo e vários tipos de exercícios permitem que estudantes pratiquem no mesmo local que possuem a teoria, de forma fácil e intuitiva. Esta nova forma de ensino pode ser interessante para o aprendizado de lógica de programação e estruturas de dados. A implementação teve como resultados a tradução parcial da plataforma Runestone para o português brasileiro, a criação de um plugin para integrar suas atividades como parte dos cursos do Moodle, por ser um dos ambientes virtuais de aprendizagem mais utilizados no Brasil, e o aprimoramento de uma plataforma do Instituto de Matemática e Estatística da Universidade de São Paulo (IME-USP) para disponibilizar conteúdos e recursos do Runestone em português para livre acesso.

**Palavras-chave:** Moodle. Runestone Interactive. Tecnologia Educacional. Educação.



# Abstract

Jiang Zhi, Larissa Goto Sala. **Facilitating distance computing teaching through Interactive Books**. Capstone Project Report (Bachelor). Institute of Mathematics and Statistics, University of São Paulo, São Paulo, 2020.

This work aims to use technological tools to better disseminate and promote distant teaching and learning of Computer Science and related subjects. It is intended to provide the tools of the interactive book platform Runestone Interactive for use in courses using the Portuguese Language. Interactive features such as an integrated development environment, a visual representation of step-by-step program execution, and various other types of exercises allow students to put to practice the theory they've learned, all using a single platform, in an easy and intuitive way. This new way of teaching allows for the better teaching of subjects related to programming logic and data structures. The implementation thus resulted in the partial translation of the Runestone platform into Brazilian Portuguese, the creation of a plugin to integrate Runestone into the Moodle courses, due to its prevalence in virtual learning environments throughout Brazil; and the improvement of a platform from the Institute of Mathematics and Statistics of the University of São Paulo (IME-USP) to provide free open access to Runestone's content and resources in Portuguese.

**Keywords:** Moodle. Runestone Interactive. Educational Technology. Education.



# Lista de Abreviaturas

API	Interface de Programação de Aplicação ( <i>Application Programming Interface</i> )
IdP	Provedor de Identidade ( <i>Identity Provider</i> )
IME	Instituto de Matemática e Estatística
LMS	Sistema de Gestão da Aprendizagem ( <i>Learning Management System</i> )
LTI	Interoperabilidade de ferramentas de aprendizagem ( <i>Learning Tools Interoperability</i> )
SAML	Linguagem de marcação de asserção de segurança ( <i>Security Assertion Markup Language</i> )
SP	Provedor de Serviço ( <i>Service Provider</i> )
USP	Universidade de São Paulo

## Lista de Figuras

1.1	Janela de exibição de um curso no Moodle. . . . .	2
1.2	Página de um livro interativo no Runestone. . . . .	4
1.3	Página de materiais disponíveis em Java na Panda. . . . .	6
3.1	Mensagens de erro de sintaxe em inglês. . . . .	9
3.2	Atividades disponíveis no Moodle 3.9. . . . .	10
3.3	Relacionamentos entre perfis e suas funções nos sistemas. . . . .	11
4.1	Exemplo de bloco <i>ActiveCode</i> . . . . .	13
4.2	Exemplo de bloco <i>CodeLens</i> . . . . .	14
4.3	Exemplo de exercício <i>Drag and Drop</i> . . . . .	14
4.4	Exemplo de exercício <i>Parsons</i> . . . . .	14
4.5	Tabela de tradução do componente <i>Parsons</i> . . . . .	15
4.6	Bloco <i>ActiveCode</i> com botões traduzidos. . . . .	16
4.7	Erro de execução traduzido. . . . .	16
4.8	Tradução pela interface administrativa do <i>web2py</i> . . . . .	17
4.9	Menu Runestone do Instrutor traduzido. . . . .	18
4.10	Página de <i>plugins</i> e suas configurações para administradores. . . . .	20
4.11	Parte da tela de configurações do <i>Plugin</i> . . . . .	21
4.12	Página de criação de uma instância Lição Runestone. . . . .	22
4.13	Visualização de uma Lição pelo Instrutor. . . . .	23
4.14	Esquematização das relações entre os dois bancos na sincronização de notas. . . . .	24
4.15	Visualização de notas Runestone no curso Moodle. . . . .	25
4.16	Vantagens do LTI. . . . .	26
4.17	Página de registro na Panda. . . . .	30
4.18	Página de autenticação na Panda. . . . .	30
A.1	Exportação de usuários na interface Moodle. . . . .	38

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Tecnologias educacionais . . . . .	1
1.2	Moodle . . . . .	2
1.3	Runestone interactive . . . . .	3
1.4	Objetivos . . . . .	5
<b>2</b>	<b>Organização e Metodologia</b>	<b>7</b>
<b>3</b>	<b>Planejamento</b>	<b>9</b>
3.1	Internacionalização do Runestone . . . . .	9
3.2	Funcionalidades do plugin . . . . .	10
3.3	Cadastro de usuários na Panda . . . . .	12
<b>4</b>	<b>Desenvolvimento</b>	<b>13</b>
4.1	Internacionalização . . . . .	13
4.2	Extensão lição Runestone para o Moodle . . . . .	18
4.2.1	Plugins base . . . . .	18
4.2.2	Configurações . . . . .	19
4.2.3	Criação, atualização e deleção . . . . .	21
4.2.4	Visualizações . . . . .	23
4.2.5	Sincronização de notas . . . . .	24
4.2.6	Autenticação . . . . .	26
4.2.7	Cadastro de estudantes no Runestone . . . . .	29
4.3	Adaptação da plataforma Panda . . . . .	29
<b>5</b>	<b>Considerações Finais</b>	<b>33</b>

## Apêndices

<b>A</b>	<b>Instruções de Uso</b>	<b>35</b>
A.1	Instalação e configuração do Runestone . . . . .	35
A.2	Tradução do Runestone . . . . .	36
A.3	Instalação do plugin do Moodle . . . . .	37
A.4	Cadastro de usuários do Moodle no Runestone . . . . .	38
A.5	Configuração da Panda . . . . .	39

	<b>Referências</b>	<b>41</b>
--	--------------------	-----------

# Capítulo 1

## Introdução

### 1.1 Tecnologias educacionais

Tecnologias educacionais<sup>1</sup> tratam da incorporação de recursos tecnológicos para facilitar qualquer processo de ensino e aprendizado.

Através da combinação de diversas ciências como pedagogia, psicologia, comunicação, sociologia, computação, entre muitas outras, as tecnologias educacionais integram técnicas, ferramentas e procedimentos na área da educação para promover um ambiente de aprendizado diverso que atenda as necessidades dos estudantes e instrutores.

Apesar de diversas tecnologias serem historicamente incorporadas como parte do ensino, como por exemplo, com o uso de rádios, gravadores de áudio e retroprojetores, o conceito de tecnologias educacionais popularizou-se juntamente ao crescimento da Tecnologia da Informação no final do século XX.

Com a criação dos computadores pessoais, a melhoria do seu processamento e a popularização da internet, tornou-se cada vez mais comum a adoção dos novos artifícios trazidos por essa revolução tecnológica, como aulas ministradas via videoconferências e compartilhamento de conhecimento através de fóruns e bibliotecas virtuais.

Essas novas tecnologias também permitem organizar cursos, ampliar o acesso a informações, aproximar a comunicação entre estudantes e instrutores e incentivar discussões. Isso ajuda a abrandar algumas das principais barreiras encontradas no ensino a distância, tais como explicitadas por GALUSHA (1998), a dificuldade na comunicação, pouco feedback dos instrutores, falta de suporte e tutoria acadêmica para os estudantes e dificuldade no planejamento dos estudos.

Uma categoria dessas ferramentas de apoio às tecnologias educacionais são os Sistemas de Gestão da Aprendizagem (Learning Management System, ou LMS<sup>2</sup>). São sistemas utilizados para a operacionalização e gerenciamento dos recursos de aprendizagem, através de ambientes virtuais que disponibilizam diversas funcionalidades para dar suporte ao

---

<sup>1</sup> [https://en.wikipedia.org/wiki/Educational\\_technology](https://en.wikipedia.org/wiki/Educational_technology)

<sup>2</sup> [https://en.wikipedia.org/wiki/Learning\\_management\\_system](https://en.wikipedia.org/wiki/Learning_management_system)

planejamento, execução e avaliação dentro dos processos de aprendizagem em programas educacionais.

Essas plataformas permitem a administração de cursos e seus conteúdos, incluindo tarefas e notas, registrando o progresso e o desempenho dos estudantes. Seus principais aspectos são a facilitação ao acesso de informações e a interatividade, permitindo construir uma aprendizagem colaborativa entre estudantes e instrutores.

Atualmente existe um grande número de sistemas de gestão. Um dos mais conhecidos e utilizados mundialmente é o LMS chamado **Moodle**.

## 1.2 Moodle

O Moodle (acrônimo para *Modular Object-Oriented Dynamic Learning Environment*, ou Ambiente Modular de Aprendizagem Dinâmica Orientada a Objetos) é uma plataforma de aprendizado criada para facilitar o uso de ambientes e projetos virtuais no ensino a distância ou misto.

Ele é um projeto de *software* livre, ou seja, seu código fonte é aberto para utilização, contribuição e distribuições pela comunidade. Sua primeira versão foi lançada há mais de 18 anos e a distribuição está sob a Licença Pública Geral GNU, ou GNU GPL<sup>3</sup>.

Dentre as principais ferramentas que a plataforma oferece está a criação de cursos por instrutores, onde é possível disponibilizar diversos tipos de recursos em que os estudantes podem interagir e colaborar entre si e com instrutores, como questionários, atividades, chats, fóruns de discussões, feedbacks, materiais de apoio, *workshops* e vários outros (Figura 1.1). Também existem funcionalidades para assistência à organização pessoal como calendários e notificações de datas importantes, além do acompanhamento do progresso no curso e das notas das atividades.

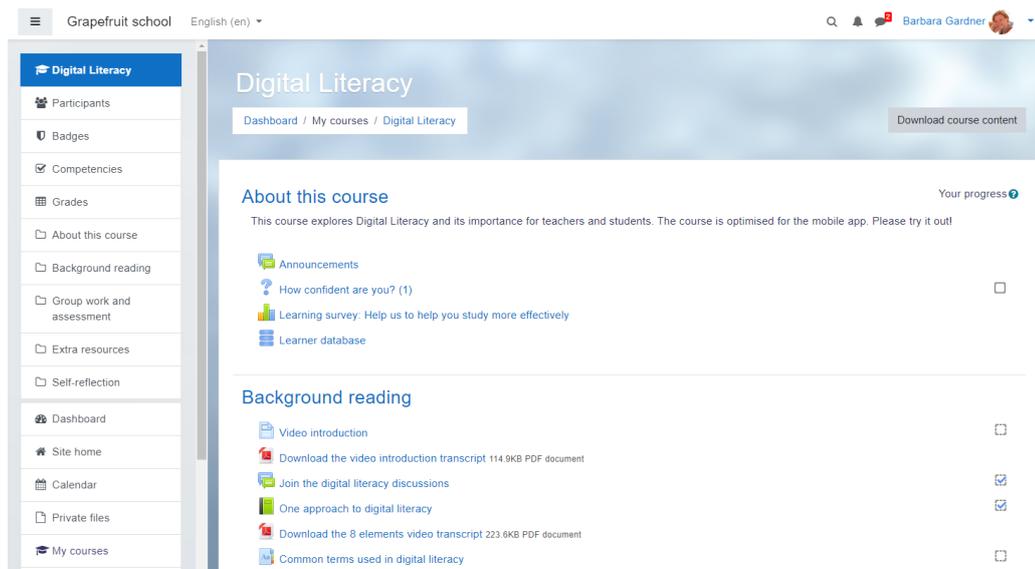


Figura 1.1: Janela de exibição de um curso no Moodle.

<sup>3</sup> <https://www.gnu.org/licenses/gpl-3.0.html>

Outra característica relevante da plataforma é a possibilidade de personalização para diferentes necessidades. Os administradores podem customizar as formas de exibir do curso e seus materiais. Também é possível criar novos temas e *plugins*, permitindo a adaptação ou criação de *layouts* e funcionalidades. Em outubro de 2019, haviam 1,619 *plugins* disponíveis para *download* na plataforma<sup>4</sup>.

Sua popularidade concentra-se na Europa, América Latina e Oceania, onde o Moodle representa pelo menos 50% do mercado de Sistemas de Gestão da Aprendizagem (HILL, 2017). O Brasil está entre os 10 países que mais utilizam o Moodle ao redor do mundo, com quase 8 mil sites registrados<sup>5</sup>.

Na Universidade de São Paulo, instâncias do Moodle foram utilizadas pelos institutos, como por exemplo os sistemas PACA<sup>6</sup> no Instituto de Matemática e Estatística (IME-USP) desde 2005, e GRAUNA<sup>7</sup> que até 2019 abrangeu cursos do IME na Escola Politécnica (POLI-USP). Atualmente, o Moodle tem sido empregado de maneira crescente na Universidade de São Paulo através da plataforma e-Disciplinas<sup>8</sup>.

### 1.3 Runestone interactive

Runestone Interactive<sup>9</sup> é um projeto de código aberto que fornece ferramentas e hospedagem gratuita para livros interativos de conteúdos de livre acesso. Sua missão é democratizar os livros didáticos do século 21 e engajar os estudantes com novos métodos de ensino.

Apesar de oferecer instrumentos que podem ser utilizados por diversas áreas, a plataforma oferece funcionalidades específicas para o ensino da Computação. Foi idealizada por Brad Miller e David Ranum e apresentada na 17ª Conferência Anual de Inovação e Tecnologia em Educação na área da Ciência da Computação no ano de 2012 (MILLER e RANUM, 2012).

Um problema que o Runestone se propõe a mitigar é a presença de diversas barreiras no ensino e aprendizagem de conceitos iniciais da computação. De acordo com Qian e Lehman (QIAN e LEHMAN, 2017), entre as principais dificuldades e equívocos cometidos pelos estudantes iniciantes da área estão os problemas de interpretação de sintaxe e semântica, a lógica de programação e as estratégias de resolução e abstração de problemas computacionais. Pensando nisso, os livros interativos buscam trazer ferramentas que clarifiquem e reforcem o conteúdo para evitar a ocorrência dessas dificuldades.

No Runestone, além de disponibilizar o material para leitura, é possível incluir vídeos, exercícios e trechos de código executáveis dentro do próprio capítulo. Os vários tipos de exercícios, como por exemplo os de múltipla escolha, complete a frase (*fill in the blank*) e

---

<sup>4</sup> <https://en.wikipedia.org/wiki/Moodle>

<sup>5</sup> <https://stats.moodle.org/>

<sup>6</sup> <https://paca.ime.usp.br/>

<sup>7</sup> <https://grauna.ime.usp.br/>

<sup>8</sup> <https://edisciplinas.usp.br/>

<sup>9</sup> <https://runestone.academy/>

de "clique e arraste" (*drag and drop*), estão inseridos nos capítulos para reforçar o conteúdo de forma periódica e consistente, contendo também um sistema de retorno de *feedbacks* automático das respostas para que os estudantes compreendam seus erros e acertos.

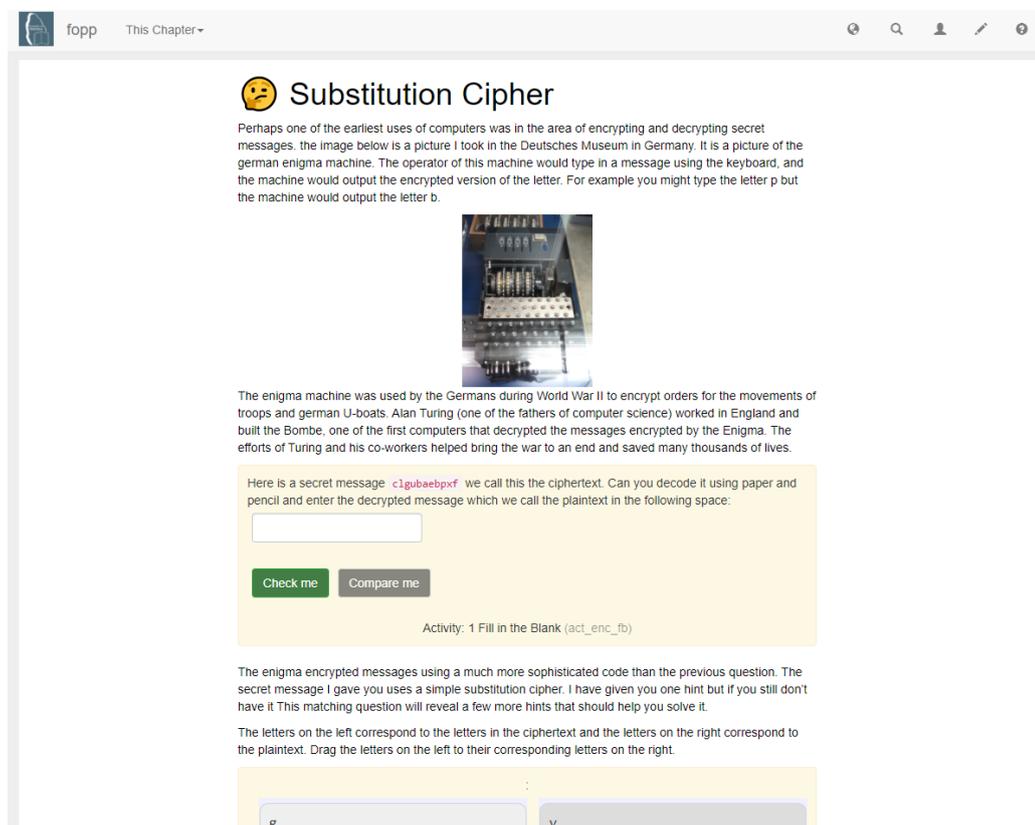


Figura 1.2: Página de um livro interativo no Runestone.

Outro dos principais recursos disponíveis é um ambiente de desenvolvimento embutido na página para que o estudante possa resolver pequenos exercícios de programação e também acompanhe a execução de seu programa, linha a linha. Isso evita a complicação de configurar ambientes de desenvolvimento, que é também um atrito para novos estudantes, e permite melhor entendimento da lógica de programação através da visualização do funcionamento do programa passo a passo. O sistema de retorno de erros de sintaxe também clarifica para o estudante o motivo do erro estar acontecendo e como corrigi-lo.

Todos esses componentes interativos dos livros também podem ser agrupados em tarefas e avaliações que os instrutores disponibilizam como parte de um curso na própria plataforma Runestone. As atividades são avaliadas automaticamente, o que facilita muito o processo para os instrutores, mas também podem ter suas notas modificadas manualmente se desejado.

A plataforma também registra dados do progresso dos estudantes na leitura do conteúdo e das interações com as funcionalidades, fornecendo ferramentas de análises e acompanhamento para dar melhor suporte ao aprendizado.

Em um experimento, a faculdade de Luther College utilizou em 2011 um dos livros interativos oferecidos pela plataforma no curso de Introdução à Ciência da Computação. Ao final da disciplina, haviam sido registrados mais de 240 mil eventos de interações com o material, 88% dos estudantes considerou que o caráter interativo do material auxiliou nos estudos. A leitura do livro e a execução de seus programas foram elencados como principais fatores que contribuíram para o aprendizado logo após o acompanhamento das aulas (MILLER e RANUM, 2012).

Atualmente seu servidor oficial com diversos materiais disponíveis em inglês é utilizado por mais de 50 mil estudantes distribuídos em mais de 1000 universidades, faculdades e colégios<sup>10</sup>.

O projeto fornece ainda um sistema provedor de serviços, ou servidor, de código aberto, que permite que os próprios indivíduos ou instituições hospedem e criem seus livros interativos. Esse servidor permite que, por exemplo, tarefas de leitura sejam oferecidas a estudantes como preparação para aulas.

## 1.4 Objetivos

A plataforma Runestone apresenta uma nova metodologia de ensino da computação através de suas funcionalidades interativas, no entanto ainda é pouco conhecida e utilizada, principalmente no Brasil. O objetivo desse trabalho é acessibilizar não apenas a ferramenta mas também democratizar seus materiais para os estudantes brasileiros, facilitando o aprendizado de uma área complexa e de extrema importância como a computação.

O primeiro passo foi traduzir as ferramentas e o servidor do Runestone para português, expandindo assim o público que consiga acessar, navegar e aproveitar o seu conteúdo.

Para incentivar o uso das atividades interativas como meio de ensino da Ciência da Computação, integrar o Runestone com o Moodle foi considerado um passo importante, tendo em vista que o Moodle é uma plataforma muito utilizada em várias instituições de ensino brasileiras, incluindo a Universidade de São Paulo.

Para isso, foi necessário o desenvolvimento de uma extensão (*plugin*) do Moodle que permite a criação e manipulação das lições interativas e suas notas dentro do curso, com uma interface unificada que não exige a constante migração de uma plataforma para a outra para os instrutores e nem os estudantes.

Por fim, para atingir maior democratização de conteúdos em português, a última etapa envolveu o aprimoramento da Panda<sup>11</sup> (Figura 1.3), uma plataforma do IME-USP que disponibiliza materiais desenvolvidos e/ou utilizados no Departamento de Ciência da Computação do instituto para qualquer pessoa que esteja interessada em seu uso.

---

<sup>10</sup> <https://runestoneinteractive.org/pages/faq.html>

<sup>11</sup> <https://panda.ime.usp.br/>

Departamento de Ciência da Computação - IME - USP

## Panda - Cursos de Computação

Principal C Java Python


**INSTITUTO DE MATEMÁTICA E ESTATÍSTICA**  
 UNIVERSIDADE DE SÃO PAULO

### Material para cursos baseados em Java

Nessa página você vai encontrar informações e material para aprender Computação usando Java.

Cursos introdutórios:

Esse material é voltado para cursos introdutórios, onde nenhum pré-requisito de computação é necessário.

- [Introdução à Ciência da Computação com Java e Orientação a Objetos](#), dos professores Alfredo Goldman, Fabio Kon e Paulo J. S. Silva, editado por Raphael Y. de Camargo.

Como instalar Java JDK em seu computador

- [Como posso começar a desenvolver programas Java com o Java Development Kit \(JDK\)?](#)
- [Download do JDKa](#)

Documentação do Java

- [Java Platform, Standard Edition \(Java SE\)](#).

**Material:**

- [Para cursos em C](#)
- [Para cursos em Java](#)
- [Para cursos em Python](#)

Copyright (c) 2015 DCC/IME/USP. All rights reserved. powered by [web2py](#)

**Figura 1.3:** *Página de materiais disponíveis em Java na Panda.*

Utilizando o Runestone como ferramenta intermediária, é possível disponibilizar livros interativos e todas as suas funcionalidades para livre acesso.

Ao final do projeto pretende-se ter implementações funcionais que poderão ser utilizadas como testes nos servidores da Universidade de São Paulo.

## Capítulo 2

# Organização e Metodologia

Como ferramenta para a organização das tarefas, foi utilizado apenas um quadro *Kanban* na plataforma Trello<sup>1</sup>. O Kanban é um método de gestão que auxilia no desenvolvimento de projetos através da visualização de fluxos de trabalhos utilizando cartões, muito utilizado em desenvolvimento de *softwares* (ARRUDA, 2012). A categorização entre tarefas que precisam ser executadas, tarefas em progresso e tarefas concluídas auxilia na compreensão da capacidade de entregas da equipe, no acompanhamento da evolução do desenvolvimento e no planejamento das etapas seguintes.

Durante o período foram realizadas reuniões semanais com os supervisores para discutir sobre o progresso e próximos passos. A documentação de detalhes importantes desses encontros foi feita em uma planilha interna.

A etapa inicial para o desenvolvimento do projeto foi o estudo das ferramentas envolvidas. Como o trabalho foi planejado para ser executado em dupla, o estudo da aplicação Runestone foi feito em paralelo com o estudo da plataforma Moodle. Essa fase também incluiu a instalação e aprofundamento das implementações a nível de código de ambos os sistemas.

O estudo do Runestone foi dividido em duas partes: seus componentes (*Runestone Components*<sup>2</sup>) e o servidor (*Runestone Server*<sup>3</sup>). Os componentes são a implementação das ferramentas interativas e o suporte para a interpretação e compilação da estrutura dos livros através de instruções personalizadas chamadas de diretivas<sup>4</sup>. Já o servidor suporta a hospedagem de diversos livros, além de disponibilizar outras funcionalidades para gerenciamento de instrutores, estudantes e cursos, assim como a criação de avaliações, correção automática e coleta de dados sobre o progresso dos estudantes ao longo do curso. A implementação da plataforma utiliza majoritariamente as linguagens de programação Python e JavaScript.

O estudo da plataforma Moodle foi focado na estrutura do código interno e na criação

---

<sup>1</sup> <https://trello.com/>

<sup>2</sup> <https://github.com/RunestoneInteractive/RunestoneComponents>

<sup>3</sup> <https://github.com/RunestoneInteractive/RunestoneServer>

<sup>4</sup> <https://runestone.academy/runestone/static/authorguide/directives.html>

de novos *plugins*, ou extensões. Seu sítio disponibiliza uma ampla documentação<sup>5</sup>, o que auxiliou na instalação e no desenvolvimento, indicando qual a estrutura dos arquivos e funções necessárias para o tipo de extensão que desejávamos criar. Também foi necessário dedicar um tempo ao estudo da linguagem de programação PHP para a compreensão e desenvolvimento do código no Moodle.

Em seguida, foram determinadas as funcionalidades desejadas da extensão do Moodle e as alterações que seriam necessárias na plataforma Runestone para tornar seu conteúdo acessível aos estudantes brasileiros. As definições serão mais detalhadas no capítulo 3: [Planejamento](#).

O desenvolvimento se iniciou sendo guiado por essas definições. A implementação foi feita primeiramente em instalações locais dos sistemas, e posteriormente testadas em um servidor externo da Amazon Web Services (AWS). Os estudos foram contínuos durante essa fase à medida que os obstáculos surgiam. Os detalhes dessa etapa estão no capítulo 4: [Desenvolvimento](#).

A última fase do projeto consistiu em aperfeiçoamento de alguns detalhes do código e realização de testes no conjunto das plataformas. As conclusões dos objetivos alcançados e sugestões de próximos passos para trabalhos futuros encontram-se no capítulo 5: [Considerações Finais](#).

---

<sup>5</sup> <https://docs.moodle.org/>

# Capítulo 3

## Planejamento

A partir do estudo de cada plataforma e suas ferramentas foram elencadas as principais modificações necessárias para facilitar, de alguma forma, o acesso aos livros interativos.

O planejamento do projeto baseou-se nas alterações do sistema Runestone e nas funcionalidades da extensão do Moodle. As especificações seguem detalhadas nas próximas seções.

### 3.1 Internacionalização do Runestone

A plataforma Runestone é implementada em inglês, não admitindo o idioma brasileiro. Por isso, um dos objetivos foi acessibilizá-la para os estudantes com o português como língua nativa.

O conteúdo dos livros pode ser escrito em português, no entanto as ferramentas interativas disponíveis trazem botões e o retorno dos exercícios em inglês, tais como mensagens de erro e dicas para o estudante (Figura 3.1).

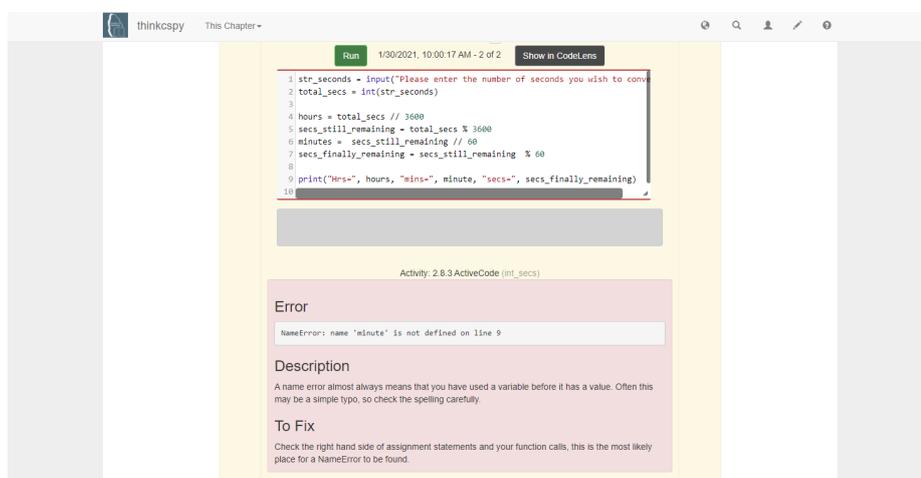


Figura 3.1: Mensagens de erro de sintaxe em inglês.

Também é necessário alterar os textos do servidor, como menus e demais páginas da plataforma, para disponibilizar uma melhor experiência de usuário e guiá-lo na navegação.

## 3.2 Funcionalidades do plugin

O principal objetivo desta extensão é permitir que as atividades criadas nos cursos do Runestone possam fazer parte do conjunto de atividades disponíveis nos cursos do Moodle (Figura 3.2)

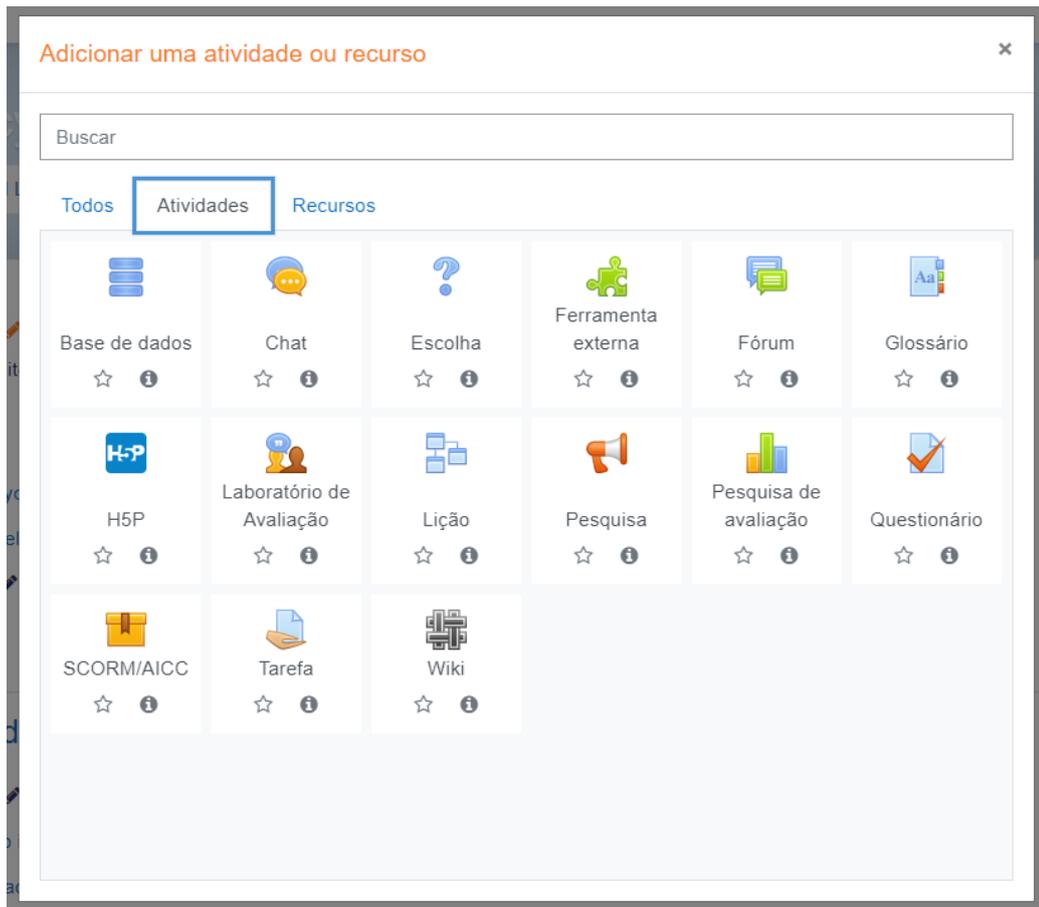


Figura 3.2: Atividades disponíveis no Moodle 3.9.

Primeiramente, foi necessário compreender a arquitetura de ambas plataformas. Tanto o Moodle quanto o Runestone possuem cursos onde há a interação entre dois papéis: os instrutores e os estudantes. Como um curso no Runestone está associado a um único livro, para que uma possível integração funcione, os estudantes do Moodle precisam estar cadastrados em todos os cursos Runestone que possuem as atividades do livro que desejam acessar, e os instrutores precisam ter o cargo de instrutores desses cursos para gerenciar as lições.

Mas para planejar as funcionalidades desejadas, é preciso saber não somente os requisitos dos sistemas mas também as necessidades de cada perfil de usuário, compreendendo

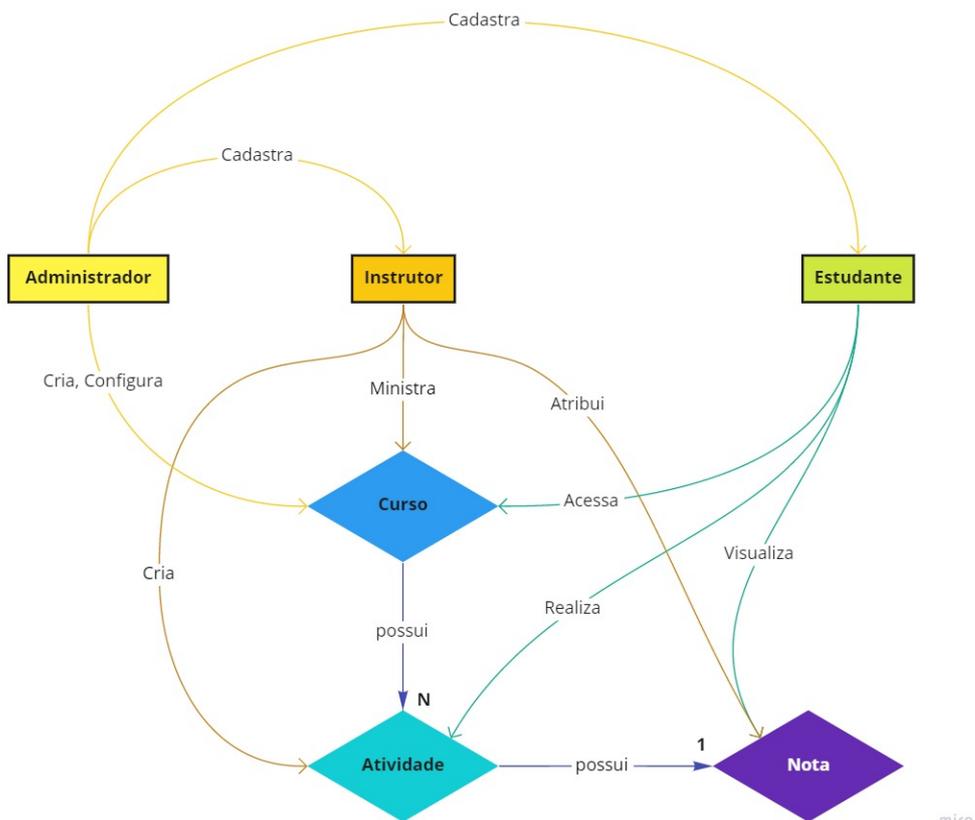
como eles interagem com o Runestone, o Moodle e suas ferramentas. Para isso, foram analisados os perfis de administradores, instrutores e estudantes (Figura 3.3).

Os administradores são responsáveis pelas configurações da plataforma, dos cursos e pelo cadastro dos usuários de outros perfis de acesso em ambos os sistemas.

Os instrutores gerenciam o conteúdo dos cursos. No Runestone, esse conteúdo se resume ao livro interativo e as atividades, as quais podem consistir de leituras e exercícios dos capítulos do livro associado àquele curso, ou até mesmo de novas questões personalizadas. Além disso, esse perfil também é responsável por avaliar as atividades, administrar as notas dos estudantes e acompanhar seu progresso.

Os usuários com perfil de acesso de monitoria no Moodle estão inclusos nas mesmas funcionalidades e permissões de um instrutor.

Já os estudantes acessam o material e realizam as atividades propostas, além de poderem visualizar suas notas no sistema.



**Figura 3.3:** Relacionamentos entre perfis e suas funções nos sistemas.

O propósito é integrar as funcionalidades de ambos os sistemas, considerando suas particularidades, de maneira que o Moodle possa ser utilizado como a principal plataforma e o Runestone seja apenas uma ferramenta auxiliar, "transparente" para os usuários.

Assim, para cada perfil foram definidas como funcionalidades a serem desenvolvidas nesta extensão:

#### 1. Administrador

- Configuração e habilitação do Plugin
- Cadastro dos estudantes nos respectivos cursos Runestone

#### 2. Instrutor

- Criação de uma Lição Runestone dentro de um determinado curso
- Edição das Lições
- Avaliação das Lições
- Acompanhamento de estatísticas do progresso dos estudantes com relação ao livro interativo
- Visualização e edição das notas da Lição Runestone dentro da relação de notas do curso Moodle

#### 3. Estudante

- Visualização e realização da Lição Runestone
- Consulta de suas notas das Lições Runestone dentro da relação de suas demais notas do curso Moodle

### 3.3 Cadastro de usuários na Panda

Atualmente a plataforma Panda disponibiliza alguns livros interativos traduzidos em português, redirecionando o visitante para uma instância do livro aberto onde é possível realizar a leitura do material e o uso dos trechos interativos.

No entanto, não é possível utilizar todos os recursos que um curso do servidor Runestone oferece pois não há a possibilidade de cadastro de usuários. Como principal desvantagem, os visitantes não conseguem salvar seu progresso no livro e perdem anotações, códigos e respostas de exercícios que foram feitos.

Assim, além de manter o acesso livre a esse conteúdo, um servidor Runestone deverá ser adicionado para gerenciar o funcionamento por trás do Panda e seus livros para os usuários que optarem salvar seu progresso. Será necessária a adaptação da interface da plataforma Panda para que o cadastro dos usuários seja feito através dela. A partir disso, o registro do usuário em todos os cursos disponíveis será automático, e o usuário passa a ter acesso ao conjunto completo de funcionalidades do servidor Runestone.

# Capítulo 4

## Desenvolvimento

Nesse capítulo é detalhado o processo de desenvolvimento do projeto de acordo com o planejamento inicial, além de obstáculos encontrados e as decisões tomadas durante o processo.

### 4.1 Internacionalização

O trabalho da tradução das *strings* do Runestone foi dividido em duas partes: primeiramente, a tradução dos textos e mensagens relacionadas aos seus componentes, e em seguida a tradução dos principais textos do servidor.

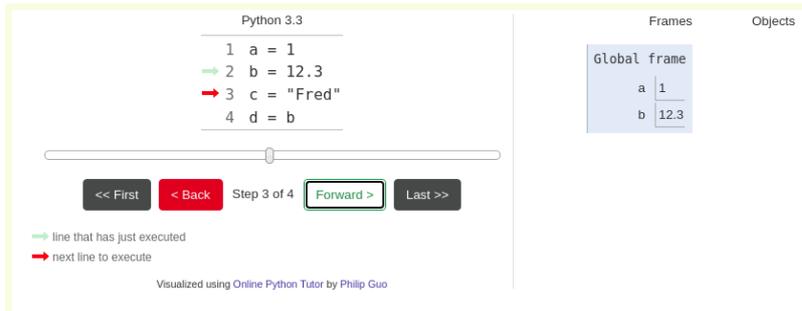
Como citado anteriormente, os componentes do Runestone consistem nas ferramentas e tipos de exercícios interativos que a plataforma oferece como funcionalidades dentro dos livros.

Considerando os principais componentes relacionados ao aprendizado de execução de código, pode-se citar o *ActiveCode* (Figura 4.1), uma caixa de texto embutida na página onde é possível escrever e executar programas em tempo real, e o *CodeLens* (Figura 4.2), que pode ser utilizado junto com o *ActiveCode* e permite acompanhar o estado do programa e de suas variáveis linha a linha do código para proporcionar maior compreensão sobre a lógica e as operações feitas durante a execução.



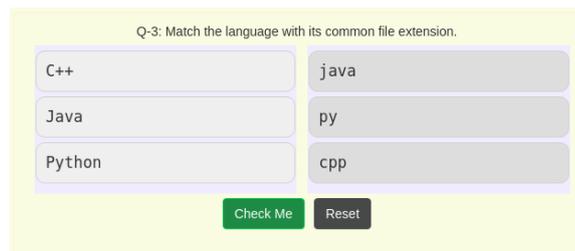
```
1 import turtle
2 t = turtle.Turtle()
3
4 for i in range(4):
5     t.forward(100)
6     t.left(90)
```

**Figura 4.1:** Exemplo de bloco *ActiveCode*.

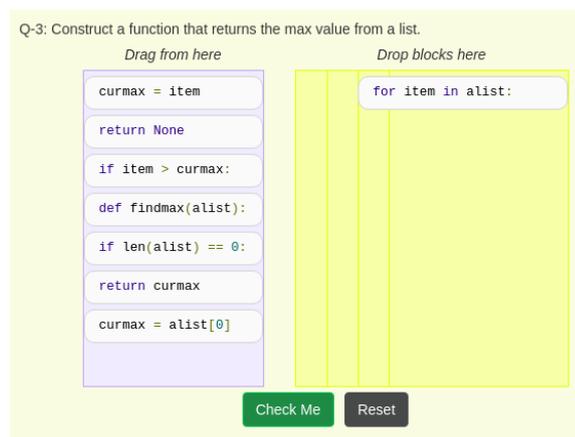


**Figura 4.2:** Exemplo de bloco CodeLens.

Já as principais ferramentas disponíveis para realização de exercícios são o *Multiple Choice* (questões de múltipla escolha), o *Drag and Drop*, que são questões de "clique e arraste" de pequenas caixas para relacionar conceitos entre múltiplas opções (Figura 4.3), o *Fill in the Blank* para que os estudantes completem frases e o *Parsons*, que consiste no exercício de lógica baseado na ordenação correta de linhas de código (Figura 4.4).



**Figura 4.3:** Exemplo de exercício Drag and Drop.



**Figura 4.4:** Exemplo de exercício Parsons.

Dentre os componentes citados acima, apenas o *CodeLens* não possui uma biblioteca de internacionalização diretamente na implementação do Runestone. Isso porque essa ferramenta possui um código fonte terceiro de um projeto chamado *Python Tutor*<sup>1</sup>, e é

<sup>1</sup> <http://pythontutor.com/>

embutido no HTML da página via um bloco *iFrame*. Por isso, para obter acesso ao código e traduzir suas *strings*, seria necessário entrar em contato com os desenvolvedores externos e, por esta razão, foi tomada a decisão de manter essa funcionalidade em inglês.

A implementação dos demais componentes utiliza a biblioteca chamada *i18n*<sup>2</sup> em JavaScript, que permite o armazenamento de tabelas de *strings* de diversas linguagens e, consequentemente, a tradução automática baseado na configuração de idioma do navegador dos usuários, desde que exista a tabela correspondente.

Os componentes que incluem essa biblioteca possuíam suporte apenas para as línguas inglês e sérvio. Para incluir a tradução para português brasileiro, o processo consistiu da criação de um novo arquivo na mesma estrutura que os demais.

Para ser reconhecido, o arquivo deve ser nomeado como "*activecode-i18n.<código-da-linguagem>.js*" na pasta "*js*" do componente correspondente. Além disso, todo componente possui um arquivo principal de configurações em *Python*, em que uma linha de código inclui o pacote *i18n* e identifica quais linguagens são suportadas na implementação. Portanto, para dar suporte ao português brasileiro, foram criados os arquivos "*activecode-i18n.pt-br.js*" para cada componente, contendo a mesma tabela de *strings* da versão em inglês mas com a tradução de todos os textos para português. Por fim, "*pt-br*" precisou ser incluído como uma das opções de linguagens disponíveis no arquivo de configuração do componente.

Importante ressaltar que nessa implementação todos os textos a serem traduzidos já estavam identificados no código fonte pelas variáveis do *i18n*. Ou seja, todas as *strings* do código utilizavam a estrutura "\$ .i18n("<chave-da-tabela>")" para serem reconhecidas. As traduções incluem principalmente textos relacionados a botões, mensagens de erro, mensagens de auxílio para correção desses erros, e o retorno (*feedbacks*) de algumas ações do usuário. Como exemplo, a tabela de tradução das *strings* do componente *Parsons* encontra-se na Figura 4.5.

```

runestone > parsons > js > parsons-i18n.pt-br.js > "en"
1  $.i18n().load({
2    "en": {
3      "msg_parson_check_me": "Verificar",
4      "msg_parson_reset": "Resetar",
5      "msg_parson_help": "Ajuda",
6      "msg_parson_too_short" : "Seu programa é muito curto. Adicione mais blocos.",
7      "msg_parson_drag_from_here" : "Arraste daqui",
8      "msg_parson_drag_to_here": "Largue os blocos aqui",
9      "msg_parson_correct_first_try": "Perfeito! Você levou apenas uma tentativa para resolver. Bom trabalho!",
10     "msg_parson correct":"Perfeito! Você levou $1 tentativas para resolver. Clique em Resetar para tentar resolver
11     "msg_parson_wrong_indent":"Este bloco não está indentado corretamente. Indente mais arrastando-o para a direita
12     "msg_parson_wrong_indents":"Estes blocos não estão indentados corretamente. Para indentar mais, arraste o bloco
13     "msg_parson_wrong_order":"Blocos destacados no seu programa estão errados ou estão na ordem errada. Isso pode s
14     "msg_parson_arrow navigate":"Use as teclas de setas para navegar. Espaço para selecionar/ desmarcar blocos para
15     "msg_parson help info":"Clique no botão Ajuda se você quiser facilitar o problema",
16     "msg_parson not solution":"Foi desabilitado um bloco de código desnecessário (que não faz parte da solução).",
17     "msg_parson provided indent":"Foi fornecida a indentação.",
18     "msg_parson combined blocks":"Dois blocos de códigos foram combinados em um.",
19     "msg_parson_remove incorrect":"Será removido um bloco de código incorreto da área de resposta",
20     "msg_parson will combine":"Serão combinados dois blocos",
21     "msg_parson atleast_three_attempts":"Você deve tentar pelo menos três vezes antes de pedir ajuda",
22     "msg_parson three blocks left" : "Restam apenas 3 blocos corretos. Você deve colocá-los em ordem",
23     "msg_parson will provide indent" : "Será fornecida a indentação"
24   }
25 });

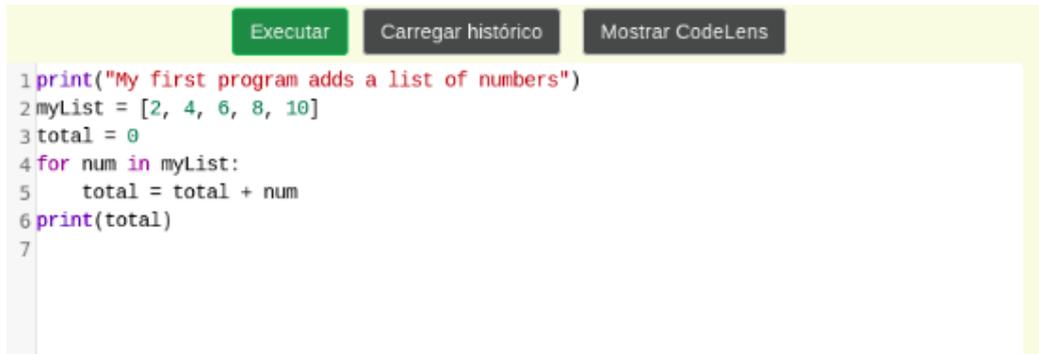
```

Figura 4.5: Tabela de tradução do componente *Parsons*.

<sup>2</sup> <https://www.npmjs.com/package/i18n>

Para a utilização dessa tradução, é necessário alterar o arquivo de configurações do livro desejado para a língua "pt-br", instalar a versão modificada dos componentes e compilar o livro utilizando essa instalação. O passo a passo está descrito no apêndice A.

Exemplos do resultado dessa primeira etapa estão nas Figuras 4.6 e 4.7.

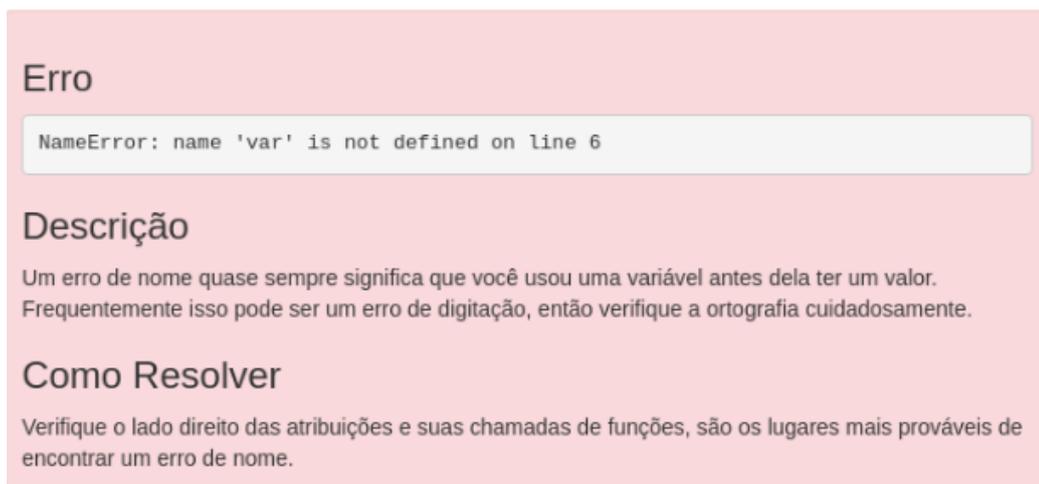


```

1 print("My first program adds a list of numbers")
2 myList = [2, 4, 6, 8, 10]
3 total = 0
4 for num in myList:
5     total = total + num
6 print(total)
7

```

**Figura 4.6:** Bloco ActiveCode com botões traduzidos.



**Erro**

NameError: name 'var' is not defined on line 6

**Descrição**

Um erro de nome quase sempre significa que você usou uma variável antes dela ter um valor. Frequentemente isso pode ser um erro de digitação, então verifique a ortografia cuidadosamente.

**Como Resolver**

Verifique o lado direito das atribuições e suas chamadas de funções, são os lugares mais prováveis de encontrar um erro de nome.

**Figura 4.7:** Erro de execução traduzido.

Já a segunda parte do processo de internacionalização do Runestone consistiu em traduzir algumas *strings* de páginas do servidor, tratando-se principalmente dos textos essenciais para a navegação como menus.

No entanto, não foi encontrado nenhum suporte implementado previamente para a internacionalização do servidor Runestone. A maneira encontrada de incorporá-la foi através do próprio *framework web2py* no qual a aplicação Runestone é instalada. Este *framework* disponibiliza uma ferramenta para identificação e tradução das *strings* por uma tabela, semelhante à biblioteca i18n.

Neste caso, as páginas não traziam os textos indexados, praticamente todas as *strings* encontravam-se "*hard-coded*" (ou seja, implementadas de maneira fixa e imutável). Para que fosse possível usar a tabela de tradução, foi necessário identificar os principais textos

e trocá-los pelas variáveis reconhecidas pelo *framework*. Para isso, foi feita a busca por textos localizados dentro de *tags* HTML através da expressão regular `/>([a-zA-Z\s]+)</`, e posteriormente a identificação manual de demais *strings* faltantes diretamente no código.

Todos os textos foram implementados com a variável `T("<string>")`, ou seja, na estrutura `{{=T("<string>")}}` dentro do código HTML, são identificados pelo *web2py* e a tradução pode ser feita em uma tabela criada em um arquivo *Python*. A nomenclatura do arquivo deve corresponder ao código da linguagem desejada, dentro da pasta "*languages*" da aplicação. As chaves da tabela são as próprias *strings* originais e os valores correspondentes são a tradução dessas *strings*.

A tabela também pode ser construída e atualizada pela interface do *web2py Admin* (Figura 4.8). Nela é possível acessar e editar os arquivos da aplicação, permitindo a atualização das tabelas de linguagens através de uma interface amigável para a inserção dos textos traduzidos. No entanto, apesar de exigir senha, não é recomendável que os sistemas em produção tenham a interface Admin disponível para livre acesso via navegador.

The image shows a screenshot of the web2py administrative interface. At the top, there is a dark header with the text "web2py™ interface administrativa". Below the header, there are three distinct sections, each with a title and a form element:

- Confirm Password:** A text input field containing "Confirmar Senha" and a button labeled "apagar".
- Constrain search to this course:** A text input field containing "Limitar busca para este curso" and a button labeled "apagar".
- Copy:** A text input field containing "Copiar" and a button labeled "apagar".

**Figura 4.8:** Tradução pela interface administrativa do *web2py*.

Nessa etapa, o maior problema encontrado foi o fato de que nem todos os textos poderiam ser identificados e traduzidos. Existem algumas *strings* geradas dinamicamente, ou seja, são construídas em códigos JavaScript e podem envolver diversas variáveis e condições, tornando impossível utilizar apenas a estrutura estática para identificá-las. Também não é ideal utilizar esse método para textos muito grandes, pois para que ele seja reconhecido não pode haver quebra de linha no código. Por isso, grandes parágrafos precisariam ser quebrados em várias frases e traduzidos por partes, o que torna o processo muito mais demorado e complicado.

O resultado final consistiu apenas na tradução de elementos principais para a compreensão da página e navegação, como menus, títulos de páginas e nomes de campos de preenchimentos como no login e cadastro. A maior parte dos textos que permaneceram em inglês é relacionada ao menu de administrador que apenas os instrutores conseguem aces-

sar, além de outros grandes trechos de texto explicativos. De maneira geral, a usabilidade foi facilitada, principalmente para os estudantes.

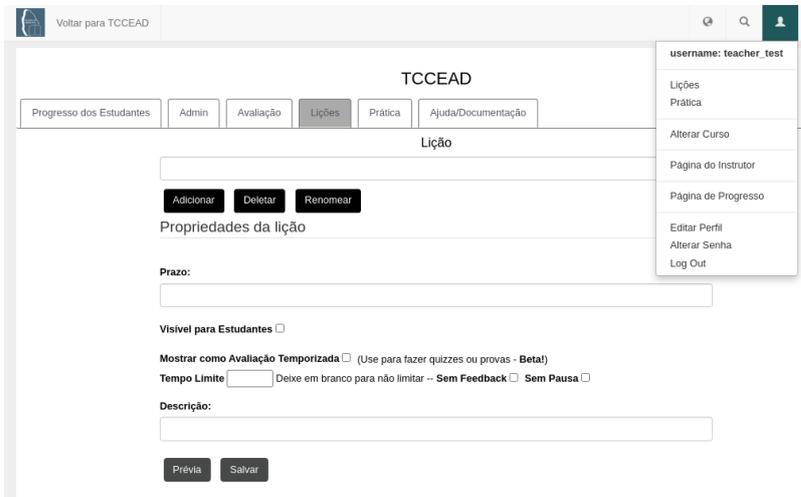


Figura 4.9: Menu Runestone do Instrutor traduzido.

## 4.2 Extensão lição Runestone para o Moodle

O desenvolvimento da extensão Moodle com todas as funcionalidades previamente planejadas iniciou-se com o estudo da estrutura de um *plugin*. O trabalho foi guiado inicialmente pela documentação da própria plataforma<sup>3</sup>, criando assim alguns arquivos básicos de configuração que todos os *plugins* possuem para gerenciamento do versionamento, permissões de acesso, linguagens e ícones.

Para a estruturação do *plugin* "Lição Runestone" foi baseado no código de demais extensões já existentes na plataforma que implementavam funcionalidades semelhantes.

### 4.2.1 Plugins base

Como uma Lição Runestone estaria dentro da categoria de atividades no Moodle e apresentaria praticamente as mesmas funcionalidades (criação, edição, visualização, deleção e possibilidade de associação a uma nota), a primeira extensão utilizada como referência foi a atividade básica chamada "Lição".

A Lição é constituída de diversas questões e páginas de conteúdo, criadas e configuradas pelo instrutor. As questões com respostas únicas e múltipla escolha são corrigidas automaticamente, mas as dissertativas precisam ser revisadas e avaliadas manualmente. Seu funcionamento é muito parecido com o das atividades do Runestone, sendo a única grande diferença o fato de que no *plugin* desenvolvido seria necessário criar as lições previamente na plataforma Runestone e depois associá-las ao Moodle.

Essa decisão de implementação se deu principalmente porque para criar uma lição Runestone via Moodle, o instrutor precisaria não somente das funcionalidades interativas

<sup>3</sup> <https://docs.moodle.org/dev/Tutorial>

do Runestone para pré-visualizar e criar leituras e questões, mas também ambos os sistemas precisariam estar totalmente integrados para criar as instâncias de forma simultânea nas duas plataformas. Dado o curto tempo disponível para a execução do projeto, não seria possível trazer as funcionalidades interativas para a interface Moodle, assim como seria inviável criar uma interface de comunicação (API) especializada em enviar e receber eventos de um sistema para o outro sem modificar totalmente suas estruturas essenciais de funcionamento, o que não é recomendado por serem sistemas extremamente complexos e de desenvolvimento terceiros.

Sendo assim, o *plugin* de Lição foi utilizado como base do *layout* das páginas (*frontend*) e do funcionamento (*backend*) relacionado às ações de criação, edição, deleção, visualizações da instância, e criação de notas associadas no banco.

Ainda pela mesma questão de não ser possível integrar as funcionalidades interativas do Runestone, foi necessário encontrar uma maneira do usuário realizar a Lição Runestone dentro de uma página de lição comum do Moodle. Para isso, a implementação foi baseado no recurso trazido pela extensão chamada "URL", na qual o usuário fornece um endereço URL e a página correspondente pode ser acessada através do Moodle de diversas maneiras. Uma dessas formas é apresentar a página web através de um *iFrame*, elemento HTML que permite a renderização de outro fragmento HTML embutido.

Dessa maneira, foi possível disponibilizar a página da lição Runestone através de sua URL e o usuário estudante pode realizar a lição pela plataforma Runestone sem sair da tela do Moodle. O maior desafio com relação a essa solução é o fato de que todos os usuários Moodle precisariam ter um usuário Runestone autenticado e matriculado no curso associado àquela lição. Esse ponto será mais explorado na seção 4.2.7.

Por fim, como nessa estratégia de implementação as lições ainda estariam sendo feitas e registradas no sistema Runestone, foi preciso estudar uma maneira de associar as instâncias de lições de uma plataforma para a outra e sincronizar as notas com o Moodle. Nesse sentido, o *plugin* chamado "Banco de Dados Externo" atendeu às necessidades, pois sua função está relacionada à sincronização de matrículas a partir de um outro banco de dados. Suas configurações e a estrutura do *plugin* foram utilizadas de maneira semelhante no desenvolvimento da lição Runestone para poder acessar o banco de dados do Runestone e importar as notas das lições. A principal diferença é que esta é uma tarefa que não deveria ter que ser acionada manualmente pelos instrutores e, por isso, deve ser executada automaticamente com certa frequência. Assim, a tarefa de sincronização de notas foi associada ao Cron<sup>4</sup> do Moodle, ferramenta que executa tarefas e scripts em períodos programados.

## 4.2.2 Configurações

Todo *plugin* para o Moodle possui uma página exclusiva para os administradores do sistema (Figura 4.10). Elas trazem configurações essenciais para o funcionamento dos *plugins* e alguns ajustes.

---

<sup>4</sup> [https://docs.moodle.org/all/pt\\_br/Cron](https://docs.moodle.org/all/pt_br/Cron)

Visão geral dos Plugins

Verificar atualizações disponíveis

Última verificação feita em 12 January 2021, 00:44 AM

Todos os plugins 412 Plugins adicionais 1

Nome do plugin	Versão	Disponibilidade	Ações	Notas
<b>Atividades</b>				
Tarefa mod_assign	2020061500	Habilitado	Configurações	Desinstalar
Tarefa 2.2 (Desabilitado) mod_assignment	2020061500	Desabilitado		Desinstalar
Livro mod_book	2020061500	Habilitado	Configurações	Desinstalar
Chat mod_chat	2020061500	Habilitado	Configurações	Desinstalar
Escolha mod_choice	2020061500	Habilitado		Desinstalar
Base de dados mod_data	2020061500	Habilitado	Configurações	Requerido por: filter_data
Pesquisa mod_feedback	2020061500	Habilitado	Configurações	Requerido por: block_feedback
Pasta mod_folder	2020061500	Habilitado	Configurações	Desinstalar
Fórum mod_forum	2020061501	Habilitado	Configurações	Requerido por: block_news_items, block_search_forums

**Figura 4.10:** Página de plugins e suas configurações para administradores.

Como o desenvolvimento do *plugin* depende da sincronização de notas com o banco de dados externo do Runestone, são necessários alguns dados para permitir essa conexão. Baseado nas configurações do *plugin* "Banco de Dados Externo" citado anteriormente, são requisitados como informações essenciais para a conexão: o *driver* do banco (por exemplo, MySQL, Postgres, ...), nome/endereço IP do servidor, credenciais para autenticação (usuário e senha), nome do banco de dados e codificação. A conexão com o banco e as operações são realizadas no Moodle pela biblioteca AdoDB<sup>5</sup>, uma camada de abstração para manipulação de bancos de dados para PHP.

Outra configuração inclusa nessa página foi o domínio (URL) do sistema Runestone que será usado, uma vez que a visualização das lições depende desse endereço para ser renderizado via *iFrame*.

<sup>5</sup> <https://adodb.org/>

## Lição Runestone

Configure o acesso para seu banco de dados para sincronizar usuários, cursos e notas. Estamos assumindo que seu banco possui os nomes de tabelas e colunas padrões do Runestone.

### Domínio do Runestone

URL Runestone <small>mod_runestone   externalurl</small>	<input type="text" value="https://runestone.academy/"/>	Padrão: Vazio
---	---	---------------

A URL do seu domínio Runestone. Ex: "https://runestone.academy/"

### Conexão do banco Runestone

Driver do banco de dados <small>mod_runestone   dbtype</small>	<input type="text" value="postgres"/>	Padrão: Vazio
---	---------------------------------------	---------------

Nome do driver de base de dados ADOdb, tipo da base de dados externa.

Nome ou número IP do servidor <small>mod_runestone   dbhost</small>	<input type="text" value="localhost"/>	Padrão: localhost
--	--	-------------------

Digite o endereço IP do servidor de banco de dados ou o nome do host. Utilize um nome DSN do sistema se estiver utilizando ODBC, ou um PDO DSN se estiver utilizando PDO.

Usuário do banco de dados <small>mod_runestone   dbuser</small>	<input type="text" value="admin"/>	Padrão: Vazio
--	------------------------------------	---------------

Senha do banco de dados  
mod\_runestone | dbpass

.....  

Figura 4.11: Parte da tela de configurações do Plugin.

A implementação dessa página consistiu apenas em um arquivo chamado `settings.php` no diretório do *plugin*, no qual é necessário apenas declarar os campos do *frontend*. A partir do nome da extensão e do identificador do campo, o Moodle automaticamente armazena todos os dados inseridos pelo usuário em uma tabela específica de configurações, permitindo resgatar em outros arquivos qualquer informação salva através da função `get_config()`.

### 4.2.3 Criação, atualização e deleção

Para a criação, atualização e deleção de uma instância de "Lição Runestone", o primeiro passo consistiu em entender as propriedades de uma lição e estruturar uma tabela no banco de dados do Moodle para armazenar esses objetos.

Os principais campos de uma lição são relacionados a propriedades básicas inseridas na criação da instância, como: nome da atividade a ser exibido na página do curso, descrição, e relação com a instância da lição no banco de dados do Runestone (*id* do curso e da lição). Todos os arquivos relacionados ao banco de dados e permissões devem estar dentro de uma pasta nomeada "db" dentro do diretório do *plugin*. A estrutura das tabelas é implementada em um arquivo `.xml` que o Moodle reconhece no momento da instalação da extensão.

A implementação do funcionamento da criação, atualização e deleção das instâncias de uma lição dependeu de três funções principais. Todas as funções essenciais de um *plugin* encontram-se em um arquivo chamado "lib.php" e devem seguir um padrão de nomenclatura para serem reconhecidas e chamadas pelo Moodle. Nesse caso, foram criadas as funções `runestone_add_instance()`, `runestone_update_instance()` e `runestone_delete_instance()`. O funcionamento do projeto também depende de algumas funções auxiliares que se encontram no arquivo "locallib.php".

A criação e a atualização de uma lição funcionam de maneira semelhante. As funções recebem o objeto com todas as propriedades da instância e as informações são salvas na tabela correspondente ao *plugin*. Já a deleção recebe apenas o `id`, e apaga do banco todas as informações relacionadas a esse registro.

Por fim, o último passo desta etapa foi criar a interface do usuário para criar ou editar uma instância de uma lição. Ambas as operações renderizam a mesma página, uma vez que a única diferença é que a edição traz e atualiza um objeto já existente. O Moodle diferencia a operação a ser executada através do fluxo de navegação do usuário. A criação de uma instância vem da opção de adição de uma atividade dentro de um curso. Já a atualização precisa estar associada à opção de edição da instância, que pode ser acessada através da lista de atividades na página principal do curso ou da página de visualização da lição.

Adicionando um(a) novo(a) Lição Runestone em Tópico 1 [Expandir tudo](#)

▼ Geral

Nome ! Lição 01 - Funções

Curso Runestone ! CursoTcc

Atividade Runestone ! Introdução a Funções

Descrição

↕ A B I
☰ ☰ ☰ ☰
🔗 🔄
😊 🖼️ 📎 🎤 🗣️ 📄 H-P

Esta lição contempla exercícios de reforço para o conceito de funções em programação

ⓘ Uma versão anterior deste texto foi automaticamente restabelecida.

Exibir descrição na página do curso !

Mostrar descrição da URL

**Figura 4.12:** Página de criação de uma instância Lição Runestone.

A visão da criação e edição é desenvolvida no arquivo "mod\_form.php" e utiliza a implementação de formulários do próprio Moodle, na qual é necessário apenas declarar quais campos deverão ser preenchidos pelo usuário, o tipo do dado, obrigatoriedade e validação. Quando o usuário preenche os dados e salva, as respectivas funções são chamadas automaticamente.

O maior desafio no desenvolvimento dessa página foi trazer como um campo de lista de seleção (*dropdown*) todos os cursos e lições cadastradas na plataforma Runestone associada ao *plugin*. Inicialmente, a instância do *plugin* foi estruturada para receber diretamente a URL da lição, o que tornava o processo do instrutor mais difícil, uma vez que ele teria que

acessar a plataforma Runestone externamente e obter a *link* correspondente. Para promover uma melhor experiência, foi construído um módulo em JavaScript, utilizando a tecnologia AJAX, para obter os nomes e os identificadores de todos os cursos e lições do banco de dados externo em tempo real na renderização. Assim, o usuário pode apenas selecionar a lição desejada pelo seu nome, e o *backend* do Moodle constrói a URL dinamicamente a partir disso.

Com relação à deleção de uma Lição, não há uma página específica que precisou ser desenvolvida, pois essa operação fica disponível na listagem de atividades dentro da tela de detalhes do próprio curso.

#### 4.2.4 Visualizações

Como citado em seções anteriores, a visualização de uma lição Runestone é uma página HTML com um elemento *iFrame* embutido que acessa e exibe a URL da lição no servidor Runestone.

O arquivo principal para a renderização dessa página é o "view.php". Seu funcionamento consiste em checar algumas condições do curso e do usuário, como certificar-se de que o usuário está logado e qual seu nível de permissão de acesso, e chamar métodos auxiliares para realizar a renderização de acordo com cada contexto.

Diferentemente dos estudantes, os administradores do curso, instrutores e monitores não apenas visualizam a lição, mas conseguem acessar outras abas para editar a lição, dar notas ou acessar as estatísticas dos estudantes no progresso do livro, desde que tenham usuários Runestone com permissão de instrutor desse curso.

Para essa diferenciação, foi criado um arquivo "tabs.php" que checa as permissões do usuário e exibe as abas "Visualização", "Edição", "Relatórios" e "Avaliar" caso ele possua acesso ao gerenciamento da atividade (Figura 4.13). Para cada uma das abas extras, foram criadas as visualizações em arquivos separados ("report.php", "edit.php" e "grade.php"), cuja diferença está em qual URL é renderizada pelo *iFrame*. Todas as URLs são geradas dinamicamente por funções auxiliares dependendo da aba ou da lição a ser renderizada.

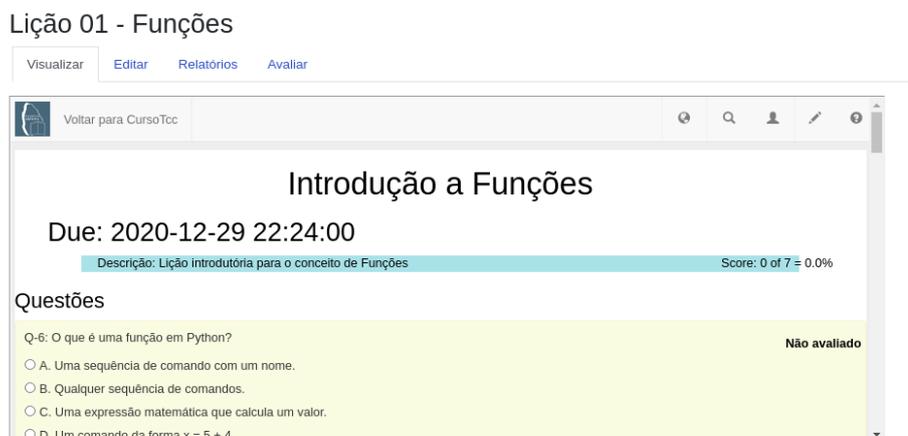


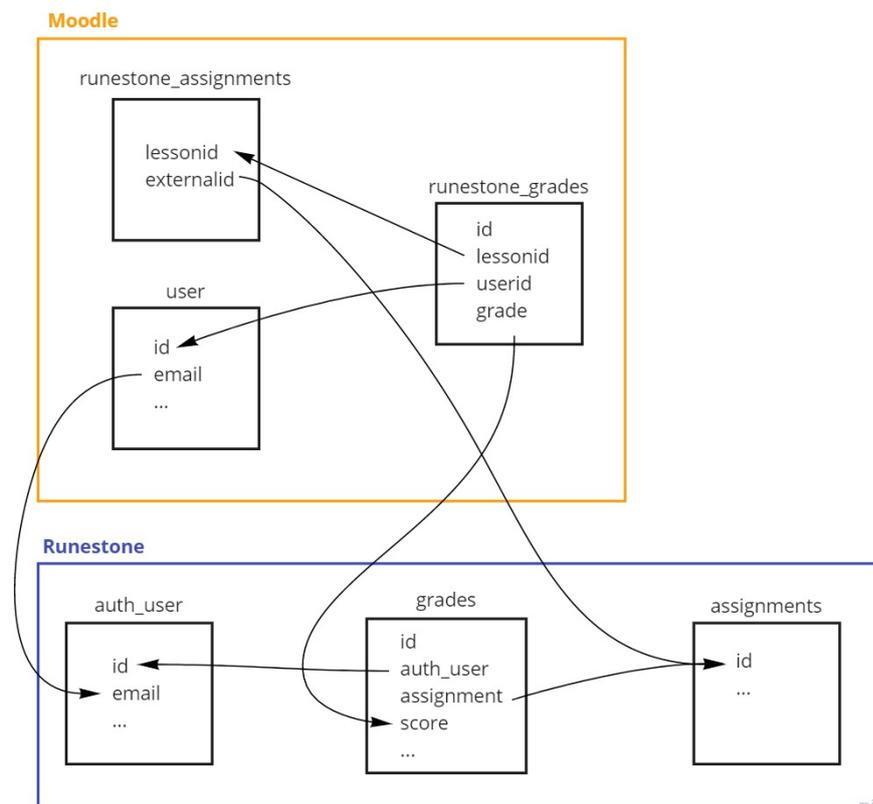
Figura 4.13: Visualização de uma Lição pelo Instrutor.

Para que essa visualização funcione, é importante garantir que o Moodle e o Runestone estejam em mesmo domínio ou que o domínio do Runestone garanta permissão de acesso CORS<sup>6</sup> ao domínio do Moodle. Isso porque, por questões de segurança, os navegadores possuem uma política de controle de requisições HTTP dentro de um *iFrame*, e sem essa permissão não é possível logar e acessar as áreas restritas do servidor do Runestone.

#### 4.2.5 Sincronização de notas

Após a inserção dos dados essenciais para conexão com o banco de dados do Runestone na página de configurações do *plugin*, a sincronização das notas é feita em uma função chamada `sync_grades()` que usa as credenciais para se conectar ao banco externo e realizar operações de leitura.

Seu processo consiste em, para cada instância de lição, identificar os estudantes que estão matriculados tanto no curso Moodle quanto no curso Runestone através da chave e-mail. A partir disso, ele procura se existem notas dessa lição para esses estudantes. Se houver, a nota é escrita no banco de dados do Moodle na tabela `runestone_grades`, criada especificamente para armazenar esses dados, relacionando uma nota a uma instância de lição Runestone e a um estudante (Figura 4.14).



**Figura 4.14:** Esquematização das relações entre os dois bancos na sincronização de notas.

O Moodle também possui uma estrutura própria de coleção e manipulação de notas

<sup>6</sup> [https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Controle\\_Acesso\\_CORS](https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Controle_Acesso_CORS)

dentro dos cursos, com funcionalidades específicas para os instrutores administrarem junto com as médias finais. Portanto, para que as notas das Lições Runestone sejam reconhecidas e utilizadas como parte do conjunto de notas oficiais do curso, também é necessário salvá-las na tabela de notas gerais chamada `grade_items`, criando assim um item de pontuação que poderá ser acessado por instrutores e estudantes junto com as demais notas do curso. No entanto, é importante ressaltar que as mudanças feitas diretamente nas notas de Lições Runestone no Moodle não são refletidas no sistema Runestone, e sim o inverso. Por isso, as avaliações precisam estar sempre atualizadas no banco de dados do Runestone.

## Relatório de notas

Ver Configurações Escalas Letras Importar Exportar

Relatório de notas Histórico de notas Relatório de resultados Relatório geral Visão única Relatório do usuário

Todos os participantes: 1/1

Nome Todos A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Sobrenome Todos A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Nome / Sobrenome		Endereço de email	Curso Teste	Total do curso
Aluno Teste		aluno@teste.com	Lição 01 - Funções 5,00	5,00
Média geral		Média geral	5,00	5,00

**Figura 4.15:** Visualização de notas Runestone no curso Moodle.

Essa rotina precisa ser executada com frequência, para que os estudantes possam acessar suas notas assim que as lições forem avaliadas. Por isso, foram desenvolvidas duas maneiras de chamar a função `sync_grades()`: chamando o script "sync.php" via linha de comando, ou executando a tarefa `sync_task` através da execução do Cron do Moodle.

No Moodle, o Cron é um script que busca e executa diversas tarefas agendadas, auxiliando na manutenção do funcionamento de diversos módulos e do próprio sistema. Por exemplo, ele é responsável por excluir definitivamente todas as atividades e arquivos enviados para a fila de deleção, enviar e-mails de notificações das postagens de fóruns, entre outras tarefas rotineiras. Apesar de também ser possível executar esse script via linha de comando, é altamente recomendado que ele seja configurado por outras aplicações que gerenciam o intervalo de tempo para a execução do Cron. O tempo recomendado de intervalo entre as execuções é de 5 a 15 minutos.

Dessa forma, ao criar a tarefa agendada no Cron para a sincronização de notas Runestone, de forma a garantir que ela seja executada de forma regular juntamente com outras atividades essenciais do sistema.

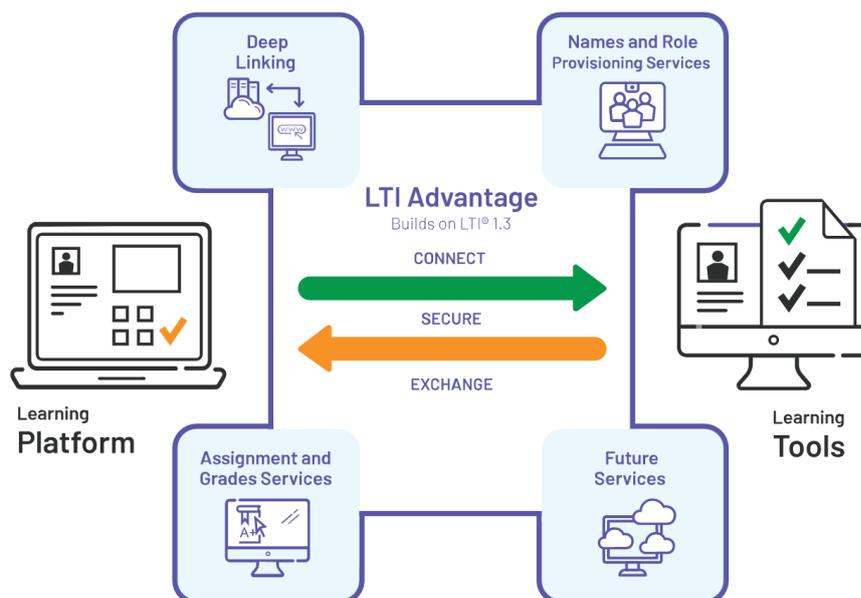
## 4.2.6 Autenticação

Um dos problemas encontrados no fluxo de utilização do *plugin* é como seria possível tornar a autenticação do Runestone transparente para os usuários Moodle, sem que cada estudante precisasse criar uma conta e se autenticar manualmente ao acessar uma Lição Runestone no Moodle.

Durante o desenvolvimento, foram priorizados estudos de soluções automáticas e simples, que gerassem menos encargo para os administradores. As formas de identificação do usuário estudadas foram o LTI (*Learning Tools Interoperability*), que tenta padronizar a passagem de informações através de plataforma educacionais, e o *Single-Sign On* que é utilizado em várias plataforma da USP a partir do sistema *idpcafe*, feito com Shibboleth.

### Learning Tools Interoperability

Learning Tools Interoperability (LTI)<sup>7</sup> é uma especificação tecnológica criado pela *IMS Global Learning Consortium* para padronizar a forma de comunicação entre vários sistemas educacionais, incluindo Canvas, Blackboard, Sakai e Moodle.



<http://www.imsglobal.org/>

Figura 4.16: Vantagens do LTI.

A ideia por trás dessa especificação é que se pode enviar e receber informações entre sistemas educacionais sem precisar se preocupar como isso é feito. O LTI possui diferentes versões e cada uma delas adiciona mais funcionalidades que auxiliam na comunicação entre as plataformas, como é possível ver em *Learning Tools Interoperability 2016*.

As informações do LTI possuem um padrão a ser seguido, como por exemplo:

<sup>7</sup> <https://www.imsglobal.org/activity/learning-tools-interoperability>

lis_person_name_given	Nome ou nome dado
lis_person_name_family	Sobrenome ou nome da família
lis_person_name_full	Nome completo
lis_person_contact_email_primary	E-mail primário

O LTI foi integrado no Runestone no final do período em que este trabalho foi feito. Em seu código há comentários que evidenciam que o sistema foi testado para as plataformas Canvas e Blackboard, mas não houve menção sobre sua utilização com o Moodle.

O Moodle possui um *plugin* para ferramentas externas que utiliza a versão 1.0/1.1, 1.3 ou 2.0 do LTI para se comunicar com plataformas via um *iFrame*. A partir de um teste, constatou-se que ambos os sistemas funcionam juntos e os estudantes poderiam realizar lições do Runestone através dessa extensão sem a necessidade de se autenticar manualmente. No entanto, sua configuração é complexa e precisa ser realizada por administradores do sistema, não sendo trivial para instrutores disponibilizarem lições específicas para os estudantes.

Assim, a solução consistiria em incluir apenas a forma de autenticação do LTI em *iFrame* para o *plugin* Lição Runestone. No entanto o código da implementação do LTI é imenso e extremamente complexo, tornando complicada a manutenção da compatibilidade com o desenvolvimento do *plugin*. Seria, então, necessário criar do zero um sistema de autenticação semelhante ao LTI para este projeto, o que foi considerado inviável no tempo restante.

## Shibboleth

Shibboleth<sup>8</sup> é um sistema com código aberto para facilitar a autenticação de usuários. Ele utiliza a informação de pessoas dentro de federações ou organizações para fazer a autenticação.

A plataforma e-Disciplinas, que é também conhecido como o Moodle unificado da USP, utiliza essa forma de autenticação para identificar seus usuários. Assim, foi considerado uma possibilidade aproveitar essa forma de identificação para autenticar os usuários no sistema do Runestone.

O sistema de autenticação do Shibboleth tem duas partes principais, sendo necessário definir primeiramente algumas terminologias desse sistema.

## Terminologias

Provedor de identificação (IdP) - esse servidor é usado para configurar as informações utilizadas no provedor de serviço.

Provedor de serviço (SP) - complemento para o servidor de identificação que é utilizado para proteger certas páginas. E quando é acessado redireciona para o provedor de identificação.

Security Assertion Markup Language (SAML) - um padrão aberto que permite que provedores de identidade e provedores de serviços passem informações de credenciais de

<sup>8</sup> <https://wiki.shibboleth.net/>

autorização entre si.

Usuário - a pessoa que quer acessar no sistema com acesso restrito.

Cliente - a máquina do usuário.

Servidor - a máquina onde acontece os serviços do provedores de autenticação.

### Processo de Autenticação

O processo de autenticação do Shibboleth será explicado de maneira simplificada, desde quando o usuário entra no *site* até o momento em que ele recebe a autorização de acesso após a autenticação.

Supondo que o site do SP está em <https://site/> e o site do IdP está em <https://default/>, o processo se inicia quando o usuário entra no site <https://site/>, o qual possui acesso restrito e é protegido pelo Shibboleth SP.

O site pede informações do usuário para o Shibboleth SP, mas como o cliente não possui nenhuma informação, o usuário é redirecionado para o IdP. Uma mensagem em SAML é enviada do SP para IdP para identificar o início do login e guarda as informações do site de origem.

O cliente é redirecionado para a interface de login de IdP. Durante esse processo, a informação do SAML é enviada para [https://default/idp/profile/SAML2/Redirect/SSO?....](https://default/idp/profile/SAML2/Redirect/SSO?...) O IdP verifica a requisição do SAML identificando a origem do SP, e também cria um *cookie* para acompanhar o processo do login.

Quando o usuário loga no sistema com seu usuário e a senha, o cliente manda um requisição POST para <https://default/idp/profile/SAML2/Redirect/SSO?execution=e1s1>. Essa requisição incluirá a informação do *cookie* do processo de login, portanto o IdP verifica o login e cria uma nova seção autenticada, retornando um *cookie* que é compartilhado apenas para *URLs* dentro do IdP.

O cliente é redirecionado para uma tela que pede para o usuário aprovar o uso das informações dentro do site. Após a permissão ser concedida, o cliente manda para <https://default/idp/profile/SAML2/Redirect/SSO?execution=e1s2> uma requisição POST que possui informação dos dois *cookies*. O usuário é redirecionado para o site original e também manda um POST para <https://site/Shibboleth.sso/SAML2/POST> para o SP criar uma sessão do Shibboleth e liberar o acesso no site.

Na USP, o servidor de Shibboleth utilizado é o *idpcafe*. Se o *plugin* "Lição Runestone" for utilizado em sua plataforma, e-Disciplinas, seria possível aproveitar o sistema de autenticação existente. No entanto, seria também necessário criar uma integração do Shibboleth com o web2py e com o Runestone, o que seria um trabalho grande e fora do escopo dos objetivos do projeto. Além disso, seu uso estaria limitado ao *idpcafe* e o *plugin* não poderia ser utilizado por outras instâncias do Moodle.

Assim, as tentativas de implementação de um sistema de autenticação entre o *plugin* do Moodle e o Runestone foram descartadas nesse primeiro protótipo, uma vez que isso não impede o uso de suas funcionalidades. A falta desse sistema apenas exige um login

manual dos usuário do Moodle. Essa melhoria pode, portanto, ser explorada em trabalhos futuros.

#### 4.2.7 Cadastro de estudantes no Runestone

Para que os usuários do Moodle consigam ter acesso às lições é necessário que todos estejam cadastrados no servidor Runestone com as permissões corretas e matriculados no curso que possui a lição desejada.

Durante os estudos de possíveis soluções para esse problema foram encontrados dois principais obstáculos. O primeiro desafio seria implementar no Runestone um módulo de conexão com o banco de dados Moodle, uma vez que a migração deveria ser feita pelo sistema Runestone para obter os dados dos usuários.

Como o Runestone, diferentemente do Moodle, não possui um sistema de modularização ou criação de *plugins*, não há uma preparação do sistema para receber extensões de funcionalidades. Assim, alterar sua arquitetura para receber essa integração não faria sentido por estar fora do escopo dos objetivos do projeto Runestone.

O segundo obstáculo seria a questão da autenticação. Como o objetivo de realizar uma autenticação transparente não foi atingido, como explorado na seção 4.2.6, os usuários precisam inserir suas senhas e logar no sistema manualmente em seu primeiro acesso. Nesse sentido, não seria possível criar usuários Runestone que utilizassem a mesma senha de seu usuário Moodle correspondente, pois o dado é criptografado. Isso criaria mais um atrito na experiência de uso.

Dada a finalidade do projeto de criar um protótipo inicial, a decisão tomada foi de manter o cadastro de estudantes de forma manual para os administradores do sistema.

No Moodle, os administradores podem extrair os dados de usuários em um arquivo .csv, enquanto o servidor Runestone aceita o cadastro de estudantes no mesmo formato, em uma estrutura de arquivo semelhante. Tendo isso em vista, foi desenvolvido um script para o auxílio desse processo. A partir da inserção do arquivo de usuários do Moodle é gerado um outro arquivo com os mesmos usuários na estrutura esperada pelo Runestone. Suas senhas também são aleatoriamente geradas neste processo.

Dessa forma, o administrador pode cadastrar o arquivo gerado no servidor Runestone e posteriormente enviar as senhas de acesso para todos os endereços de e-mail. Trabalhos futuros podem investir na automatização desse processo como um todo.

### 4.3 Adaptação da plataforma Panda

Panda é uma plataforma criada sobre o *framework* web2py. Ela é utilizada para disponibilizar materiais relacionados à ciência da computação que são desenvolvidos e usados no IME. Possui, por exemplo, materiais para introdução a algumas linguagens de programação como C, Java, Python.

A Panda atualmente disponibiliza alguns dos livros do Runestone, como o *How to Think like a Computer Scientist* traduzido para o Português. Porém, os usuários acessam os livros

como visitantes e por isso não conseguem salvar seu progresso, incluindo anotações e tentativas de exercícios.

Assim, foi criado um sistema compartilhado das sessões de usuário do Panda com o Runestone utilizando a autenticação padrão do web2py. Dessa forma, os visitantes podem se registrar e se autenticar na Panda, e suas informações de sessão são automaticamente passadas para a aplicação web2py do Runestone.

Departamento de Ciência da Computação - IME - USP

- [Autentique-se](#)
- [Registre-se](#)

## Panda - Cursos de Computação

Principal
C
Java
Python



**IME** INSTITUTO DE MATEMÁTICA  
E ESTATÍSTICA  
UNIVERSIDADE DE SÃO PAULO

### Registre-se

Usuário:

Nome:

Sobrenome:

Email:

Senha:

Repita a Senha:

Eu aceito o termo de serviço:

[Registre-se](#)

### Material:

[Para cursos em C](#)

[Para cursos em Java](#)

[Para cursos em Python](#)

Copyright (c) 2015 DCC/IME/USP. All rights reserved. powered by [web2py](#)

**Figura 4.17:** *Página de registro na Panda.*

Departamento de Ciência da Computação - IME - USP

- [Autentique-se](#)
- [Registre-se](#)

## Panda - Cursos de Computação

Principal
C
Java
Python



**IME** INSTITUTO DE MATEMÁTICA  
E ESTATÍSTICA  
UNIVERSIDADE DE SÃO PAULO

### Autentique-se

Usuário:

Senha:

Lembre-me (por 30 dias):

Log In [Registre-se](#)

### Material:

[Para cursos em C](#)

[Para cursos em Java](#)

[Para cursos em Python](#)

Copyright (c) 2015 DCC/IME/USP. All rights reserved. powered by [web2py](#)

**Figura 4.18:** *Página de autenticação na Panda.*

Para o registro do progresso no banco de dados, os livros precisam estar dentro de um curso Runestone. Assim, um usuário não autenticado é redirecionado para a página do livro aberto, e um usuário autenticado é redirecionado para o curso contendo esse livro.

Um obstáculo encontrado na arquitetura do Runestone foi que usuários recém logados guardam um *cookie* contendo informações do curso acessado, e isso causaria erros de permissão de acesso caso a pessoa tentasse acessar o livro aberto em seguida sem autenticação. Isso acontece porque o *link* de redirecionamento para um curso e o *link* para seu livro aberto é o mesmo. Para solucionar esse problema, foram criadas duas instâncias diferentes de cada livro a ser disponibilizado na Panda.

Para permitir o acesso dos usuários aos cursos, todo novo cadastro automaticamente matricula o usuário a todos os cursos disponíveis na instância do Runestone da Panda. Por isso, o Runestone utilizado na Panda não deve ser compartilhado, como por exemplo, com a instância utilizada com o Moodle.



## Capítulo 5

# Considerações Finais

Ao final do projeto obteve-se um resultado satisfatório aos objetivos estabelecidos, uma vez que o Runestone se tornou um pouco mais acessível aos estudantes brasileiros através da tradução de partes do sistema para português e também pela integração com as plataformas Moodle e Panda.

Porém, muitos dos planejamentos iniciais do projeto não foram atendidos da maneira ideal, pois no decorrer do processo de desenvolvimento surgiram diversos desafios.

Uma das principais dificuldades encontradas durante o processo foi trabalhar a partir de grandes sistemas prontos, projetados em uma estrutura fixa e desenvolvidos por terceiros. Isso acaba limitando a flexibilidade na implementação de novas soluções.

O Moodle, por exemplo, é uma plataforma mais consolidada e conhecida ao redor do mundo, possui uma boa documentação auxiliar para desenvolvedores (porém longe de ser completa) e uma comunidade relativamente grande. No entanto, é um sistema cujo funcionamento é muito complexo e isso torna o desenvolvimento e entendimento do código complicados, principalmente para um trabalho com curto período de tempo dedicado.

Por outro lado, o Runestone é um projeto mais recente e ainda não tão conhecido, não dispondo de muitas informações disponíveis sobre sua arquitetura e nem uma comunidade ativa que possa tirar dúvidas e auxiliar no desenvolvimento. Grande parte do tempo, portanto, foi destinado ao entendimento de código, detecção e solução de problemas que inicialmente pareciam simples.

Além disso, o resultado ideal seria uma integração total entre as duas plataformas, o que não foi possível de ser realizado neste projeto porque seria necessário compreender e alterar o funcionamento de partes essenciais de cada sistema. Isso não apenas traria o risco de causar grandes efeitos colaterais e gerar problemas de funcionamento, mas também seriam modificações muito específicas para este projeto e que não teriam suporte em futuras atualizações, tornando trabalhos futuros não escaláveis.

De modo geral, o resultado do trabalho foi um bom primeiro passo para facilitar a utilização de livros interativos e novas metodologias no ensino da Computação. Espera-se

que o projeto possa ser futuramente continuado para impactar positivamente ainda mais estudantes e instrutores da área.

## Trabalhos Futuros

Os próximos passos consistem em revisar e aperfeiçoar as implementações realizadas. A internacionalização do Runestone precisa ser aprimorada para reconhecer todas as *strings* e facilitar a tradução para qualquer linguagem.

Com relação às funcionalidades do *plugin* Moodle, é necessário criar ou encontrar um sistema de autenticação que permita maior transparência do sistema Runestone aos usuários Moodle. Para isso também é necessário implementar uma arquitetura que crie e associe usuários do Moodle a usuários do Runestone, preferencialmente sendo cadastrados automaticamente nos cursos desejados, ou ao menos disponibilizando uma interface administrativa para essa finalidade.

Além disso, existem várias possibilidades de funcionalidades a serem exploradas de acordo com a necessidade de cada tipo de usuário. Por exemplo, para os instrutores pode ser útil desenvolver uma interface Moodle para criar e administrar cursos Runestone. Ou criar um outro *plugin* mais focado em disponibilizar capítulos dos livros para leitura, e não somente as lições. O papel de administrador também precisa ser revisado, pois neste momento a solução desenvolvida possui diversas funções manuais sem uma interface unificada para que o *plugin* possa gerenciar os sistemas.

Por fim, é necessário compreender como estudantes e instrutores irão utilizar essas ferramentas. O trabalho deve ser continuamente testado dentro e fora da Universidade de São Paulo para que, através de *feedbacks*, sejam estabelecidas as prioridades de desenvolvimentos futuros, focando sempre em melhorar a experiência geral dos usuários e expandir seu uso.

# Apêndice A

## Instruções de Uso

As próximas seções foram estruturadas para auxiliar na instalação e configuração das ferramentas desenvolvidas neste projeto. Estão separadas em: instalação e configuração do Runestone, como utilizar a tradução para português, instalação e configuração do plugin Moodle para Lição Runestone e configuração da Panda.

### A.1 Instalação e configuração do Runestone

Para fazer a instalação do Runestone, pode-se seguir as instruções da documentação oficial do Runestone em <https://runestoneserver.readthedocs.io/en/latest/docs/installation.html>.

Durante o projeto foi utilizado o wsgi do web2py para disponibilizar o site, então algumas modificações extras foram feitas nesse processo de instalação e configuração.

O código do wsgi pode ser encontrado em `handlers/wsgihandler.py`, para conseguir inserir as variáveis de ambiente no wsgi foi necessário a inserção do seguinte trecho de código para configurar as informações do bancos de dados.

```
1 os.environ['WEB2PY_CONFIG'] = 'production'
2 os.environ['WEB2PY_MIGRATE'] = 'Yes'
3 os.environ['DBURL'] = 'postgresql://{username}:{password}@{hostname}/{
    database}'
4 os.environ['TEST_DBURL'] = 'postgresql://{username}:{password}@{hostname}/{
    database}'
5 os.environ['DEV_DBURL'] = 'postgresql://{username}:{password}@{hostname}/{
    database}'
```

O *username* e o *password* precisam ser trocados pelo usuário e pela senha e o *hostname* e o *database* pelo nome do servidor e pelo nome do banco de dados.

Para acessar a configuração das internacionalizações do web2py é necessário criar uma senha para o sistema administrativo a partir do seguinte comando:

```
1 $ python3 -c "from gluon.main import save_password; save_password('{
    password_web2py}',443)"
```

É necessário trocar `password_web2py` para a senha utilizada no sistema administrativo do web2py. Após a criação da senha é possível entrar no admin e no appadmin.

A configuração do Apache é feita usando um script do web2py existente em `scripts/setup-web2py-ubuntu.sh`. Para executá-lo use `sh setup-web2py-ubuntu.sh`

Depois de instalar e configurar o Apache, é preciso atualizar a versão do wsgi de Python2 para Python3. O seguinte pacote deve ser instalado:

```
$ sudo apt-get install libapache2-mod-wsgi-py3
```

É necessário dar permissão de acesso para o wsgi, executando o seguinte comando na pasta que se quer dar acesso:

```
$ sudo chown -R www-data:www-data /home/www-data/web2py
```

O gerenciador de pacotes `venv` foi criado na pasta `/home/www-data` para o usuário `www-data`, e esse usuário conseguirá acessar os pacotes do Python3.

No início do código do Apache2 é preciso colocar para o web2py ter acesso aos pacotes de Python e permissões do usuário `www-data`:

```
1 WSGIDaemonProcess web2py user=www-data group=www-data python-path=/home/www-
  data:/home/www-data/lib/python3.6/site-packages
```

Para modificar as rotas de urls do web2py do padrão com `welcome` para `runestone` podemos usar o seguinte script:

```
1 $ cp examples/routes.patterns.example.py routes.py
2 $ sed -i "s/default_application = 'isnit'/default_application = 'runestone'/
  g" routes.py
3 $ sed -i "s/r'myapp'/r'runestone'/g" routes.py
```

## A.2 Tradução do Runestone

A tradução dos arquivos do servidor pode ser encontrada no repositório <https://github.com/larisala/ime-runestone-server>. Importante ressaltar que essa versão inclui também as alterações de algumas páginas para adaptá-las aos servidores da universidade. Por conta disso e da tradução incompleta, essas mudanças não foram incorporadas no desenvolvimento oficial do Servidor Runestone. Assim, não é possível garantir a compatibilidade das modificações em versões futuras.

Para utilizar essa versão da aplicação, é necessário primeiramente realizar a instalação do Runestone 5.1.2. A partir dela, baixe e coloque as pastas do repositório no diretório do runestone dentro das aplicações do web2py.

A tradução dos Componentes Runestone, por sua vez, foi feita na versão 4.2.5, e pode ser encontrada no repositório <https://github.com/larisala/tcc-ead-2020>.

Para utilizar a tradução, primeiro instale o conteúdo do repositório com o comando:

```
$ pip install git+git://github.com/larisala/tcc-ead-2020.git
```

Em seguida, garanta que a linguagem do seu livro esteja em Português. Basta descomentar no arquivo `conf.py` a linha `"language = None"` e substituir `'None'` por `'pt-br'`. Para aplicar a tradução, compile o livro novamente com o comando `runestone build` (ou `runestone build --all deploy` caso esteja no servidor). É possível que a versão do seu servidor não seja mais compatível com essa versão dos Componentes. Nesse caso, será necessário instalar o comando `runestone` das fontes oficiais.

Para traduzir os menus do livro é necessário mudar seu template e alterar a localização do template em `conf.py` alterando o `html_theme_path` ou `templates_path`.

A tradução dos componentes feita nesse projeto foi aceita como contribuição do Runestone e encontra-se na última versão do *Runestone Components* (5.4). No entanto, houveram alterações na maneira como a internacionalização é implementada, por isso a tradução não está completa nas novas versões.

## A.3 Instalação do plugin do Moodle

Para aplicar a extensão da Lição Runestone no Moodle, primeiro é necessário uma instalação do Moodle. A extensão foi desenvolvida e testada apenas nas versões 3.8 e 3.9 da plataforma, por isso o funcionamento de todas as funcionalidades em versões futuras não é garantido.

Caso você não possua uma instalação do Moodle, siga os passos no [site oficial](#).

Os arquivos do *plugin* podem ser encontrados no repositório [https://github.com/larisala/mdl\\_runestone](https://github.com/larisala/mdl_runestone). Para instalar, baixe os arquivos e coloque a pasta `runestone` no diretório `moodle/mod/`. A próxima vez que o Moodle for aberto no browser o *plugin* será instalado e ele fará a requisição dos dados de configuração:

- **URL Runestone:** o domínio no qual o servidor Runestone está instalado, por exemplo: `https://runestone.academy/` ou `https://runestone.academy/runestone`. É importante que a aplicação se encontre em `/runestone/`, porque o *plugin* cria a URL das lições dinamicamente a partir desse domínio.
- **Driver do Banco de Dados:** O tipo de Banco de Dados do servidor Runestone. Normalmente o Runestone utiliza Postgres.
- **Endereço IP do Servidor:** O endereço para conexão com a máquina que possui o banco de dados Runestone. Se este estiver no mesmo local que a instalação Moodle, basta manter como `localhost`.
- **Usuário e Senha do Banco de Dados:** As credenciais de um usuário com permissões de acesso ao banco de dados Runestone.
- **Nome do Banco de Dados:** O nome dado ao banco de dados, por padrão é configurado como `runestone`.
- **Codificação:** Mais conhecido como o tipo de *character encoding*, ou codificação de caracteres do banco. O padrão é configurado como `utf-8`.

Para que a conexão com o Banco de Dados do Runestone funcione, também é imprescindível que a extensão do respectivo *driver* esteja instalado para PHP. No caso do Postgres, rode o comando:

```
$ sudo apt install php-pgsql
```

Em seguida, ative a extensão no arquivo `php.ini`, normalmente localizado no diretório `/etc/php/<versão>/apache2/`. Basta descomentar a linha

```
; extension=pgsql
```

E reiniciar o servidor Apache com o comando

```
$ sudo service apache2 restart
```

O Moodle deve então passar a reconhecer e se conectar com o banco de dados do Runestone.

## A.4 Cadastro de usuários do Moodle no Runestone

Como explorado na seção 4.2.7, um dos maiores desafios do projeto foi lidar com a sincronização e autenticação automáticas do Moodle para o Runestone, não sendo possível desenvolver uma solução ideal na primeira versão. Por isso, o administrador dos sistemas precisará cadastrar os estudantes de forma manual.

O Moodle permite exportação dos dados de estudantes em um arquivo `.csv`. Para fazer isso pela interface do sistema, basta acessar a página de Ações em lote sobre usuários na seção de Administração do Site (Figura A.1).



Figura A.1: Exportação de usuários na interface Moodle.

É possível adicionar diversos filtros e exportar todos os usuários correspondentes, sendo possível, por exemplo, separar todos os estudantes de um determinado curso. Apesar da plataforma também disponibilizar a exportação de alguns dados dos estudantes através da página de Notas do curso, é desejável que ela seja feita pelo primeiro método, uma vez que o segundo não possui a informação de nome de usuário.

Com os dados dos estudantes num arquivo `.csv`, é necessário transportá-los para outro arquivo chamado `students.csv` com as colunas `username`, `email`, `first_name`, `last_name`, `password`, `course`. Para facilitar esse processo, o *script* desenvolvido pode ser encontrado no mesmo repositório dos arquivos do plugin ([https://github.com/larisala/mdl\\_runestone](https://github.com/larisala/mdl_runestone)). Basta baixar o arquivo `generate_students.py` e rodar o comando

```
$ python3 generate_students.py <arquivo.csv> <nome do curso>
```

O script gera senhas aleatórias de 8 caracteres para todos os usuários e salva o arquivo `students.csv` no mesmo diretório, que pode ser cadastrado na base de dados Runestone pela página "Admin/Cadastrar Estudantes" de um curso, ou rodando o comando `rsmanage inituser` de administrador no terminal. Também é possível tornar um usuário existente em instrutor ou editor de um curso através dos comandos `rsmanage addinstructor` e `rsmanage addeditor`, respectivamente. É possível consultar todos os comandos disponíveis em <https://runestoneserver.readthedocs.io/en/latest/rsmanage/toctree.html>.

Com os usuários criados, é preciso passar manualmente a senha gerada para cada estudante. Isso pode ser feito, por exemplo, através de alguma automatização de serviço de e-mail.

## A.5 Configuração da Panda

A Panda modificada pode ser encontrada no repositório <https://github.com/kokosha/panda>.

Após a importação do código da Panda, o sistema de autenticação já estará pronto para usar, mas ainda é necessário configurar a autenticação e o redirecionamento para os livros dependendo da situação do login.

Para a autenticação entre a Panda e o Runestone será necessário usar a mesma chave de criptografia que está localizado em `private\auth.key`. Para a sincronização entre as sessões da Panda e do Runestone, as duas aplicações precisam usar o mesmo banco de dados. Isso pode ser modificado em `models\db.py`, mudando a variável `table_migrate_prefix` para `"runestone_"`, onde `runestone` é o nome do banco de dados utilizado.

Para o redirecionamento para o livro ou o curso correto, é preciso verificar se o usuário está autenticado ou não. Quando o usuário está autenticado, ele é redirecionado para `/default/coursechooser/thinkcspy_ptbr`, que é a URL para a página de seleção do curso `thinkcspy_ptbr` no Runestone. Quando o usuário não estiver autenticado, ele é redirecionado para `books/published/thinkcspy_ptbr/index.html`, que é a localização do livro aberto `thinkcspy_ptbr`.

É possível utilizar o seguinte template de código para fazer esse redirecionamento dentro de qualquer arquivo `.html` no *view* do web2py.

```
1  {{if auth.is_logged_in():}}
2  <a href="http://site/default/coursechooser/thinkcspy_ptbr"></a>.
3  {{else:}}
4  <a href="http://site/books/published/thinkcspy_ptbr/index.html"></a>.
5  {{pass}}
```

## Referências

- [ARRUDA 2012] Ludmila Varela ARRUDA. “Desenvolvimento ágil de software: uma análise sintética a partir da metodologia kanban”. Em: *VII CONNEPI - Congresso Norte Nordeste de Pesquisa e Inovação*. Palmas Tocantins Brasil, 2012. ISBN: 978-85-62830-10-5 (citado na pg. 7).
- [GALUSHA 1998] Jill M GALUSHA. “Barriers to learning in distance education.” Em: (1998) (citado na pg. 1).
- [HILL 2017] Phil HILL. *Academic LMS Market Share: A view across four global regions*. 2017. URL: <https://eliterate.us/academic-lms-market-share-view-across-four-global-regions/> (citado na pg. 3).
- [*Learning Tools Interoperability* 2016] *Learning Tools Interoperability*. 2016. URL: <https://www.imslobal.org/activity/learning-tools-interoperability> (citado na pg. 26).
- [MILLER e RANUM 2012] Bradley N. MILLER e David L. RANUM. “Beyond pdf and epub: toward an interactive textbook”. Em: *Proceedings of the 17th ACM Annual Conference on Innovation and Technology in Computer Science Education*. ITiCSE '12. Haifa, Israel: ACM, 2012, pgs. 150–155. ISBN: 978-1-4503-1246-2. DOI: [10.1145/2325296.2325335](https://doi.org/10.1145/2325296.2325335). URL: <http://doi.acm.org/10.1145/2325296.2325335> (citado nas pgs. 3, 5).
- [QIAN e LEHMAN 2017] Yizhou QIAN e James LEHMAN. “Students’ misconceptions and other difficulties in introductory programming: a literature review”. Em: *ACM Transactions on Computing Education (TOCE)* 18.1 (2017), pgs. 1–24 (citado na pg. 3).