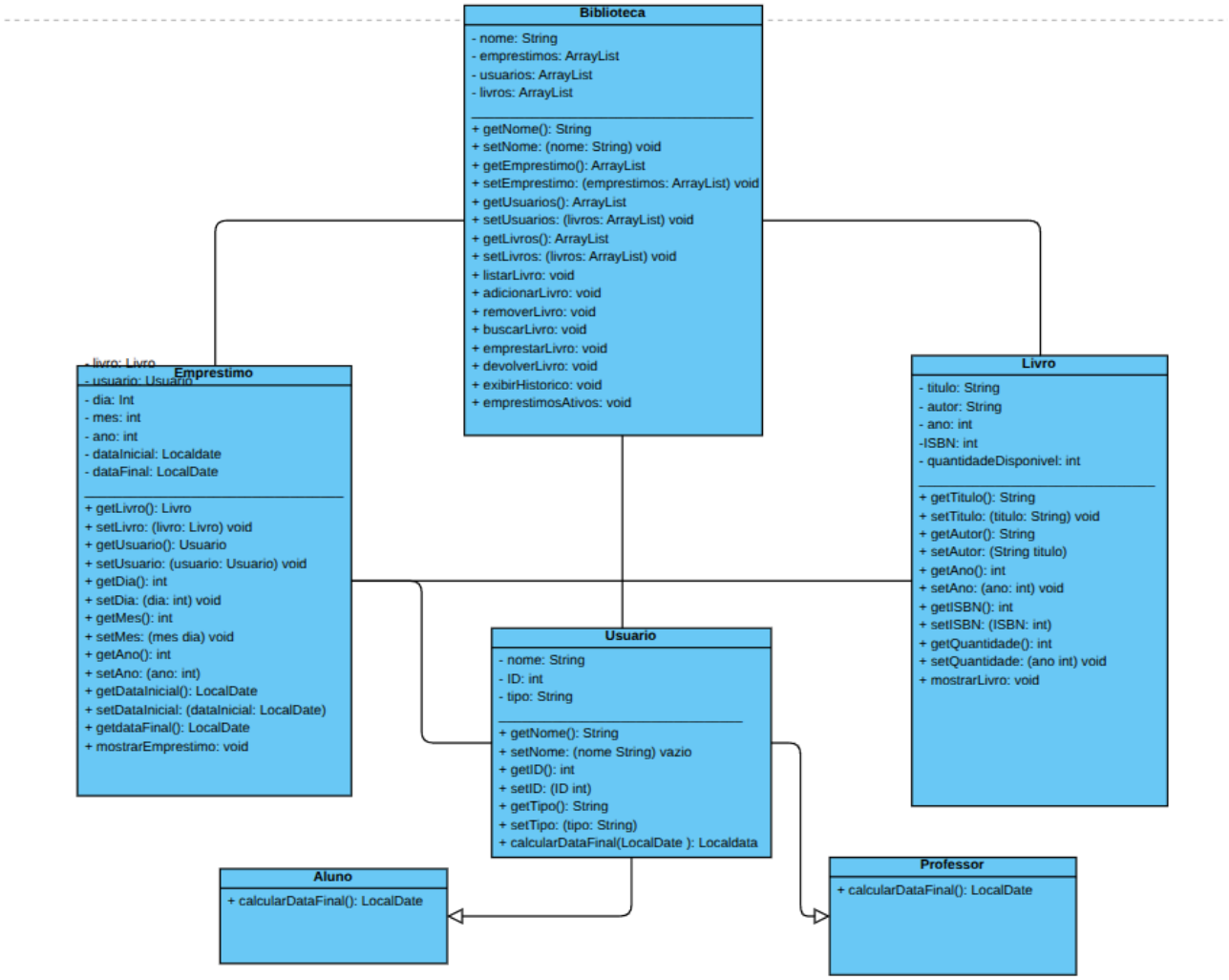
1. DESCRIÇÃO DO SISTEMA E SEUS OBJETIVOS:

O sistema desenvolvido neste TDE se trata de um sistema para o controle e gerenciamento de uma biblioteca. Seus objetivos são: registrar as entradas e saídas de livros na biblioteca com os métodos: adicionarLivro() e removerLivro, criação de objetos: Livro, Usuario (classe pai) que se deriva em duas sub-classes Professor e Aluno, Empréstimo e Biblioteca, exibir livros existentes na biblioteca com o método: listarLivros(), procurar se determinado livro existe na biblioteca em questão com o método: buscarLivro(), emprestar livros por um período determinado com o método: emprestarLivro(), devolver os livros emprestados com o método: devolverLivro(), exibir histórico de empréstimos de um usuário com o método: exibirHistorico() e visualizar empréstimos de livros que estão ativos com o método: emprestimosAtivos();

2. DIAGRAMA DE CLASSES

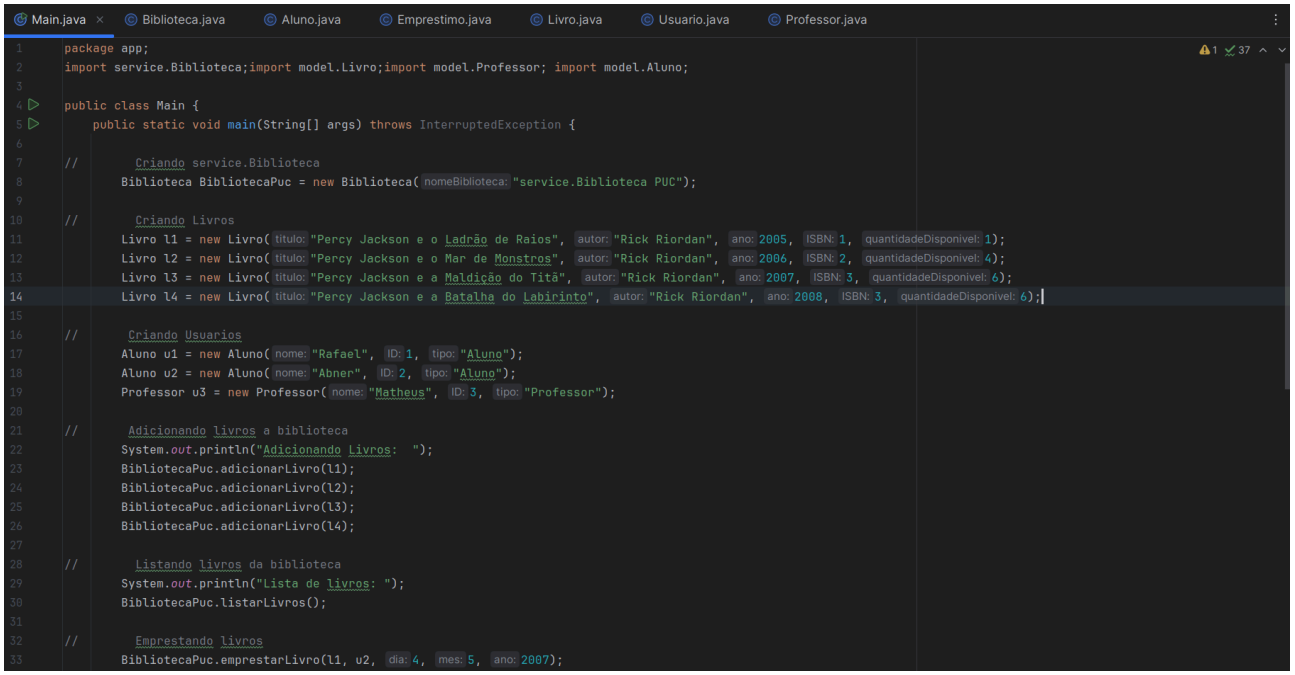


3. PRINCIPAIS DECISÕES

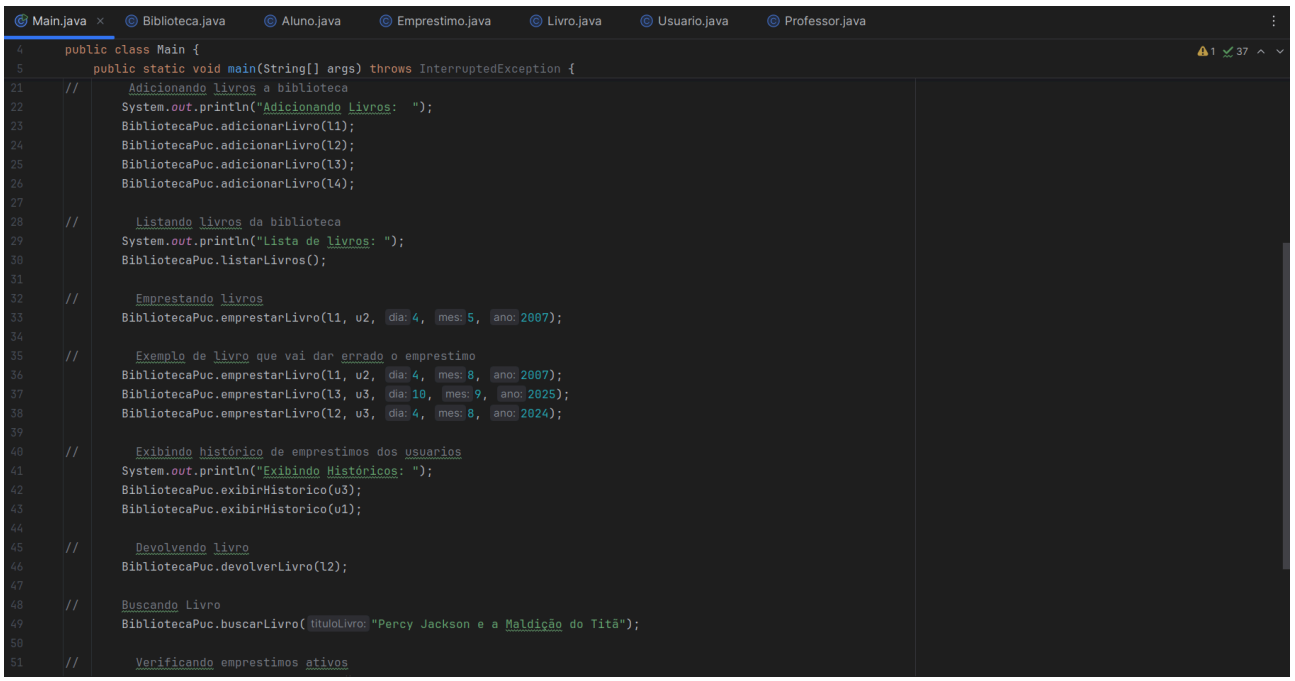
Esse projeto envolveu diversas camadas de desenvolvimento e validações, onde tarefas aparentemente simples se tornaram complexas ao exigir verificações que refletissem situações reais. A implementação de funcionalidades como controle de empréstimos, distinção entre perfis de usuários e cálculo de prazos demandou grande atenção às regras de negócio, resultando em um sistema mais fiel à dinâmica de uma biblioteca de verdade. As principais decisões foram principalmente como fazer as verificações e quais classes seriam atributos de outras, onde a primeiro momento achei que poderia fazer de determinada maneira e percebi que não daria certo, precisando assim retomar tudo de volta.

4. EVIDÊNCIAS DE EXECUÇÃO

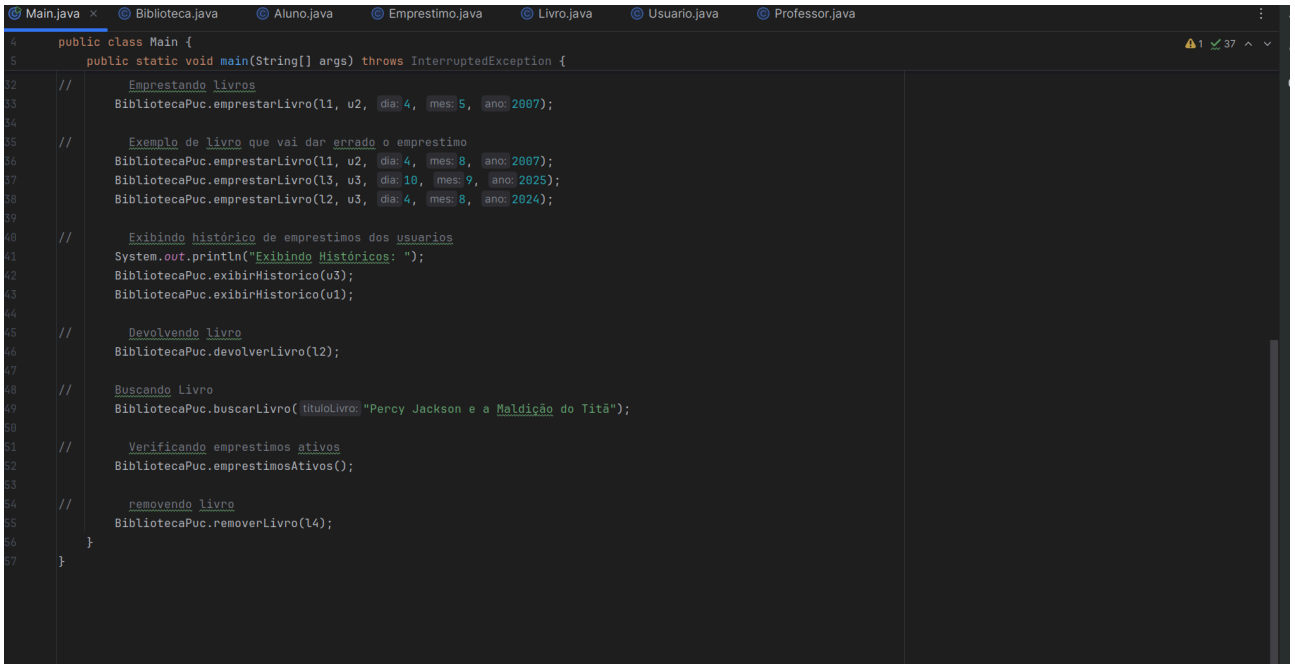
4.1 FOTOS DO CÓDIGO MAIN



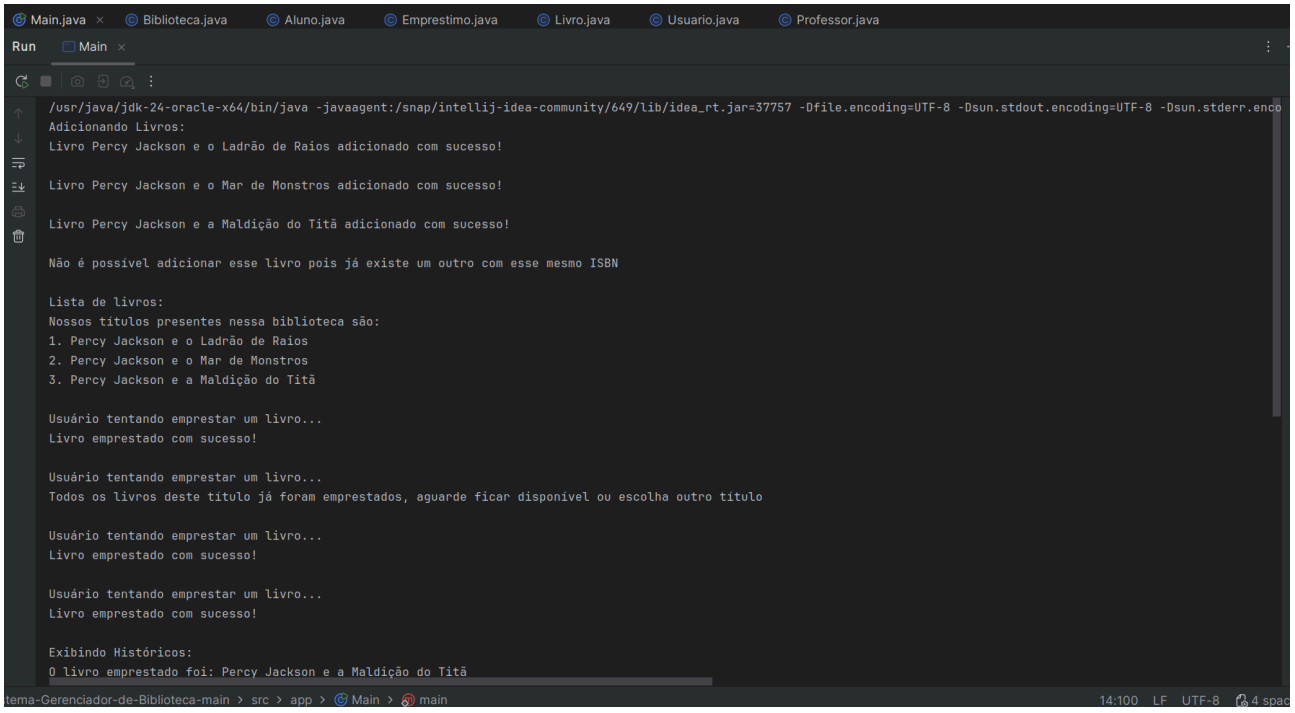
```
Main.java | Biblioteca.java | Aluno.java | Emprestimo.java | Livro.java | Usuario.java | Professor.java

package app;
import service.Biblioteca;import model.Livro;import model.Professor; import model.Aluno;

public class Main {
    public static void main(String[] args) throws InterruptedException {

//        Criando service.Biblioteca
        Biblioteca BibliotecaPuc = new Biblioteca( nomeBiblioteca: "service.Biblioteca PUC");

//        Criando Livros
        Livro l1 = new Livro( titulo: "Percy Jackson e o Ladrão de Raios",  autor: "Rick Riordan",  ano: 2005,  ISBN: 1,  quantidadeDisponivel: 1);
        Livro l2 = new Livro( titulo: "Percy Jackson e o Mar de Monstros",  autor: "Rick Riordan",  ano: 2006,  ISBN: 2,  quantidadeDisponivel: 4);
        Livro l3 = new Livro( titulo: "Percy Jackson e a Maldição do Titã",  autor: "Rick Riordan",  ano: 2007,  ISBN: 3,  quantidadeDisponivel: 6);
        Livro l4 = new Livro( titulo: "Percy Jackson e a Batalha do Labirinto",  autor: "Rick Riordan",  ano: 2008,  ISBN: 3,  quantidadeDisponivel: 6);

//       Criando Usuarios
        Aluno u1 = new Aluno( nome: "Rafael",  ID: 1,  tipo: "Aluno");
        Aluno u2 = new Aluno( nome: "Abner",  ID: 2,  tipo: "Aluno");
        Professor u3 = new Professor( nome: "Matheus",  ID: 3,  tipo: "Professor");

//       Adicionando livros a biblioteca
        System.out.println("Adicionando Livros:  ");
        BibliotecaPuc.adicionarLivro(l1);
        BibliotecaPuc.adicionarLivro(l2);
        BibliotecaPuc.adicionarLivro(l3);
        BibliotecaPuc.adicionarLivro(l4);

//        Listando livros da biblioteca
        System.out.println("Lista de livros: ");
        BibliotecaPuc.listarLivros();

//        Emprestando livros
        BibliotecaPuc.emprestarLivro(l1, u2,  dia: 4,  mes: 5,  ano: 2007);
```

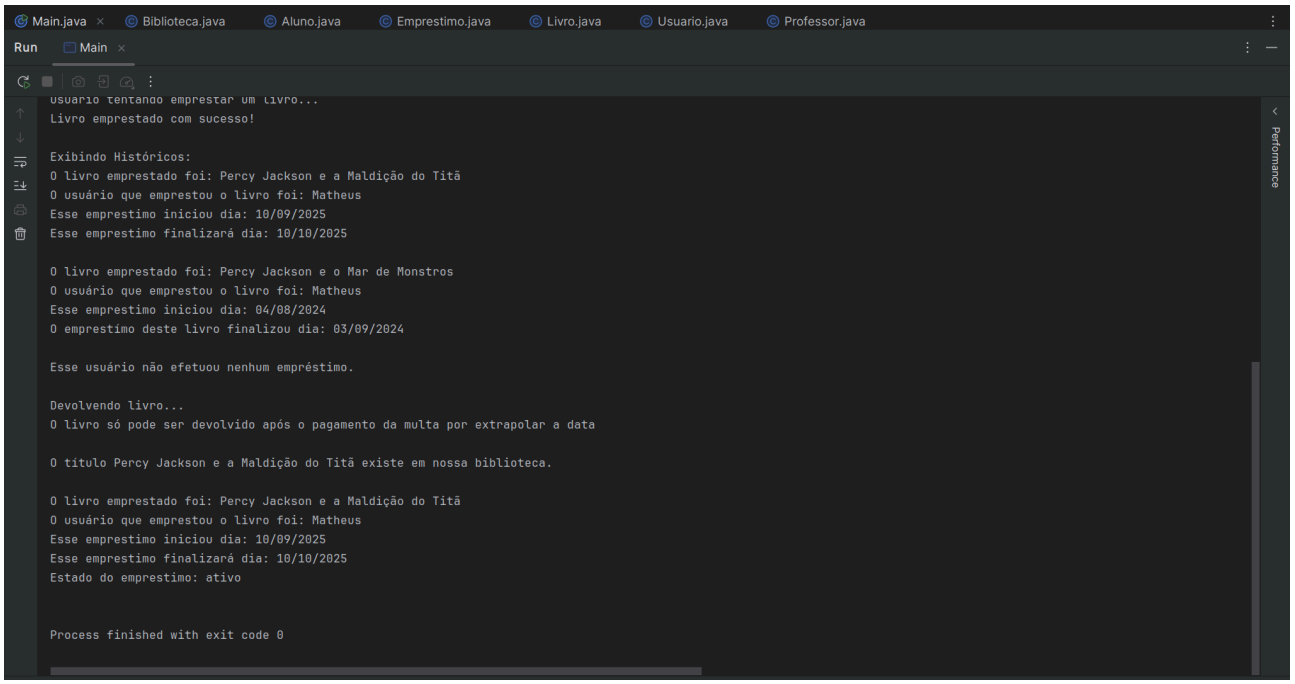


```
Main.java | Biblioteca.java | Aluno.java | Emprestimo.java | Livro.java | Usuario.java | Professor.java

public class Main {
    public static void main(String[] args) throws InterruptedException {
//       Adicionando livros a biblioteca
        System.out.println("Adicionando Livros:  ");
        BibliotecaPuc.adicionarLivro(l1);
        BibliotecaPuc.adicionarLivro(l2);
        BibliotecaPuc.adicionarLivro(l3);
        BibliotecaPuc.adicionarLivro(l4);

//        Listando livros da biblioteca
        System.out.println("Lista de livros: ");
        BibliotecaPuc.listarLivros();

//        Emprestando livros
        BibliotecaPuc.emprestarLivro(l1, u2,  dia: 4,  mes: 5,  ano: 2007);

//        Exemplo de livro que vai dar errado o emprestimo
        BibliotecaPuc.emprestarLivro(l1, u2,  dia: 4,  mes: 8,  ano: 2007);
        BibliotecaPuc.emprestarLivro(l3, u3,  dia: 10,  mes: 9,  ano: 2025);
        BibliotecaPuc.emprestarLivro(l2, u3,  dia: 4,  mes: 8,  ano: 2024);

//        Exibindo histórico de emprestimos dos usuarios
        System.out.println("Exibindo Históricos: ");
        BibliotecaPuc.exibirHistorico(u3);
        BibliotecaPuc.exibirHistorico(u1);

//        Devolvendo livro
        BibliotecaPuc.devolverLivro(l2);

//      Buscando Livro
        BibliotecaPuc.buscarLivro( tituloLivro: "Percy Jackson e a Maldição do Titã");

//        Verificando emprestimos ativos
```

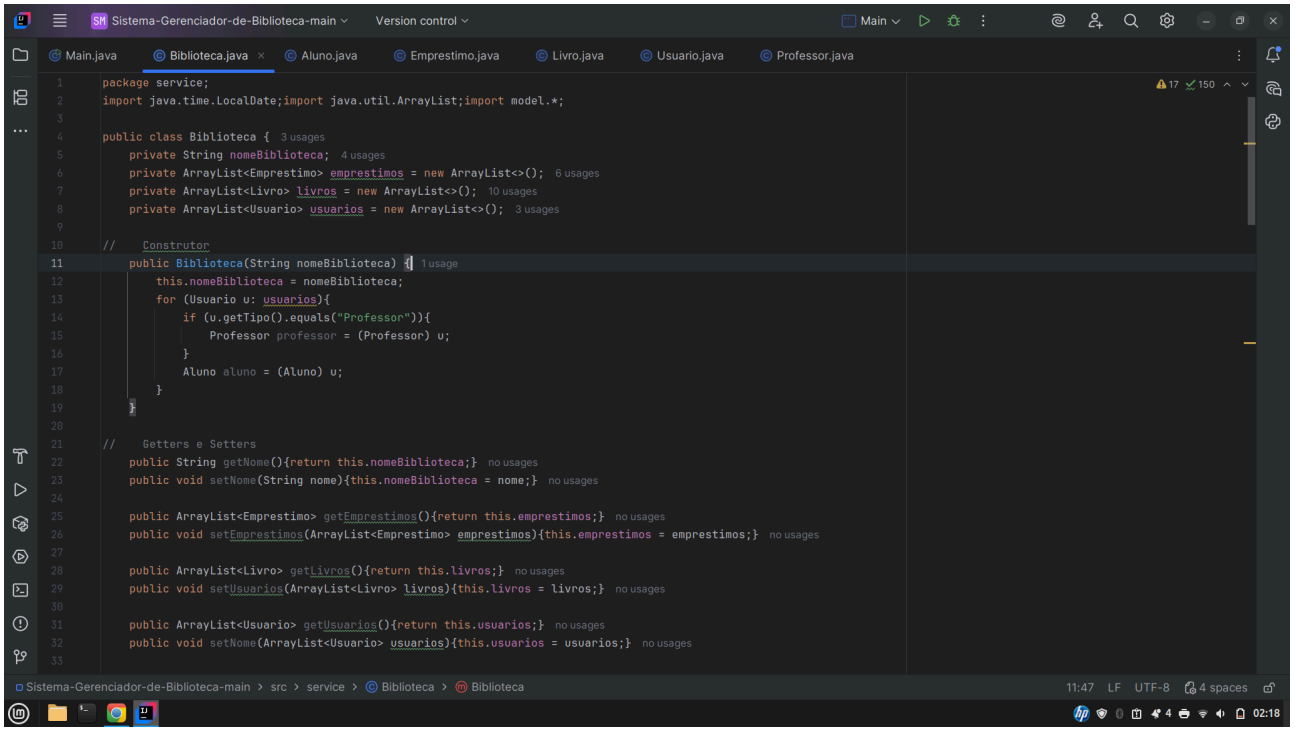


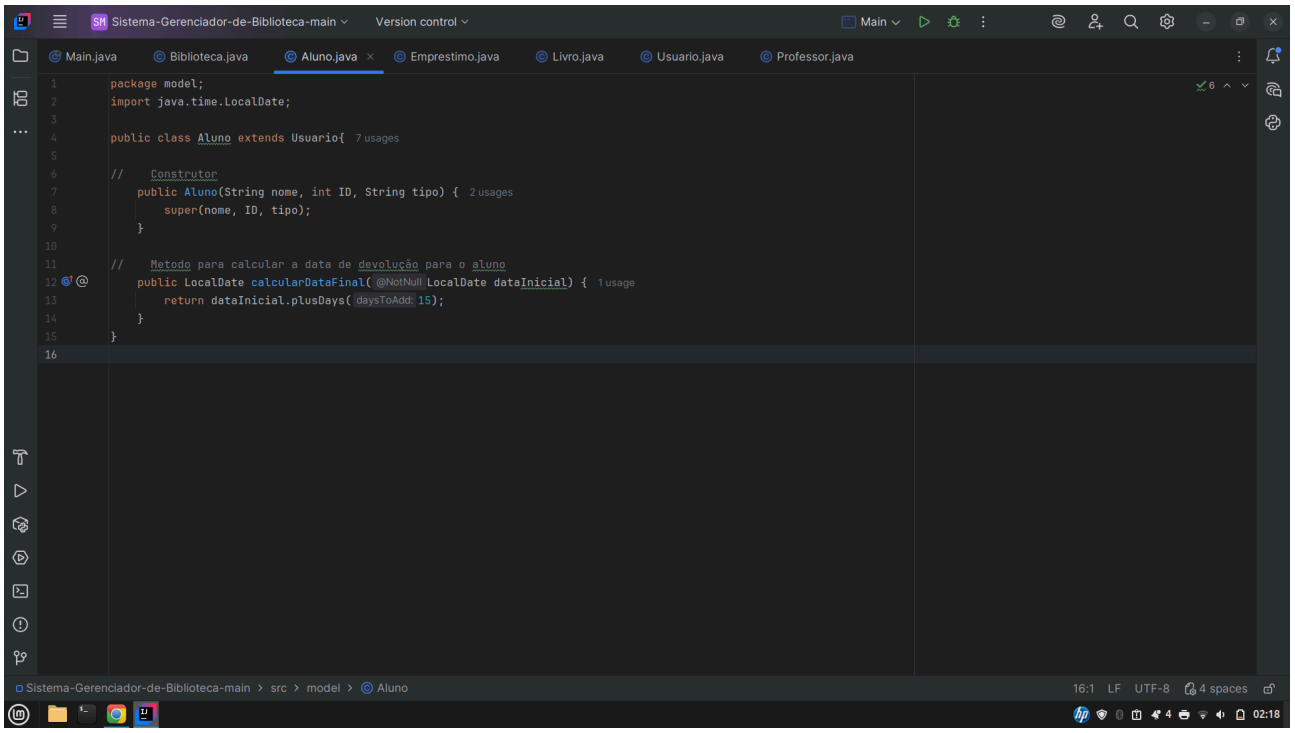
4.2 FOTOS DA EXECUÇÃO NO TERMINAL

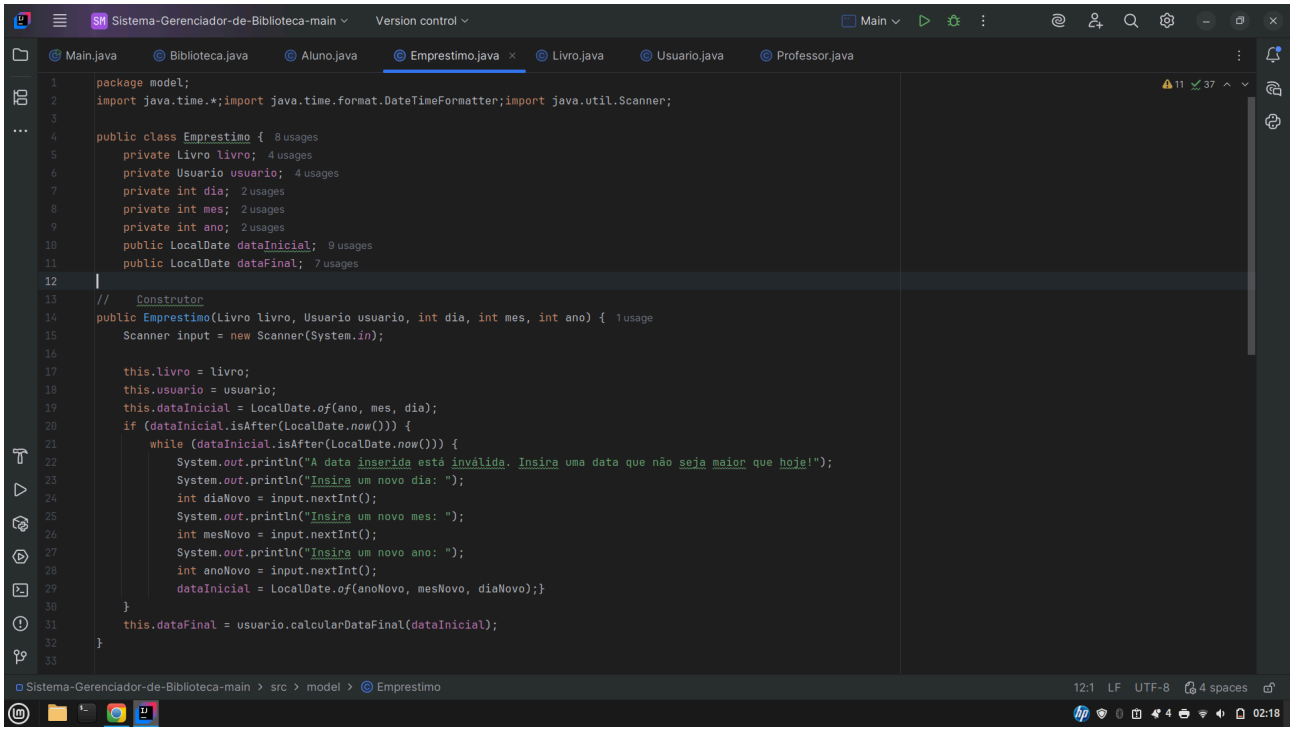


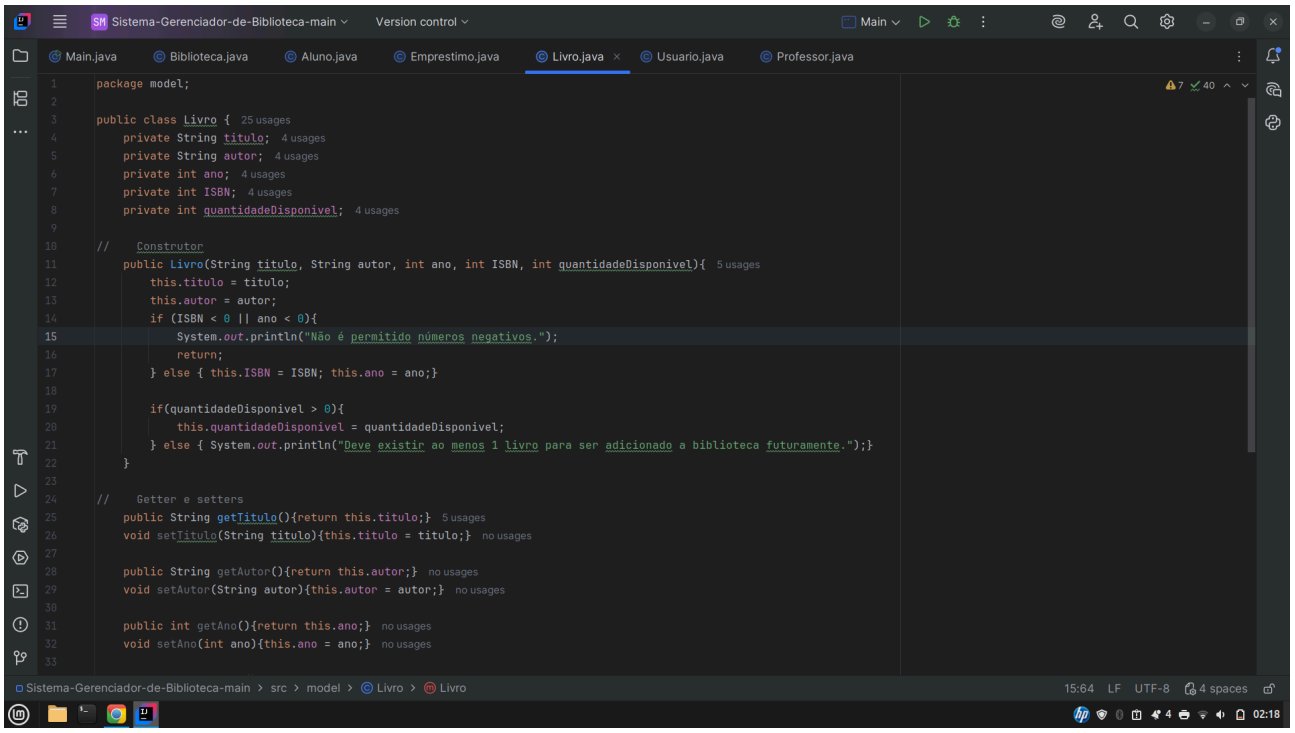


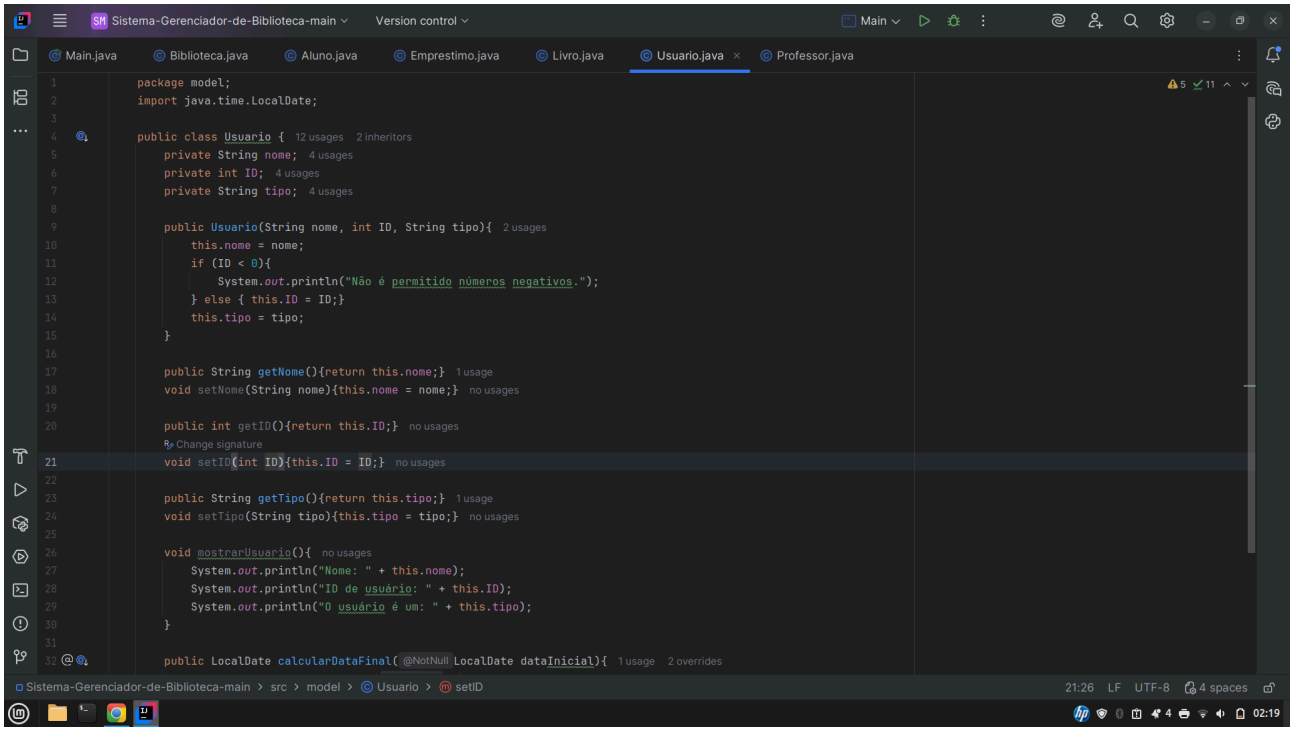
4.3 FOTOS DOS CÓDIGOS DAS CLASSES

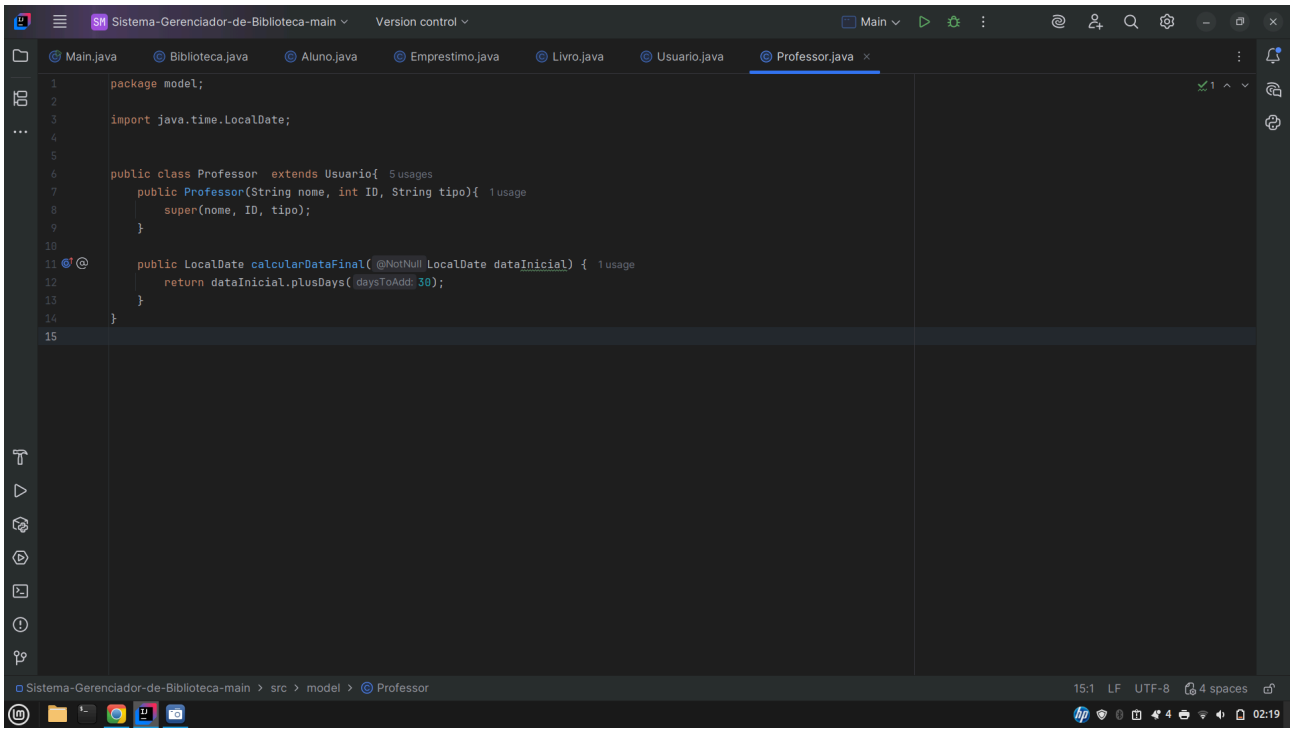



Sistema-Gerenciador-de-Biblioteca-main
Version control
Main
Main.java
Biblioteca.java
Aluno.java
Emprestimo.java
Livro.java
Usuario.java
Professor.java
package model;
import java.time.LocalDate;
public class Aluno extends Usuario{ 7 usages
// Construtor
public Aluno(String nome, int ID, String tipo) { 2 usages
super(nome, ID, tipo);
}
// Metodo para calcular a data de devolução para o aluno
public LocalDate calcularDataFinal(@NotNull LocalDate dataInicial) { 1 usage
return dataInicial.plusDays(daysToAdd: 15);
}
}
Sistema-Gerenciador-de-Biblioteca-main › src › model › Aluno
16:1 LF UTF-8 4 spaces
02:18


Sistema-Gerenciador-de-Biblioteca-main
Version control
Main
Main.java
Biblioteca.java
Aluno.java
Emprestimo.java
Livro.java
Usuario.java
Professor.java
package model;
import java.time.*;import java.time.format.DateTimeFormatter;import java.util.Scanner;
public class Emprestimo { 8 usages
private Livro livro; 4 usages
private Usuario usuario; 4 usages
private int dia; 2 usages
private int mes; 2 usages
private int ano; 2 usages
public LocalDate dataInicial; 9 usages
public LocalDate dataFinal; 7 usages
// Construtor
public Emprestimo(Livro livro, Usuario usuario, int dia, int mes, int ano) { 1 usage
Scanner input = new Scanner(System.in);
this.livro = livro;
this.usuario = usuario;
this.dataInicial = LocalDate.of(ano, mes, dia);
if (dataInicial.isAfter(LocalDate.now())) {
while (dataInicial.isAfter(LocalDate.now())) {
System.out.println("A data inserida está inválida. Insira uma data que não seja maior que hoje!");
System.out.println("Insira um novo dia: ");
int diaNovo = input.nextInt();
System.out.println("Insira um novo mes: ");
int mesNovo = input.nextInt();
System.out.println("Insira um novo ano: ");
int anoNovo = input.nextInt();
dataInicial = LocalDate.of(anoNovo, mesNovo, diaNovo);}
}
this.dataFinal = usuario.calcularDataFinal(dataInicial);
}
Sistema-Gerenciador-de-Biblioteca-main › src › model › Emprestimo
12:1 LF UTF-8 4 spaces
02:18


Sistema-Gerenciador-de-Biblioteca-main
Version control
Main
Main.java
Biblioteca.java
Aluno.java
Emprestimo.java
Livro.java
Usuario.java
Professor.java
package model;
public class Livro { 25 usages
private String titulo; 4 usages
private String autor; 4 usages
private int ano; 4 usages
private int ISBN; 4 usages
private int quantidadeDisponivel; 4 usages
// Construtor
public Livro(String titulo, String autor, int ano, int ISBN, int quantidadeDisponivel){ 5 usages
this.titulo = titulo;
this.autor = autor;
if (ISBN < 0 || ano < 0){
System.out.println("Não é permitido números negativos.");
return;
} else { this.ISBN = ISBN; this.ano = ano;}
if(quantidadeDisponivel > 0){
this.quantidadeDisponivel = quantidadeDisponivel;
} else { System.out.println("Deve existir ao menos 1 livro para ser adicionado a biblioteca futuramente.");}
}
// Getter e setters
public String getTitulo(){return this.titulo;} 5 usages
void setTitulo(String titulo){this.titulo = titulo;} no usages
public String getAutor(){return this.autor;} no usages
void setAutor(String autor){this.autor = autor;} no usages
public int getAno(){return this.ano;} no usages
void setAno(int ano){this.ano = ano;} no usages
Sistema-Gerenciador-de-Biblioteca-main › src › model › Livro › Livro
15:64 LF UTF-8 4 spaces
02:18


Sistema-Gerenciador-de-Biblioteca-main
Version control
Main
Main.java
Biblioteca.java
Aluno.java
Emprestimo.java
Livro.java
Usuario.java
Professor.java
package model;
import java.time.LocalDate;
public class Usuario { 12 usages 2 inheritors
private String nome; 4 usages
private int ID; 4 usages
private String tipo; 4 usages
public Usuario(String nome, int ID, String tipo){ 2 usages
this.nome = nome;
if (ID < 0){
System.out.println("Não é permitido números negativos.");
} else { this.ID = ID;}
this.tipo = tipo;
}
public String getNome(){return this.nome;} 1 usage
void setNome(String nome){this.nome = nome;} no usages
public int getID(){return this.ID;} no usages
Change signature
void setID(int ID){this.ID = ID;} no usages
public String getTipo(){return this.tipo;} 1 usage
void setTipo(String tipo){this.tipo = tipo;} no usages
void mostrarUsuario(){ no usages
System.out.println("Nome: " + this.nome);
System.out.println("ID de usuário: " + this.ID);
System.out.println("O usuário é um: " + this.tipo);
}
public LocalDate calcularDataFinal(@NotNull LocalDate dataInicial){ 1 usage 2 overrides
Sistema-Gerenciador-de-Biblioteca-main › src › model › Usuario › setID
21:26 LF UTF-8 4 spaces
02:19


Sistema-Gerenciador-de-Biblioteca-main
Version control
Main
Main.java
Biblioteca.java
Aluno.java
Emprestimo.java
Livro.java
Usuario.java
Professor.java
package model;
import java.time.LocalDate;
public class Professor extends Usuario{ 5 usages
public Professor(String nome, int ID, String tipo){ 1 usage
super(nome, ID, tipo);
}
public LocalDate calcularDataFinal(@NotNull LocalDate dataInicial) { 1 usage
return dataInicial.plusDays(daysToAdd: 30);
}
}
Sistema-Gerenciador-de-Biblioteca-main › src › model › Professor
15:1 LF UTF-8 4 spaces
02:19