

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SÃO PAULO - Campus Bragança Paulista**  
**Curso: Análise e Desenvolvimento de Sistemas - ADS**  
**Disciplina: Desenvolvimento de Componentes (BRADECO)**  
**Prof. Prof. Luiz Gustavo D. de O. Véras**

**Nome: Enzo José R. J. T. de Oliveira**      **Prontuário: BP3044271**  
**Nome: Lincoln Silva Moraes**      **Prontuário: BP3044351**

### **Projeto Desenvolvimento de Componentes: Livraria**

#### **1. Introdução**

O presente relatório descreve o desenvolvimento de um sistema de E-commerce de Livraria, cujo objetivo é permitir que clientes pesquisem, visualizem e adquiram livros de forma online. O projeto foi desenvolvido utilizando a abordagem de componentes em Java, onde cada componente é empacotado em um arquivo .jar, representando funcionalidades independentes e coesas.

Este trabalho é uma continuidade do projeto de Arquitetura de Software (BRAARQS), adaptado para a implementação baseada em componentes, com foco na modularização, baixa complexidade, reutilização de código, e aplicação dos princípios SOLID.

#### **2. Objetivo do Trabalho**

O objetivo principal é demonstrar a capacidade de desenvolver um sistema modularizado, onde:

- Cada integrante é responsável por um componente individual (CSU01–CSU05).
- O sistema como um todo funciona de maneira integrada, permitindo que todos os casos de uso sejam executados corretamente.
- Os componentes seguem boas práticas de desenvolvimento, com coesão e baixo acoplamento, e aplicam os princípios SOLID.
- A arquitetura permite manutenção futura, escalabilidade e fácil integração com novos componentes.

#### **3. Casos de Uso**

CSU01 – Pesquisar Livro

- Objetivo: Permitir ao cliente pesquisar livros por título, autor, editora ou categoria.
- Fluxo principal:
  - Cliente escolhe critério de pesquisa e insere parâmetro.
  - Sistema exibe listagem com título, autor, editora, categoria, status e preço.
  - Cliente pode visualizar detalhes do livro e adicionar ao carrinho.
- Extensões:
  - Exibir Detalhes
  - Adicionar ao Carrinho

#### CSU02 – Efetuar Pedido

- Objetivo: Confirmar e finalizar pedidos de livros adicionados ao carrinho.
- Fluxo principal:
  - Cliente acessa carrinho com itens selecionados.
  - Se não estiver cadastrado, realiza cadastro.
  - Informa endereço de entrega e escolhe forma de pagamento (Cartão ou Pix).
  - Sistema atualiza estoque e finaliza o pedido.
- Inclusões:
  - Visualizar Carrinho
  - Calcular Frete
  - Efetuar Pagamento
- Especializações:
  - Pagar via Cartão
  - Pagar via Pix

#### CSU03 – Manter Cliente

- Objetivo: Registrar ou atualizar dados pessoais do cliente.
- Fluxo principal:
  1. Cliente fornece nome, CPF, e-mail, telefone, data de nascimento e endereços.
  2. Sistema valida e armazena os dados.
- Extensão: CSU03 é extensão do CSU02 (Efetuar Pedido) quando cliente não cadastrado ou dados alterados.

#### CSU04 – Visualizar Pedido

- Objetivo: Permitir ao cliente acompanhar pedidos recentes e consultar histórico de compras.
- Requisito: Cliente deve estar autenticado.

#### CSU05 – Manter Livro

- Objetivo: Registrar ou atualizar informações de livros, autores e editoras.
- Dados do Livro: Título, ISBN, número de páginas, ano, categorias, formato, status, preço, resumo, quantidade em estoque.
- Dados do Autor: Nome, data de nascimento, nacionalidade.
- Dados da Editora: Nome, CNPJ, telefone e e-mail.
- Extensões: Manter Autor e Manter Editora.

### 4. Estrutura de Componentes

O sistema foi modularizado em componentes independentes, cada um empacotado em .jar. A divisão considera baixa dependência entre componentes e alta coesão.

**Livro.jar (desenvolvido por Lincoln):**

Este componente é responsável por armazenar todas as informações de livros no sistema. Ele permite que o cliente busque livros por título, autor, editora ou categoria, visualize detalhes e adicione itens ao carrinho.

**Cliente.jar (desenvolvido por Lincoln):**

Responsável pelo gerenciamento dos dados pessoais dos clientes, este componente registra, atualiza e valida informações como nome, CPF, e-mail, telefone e endereços.

Carrinho.jar (desenvolvido por Lincoln):

O componente Carrinho.jar processa toda a lógica de adição de um livro para o componente de pedido. Ele controla o fluxo de visualização do carrinho até a atualização de estoque.

Frete.jar (desenvolvido por Enzo):

Este componente permite o cálculo do valor a ser acrescido ao componente de pagamento

Pedido.jar (desenvolvido por Lincoln):

Responsável por registrar os itens adicionados ao carrinho e o frete a ser passado ao componente de pagamento.

Pagamento.jar (desenvolvido por Enzo):

Este componente salva os dados do cliente recorrentes a pagamento e envia ao banco de dados, além de levar a api externa para realização do pagamento do livro.

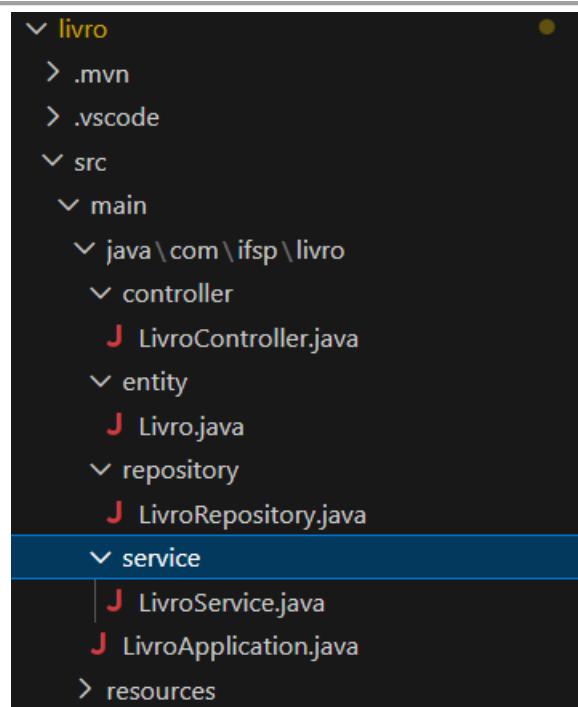
Gateway.jar (desenvolvido por Enzo):

É responsável por unir o back-end com o front-end, toda os dados passam por ele.

## 5. Implementação do SOLID

Os integrantes desenvolveram os componentes seguindo princípios SOLID.

### 5.1. SRP – Single Responsibility Principle:



segue SRP porque separa responsabilidades em **controller, service, repository e entity**.

## 5.2. OCP — Open/Closed Principle:

```
@RestController  
@RequestMapping("/livros")  
@CrossOrigin(origins = {"http://localhost:5173", "http://localhost:3000"})  
public class LivroController {
```

crescer sem mexer nos outros **controllers**.

## 5.3. LSP — Liskov Substitution Principle:

- ↳ Pagamento.java
- ↳ PagamentoCartao.java
- ↳ PagamentoPix.java

A arquitetura segue o LSP porque foi implementada a herança para diversos tipos de pagamento.

## 5.4. ISP — Interface Segregation Principle:

```
package com.ifsp.cliente.repository;  
  
import java.util.Optional;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
import com.ifsp.cliente.entity.Cliente;  
  
public interface ClienteRepository extends JpaRepository<Cliente, String> {  
  
    Optional<Cliente> findByEmail(String email);  
    Optional<Cliente> findByCpf(String cpf);  
}
```

A arquitetura segue o ISP porque foi implementada pois cada entidade tem seu próprio repositório.

## 5.4. DIP — Dependency Inversion Principle:

```
@Service
public class PedidoService {

    private final PedidoRepository repository;
    private final CarrinhoService carrinhoService;
    private final LivroService livroService;

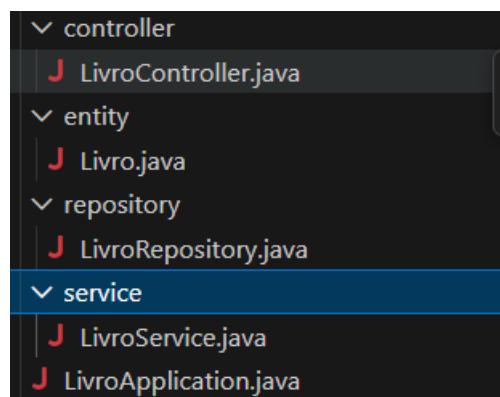
    public PedidoService(PedidoRepository repository,
                         CarrinhoService carrinhoService,
                         LivroService livroService) {
        this.repository = repository;
        this.carrinhoService = carrinhoService;
        this.livroService = livroService;
    }
}
```

A arquitetura segue o DIP porque ele depende da abstração não da implementação concreta.

## 6. Princípios de Desenvolvimento de Componentes

### 6.1. REP — Reuse/Release Equivalence Principle:

---



O princípio REP foi aplicado, pois se for atualizado “LivroController”, todo o pacote muda junto e é reutilizado junto.

### 6.2. CRP — Common Reuse Principle:

```
package com.ifsp.cliente.controller;

import java.util.List;
import java.util.Map;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.http.HttpStatus;
import org.springframework.web.bind.annotation.CrossOrigin;
import org.springframework.web.bind.annotation.DeleteMapping;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.PutMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.ResponseStatus;
import org.springframework.web.bind.annotation.RestController;

import com.ifsp.cliente.entity.Cliente;
import com.ifsp.cliente.service.ClienteService;

@RestController
@RequestMapping("/clientes")
@CrossOrigin(origins = {"http://localhost:5173", "http://localhost:3000"})
public class ClienteController {
```

Isso demonstra a implementação CRP, pois não existe dependência cruzada desnecessária.

### 6.3. CCP — Common Closure Principle:

```
package com.ifsp.livro.service;

import java.util.List;

import org.springframework.stereotype.Service;

import com.ifsp.livro.entity.Livro;
import com.ifsp.livro.repository.LivroRepository;

@Service
public class LivroService {

    private final LivroRepository repository;

    public LivroService(LivroRepository repository) {
        this.repository = repository;
    }
}
```

O CCP é aplicado aqui, pois ele agrupa classes que mudam pelo mesmo motivo.

#### 6.4. ADP — Acyclic Dependencies Principle:

```
package com.ifsp.carrinho.service;

import com.ifsp.carrinho.entity.Carrinho;
import com.ifsp.carrinho.entity.ItemCarrinho;
import com.ifsp.carrinho.repository.CarrinhoRepository;
import com.ifsp.cliente.entity.Cliente;
import com.ifsp.cliente.service.ClienteService;
import com.ifsp.livro.entity.Livro;
import com.ifsp.livro.service.LivroService;
import org.springframework.stereotype.Service;

@Service
public class CarrinhoService {

    private final CarrinhoRepository carrinhoRepo;
    private final ClienteService clienteService;
    private final LivroService livroService;
}
```

A arquitetura segue o ACP, pois não tem ciclos de dependencia entre ecomponentes.

#### 6.5. SDP — Stable Dependencies Principle:

```
public class PedidoController {

    private final PedidoService service;
```

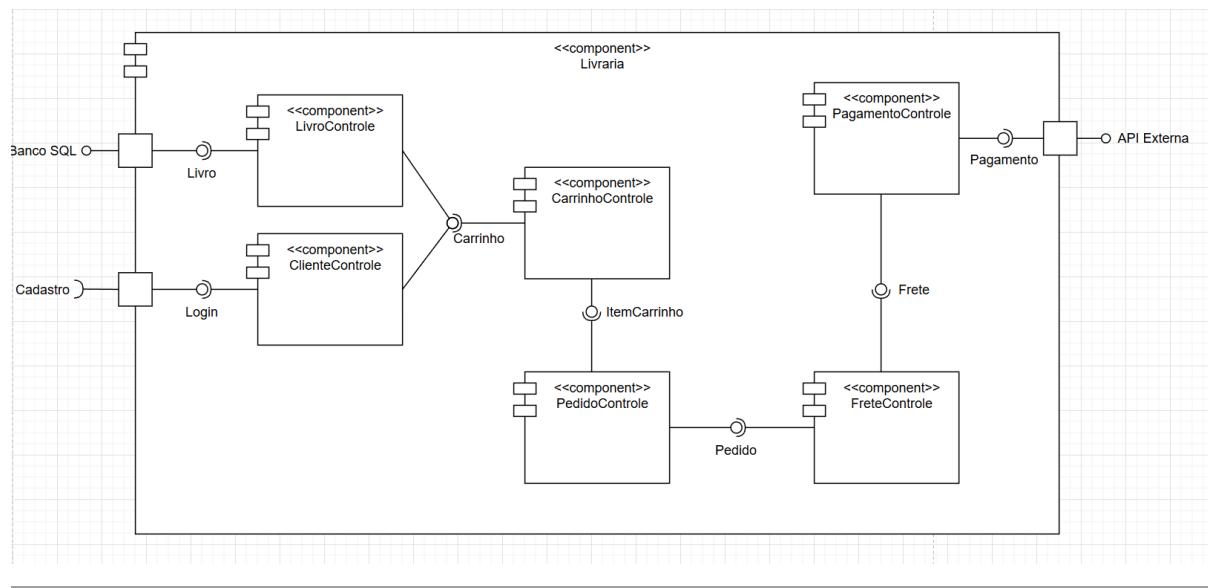
É aplicado o SDP, pois componentes como controllers que são instáveis dependem de repositórios que são componentes estáveis.

#### 6.6. SAP — Stable Abstractions Principle:

```
public interface CarrinhoRepository extends JpaRepository<Carrinho, String> {  
    Optional<Carrinho> findByCliente(Cliente cliente);  
}
```

O SAP é aplicado, pois componentes mais estáveis devem depender mais de interfaces.

### 7. Diagrama de Componentes



### 8. Testes e Funcionamento

#### Testes individuais

Livro:

```

PS C:\Users\enzor\Downloads\Livraria-Projeto\livraria\livro> mvn spring-boot:run
2025-12-02T00:14:08.116-03:00 INFO 37192 --- [livro] [           main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2025-12-02T00:14:08.228-03:00 INFO 37192 --- [livro] [           main] com.zaxxer.hikari.pool.HikariPool   : HikariPool-1 - Added connection com.mysql.cj.jdbc.ConnectionImpl@1c93b51e
2025-12-02T00:14:08.229-03:00 INFO 37192 --- [livro] [           main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2025-12-02T00:14:08.277-03:00 INFO 37192 --- [livro] [           main] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo [name: default]
2025-12-02T00:14:08.323-03:00 INFO 37192 --- [livro] [           main] org.hibernate.Version      : HHH000412: Hibernate ORM core version 6.5.2.Final
2025-12-02T00:14:08.347-03:00 INFO 37192 --- [livro] [           main] o.h.c.internal.RegionFactoryInitiator : HHH00026: Second-level cache disabled
2025-12-02T00:14:08.541-03:00 INFO 37192 --- [livro] [           main] o.s.o.j.p.SpringPersistenceUnitInfo : No LoadTimeWeaver setup: ignoring JPA class transformer
2025-12-02T00:14:08.592-03:00 WARN 37192 --- [livro] [           main] org.hibernate.orm.deprecation : HHH0000025: MySQLDialect does not need to be specified explicitly using 'hibernate.dialect' (remove the property setting and it will be selected by default)
2025-12-02T00:14:09.087-03:00 INFO 37192 --- [livro] [           main] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000489: No JTA platform available (set 'hibernate.transaction.jta.platform' to enable JTA platform integration)
2025-12-02T00:14:09.120-03:00 INFO 37192 --- [livro] [           main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2025-12-02T00:14:09.306-03:00 WARN 37192 --- [livro] [           main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be performed during view rendering. Explicitly configure spring.jpa.open-in-view to disable this warning
2025-12-02T00:14:09.489-03:00 INFO 37192 --- [livro] [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8081 (http) with context path '/'
2025-12-02T00:14:09.492-03:00 INFO 37192 --- [livro] [           main] com.ifsp.livro.LivroApplication      : Started LivroApplication in 2.427 seconds (process running for 2.635)

```

## Cliente:

```

PS C:\Users\enzor\Downloads\Livraria-Projeto\livraria\cliente> mvn spring-boot:run
transformer
2025-12-02T00:15:01.664-03:00 INFO 4164 --- [cliente] [           main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2025-12-02T00:15:01.747-03:00 INFO 4164 --- [cliente] [           main] com.zaxxer.hikari.pool.HikariPool   : HikariPool-1 - Added connection com.mysql.cj.jdbc.ConnectionImpl@2cbe455c
2025-12-02T00:15:01.778-03:00 WARN 4164 --- [cliente] [           main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2025-12-02T00:15:01.778-03:00 WARN 4164 --- [cliente] [           main] org.hibernate.orm.deprecation : HHH0000025: MySQLDialect does not need to be specified explicitly using 'hibernate.dialect' (remove the property setting and it will be selected by default)
2025-12-02T00:15:02.221-03:00 INFO 4164 --- [cliente] [           main] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000489: No JTA platform available (set 'hibernate.transaction.jta.platform' to enable JTA platform integration)
2025-12-02T00:15:02.245-03:00 DEBUG 4164 --- [cliente] [           main] org.hibernate.SQL          :
alter table clientes
    modify column nome varchar(255) not null
Hibernate:
    alter table clientes
        modify column nome varchar(255) not null
2025-12-02T00:15:02.332-03:00 INFO 4164 --- [cliente] [           main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2025-12-02T00:15:02.500-03:00 WARN 4164 --- [cliente] [           main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be performed during view rendering. Explicitly configure spring.jpa.open-in-view to disable this warning
2025-12-02T00:15:02.669-03:00 INFO 4164 --- [cliente] [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8082 (http) with context path '/'
2025-12-02T00:15:02.673-03:00 INFO 4164 --- [cliente] [           main] com.ifsp.cliente.ClienteApplication     : Started ClienteApplication in 2.14 seconds (process running for 2.307)

```

## Carrinho:

```

PS C:\Users\enzor\Downloads\Livraria-Projeto\livraria\carrinho> mvn spring-boot:run
at org.springframework.boot.SpringApplication.run(SpringApplication.java:335) ~[spring-boot-3.3.2.jar:3.3.2]
at org.springframework.boot.SpringApplication.run(SpringApplication.java:1363) ~[spring-boot-3.3.2.jar:3.3.2]
at org.springframework.boot.SpringApplication.run(SpringApplication.java:1352) ~[spring-boot-3.3.2.jar:3.3.2]
at com.ifsp.carrinho.CarrinhoApplication.main(CarrinhoApplication.java:21) ~[classes/:na]
Caused by: java.sql.SQLException: Data truncated for column 'telefone' at row 1
at com.mysql.cj.jdbc.exceptions.SQLException.createSQLException(SQLException.java:130) ~[mysql-connector-j-8.3.0.jar:8.3.0]
at com.mysql.cj.jdbc.exceptions.SQLExceptionsMapping.translateException(SQLExceptionsMapping.java:122) ~[mysql-connector-j-8.3.0.jar:8.3.0]
at com.mysql.cj.jdbc.StatementImpl.executeInternal(StatementImpl.java:770) ~[mysql-connector-j-8.3.0.jar:8.3.0]
at com.mysql.cj.jdbc.StatementImpl.execute(StatementImpl.java:653) ~[mysql-connector-j-8.3.0.jar:8.3.0]
at com.zaxxer.hikari.pool.ProxyStatement.execute(ProxyStatement.java:94) ~[HikariCP-5.1.0.jar:na]
at com.zaxxer.hikari.pool.HikariProxyStatement.execute(HikariProxyStatement.java) ~[HikariCP-5.1.0.jar:na]
at org.hibernate.tool.schema.internal.exec.GenerationTargetToDatabase.accept(GenerationTargetToDatabase.java:80) ~[hibernate-core-6.5.2.Final.jar:6.5.2.Final]
Final] ... 37 common frames omitted
2025-12-02T00:18:39.215-03:00 INFO 30276 --- [carrinho] [           main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2025-12-02T00:18:39.405-03:00 WARN 30276 --- [carrinho] [           main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be performed during view rendering. Explicitly configure spring.jpa.open-in-view to disable this warning
2025-12-02T00:18:39.567-03:00 INFO 30276 --- [carrinho] [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8083 (http) with context path '/'
2025-12-02T00:18:39.571-03:00 INFO 30276 --- [carrinho] [           main] com.ifsp.carrinho.CarrinhoApplication : Started CarrinhoApplication in 2.304 seconds (process running for 2.479)

```

## Pedido:

```
PS C:\Users\enzo\Downloads\Livraria-Projeto\livraria\pedido> mvn spring-boot:run
2025-12-02T00:16:55.656-03:00 INFO 37852 --- [pedido] [           main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 630 ms
2025-12-02T00:16:55.723-03:00 INFO 37852 --- [pedido] [           main] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo [name: default]
2025-12-02T00:16:55.748-03:00 INFO 37852 --- [pedido] [           main] org.hibernate.Version      : HHH000412: Hibernate ORM core version 6.5.2.Final
2025-12-02T00:16:55.764-03:00 INFO 37852 --- [pedido] [           main] o.h.c.internal.RegionFactoryInitiator : HHH000026: Second-level cache disabled
2025-12-02T00:16:55.906-03:00 INFO 37852 --- [pedido] [           main] o.s.o.j.p.SpringPersistenceUnitInfoTransformer : No LoadTimeWeaver setup: ignoring JPA class
2025-12-02T00:16:55.919-03:00 INFO 37852 --- [pedido] [           main] com.zaxxer.hikari.HikariDataSource   : HikariPool-1 - Starting...
2025-12-02T00:16:56.013-03:00 INFO 37852 --- [pedido] [           main] com.zaxxer.hikari.pool.HikariPool    : HikariPool-1 - Added connection conn0: url=jdbc:mysql://127.0.0.1:3306/livraria?serverTimezone=UTC&useSSL=false&allowPublicKeyRetrieval=true&useLegacyDatetimeCode=false&characterEncoding=UTF-8
2025-12-02T00:16:56.013-03:00 INFO 37852 --- [pedido] [           main] com.zaxxer.hikari.HikariDataSource   : HikariPool-1 - Start completed.
2025-12-02T00:16:56.567-03:00 INFO 37852 --- [pedido] [           main] o.h.e.t.j.p.i.JtaPlatformInitiator  : HHH000489: No JTA platform available (set 'hibernate.transaction.jta.platform' to enable JTA platform integration)
2025-12-02T00:16:56.603-03:00 INFO 37852 --- [pedido] [           main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2025-12-02T00:16:56.798-03:00 WARN 37852 --- [pedido] [           main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be performed during view rendering. Explicitly configure spring.jpa.open-in-view to disable this warning
2025-12-02T00:16:56.968-03:00 INFO 37852 --- [pedido] [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8084 (http) with context path '/'
2025-12-02T00:16:56.972-03:00 INFO 37852 --- [pedido] [           main] com.ifsp.pedido.PedidoApplication  : Started PedidoApplication in 2.173 seconds (process running for 2.342)
```

## Frete:

```
PS C:\Users\enzo\Downloads\Livraria-Projeto\livraria\frete> mvn spring-boot:run
:: Spring Boot ::        (v3.3.2)

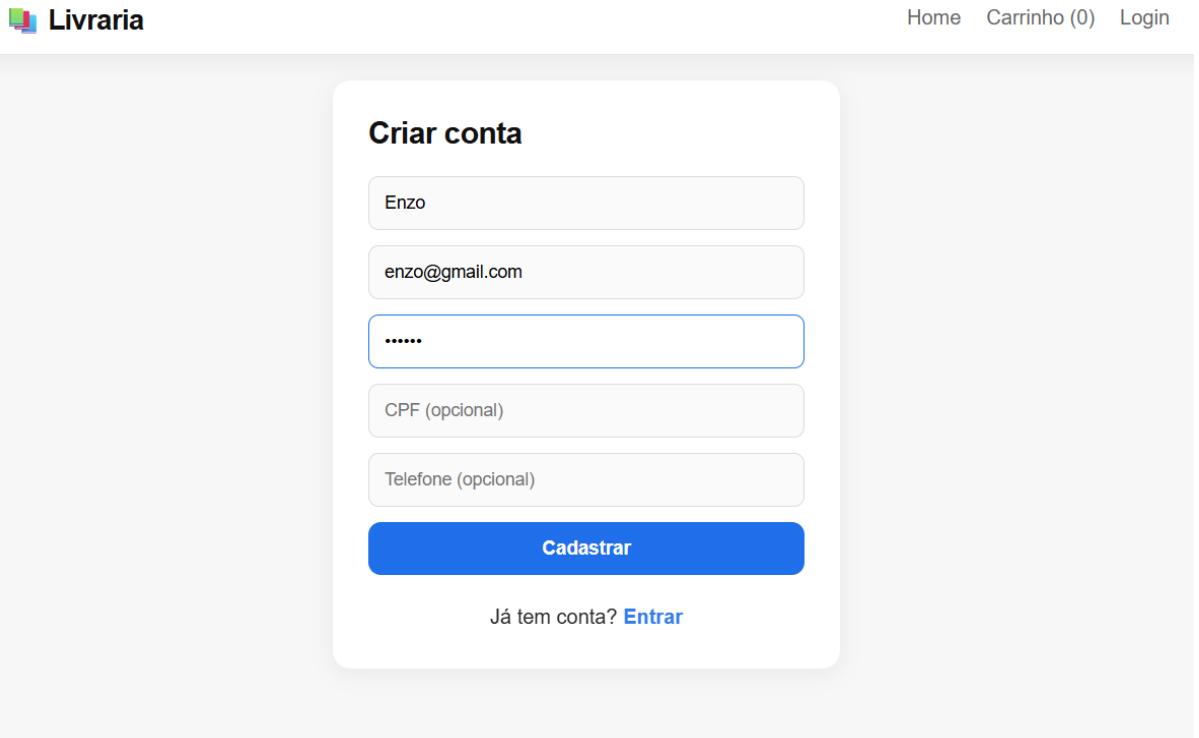
2025-12-02T00:17:50.152-03:00 INFO 31572 --- [frete] [           main] com.ifsp.frete.FreteApplication      : Starting FreteApplication using Java 21.0.8 with PID 31572 (c:\Users\enzo\Downloads\Livraria-Projeto\livraria\frete\target\classes started by enzo in c:\Users\enzo\Downloads\Livraria-Projeto\livraria\frete)
2025-12-02T00:17:50.156-03:00 INFO 31572 --- [frete] [           main] com.ifsp.frete.FreteApplication      : No active profile set, falling back to 1 default profile: "default"
2025-12-02T00:17:50.579-03:00 INFO 31572 --- [frete] [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8086 (http)
2025-12-02T00:17:50.588-03:00 INFO 31572 --- [frete] [           main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2025-12-02T00:17:50.588-03:00 INFO 31572 --- [frete] [           main] o.apache.catalina.core.StandardEngine  : Starting Servlet engine: [Apache Tomcat/10.1.26]
2025-12-02T00:17:50.618-03:00 INFO 31572 --- [frete] [           main] o.a.c.c.C.[Tomcat].[localhost].[]       : Initializing Spring embedded WebApplicationContext
2025-12-02T00:17:50.619-03:00 INFO 31572 --- [frete] [           main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 439 ms
2025-12-02T00:17:50.772-03:00 INFO 31572 --- [frete] [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8086 (http) with context path '/'
2025-12-02T00:17:50.775-03:00 INFO 31572 --- [frete] [           main] com.ifsp.frete.FreteApplication      : Started FreteApplication in 0.801 seconds (process running for 0.954)
```

## Pagamento:

```
PS C:\Users\enzo\Downloads\Livraria-Projeto\livraria\pagamento> mvn spring-boot:run
2025-12-02T00:15:47.768-03:00 INFO 35876 --- [pagamento] [           main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 694 ms
2025-12-02T00:15:47.849-03:00 INFO 35876 --- [pagamento] [           main] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo [name: default]
2025-12-02T00:15:47.876-03:00 INFO 35876 --- [pagamento] [           main] org.hibernate.Version      : HHH000412: Hibernate ORM core version 6.5.2.Final
2025-12-02T00:15:47.892-03:00 INFO 35876 --- [pagamento] [           main] o.h.c.internal.RegionFactoryInitiator : HHH000026: Second-level cache disabled
2025-12-02T00:15:48.038-03:00 INFO 35876 --- [pagamento] [           main] o.s.o.j.p.SpringPersistenceUnitInfoTransformer : No LoadTimeWeaver setup: ignoring JPA class
2025-12-02T00:15:48.051-03:00 INFO 35876 --- [pagamento] [           main] com.zaxxer.hikari.HikariDataSource   : HikariPool-1 - Starting...
2025-12-02T00:15:48.188-03:00 INFO 35876 --- [pagamento] [           main] com.zaxxer.hikari.pool.HikariPool    : HikariPool-1 - Added connection conn0: url=jdbc:mysql://127.0.0.1:3306/livraria?serverTimezone=UTC&useSSL=false&allowPublicKeyRetrieval=true&useLegacyDatetimeCode=false&characterEncoding=UTF-8
2025-12-02T00:15:48.189-03:00 INFO 35876 --- [pagamento] [           main] com.zaxxer.hikari.HikariDataSource   : HikariPool-1 - Start completed.
2025-12-02T00:15:48.808-03:00 INFO 35876 --- [pagamento] [           main] o.h.e.t.j.p.i.JtaPlatformInitiator  : HHH000489: No JTA platform available (set 'hibernate.transaction.jta.platform' to enable JTA platform integration)
2025-12-02T00:15:48.844-03:00 INFO 35876 --- [pagamento] [           main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2025-12-02T00:15:49.063-03:00 WARN 35876 --- [pagamento] [           main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be performed during view rendering. Explicitly configure spring.jpa.open-in-view to disable this warning
2025-12-02T00:15:49.233-03:00 INFO 35876 --- [pagamento] [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8085 (http) with context path '/'
2025-12-02T00:15:49.238-03:00 INFO 35876 --- [pagamento] [           main] c.ifsp.pagamento.PagamentoApplication : Started PagamentoApplication in 2.398 seconds (process running for 2.571)
```

## Gateway:

## Testes integrados



## Descubra livros

**Clean Code***Robert C. Martin***R\$ 129.90****Domain-Driven Design***Eric Evans***R\$ 199.90****Refactoring***Martin Fowler***R\$ 159.90****Clean Architecture***Robert C. Martin***Estruturas de Dados e  
Algoritmos em Java****1984***George Orwell*



## Meu Carrinho

### Itens (2)



#### Domain-Driven Design

Eric  
Evans

R\$  
199.90

Quantidade:

- 1 +

Subtotal

R\$  
199.90



#### Clean Code

Robert  
C.  
Martin

R\$  
129.90

Quantidade:

- 1 +

Subtotal

R\$  
129.90



### Resumo do Pedido

Subtotal (2 itens)

R\$ 329.80

Calcular Frete

00000-000

OK

Frete

Calcular

Total

**R\$ 329.80**

**Finalizar Compra**

**Continuar Comprando**

Limpar Carrinho

✓ Frete grátis para todo Brasil

✓ Entrega em até 7 dias úteis

✓ Compra 100% segura

 **Dados de Entrega**

CEP \*

12900-480



Estado \*

SP

Cidade \*

Bragança Paulista

Endereço \*

Rua Coronel Assis Gonçalves

Número \*

10

Complemento

Apto, Bloco...

Bairro \*

Centro

✓ Frete calculado: R\$ 15.00

## Finalizar Compra

### Dados de Entrega

CEP \*

12900-480



Estado \*

SP

Cidade \*

Bragança Paulista

Endereço \*

Rua Coronel Assis Gonçalves

Número \*

10

Complemento

Apto, Bloco...

Bairro \*

Centro

✓ Frete calculado: R\$ 15.00

### Dados de Pagamento



Cartão de Crédito



Cartão de Débito



PIX



Boleto

Após confirmar o pedido, você receberá um QR Code para pagamento via PIX.

O pedido será processado após a confirmação do pagamento.

### Resumo do Pedido



Domain-Driven Design

Qtd: 1

R\$ 199.90



Clean Code

Qtd: 1

R\$ 129.90

Subtotal

R\$ 329.80

Frete

R\$ 15.00

Total

R\$ 344.80

**Confirmar Pedido**[← Voltar ao Carrinho](#)

Compra 100% segura e protegida

## Dados de Pagamento



Cartão  
de  
Crédito



Cartão  
de  
Débito



PIX



Boleto

 Após confirmar o pedido, você receberá um QR Code para pagamento via PIX.

O pedido será processado após a confirmação do pagamento.