Instituto Politécnico de Viana do Castelo  
Escola Superior de Tecnologia e Gestão

**Tecnologias e Programação de Sistema de Informação**

**Programação Orientada por Objetos**

**Relatório Trabalho Prático Nº.1**

Shape

Lucas de Linhares N.º 32187

Maio 2024

[1. Introdução 2](#_Toc336326673)

[2. Objetivos 3](#_Toc1691724191)

[3. Fundamentação Teórica 3](#_Toc268999297)

[3.1. Rust 3](#_Toc1006631526)

[3.2. Java 4](#_Toc578967707)

[3.3. Gradle 4](#_Toc964999298)

[3.4. FastJson 4](#_Toc1572820796)

[4. Implementação 4](#_Toc479436922)

[6. Dificuldades Encontradas e Resolução 6](#_Toc164895411)

[7. Conclusão 6](#_Toc1246571376)

# Introdução

No contexto da disciplina de Programação Orientada por Objetos, desenvolvi um projeto prático que visa a implementação de um sistema de gestão para uma oficina utilizando a linguagem de programação Java. Este trabalho tem como objetivo principal a aplicação dos conceitos aprendidos ao longo do curso, através da criação de um programa que gerencie de forma eficiente as informações referentes aos trabalhadores da oficina, aos clientes e aos arranjos de veículos.

Para atingir este objetivo, estruturei o programa com base nos princípios da programação orientada por objetos, onde cada componente principal do sistema (trabalhadores, clientes e arranjos de veículos) é representado por uma classe específica. Desenvolvi cada classe de maneira a conter atributos, construtores e métodos que permitem a manipulação dos dados de forma organizada e modular.

Adicionalmente, para criar uma interface no terminal que fosse funcional e agradável ao utilizador, recorri a uma linguagem de baixo nível, tendo escolhido Rust para este propósito. Esta escolha permitiu-me um controlo mais preciso sobre os recursos do sistema e uma melhor performance da interface.

O código fonte do projeto está disponível em: <https://github.com/lucascompython/TP1-POP>

# Objetivos

Na realização deste trabalho tivemos os seguintes objetivos:

* Aplicar conhecimentos adquiridos na sala de aula num trabalho pratico;
* Adquirir conhecimentos sobre novas tecnologias como:
  + O API FFM introduzido no JDK 22
  + Rust
  + Programação de baixo nível
* Garantir uma ótima performance
* Escrever código rápido e limpo

# Fundamentação Teórica

Aqui explicarei o que cada uma das tecnologias envolvidas no desenvolvimento deste projeto são.

# Rust

Rust é uma linguagem de programação multiparadigma compilada desenvolvida pela Mozilla. É projetada para ser "segura, concorrente e prática", mas diferente de outras linguagens seguras, Rust não usa coletor de lixo.

# Java

Java é uma linguagem de programação orientada a objetos. Diferente das linguagens de programação modernas, que são compiladas para código nativo, Java é compilada para um bytecode que é interpretado por uma máquina virtual (Java Virtual Machine, abreviada JVM). A linguagem de programação Java é a linguagem convencional da Plataforma Java, mas não é a sua única linguagem.

# Gradle

Gradle é uma ferramenta de automação de build para desenvolvimento de software multilíngue. Ele controla o processo de desenvolvimento nas tarefas de compilação e empacotamento para teste, implantação e publicação. As linguagens suportadas incluem Java (bem como Kotlin, Groovy, Scala), C/C++ e JavaScript. Gradle baseia-se nos conceitos de Apache Ant e Apache Maven e introduz uma linguagem específica de domínio baseada em Groovy e Kotlin em contraste com a configuração de projeto baseada em XML usada pelo Maven. Gradle usa um gráfico acíclico direcionado para determinar a ordem em que as tarefas podem ser executadas, fornecendo gerenciamento de dependências. Ele é executado na Java Virtual Machine.

# FastJson

Fastjson é uma biblioteca Java que pode ser usada para converter objetos Java em sua representação JSON. Também pode ser usado para converter uma string JSON em um objeto Java equivalente.

# Implementação

A biblioteca que escreci em Rust comunica com a JVM usando o API Foreign Function & Memory (FFM)introduzido no JDK 22.

Usei a ferramenta Jextract, para gerar automaticamente “bindings” em Java ao ler um header em C. O Jextract roda a cada build (ver o código abaixo).

Código responsável por gerar os bindings em Java utilizando o Jextract, no *build.gradle.kts*:

tasks.register<Exec>("jextract") {

val ext = if (Os.isFamily(Os.FAMILY\_WINDOWS)) ".bat" else ""

commandLine = listOf(

"../jextract/jextract$ext",

"--output", "src/main/java",

"--target-package", "org.tp1.bindings",

"--library", "terminal\_utils",

"../terminal\_utils/bindings.h"

)

}

tasks.withType<JavaCompile> {

dependsOn("jextract")

}

Neste trecho de código, registamos uma nova tarefa do tipo *Exec* fazemos com que a tarefa de compilar o Java dependa de criar os bindings. Assim a cada build o Jextract vai ser executado.

Para gerar o header em C usei uma build-time depency, Cbindgen para gerar automaticamente o header ao ler as funções e structs em Rust que são sujeitas a ser partilhadas na biblioteca. Este header também é gerado a cada build (ver o código abaixo).

Código responsável por gerar o header em C utilizando o cbindgen no ficheiro *build.rs*:

fn main() {

let crate\_dir = std::env::var("CARGO\_MANIFEST\_DIR").unwrap();

cbindgen::Builder::new()

.with\_crate(crate\_dir)

.with\_language(cbindgen::Language::C)

.with\_no\_includes()

.with\_sys\_include("stdint.h")

.with\_sys\_include("stdbool.h")

.with\_include\_guard("BINDINGS\_H")

.generate()

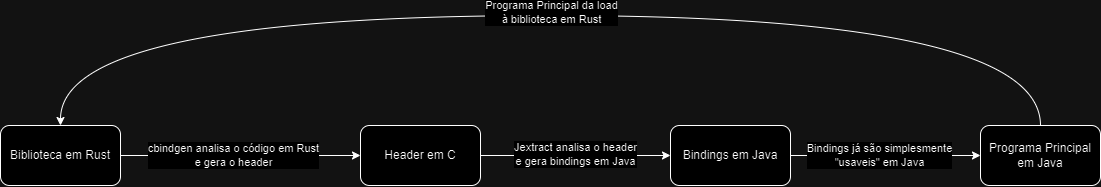
.expect("Unable to generate bindings")

.write\_to\_file("bindings.h");

}

Aqui eu decidi manualmente adicionar dois *includes* em vez de deixar o Cbindgen tratar disso, pois o Cbindgen também adiciona outros headers que não são necessários. Decidi então removê-los e adicionar manualmente apenas os necessários pois senão o Jextract iria gerar bindings desnecessários.

Ver o diagrama abaixo:



# 6. Dificuldades Encontradas e Resolução

Tive algumas dificuldades ao passar informação entre Java e Rust. Mas ao fim de várias idas a documentação do Jextract, consegui fazer o que queria.

# 7. Conclusão

Este projeto deu me uma oportunidade de aprofundar os meus conhecimentos com o ecossistema de Java e programação de baixo nível.

# 8. Bibliografia

<https://en.wikipedia.org/wiki/Rust_(programming_language>)

<https://en.wikipedia.org/wiki/Java_(programming_language>)

<https://en.wikipedia.org/wiki/Gradle>

<https://github.com/alibaba/fastjson>