

## 1. IDENTIFICAÇÃO

|          |                                                                    |
|----------|--------------------------------------------------------------------|
| Padrão   | <b>Desenvolvimento Java em Microserviços com Quarkus e JDK 17+</b> |
| Segmento | <b>Arquitetura de Soluções</b>                                     |
| Código   | <b>P05.010</b>                                                     |
| Revisão  | <b>v.2026</b>                                                      |

## 2. PUBLICAÇÃO

| Versão        | Data para adoção            | Publicação |
|---------------|-----------------------------|------------|
| <b>v.2026</b> | <b>24 de agosto de 2026</b> |            |

## 3. PROPÓSITO DO PADRÃO

Padronização do desenvolvimento e da arquitetura de novas aplicações de software baseadas em microserviços e APIs RESTful, utilizando a linguagem de programação Java na Prefeitura do Rio de Janeiro. Com a evolução tecnológica e a demanda por serviços públicos mais eficientes, escaláveis e resilientes, estabelece-se a transição das arquiteturas Web monolíticas tradicionais (como o MVC legado definido em P05.003) para uma arquitetura ágil e cloud-native de microserviços. Este padrão visa uniformizar a construção das soluções municipais para propiciar:

- **Aumento de Produtividade:** Utilização do Quarkus e suas capacidades de recarregamento rápido em tempo de desenvolvimento (Live Coding).
- **Otimização de Infraestrutura:** Uso de microserviços Java otimizados para execução eficiente em ambientes de containers Kubernetes / OpenShift.
- **Segurança e Conformidade Corporativa:** Uso mandatório de imagens de base homologadas e mantidas com suporte corporativo da Red Hat.
- **Interoperabilidade Facilitada:** Padronização dos protocolos de comunicação em APIs RESTful JSON, integradas nativamente com soluções corporativas de identidade (Keycloak OIDC).
- **Garantia de Manutenibilidade:** Uniformização de frameworks de persistência, validação e observabilidade de modo a facilitar a sustentação e transferência de conhecimento entre equipes da IPLANRIO.

## 4. RESPONSÁVEL PELO PADRÃO

|             |                                                   |
|-------------|---------------------------------------------------|
| Órgão       | IplanRio                                          |
| Diretoria   | DSI - Diretoria de Sistemas                       |
| Setor       | GDS-1 – Gerência de Desenvolvimento de Sistemas 1 |
| Responsável | Supervisor da Equipe Java                         |

## 5. DESCRIÇÃO DO PADRÃO

Este padrão estabelece que todas as novas aplicações ou manutenções evolutivas críticas de sistemas Java na Prefeitura do Rio de Janeiro devem utilizar o framework Quarkus, rodando sobre o Java Development Kit (JDK) na versão 17 ou superior (LTS). A arquitetura de implantação deve se basear estritamente em containers Docker executados sobre a infraestrutura de orquestração municipal.

### 5.1 Execução Quarkus com Máquina Virtual Java (JVM)

O ambiente de execução padrão estabelecido para o Quarkus é o modo JVM (Hotspot). Embora o Quarkus suporte a compilação nativa, a execução baseada em JVM tradicional de mercado é o modelo homologado principal devido à sua compatibilidade nativa com 100% das bibliotecas do ecossistema Java, facilidade de diagnóstico em tempo de execução (dumps de memória e thread dumps), estabilidade comprovada do coletor de lixo (Garbage Collector) corporativo e maturidade do compilador JIT.

Referência Tecnológica (GraalVM / Mandrel): A compilação nativa baseada em GraalVM ou Red Hat Mandrel é mantida neste padrão apenas como referência opcional de alta performance para casos de uso específicos. O seu uso deve ser restrito a microsserviços com requisitos extremos de tempo de inicialização (Cold Start), como funções Serverless, e requer aprovação prévia formal do Comitê de Arquitetura de Software da IPLANRIO.

### 5.2 Dependência de Imagens Docker Homologadas pela Red Hat

Visando mitigar vulnerabilidades de segurança (CVEs) em produção e garantir a estabilidade do DataCenter municipal, é estritamente obrigatório que todas as imagens de container geradas dependam de imagens de base oficiais e homologadas da Red Hat, especificamente a Universal Base Image (UBI) e o Red Hat Build of OpenJDK. Fica proibido o uso de imagens de base não homologadas como Alpine, Debian puras ou OpenJDK genéricos de repositórios públicos sem verificação de segurança.

Vantagens do uso obrigatório de imagens homologadas pela Red Hat:

- **Suporte e Estabilidade Corporativa:** Assegura compatibilidade e estabilidade com a plataforma Kubernetes/OpenShift operada pelo município.
- **Mitigação de CVEs:** Red Hat publica correções de segurança críticas em suas imagens UBI com tempo de resposta ágil, garantindo que as imagens de produção estejam sempre protegidas.
- **Conformidade Regulatória (FIPS):** Imagens UBI suportam criptografia em conformidade com os padrões federais de segurança (FIPS), exigida para sistemas municipais críticos.

## 6. POLÍTICA E NORMATIZAÇÃO DE USO

Fica estabelecido o padrão tecnológico de Desenvolvimento Java baseado em Quarkus JVM e JDK 17+ a partir dos componentes listados na especificação técnica deste documento.

Todo desenvolvimento interno, terceirizado ou aquisição de novas soluções sistêmicas na linguagem Java na Prefeitura do Rio de Janeiro deverá estar em estrita conformidade com as referências técnicas deste padrão.

O pipeline de Integração Contínua (CI/CD) da IPLANRIO bloqueará automaticamente qualquer implantação (deploy) cujo arquivo Dockerfile utilize imagens de base diferentes das homologadas Red Hat UBI / OpenJDK.

Qualquer dúvida técnica ou solicitação de exceção (como uso de compilação nativa GraalVM ou frameworks externos) deve ser formalizada e avaliada pela Gerência de Desenvolvimento de Sistemas 1 (GDS-1) da IPLANRIO.

Visando manter a tecnologia atualizada e segura, os componentes deste padrão serão revisados pela Diretoria de Sistemas e pelo profissional responsável por esse padrão com periodicidade máxima de 365 dias.

## 7. ESPECIFICAÇÃO TÉCNICA

Abaixo são especificados os componentes mínimos adotados e homologados para compor o ecossistema de microsserviços Java Quarkus na Prefeitura do Rio de Janeiro:

| Componente                 | Especificação Homologada                                                                                        | Situação              |
|----------------------------|-----------------------------------------------------------------------------------------------------------------|-----------------------|
| Runtime de Linguagem       | Red Hat Build of OpenJDK 17 ou superior (LTS)                                                                   | Adotado (Obrigatório) |
| Framework Base             | Quarkus Framework (versão LTS 3.x)                                                                              | Adotado (Obrigatório) |
| Imagem de Base Docker      | Red Hat Universal Base Image (UBI 8 / UBI 9) com JDK integrado (Ex: registry.access.redhat.com/ubi8/openjdk-17) | Adotado (Obrigatório) |
| Runtime alternativo        | Red Hat Mandrel / GraalVM (Apenas compilação nativa)                                                            | Referência Opcional   |
| Camada REST API            | Quarkus RESTEasy Reactive (com serialização JSON via Jackson)                                                   | Adotado (Obrigatório) |
| Persistência de Dados      | Quarkus Hibernate ORM com Panache (Drivers JDBC homologados PostgreSQL / MariaDB)                               | Adotado (Recomendado) |
| Validação de Dados         | Hibernate Validator (Jakarta Validation API)                                                                    | Adotado (Obrigatório) |
| Segurança / Autenticação   | Quarkus OIDC (OpenID Connect) integrado corporativamente com Keycloak                                           | Adotado (Obrigatório) |
| Observabilidade / Métricas | Micrometer Metrics integrados com SmallRye Health (Métricas de Liveness e Readiness)                            | Adotado (Obrigatório) |
| Documentação de API        | Quarkus SmallRye OpenAPI (Geração automatizada de Swagger UI em ambiente de desenvolvimento)                    | Adotado (Recomendado) |

## 8. DEFINIÇÃO E ABREVIATURAS

|                      | Definição                                                                                                                                                  |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Quarkus</b>       | Framework Java nativo do Kubernetes de pilha cheia (Full-stack), otimizado para containers e focado em baixíssimo consumo de recursos.                     |
| <b>JVM</b>           | Java Virtual Machine. Ambiente de execução em que os programas Java compilados (bytecodes) são executados em tempo real.                                   |
| <b>UBI</b>           | Universal Base Image. Imagem de container base da Red Hat que provê um ecossistema operacional Linux seguro, estável e livre de royalties.                 |
| <b>Keycloak OIDC</b> | OpenID Connect implementado através do Keycloak corporativo, utilizado para federação de identidades e controle de acessos unificado (RBAC) na Prefeitura. |
| <b>GraalVM</b>       | Máquina virtual de alto desempenho capaz de compilar código Java antecipadamente (AOT) em um executável nativo do sistema operacional.                     |

## 9. REFERÊNCIAS

[1] PORTAL OFICIAL DO E-PINGRIO. Catálogo de Interoperabilidade e Padrões Tecnológicos da Prefeitura do Rio de Janeiro. Disponível em: [https://iplanrio.prefeitura.rio/e-pingrio\\_catalogo/](https://iplanrio.prefeitura.rio/e-pingrio_catalogo/).

[2] RED HAT. Red Hat Build of OpenJDK & Universal Base Images (UBI). Disponível em: <https://access.redhat.com/>.

[3] QUARKUS COMMUNITY. Quarkus - Supersonic Subatomic Java. Documentação de referência e extensões. Disponível em: <https://quarkus.io/>.

[4] IPLANRIO. e-PINGRIO – P05.003: Desenvolvimento em Java para Aplicações Web, v. 2014.

[5] IPLANRIO. e-PINGRIO – P05.005: Desenvolvimento em Python para Aplicações Web (API RESTful), v. 2025.

## 10. GRUPO TÉCNICO RESPONSÁVEL PELA ELABORAÇÃO DO PADRÃO

Diretoria de Sistemas IplanRio  
**Wesley Cunha do Nascimento**  
**Marcia Pereira Fontes**  
**Eugenio Carlos da Cunha Pereira**