**CENTRO UNIVERSITÁRIO IMEPAC ARAGUARI**

**FERNANDO CORSINO RESENDE SANTOS**

**APLICAÇÃO DE MICROSSERVIÇOS NA CLÍNICA MÉDICA: USO DA TECNOLOGIA PARA MELHORAR O ATENDIMENTO DE CLINICA MEDICA**

**ARAGUARI**

**DEZEMBRO/2024**

**FERNANDO CORSINO RESENDE SANTOS**

**APLICAÇÃO DE MICROSSERVIÇOS NA CLÍNICA MÉDICA: USO DO MAVEN**

Trabalho de Pesquisa apresentado ao Curso de Tecnologia em Análise e Desenvolvimento de Sistemas do Centro Universitário IMEPAC, como requisito parcial à obtenção do aproveitamento da disciplina de Programação Orientada a Objetos.

Orientador: Prof. Me. Everton Hipólito de Freitas

**ARAGUARI**

**DEZEMBRO/2024**

### **4. REVISÃO TEÓRICA**

O Apache Maven é descrito como uma ferramenta essencial para equipes que trabalham com projetos Java, mas que também pode ser usada em outras linguagens. Sua principal finalidade é oferecer automação e padronização no processo de construção, gerenciamento e publicação de aplicações. Além disso, sua flexibilidade é destacada pela possibilidade de extensão via plugins, tornando-o adaptável a diferentes necessidades.

A seção inclui uma análise das características de fases de um projeto, repositório de dependência e automatizador de build.

##### **4.1.1 O QUE É O MAV****EN**

O Maven é uma ferramenta de automação de build e gerenciamento de projetos desenvolvida pela Apache. Seu principal objetivo é simplificar o processo de construção (build) de projetos, gerenciar dependências externas e padronizar o ciclo de vida de um projeto Java. Com o Maven, é possível gerenciar de forma centralizada as bibliotecas e plugins usados em seu projeto, garantindo que tudo funcione de forma integrada e eficiente.

##### **4.1.1 FASES DE UM PROJETO**

Uma fase nada mais é do que um estágio onde são executadas algumas regras sobre o projeto e se obtêm algum resultado no final.

As fases do Maven permitem dividir o ciclo de vida do build em partes claras e automatizadas, garantindo que todas as etapas sejam executadas de forma consistente. Dependendo do objetivo, você pode escolher executar apenas fases específicas ou o ciclo completo.

Como por exemplo:

**Compile**: a fase compile compila o código-fonte do projeto

**Test**: a fase test testa o código-fonte compilado usando um framework de teste adequado

**Package**: a fase package leva o código compilado e o empacota em um formato distribuível (como JAR, WAR, etc.)

**Validate**: verifica se todas as informações necessárias estão disponíveis para o projeto.

**Verify**: executa verificações para garantir que o pacote é válido e atende a critérios de qualidade.

**Install**: instala o pacote no repositório local para uso como dependência em outros projetos locais.

**Deploy**: copia o pacote final para o repositório remoto para compartilhamento com outros desenvolvedores e projetos.

**4.1.2 REPOSITÓRIO DE DEPENDÊNCIAS**

No **Maven**, um **repositório de dependências** é um local onde as bibliotecas, plugins e outros artefatos necessários para um projeto são armazenados e de onde podem ser recuperados. Ele é uma parte central do funcionamento do Maven, já que o sistema de gerenciamento de dependências permite que os desenvolvedores declarem as dependências no arquivo pom.xml, e o Maven automaticamente baixa e gerencia essas bibliotecas a partir dos repositórios.

Tipos de repositórios Maven:

Repositório local: é armazenado no computador do desenvolvedor

Repositório Maven Central: O Maven Central é o repositório remoto padrão, mantido pela comunidade Maven, que contém milhares de bibliotecas disponíveis publicamente.

Repositórios Remotos Personalizados: São repositórios adicionais que você pode configurar no arquivo pom.xml ou settings.xml

**4.1.3**. **AUTOMATIZADOR DE BUILD**

#### O Maven é um dos automatizadores de build mais populares para projetos Java (e outras linguagens baseadas na JVM). Ele gerencia todo o ciclo de vida de um projeto, incluindo tarefas como compilação, teste, empacotamento, instalação e implantação. Além disso, o Maven é amplamente utilizado por seu eficiente gerenciamento de dependências.

#### Com o Maven, você pode automatizar o processo de build, compilando o código, realizando testes e gerando artefatos (como arquivos JAR, WAR ou EAR). Isso garante que o processo de construção do projeto seja reproduzível e consistente, independentemente de onde o código esteja sendo construído.

#### **4.2 CONCLUSÃO**

O Maven é uma ferramenta indispensável para desenvolvedores Java, pois facilita o gerenciamento de dependências e automatiza o processo de build, tornando o ciclo de vida de um projeto muito mais organizado e eficiente. Ao adotar o Maven, você estará economizando tempo e recursos, além de garantir que seu projeto siga boas práticas de desenvolvimento e tenha um processo de build consistente e reproduzível.

**5. REFERÊNCIA**

Apache Maven Project. (2021). Introduction to the Build Lifecycle. Disponível em: <https://maven.apache.org/guides/introduction/introduction-to-the-lifecycle.html>.