

## GUIA TÉCNICO: CRIAÇÃO DE PLUG-INS PARA O ARCHI (ARCHIMATE TOOL)

### 1. ENTENDENDO A ARQUITETURA DO ARCHI

---

O Archi é construído sobre o Eclipse Rich Client Platform (RCP) e usa o Eclipse Plug-in Framework (Equinox + OSGi).

Cada módulo do Archi é um plug-in Eclipse. Você pode criar o seu próprio, empacotado como um bundle OSGi.

O modelo é baseado em EMF (Eclipse Modeling Framework):

- com.archimatetool.model → contém interfaces como:
  - IArchimateModel
  - IArchimateElement
  - IFolder
  - IRelationship
  - IProperty
  - IArchimateFactory

### 2. PREPARAR O AMBIENTE DE DESENVOLVIMENTO

---

Dependências:

- Eclipse IDE for RCP and RAP Developers
- Java 17+
- Archi source code (<https://github.com/archimatetool/archi>)
- Plug-in Development Environment (PDE)

Passos:

1. Clone o repositório do Archi
2. Importe os projetos no Eclipse
3. Verifique que módulos como com.archimatetool.editor e com.archimatetool.model estão disponíveis

### 3. CRIAR SEU PLUG-IN

---

No Eclipse:

- File → New → Plug-in Project
- Nome: com.meuplugin.archi.extension
- Escolha "No" para aplicação RCP
- Configure o MANIFEST.MF com dependências:

Require-Bundle: com.archimatetool.editor,  
com.archimatetool.model,  
org.eclipse.ui,

org.eclipse.core.runtime

#### 4. ESTRUTURA DO PROJETO

---

```
com.meuplugin.archi.extension/
  META-INF/
    MANIFEST.MF
  plugin.xml
  src/
    com/meuplugin/archi/handlers/AddCustomElementHandler.java
  icons/
```

#### 5. EXEMPLO DE PLUGIN.XML

---

```
<plugin>
  <extension point="org.eclipse.ui.commands">
    <command id="com.meuplugin.archi.extension.addElement"
      name="Add Custom Element"
      description="Adiciona um novo elemento Archi personalizado"/>
  </extension>

  <extension point="org.eclipse.ui.handlers">
    <handler commandId="com.meuplugin.archi.extension.addElement"
      class="com.meuplugin.archi.handlers.AddCustomElementHandler"/>
  </extension>

  <extension point="org.eclipse.ui.menus">
    <menuContribution locationURI="menu:tools">
      <command commandId="com.meuplugin.archi.extension.addElement"
        label="Add Custom Archi Element" style="push"/>
    </menuContribution>
  </extension>
</plugin>
```

#### 6. HANDLER JAVA

---

```
public class AddCustomElementHandler extends AbstractHandler {
  @Override
  public Object execute(ExecutionEvent event) throws ExecutionException {
    IArchimateModel model = getActiveModel();
    if (model == null) return null;
```

```

IArchimateFactory factory = IArchimateFactory.eINSTANCE;
IArchimateElement element = factory.createBusinessActor();
element.setName("Meu Novo Elemento");

IFolder folder = model.getDefaultFolderForElement(element);
folder.getElements().add(element);

model.eNotify(null);
return null;
}

private IArchimateModel getActiveModel() {
return EditorModelManager.INSTANCE.getActiveArchimateModel();
}
}
}

```

## 7. EXECUTAR E TESTAR

---

- Run As → Eclipse Application
- O Archi abrirá com o plugin
- Vá em Tools → Add Custom Archi Element

## 8. EMPACOTAR O PLUGIN

---

- File → Export → Deployable plug-ins and fragments
- Copie o .jar para Archi/plugins/
- Reinicie o Archi

## 9. MANIPULAÇÃO AVANÇADA

---

Iterar sobre elementos:

```

for (IArchimateElement e : model.getAllElements()) {
System.out.println(e.getName());
}

```

Criar relações:

```

IRelationship rel = factory.createAssignmentRelationship();
rel.setSource(sourceElement);
rel.setTarget(targetElement);
model.getDefaultFolderForElement(rel).getElements().add(rel);

```

Adicionar propriedades:

```

IProperty prop = factory.createProperty();

```

```
prop.setKey("status");
prop.setValue("ativo");
element.getProperties().add(prop);
```

## 10. REFERÊNCIAS

---

- Archi GitHub – Developer Docs: <https://github.com/archimatetool/archi/wiki>
- Eclipse Plug-in Development Guide:  
[https://help.eclipse.org/latest/topic/org.eclipse.pde.doc.user/guide/pde\\_intro.htm](https://help.eclipse.org/latest/topic/org.eclipse.pde.doc.user/guide/pde_intro.htm)
- Archi Model API – Javadoc: <https://www.archimatetool.com/dev/docs/javadoc/latest/>