

# Ferramentas

## 1. Introdução

Neste documento será apresentada a ambientação e o ferramental padrão para o desenvolvimento do projeto. A escolha de ferramentas para a produção de artefatos que não tiverem sido especificados ficará a critério do responsável pelo desenvolvimento do respectivo artefato.

## 2. *Templates*

A documentação do projeto é baseada nos *templates* da última versão do *OpenUP*, quando disponíveis. Para os artefatos em que um arquivo de *template* não existir no *OpenUP*, então a equipe produzirá um *template* próprio com base nas páginas explanatórias do próprio *OpenUP*, com formatação semelhante ao dos *templates* existentes. Para os artefatos em que não houver uma página explanatória no *OpenUP*, a equipe produzirá seus próprios *templates* com base em pontos que considerar importantes para aquele artefato segundo o conteúdo presente em livros texto e ministrado em sala de aula, a serem decididos durante reunião da equipe.

## 3. Documentação

A produção e edição da documentação deve ser realizada utilizando arquivos em formato RTF (*Rich Text Format*), uma vez que os *templates* dos artefatos são disponibilizados neste formato. Para tanto, devem ser utilizados os processadores de texto offline ou online: LibreOffice Writer, Microsoft Word (2010 ou mais recente), WordPad ou Google Docs.

A submissão da versão final de cada documento deve ser realizada em formato PDF (*Portable Document Format*), onde a conversão pode ser realizada utilizando algum dos processadores supracitados ou alguma outra ferramenta a escolha do responsável pela produção do artefato.

## 4. Código-fonte

A produção e edição de arquivos referentes ao código-fonte pode ser feita no ambiente de desenvolvimento da preferência do implementador. A submissão dos arquivos de código-fonte deve ser por texto em claro (*plain text*), codificação UTF-8 ou UTF-16, com a extensão padrão dos arquivos de código-fonte referentes à linguagem de programação utilizada.

## 5. Arquivos objeto, executáveis e testes

A produção de arquivos objeto e executáveis deve ser realizada na linguagem de programação *Swift*, através do ambiente de desenvolvimento integrado (*IDE*) *XCode*, que inclui o compilador da linguagem *Swift SDK*, o framework de testes *XCTests* e o framework de automação de testes *Fastlane*, que possibilitam a automação do processo de execução de testes de unidade, de alguns testes de integração e o *checkout* automático no repositório em caso de sucesso nos testes.

## 6. Canais de comunicação, compartilhamento de arquivos e controle de versão

A equipe deve se comunicar durante a maior parte do tempo através do aplicativo de mensageria por texto e voz *Discord*, por meio de um servidor aberto apenas aos membros da equipe de desenvolvimento do projeto. Uma vez que este aplicativo possui um histórico persistente das mensagens trocadas e também possibilita o compartilhamento de arquivos, ele deve ser utilizado para a avaliação dos artefatos submetidos, e inicialmente para o controle de versões enquanto o repositório não for propriamente configurado.

Posteriormente, uma vez configurado o repositório do projeto no GitHub, o controle de versões pode ser realizado também através de um cliente *git* a escolha de cada membro. Nesta situação, a avaliação dos artefatos continuará a ser realizada através do *Discord*, e a configuração do repositório não excluindo a possibilidade de arquivos continuarem a ser compartilhados no servidor do *Discord*.

## 7. Outras ferramentas

### 7.1. Diagramas

Para a produção de diagramas, recomenda-se a utilização do serviço *web draw.io*, que permite a produção de diferentes classes de diagramas, incluindo UML e fluxogramas, e seu posterior compartilhamento e/ou exportação para arquivos de imagem.