

## NORMAS PARA AS IMPLEMENTAÇÕES DOS PADRÕES

1) As implementações dos padrões serão realizadas a partir de operações de *refactoring* e evolução do jogo Blok, disponível em Java e C++/Qt nos seguintes repositórios git:

**Java:** git@gitorious.org:inf011/blok-java.git

**C++/Qt:** git@gitorious.org:inf011/blok-qt.git

2) Instalando o Git no seu computador e criando uma conta no Gitorious

No Linux, utilize o gerenciador de pacotes da sua distribuição para instalar o Git. No Windows, faça o *download* do arquivo de instalação em <http://msysgit.googlecode.com/files/Git-1.7.9-preview20120201.exe> e execute o setup com as opções *default*.

O próximo passo é a criação de uma conta no <http://www.gitorious.org>. Realize o procedimento normal no site e valide sua conta através do seu e-mail.

Para obter permissões de *commit* no repositório voce precisa criar um par de chaves SSH de criptografia e indicar, no Gitorious, a chave pública a ser utilizada.

Para criar um par de chaves no Linux abra um terminal e execute os comandos abaixo. No Windows, execute o programa *Git Bash* instalado pelo Git e execute os mesmos comandos abaixo.

```
ssh-keygen -t rsa
```

Indique, duas vezes, a senha da sua chave. Esta senha será solicitada pelo git sempre que você for clonar um novo repositório ou realizar um *commit*. O comando acima irá gerar dois arquivos: *id\_rsa* (sua chave privada – guarde com cuidado) e *id\_rsa.pub* (sua chave pública – pode ser fornecida a outras pessoas).

Estes dois arquivos deverão ser copiadas para a pasta *.ssh/* presente na pasta home do seu usuário.

Execute:

```
cd ~/.ssh/
```

Para verificar se a pasta já existe. Caso ainda não exista execute os seguintes comandos:

```
mkdir ~/.ssh/
```

```
cd ~/.ssh
```

Estes arquivos devem ser guardados (envie por exemplo para o seu *email*) caso você precise usar o Git em uma outra máquina. Neste caso é só copiar estes dois arquivos para a pasta *.ssh/* da máquina nova e executar:

```
chmod 600 ~/.ssh/id_rsa
```

Para limitar o acesso à sua chave privada.

Feito isso, o próximo passo é cadastrar, no Gitorious, a sua chave pública. Para isso, faça o *login* no site, clique em *Dashboard* e depois em *Manage SSH Keys*. Clique em *Add SSH Key* e cole o conteúdo do arquivo `id_rsa.pub`.

O último é enviar um e-mail para [sandroandrade@ifba.edu.br](mailto:sandroandrade@ifba.edu.br), informando o seu usuário no Gitorious e solicitando a inclusão deste usuário no grupo de desenvolvedores que poderão fazer *commit* nos repositórios do trabalho prático.

### 3) Clonando o repositório do trabalho prático no seu computador

Na pasta onde você deseja baixar o código-fonte do trabalho prático digite:

```
git clone git@gitorious.org:inf011/blok-java.git
```

para obter a versão Java do jogo ou

```
git clone git@gitorious.org:inf011/blok-qt.git
```

para obter a versão Qt.

**OBS:** as operações de criação/obtenção e instalação das chaves SSH e clonagem do repositório deverão ser realizados apenas uma vez por máquina sendo utilizada.

### 4) Criando o seu *branch* de trabalho

Cada aluno irá trabalhar em um *branch* distinto. O *branch master* NÃO deverá ser modificado, sendo este somente de uso do professor. Para verificar todos os *branches* existentes digite:

```
git branch -a
```

se não existir ainda um *branch* com o seu nome, você pode criar um novo *branch* com o comando:

```
git branch <seu-nome-mais-ultimo-nome>      Ex: git branch sandroandrade
```

```
git checkout <seu-nome-mais-ultimo-nome>    // para mudar para o seu branch
```

```
git push origin <seu-nome-mais-ultimo-nome>
```

Este último comando envia para o servidor do Gitorious todos os arquivos do projeto, criando assim o seu *branch* no servidor.

### 5) Implementando os padrões e fazendo os *commits*

Após a implementação e testes de uma certa funcionalidade (pode ser um padrão ou parte de um padrão) você deve enviar as alterações para o servidor do Gitorious com os seguintes comandos:

```
git status
```

Este comando irá exibir todos os arquivos modificados deste o último *commit*. Certifique-se que está tudo conforme o esperado. Você pode executar `git diff <nome-do-arquivo>` para ver as diferenças entre a versão atual e a do último *commit* para o arquivo informado. Se quiser descartar as mudanças que você fez em um arquivo e retornar para a última versão submetida execute `git checkout -- <nome-do-arquivo>`.

O processo de submissão das alterações para o servidor do Gitorious é feito em três etapas:

1) Marcar os arquivos que você deseja atualizar utilizando o comando:

```
git add <nome-do-arquivo-ou-pasta>
```

Perceba que ao informar uma pasta todos os seus arquivos e sub-pastas serão recursivamente submetidos ao servidor.

**OBS:** arquivos resultantes do processo de compilação (ex: arquivos `.class` ou `.o`), arquivos temporários ou arquivos de *backup* NÃO devem ser incluídos no repositório. Você não precisa remover estes arquivos, basta não executar o `git add` para eles.

2) Realizar o *commit* no seu repositório git local (na sua máquina):

```
git commit -m "<mensagem-informando-o-que-voce-fez>"
```

3) Realizar o *push* no servidor do Gitorious:

```
git push origin <nome-do-seu-branch>
```

## OBSERVAÇÕES GERAIS SOBRE A IMPLEMENTAÇÃO

1) Para cada classe implementada identifique em um comentário, exatamente antes da declaração, qual **participante** do padrão a classe está implementando. Ex:

<pre>// AbstractFactory class WidgetFactory { ... { };</pre>	<pre>// ConcreteFactory class PMWidgetFactory : public WidgetFactory { } };</pre>	<pre>// Client public static void main } };</pre>
--	---	---

2) Todas as implementações deverão apresentar soluções para o empacotamento dos padrões. Cada pacote será representado por um *package* e cada pacote será armazenado em um arquivo `.jar`.

3) Não basta o código compilar e funcionar para estar correto. Aspectos tais como: legibilidade do código, desempenho, elegância da solução, expressividade do problema resolvido, reutilização e facilidade de modificação futura estão incluídos nos critérios de avaliação.

4) Se houver dúvida quanto a onde aplicar o padrão (problema a ser resolvido) contacte o professor com urgência.

5) Para alguns padrões o trabalho consistirá em uma análise de uma implementação já existente, ao invés da elaboração de uma implementação própria.