



REINALDO_{JR}

REFACTORING CODE

ROTEIRO

1. O que é refatoração
2. Padrões de projeto
3. Prolemáticas de código
4. Command
5. Strategy
6. Decorator
7. Factory Method



REFATORAÇÃO é uma “transformação que preserva comportamento”.

“Uma alteração feita na estrutura interna do software para torná-lo mais fácil de ser entendido e menos custoso de ser modificado sem alterar seu comportamento observável” (Martin Fowler).

REFATORAÇÃO

O processo de refatoração, envolve a remoção de duplicação, a simplificação de lógica condicional e a clarificação de código que não esta claro.

O QUE NOS MOTIVA REFATORAR?

1. Tornar mais fácil a adição de código novo
2. Melhorar o projeto de código existente
3. Obter um melhor entendimento do código
4. Tornar a programação menos irritante

“Qualquer tolo consegue escrever código que um computador entenda. Bons programadores escrevem código que humanos possam entender” (Martin Fowler)

REFATORAÇÃO

A linguagem técnica de refatoração, geralmente não comunica efetivamente com a maioria das gerências. Utilize a metáfora do **débito técnico**. Débito de projeto, ocorre quando você não realiza os seguintes itens:

1. Remover duplicação
2. Simplificar seu código
3. Deixar clara a intenção do seu código

PADRÕES DE PROJETO

“Cada padrão escreve um problema que ocorre repetidamente em nosso ambiente, e então descreve o núcleo da solução para esse problema, de uma forma que você possa utilizar essa solução um milhão de vezes, sem nunca fazê-lo da mesma forma duas vezes.” (Alexander)

PROBLEMAS DE CÓDIGO

Os problemas de projeto mais comuns, resultam de código que:

1. Duplicado
2. Não é claro
3. É complicado

SOLUÇÃO

Simplificando
Expressão
Condicional



#STRATEGY

DEFINIÇÃO

Permitir de maneira simples a variação dos algoritmos utilizados na resolução de um determinado problema.

*“Uma das fontes mais
comuns de complexidade em
um programa encontra-se
na lógica condicional –
Martin Fowler”*

CENÁRIO

Nós frequentemente encontramos tal lógica em algoritmos porque eles tendem a crescer, devido as diversas novas situações do sistema.

CENÁRIO

Realização de cálculo de salários, de acordo, com o seu enquadramento funcional.

CÓDIGO - LEGADO

```
*
* @var boolean
*/
define('PSI_INTERNAL_XML', false);

if (version_compare("5.2", PHP_VERSION, ">")) {
    die("PHP 5.2 or greater is required!!!");
}
if (!extension_loaded("pcre")) {
    die("phpSysInfo requires the pcre extension to php in order to work properly.");
}

require_once APP_ROOT.'/includes/autoloader.inc.php';

// Load configuration
require_once APP_ROOT.'/config.php';

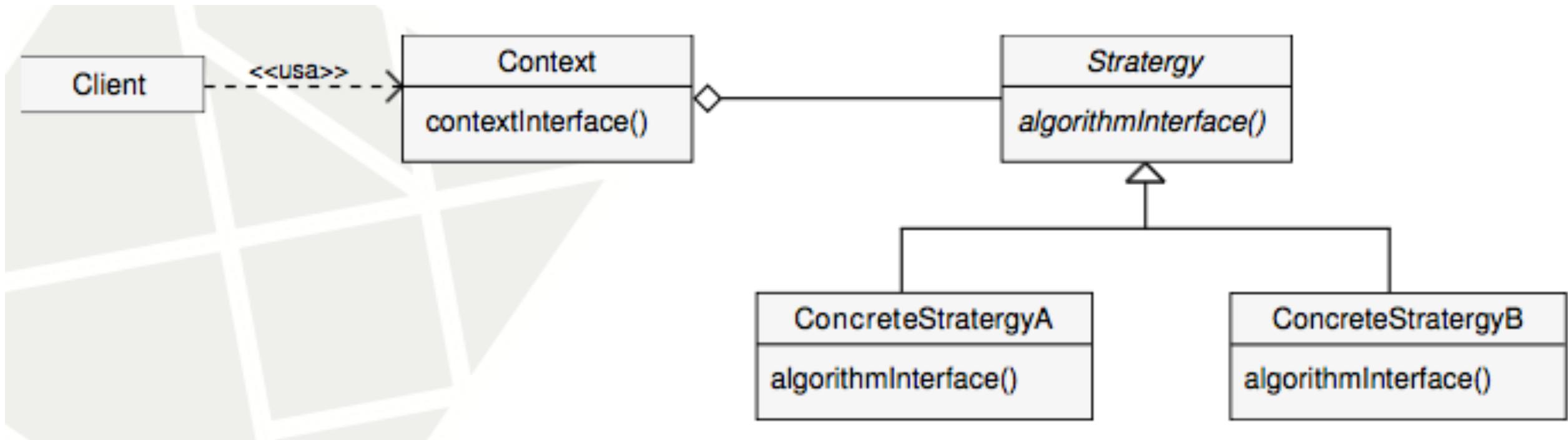
if (!defined('PSI_CONFIG_FILE') || !defined('PSI_DEBUG')) {
    $tpl = new Template("/templates/html/error_config.html");
    echo $tpl->fetch();
    die();
}

+ javascript
1. strtolower(
```

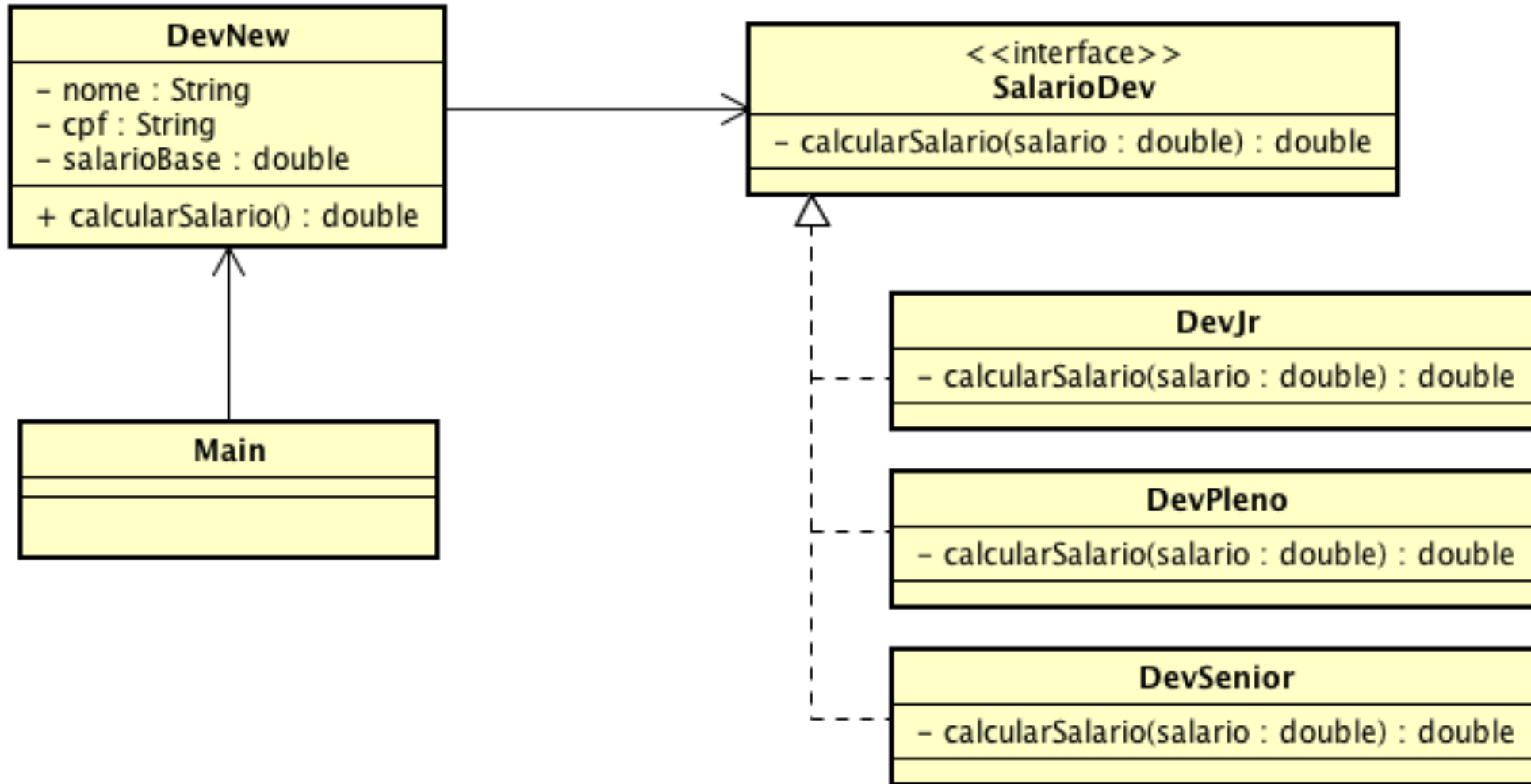

BENEFÍCIOS

- Clarifica algoritmos ao diminuir ou remover lógica condicional.
- Simplifica uma classe ao mover variações de um algoritmos para uma hierarquia.
- Habilita que um algoritmo seja substituído por outro em tempo de execução.

STRATEGY



SOLUÇÃO



CÓDIGO - SOLUÇÃO

```
*
* @var boolean
*/
define('PSI_INTERNAL_XML', false);

if (version_compare("5.2", PHP_VERSION, ">")) {
    die("PHP 5.2 or greater is required!!!");
}
if (!extension_loaded("pcre")) {
    die("phpSysInfo requires the pcre extension to php in order to work properly.");
}

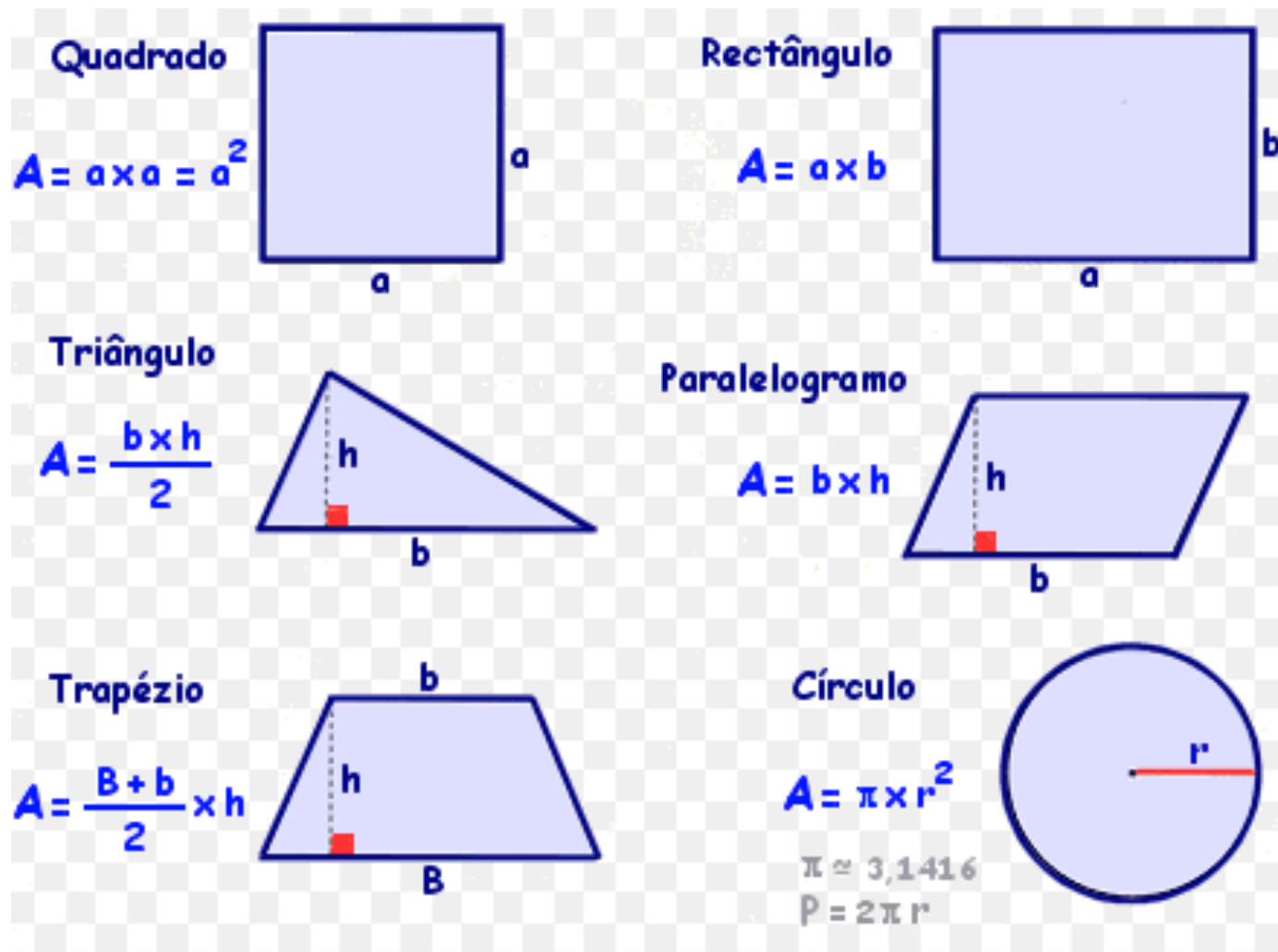
require_once APP_ROOT.'/includes/autoloader.inc.php';

// Load configuration
require_once APP_ROOT.'/config.php';

if (!defined('PSI_CONFIG_FILE') || !defined('PSI_DEBUG')) {
    $tpl = new Template("/templates/html/error_config.html");
    echo $tpl->fetch();
    die();
}

+ javascript
1. strtolower(
```

EXERCÍCIO (STRATEGY)



SOLUÇÃO

Diminuir o tamanho das
classes e diminuir o
envio condicional

#COMMAND

DEFINIÇÃO

Controlar as chamadas a um determinado componente, modelando cada requisição como um objeto. Permitir que as operações possam ser desfeitas, enfileiradas e registradas.

CENÁRIO

Emissão para diversos modelos de impressoras.

CÓDIGO - LEGADO

```
*
* @var boolean
*/
define('PSI_INTERNAL_XML', false);

if (version_compare("5.2", PHP_VERSION, ">")) {
    die("PHP 5.2 or greater is required!!!");
}
if (!extension_loaded("pcre")) {
    die("phpSysInfo requires the pcre extension to php in order to work properly.");
}

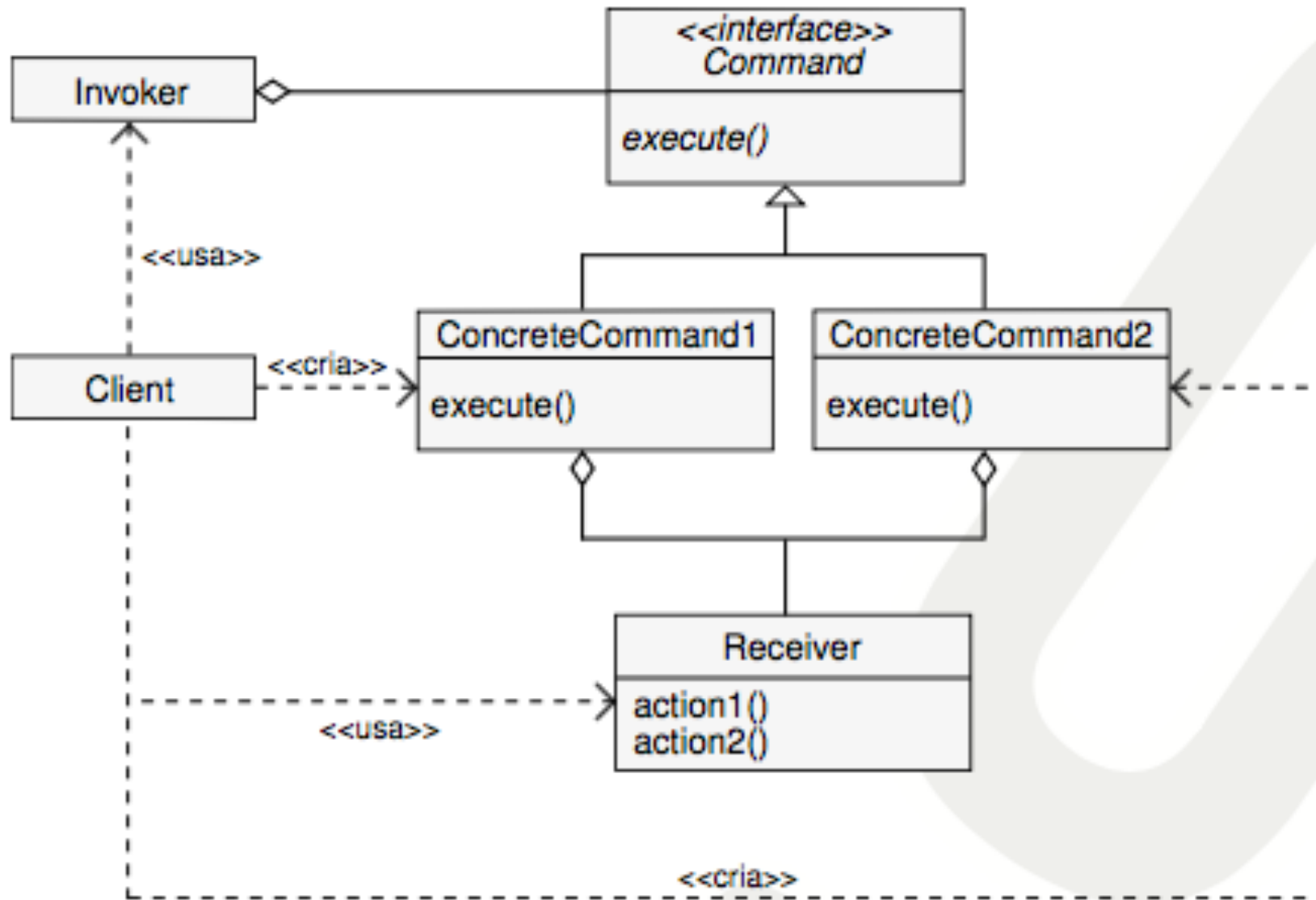
require_once APP_ROOT.'/includes/autoloader.inc.php';

// Load configuration
require_once APP_ROOT.'/config.php';

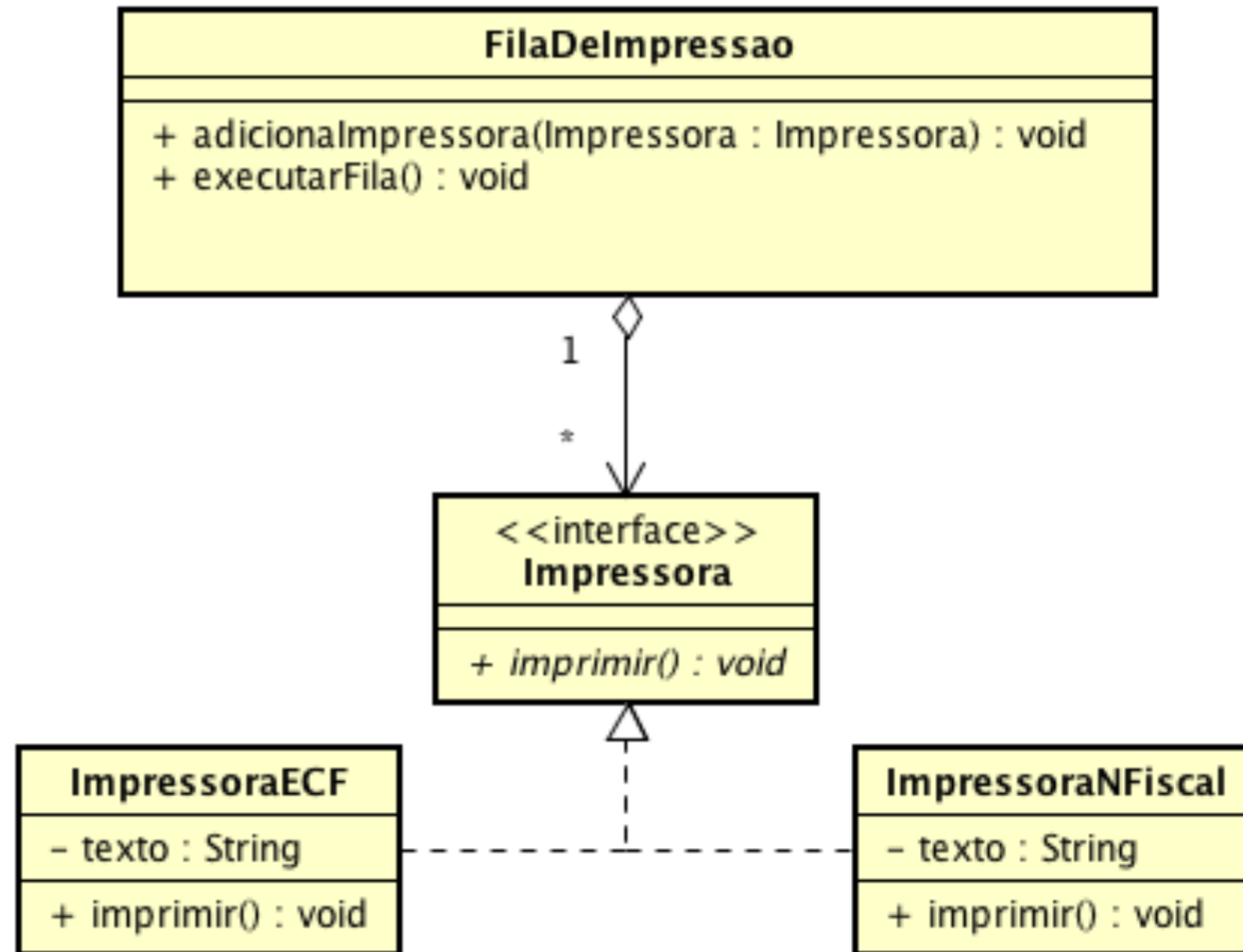
if (!defined('PSI_CONFIG_FILE') || !defined('PSI_DEBUG')) {
    $tpl = new Template("/templates/html/error_config.html");
    echo $tpl->fetch();
    die();
}

+ javascript
1. strtolower(
```

COMMAND



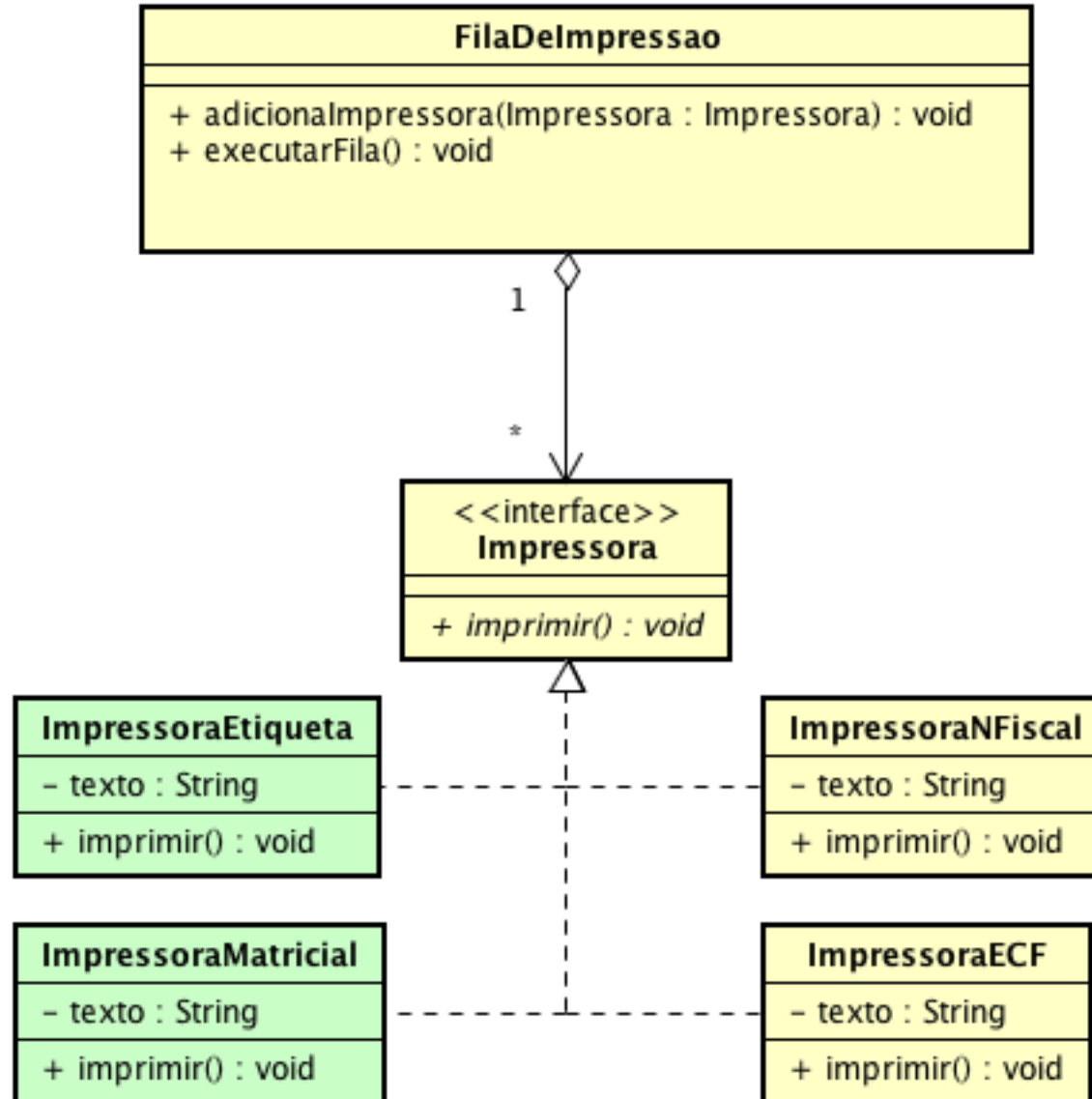
SOLUÇÃO (Envio Condicional)



CÓDIGO - SOLUÇÃO

```
*  
* @var boolean  
*/  
define('PSI_INTERNAL_XML', false);  
  
if (version_compare("5.2", PHP_VERSION, ">")) {  
    die("PHP 5.2 or greater is required!!!");  
}  
if (!extension_loaded("pcre")) {  
    die("phpSysInfo requires the pcre extension to php in order to work  
    properly.");  
}  
  
require_once APP_ROOT.'/includes/autoloader.inc.php';  
  
// Load configuration  
require_once APP_ROOT.'/config.php';  
  
if (!defined('PSI_CONFIG_FILE') || !defined('PSI_DEBUG')) {  
    $tpl = new Template("/templates/html/error_config.html");  
    echo $tpl->fetch();  
    die();  
}
```

EXERCÍCIO



SOLUÇÃO

Mover embelezamento
(constantes adaptações)

#DECORATOR

DEFINIÇÃO

Adicionar funcionalidade a um objeto dinamicamente.

CENÁRIO

Montagem de objetos
dinamicamente em tempo de
execução.

CÓDIGO - LEGADO

```
*
* @var boolean
*/
define('PSI_INTERNAL_XML', false);

if (version_compare("5.2", PHP_VERSION, ">")) {
    die("PHP 5.2 or greater is required!!!");
}
if (!extension_loaded("pcre")) {
    die("phpSysInfo requires the pcre extension to php in order to work properly.");
}

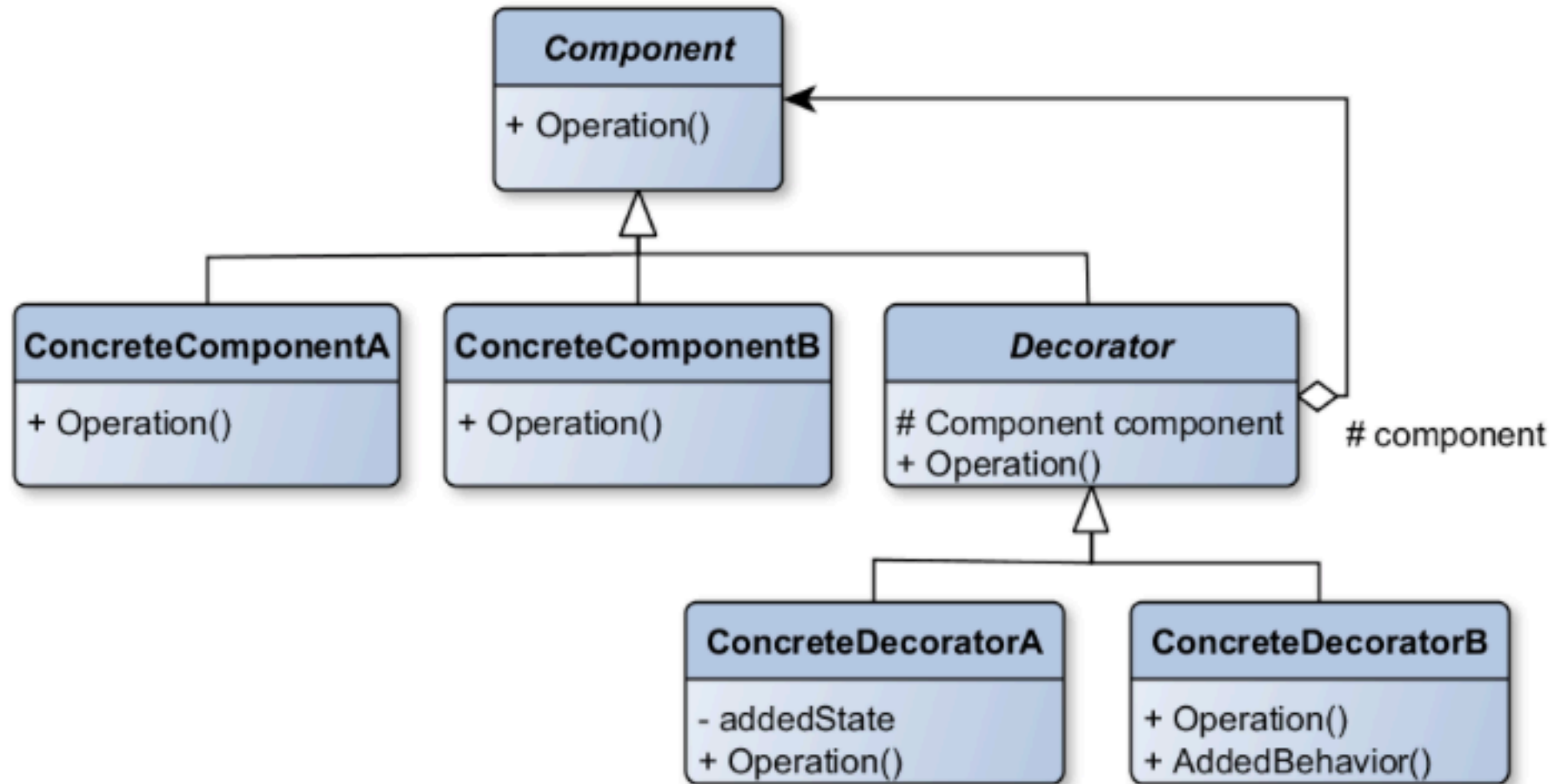
require_once APP_ROOT.'/includes/autoloader.inc.php';

// Load configuration
require_once APP_ROOT.'/config.php';

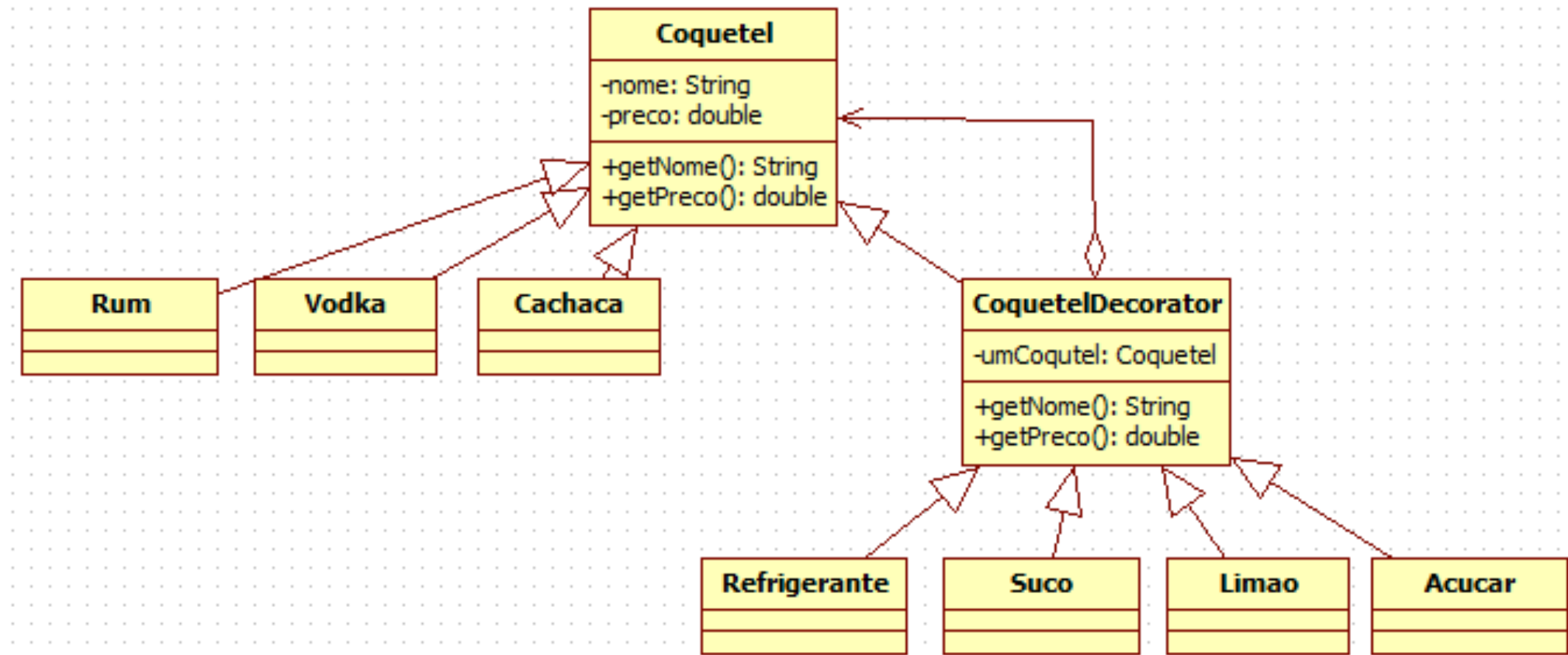
if (!defined('PSI_CONFIG_FILE') || !defined('PSI_DEBUG')) {
    $tpl = new Template("/templates/html/error_config.html");
    echo $tpl->fetch();
    die();
}

+ javascript
1. strtolower(
```

DECORATOR



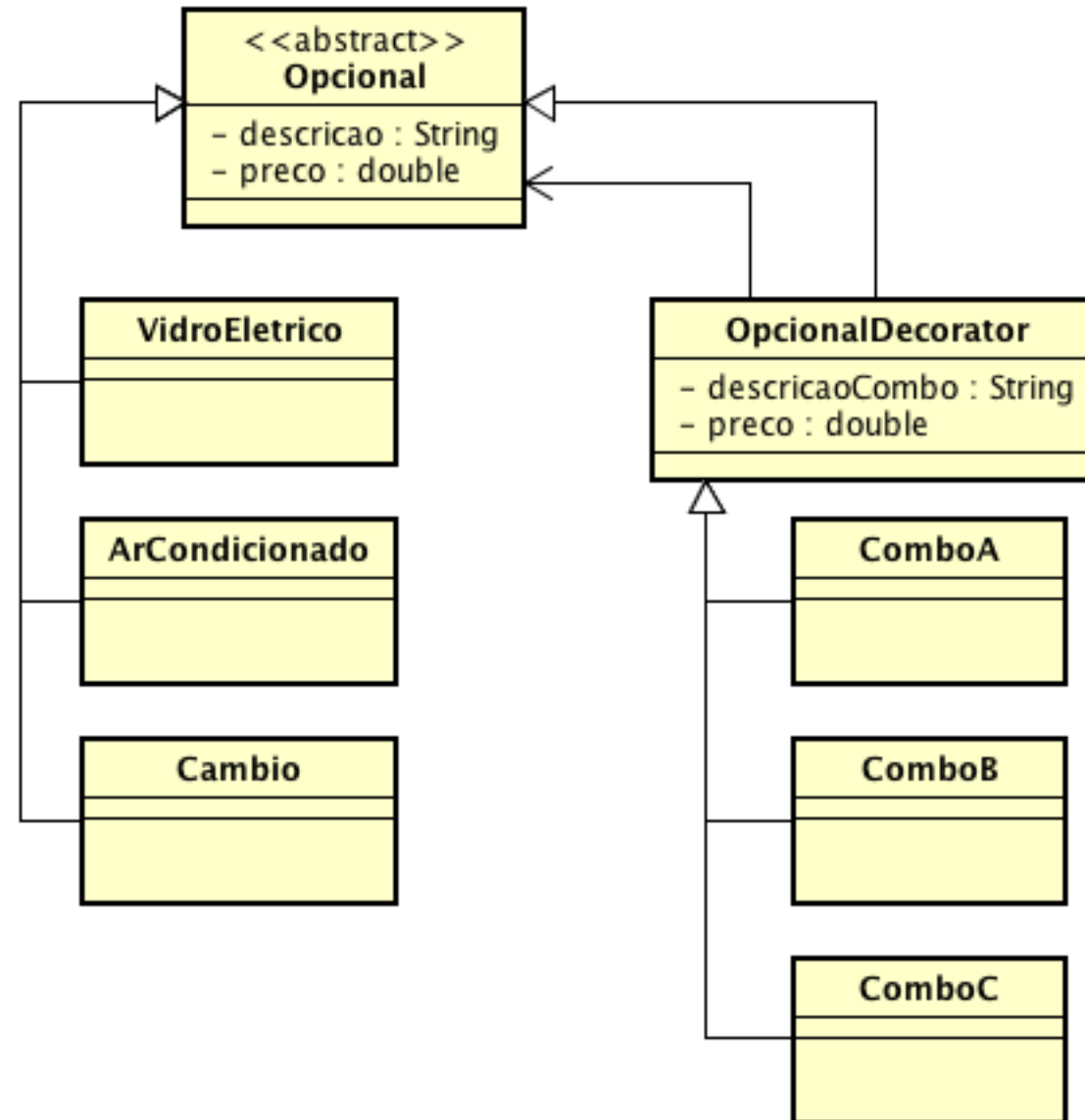
SOLUÇÃO (mover embelezamento)



CÓDIGO - SOLUÇÃO

```
*  
* @var boolean  
*/  
define('PSI_INTERNAL_XML', false);  
  
if (version_compare("5.2", PHP_VERSION, ">")) {  
    die("PHP 5.2 or greater is required!!!");  
}  
if (!extension_loaded("pcre")) {  
    die("phpSysInfo requires the pcre extension to php in order to work properly.");  
}  
  
require_once APP_ROOT.'/includes/autoloader.inc.php';  
  
// Load configuration  
require_once APP_ROOT.'/config.php';  
  
if (!defined('PSI_CONFIG_FILE') || !defined('PSI_DEBUG')) {  
    $tpl = new Template("/templates/html/error_config.html");  
    echo $tpl->fetch();  
    die();  
}
```

EXERCÍCIO (DECORATOR)



#FACTORY METHOD

DEFINIÇÃO

Encapsular a escolha da classe concreta a ser utilizada na criação de objetos de determinado tipo

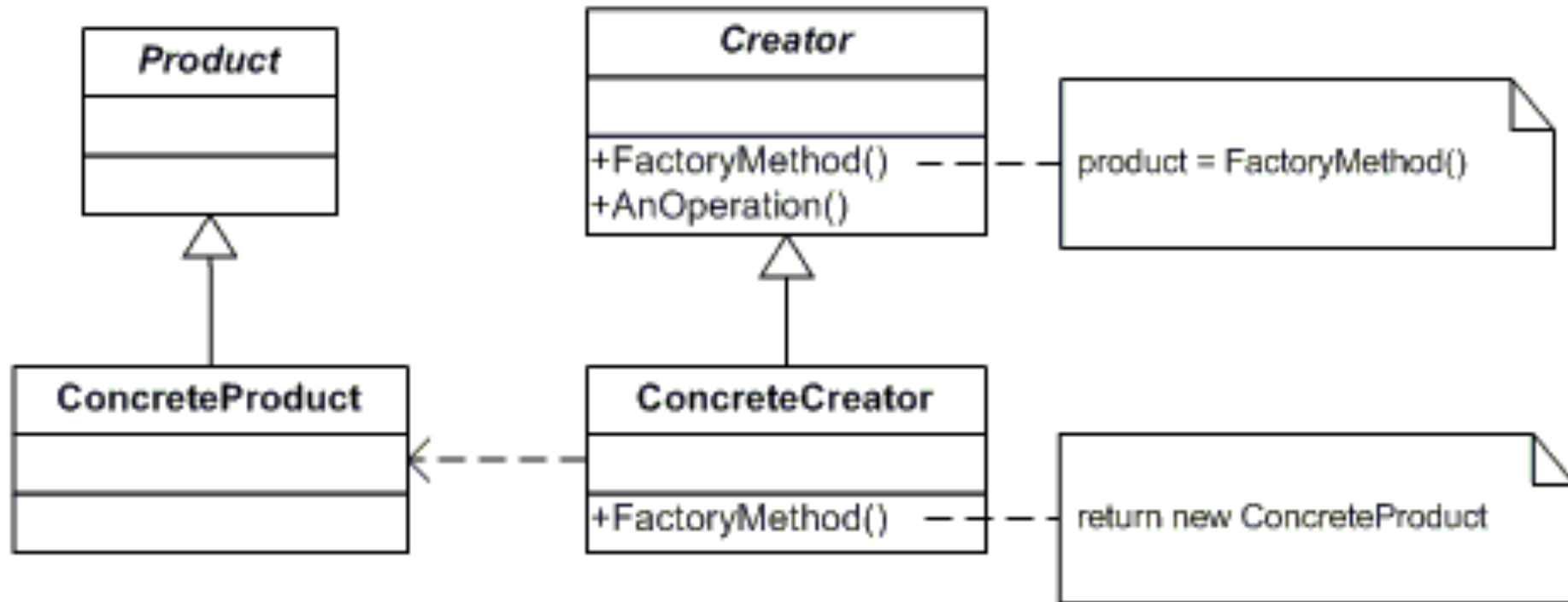
CENÁRIO

Criação de diversos tipos de arquivo (TXT, DOC, XML, etc).

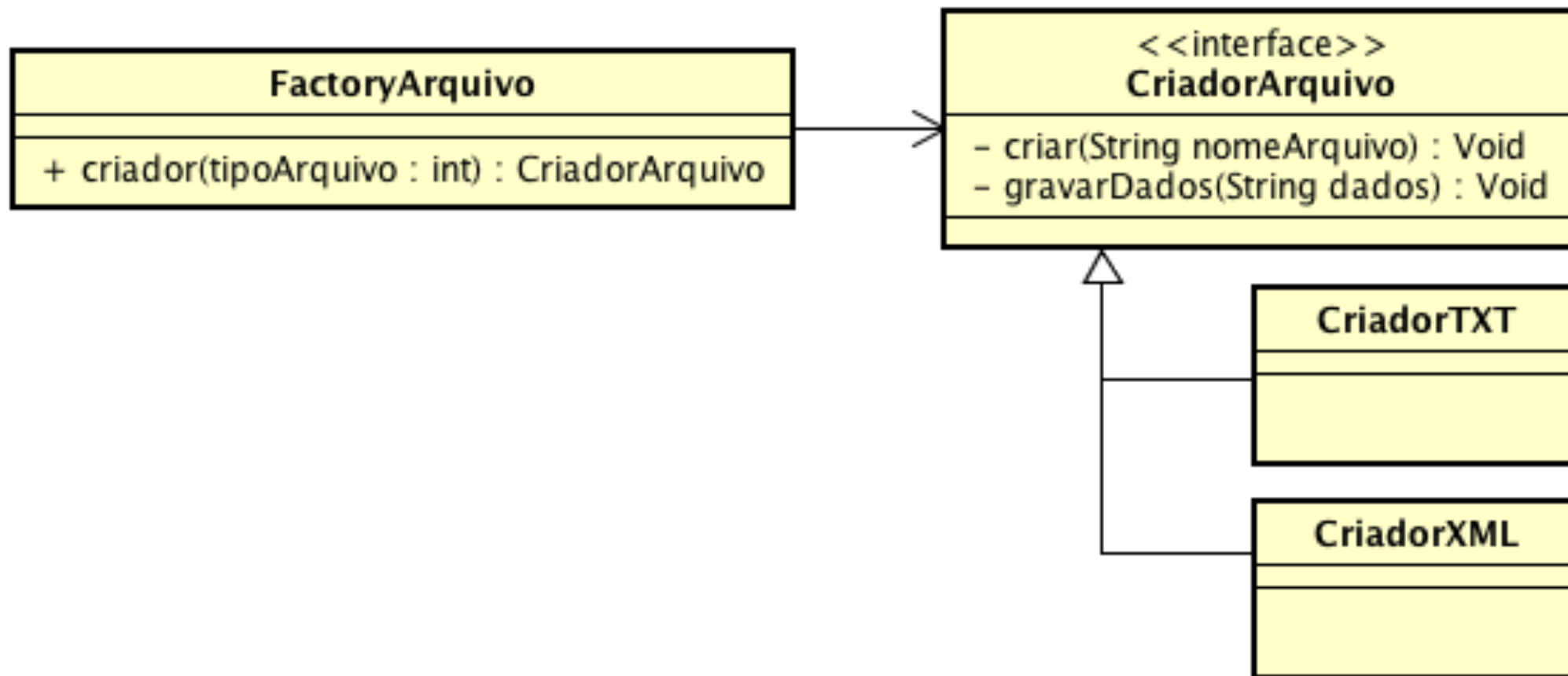
CÓDIGO - LEGADO

```
*  
* @var boolean  
*/  
define('PSI_INTERNAL_XML', false);  
  
if (version_compare("5.2", PHP_VERSION, ">")) {  
    die("PHP 5.2 or greater is required!!!");  
}  
if (!extension_loaded("pcre")) {  
    die("phpSysInfo requires the pcre extension to php in order to work  
    properly.");  
}  
  
require_once APP_ROOT.'/includes/autoloader.inc.php';  
  
// Load configuration  
require_once APP_ROOT.'/config.php';  
  
if (!defined('PSI_CONFIG_FILE') || !defined('PSI_DEBUG')) {  
    $tpl = new Template("/templates/html/error_config.html");  
    echo $tpl->fetch();  
    die();  
}
```

FACTORY METHOD



SOLUÇÃO



CÓDIGO - SOLUÇÃO

```
*
* @var boolean
*/
define('PSI_INTERNAL_XML', false);

if (version_compare("5.2", PHP_VERSION, ">")) {
    die("PHP 5.2 or greater is required!!!");
}
if (!extension_loaded("pcre")) {
    die("phpSysInfo requires the pcre extension to php in order to work properly.");
}

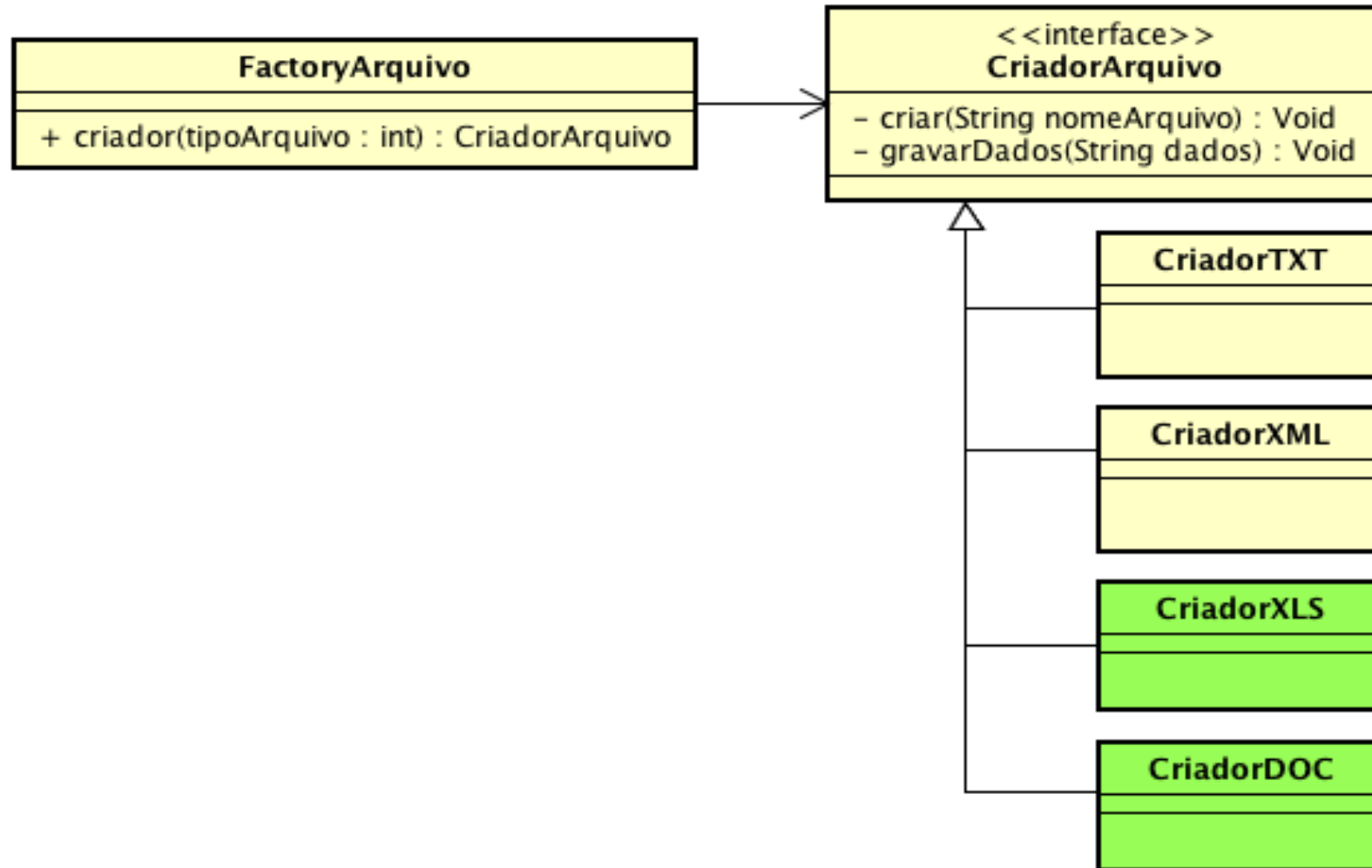
require_once APP_ROOT.'/includes/autoloader.inc.php';

// Load configuration
require_once APP_ROOT.'/config.php';

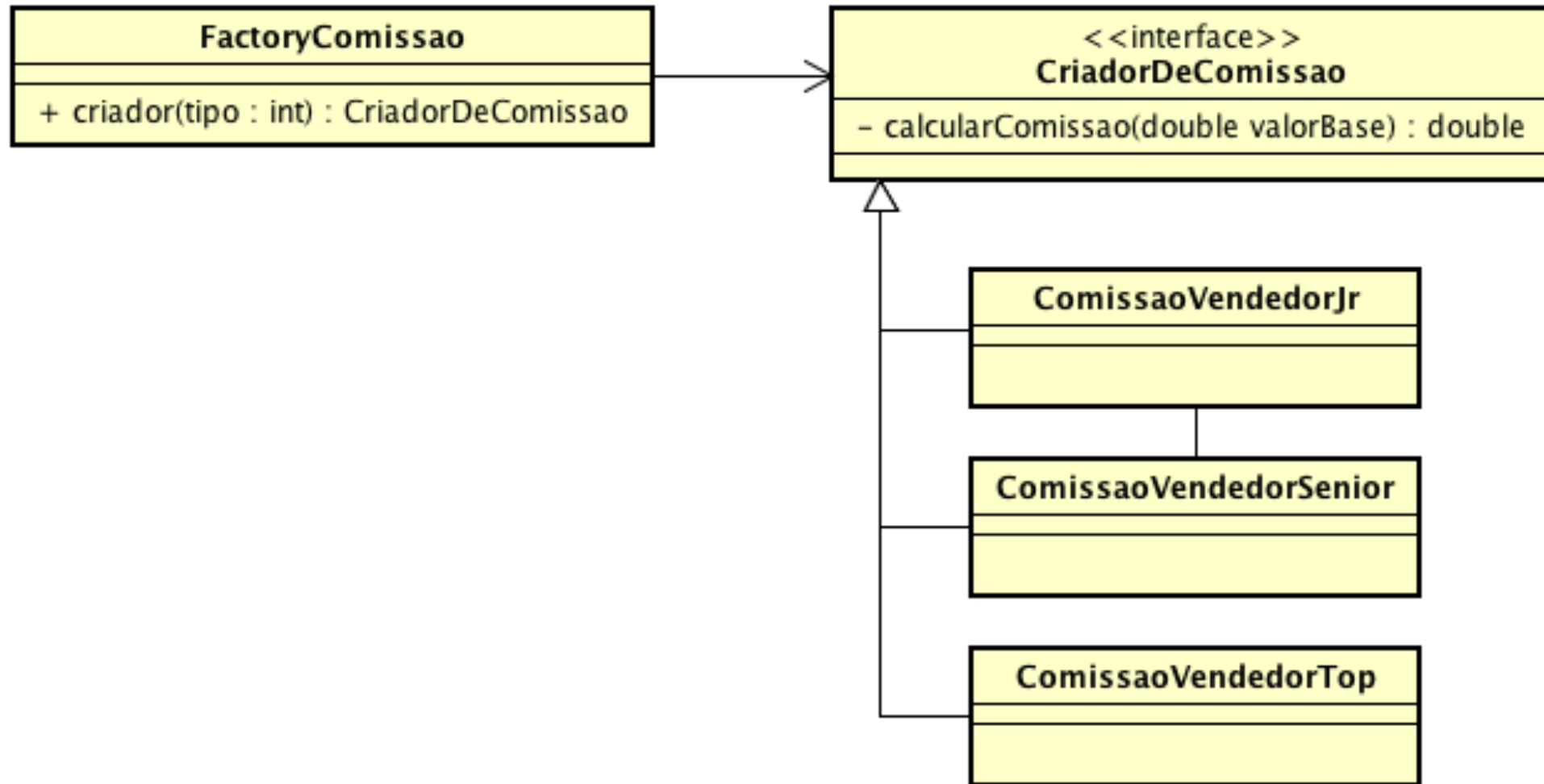
if (!defined('PSI_CONFIG_FILE') || !defined('PSI_DEBUG')) {
    $tpl = new Template("/templates/html/error_config.html");
    echo $tpl->fetch();
    die();
}

+ javascript
1. strtolower(
```

EXERCÍCIO



EXERCÍCIO



Exercício

Dentro do seu ambiente de trabalho, encontrar 4 situações que podem ser aplicados, todos os padrões de projeto, vistos nesta aula.

- Data de Entrega: Até o dia 26/Fev às 23h59m
- Pode fazer em duplas ou individual.
- Entregar o código
- Forma de Entrega do código (via **GITHUB** ou e-mail)

DICAS DE LIVRO

Padrões de Projeto

Soluções reutilizáveis de software orientado a objetos



ERICH GAMMA
RICHARD HELM
RALPH JOHNSON
JOHN VLISSIDES

Design Patterns



REFACTORING

IMPROVING THE DESIGN
OF EXISTING CODE

MARTIN FOWLER

With Contributions by Kent Beck, John Brant,
William Opdyke, and Don Roberts

Foreword by Erich Gamma
Object Technology International Inc.



REFATORAÇÃO PARA PADRÕES

JOSHUA KERIEVSKY



Apresentações de Ralph Johnson e Martin Fowler
Prefácio de John Brant e Don Roberts



rsjrnet@gmail.com



APOIADOR