

# Ferramenta de controle de acesso baseado em gamificação

Weslei de Paula Pinto<sup>1</sup>, Eduardo Guerra<sup>1</sup>, Gerson da Penha Neto<sup>1</sup>, Phyllipe Lima<sup>1</sup>

<sup>1</sup>Instituto Nacional de Pesquisas Espaciais (INPE)  
Av. dos Astronautas, 1758, Jardim da Granja, 12227-010  
São José dos Campos – SP – Brasil

weslei.paula@fatec.sp.gov.br, guerraem@gmail.com  
gerson.penha@fatec.sp.gov.br, phyllipe\_slf@yahoo.com.br

**Abstract.** *Applications using gamification unlock new features as users gain achievements or reach a number of points. However, integrating this concept to access control may become complex, and the implementation tends to occur inside the methods of business classes, increasing the coupling. This paper presents a tool that integrates the access control functionalities of the framework Guardian to the gamification mechanism offered by the Esfinge Gamification framework. To validate the tool, we performed a case study developing an online forum application that uses access control combined with gamification mechanism.*

**Resumo.** *Aplicações que usam gamificação desbloqueiam novas funcionalidades conforme o usuário obtém conquistas ou atinge certa pontuação. Porém, fazer o controle do acesso integrado a estes conceitos se torna uma atividade complexa, e sua implementação tende a acontecer isoladamente nos métodos das classes de negócio, aumentando o acoplamento entre as regras de negócio e de controle de acesso. Este artigo apresenta uma ferramenta que integra as funcionalidades de controle de acesso oferecidas pelo framework Guardian aos mecanismos de gamificação oferecido pelo framework Esfinge Gamification. A avaliação da ferramenta foi feita com um estudo de caso de uma aplicação web de fórum, onde são implementadas restrições de acesso baseadas no mecanismo de gamificação.*

**Video:** [https://youtu.be/Ab\\_EeHCvLQ](https://youtu.be/Ab_EeHCvLQ)

## 1. Introdução

A gamificação é a utilização de técnicas de *design* de jogos para melhorar a experiência de pessoas em outras áreas, através de desafios e bonificações [Robson et al. 2015]. Em sistemas de *software*, é comum que a utilização de gamificação libere acesso a novas funcionalidades a medida que o usuário obtém conquistas, o que é reconhecido na área da Psicologia Comportamental como reforço positivo [Skinner 1990]. Porém, o controle de acesso pelas informações de gamificação dificilmente se integra ao mecanismo de autorização da arquitetura e acaba sendo implementado de forma individual dentro dos métodos de negócio, agregando complexidade e responsabilidades adicionais à estes métodos.

Esse trabalho apresenta uma ferramenta denominada Esfinge GameGuardian, para a integração dos *frameworks* Guardian [Silva et al. 2013], que implementa um modelo extensível para controle de acesso, e Esfinge Gamification [Guerra et al. 2017], que possui funcionalidades para o controle de um mecanismo de gamificação. Ambos os *frameworks*

são baseados em metadados [Guerra et al. 2013], suportam a linguagem de programação Java e utilizam anotações para desacoplar esses interesses das classes da aplicação. A ferramenta apresentada nesse artigo segue essa mesma linha, propondo um conjunto de anotações para adicionar controle de acesso baseado nas informações de gamificação do Esfinge Gamification, sem alterações nas funcionalidades já existentes nos *frameworks*, ou seja, ainda é possível utilizar o Esfinge Gamification sem controle de acesso e o Guardian com os modelos já existentes. Para a avaliação da ferramenta, foi feito o desenvolvimento de uma aplicação que possui controle de acesso em funcionalidades, incremento de pontos e adição de *Achievements* em determinadas situações.

## 2. Visão Geral

Esta seção tem como objetivo apresentar a visão geral da ferramenta e mostrar a utilização dos *frameworks*. Para utilizar o *framework* Esfinge Gamification é necessária uma configuração inicial, assim como o encapsulamento da classe de negócio com as anotações em um *proxy* dinâmico criado pela classe `GameProxy`, apresentado na Figura 1.

```
1 // Configuracao do Esfinge Gamification
2 Game game = new GameMemoryStorage();
3 GameInvoker.getInstance().setGame(game);
4
5 // Configura usuario corrente
6 UserStorage.setUserID(user.getID());
7
8 // Proxy na classe de negocio
9 Negocios negocios = GameProxy.createProxy(new NegociosImpl());
```

Figura 1. Configuração e utilização do *framework* Esfinge Gamification

A Figura 2 apresenta um exemplo de utilização das anotações. Na linha 3, a anotação `@PointsToUser` configura que a cada vez que esse método for invocado serão adicionados para esse usuário 10 pontos do tipo “ESTRELA”. Essa é uma anotação já existente no *framework* Esfinge Gamification e deve ser adicionada na interface. Na linha 13, a anotação `@AllowPointGreaterThan` seria uma das anotações propostas nessa ferramenta, configurando que o usuário seria autorizado a executa-lo apenas se possuir mais de 100 pontos do tipo “ESTRELA”. As anotações de controle de acesso são colocadas nas implementações, ou seja, nas classes.

```
1 public interface Negocios {
2
3     @PointsToUser(name = "ESTRELA", quantity = 10)
4     void metodoNegocios();
5
6     void metodoRestrito();
7 }
8 public class NegociosImpl implements Negocio{
9
10    void metodoNegocios(){...}
11
12    @AllowPointGreaterThan(name = "ESTRELA", quantity = 100)
13    void metodoRestrito(){...}
14 }
```

Figura 2. Interface e classes com anotações de gamificação

**Tabela 1. Anotações de autorização desenvolvidas**

Anotação	Comportamento
@AllowPointGreaterThan @AllowPointLessOrEqualsThan @DenyPointLessOrEqualsThan @DenyPointGreaterThan	Estas anotações verificam se os pontos respeitam as restrições de maior ou menor igual a uma determinada quantidade definida na anotação para que a autorização seja concedida.
@AllowRanking @AllowLevel @AllowRankingAndLevel @AllowRankingOrLevel @DenyLevel @DenyRanking @DenyRankingAndLevel @DenyRankingOrLeve	As anotações de Ranking verificam as condições das propriedades ranking e level das conquistas, para permitir ou não o acesso a recursos.
@AllowTrophy, @DenyTrophy	Anotações de Trophy permitem ou não que um usuário acesse o recurso.
@AllowReward, @DenyReward	Anotações de Reward verificam se o usuário poderá ou não acessar um recurso com determinado Reward.

### 3. Funcionalidades

O Esfinge Gamification utiliza a interface `Achievement` para representar conquistas que o usuário pode receber a partir do uso do sistema. Os tipos de conquistas suportadas são pontos, troféus, recompensa e *ranking*. As anotações presentes neste *framework* permitem o gerenciamento dessas conquistas permitindo que sejam adicionadas ou removidas de usuários ligados a uma chamada de método.

A ferramenta descrita nesse artigo adicionou ao *framework* de gamificação anotações de controle de acesso que permitem a restrição de acordo com as informações do usuário referentes aos conceitos apresentados. Essas anotações estão disponíveis no pacote `net.sf.esfinge.gamification.annotation.auth` e estão apresentadas na Tabela 1.

Para que às anotações descritas tenham efeito, basta que a interface que as possui seja encapsulada em um *proxy* a partir da classe `GameProxy`, conforme apresentado na Figura 1.

### 4. Arquitetura

O ponto de entrada da arquitetura é a classe `GameProxy`, que cria *proxies* dinâmicos que encapsulam as classes da aplicação. Dessa forma, quando um dos métodos dessa classe for invocado, a funcionalidade de registrar as conquistas do usuário e verificar a autorização do acesso será realizada. A Figura 3 apresenta o código da classe `GameProxy`. A linha 8 encapsula o objeto com o *proxy* do *framework* `Guardian`. Isso faz com que a funcionalidade relativa as anotações de controle de acesso sejam executadas. A linha 15, executada quando um método do objeto encapsulado no *proxy* é invocado, chama a funcionalidade responsável por registrar as conquistas do usuário.

O *framework* possui o fluxo de funcionamento apresentado na Figura 4. Neste

```

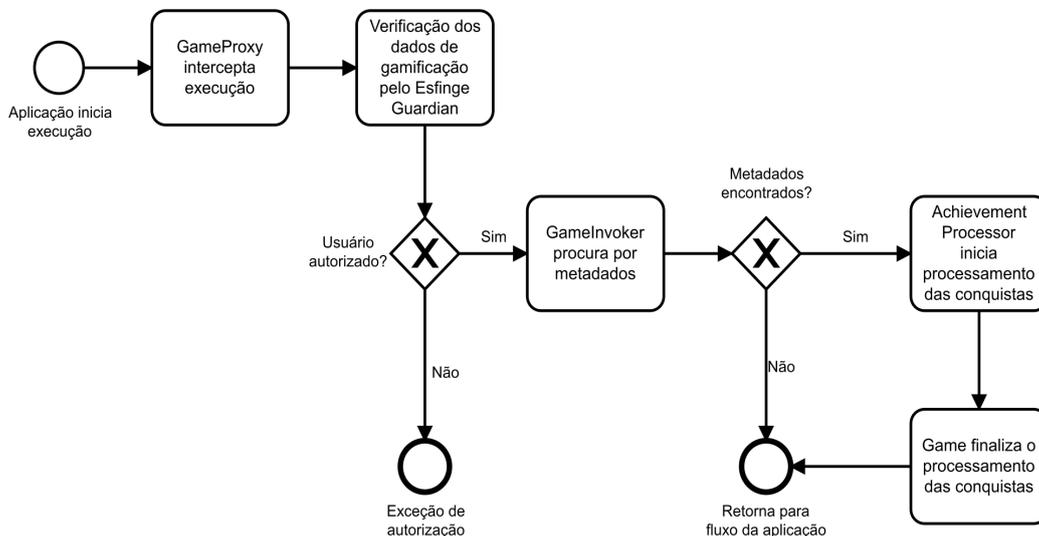
1 public class GameProxy implements InvocationHandler {
2     private Object encapsulated;
3     private Object guardedObject;
4
5     private GameProxy(Object encapsulated) {
6         this.encapsulated = encapsulated;
7         // tratativa de excecao omitida
8         this.guardedObject = AuthorizationContext.guardObject(encapsulated);
9     }
10    public Object invoke(Object proxy, Method method, Object[] args) throws Throwable {
11        try {
12            Object returnValue = method.invoke(guardedObject, args);
13            GameInvoker gameInvoker = GameInvoker.getInstance();
14            gameInvoker.registerAchievement(encapsulated, method, args);
15            return returnValue;
16        } catch (InvocationTargetException e) {
17            throw e.getTargetException();
18        }
19    }
20    //metodo createProxy() omitido
21 }

```

Figura 3. Proxy dinâmico do Esfinge Gamification com autorização

fluxo, aplicações em que o Esfinge Gamification foi configurado, são interceptadas via *Proxy* dinâmico. Antes do fluxo de gamificação ser executado, o Guardian é invocado para que validações de autorização do usuário sejam realizadas.

Figura 4. Fluxo de funcionamento atual do Esfinge Gamification



Vale ressaltar que tanto o Esfinge Gamification quanto o Guardian possuem pontos de extensão que possibilitam a criação de novas anotações. Dessa forma, é possível estender a funcionalidade da ferramenta para a criação de novas anotações com uma lógica de autorização diferente.

## 5. Avaliação

Para avaliar o uso da ferramenta desenvolvida, foi criada uma aplicação para a realização de um estudo de caso. Esta possui características de gamificação e restrições de acesso

em suas funcionalidades baseado nas conquistas do usuário. Nessa aplicação, foram testados cenários que verificam o funcionamento do *framework* em relação ao controle de acesso. A aplicação utiliza um mecanismo de *logging* é possível verificar as autorizações concedidas e negadas. Adicionalmente, a aplicação foi submetida a uma análise de acoplamento a partir do uso de uma *Dependency Structure Matrix* (DSM) [Browning 2001], no qual foi possível avaliar as dependências entre a aplicação e o *framework*.

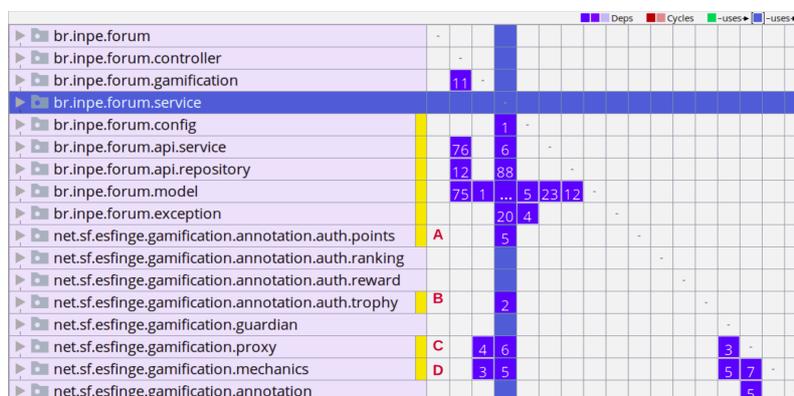
**Aplicação** : Consiste em um fórum para compartilhamento de ideias e conversas sobre assuntos gerais<sup>1</sup>. A gamificação se faz presente através de regras que fazem o usuário ganhar pontos pela adição de tópicos e respostas. Assim, novas funcionalidades são desbloqueadas com a utilização da aplicação, estimulando a interação entre usuários.

**Cenários de uso** : Foram testados cenários de uso com as seguintes características: (a) sem restrições de segurança; (b) com restrições de segurança e o usuário com as conquistas necessárias para permitir o acesso; e (c) com restrições de segurança e o usuário sem as conquistas necessárias para permitir o acesso. Foram criados cenários baseados em pontos e troféus. Todos os testes executados tiveram o comportamento esperado. A resposta do mecanismo de controle de acesso também foi verificada pelo *logging* da aplicação.

**Análise DSM** : Foi gerada a matriz DSM para observar o acoplamento envolvendo a aplicação e o Esfinge Gamification. A Figura 5 apresenta a DSM envolvendo classes da aplicação, com pacotes começados em `br.inpe.forum`, e classes do *framework*, com pacotes começados em `net.sf.esfinge.gamification`.

Conforme ressaltado na figura, o pacote `br.inpe.forum.service` depende dos pacotes com anotações de autorização a respeito de pontos e troféus (linhas A e B). Adicionalmente, serviços que acessam informações de gamificação diretamente, como para exibir as conquistas, dependem também dos pacotes “proxy” e “mechanics” do *framework* (linhas C e D). Além disso, apenas o pacote `br.inpe.forum.gamification` também depende do *framework*, nesse caso, para realizar as configurações iniciais (linhas C e D).

**Figura 5. DSM das classes da aplicação desenvolvida**



Analisando essa DSM é possível ver que a aplicação possui baixo acoplamento com o *framework* desenvolvido, sendo utilizado diretamente apenas por classes para configuração e por classes responsáveis por recuperar informações de gamificação de usuá-

<sup>1</sup>forum-tg.netlify.com

rios para exibição. Fora isso, todas as regras de segurança são definidas com as anotações, como é possível ver nas dependências. Assim, é possível dizer que existe uma separação explícita entre interesses que lidam com a gamificação e com a lógica de negócios da aplicação. Isto é um benefício ao utilizar a ferramenta, pois não é necessário realizar muitas alterações no desenvolvimento e caso a ferramenta seja modificada, atualizações não afetarão o que já foi desenvolvido devido ao baixo acoplamento entre o *framework* e as soluções que dependem deste.

## 6. Conclusão

O objetivo desse artigo é apresentar uma ferramenta que integra um mecanismo de gamificação com o componente de controle de acesso da arquitetura. Ele provê funcionalidades importantes para a gamificação de um sistema de *software*, permitindo a liberação de funcionalidades à medida que o usuário obtém conquistas no contexto da aplicação. A ferramenta utiliza anotações para configurar as restrições de acesso aos métodos.

A partir de um estudo de caso foi possível avaliar a utilização desta ferramenta. Testes em cenários de uso mostraram que ela é capaz de prover a funcionalidade almejada e uma avaliação de acoplamento mostrou que a abordagem proposta permite separar os interesses relacionados a restrições de acesso baseadas em gamificação da lógica da aplicação.

## Agradecimento

Este trabalho teve o apoio da Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), processo nº 2014/16236-6.

## Referências

- Browning, T. R. (2001). Applying the design structure matrix to system decomposition and integration problems: a review and new directions. *IEEE Transactions on Engineering management*, 48(3):292–306.
- Guerra, E., Alves, F., Kulesza, U., and Fernandes, C. (2013). A reference architecture for organizing the internal structure of metadata-based frameworks. *Journal of Systems and Software*, 86(5):1239 – 1256.
- Guerra, E. M., Fornari, G., Costa, W. S., Porto, S. M., Candia, M. P. L., and da Silva, T. S. (2017). An approach for modularizing gamification concerns. In *Computational Science and Its Applications – ICCSA 2017*, pages 635–651, Cham. Springer International Publishing.
- Robson, K., Plangger, K., Kietzmann, J. H., McCarthy, I., and Pitt, L. (2015). Is it all a game? understanding the principles of gamification. *Business Horizons*, 58(4):411–420.
- Silva, J. O., Guerra, E. M., and Fernandes, C. T. (2013). An extensible and decoupled architectural model for authorization frameworks. In Murgante, B., Misra, S., Carlini, M., Torre, C. M., Nguyen, H.-Q., Taniar, D., Apduhan, B. O., and Gervasi, O., editors, *Computational Science and Its Applications – ICCSA 2013*, pages 614–628, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Skinner, B. F. (1990). *The behavior of organisms: An experimental analysis*. BF Skinner Foundation.