

Elementos de Gamificação Aplicados no Ensino-Aprendizagem de Programação Web

Benediel Gonçalves¹, Alison Nascimento¹, Enio Monteiro¹, Carlos Portela¹,
Sandro Oliveira²

¹Faculdade de Sistemas de Informação/Universidade Federal do Pará (UFPA)

²Programa de Pós-Graduação em Ciência da Computação (ICEN/UFPA)

{bene_diel2012,elison_cn,eniojpmonteiro}@hotmail.com,{csp,srbo}@ufpa.br

Abstract. *Gamification consists of the use of game mechanics and dynamics to engage people, solve problems, and improve learning. When applied in the classroom, it allows to arouse students' curiosity and increase their motivation. This paper presents an experience report about the adoption of gamification elements in an elective course in the Information Systems at the Federal University of Pará, whose objective is to teach Web Programming. Thus, the activities of the discipline were organized in phases so that the student-player could follow a pre-established path of teaching-learning. Additionally, he scores system, medals and awards were defined. This report is described both from the point of view of teacher-facilitator (teaching) and student-player (learning).*

Resumo. *Gamificação consiste no uso de mecânicas e dinâmicas de jogos para engajar pessoas, resolver problemas e melhorar o aprendizado. Quando aplicada em sala de aula, permite despertar a curiosidade dos alunos e aumentar a sua motivação. Este artigo apresenta um relato de experiência da adoção de elementos de gamificação em uma disciplina optativa do curso de Sistemas de Informação da Universidade Federal do Pará cujo objetivo é ensinar Programação Web. Sendo assim, as atividades da disciplina foram organizadas em fases a fim de que o aluno-jogador pudesse seguir um caminho pré-estabelecido de ensino-aprendizagem. Adicionalmente, definiram-se as pontuações, medalhas e premiação. Esse relato é descrito tanto do ponto de vista do professor-facilitador (ensino) quanto dos alunos-jogadores (aprendizagem).*

1. Introdução

Cada vez mais as aplicações web estão presentes no cotidiano das pessoas e, conseqüentemente, o desenvolvimento desse tipo de aplicação representa boa parte da produção de organizações de software [Suarez et al. 2015]. Esse desenvolvimento contempla o processo de construção e testes de softwares específicos para a web, podendo variar desde simples páginas estáticas a aplicações dinâmicas, como comércios eletrônicos ou redes sociais. Cerca de 220 mil profissionais trabalham na área, conforme dados da Associação Brasileira de WebDesigners e WebMasters [ABRAWEB 2018].

No entanto, de acordo com a pesquisa *Networking Skills in Latin America*, estima-se que somente em 2015, o Brasil teve um *déficit* de 195.365 especialistas qualificados e empregados na área, número que deve cair para cerca de 161.581, em 2019 [Pineda e Gonzalez 2016]. Dada esta demanda e o *déficit* de profissionais na área, foi ofertada no primeiro semestre de 2018 (18 a 31 de janeiro) do curso de Bacharelado em Sistemas de Informação da Universidade Federal do Pará a disciplina optativa “Tópicos Especiais em Desenvolvimento para a Internet” com carga horária de 68 horas-aula. De acordo com a ementa, o objetivo é fazer com que o aluno, ao final dessa disciplina, seja capaz de

construir aplicações para web a partir do uso das principais tecnologias e técnicas empregadas na indústria.

Essa disciplina é ministrada de forma intervalar, tendo duração de apenas 3 semanas. Esse é um período muito curto para o professor abordar as principais linguagens de programação e tecnologias web da forma tradicional. Conseqüentemente, a maioria dos professores escolhe uma linguagem de programação e propõe o desenvolvimento de um site como projeto avaliativo. Isso acaba por desmotivar os alunos, pois além de ficarem limitados a uma única linguagem, não aprendem a configurar servidor, conectar com banco de dados, enviar email, dentre outras técnicas importantes na programação web.

A fim de utilizar uma metodologia de ensino que permitisse aumentar o engajamento dos alunos nesse curto período de tempo, essa disciplina adotou elementos de Gamificação [Deterding et al. 2011]. O objetivo desta adoção foi fazer com que os alunos se tornassem agentes ativos do processo de ensino-aprendizagem e buscassem aprender a configurar e implementar funcionalidades básicas em diferentes linguagens de programação web.

Além desta seção introdutória, na Seção 2 é abordada o referencial teórico deste relato de experiência, a partir da contextualização, principais conceitos e trabalhos relacionados. Já na Seção 3 são descritos os elementos de gamificação propostos para a disciplina. A Seção 4 mostra o relato de experiência do professor que ministrou a disciplina e de dois alunos que cursaram a mesma. Por fim, a Seção 5 apresenta considerações finais e as conclusões do artigo, além de destacar as próximas etapas a serem realizadas.

2. Referencial Teórico

2.1. Desenvolvimento Web

De maneira geral, as linguagens de desenvolvimento para web podem ser classificadas em duas categorias: *client-side* e *server-side* [Getting Started 2018]. Estas também podem ser identificadas por *front-end*, se referindo às linguagens *client-side*, e *back-end* se referindo às linguagens *server-side*. As *server-side* são linguagens que o servidor interpreta. Isso significa que permitem escrever um código onde o servidor vai processar e, em seguida, mandar para o seu navegador a resposta. Já as *client-side* são linguagens onde apenas o navegador vai interpretar.

De acordo com a pesquisa da Umblar (2017), o desenvolvimento *front-end* se destaca quando se trata das linguagens mais procuradas do mercado. Nessa categoria, a linguagem JavaScript representa 67% das vagas de emprego ofertadas. A linguagem *back-end* que mais tem aceitação do mercado é o PHP, presente como pré-requisito em 40% das vagas. Além destas, destacam-se no *back-end* .NET e Java e, no *front-end*, a linguagem HTML e CSS, que são requisitadas em 56% e 54% das vagas, respectivamente.

Para a disciplina, foram selecionadas 4 linguagens de programação dentre aquelas com melhores médias de remunerações das vagas ofertadas para desenvolvedores, segundo a pesquisa da Umblar (2017). Desta forma, abordou-se Java (média de R\$8309), Ruby (média de R\$7595), Python (média de R\$7350) e PHP (média de R\$4878).

2.2. Gamificação

Segundo Deterding et al. (2011), o termo Gamificação teve origem na indústria de mídia digital em 2008. Zichermann e Cunningham (2011) caracterizam gamificação como a inclusão de mecânica, estilo, pensamento e técnicas de *design* de jogos eletrônicos para envolver pessoas na solução de um problema. De maneira complementar, Vianna et al.

(2013) define esse termo como o uso de mecanismos e dinâmicas de jogos para a resolução de problemas e para a motivação e engajamento de um determinado público.

Atualmente, as técnicas de gamificação são bastante utilizadas como ferramentas de ensino [Khaled 2011]. Componentes como bonificação, *ranking*, desafios, ganho de medalhas e troféus, são alguns exemplos de gamificação utilizados em jogos digitais e que podem ser incorporados em sala de aula com o objetivo de motivar e engajar os alunos no desenvolvimento das atividades. Sugere-se o uso destas e de outros elementos de gamificação em disciplinas da área de Computação, com o intuito de motivar e aumentar a participação dos alunos nas aulas, mais especificamente em disciplinas de programação.

2.3. Trabalhos Relacionados

Figueiredo et al. (2015) apresentam um relato de experiência da aplicação de uma abordagem gamificada para o ensino de programação orientada a objetos com o objetivo de engajar e motivar os alunos na disciplina Algoritmos III. Dentre as estratégias adotadas, as medalhas foram classificadas em dois tipos: medalhas temáticas relacionadas a um conteúdo específico da disciplina, que eram conquistadas por entregar uma tarefa no prazo e medalhas de comportamento. No entanto, a forma que esta estratégia foi implementada apenas força os alunos a manterem comportamentos artificiais em sala de aula, como fazer perguntas (medalha para cada 10 perguntas) e comentários (medalha para cada 20 comentários) mesmo que não tenham uma dúvida ou contribuição real.

Já Campos, Gardiman e Madeira (2015) relatam o uso de uma ferramenta gamificada, denominada Kodesh, em uma disciplina introdutória de programação. Os resultados dessa aplicação demonstraram um alto nível de aceitação por parte de professores e alunos, resultando em índices de aceitação entre 81% e 83% para os elementos de gamificação. Apesar dos resultados positivos, a ferramenta não foca no ensino de programação web e nem se encontra disponível para *download*.

Diferentemente destas abordagens, optou-se por especificar elementos de gamificação diretamente relacionadas com o processo de ensino-aprendizagem de programação web, a fim de explorar o potencial desta abordagem. Desta forma, incorporou-se o histórico das linguagens, as etapas de configuração dos servidores e níveis de dificuldade de aprendizagem nas fases, medalhas, desafios e premiação da disciplina.

3. Gamificação da Disciplina

3.1. Elementos Adotados

Na gamificação, os elementos podem ser divididos conforme seu nível de abstração entre dinâmicas, mecânicas e componentes, conforme ilustra a Figura 1.

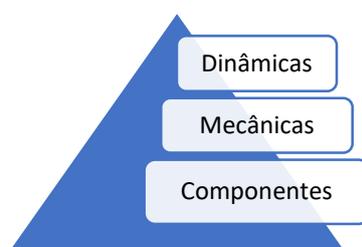


Figura 1. Elementos de Jogos de Gamificação.

No nível mais alto de abstração encontram-se as dinâmicas, que são aspectos controlados pela gamificação, mas obtidos pelos demais elementos [Werbach e Hunter 2016]. As dinâmicas adotadas para a disciplina foram:

- **Progresso** que fornece o *feedback* da evolução do usuário no decorrer do tempo;
- **Regras** que são características impostas (limitações) aos usuários;
- **Relacionamentos** que são as iterações sociais que o usuário tem com outros durante as atividades, como por exemplo, com membros de equipe e adversários.

As mecânicas estão no nível intermediário de abstração, sendo processos que estimulam a ação e o engajamento do usuário e são alcançados através dos componentes [Werbach e Hunter 2016]. As mecânicas adotadas na disciplina foram:

- **Competições e Cooperações** que promovem a interação entre os usuários. Enquanto na competição os alunos disputam contra outros, eles trabalham em equipe na cooperação para atingir uma meta em comum;
- **Desafios** que são atividades que exigem esforço para serem resolvidas e direcionam os usuários ao principal objetivo da gamificação, como por exemplo, aprender uma nova linguagem de programação web;
- **Feedback** que retorna informações relevantes ao usuário sobre a gamificação a fim de gerar engajamento e motivação;
- **Recompensas** que são benefícios oferecidos a fim de reconhecer o esforço.

Na base da gamificação estão os componentes, que são instâncias específicas das mecânicas e dinâmicas no sistema [Werbach e Hunter 2016]. Os componentes incorporados na disciplina foram:

- **Emblemas** que representam as conquistas do usuário, sendo concedidos ao alcançar alguma meta, se apresentando nesta disciplina na forma de medalhas;
- **Missões** que são metas pré-definidas que possuem recompensas específicas;
- **Pontos** que representam o progresso do usuário em relação às missões propostas;
- **Níveis ou Fases** que são identificadores do progresso do usuário ao longo do tempo, baseado nas missões cumpridas;
- **Tabelas de Classificação ou Ranking** que exibem o progresso do usuário, dando significado a outros componentes e contextualizando as pontuações.

3.2. Fases e Pontuação

As atividades da disciplina foram organizadas em fases a fim de que o aluno-jogador seguisse um caminho pré-estabelecido de ensino-aprendizagem. Assim, cada fase possuía um grau de dificuldade e pontuação de acordo com a ementa e objetivos da disciplina. Neste sentido, as fases, pontuações e objetivos são descritos na Tabela 1.

Tabela 1. Fases e Pontuação da Disciplina

Etapa	Pontuação	Objetivo
Fase 1	1,0 ponto	Criar um "hello world!" através de <i>request-response</i> no cliente-servidor.
Fase 2	2,0 pontos	Criar um CRUD (<i>Create, Retrieve, Update e Delete</i>) de uma entidade com persistência no banco de dados.
Fase 3	3,0 pontos	Gerar uma classe para gerenciar o envio de e-mail usando o protocolo SMTP (<i>Simple Mail Transfer Protocol</i>).
Fase 4	4,0 pontos	Aplicar no mínimo 4 (quatro) <i>Web Patterns</i> no site desenvolvido durante a disciplina.
Fase Bônus	1,0 ponto	Aplicar máscaras e realizar a validação dos campos dos formulários do site desenvolvido.
Final Boss	10,0 pontos	Criar um site em HTML 5 que implemente os requisitos de todas as fases e que seja totalmente responsivo.

Assim, para a Fase 1 os alunos deveriam criar uma página *no front-end* que possuísse um botão. Esse botão deveria disparar uma requisição ao servidor. A página do *back-end* deveria responder à requisição com a mensagem "Hello World!". O "Olá Mundo" é um famoso algoritmo de programação que usualmente imprime "Olá Mundo!" (ou "Hello World!"), com algumas variações como inexistência do ponto de exclamação e letras em minúscula, no dispositivo de saída. É utilizado como um teste ou como um exemplo de código minimalista para se aprender a sintaxe de uma linguagem de programação. Na apresentação, os alunos deveriam mostrar o passo-a-passo da configuração do servidor e os códigos-fonte do *front-end* e *back-end*.

Na Fase 2, os alunos deveriam criar um formulário *no front-end* que possuísse os campos solicitados pelo cliente (professor). Esse formulário deveria permitir que o usuário inserisse informações e clicasse no botão "Enviar". Então, esse botão deve disparar uma requisição ao servidor para que estas informações fossem salvas no banco de dados do *back-end*. Após a persistência dos dados, o site deveria permitir que os usuários realizassem o restante das operações do CRUD, acrônimo em inglês de *Create* (Criação), *Retrieve* (Consulta), *Update* (Atualização) e *Delete* (Destrução). Este acrônimo é comumente utilizado para definir as quatro operações básicas usadas em Banco de Dados Relacionais. Na apresentação, os alunos deveriam mostrar o banco de dados e os códigos-fonte do *front-end* e *back-end*.

Em seguida, na Fase 3, os alunos tinham que criar um formulário *no front-end* que possuísse os campos solicitados pelo cliente (professor). Esse formulário deveria permitir que o usuário inserisse informações e clicasse no botão "Enviar E-mail". Esse botão deveria disparar uma requisição ao servidor para fazer uso do protocolo SMTP (*Simple Mail Transfer Protocol*) no *back-end*. Após o envio do e-mail, o site deveria fornecer um *feedback* ao usuário sobre a operação: "E-mail enviado com sucesso!" ou "Falha no envio do E-mail!". O SMTP é um protocolo relativamente simples, em texto plano, onde um ou vários destinatários de uma mensagem são especificados sendo, posteriormente, a mensagem transferida. Durante a apresentação, os alunos deveriam apresentar o código-fonte da página do *front-end* e as classes e métodos do *back-end*.

Por fim, na Fase 4, os alunos deveriam aplicar no mínimo 4 (quatro) *Web Patterns* no site que estava sendo desenvolvido. *Web Patterns* consistem em padrões para criação e renderização de páginas web, incluindo HTML, CSS, SVG, APIs de dispositivos e outras tecnologias para aplicativos da web ("WebApps"). De maneira geral, objetivam tornar as páginas acessíveis a pessoas com deficiência (WCAG), internacionalizá-las e fazê-las funcionar em dispositivos móveis [W3C 2018]. Durante a apresentação, os alunos deveriam apresentar o código-fonte referente aos padrões aplicados no *front-end* e, se fosse o caso, os métodos usados no *back-end*.

Adicionalmente, a Fase Bônus consistia em inserir máscara de entrada e realizar a validação de dados nos formulários do site. Quando se trata de validar valores de campos de formulários, isso pode ser feito tanto no lado do cliente (navegador), através de máscaras de entrada, quanto no lado do servidor (usando uma linguagem específica). Uma alternativa a todas essas 4 fases era o *Final Boss* (Chefão Final), onde o aluno deveria desenvolver um site responsivo em HTML5 que contemplasse todos os requisitos das fases 2, 3, 4 e Bônus.

As atividades da disciplina foram realizadas em grupos. Assim, foram formadas 4 equipes de 4 alunos (16 participantes). Para evitar empates e a falta de comprometimento, ao final da disciplina, os alunos realizaram uma Avaliação 360°, ou seja, avaliaram o professor da disciplina, seus colegas de equipe e a si próprio. Desta forma, buscou-se que todos participassem ativamente das atividades propostas.

3.3. Desafios e Estratégia

O desafio da disciplina consistiu em realizar cada fase utilizando uma linguagem de programação web diferente. Estabeleceu-se esse desafio a fim de que os alunos pudessem explorar não apenas uma linguagem de programação web, como normalmente acontece nesse tipo de disciplina, e sim aprender a configurar e conhecer a sintaxe básica das principais linguagens do mercado.

Caso repetisse uma linguagem, seria subtraído 1 ponto da pontuação da fase. Assim, se um aluno realizar todas as fases na mesma linguagem, poderia obter no máximo 7,0 pontos. Desta forma, os alunos deveriam criar uma estratégia na hora de escolher qual linguagem usariam em cada fase.

No entanto, além das 4 fases, havia uma Fase Bônus que valia 1 ponto extra, podendo auxiliar na estratégia do aluno. Por fim, caso desejasse, o aluno poderia optar por enfrentar o Chefão Final (*Final Boss*) diretamente, usando apenas uma linguagem, ao invés de cumprir essas 4 fases.

3.4. Medalhas

A conclusão de cada fase dava direito a uma medalha, para os membros da equipe, correspondente ao grau de dificuldade da mesma. Adicionalmente, o uso com sucesso de uma linguagem dava direito a uma medalha referente ao criador desta linguagem.

Houve ainda uma medalha específica para a equipe que matasse o *Final Boss* e uma especial para aquela que concluísse todas as fases. Os alunos solicitaram a adição da medalha “Mochila”, para aqueles colegas que são “carregados” pelos demais da equipe.

A Tabela 2 apresenta as medalhas criadas para a gamificação da disciplina.

Tabela 2. Medalhas da Disciplina

Medalha	Descrição
 Noob	Medalha adquirida pela equipe que concluísse a Fase 1. Mensagem: Parabéns, vocês são uns <i>Noobs</i> que conseguem fazer um "hello world!".
 Nerd	Medalha adquirida pela equipe que concluísse a Fase 2. Mensagem: Parabéns, vocês são uns <i>Nerds</i> que conseguem fazer um CRUD usando uma Linguagem Web.
 Developer	Medalha adquirida pela equipe que concluísse a Fase 3. Mensagem: Parabéns, vocês são uns <i>Developers</i> que conseguem fazer envio de e-mail usando SMTP.
 Hacker	Medalha adquirida pela equipe que concluísse a Fase 4. Mensagem: Parabéns, vocês são uns <i>Hackers</i> que sabem usar <i>Web Patterns</i> no desenvolvimento dos seus sites.
 Ninja	Medalha adquirida pela equipe que matasse o <i>Final Boss</i> . Mensagem: Parabéns, vocês são uns Ninjas da Web que mataram o <i>Final Boss</i> .

 James Gosling	Medalha adquirida pela equipe que concluiu uma das fases utilizando Java. Mensagem: James Gosling está orgulhoso de vocês. Ele fez o projeto original da linguagem Java e implementou o seu primeiro compilador e máquina virtual.
 Rasmus Lerdorf	Medalha adquirida pela equipe que concluiu uma das fases utilizando PHP. Mensagem: Rasmus Lerdorf está orgulhoso de vocês. Ele é o autor da primeira versão da linguagem de programação PHP.
 Yukihiro Matsumoto	Medalha adquirida pela equipe que concluiu uma das fases utilizando Ruby. Mensagem: Yukihiro Matsumoto está orgulhoso de vocês. Matz é o idealizador/criador da linguagem de programação Ruby.
 Guido van Rossum	Medalha adquirida pela equipe que concluiu uma das fases utilizando Python. Mensagem: Guido van Rossum está orgulhoso de vocês. Ele é o autor da linguagem de programação Python.
 Ada Lovelace	Medalha para equipe que concluiu todos os desafios. Mensagem: Ada Lovelace estaria orgulhosa de vocês. Ela foi a primeira programadora da História, tendo criado o primeiro algoritmo para a máquina analítica de Charles Babbage.
 Mochila	Medalha individual adquirida pelo aluno que obtivesse a menor nota na avaliação 360°. Mensagem: De acordo com a sua equipe, você foi carregado nas costas durante toda a disciplina. Você concorda?

3.4. Ranking e Premiação

O *ranking* da disciplina foi gerado a partir da nota da pontuação + classificação obtida em cada fase. Por exemplo, se existissem 5 equipes, a que melhor concluiu a Fase 1 poderia obter 1,0 ponto + 0,5 pela classificação. A que ficasse em 2º na fase teria somado a pontuação + 0,4 pela classificação, a que ficasse em 3º + 0,3 e assim sucessivamente. Ao final da disciplina, quem estivesse em 1º lugar no *ranking* ganhava um prêmio.

No caso específico desta disciplina, o prêmio foi um curso *online* completo de Desenvolvimento Web, com carga horária de 34h, onde o 1º colocado poderia aprender a desenvolver sites e sistemas web utilizando HTML, CSS, Bootstrap, JavaScript, JQuery, PHP, POO e MySQL.

4. Relato de Experiência

4.1. Professor-Facilitador (Ensino)

Inicialmente, o professor analisou a ementa da disciplina e selecionou as 4 linguagens de programação web de acordo com sua importância para o mercado em termos de uso e remuneração profissional. A fim de que os alunos pudessem experimentar essas linguagens, o professor estabeleceu as 4 fases e a regra de que cada uma deveria ser completada com uma linguagem diferente. Buscando contemplar os tópicos da ementa, definiu um percurso gradativo de dificuldade, partindo do nível fácil (Fase 1 - 1,0 ponto) até o mais difícil (Fase 4 - 4,0 pontos), refletindo na pontuação das atividades.

De acordo com o professor, foi a primeira vez que o mesmo ministrou a disciplina de “Tópicos Especiais em Desenvolvimento para a Internet”. No entanto, já possui experiência acadêmica com a linguagem Java e Ruby e profissional com PHP. Quanto à Python, não havia tido nenhum contato antes da disciplina. Desta forma, atuou como um facilitador no processo de ensino-aprendizagem, fornecendo uma determinada quantidade de orientação e apoio às equipes. Esse apoio buscou viabilizar a aprendizagem sem gerar forte influência nas atividades, ou seja, não deveria desviar os alunos de encontrar por si mesmos o “caminho certo”. Por exemplo, na Fase 3, o professor apresentou as bibliotecas SMTP para cada linguagem, mas não tratou do *front-end* da página.

Quanto às medalhas, o professor aproveitou esse componente de gamificação para ensinar sobre o histórico das linguagens, homenageando os criadores das mesmas. O mesmo pode observar que estas foram uma motivação a mais para os alunos, gerando uma competição para ver quem conseguia mais medalhas. Adicionalmente, o professor atendeu à requisição dos alunos de criar a medalha “Mochila” (uma brincadeira interna da turma). Isso fez com que a turma se divertisse, aumentando sua imersão no desenvolvimento das atividades e, conseqüentemente, a competitividade, pois ninguém queria essa medalha. No entanto, por ser uma brincadeira da turma, o professor destaca que sua adoção é opcional.

O *ranking* era atualizado ao final da entrega (apresentação) de cada fase, através do site <http://carlosportela.com.br/tedi>. Assim, os alunos podiam acompanhar a sua classificação. Quanto aos resultados das fases, após cada apresentação, o professor atribuía uma nota (de acordo com a pontuação máxima de cada fase) e definia uma classificação de acordo com a qualidade do código-fonte entregue e da apresentação da equipe. Os resultados de cada fase e a classificação final por equipe são apresentados na Tabela 3.

Tabela 3. Resultados das Equipes por Fases.

#	Fase 1	Fase 2	Fase 3	Fase 4	FINAL	Pontuação
1°	Equipe D	Equipe C	Equipe C	Equipe A	Equipe C	13pts
2°	Equipe C	Equipe A	Equipe A	Equipe D	Equipe A	12pts
3°	Equipe A	Equipe D	Equipe B	Equipe C	Equipe D	10pts
4°	Equipe B	Equipe B	Equipe D	Equipe B	Equipe B	5pts

Nenhum aluno fez a Fase Bônus ou enfrentou o *Final Boss* (ambas deveriam ser feitas individualmente), pois preferiram trabalhar em equipe. Como as atividades foram realizadas em grupo, o desempate entre os membros das equipes aconteceu após a avaliação 360° da turma. Nesta avaliação, os alunos tinham que responder, de 0 a 5, qual foi a contribuição do membro da equipe durante as fases. Assim, obteve-se o percentual de participação do aluno nas atividades, que foi multiplicado pela nota final da equipe. Neste sentido, o professor, tendo ministrado outras disciplinas para a turma, observou o aumento no engajamento dos alunos e a diminuição na quantidade de alunos “mochilas”.

4.2. Aluno-Jogador (Aprendizagem)

Em relação às expectativas para a disciplina, os alunos relataram que pensaram, a princípio, que a disciplina adotaria uma abordagem tradicional de ensino, ou seja, uma metodologia na qual o professor ensinaria o básico de uma única linguagem de programação web. Quanto à avaliação, esperavam que esta fosse por meio de uma prova e/ou desenvolvimento de um sistema web específico.

De acordo com os alunos, o processo de ter que pesquisar o conteúdo foi desafiador a todos da turma, pois estavam acostumados com o professor ensinando em sala de aula.

Citaram que a abordagem tradicional normalmente desperta o interesse de alguns (aqueles que possuem maior inclinação pelo assunto abordado), entretanto, outros acabam se acomodando ou não mostrando o devido interesse pela disciplina. Nessa experiência com gamificação, observaram que todos assumiram uma postura proativa. Destacaram que o desafio se mostrou ainda maior, pois a disciplina abordou 4 linguagens de programação, exigindo empenho e dedicação de cada membro da equipe, uma vez que, algumas dessas linguagens eram até então desconhecidas da maioria. Ressaltaram que o professor sempre contribuía quando surgiam dúvidas em relação à execução das atividades.

De acordo com um dos alunos, “as medalhas foram um diferencial para a disciplina, pois nos motivou no aprendizado das novas linguagens de programação. A utilização de medalhas com os nomes dos criadores das linguagens, bem como medalhas com nomes motivadores (Ninja, por exemplo), acirrou a competição saudável entre a turma, dado que cada um dos alunos buscou acumular o maior número de medalhas possíveis. A medalha Mochila representava acomodação, portanto, ninguém gostaria de ganhar esta medalha ao final da disciplina. Isto motivou a participação de todos na realização das atividades”.

Segundo os alunos entrevistados, o resultado final mostrou-se justo, uma vez que as equipes mais engajadas no cumprimento das atividades foram as que obtiveram o maior número de medalhas, ocuparam as primeiras posições do *ranking* e conseguiram os maiores conceitos (notas) na disciplina.

Ao final da disciplina, o professor pediu para que a turma classificasse as 4 linguagens (em ordem crescente) de acordo com a facilidade de uso para desenvolvimento web. Em 1º lugar, os alunos escolheram a linguagem PHP, que é uma das linguagens de programação mais requisitadas pelo mercado de desenvolvimento web. A maioria dos alunos possuíam conhecimento prévio em PHP e citaram como um facilitador o fato da sua sintaxe ser concatenada com HTML. Assim, essa foi a linguagem mais utilizada na Fase 4. Ruby, que ficou em 2º lugar, tornou-se uma das linguagens de programação preferidas pelos alunos na Fase 2, pois, através de seu comando *scaffold*, permite desenvolver o CRUD de uma aplicação de maneira automatizada e consistente. Java ficou em 3º lugar, pois, de acordo com os alunos, para o desenvolvimento web é necessário que se conheça as suas APIs (*Application Programming Interface*), como *servlets* e JSP. Assim, foi usada principalmente na Fase 3 devido o suporte da biblioteca *javax.mail*. Por fim, em 4º lugar, a mais difícil na opinião dos alunos, Python. Apesar da simplicidade de sua sintaxe e da baixa curva de aprendizagem, era novidade para a maioria da turma que teve dificuldades na configuração do seu servidor. Desta forma, foi amplamente usada na Fase 1.

5. Considerações Finais

Este artigo apresentou um relato de experiência da adoção de elementos de gamificação em uma disciplina cujo foco é ensinar Programação Web. Inicialmente, descreveram-se os elementos adotados para a disciplina, como fases e desafios, medalhas, *ranking* e pontuação, bem como as mecânicas de estratégias e avaliação das atividades. Em seguida, o professor descreveu a aplicação destes elementos com foco no ensino e os alunos relataram a sua experiência com foco na aprendizagem.

As principais dificuldades foram relacionadas à mudança de paradigmas de ensino-aprendizagem: os alunos e professores já estão acostumados com o método tradicional. Sendo assim, o professor destacou o esforço para definir os elementos de gamificação e preparar os materiais de apoio. No entanto, ele considerou o trabalho necessário análogo ao de criar o material de uma disciplina tradicional pela primeira vez, destacando semelhanças como: pesquisa e estudo de fontes bibliográficas, definição dos exercícios (desafios) e avaliação (pontuação).

Já os alunos relataram que muitas das vezes ficam desmotivados com o método tradicional de ensino. Com o uso da gamificação, acharam a disciplina mais atrativa em relação às demais do curso de Sistemas de Informação. De acordo com seus relatos, sentiram dificuldades na busca por fontes de informação para aprender a configurar os servidores, aprender a sintaxe das linguagens e, conseqüentemente, concluir as fases. No entanto, eles destacaram que a utilização dos elementos contribuiu significativamente para o maior engajamento e participação de todos nas atividades em sala de aula, pois estimulam a busca pelo cumprimento de desafios e metas propostas pelo professor.

Como foi a primeira vez que essa disciplina foi ministrada, não há como comparar os resultados desta experiência com turmas anteriores. A próxima etapa desta pesquisa será a reoferta da disciplina, no primeiro semestre de 2019, a fim de coletar dados e comparar os resultados obtidos entre duas turmas. Desta forma, serão analisadas as diferenças entre os perfis dos alunos e o impacto destes na adoção da gamificação. Assim, espera-se definir elementos de gamificação mais abrangentes, que considerem diferentes perfis de alunos.

Referências

- ABRAWEB (2018). “Associação Brasileira de Profissionais da Internet”, <https://www.abraweb.com.br/>, Março.
- Campos, A., Gardiman, R., Madeira, C. (2015). “Uma Ferramenta Gamificada de Apoio à Disciplina Introdutória de Programação”. Anais do XXIII Workshop sobre Educação em Informática (WEI). Recife-PE, Brasil.
- Deterding, S.; Dixon, D.; Khaled, R.; Nacke, L. (2011). “From Game Design Elements to Gamefulness: Defining Gamification”. 15th International Academic MindTrek Conference: Envisioning Future Media Environments.
- Figueiredo, K., Ribeiro, J., Souza, R., Vinicius, R. (2015). “Uma Abordagem Gamificada para o Ensino de Programação Orientada a Objetos”. Anais do XXIII Workshop sobre Educação em Informática (WEI). Recife-PE, Brasil.
- Getting Started. (2018). “O que é client-side e server-side?”, <http://tableless.github.io/iniciantes/manual/obasico/o-que-front-back.html>, Março.
- Khaled, R. (2011). “It’s Not Just Whether You Win or Lose: Thoughts on Gamification and Culture”. In: CHI 2011 Workshop Gamification: Using Game Design Elements in Non-Game Contexts. Vancouver, Canadá.
- Pineda, E., Gonzalez, C. (2016). “*Networking Skills in Latin America*”, IDC White Paper, Cisco Systems.
- Suarez, P., Andrade, G., Dantas, A., Oliveira, K., Figueira, G. 2015. “Estudo e Avaliação de Ferramentas para Desenvolvimento Ágil. Revista Principia. Divulgação Científica e Tecnológica do IFPB, Nº 26.
- Umbler (2017). “Mercado de desenvolvimento web: panorama 2016 e expectativas 2017”, <https://blog.umbler.com/br/mercado-de-desenvolvimento-web-panorama-2016-e-expectativas-2017/>, Março.
- Vianna, Y., Vianna, M., Medina, B., Tanaka, S. (2013). “Gamification, Inc.: como reinventar empresas a partir de jogos”. Rio de Janeiro: MJV.
- W3C (2018). “Web Design”, <https://www.w3.org/standards/webdesign/>, Março.
- Werbach, K. Hunter, D. (2016). “For the Win: How Game Thinking Can Revolutionize Your Business”, Wharton Digital Press, Philadelphia.
- Zichermann, G.; Cunningham, C. (2011). “Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps” 1st ed., Sebastopol (CAN): O’Reilly Media, Inc.