

MyTeams

Gabriel Ferreira Silva
Curso Sistemas de Informação
Centro Universitário Unieuro
Brasília, Brasil
gabrielferreira2008@gmail.com

Mateus de Cerqueira Tavares
Curso Sistemas de Informação
mateusctm@gmail.com

Rafael Sampaio Tibery Silva
Curso Sistemas de Informação
rafael.204@hotmail.com

Welson de Lima Teles
Curso Sistemas de Informação
welsonlimawlsn@gmail.com

Resumo—Esse artigo propõe descrever a criação de uma plataforma web com a implementação de linguagens como HTML, java, javascript, CSS, fazendo a interação com o banco de dados. Trata-se de uma plataforma que disponibilizará para os seus navegantes as opções de visualizar, selecionar e adquirir produtos relacionados a games.

Palavras-Chaves—Desenvolvimento web, games, banco de dados.

Abstract—This article proposes to describe the creation of a web platform with the implementation of language such as HTML, java, javascript, CSS, interacting with the database. This is a platform that offers your browsers options for viewing, selecting and purchasing products related a games.

Index Terms—Web development, games, database.

I. INTRODUÇÃO

Com a falta de sites confiáveis na venda de produtos relacionados a games, a MyTeams apresenta se como uma solução para a ausência de páginas voltadas para jogos eletrônicos e prestar a devida atenção aos seus consumidores. A proposta envolve a criação de uma plataforma exclusivamente voltada para o mercado de jogos, disponibilizando os melhores preços e promoções e sempre a mantendo cem por cento atualizada.

Para que fosse possível a implementação dessa aplicação a equipe da MyTeams foi em busca dos conhecimentos necessários para o desenvolvimento web que garantisse a melhor prestação de serviço. Sendo assim a plataforma teria que ser capaz de operações CRUD (Create, Retriave, Update, Delete), utilizando tecnologias como Servlets, HTML, Javascript, CSS, java, buscando um meio seguro e eficiente de integrar um banco de dados.

Para dar início ao projeto e o rápido entendimento de toda a estrutura necessária que a equipe teria que possuir para o desenvolvimento da plataforma, foram feitos diagramas que auxiliaram os programadores e que ajudaram a ilustrar aos stakeholders os mecanismos e objetivos que a plataforma atenderia.

II. DIAGRAMAS

A. Casos de uso

O diagrama de caso de uso é utilizado para facilitar o levantamento de requisitos e melhorar o entendimento de suas principais funcionalidades. Após o levantamento dos requisitos do projeto MyTeams, a equipe responsável conseguiu analisar o cenário, identificar seus atores e suas relações [7].

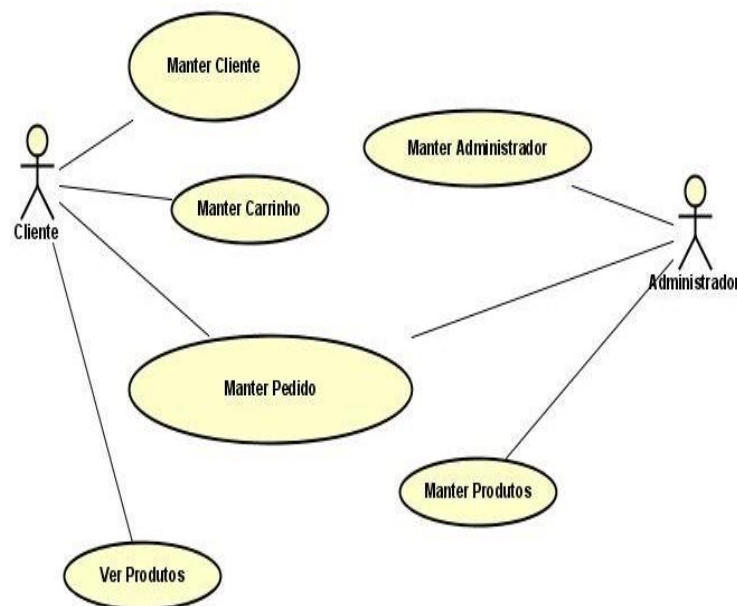


Figura 1 – Diagrama de casos de uso mostrando seus atores e suas respectivas ações.

B. Classes

O diagrama de classes serve para ilustrar como será a estrutura do software, agindo como um guia para seus programadores. Com o diagrama de classes do projeto bem elaborado, serviu para facilitar a parte do desenvolvimento, da

criação de cada classe e posteriormente na criação do bando de dados [7].

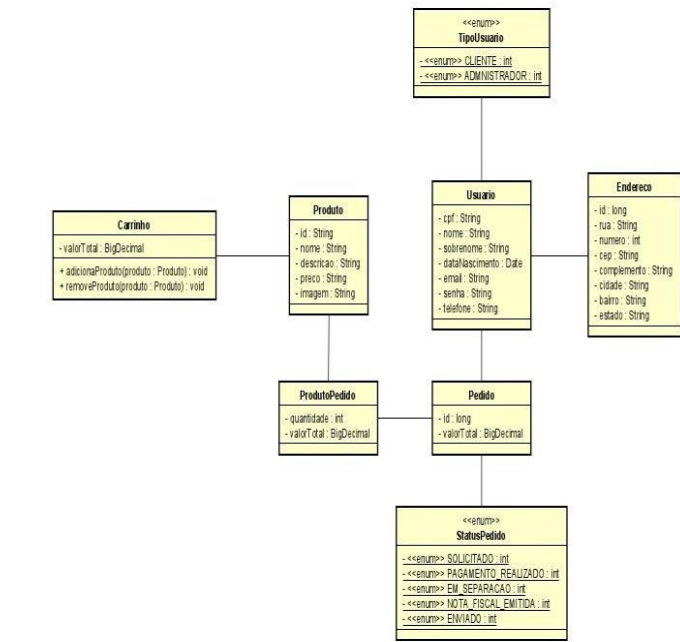


Figura 2 – Diagrama de classe do projeto MyTeams.

C. Sequência

Esse diagrama serve como base para analisar de forma temporal o comportamento dos objetos envolvidos no processo [7]. Com o diagrama de sequência, os desenvolvedores da MyTeams foram capazes de identificar os eventos, seus processos e o métodos necessários para a execução dos mesmos, bem como qual ator seria responsável já que por sua vez esse diagrama é baseado no diagrama de classes.

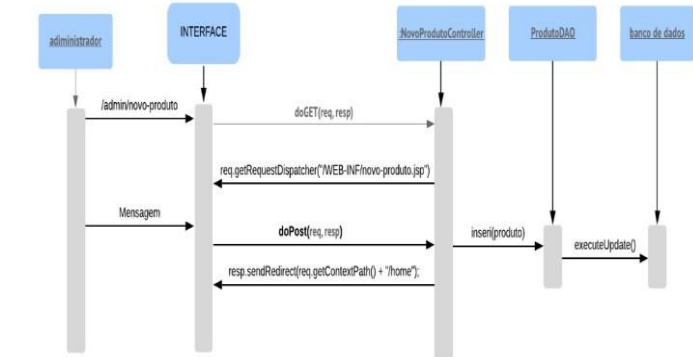


Figura 3 – Exemplo do Diagrama de Sequência utilizado no projeto.

III. IMPLEMENTAÇÃO

Conhecer o funcionamento de um ambiente *web* e a arquitetura cliente-servidor é vital para solucionar os problemas de usabilidade que poderão no futuro prejudicar o negócio da MyTeams que propõe oferecer um serviço completo aos seus navegantes. A conexão funciona em um modelo de pedido-resposta (*request/respost*), o navegante solicita um documento ao servidor que processa a chamada, envia o documento ao cliente e encerra a comunicação [2].

Seguindo o modelo MVC (*Model-View-Controller*), a estrutura web define um esquema de endereçamento (URL), estabelece protocolos de sessão (TCP/IP) e apresentação (HTML e auxiliares), define um protocolo (HTTP) no nível da aplicação seguindo o modelo cliente-servidor [4].

Citados na introdução desse artigo, os *servlets* que são uma classe *java*, serão os responsáveis por tratar as requisições recebidas dos clientes [3]. Ao receber um chamado, a classe *servlet* é capaz de capturar os parâmetros dessa requisição e efetuar qualquer processamento inerente a uma classe *java*, podendo desenvolver uma página HTML [5].

A implementação do sistema consiste no desenvolvimento em etapas começando pelo *front-end*, depois pela criação do banco de dados utilizando um SGBD eficiente, em seguida a construção do *back-end*. Cada uma dessas etapas consiste em adquirir conhecimentos das linguagens HTML e CSS, *java*, Javascript e do próprio SQL.

A. Front-end

Para a criação de um site funcional e interativo para seus usuários, a parte do *front-end* é o componente onde o cliente interage com o sistema, e por ele que constam todas as ligações feitas do *back-end* com o banco de dados. A MyTeams utilizou como forma de facilitar a criação da parte frontal do site o framework web conhecido como *Bootstrap* para melhorar ainda mais a experiência do usuário [1].

```

<body>
  <!--Inicio Navbar -->
  <section>
    <nav class="navbar bg-info navbar-expand-sm navbar-dark justify-content-between">
      <a class="navbar-brand" href="/index.html">
        
        MyTeams
      </a>
      <ul class="navbar-nav">
        <li class="nav-item">
          <a class="nav-link active" href="index.html">Home</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#sobre">Sobre</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="pesquisa.html">Produtos</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#section-contato">Contato</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="/login/index.html">Login</a>
        </li>
      </ul>
    </nav>
  </section>

```

Figura 4 – Uma parte do *front-end* referente à tela principal.

B. HTML

Linguagem utilizada para o desenvolvimento de websites, o HTML é baseado em marcações, essas marcações são chamadas *tags* que são inicializadas e quando finalizadas o sistema entende do que se trata aquela informação [2].

```

<html>
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title>ShowProduto</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" type="text/css" media="screen" href="css/main.css">
  <script src="js/main.js"></script>
  <!--Caso precise usar o js, crie um arquivo .js e altere essa linha para:
  <!-- Bootstrap CSS -->
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
</head>
<body>
  <!--Inicio Navbar -->
  <nav class="navbar bg-info navbar-expand-sm navbar-dark justify-content-between">

```

Figura 5 – Exemplo de uma utilização do HTML no projeto.

C. CSS

Cascading Style Sheets, ou CSS é uma linguagem de folhas de estilo, utilizada para auxiliar a formatar conteúdos estruturados por documentos escritos na linguagem de marcação, como o HTML [2].

```

.recuo p {
  text-indent: 40px;
  max-width: 1100px;
  margin-left: auto;
  margin-right: auto;
}

.slider div{
  margin-left: auto;
  margin-right: auto;
}

.carousel .item {
  min-height: 330px;
  text-align: center;
  overflow: hidden;
}

```

Figura 6 – Uma parte referente ao CSS no projeto.

D. Javascript

Essa linguagem possibilita que sua página web seja capaz de ações como programar, processar e transformar os dados recebidos e enviados, interagindo com a linguagem de marcação como o HTML e com a formatação e estilização do CSS, tornando a experiência do usuário muito melhor uma vez que essa linguagem interage com seu navegador [2].

```
function sendImage() {
    var PATH = '${pageContext.request.contextPath}' + '/imagem/';
    var DONE = 4;
    var STATUS_OK = 200;
    var METHOD_POST = 'POST';

    var httpRequest = new XMLHttpRequest();

    function completed(response) {
        return response.readyState === DONE && response.status === STATUS_OK;
    }

    httpRequest.onreadystatechange = function () {
        if (completed(this)) {
            var img = document.getElementById('imagemUpload');
            var input = document.getElementById('imagem');
            img.style.display = 'block';
            img.src = JSON.parse(this.responseText).url;
            input.value = JSON.parse(this.responseText).url;

            console.log(input);
        }
    };

    httpRequest.open(METHOD_POST, PATH, true);
    httpRequest.send(new FormData(document.getElementById('form')));
}
```

Figura 7 – Utilização do javascript para enviar uma imagem ao servidor.

E. Banco de Dados

Talvez a parte mais complexa e sensível do projeto consiste na parte de criar um banco de dados bem estruturado e que atenda todas as necessidades e aja com eficiência e desempenho na hora de se integrar com o *back-end*. A figura 8 mostra o modelo ER (entidade relacionamento) utilizado na implementação do banco de dados.

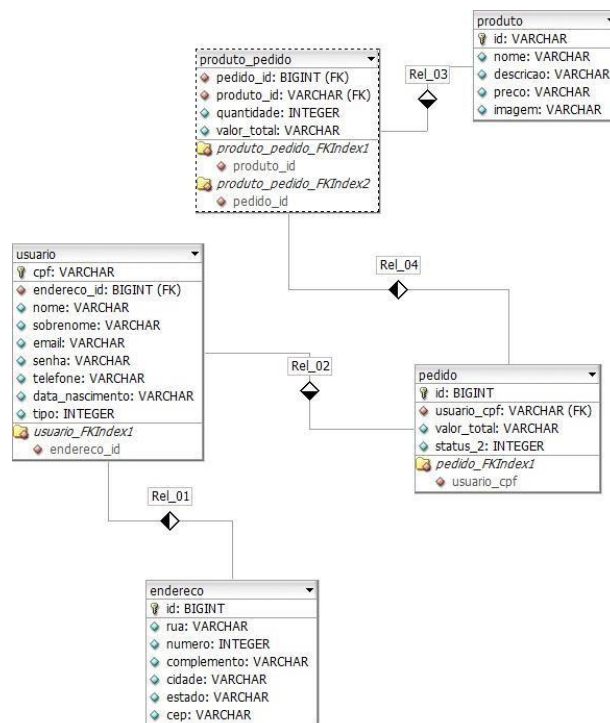


Figura 8 – Modelo de entidade e relacionamento.

```
create table usuario
(
    cpf          varchar(14) primary key,
    nome         varchar(20) not null,
    sobrenome    varchar(20) not null,
    email        varchar(100) not null unique,
    senha        varchar(300) not null,
    telefone     varchar(11) not null,
    data_nascimento date,
    tipo         int          not null,
    endereco     bigint       not null references endereco (id)
);

create table produto
(
    id           varchar(100) not null primary key,
    nome         varchar(100) not null,
    descricao    varchar(500) not null,
    preco        varchar(13) not null,
    imagem       varchar(200) not null
);
```

Figura 9 – Criação das classes usuário e produto no banco de dados.

F. Back-end

Situado no servidor onde a aplicação está hospedada, a parte do *back-end* é responsável por garantir a execução das funções

do site, a segurança de informações contidas no banco de dados e a implementação das regras de negócio.

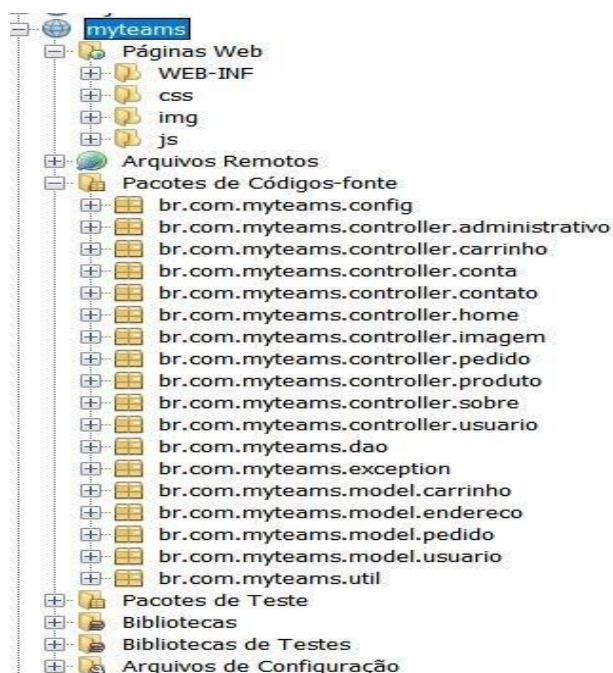


Figura 10 – Uma visão da estrutura do *back-end* com algumas de suas pastas *model* e *controller*.

IV. RESULTADOS E DISCUSSÕES

A MyTeams no seu diferencial propôs entregar uma plataforma confiável e eficiente na sua conexão, garantindo o desempenho do usuário durante a execução de tarefas e sendo capaz de proporcionar informações seguras a respeito dos novos jogos, dos seus preços e a possibilidade da compra, além de contar com as avaliações de técnicos especializados.

Com o público alvo bem definido foram utilizadas estratégias que garantem um destaque no mercado, estar online não é suficiente para assegurar que sejam alcançados os objetivos e metas do projeto, mas ajuda na aproximação com a comunidade *gamer* que irá usufruir do serviço da plataforma [8].

Uma estratégia é evitar muitos dos problemas relacionados a criação de uma plataforma *web* e assim garantir uma boa experiência para o seu usuário. A usabilidade é uma questão primordial já que a falta da mesma ocasiona uma rejeição do *software* por parte dos usuários [2].

Durante o levantamento dos requisitos do projeto foi observado a necessidade do usuário logar para obter vantagens como a compra de produtos, contudo a parte da visualização dos mesmos e seus respectivos preços e promoções não haveria a necessidade do *login*, com o intuito de despertar o interesse do cliente em efetuar o cadastro no site da MyTeams.

Funcionando de forma intuitiva, o fluxo principal da plataforma começa com a página principal que possibilita ao usuário ter acesso a informações sobre a iniciativa da MyTeams

como *Startup*, a opção de visualizar os produtos disponíveis e selecioná-los e adicionar no carrinho de compras.

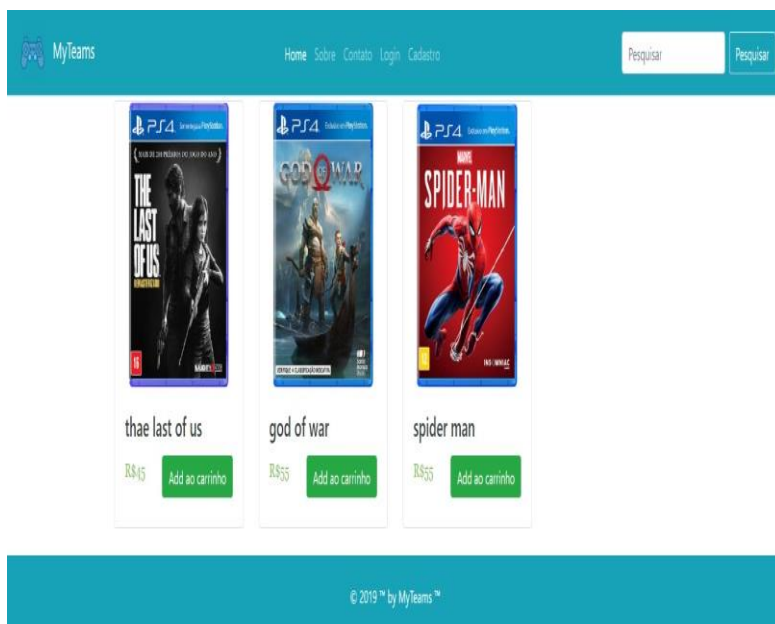


Figura 11 – Página principal da MyTeams.

Selecionando a opção de cadastro o usuário terá de preencher dados necessários para a efetivação do mesmo. Informações relevantes para que posteriormente o usuário seja capaz de logar na plataforma e efetue a compra de jogos.

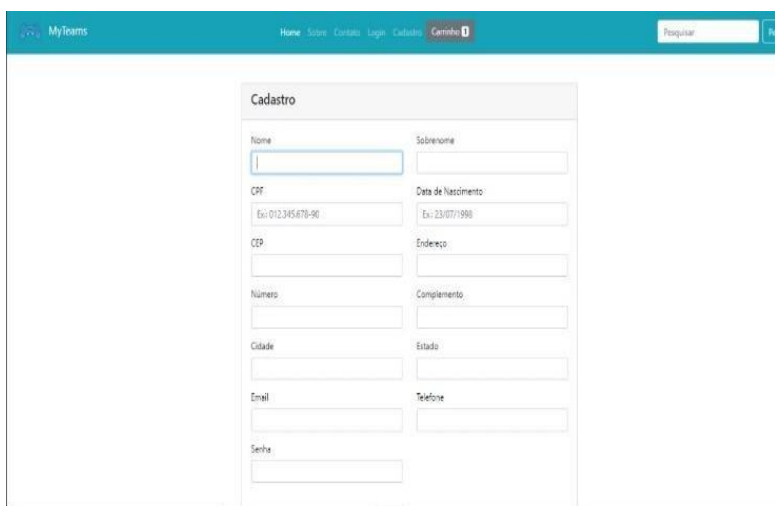


Figura 12 – Página referente ao formulário de cadastro do usuário.

Para a efetivação do cadastro o usuário deverá informar seu nome e sobrenome, seu cpf, data de nascimento, endereço, e-mail e senha. Caso o navegador já tenha login e senha, basta selecionar a opção de login na tela principal e preencher os campos solicitados.

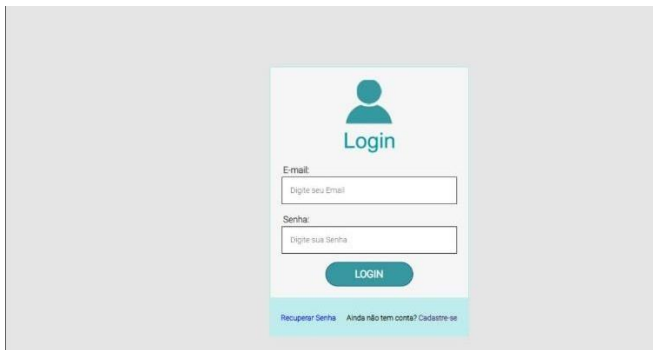


Figura 13 – Tela de login do usuário.

Com o usuário já cadastrado ele habilita seu poder de compra, assim podendo adicionar itens ao carrinho.

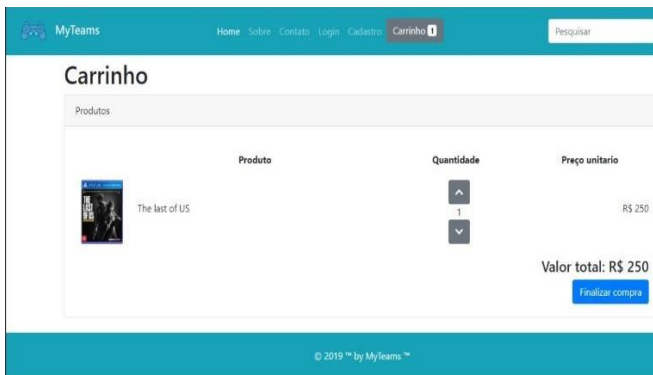


Figura 14 – Demonstração da escolha de um produto guardado no carrinho.

Depois de preenchidos os campos referentes ao número do cartão, seu código de segurança e o nome que consta no mesmo, o usuário poderá finalizar a compra.

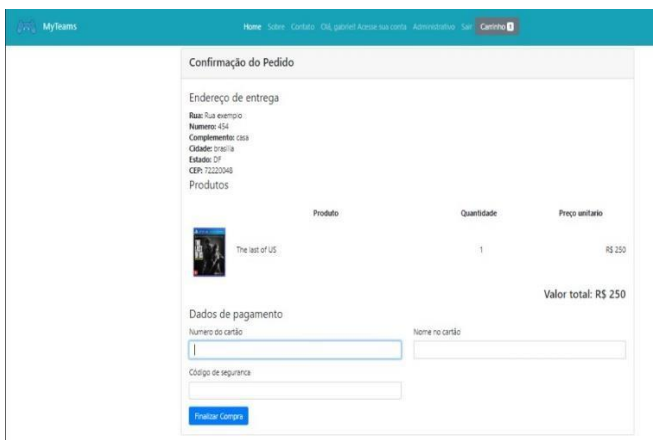


Figura 15 – Na finalização da compra é feita a confirmação de todos os pedidos.

Na tela de finalização de compra o usuário poderá conferir o endereço previamente informado na hora do seu cadastro, o valor total da compra baseado na quantidade de itens e seus respectivos preços. Logo após conferir seus dados o cliente poderá finalizar a compra.

V. CONCLUSÃO

Após serem feitas pesquisas sobre o comércio eletrônico, a equipe foi capaz de analisar que mesmo ao passar dos anos a utilização da internet como forma de comércio se faz cada vez mais presente e muito eficiente, além de ser uma ferramenta eficaz para conhecer melhor as necessidades dos seus clientes [6].

Satisfeita com o término do projeto e aproveitando o aumento do poder de compra dos brasileiros e os inúmeros caminhos que facilitam a compra de produtos via *web*, a MyTeams como uma *Startup* apaixonada por jogos eletrônicos e atendida com os desejos da comunidade *gamer* no Brasil, visa contribuir e servir de referência com a sua plataforma.

AGRADECIMENTOS

A equipe do MyTeams agradece a todo corpo docente de mestres e professores do curso de sistemas de informação da universidade Unieuro.

REFERÊNCIAS

- [1] N. Jain, "Review of different responsive CSS front-end frameworks", Journal of Global Research in Computer Science, 5 (11), November 2014, 5-10.
- [2] M. Winckler, M. Soares Pimenta, "Avaliação de Usabilidade de Sites Web", Instituto de Informática – UFRGS.
- [3] C. Renato de Souza Perri, F. Assis da Silva, L. Luiz de Almeida, "Desenvolvimento de uma ferramenta assistente para criação de aplicações CRUD em java na web", Faculdade de Informática – Universidade do Oeste Paulista (UNOESTE) – Presidente Prudente – SP – Brasil.
- [4] I. Luiz Marques Ricarte, "Programação orientada a objetos: Uma Abordagem com Java.", Faculdade de engenharia elétrica e de computação (FEEC) – Universidade Estadual de Campinas (UNICAMP).
- [5] A. Temple, R. Fernandes de Mello, D. Taffarel Calegari, M. Schiezero, "Programação WEB com Jsp, Servlets e J2EE", CpqD Telecom & IT Solutions – Departamento de Ciências da computação – Instituto de Ciências Matemáticas e de Computação – Universidade de São Paulo.
- [6] L. Leis Diniz, L. Garcia Alves de Souza, L. Rodrigues da Conceição, M. Rangel Faustini, "O Comércio eletrônico como ferramenta estratégica de vendas para empresas", Unisaesiano – Lins, 17 – 21 de outubro de 2011
- [7] G. Thorwald Araujo, "UML 2 – Uma abordagem prática", 2ª edição. São Paulo: Novatec Editora Ltda, 2011.
- [8] V. Cardozo Brandão, "Comunicação e marketing na era digital: A internet como mídia e canal de vendas", PUC Minas – Pontifícia Universidade de Minas Gerais.