

# Desenvolvimento de Sistema de Gerenciamento e Controle para Academias

João Antonio Aires<sup>1</sup>, Sergio Ribeiro<sup>2</sup>, Regiane Orlovski<sup>3</sup>

<sup>1</sup>Dicente no curso de Tecnologia em Análise e Desenvolvimento de Sistemas - Faculdade Guairacá (FAG) 85010-000 – Guarapuava – PR – Brasil

<sup>2,3</sup>Docentes no curso de Tecnologia em Análise e Desenvolvimento de Sistemas - Faculdade Guairacá (FAG) – Guarapuava, PR – Brasil

<sup>1</sup>jantonioayres@hotmail.com, <sup>2</sup>sergio.ribeiro@docase.com.br, <sup>3</sup>regianeorlovski@hotmail.com

**Abstract.** *This article aims to present the development of a web software for management and control academies. For its development requirements were raised and modeled in UML diagrams, also used the PHP language development in relation to information storage was chosen a MySQL database and to enrich the interface a framework CSS. Achieved satisfactory results with the development and deployment of the system, because it met the requirements gathering. Therefore constitutes a software that can be displayed and used easily, keeping track of information, convenience to customers and employees and improved organizational issue.*

**Resumo.** *Este artigo tem como objetivo apresentar o desenvolvimento de um software Web para o gerenciamento e controle de academias. Para o seu desenvolvimento foram levantados requisitos e modelados diagramas em UML, também utilizado a linguagem PHP para o desenvolvimento, em relação ao armazenamento de informação foi escolhido um banco de dados MySQL e para enriquecimento da interface um framework CSS. Obteve resultados satisfatórios com o desenvolvimento e implantação do sistema, pois o mesmo atendeu o levantamento de requisitos. Portanto, constitui em um software que pode ser apresentado e utilizado facilmente, mantendo controle de informação, comodidade aos clientes e funcionários e melhoria na questão de organização.*

## Introdução

O aumento de pessoas que buscam manter o corpo e a saúde é frequente, assim as academias precisam estar preparadas para melhor atendimento e controle de clientes. Dessa forma a implementação de sistemas informatizados oferecem benefícios como agilidade no atendimento ao cliente e ajudam as empresas a ter maior competitividade no mercado.

Além disso, um sistema *Web* apresenta informações precisas e atualizadas, ajuda a empresa estabelecer um equilíbrio na conquista de espaço de mercado, melhorar o ambiente para futuros clientes, e torna-se uma referência na hora de efetuar um atendimento diferenciado, seja pela tecnologia ou pela rapidez no atendimento.

Para o desenvolvimento do sistema foram utilizadas algumas ferramentas, como a linguagem de programação PHP Orientado a Objeto, para a interface o *framework Bootstrap*, em validações de formulários a biblioteca *JQuery*, na modelagem de diagramas, o *software Astah Community* e para a criação do Banco de Dados o *MySQL Workbench*.

O objetivo deste trabalho é apresentar o desenvolvimento de um sistema para facilitar a organização em academias, melhorar o acompanhamento dos clientes, e o fornecimento de informações rápidas e atualizadas, assim como acompanhar atrasos de mensalidades e evitar conflitos de horários.

## **Fundamentação Teórica**

Nos dias atuais pessoas estão buscando manter a boa forma e cuidar da saúde, portanto procuram a solução em academias, fazendo com que essas empresas necessitem investir em soluções para melhorar o atendimento, acompanhamento e o gerenciamento. Pinheiro (2013) complementa que as atividades físicas melhoram a circulação sanguínea, ajuda a aumentar o metabolismo, fortalecer o sistema imunológico, diminuir os riscos de doenças cardíacas, aumentar a resistências dos ossos, aumentar a boa disposição e o bom humor. Assim, frequentar uma academia passa a ser um fator de qualidade de vida.

Com isso a implantação de um sistema de gerenciamento e controle nas academias, tem como resultado manter a empresa organizada, fornecer informações rápidas e atualizadas, para uma melhor tomada de decisão e a excelência no atendimento ao cliente. Laudon (2007) explica que, se melhorar a atenção relacionada ao cliente, e tornar isso como uma prioridade, melhora a qualidade do produto em si. Incluindo também benefícios que a linguagem de programação traz, como por exemplo, fácil armazenamento de dados e segurança nas informações.

Nesse aspecto, para o desenvolvimento faz-se necessário a escolha de um ciclo de vida, Sommerville (2003) explica que um ciclo de vida determina onde inicia e quando termina o desenvolvimento do sistema, funciona de tal maneira que erros sejam reduzidos, e que interpretação do mesmo seja feita com facilidade e evitando perda de tempo. Além, um ciclo de vida enumera objetivamente as partes essenciais do processo, e cada parte contem atividades reais do sistema.

O modelo de ciclo de vida é determinado logo quando se pensa em criar um sistema, para o desenvolvimento em questão foi selecionado o modelo em cascata, Pressman (2011) descreve o modelo cascata como o mais antigo dos modelos, e, no entanto ao longo do tempo, sua eficácia se tornou reconhecida até mesmo pelos seus críticos. Sommerville (2003) explica que as principais etapas do modelo são:

1. Análise e definição de requisito: definir detalhadamente as funções, restrições e objetivos do sistema;
2. Projeto de sistemas e de *software*: elaborar modelagem por meio da arquitetura estabelecida;
3. Implementação e teste de unidades: o sistema é colocado em funcionamento como um todo, e depois testado confirmando se cada unidade atende aos

requisitos;

4. Integração e teste de sistema: elementos são integrados e testados, então é entregue ao cliente;
5. Operação e manutenção: realiza-se a instalação e treinamento do sistema, e correção de futuros erros.

Dessa forma são levantados os principais requisitos que compõem o sistema. Pressman (2011) explica que o levantamento de requisitos é um método fundamental para entender aquilo que o cliente deseja, partes desse método são por meio da identificação de necessidade e análise do mesmo, negociação de soluções e determinação de quais os elementos do sistema.

Em seguida são modelados todos os requisitos do sistema, usando diagramas com a linguagem de modelagem *Unified Modeling Language (UML)*, Segundo Melo (2007) a UML é utilizada para criar várias visões do projeto, pois é descrito o que cada elemento deve fazer, o seu significado, e o que cada elemento tem de relação com o outro. Lima (2011) explica que a modelagem é necessária porque não se consegue entender os sistemas completamente, são necessárias várias modelagens antes de criar um novo *software*. Alguns diagramas usados são de caso de uso, classe e sequência, assim quanto melhor o modelo criado mais próximo fica da realidade.

Os diagramas têm as seguintes funções: Diagrama de caso de uso, Lima (2011) afirma que este, ajuda a melhorar o relacionamento entre analista e o cliente, o diagrama mostra as funções do sistema segundo o usuário. Diagrama de classe, que é a modelagem que representa as classes que o sistema possui, segundo Bezerra (2007) o diagrama de classe é utilizado para montar modelos desde níveis de análise até níveis de especificação, dessa forma é mais rico em informações. Diagrama de sequência é modelado qual o caminho de cada mensagem enviada por determinado objeto, de onde ela sai e onde vai chegar, Bezerra (2007) explica que a finalidade do diagrama é mostrar a ordem que as mensagens acontecem.

No desenvolvimento de diagramas, é necessário a escolha de um sistema para que seja possível efetuar a modelagem, nesse aspecto foi escolhido o *software Astah Community*, que facilita na estruturação e oferece diversas opções de diagrama, possui algumas ferramentas, como sugerir recursos para modelos e alinhar automaticamente atores, caso de uso e as classes (Change Vision, 2013).

Assim que os diagramas são definidos, é determinado qual o padrão de projeto de *software* vai ser utilizado, segundo Dall'Oglio (2007) aponta que o padrão mais conhecido é o *Model View Controller (MVC)*, onde *Model* é o domínio de toda aplicação do sistema, a *View* onde são definidas todas as telas apresentadas ao usuário e *Controller* é executado todas as tarefas do usuário recebendo informações principalmente da camada *View*. Os benefícios desse padrão de projeto atingem principalmente ao desenvolvedor, pois permite reutilizar de um mesmo objeto.

Após a aplicação do padrão de projeto, inicia a parte de programação, para isso foi aplicado à linguagem *Hypertext Preprocessor (PHP)*. Beighley e Morrison (2010)

explicam que a linguagem tem grande capacidade de alterar o conteúdo das páginas, ele atua antes mesmo que o PHP seja carregado. Assim depois que o servidor executa os comandos do mesmo, os códigos são apresentados pelo *browser*. Xavier (2008) aponta que o código é executado no servidor e o que é visualizado pelo navegador é apenas *HyperText Markup Language (HTML)* afim de disponibilizar uma interação a mais com o usuário. Melo (2007) aponta que o PHP trabalha com diversas extensões de arquivos e pode tanto inserir, alterar, consultar e deletar (CRUD). Dall'Oglio (2007) descreve que a linguagem é compatível com uma grande variedade de banco de dados e a cada banco existem diferentes funções que realizam conexões, consultas, retorno, dentre outras.

A linguem PHP é implementada em orientado a objeto, tanto Xavier (2008) quanto Dall'Oglio (2007) afirmam que este se desenham os projetos mais próximos a realidade para que possam ser melhor compreendidos. A utilização do PHP orientado a objeto se alimenta pelo levantamento de requisitos do sistema, então Melo (2007) explica que essa etapa ajuda a encontrar futuras falhas que possam surgir durante o inicio do projeto.

O *layout* do sistema é apresentado em HTML utilizando todas as ferramentas disponíveis, formulários, tabelas, *links*, entre outros, tanto Melo (2007) como Silva (2011) afirmam que o HTML é uma linguagem de marcação que é implementado em um documento direcionado para *Web*, e fornece a comunicação com outros documentos. Segundo Lemay (2002) o HTML é um grupo de estilos para páginas *Web*, como cabeçalhos, parágrafos, listas e tabelas. E também para caracteres como negrito, itálico, entre outros, cada um desses estilos possui um nome e está denominada como *tag*. Essas etiquetas são responsáveis por diferenciar diversos elementos dentro de uma página *Web*. Silva (2011) complementa que o HTML deve ser visualizado completamente por qualquer navegador e devem sempre seguir os princípios de compatibilidade previstos na nota do W3C.

Para formatação do HTML é utilizado uma linguagem de estilo, mais conhecido como *Cascading Style Sheets (CSS)*, essa por sua vez separa os códigos de formatação dos códigos de conteúdo, portanto apenas se estabelece um *link* para a página onde serão inseridos os mesmo. Lemay (2002) explica que as folhas de estilo externas mantém toda a estrutura em uma página separada, tem como benefício usar o mesmo código de estilo em varias páginas, assim facilitando a modificação de interface. Melo (2007), Eis e Ferreira (2012), explicam que o CSS é utilizado para formatar documentos que utilizam marcação, assim quando utilizada pode alterar o modo de visualização em diversas maneiras, tornando-as mais agradáveis. Eis e Ferreira (2012) complementam que todas as páginas *Web* são acessadas por diferentes dispositivos, o CSS é responsável por formatar os elementos das páginas para que elas sejam apresentadas a todos os usuários de forma simples.

No entanto, Junior (2013) observa que o CSS tem como prioridade desenvolver padrões de interfaces, como por exemplo, definir um tema para o *layout*, isso significa definir um estilo para todos os elementos usados na página e utilizá-los quando necessários. Toda essa implementação gasta tempo, para resolver esse problema, orienta-se que é recomendável optar por um *framework* para CSS, assim todos os elementos estão prontos para uso.

O *framework* CSS utilizado no sistema é o *Bootstrap*, que tem como configuração padrão, temas para botões, formulários, tabelas, *link*, imagens, e muitos outros, Silva (2011) explica que além de ser uma compilação de arquivos CSS, HTML e *JavaScript*, possuem códigos que mostram avisos *pop-up*, caixas de diálogo sobrepostas, barras de erro e mais interações.

Para melhor utilização dos recursos do *framework* foi configurado uma biblioteca *JavaScript*, que para Silva (2000) é uma linguagem interpretada, ou seja, quando acessado pelo navegador o *script* é executado. A biblioteca denominada é o *JQuery*, com prioridade de adicionar interatividade e dinamismo às páginas *Web*, segundo Silva (2008) o *JQuery* oferece funcionalidades para a criação de *scripts*, que buscam acrescentar a usabilidade e acessibilidade para melhor experiência do usuário. Para enriquecimento das páginas são adicionados, efeitos visuais, busca de informações sem necessidade de recarregar a página, mais interatividade, modificação de conteúdos e alteração de estilos. Além disso, tem como principais características a utilização de seletores CSS para identificar elementos na estrutura HTML e são compatível com *plug-ins* e extensões.

Para escrita de todos os códigos em PHP, HTML, CSS e *JQuery* a opção foi por um *software* de editor de códigos, o *Sublime Text 2*. Esse *software* oferece ferramentas simples e muito úteis, Yung (2011) explica que *Sublime Text 2* possui o fundo preto para melhor leitura dos códigos, apresenta recursos a mais que outros editores, alguns *plug-ins* ajudam muito na hora de escrever, caso esqueça algum comando o programa ajuda a lembrar dando varias opções do mesmo, também é possível selecionar todas as *tags* de um mesmo grupo.

O sistema oferece uma interface agradável, interatividade e rapidez, no entanto precisa ser alimentando com informações, elas são fornecidas ao usuário que manipulam o programa por meio de pessoas cadastradas. Essas informações são armazenadas em um banco de dados, que segundo Elmasri (2011) é projetado e alimentado com informações específicas, assim todos os dados são derivados de um fator real. Silberschatz (2006) explica que o banco de dados ajuda a manter informações organizadas e apresentando diversas vantagens, evitando redundância e inconsistência de dados, fácil acesso de dados e melhoramento na segurança.

Na modelagem de banco de dados uma ferramenta essencial é o *MySQL Workbench* o mesmo é capaz de realizar diversas funções, além de fornecer uma interface agradável ao usuário. Segundo Oracle Corporation (2003) explica que o mesmo pode projetar visualmente o modelo, gerar e gerenciar o banco de dados. Também oferece melhor aparência para a sintaxe, auto completar códigos do banco de dados e um histórico de execução dos mesmos, proporcionando total controle do banco, podendo acessar diretamente do programa.

A linguagem padrão de acesso do banco de dados é *Structured Query Language (SQL)*, responsável por qualquer ação que deseje realizar no banco de dados, armazenar, excluir, alterar e pesquisar, Silberschatz (2006) afirma que a linguagem de banco de dados pode ser chamada de linguagem de manipulação de dados e podem ter vários tipos de acesso, recuperação de informações, inserir informações, excluir informações e modificar cada uma delas. Também explica que consultas em banco de dados podem

tanto ser realizadas em uma única tabela como também em varias tabelas. Elmasri (2011) aponta que linguagem SQL é muito grande, com diversos atributos e privilégios, mas são essenciais para desenvolvimento padrão do banco.

## **Etapas do Desenvolvimento do Projeto**

O projeto desse sistema orientou-se primeiramente na coleta de requisitos, parte fundamental para o desenvolvimento do *software*, por meio de uma entrevista com o usuário o mesmo apontou os requisitos essenciais, onde cita-se cadastro e manutenção do cliente, acompanhamento das mensalidades, medidas corporais, relatórios, entre outros, que se encontra no Apêndice A.

Após a definição dos requisitos foi escolhido o ciclo de vida do sistema, pois ajuda a reduzir erros e melhorar a interpretação do mesmo. Por descrever de atividades fundamentais foi determinado o modelo cascata, facilitando o planejamento e a fácil visualização do processo.

Posteriormente ao levantamento de requisitos e escolha do ciclo de vida efetuou-se a modelagem do sistema por meio dos diagramas UML. Primeiramente foi elaborado o diagrama de caso de uso, que por sua vez, definiu dois autores, usuário e administrador, que conseqüentemente realizam atividades relacionadas com seu cargo perante a empresa, que apresenta-se no Apêndice B. Da mesma maneira foi modelado o diagrama de classe, que descreve quais as classes existentes e quais as relações entre elas, apresentando que existe uma classe responsável por carregar outras ligadas ao mesmo, assim permitindo que usuário faça uso de determinados métodos de cada classe, essa modelagem é apresentado na Apêndice C. O diagrama de sequência é dividido em dois diagramas, no Apêndice D o diagrama demonstra o fluxo do sistema, com as funções primarias, como cadastrar cliente, professor e agenda. No Apêndice E são as funções secundarias, que estão relacionadas à classe cliente, como cadastrar mensalidades e medidas. Nesse contexto a modelagem torna-se necessária, pois ajuda a entender melhor o que usuário necessita. Para a realização das modelagens dos diagramas, a ferramenta utilizada foi a *Astah Communit*, pois ela é capaz de criar diagramas com facilidade, e tem a disposição vários tipos de diagrama para ser utilizado.

O próximo passo foi à elaboração da modelagem lógica do banco de dados, com base no que foi definido nos diagramas. Essa modelagem tornou-se necessária, pois mantém informações organizadas, sem risco de perda e fácil acesso aos dados. A modelagem desenvolvida para o sistema se encontra no Apêndice F, que descreve quais tabelas compõem o banco de dados do sistema e suas relações, são elas, tabela “cliente”, “agenda”, “professor”, “mensalidades” e “medidas”, as demais tabelas não possuem relação, são apenas tabelas para armazenamento de relatório, “clienteLog”, “agendaLog” e “mensalidadeLog”, essas tabelas são incrementadas com gatilhos, que conseqüentemente são ativados com inserções em tabelas referente as seus nomes. O *software* usado para implementação da modelagem do banco de dados é o *MySQL Workbench* visto este ser capaz de criar a modelagem e ao mesmo tempo atualizar dentro do servidor do Banco de Dados.

Na sequência, os arquivos seguiram um padrão de projeto, denominada MVC,

esse estilo de estrutura favorece o desenvolvimento de forma que os objetos possam ser reutilizados. Na pasta *Model* são configuradas todas as classes de funcionamento do sistema, no diretório *View* encontram-se todas as telas que serão apresentadas ao usuário e em *Controller* realiza a conexão com o banco de dados, definição de variáveis por meio da URL e início da *session*. Também é adicionada uma pasta *config* para definição de constantes e atributos para acesso ao banco dados e uma pasta *public* responsável por guardar arquivos CSS, imagens do sistema e *JavaScript*

O desenvolvimento dos códigos em PHP tornou-se indispensável, já que se trata de um sistema *online*, a ferramenta para a edição de códigos é *Sublime Text 2*, considerando que esta ferramenta facilita muito na hora de editar algum erro ou procurar um código. Uma função essencial foi a de *\_\_autoload()*, responsável por carregar as classes automaticamente, evitando a perda tempo de estanciar a classe dentro do arquivo para poder utiliza-la, além disso, a utilização da função *explode()* faz as separação de elementos vindo da URL, separando qual a classe pertence, a função a ser realizada e qual o registro dentro do banco de dados.

Junto com a função *explode()* foi implementado duas funções, a *array\_reverse()* que permite inverter uma variável do tipo *string* armazenando-a em uma *array*, e a *implode()* que faz uma ligação de todos os elementos do *array* separando-os por um determinado ponto. Essas três funções utilizadas juntas são responsáveis por realizar um tratamento no atributo de tipo *date*, que por sua vez é cadastrada no banco de dados como uma data americana, ou seja, primeiro o ano depois o mês e então o dia, o tratamento converte a mesma para uma data brasileira, primeiro o dia, depois o mês e então o ano.

Outra função PHP utilizada é *header()*, tem a função de gerar documentos do *excel*, que são dividido em quatro métodos.

1. *header("Content-type: application/vnd.ms-excel");* – código que determina que o arquivo é uma planilha do *Excel*.
2. *header("Content-type: application/force-download");* – foçar o início do *download* do arquivo.
3. *header("Content-Disposition: attachment; filename=file.xls");* – determina o nome do arquivo.
4. *header("Pragma: no-cache");* – imprime o conteúdo da tabela no arquivo gerado.

A escolha do *designer* do sistema deu-se por um *framework* CSS que fornece facilidade no desenvolvimento de códigos, e conseqüentemente possui o padrão de elementos usados na internet. A ferramenta utilizada *Bootstrap*, esta disponível na internet para estudos e implementações, a ferramenta é capaz de configurar estilos de todos os elementos usados para compor o sistema, desde botões, tabelas, menus, entre outros, mas para melhor adaptação do mesmo é criado um arquivo CSS *style*, responsável por configurações separadas do *framework*. Exemplo é retirar margens das *tags* *<h1>* ao *<h6>* responsáveis por identificar títulos na página, já que o mesmo não possui essa configuração. Além disso, beneficia *tags*, como *address*, que é específica para informações do tipo endereço e a *tag dl* utilizada para uma lista de termos que

possuem descrições, oferece também uma classe de estilo que denomina uma largura fixa a diferentes elementos, permitindo que o conteúdo se mantenha sempre padrão.

O *framework* oferece diferentes componentes para promover a navegação do usuário, um componente utilizado é o *Menu dropdown* que ao clicar apresenta uma lista de *links* como mostra no Apêndice G. Também a utilização do *Breadcrumbs*, que mostra qual o caminho que usuário percorre evitando erros dentro do sistema, apresentado no Apêndice H. Além disso, oferece alguns *plug-ins* como *modal*, que quando ativado por um botão uma caixa de diálogo é apresentado na tela, apresentado no Apêndice I, e o *tooltip* realiza uma animação com o título dado a um link, exemplo no Apêndice J.

Para tornar mais simples a navegação do sistema, a utilização da biblioteca *jQuery* ajuda a criar animações e manipular eventos, no sistema o mesmo utiliza *plug-ins* responsáveis pela verificação de *inputs* dentro de formulários, assim obrigando o usuário a preencher todos eles corretamente. Também foi criadas funções específicas para determinados campos, como a validação do número de CPF e da data de nascimento, assim como a configuração de máscaras em *inputs* que predominam a utilização de apenas números, dessa forma não é necessário digitar caracteres como pontos, traços ou parênteses, os mesmos são adicionados automaticamente, demonstrado no Apêndice K.

Outro recurso desenvolvido para o sistema por meio do *jQuery*, é a criação de um *plug-in* que determina uma condição dependo do valor que se encontra dentro de um elemento. Como por exemplo, na visualização de mensalidade, onde existe um atributo que por sua vez determina se a mesma está paga ou não, assim o *plug-in* realiza a verificação desse atributo, e adiciona uma classe de estilo que representa um dos seus estados, o mesmo se encontra no Apêndice L. Além disso, foi criada uma função que realiza uma pesquisa rápida dentro de um conteúdo de uma tabela de HTML, sem uso do banco de dados, assim buscando todos os dados contidos em uma tabela e oculta os demais.

## **Resultado**

O resultado obtido por meio do desenvolvimento desse trabalho foi de um sistema que tem como característica a fácil operação, interface simples e moderna. Apresenta-se como uma importante contribuição para o gerenciamento em academias visto atender os principais requisitos para o funcionamento das mesmas.

No desenvolvimento do sistema, o levantamento de requisitos auxiliou para melhor entender o que era necessário ser desenvolvido. Com tudo a modelagem de diagramas evitou a criação de classes inúteis e uma base de quais elementos foi composta o banco de dados. Com a modelagem lógica do banco de dados, foi possível a melhor elaboração do mesmo, evitando a perda de tempo com criação de tabelas inutilizáveis.

A linguagem de programação procedeu a um sistema dinâmico e facilitou na questão de suporte e manutenção, acrescenta-se também que o *framework* para CSS aumenta ainda mais a fácil operação como também mantém o sistema com padrões de

interfaces, e para melhor interação do usuário com o sistema, a escolha da biblioteca *jQuery* resulta em uma validação pelo lado do usuário, que ajuda a evitar dados inválidos e campos vazios.

O sistema implantado facilitou a organização em academias, melhor acompanhamento dos clientes, fornecendo informações rápidas e atualizadas, e pode também ser utilizado para verificar atrasos de mensalidades e conflitos em horários. Além disso, torna-se uma característica fundamental, para a escolha de uma academia.

## **Considerações Finais**

O objetivo desse artigo foi apresentar o desenvolvimento de um sistema direcionado às academias de ginástica. O mesmo tem como base o controle de atividades de funcionamento da academia, nesse aspecto o sistema foi desenvolvido com uma linguagem de programação, capaz de desenvolver um sistema para atender as necessidades apontadas pelo levantamento de requisitos.

O PHP apresentou eficiência no desenvolvimento do sistema, mostrou muitas vantagens, como rapidez e facilidade na execução nos principais sistemas operacionais. Além disso, interage muito bem com o *MySQL* resultando em um bom desempenho em consultas e outras operações realizadas.

Em funcionamento o sistema faz o controle de clientes, mensalidades, medidas corporais, professores e agendamentos, trazendo muitas vantagens para a empresa, como por exemplo: praticidade de operação, controle de informação, comodidade aos clientes e funcionários e a melhoria organizacional, acrescenta-se também, que a empresa pode se manter em competitividade no mercado, reduzir chances de erros que podem resultar a uma possível perda de clientes.

Por meio desse artigo, junto com o conhecimento técnico adquirido ao longo do curso, foi possível criar um sistema eficiente para gerenciamento e controle de academias, com base em PHP Orientado a Objeto e a implementação de ferramentas como *Bootstrap* e *jQuery*.

## **Referências**

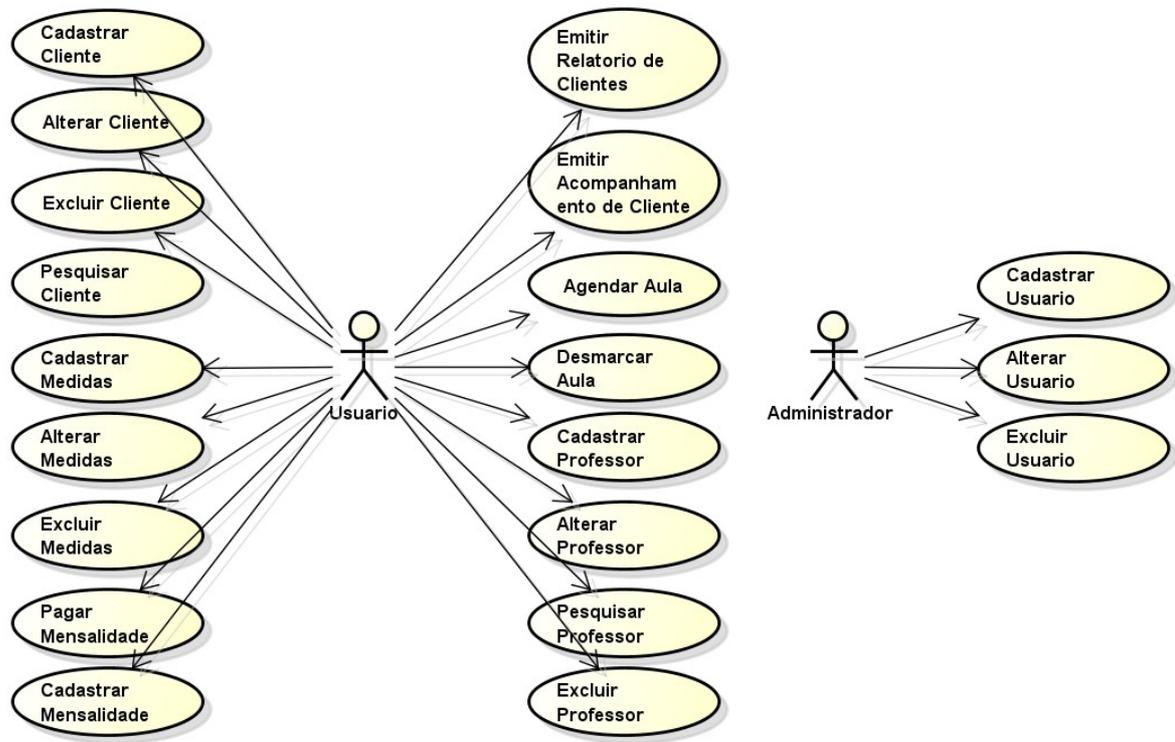
- Beighley, L. & Morrison, M. (2010) “Use a Cabeça! PHP & web”, Rio de Janeiro, Alta Books Editora.
- Bezerra, E. (2007) “Princípios de análise e projeto de sistemas com UML”, Rio de Janeiro, Elsevier.
- Change Vision. Astah. (2013) “Astah Community”, Disponível em: <http://astah.net/editions/community>, acessado em 21 de out. de 2013.
- Dall’Oglio, P. (2007) “PHP: programando com orientação a objetos”, São Paulo, Novactec Editora.
- Eis, D. & Ferreira, E. (2012) “HTML5 e CSS3 com farinha e pimenta”, São Paulo, Tableless.
- Elmasri, R. (2011) “Sistemas de banco de dados”, 6 ed. São Paulo, Pearson Addison Wesley.

- Junior, E. M. (2013) “Seja produtivo usando o Twitter Bootstrap!”, Disponível em: <http://www.devmedia.com.br/seja-produtivo-usando-o-twitter-bootstrap/24967>, Outubro.
- Laudon, K. C. (2007) “Sistemas de informação gerenciais”, 7 ed. São Paulo, Pearson Prentice Hall.
- Lemay, L. (2002) “Aprenda a Criar Páginas Web com HTML e XHTML em 21 Dias”, São Paulo, Pearson Education do Brasil.
- Lima, A. da S. (2011) “UML 2.3: do requisito à solução”, São Paulo, Érica.
- Melo, A. A. de. (2007) “PHP profissional: aprenda a desenvolver sistemas profissionais orientados a objetos com padrões de projeto”, São Paulo, Novatec Editora.
- Oracle Corporation. MySQL Workbench. (2013) “MySQL Workbench”, Disponível em: <http://www.mysql.com/products/workbench/>, Outubro.
- Pinheiro, M. (2013) “Benefícios da atividade física”, <http://www.tuasaude.com/beneficios-da-atividade-fisica/>, Outubro.
- Pressman, R. S. (2011) “Engenharia de software: Uma Abordagem Profissional”, 7 ed. Porto Alegre, AMGH.
- Silberschatz, A. (2006) “Sistemas de banco de dados”, Rio de Janeiro, Elsevier.
- Silva, M. S. (2008) “jQuery: a biblioteca do programador JavaScript”, São Paulo, Novatec Editora.
- Silva, M. S. (2011) “HTML5”, São Paulo, Novatec Editora.
- Silva, O. J. (2000) “JavaScript: guia prático do Webmaster”, São Paulo, Érica.
- Silva, R. (2013) “Twitter lança BootStrap para ajudar desenvolvedores de serviços”, Disponível em: <http://www.techtudo.com.br/artigos/noticia/2011/08/twitter-lanca-bootstrap-para-ajudar-desenvolvedores-de-servicos.html>, Outubro.
- Sommerville, I. (2003) “Engenharia de software”, São Paulo, Addison Wesley.
- Xavier, F. S. V. (2008) “PHP – do Básico à Orientação a Objeto”, Rio de Janeiro, Editora Ciência Moderna Ltda.
- Yung, R. (2013) “Sublime Text 2”, Disponível em: <http://codigofonte.uol.com.br/ferramentas/sublime-text-2>, Outubro.

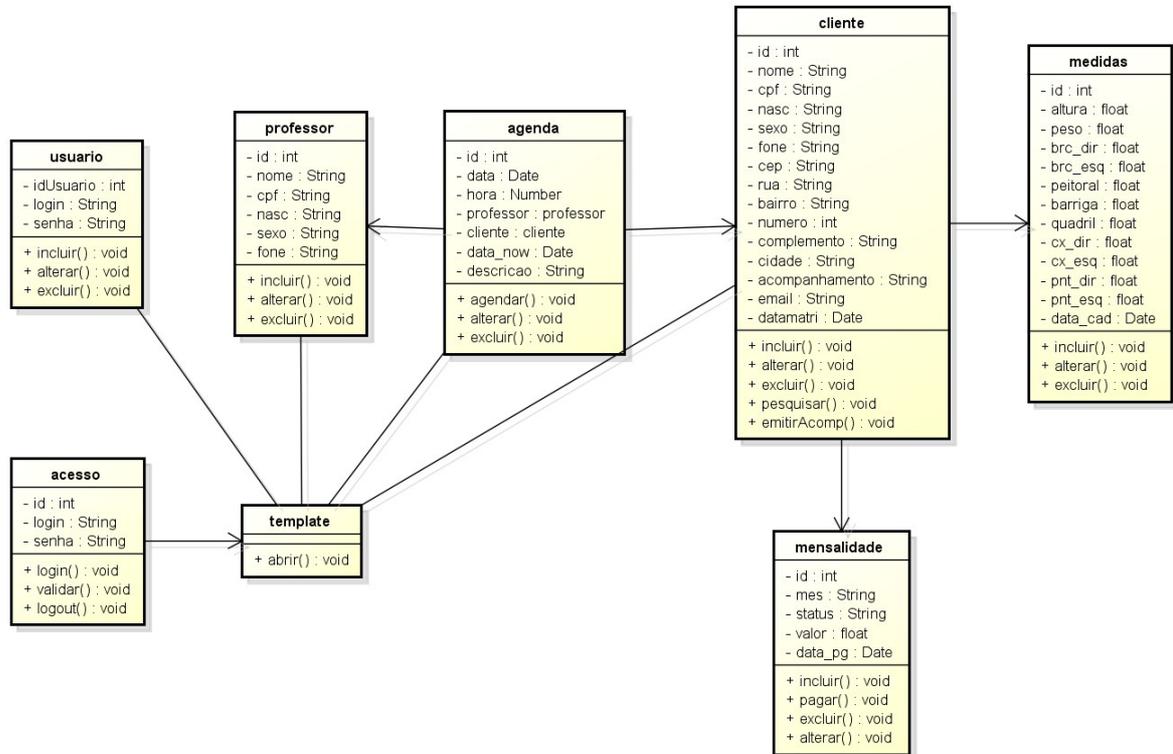
## Apêndice A: Levantamento de Requisitos

- Cadastro de Clientes
- Cadastro de Professores
- Cadastro de Mensalidade
- Cadastro de Medidas Corporais
- Agendamento
- Relatórios

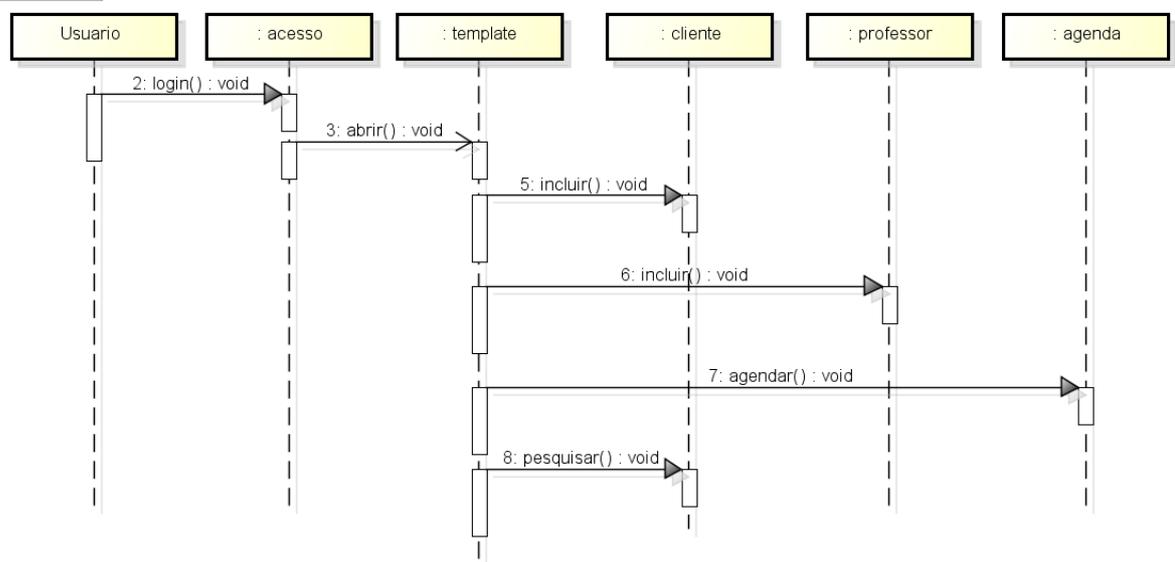
## Apêndice B: Diagrama de Caso de Uso



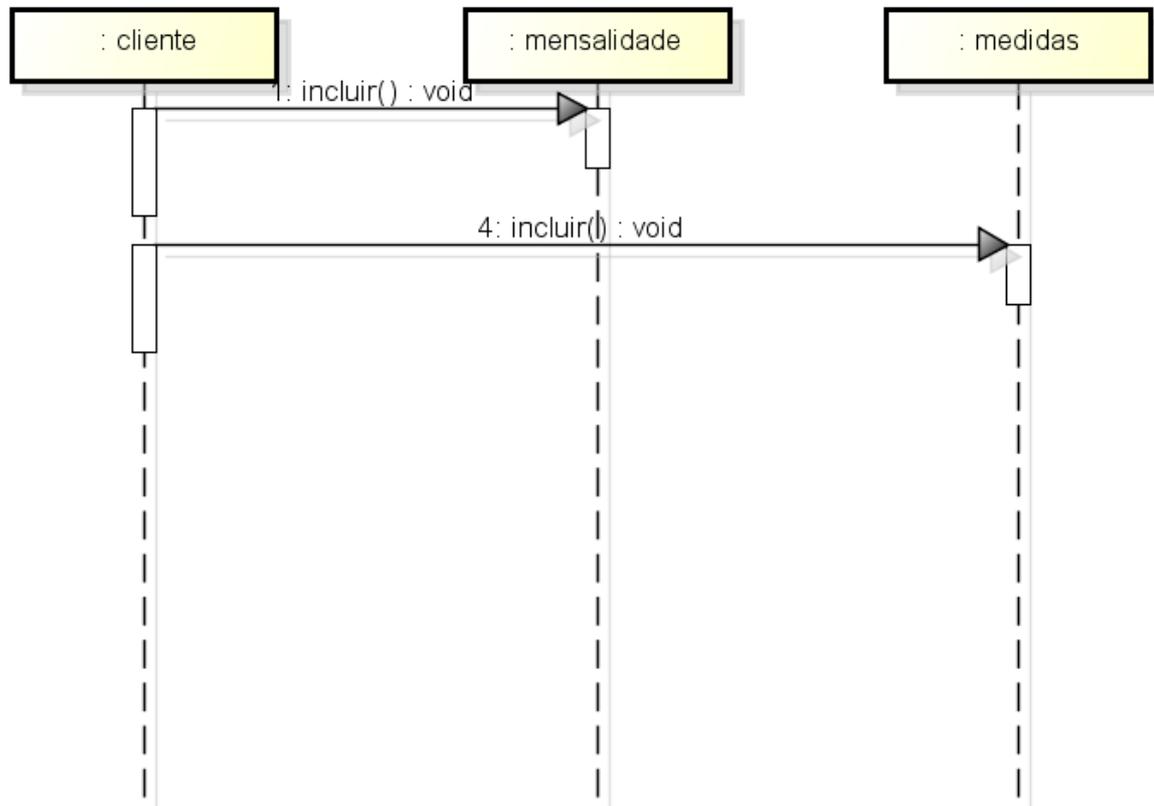
## Apêndice C: Diagrama de Classe



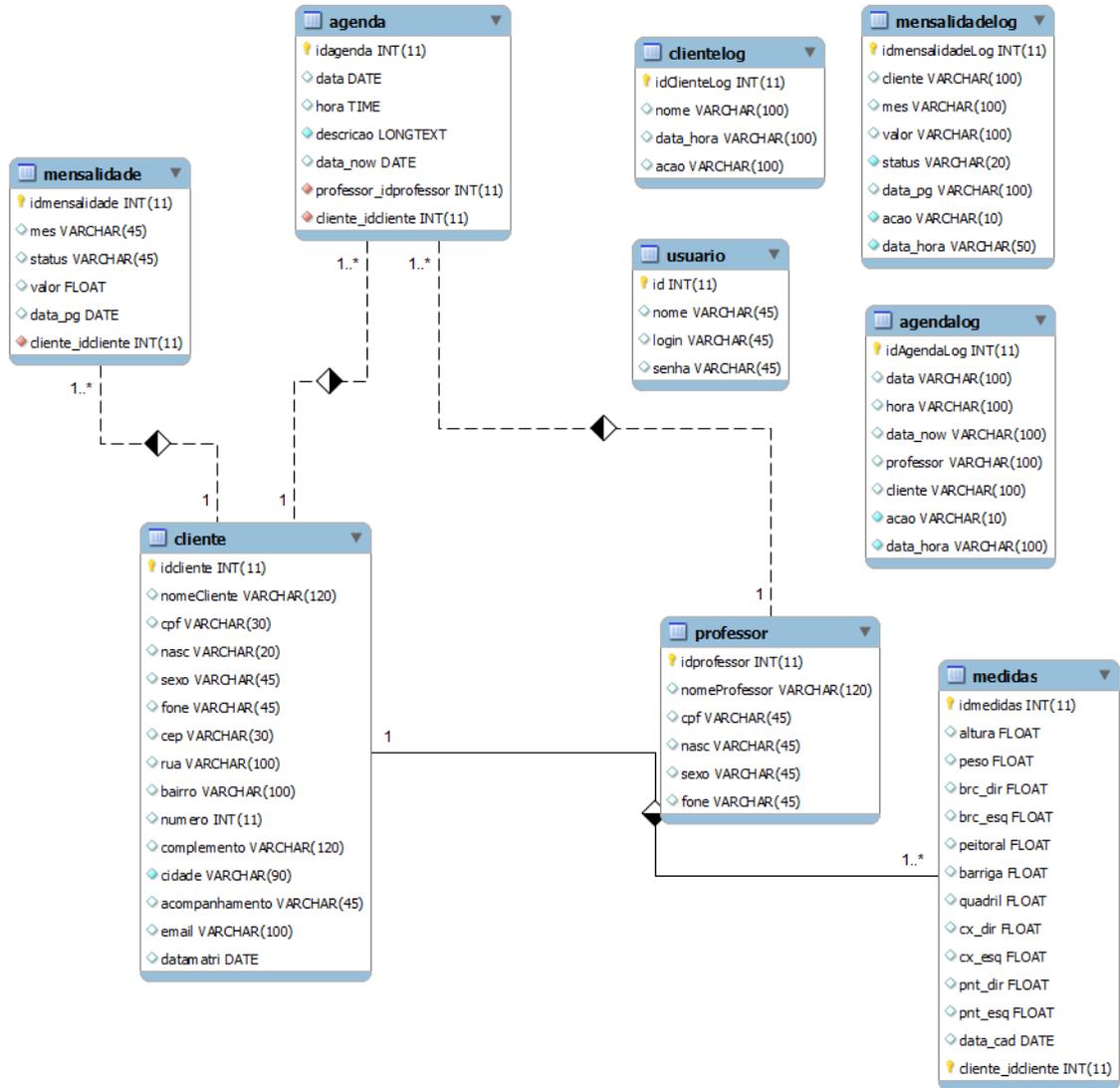
## Apêndice D: Diagrama de Sequência (Um)



## Apêndice E: Diagrama de Sequência (Dois)



## Apêndice F: Modelagem Lógica do Banco de Dados



## Apêndice G: Modelo de Menu Dropdown



localhost/teste/Professor/show

## Apêndice H: Utilização do Breadcrumbs

Sistema de Academia   Cliente   Agenda   Outros ▾   Relatório ▾   Sair

[Cliente](#) / [Detalhes James Paul McCartney](#) / Incluir Mensalidade

Cliente: James Paul McCartney

## Apêndice I: Exemplo de Modal

The screenshot displays a web application interface with a dark theme. At the top, a navigation bar contains the text "Sistema de Academia" and several menu items: "Cliente", "Agenda", "Outros", "Relatório", and "Sair". Below the navigation bar, the main content area is titled "Agenda" and features a table with the following data:

Id	Data/Hora
2	11/12/2013 às 17:00:00
3	11/11/2014 às 18:00:00
6	14/11/2013 às 17:59:00

A modal window titled "Agenda - João Antonio Aires" is open, showing a calendar for "Dezembro 2013" with the number "11" highlighted. The modal content includes:

- Quarta-Feira às 17:00:00
- Acompanhamento com Jackie Chan
- Descrição: Corrida

At the bottom right of the modal is a "Fechar" button. To the right of the modal, there is a search bar labeled "Buscar" and a section titled "Ações" containing three rows of buttons: "Visualizar", "Editar", and "Excluir".



# Apêndice K: Mascaras para campos

## Cliente

Nome:

Sexo:

Telefone:

CPF:

Email:

## Apêndice L: Visualização de Mensalidade

### Mensalidade

Incluir Mensalidade

Mês	Status	Valor	Data de Pagamento
Dezembro	Não Pago	R\$ 150.00	01/11/2013 
Novembro	Não Pago	R\$ 150.00	30/10/2013 
Outubro	Pago	R\$ 150.00	29/10/2013 