



UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO

CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA

ESCOLA DE INFORMÁTICA APLICADA

# Black Belt

[Cristiano Muniz e Claudia Martins]

**Orientador**

[Leila Cristina Vasconcelos de Andrade]

RIO DE JANEIRO, RJ – BRASIL

[DEZEMBRO] DE [2014]

[BlackBelt]

[Cristiano Muniz e Claudia Martins]

Projeto de Graduação apresentado à Escola de  
Informática Aplicada da Universidade Federal do  
Estado do Rio de Janeiro (UNIRIO) para obtenção do  
título de Bacharel em Sistemas de Informação.

Aprovada por:

---

[Leila Cristina Vasconcelos de Andrade] (UNIRIO)

---

[Luiz Amâncio Machado de Sousa Júnior]

---

[Mariano Pimentel]

RIO DE JANEIRO, RJ – BRASIL.

[DEZEMBRO] DE [2014]

## **Agradecimentos**

[Cristiano Muniz]

Gostaria de agradecer à todos os mestres de luta com os quais fizemos as pesquisas que tornaram possíveis a verossimilhança do sistema com a necessidade do mundo real, pois é a partir dessa necessidade que o produto surge. Menção especial neste caso para o mestre Raphael Britto, presidente da confederação de Sambo RJ e membro da confederação e da associação RJ de Karate Shotokan, por ter sido o maior colaborador do projeto, tornando-se assim sócio do mesmo. Gostaria de agradecer à minha orientadora Leila Cristina pois sempre foi de uma grande paciência, carinho e atenção por todo o período no qual cursei Sistemas de Informação na UNIRIO, e me deu suporte sempre que possível. Agradecimentos especiais à minha namorada Cristiane Viamonte que me incentivou à não desistir em momentos de dificuldade, e à minha mãe Aldacy Pereira da Silva porque desempenhou, sempre com maestria, a maioria dos papéis citados acima.

[Claudia Martins]

Meus sinceros agradecimentos a todos aqueles que de uma forma ou de outra doaram um pouco de si para que a conclusão deste trabalho se tornasse possível.

À minha família por acreditar que eu era capaz e investir em mim, não medindo esforços para que eu chegasse até esta etapa de minha vida. Mãe, seu cuidado e dedicação foram o que me deram a esperança para seguir e a certeza de que não estou sozinha nesta caminhada. Você é minha heroína. Ao meu pai que apesar de todas as dificuldades me fortaleceu e que para mim foi muito importante.

À todo corpo docente do curso de Sistemas de Informação por me proporcionar o conhecimento não somente racional, mas na demonstração do caráter e ética profissional, além da atenção infindável e principalmente por não somente terem me ensinado, mas por terem me feito aprender. A palavra mestre, nunca fará justiça aos professores dedicados aos quais sem nominar terão os meus eternos agradecimentos.

Agradeço também a cada um dos amigos e colegas por toda compreensão e apoio sempre prestado não somente durante este período árduo e de muito esforço, mas em toda minha vida.

E, finalmente, a DEUS pela oportunidade e pelo privilégio que me foram dados em compartilhar tamanha experiência de vida.

*“Se enxerguei mais longe, foi porque me apoiei sobre os ombros de gigantes.”*

*- Isaac Newton*

## **RESUMO**

O desenvolvimento de sistemas para o auxílio nas atividades cotidianas é considerado cada vez mais relevante nos diversos contextos do cenário de vida real, pois proporcionam maior praticidade, organização, agilidade e outros aspectos positivos. Contudo, ainda existem diversas áreas que contam com sistemas precários, ou até mesmo não possuem uma aplicação que as auxiliem - como é o caso do sistema de artes marciais, assunto abordado neste trabalho.

Atualmente, no Brasil, não existe um software unificado que armazene, compartilhe e documente os dados e informações relevantes a todas as entidades relacionadas às artes marciais de maneira integrada. Este trabalho apresenta uma proposta para resolver o problema da informatização neste universo de lutas: o software BlackBelt.

Nossa proposta consiste em utilizar um sistema integrado de quatro módulos, onde cada módulo representa uma entidade forte dentro do sistema mundial de organização de artes marciais, sendo elas: Federação, Confederação, Associação e Professor. Através da associação de técnicas de programação e banco de dados, é possível informatizar tarefas destas artes que hoje são feitas em papel de forma arcaica, muitas vezes em papel e com pouca ou nenhuma segurança.

**Palavras-chave:** Artes Marciais, Professor, Módulo, Integrado

## **ABSTRACT**

The system development for helping everyday's activities is considered more and more relevant in real life's scenery contexts because they bring practical improvements, organizational benefits, agility and other positive aspects. Yet there are some areas that still don't have the proper systems to help or solve integratetional problems, sometimes no system at all. The problem adressed in this paper concerns about one of these problems, the one found in martial arts organizations.

Currently, in Brazil, there is no integrated software that stores, shares and documents all data and relevant information to all entities related to martial arts in an integrated context. This work presents a proposal to solve the problem regarding computational issues in the martial arts universe: The BlackBelt software.

Our proposal consists in a four-module integrated system, each module represented by one of the main entities in the worldwide martial arts system, they are: Teacher, Association, Federation and Confederation. Using programming and database techniques, it is possible to bring technological solutions to these arts work methods that are still used in archaic ways such as paper archivements with few or none security involved.

**Keywords:** Martial Arts, Teacher, Module, Integrated

## Índice

1 Introdução .....	9
1.1 Motivação.....	9
1.2 Objetivos .....	9
1.3 Organização do texto.....	10
2 A Proposta.....	11
3 Tecnologias .....	19
3.1 Tecnologias Utilizadas .....	19
3.1.1. MySQL Workbench.....	19
3.1.2. MySQL .....	20
3.1.3. Java.....	22
3.1.4. JPA .....	23
3.1.5. Hibernate.....	24
3.2 Estratégias de Desenvolvimento .....	25
3.2.1. Padrões de Projeto (Design Patterns) .....	25
3.2.2. Internacionalização .....	27
3.2.3. Encriptação.....	27
3.2.4. GUI (SWING).....	28
3.2.5. Sincronização.....	28
4 Implementação .....	30
4.1 Acesso .....	30
4.2 Gerenciamento de Lutas, Academias e Horários .....	32
4.2.1. Lutas .....	32
4.2.2. Aba de academias .....	33
4.2.3. Aba de Horários.....	34
4.3 Alunos .....	34
4.3.1. Informações gerais.....	35
4.3.2. Luta e graduações .....	35
4.3.3. Academias .....	36
4.3.4. Horários.....	36
4.3.5. Pagamentos.....	36

4.3.6.	Filtrar aluno .....	37
4.3.7.	Deletar aluno.....	37
4.4	Chamadas e Quadros de Horários .....	37
4.4.1.	Filtrar por luta .....	38
4.4.2.	Filtrar por academia .....	39
4.5	Registros de Professor.....	40
4.6	Registro de Alunos.....	41
4.7	Menu .....	41
4.7.1.	Configurações de Servidor .....	42
4.7.2.	Preferências.....	42
4.7.3.	Sincronização.....	43
4.7.4.	Sobre .....	44
5	Conclusão.....	45



# **1 Introdução**

## **1.1 Motivação**

Considerando que os principais e mais urgentes problemas relacionados ao sistema de artes marciais não são relacionados ao aprendizado, mas sim ao compartilhamento e documentação de informações e dados pertinentes às academias, foi elaborado um estudo de caso, e, a partir deste, tornou-se evidente a necessidade de se desenvolver um sistema capaz de possibilitar a comunicação integrada por parte de certas entidades.

O estudo de caso envolveu entrevistas com stakeholders, e teve o cuidado em identificar os meios e selecionar a tecnologia mais eficaz para então, sintetizar e informatizar as diretrizes burocráticas a fim de promover a melhoria do desempenho, fidelização e difusão de dados dos atuais métodos correntes.

Não há hoje uma validação de dados, históricos de graduação, dados de professores, ou integridade de informações obtidas entre uma localidade e outra, além da total ausência de dados em situações como: Dados de alunos mantidos em documentos manuscritos, difícil acesso de entidades externas a tais dados e diversos cadastros em duplicata de professores, gerados durante o procedimento de intercâmbio e/ou troca de instituição, que supostamente deveriam estar fortemente relacionados. Nunca é demais lembrar o peso e o significado destes problemas, uma vez que há um consenso sobre a necessidade do desenvolvimento do aplicativo entre os envolvidos.

## **1.2 Objetivos**

A partir da problemática apresentada anteriormente, desenvolvemos o conceito de um sistema integrado de quatro módulos: confederação, federação, associação e professor, cada um correspondente a uma das entidades do universo das lutas.

A confederação é a agregação de todas as federações dos estados ou regiões de uma nação. A federação, por sua vez, é responsável pela regulamentação das associações e é formada por representantes das mesmas. E as associações são as entidades ou clubes ao qual o professor está vinculado.

Cada um destes módulos visa resolver os problemas relacionados apenas àquela entidade, bem como definir as permissões das informações que cada módulo poderá ter.

Neste trabalho trataremos do módulo da entidade Professor, suas implicações, problemas relacionados, casos de uso, implementação e soluções.

A grande importância do módulo Professor se deve ao fato de que os professores serão os maiores responsáveis pelo abastecimento do sistema integrado. A partir de dados coletados e cadastrados pelos professores, outras entidades poderão gerar relatórios e acessar informações que sejam de seu interesse sobre seus professores cadastrados, e os alunos dos mesmos.

Professores serão os principais responsáveis por manter os dados de seus alunos sempre atualizados e coesos, para que sejam acessados sempre que necessário. Por ser a entidade que vai ocasionalmente se relacionar com a maior quantidade de entidades alvo, os alunos, faz sentido que seja a primeira entidade a ter uma solução implementada.

### **1.3 Organização do texto**

O presente trabalho está estruturado em capítulos e, além desta introdução, será desenvolvido da seguinte forma:

Capítulo II: apresenta a proposta de solução para o problema apresentado, bem como diagramas UML e o esquema do banco de dados.

Capítulo III: apresenta os conceitos das tecnologias envolvidas no projeto.

Capítulo IV: apresenta toda a implementação do software BlackBelt.

Capítulo V: apresenta as considerações finais, assinala as contribuições do projeto e sugere possibilidades de aprofundamento posterior.

## 2 A Proposta

A ideia principal deste projeto é utilizar e integrar toda a tecnologia disponível de forma a prestar toda a assistência necessária ao professor (e futuramente a todas as entidades envolvidas no sistema de artes marciais). Assim, o que anteriormente era realizado de forma arcaica através de anotações e cadastros em folhas de papel, agora poderá ser informatizado e automatizado, proporcionando maior praticidade durante cadastros e pesquisas de dados e segurança no armazenamento e troca de informações.

Dessa forma, não só o professor (que é o principal usuário deste sistema), como também alunos e posteriormente federações, confederações e associações de artes marciais poderão ser beneficiados com o software.

Como este trabalho apresenta exclusivamente o módulo Professor, iremos apresentar somente o seu objetivo, que é possibilitar aos professores de artes marciais:

- Criação e manutenção das lutas em que possui graduação.
- Criação e manutenção das academias nas quais ministra suas aulas.
- Criação e manutenção dos horários específicos para cada luta em cada academia.
- Criação e manutenção de seus alunos, e cadastro dos mesmos nas lutas, academias e horários predefinidos.
- Salvar e imprimir listas de chamadas por turmas e horários.
- Salvar e imprimir relatórios de alunos por academia.
- Verificação de seus próprios registros nas entidades de Federação, Confederação e Associação.
- Verificação dos registros de todos os seus alunos nas entidades de Federação, Confederação e Associação.
- Outras configurações e preferências do software

## Diagramas UML

### 1 - Criação e manutenção das lutas em que possui graduação.

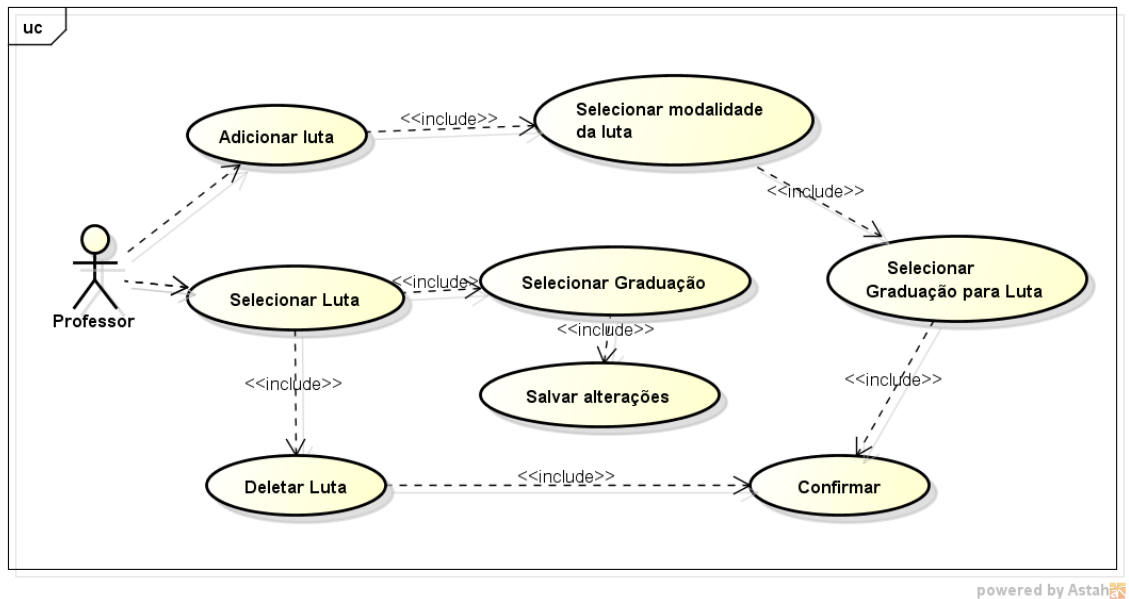


Figura 1: Criação e manutenção de lutas

### 2 - Criação e manutenção das academias nas quais ministra suas aulas.

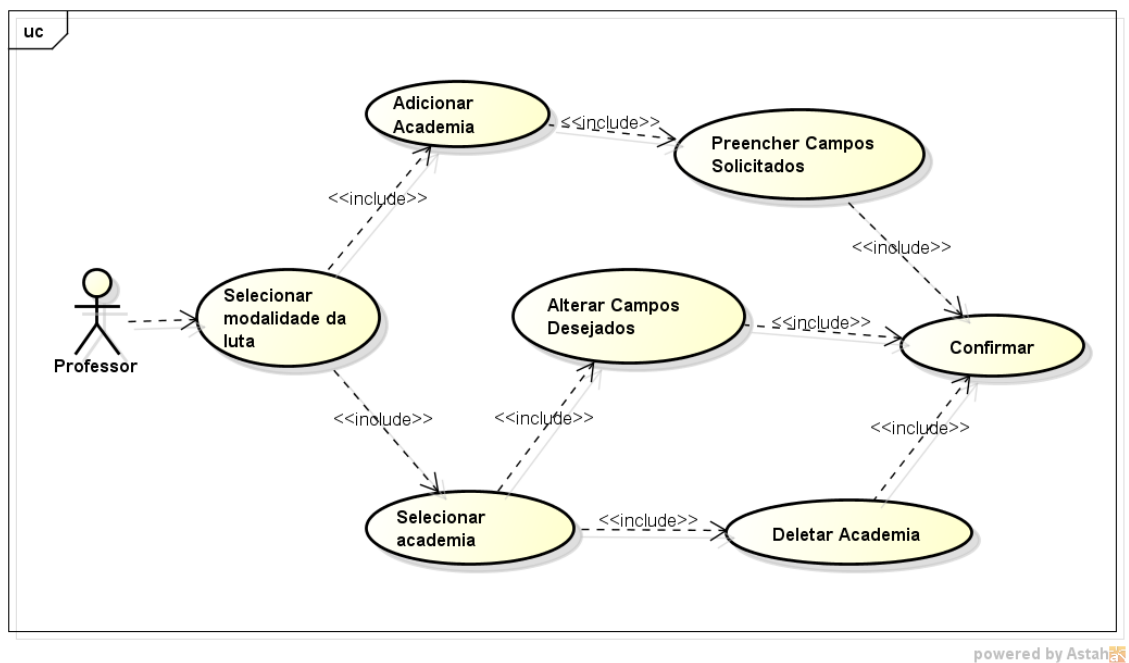


Figura 2: Criação e manutenção de academias

### 3 - Criação e manutenção dos horários específicos para cada luta em cada academia.

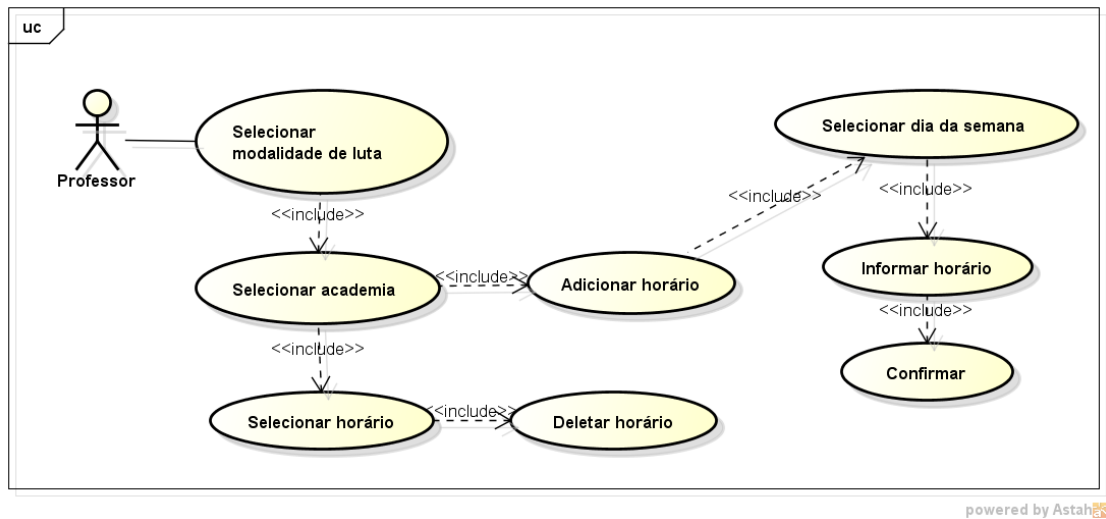


Figura 3: Criação e manutenção do horário para lutas

### 4 - Criação e manutenção de seus alunos, e cadastro dos mesmos nas lutas, academias e horários predefinidos.

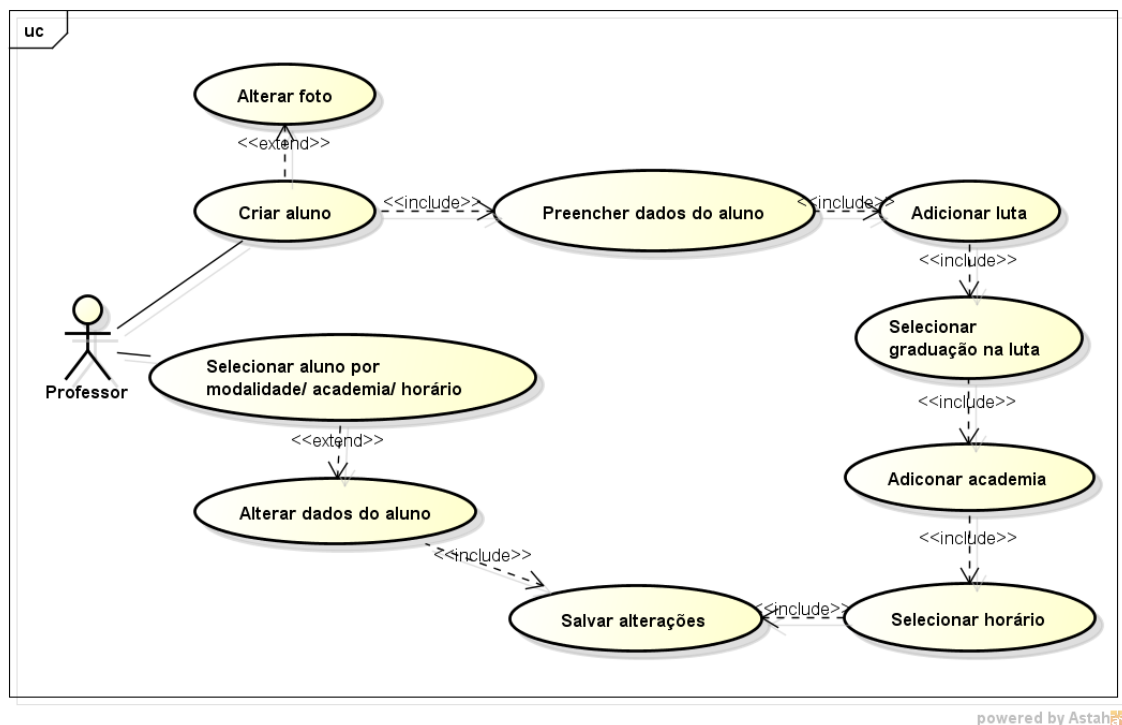
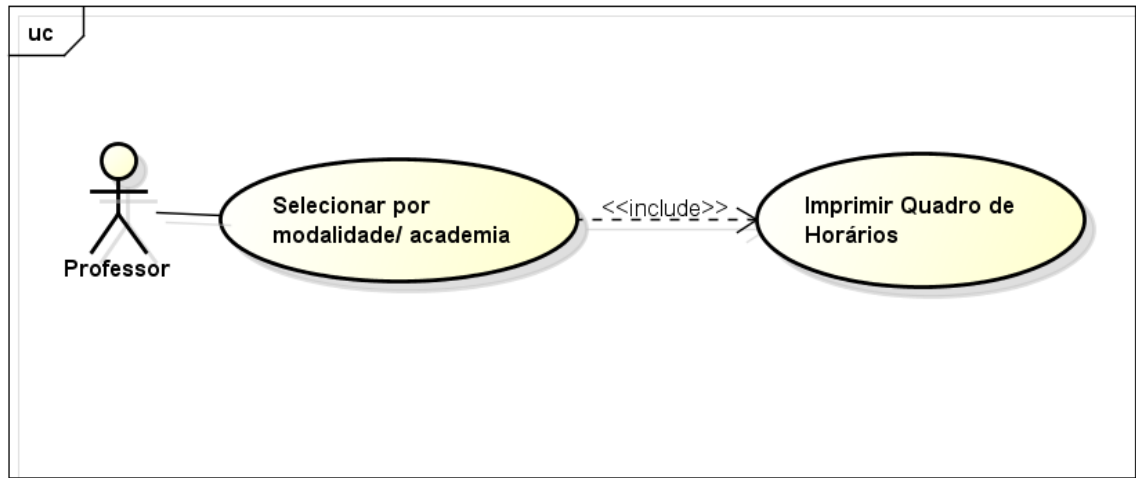


Figura 4: Criação e manutenção de alunos

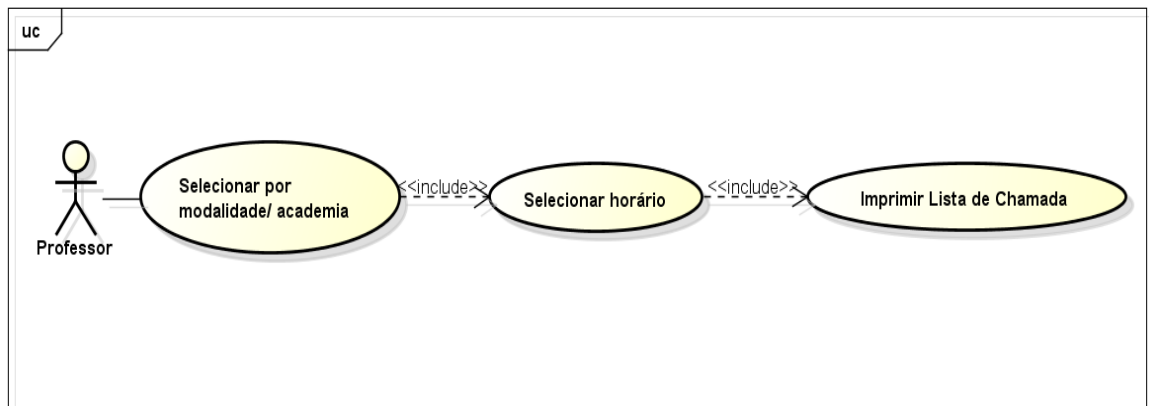
## 5 - Salvar e imprimir listas de chamadas por turmas e horários.



powered by Astah

Figura 5: Criação e manutenção de alunos

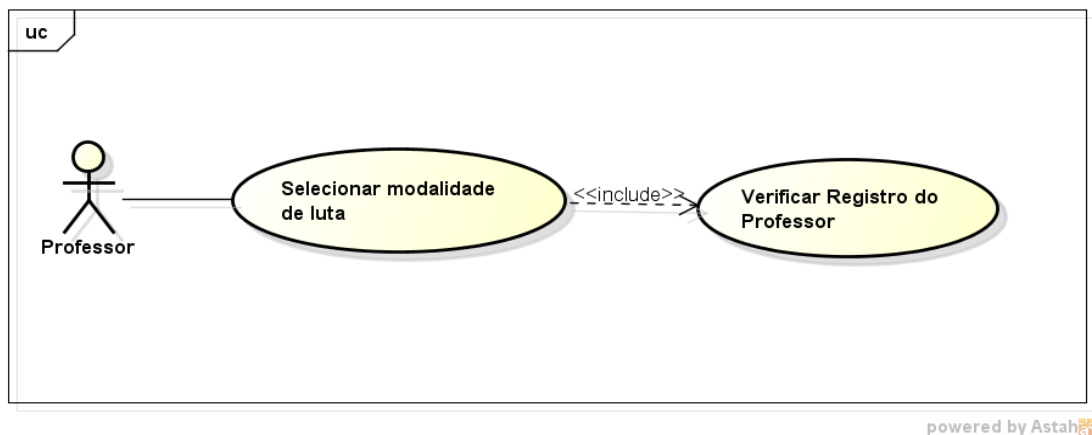
## 6 - Salvar e imprimir relatórios de alunos por academia.



powered by Astah

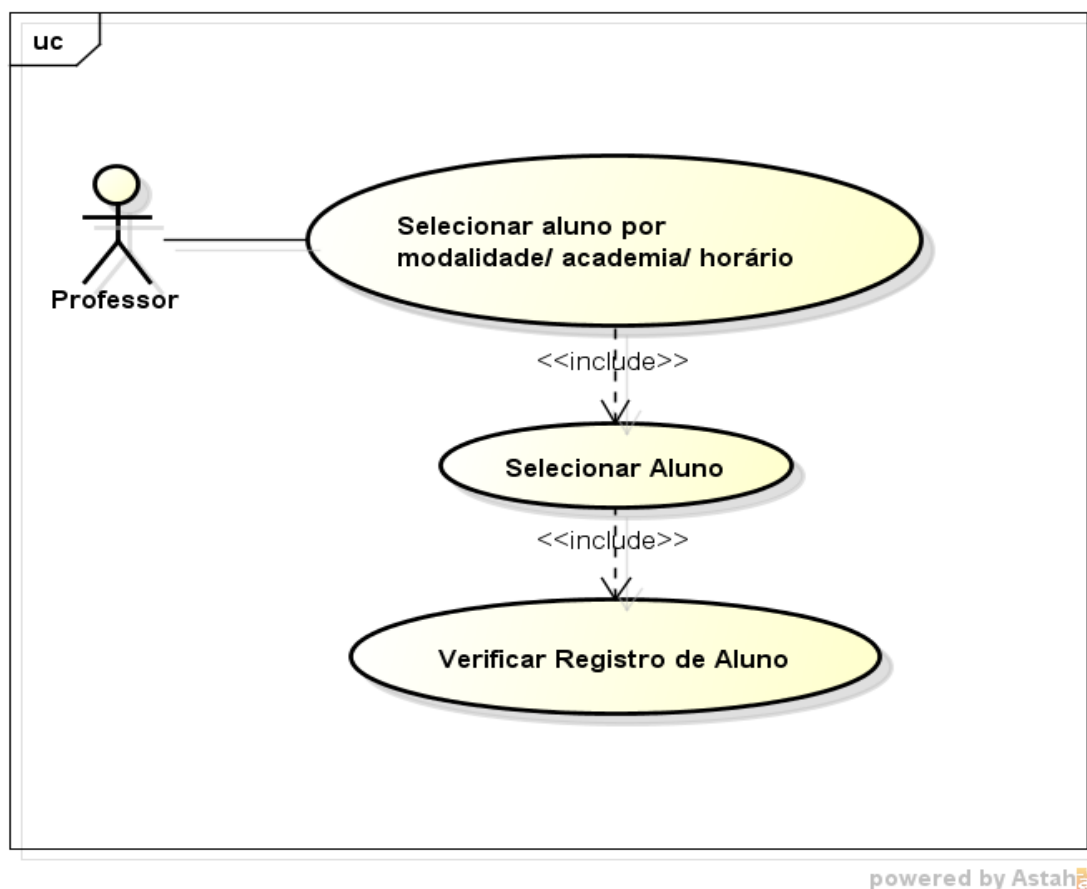
Figura 6: Salvar e imprimir relatórios de alunos

**7 - Verificação dos registros de Professor nas entidades de Federação, Confederação e Associação.**



**Figura 7: Verificação dos registros de professor**

**8 - Verificação dos registros de todos os seus alunos nas entidades de Federação, Confederação e Associação.**



**Figura 8: Verificação dos registros de alunos**

## Diagrama Geral (Todos os Módulos)

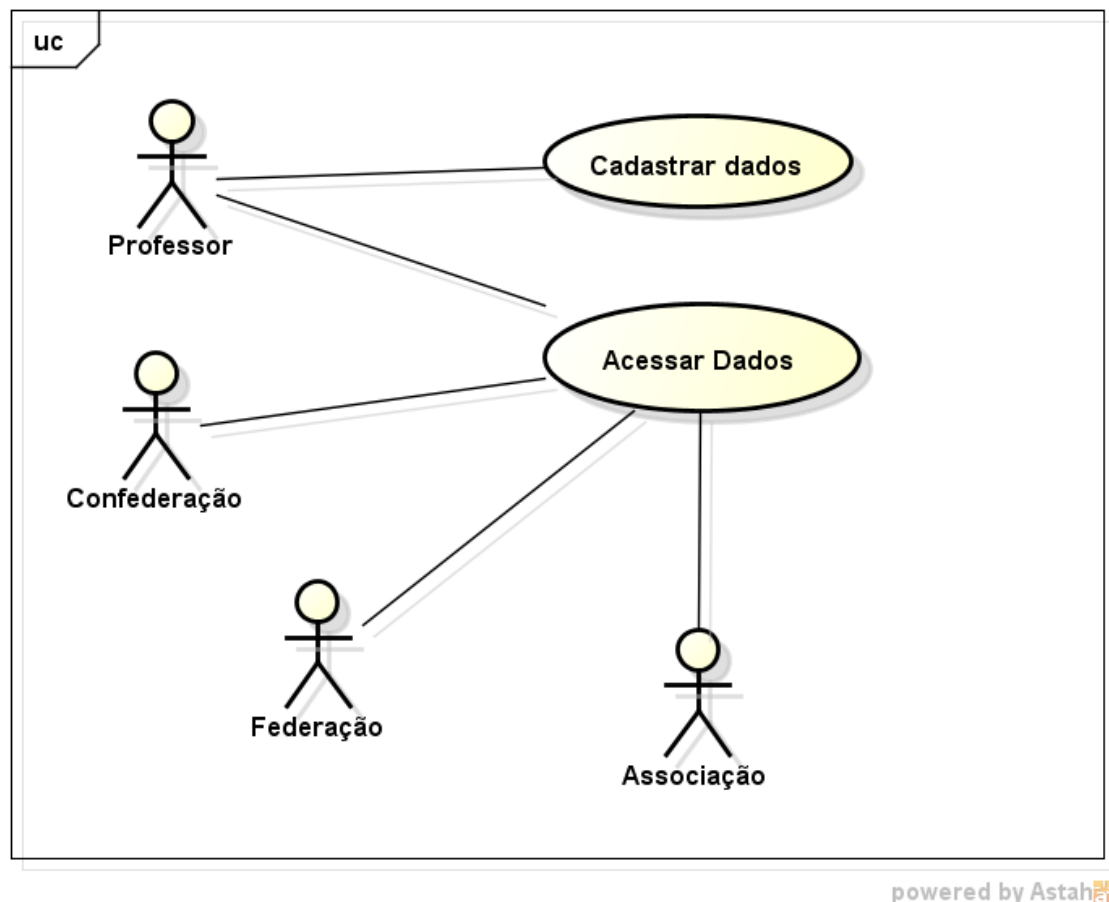


Figura 9: Verificação dos registros de alunos

### Esquema do Banco:

O esquema de banco de dados apresentado na figura 10 foi gerado pelo MySQL Workbench e se refere à versão final do módulo professor utilizada no desenvolvimento do sistema BlackBelt. No primeiro momento de implementação do banco os primeiros dados a serem carregados foram a hierarquia completa de estados, cidades, bairros e códigos postais.

A entidade Lutas do banco diz respeito a uma determinada luta num certo país. Esta decisão foi tomada porque lutas têm conjuntos diferentes de estilos entre si em diferentes países. Um país possui confederações, que por sua vez possuem federações. Estas federações estão associadas a um ou mais estilos de lutas e possuem Associações que por sua vez também estão associadas a um Estilo. Cada Estilo de Luta possui seu próprio conjunto de Graduações, que definem o nível de um Aluno dentro daquela Luta.



Um Professor pode estar associado a Associações, Federações e Confederações em apenas uma de cada uma destas entidades por Luta que ministra. Professores ministram aulas de determinados Estilos de Luta em Academias e em determinados Horários. Possuem também Alunos aos quais podem cadastrar, além das informações básicas, pagamentos para um determinado conjunto de Horários.

Alunos possuem Graduações de Luta para cada uma das Lutas em que um Professor o cadastrou, estão associados a academias e horários cadastrados pelo professor e podem ou não estar associados às entidades de Associação, Federação ou Confederação, pelas mesmas regras com as quais os Professores se relacionam com estas entidades.

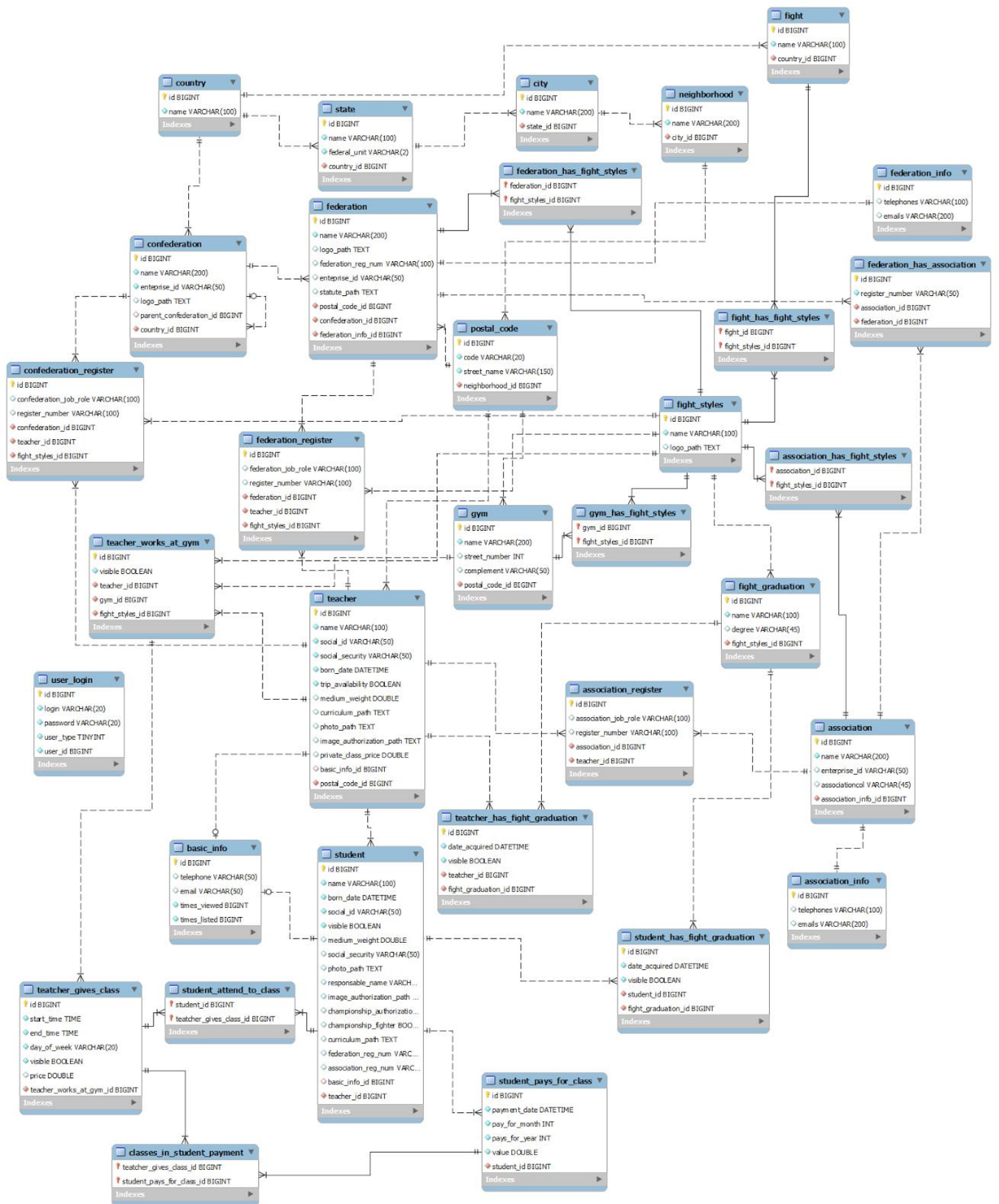


Figura 10: Esquema de banco de dados

## 3 Tecnologias

### 3.1 Tecnologias Utilizadas

#### 3.1.1. MySQL Workbench

O MySQL Workbench é uma ferramenta para modelagem e teste do banco de dados MySQL. O Workbench possui uma interface que permite planejar e desenvolver a solução da diagramação de maneira gráfica, o que possibilita uma fácil implementação e verificação simultânea do funcionamento dessa modelagem.

Dentro do MySQL Workbench temos as seguintes funcionalidades:

- **EER Diagram:** Apresenta todos os diagramas disponíveis para o banco de dados em questão. Ele permite criar, editar e modularizar os diagramas. Isso permite à ferramenta suportar múltiplos diagramas.
- **Physical Schemata:** Apresenta todos os objetos disponíveis para criação, bem como a listagem dos que já foram criados de acordo com seu tipo. Se o objeto for uma tabela, ficará categorizado abaixo do item “Tables”, análogo como ocorre para os outros itens (Views no item de “Views” e assim por diante).
- **SQL Script's:** É uma área onde pode ser criado um script SQL adicional para o banco ou carregar um arquivo externo criado anteriormente. É útil porque possibilita carregar o banco uma vez e cadastrar alguns usuários, mas você quer garantir que os usuários permaneçam lá caso você queira resetar o banco. Podemos exportar o SQL dos inserts<sup>1</sup> e carregá-lo nesta área e quando gerarmos o script final ele vai ser colocado junto.

---

<sup>1</sup> Dados que serão inseridos ao banco por script SQL.

- **Model Notes:** É uma área similar a um bloco de notas sobre o banco de dados onde podem ser documentados detalhes sobre o banco para consultas futuras como, por exemplo, regras de campos.
- **Visualização total do Dashboard:** Uma espécie de miniatura de todo o projeto onde podemos navegar entre todas as páginas que compõem o diagrama com facilidade e rapidez.
- **Catálogo do Banco de dados:** Uma apresentação de forma hierárquica dos objetos que compõem o banco de dados em produção. Na mesma guia podemos encontrar mini abas na parte inferior que possuem a finalidade de controlar os **Layers** que são grupos de objetos dentro do diagrama demarcados por uma área colorida. Contamos também com o **User Types** caso você venha a precisar de um tipo de dado específico que não tenha nas opções do programa. Nesta guia podemos criar um novo tipo de dado totalmente personalizado de acordo com necessidades específicas.
- **Informações Gerais sobre o Objeto selecionado:** Exibe informações sobre o objeto que estiver selecionado naquele exato momento seja ele uma tabela, view ou rotina.

### 3.1.2. MySQL

MySQL é um sistema de gerenciamento de banco de dados relacional que foi considerado, de acordo com o ranking publicado pela DB-Engines, o segundo SGBD mais popular, sendo amplamente utilizado por grandes corporações como, NASA, Google e Facebook.

Este sistema utiliza as linguagens C e C++, possui código aberto e é desenvolvido e distribuído por meio de duas licenças que dependem do tipo de sua utilização:

- Software livre baseado nas cláusulas da GNU-GPL (General Public Licence) onde é definido o que se pode ou não fazer com a ferramenta e seus recursos (utilização, distribuição, didática e colaboração).
- E comercial para situações onde os termos da GNU-GPL não se aplicam, como inclusão do MySQL em aplicações comerciais, ou para pacotes com mais ferramentas, suporte diferenciado, etc.

Seu desenvolvimento foi iniciado na Suécia em 1994, pela Companhia MySQL AB - fundada por dois suecos (David Axmark e Allan Larsson) e um finlandês (Michael "Monty" Widenius) - tendo sua primeira versão disponível em 23 de maio de 1995. Seu objetivo inicial era otimizar a utilização do mSQL baseado na linguagem de baixo nível ISAM, que era considerada inflexível e muita lenta. Mantendo-se a mesma API do mSQL, foi criada uma nova interface SQL, agradando desde então desenvolvedores, que puderam substituir o software licenciado pelo novo sistema "open source"<sup>2</sup>.

Após processos e disputas judiciais entre as empresas MySQL AB e NuSphere, envolvendo basicamente quebras de contrato e violação de direitos autorais, em 16 de janeiro de 2008, a companhia Sun Microsystems adquiriu a MySQL AB por US\$ 1 bilhão, enquanto em paralelo a Oracle adquiria diversas empresas desenvolvedoras de mecanismos cujas abrangências melhoravam as funcionalidades do MySQL. Finalmente, em 20 de abril de 2009, a Oracle anuncia a compra da então detentora do MySQL, Sun.

O Departamento de Justiça e Defesa da Concorrência dos Estados Unidos acabou por pressionar a aprovação da compra, à pedido da Oracle, e a Comissão Europeia acabou autorizando a aquisição do MySQL pela Oracle em 21 de janeiro de 2010.

O MySQL já possui 19 anos e, mesmo tendo sido projetado inicialmente para trabalhar com aplicações de pequeno e médio porte (em torno de 100 milhões de registros por tabela - aproximadamente 100MB por tabela), hoje atende aplicações de grande porte, concorrendo até mesmo com programas similares de código fechado (como SQL Server e Oracle).

Existem diversas vantagens na utilização do MySQL, entre elas:

- Compatibilidade com praticamente todos os sistemas operacionais do mercado (entre eles Linux, Unix, FreeBSD, MAC OS X Server e Windows);
- Ferramenta altamente portátil entre diferentes sistemas, plataformas e compiladores. Além disto, possui drivers como ODBC, JDBC e .NET e módulos de interfaces para as diversas linguagens de programação existentes (C, C++, Java, Delphi, PHP, etc);

---

<sup>2</sup> Código aberto. Os arquivos fonte podem ser baixados e alterados por qualquer pessoa.

- Maior velocidade para acesso aos dados, pois utiliza cachês em suas consultas, indexação BTREE para tabelas HEAP, algoritmos de buscas, entre outros recursos;
- Como utiliza multithreads, possui facilidades para integração entre hardwares com mais de uma CPU, além de também aumentar consideravelmente a velocidade de processamento;
- Além de possuir um sistema gerenciador de conexões que criptografa as senhas, apresenta uma espécie de firewall que habilita conexões apenas para estações de domínios especificados em sua lista de acessos, o que o torna extremamente seguro e confiável;
- Suporta Triggers, Cursors, controle transacional, Stored Procedures e Functions;

### **3.1.3. Java**

O Java é uma linguagem de programação orientada a objetos que tem como uma de suas principais características (e também vantagens) a independência de plataformas, ou seja, uma vez compilado o código em determinada plataforma, não será necessário compilá-lo novamente caso decida-se adotar alguma outra plataforma.

Foi criado em junho de 1991, pela Sun Microsystems, o Green Project, composto por Patrick Naughton, Mike Sheridan, e James Gosling - este último conhecido como o pai da linguagem de programação Java.

A primeira criação desse time foi um controle remoto com interface touchscreen capaz de controlar vários dispositivos e aplicações chamado \*7 (StarSeven). Este equipamento possuía um mascote com o nome de Duke (um guia virtual hoje muito popular no mundo Java), que auxiliava na utilização do controle. A linguagem de programação utilizada, criada por James Gosling, foi denominada de "Oak" (Carvalho) devido a uma árvore em que o mesmo podia observar em frente a sua janela.

Ainda que hoje esta ideia de interação já seja possível e real, naquela época ainda era visto como algo muito visionário e as empresas de TV a cabo não possuíam tecnologia suficiente para entender e comprar esta proposta. "A ideia certa, na época errada".

Enquanto isso, no entanto, a internet vinha crescendo e se expandindo cada vez mais, tornando-se o ambiente ideal para a equipe do Green Project investir, visto que

toda a infraestrutura necessária já existia. E em janeiro de 1995 foi lançada a nova versão do Oak adaptada para a internet, agora chamada de Java ("Java Coffee", café consumido pela equipe em grandes quantidades durante seu projeto).

Utilizando uma sintaxe similar ao C/C++, Java era uma promessa de linguagem WORA ("Write Once, Run Anywhere" - Escreva uma vez, rode em qualquer lugar). Graças a sua portabilidade e dinâmica tecnologia, logo os principais browsers (como o Netscape Navigator) começaram a executar Applets Java em suas páginas web e grandes empresas (como a IBM) passaram a também utilizar aplicativos em Java em seus produtos, prestando suporte para os mesmos.

A plataforma Java foi adotada mais rapidamente que qualquer outra linguagem de programação na história da informática e se tornou popular tão rapidamente que desde seu lançamento até 2004 já detinha três milhões de desenvolvedores em todo o mundo.

Java utiliza o conceito de máquina virtual e bytecode, o que o difere de outras linguagens. Em linguagens tradicionais, cada plataforma utiliza um compilador diferente, e mesmo que o código fonte seja escrito de forma facilmente portátil, ainda existe a necessidade de o recompilar, gerando binários diferentes em cada utilização. Já no Java, é necessário gerar apenas um binário que será executado em qualquer plataforma/sistema operacional. Este binário é chamado de bytecode.

Para que isso seja possível, é necessário que a plataforma em questão esteja utilizando uma máquina virtual, conhecida como JVM (Java Virtual Machine). Ela será responsável por converter o bytecode em comandos que o sistema operacional consiga executar. Desta forma, apenas a máquina virtual deve ser modificada para cada plataforma, o que é feito pelos desenvolvedores da linguagem e não pelo programador.

Entretanto, esta técnica também apresenta desvantagens. Por ser necessária a utilização de máquinas virtuais, o processamento delas pode tornar o programa mais pesado do que no caso de ter sido escrito diretamente para a plataforma.

#### **3.1.4. JPA**

Utilizar banco de dados com a linguagem Java nem sempre foi uma tarefa fácil. Ainda que possua um padrão (ANSI), SQL possui variações de acordo com seus fabricantes, e a troca deles em geral se torna complicada. Além disso, banco de dados

utilizam modelos de entidade e relacionamento, onde a representação das informações é feita através de tabelas e colunas e estas são relacionadas através de chaves estrangeiras. Já as linguagens de programação orientadas a objeto utilizam classes e atributos para representar as informações e estas são interligadas por meio de conceitos como herança, polimorfismo, etc. Sendo assim, para a utilização do Java e do SQL juntos, faz-se necessário sempre adaptar objetos e registros.

Para otimizar o tempo gasto com essas adaptações foi criado o Java Persistence API ou simplesmente JPA, uma API (Application Programming Interface) da linguagem Java que descreve um meio de gerenciar dados relacionais em aplicações que utilizem Java.

Existem algumas ferramentas conhecidas hoje em dia que implementam o JPA, como: Enterprise JavaBeans, Java Data Objects API, Service Data Object API e Hibernate. Neste trabalho iremos descrever apenas a ferramenta Hibernate.

### **3.1.5. Hibernate**

O Hibernate é um framework open source de mapeamento objeto-relacional (ORM, do inglês Object-relational mapping). Esta técnica consiste em transformar as tabelas (utilizadas em bancos de dados) em classes (utilizadas nas linguagens de programação orientada a objeto). Dessa forma, cada registro de cada tabela corresponderá a uma instância da classe relacionada. Assim, o desenvolvedor não precisa se preocupar com a linguagem SQL, precisando apenas definir a relação que terá entre as tabelas e as classes (através de um código separado, possibilitando assim que possíveis alterações no banco de dados não interfiram no programa). Cada ferramenta ORM utiliza uma forma de mapear, no caso do Hibernate é utilizado arquivos XML ou anotações Java (@).

No Hibernate existem duas possibilidades de linguagem: o próprio SQL, ou a HQL (Hibernate Query Language), uma linguagem de consulta que se assemelha ao SQL, porém ao invés de utilizar tabelas e colunas, é totalmente orientada a objetos.



## 3.2 Estratégias de Desenvolvimento

### 3.2.1. Padrões de Projeto (Design Patterns)

Os padrões de projeto, mais conhecidos na área de desenvolvimento como Design Patterns, são estratégias de modularização do código e da estrutura de chamadas dos objetos de uma determinada linguagem de programação (geralmente orientada a objetos) que objetivam facilitar os processos relacionados à codificação (rapidez na codificação, reusabilidade, manutenção do código, etc). Não existe regra para a criação destes padrões, mas alguns são tão efetivos que se popularizaram e são utilizados em projetos por toda a comunidade de desenvolvimento. Para o desenvolvimento do BlackBelt foram utilizados três padrões de projetos na estrutura do Java: MVC, Singleton e Factory.

O padrão Singleton garante a existência de apenas uma instância de uma classe, mantendo um ponto global de acesso ao seu objeto. O Padrão MVC (Model-View-Controller), é um modelo de arquitetura de desenvolvimento que separa e modulariza as classes responsáveis pelos modelos de dados (model), que são as classes que representam os dados no banco, as visões (view) que são a parte visual de um software e os controladores (controller) que gerenciam as funcionalidades presentes nas visões. O padrão Factory permite às classes java delegar funcionalidades para que subclasses desta primeira decidam como agir.

Cada estrutura de pastas do BlackBelt (arquivos descompactados do arquivo .zip que deve ser baixado no site) é uma instância do software. O BlackBelt não é instalado na máquina, o que permite que o software possua múltiplas instâncias e estruturas de pasta dentro de uma mesma máquina.

Este sistema possui dois modos, online e off-line. É possível que um professor cadastre um usuário chamado de “Usuário Premium” uma vez que esteja logado. Cada instância do BlackBelt pode ter apenas um usuário premium cadastrado. Ao cadastrar um Usuário Premium, um Professor será capaz de utilizar o modo off-line, similar em quase todos os aspectos ao modo online, excluindo-se apenas algumas funcionalidades de menus e a integração com os outros módulos. É possível gerenciar lutas, academias, horários, alunos e gerar todos os relatórios e listas de chamada, para que todas as novas informações sejam sincronizadas posteriormente quando o Professor fizer login online,

ou no momento em que for mais conveniente com o uso da funcionalidade “Sincronizar Agora” contida no menu Sincronização (disponível apenas no módulo off-line). Isso possibilitará aos Professores levar notebooks à locais de trabalho onde não há acesso à internet e trabalhar normalmente, podendo sincronizar tudo posteriormente.

Para conseguir criar um sistema de sincronização eficiente, foi necessário primeiro escolher uma estratégia de armazenamento, modificação ou deleção dos objetos que são utilizados no modo off-line. Optamos por utilizar a técnica da serialização do Java. A serialização consiste em transformar em arquivos serial (.ser) os objetos utilizados pelo Java e armazenar em pastas separadas pelo tipo de objeto. Quando um Usuário Premium está utilizando o BlackBelt no modo online, tudo que ele faz está sendo salvo automaticamente no banco e similarmente na máquina local nesta estrutura de pastas serializadas. Ao utilizar o modo off-line e realizar uma sincronização ou realizar um login, o BlackBelt possui uma lógica que vai determinar qual é a ordem das modificações que vão ser realizadas no banco:

1 - Quais são os arquivos serializáveis que não existem mais?

- Primeiramente, todos os objetos que já foram removidos deverão ser removidos do banco de dados.

2 - Quais são os arquivos serializáveis que já existiam?

- Todos os objetos já existentes sofrem um processo automático de update, para que novas informações sobre eles sejam atualizadas caso existam.

3 - Quais são os arquivos serializáveis novos?

- Os arquivos serializáveis que ainda não tem seus correspondentes no banco de dados serão sempre inseridos.

Para fazer a validação de correspondência entre um objeto serializado e o seu correspondente no banco de dados, os arquivos serializados são salvos com o seu nome sendo a criptografia Hash do seu id do banco.

Foi necessário também determinar de acordo com o esquema de banco de dados, qual a ordem ideal de sincronização em cada tipo de processo para que não haja falhas no processo, ou seja, deleção, atualização e inserção têm ordens ideais diferentes do fluxo de sincronização por tabelas para que não haja impactos no processo como um todo. Esta ordem é determinada de acordo com as funcionalidades do software em conjunto com a relação de dependências das tabelas no banco de dados.

### **3.2.2. Internacionalização**

O BlackBelt está disponível a princípio em três línguas: Português, Inglês e Espanhol. Para possibilitar um fácil sistema de internacionalização, foi utilizado o padrão de projeto Factory com arquivos de propriedades (.properties) sobre a camada MVC do projeto.

Os arquivos de propriedades do Java são arquivos que guardam simplesmente o nome de uma determinada propriedade e seu respectivo valor. Este tipo de arquivo permite fácil atribuição de valores que serão comuns ao software e que não tendem a sofrer mudanças constantes em tempo de execução.

Para realizar o processo de internacionalização, definimos todas as chamadas aos nomes dos atributos em uma língua padrão (inglês no nosso caso), e para cada arquivo de propriedades de línguas inserimos este atributo como o nome de uma propriedade. A essa propriedade daremos o valor daquela sentença traduzida para os arquivos. Em outro arquivo de propriedade definimos qual é a língua utilizada no momento, então para realizar a tradução bastou criar um handler que seleciona a língua corrente e nomeia todos os campos do software com as propriedades contidas no arquivo para o qual ele foi direcionado.

### **3.2.3. Encriptação**

Para garantir a segurança no sistema de senhas dos usuários foi utilizada uma encriptação utilizando o algoritmo SHA-512, uma função de hash criptográfica da família SHA-2.

Com a necessidade de organizar grandes volumes de dados e manter sua integridade, foi criada uma forma para, através de cálculos matemáticos, estrutura-las de modo que possibilite estrutura-las em um mapeamento entre valores de chaves e entradas na tabela. Resultado destes cálculos é chamado de Hash e é matematicamente inviável através do valor gerado retornar ao dado original.

### 3.2.4. GUI (SWING)

O estilo de interface visual do BlackBelt foi baseado no Java SWING.

Inicialmente a única API que o Java suportava era AWT, porém posteriormente esta foi ultrapassada pela SWING (a partir do Java 1.2). Chegando as duas bibliotecas gráficas suportadas oficialmente pelo Java hoje, pertencentes a qualquer JRE ou JDK.

A conhecida ideologia de portabilidade da plataforma Java é facilitada pelas interfaces gráficas que este oferece. *Look-and-feel* é a denominação da aparência de uma interface. No Swing ele se comporta da mesma forma independentemente do Sistema Operacional que esteja sendo executado, logo a aplicação será exibida igualmente em todos os formatos e tamanhos de tela.

Classes e métodos do Swing foram desenvolvidos com foco na maior portabilidade possível e isto resultou em boa parte da complexidade da API. Grande parte da complexidade das classes e métodos do Swing está no fato da API ter sido desenvolvida tendo em mente o máximo de portabilidade possível. Favorece-se, por exemplo, o posicionamento relativo de componentes, em detrimento do uso de posicionamento fixo, que poderia prejudicar usuários com resoluções de tela diferentes da prevista.

### 3.2.5. Sincronização

Cada estrutura de pastas do BlackBelt (arquivos descompactados do arquivo .zip que deve ser baixado no site) é uma instância do software. O BlackBelt não é instalado na máquina, o que permite que o software possua múltiplas instâncias e estruturas de pasta dentro de uma mesma máquina.

Este sistema possui dois modos, online e off-line. É possível que um professor cadastre um usuário chamado de “Usuário Premium” uma vez que esteja logado. Cada instância do BlackBelt pode ter apenas um usuário Premium cadastrado. Ao cadastrar um Usuário Premium, um Professor será capaz de utilizar o modo off-line, similar em quase todos os aspectos ao modo online, excluindo-se apenas algumas funcionalidades de menus e a integração com os outros módulos. É possível gerenciar lutas, academias, horários, alunos e gerar todos os relatórios e listas de chamada, para que todas as novas informações sejam sincronizadas posteriormente quando o Professor fizer login online, ou no momento em que for mais conveniente com o uso da funcionalidade “Sincronizar

Agora” contida no menu Sincronização (disponível apenas no módulo off-line). Isso possibilitará aos Professores levar notebooks à locais de trabalho onde não há acesso à internet e trabalhar normalmente, podendo sincronizar tudo posteriormente.

Para conseguir criar um sistema de sincronização eficiente, foi necessário primeiro escolher uma estratégia de armazenamento, modificação ou deleção dos objetos que são utilizados no modo off-line. Optamos por utilizar a técnica da serialização do Java. A serialização consiste em transformar em arquivos serial (.ser) os objetos utilizados pelo Java e armazenar em pastas separadas pelo tipo de objeto. Quando um Usuário Premium está utilizando o BlackBelt no modo online, tudo que ele faz está sendo salvo automaticamente no banco e similarmente na máquina local nesta estrutura de pastas serializadas. Ao utilizar o modo off-line e realizar uma sincronização ou realizar um login, o BlackBelt possui uma lógica que vai determinar qual é a ordem das modificações que vão ser realizadas no banco:

1 - Quais são os arquivos serializáveis que não existem mais?

- Primeiramente, todos os objetos que já foram removidos deverão ser removidos do banco de dados.

2 - Quais são os arquivos serializáveis que já existiam?

- Todos os objetos já existentes sofrem um processo automático de update, para que novas informações sobre eles sejam atualizadas caso existam.

3 - Quais são os arquivos serializáveis novos?

- Os arquivos serializáveis que ainda não tem seus correspondentes no banco de dados serão sempre inseridos.

Para fazer a validação de correspondência entre um objeto serializado e o seu correspondente no banco de dados, os arquivos serializados são salvos com o seu nome sendo a criptografia Hash do seu id do banco.

Foi necessário também determinar de acordo com o esquema de banco de dados, qual a ordem ideal de sincronização em cada tipo de processo para que não haja falhas no processo, ou seja, deleção, atualização e inserção têm ordens ideais diferentes do fluxo de sincronização por tabelas para que não haja impactos no processo como um todo. Esta ordem é determinada de acordo com as funcionalidades do software em conjunto com a relação de dependências das tabelas no banco de dados.

## 4 Implementação

### 4.1 Acesso

Para ter acesso ao sistema BlackBelt, primeiro um professor deverá realizar um cadastro no site [BlackBeltArts](#), confirmar seu cadastro por e-mail e preencher as informações do perfil que deseja ter acesso. Ao realizar o cadastro no site, ele terá acesso ao download do software BlackBelt através dos links de download conforme a Figura 11 abaixo:



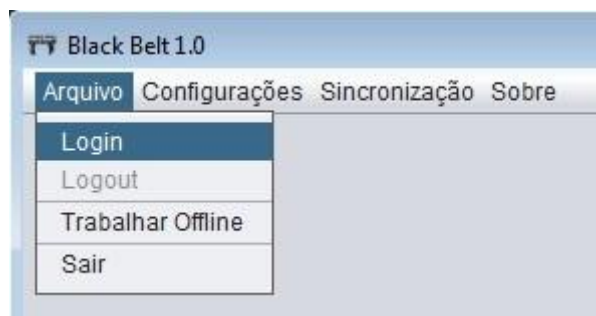
**Figura 11: BlackBelt Download**

O Professor deve escolher uma pasta de fácil acesso e mover o arquivo baixado para ela e em seguida descompactar o arquivo baixado com qualquer descompactador de preferência. Para executar o software, basta um duplo clique no arquivo nomeado como “Black Belt, a versão atual do software e extensão .jar”. Importante: as demais pastas contém importantes arquivos de configuração e outras usabilidades do sistema

que não devem ser movimentados ou deletados incluindo quaisquer subpastas ou arquivos abaixo delas.

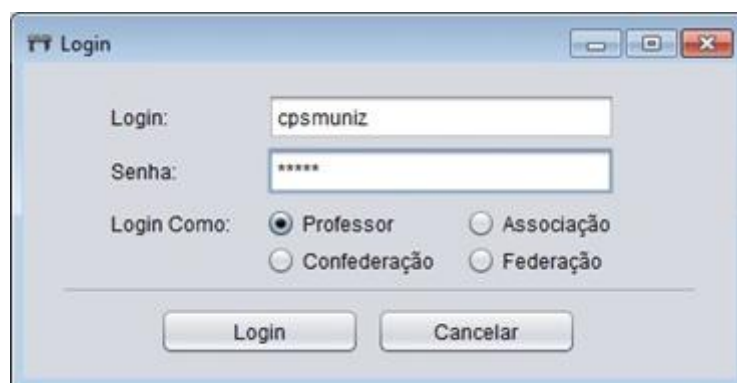
Os Professores podem deixar a pasta escondida, e criar um link pessoal na área de trabalho ou em outro local de preferência, basta mover a pasta extraída normalmente, dar um clique simples no arquivo nomeado como “Black Belt, a versão atual do software e extensão .jar” e copiar. Em seguida, na área de trabalho, ou outra pasta de preferência, dar um clique simples e escolher a opção “Colar atalho”.

Ao acessar o arquivo, um Professor deverá realizar o login no modo online ou off-line para ter acesso às funcionalidades do sistema. Estas modalidades estão disponíveis no menu “Arquivo” como mostra a figura 12:



**Figura 12: Métodos de login**

Para realizar um login do tipo “Trabalhar Off-line” é necessário que o professor tenha cadastrado um Usuário Premium, área que só é possível ser acessada após realizar um login online (cobriremos este processo de cadastro posteriormente). Para realizar um login tradicional, um Professor deverá utilizar seu usuário e senha criados no site do BlackBelt e selecionar o tipo de login como tipo professor assim como na Figura 13:



**Figura 13: Login de Professor**

## 4.2 Gerenciamento de Lutas, Academias e Horários

Para iniciar suas atividades no sistema, o professor deverá cadastrar as lutas as quais ele ministra aulas com suas respectivas graduações, as academias para as quais ele ministra aulas de determinadas lutas, e os horários para os quais cada uma destas lutas é ministrada dentro de uma determinada academia como podemos verificar na Figura 14:

The screenshot shows the 'Black Belt 1.0' application window. The menu bar includes 'Arquivo', 'Configurações', 'Sincronização', and 'Sobre'. The tab bar at the top has 'Alunos', 'Lutas, Academias e Horários' (selected), 'Chamadas e Quadro de Horários', 'Registros De Professor', and 'Registros De Alunos'. The main area is divided into three panels. The left panel, titled 'Lutas:', contains a large empty box, a 'Graduação:' dropdown, and buttons for 'Deletar Luta', 'Salvar Alterações', and 'Adicionar Luta'. The middle panel, titled 'Lutas:', contains a dropdown, an 'Academias:' dropdown, a large empty box, buttons for 'Deletar Academia' and 'Adicionar Academia', and input fields for 'CEP:', 'Nome:', 'Número:', and 'Complemento:'. The right panel, titled 'Lutas:', contains a dropdown, an 'Academias:' dropdown, a large empty box, buttons for 'Deletar Horário' and 'Adicionar Horário', and a time selection area with 'Confirmar' buttons.

**Figura 14: Tela “Lutas, Academias e Horários”**

### 4.2.1. Lutas

Para gerenciar suas lutas, um professor deverá utilizar a subdivisão à esquerda da tela “Lutas. Academias e Horários” como mostra a figura 15:



Lutas:

Graduação:

Deletar Luta

Salvar Alterações

Adicionar Luta

Confirmar

**Figura 15: Gerenciamento de Lutas**

Nesta subdivisão, um Professor será capaz de gerenciar as lutas às quais ministra aulas ou apenas é graduado, alterar as graduações das lutas em que realizou cadastro e deletar suas lutas.

#### **4.2.2. Aba de academias**

A aba de academias é a subdivisão ao centro da tela “Lutas. Academias e Horários” como mostra a figura 16:

Lutas:

Academias:

Deletar Academia

Adicionar Academia

CEP:

Nome:

Número:

Complemento:

Confirmar

**Figura 16: Gerenciamento de Academias**

Nesta subdivisão, um Professor será capaz de gerenciar os horários em que ele ministra suas aulas de acordo com as academias ou instituições às quais ele se cadastrou em cada uma de suas lutas.

#### 4.2.3. Aba de Horários

A aba de horários é a aba ao centro da tela “Lutas. Academias e Horários” como mostra a figura 17:

A interface de gerenciamento de horários é apresentada em um formulário vertical. No topo, há um campo rotulado "Lutas:" com uma seta para baixo. Abaixo dele, um campo rotulado "Academias:" também com uma seta para baixo. O campo principal é rotulado "Horários:" e contém uma área vazia para a lista de horários. Abaixo desta área, há dois botões: "Deletar Horário" e "Adicionar Horário". Logo abaixo dos botões, há um campo com uma seta para baixo, seguido por uma linha com campos para hora, minutos e segundos, separados por pontos e um hífen. No final, há um botão "Confirmar".

**Figura 17: Gerenciamento de Horários**

### 4.3 Alunos

Na tela “Alunos” um Professor será capaz de cadastrar e gerenciar seus alunos e definir suas informações gerais e foto, lutas que pratica com suas respectivas graduações, academias e horários em que estuda, pagamentos que foram efetuados, gerenciar a autorização de imagem para campeonatos e filtrar sua lista de alunos pelos critérios de Lutas, Academias e Horários. Podemos ver a tela “Alunos” na Figura 18:

Black Belt 1.0

Arquivo Configurações Sincronização Sobre

Alunos Lutas, Academias e Horários Chamadas e Quadro de Horários Registros De Professor Registros De Alunos

Selecionar Aluno Por...

Karate Shotokan (Brasil)

Aluno

Alterar Foto

Peso Médio ( Em Kg ):

Nascimento:

Nome:

CPF:

Identidade:

Responsável:

Telefones:

Email:

Autorização de Imagem:

Visualizar Atual

Alterar

Criar Aluno

Lutas:

Adicionar Luta

Graduação na Luta:

Academia:

Nova Academia

Horários:

Novo Horário

Pagamentos:

Histórico

Adicionar

Salvar Alterações

Fechar Aluno

**Figura 18: Tela “Alunos”**

#### **4.3.1. Informações gerais**

Para adicionar as informações gerais dos alunos, um Professor deverá simplesmente acessar os campos em branco (quando forem campos de texto) e adicionar as informações desejadas. Alguns destes campos são obrigatórios. Todos os campos possuem limitações de caracteres (por exemplo: Data de Nascimento só aceita números e “/”, Telefones só aceita números, “(”, “)”, “,” e “-”, etc). Para foto e autorização de imagem basta selecionar arquivos que estejam dentro das extensões permitidas para cada um destes tipos.

#### **4.3.2. Luta e graduações**

Ao criar um aluno, um professor poderá adicionar somente uma única luta para ele. Após salvar com sucesso o novo aluno será habilitada a opção de adicionar novas lutas e graduações para o mesmo. Para cada luta adicionada a um aluno, a combo box de graduações será populada com as graduações referentes àquela luta para que o Professor possa definir qual a graduação corrente do aluno na luta determinada. A graduação e

mandatória, e caso não seja escolhida uma, ela será definida sempre como a graduação mais baixa daquela luta.

#### 4.3.3. Academias

Para adicionar uma academia é mandatório que uma luta esteja selecionada para o aluno corrente. O Professor terá a possibilidade de escolher entre as academias que já foram cadastradas para a luta que está selecionada.

#### 4.3.4. Horários

Para adicionar um horário é mandatório que uma academia esteja selecionada para o aluno corrente. O Professor terá a possibilidade de escolher entre os horários que já foram cadastradas para a academia que está selecionada na luta selecionada.

#### 4.3.5. Pagamentos

Para adicionar um horário é mandatório que uma academia esteja selecionada para o aluno corrente e acionar o botão “Adicionar Pagamento”. Uma tela auxiliar se abrirá e o Professor poderá configurar o pagamento do aluno como mostra a Figura 19:

A imagem mostra uma janela de software intitulada "Adicionar Pagamento". No topo, há uma barra de título com ícones de minimizar, maximizar e fechar. O conteúdo da janela é dividido em duas seções principais. À esquerda, sob o rótulo "Horários Na Academia:", há uma lista de horários disponíveis: "09:00:00 - 10:00:00 ( Segunda )", "14:00:00 - 16:00:00 ( Terça )", "09:00:00 - 10:00:00 ( Quarta )", e "14:00:00 - 16:00:00 ( Quinta )". À direita, sob o rótulo "Horários Selecionados:", há um campo vazio. Entre as duas seções, há dois botões de seta: ">>" e "<<". Abaixo dessas seções, há campos para "Ano:", "Mês:", "Data:" e "Preço:". No rodapé da janela, há dois botões: "Adicionar" e "Cancelar".

**Figura 19: Tela “Adicionar Pagamento”**

Para verificar o histórico de pagamentos do aluno selecionado basta acionar o botão “Histórico” abaixo do texto “Pagamentos”. Nesta outra tela auxiliar um Professor poderá verificar sequencialmente os pagamentos cadastrados para aquele aluno na academia selecionada.

#### 4.3.6. Filtrar aluno

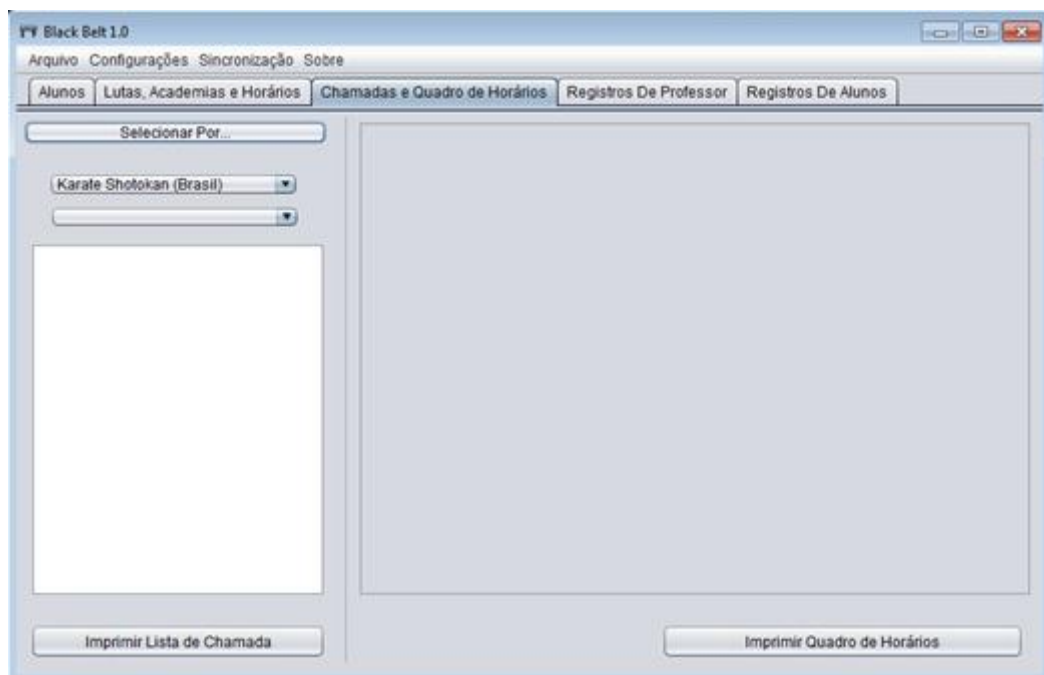
Para facilitar o processo de encontrar os alunos na lista, um Professor poderá filtra-los por luta, academia e horário. Para fazer a filtragem, basta selecionar sequencialmente os itens citados na ordem desejada nas respectivas combo box. Os alunos que são exibidos na tabela serão filtrados de acordo com os quesitos escolhidos.

#### 4.3.7. Deletar aluno

É possível remover um aluno da lista de alunos com a ação Deletar. Um aluno nunca será deletado de fato, todas as informações sobre este aluno ficarão apenas invisíveis, pois, caso um Professor deseje reaver estas informações elas estarão disponíveis para recuperação. Para realizar a recuperação basta acessar a aba de alunos no menu de preferências.

### 4.4 Chamadas e Quadros de Horários

Na tela “Chamadas e Quadros de Horários” um Professor poderá gerar relatórios e gerar listas de chamada mensais de acordo com algumas preferências como mostra a Figura 20:



**Figura 20: Tela “Chamadas e Quadros de Horários”**

#### 4.4.1. Filtrar por luta

Para mostrar a primeira tabela e habilitar a primeira funcionalidade do botão “Imprimir Quadro de Horários”, um Professor deverá fazer uma filtragem por luta selecionando a luta desejada na combo box de lutas conforme a Figura 21:



Figura 21: Filtragem por Luta

Uma tabela será exibida com todas as academias às quais o Professor ministra aulas daquela luta e todos os estilos de luta que estão relacionados àquela academia. Estas academias serão também carregadas na combo box de academias. Veja um exemplo na Figura 22:

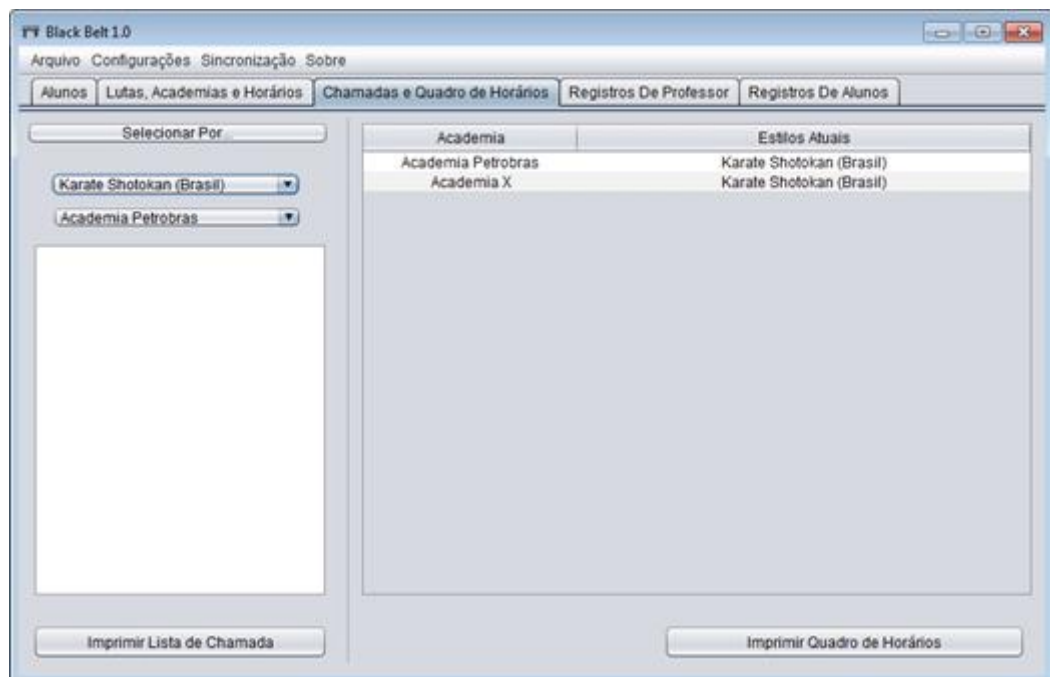


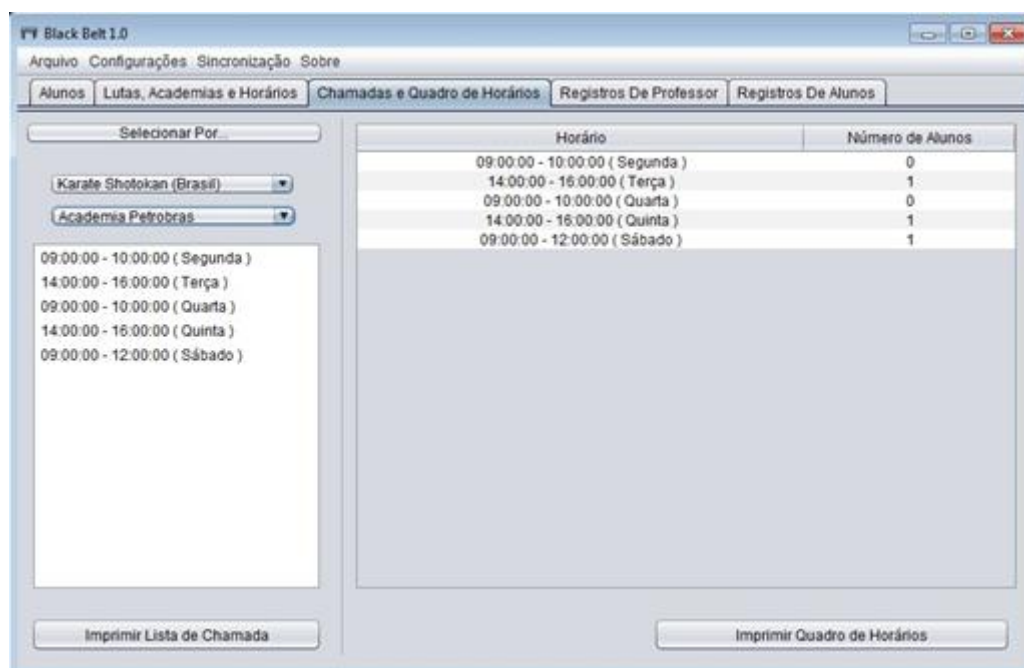
Figura 22: Resultado da Filtragem por Lutas

Caso um Professor deseje salvar a tabela gerada em formato Excel, basta acionar o botão “Imprimir Quadro de Horários” e selecionar o local onde deseja salvar o arquivo Excel que será gerado.

#### 4.4.2. Filtrar por academia

Para mostrar a segunda tabela, habilitar a segunda funcionalidade do botão “Imprimir Quadro de Horários” e ainda habilitar a funcionalidade do botão “Imprimir Lista de Chamada”, um professor deverá fazer uma filtragem por academia selecionando a academia desejada na combo box de academias.

Uma tabela será exibida com os horários relacionados àquela academia e a quantidade total de alunos que estão cadastrados para a mesma. Veja um exemplo na Figura 23:



**Figura 23: Filtragem por Academias**

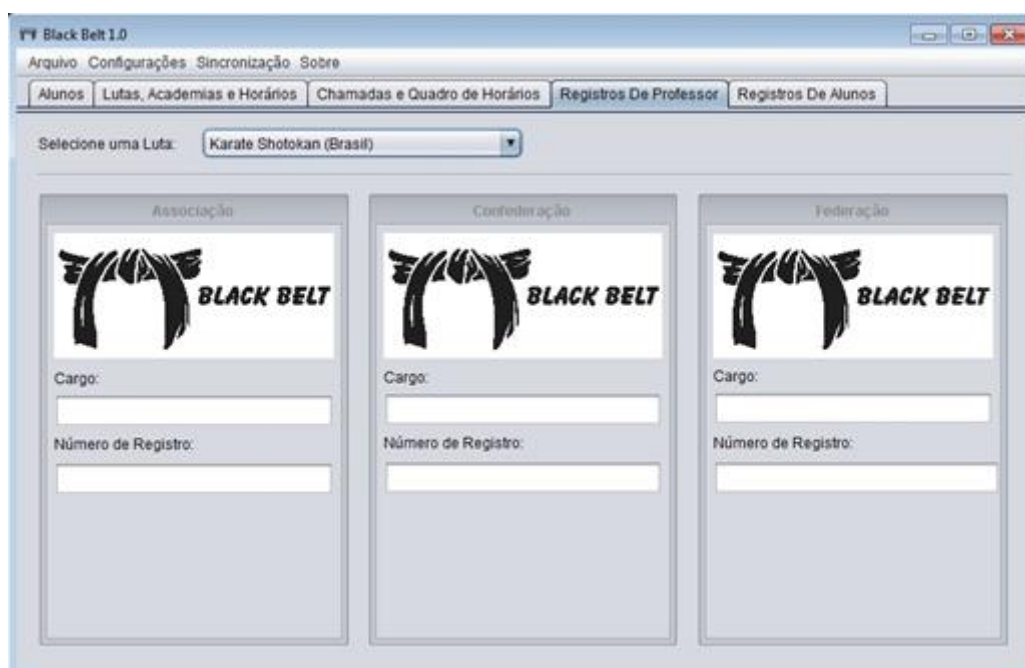
Caso um Professor deseje salvar a tabela gerada em formato Excel, basta acionar o botão “Imprimir Quadro de Horários” e selecionar o local onde deseja salvar o arquivo Excel que será gerado.

Para habilitar a função “Imprimir Lista de Chamada” para um horário, um Professor deverá primeiro selecionar um horário na lista de horários. Em seguida

acionar o botão “Imprimir Lista de Chamada” e selecionar o local onde deseja salvar o arquivo Excel que será gerado. Neste arquivo será gerada uma lista de chamada com todos os nomes dos alunos que estão cadastrados para a Luta, Academia e Horário selecionados, e colunas com todas as datas do mês do dia da semana do horário selecionado (todas as quartas feiras do mês, por exemplo).

## 4.5 Registros de Professor

Na tela “Registros De Professor” um Professor poderá verificar suas matrículas e cargos nas entidades de Associação, Federação e confederação de todas as lutas que ministra. Só poderá ser possível esta verificação caso um Professor já tenha se cadastrado com sucesso em alguma destas entidades. A tela “Registros De Professor” é exibida na Figura 24:



**Figura 24: Tela “Registros de Professor”**

Para que um professor possa verificar seus registros em uma determinada luta, basta selecionar a luta desejada na combo box de lutas. Se houverem registros nas entidades relacionadas, eles aparecerão automaticamente.



## 4.6 Registro de Alunos

Na tela “Registros De Alunos” um Professor poderá verificar as matrículas e cargos de seus alunos nas entidades de Associação, Federação e Confederação, bem como fazer a filtragem da lista de alunos por Luta, Academia e Horário como mostra a Figura 25:

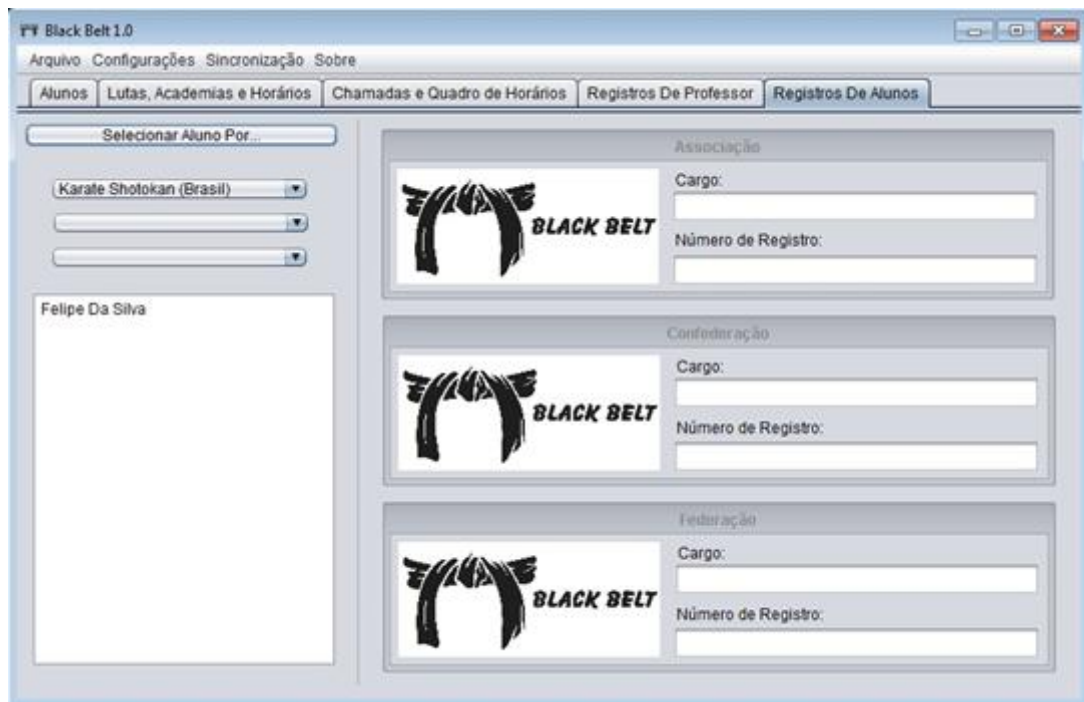


Figura 25: Tela “Registros de Alunos”

Para verificar os registros de um determinado aluno, um Professor deve simplesmente selecioná-lo com um clique duplo no nome do aluno desejado.

Os resultados exibidos para os cargos dos alunos serão definidos pela luta selecionada na combo box de lutas (ou pela primeira luta da lista caso nenhuma tenha sido selecionada). O filtro de alunos da tela “Registros de alunos” funciona de maneira análoga ao da tela “Alunos” (Item 3.7).

## 4.7 Menu

Além da função de login online e off-line, o BlackBelt possui outros menus com funcionalidades e configurações de software.

#### 4.7.1. Configurações de Servidor

Atualizações de configuração de servidor normalmente ocorrerão em caráter automático, ou seja, as configurações de servidor virão a partir de uma tabela servidor externo. Em caso de necessidade um professor poderá realizar essa troca de configuração manualmente ou tentando realizar o update automático. Todas estas opções estão disponíveis como mostra a Figura 26 abaixo:



Figura 26: Menu “Configurar Servidor”

#### 4.7.2. Preferências

No menu de preferências um Professor poderá realizar ações para troca de senha, alteração do usuário Premium, alteração da linguagem do software e recuperação de alunos excluídos. A Figura 27 abaixo exhibe a aba da troca de senhas:



Figura 27: Preferências: Troca de Senha

Na aba de usuário Premium um Professor poderá definir seu usuário e senha do login atual como usuário Premium da instância que estiver usando, ou seja, ele estará apto a trabalhar off-line e poderá utilizar os recursos de sincronização do BlackBelt. Nesta tela ele poderá também alterar a língua corrente do software como mostra a Figura 28:



**Figura 28: Preferências: Usuário Premium**

Na aba de alunos um Professor poderá recuperar os alunos que foram deletados do seu perfil selecionando o aluno desejado na lista e acionando a ação de recuperar aluno como mostra a Figura 29:



**Figura 29: Preferências: Alunos**

#### **4.7.3. Sincronização**

No menu “Sincronização” um Professor poderá utilizar a ação “Sincronizar Agora” para realizar uma sincronização de seus dados sempre que estiver no modo off-

line, sem a necessidade de realizar um login online. Este método é interessante para lugares onde a conectividade é limitada e o Professor não deseja esperar tanto para sincronizar seus dados com o servidor e a base de dados.

#### **4.7.4. Sobre**

No menu “Sobre” um Professor poderá verificar a versão do software e alguns outros detalhes sobre o BlackBelt, quando estes estiverem disponíveis.

#### **Limitações:**

Como pudemos observar, algumas das funcionalidades do módulo professor são incapazes de ser testadas à priori, pois parte do módulo depende da integração com os outros módulos, que serão desenvolvidos no futuro.

Podemos citar entre estes, as verificações de cadastros dos professores nas entidades de Associação, Federação e Confederação, bem como seus números de registro e cargos. O mesmo se aplica para o registro de alunos nas citadas entidades de acordo com a tela “Registros de Alunos”. Nesta aba possuímos as mesmas funcionalidades de “Registros de Professor”, porém com duas funcionalidades a mais de filtros: Filtragem por academias e por horários. Estes filtros funcionam de maneira similar à tela de cadastro de alunos.

## 5 Conclusão

### Considerações Finais:

Este trabalho apresentou a necessidade de implantar uma tecnologia capaz de trazer ao universo das artes marciais um fator facilitador de processos burocráticos. Este sistema permite o acesso à informações que de outra maneira seriam difíceis ou até impossíveis de ser acessadas, além de gerar relatórios inteligentes e instantâneos que poderão ser utilizados para criar novas estratégias de marketing e ainda análises demográficas para organização de campeonatos.

Para isso, após a elaboração de um estudo de caso junto a *stakeholders*, foram definidas as tecnologias a serem utilizadas para a criação do novo software. Com relação ao desenvolvimento do Blackbelt, foi empregada a linguagem de programação Java e suas facilidades para realizar a associação ao sistema de gerenciamento de banco de dados MySQL.

As facilidades e praticidades envolvidas no uso do produto final deverão ser incluídas no dia a dia dos profissionais da área até que o produto tenha a sua usabilidade considerada indispensável no fluxo de trabalho dos mesmos.

Ainda como parte desta estratégia, e a fim de facilitar o processo de disseminação massiva da solução, as entidades de nível hierárquico mais elevado deverão ser contatadas e persuadidas a contribuir com a divulgação do produto, com a promessa de que os benefícios a curto e médio prazo poderão de fato trazer os benefícios prometidos na proposta de solução.

Para a solução do módulo “Professor” implementada neste trabalho, por exemplo, as entidades de Associação seriam os principais alvos de venda da solução. Se, para estas associações, for interessante adquirir seu respectivo módulo, visto que a prática antes utilizada era rústica e burocrática, então o processo passará a ser em tempo

real e com compartilhamento global, se tornando uma solução simples e coerente para um problema existente.

O módulo professor é de certa forma um dos módulos mais importantes dentro do quadro geral da solução completa apresentada. Alguns dos dados que serão gerados e mantidos por este módulo poderão ser acessados por alguns ou talvez todos os outros módulos com a finalidade de geração de relatórios e algumas outras verificações. Os dados de professores e alunos são fundamentalmente os de maior importância dentro do quadro geral da solução.

Os próximos passos para o desenvolvimento completo da solução serão as modelagens e implementações dos três módulos ainda não implementados: Associação, Confederação e Federação. Cada um destes módulos deverá contar com um estudo de caso separado de acordo com a necessidade de cada uma das entidades associadas.

A importância dos outros módulos se faz óbvia se analisarmos o ponto de vista do objetivo inicial, que é criar módulos integrados que realizam tarefas diferentes para as diferentes entidades.

Em paralelo com o desenvolvimento dos outros módulos, o projeto visa à inscrição em editais do governo objetivando a criação de aplicativos para as principais plataformas de smartphones (Android, iOS e Windows Phone) que realizem as mesmas funcionalidades dos módulos já desenvolvidos e que tragam aos profissionais da área a praticidade do gerenciamento das tarefas de sua profissão combinado ao já tão difundido diário do celular.

## Referências Bibliográficas

SHNEIER, B. (2009) “Ever Better Cryptanalytic Results Against Sha-1”. Acessado 14 de Novembro de 2014,

[https://www.schneier.com/blog/archives/2009/06/ever\\_better\\_cry.html](https://www.schneier.com/blog/archives/2009/06/ever_better_cry.html)

SHNEIER, B. (2009) “Cryptanalysis of SHA-1”. Acessado 14 de Novembro de 2014,

[http://www.schneier.com/blog/archives/2005/02/cryptanalysis\\_o.html](http://www.schneier.com/blog/archives/2005/02/cryptanalysis_o.html)

WANGENHEIM, A. V. (2002) “Hashing”. Acessado 14 de Novembro 2014,

<http://www.inf.ufsc.br/~ine5384-hp/Hashing/>

Caelum. “Testes, XML e Design Patterns”. Acessado 14 de Novembro de 2014,

<http://www.caelum.com.br/apostila-java-testes-xml-design-patterns/>

MILANI, A. (2007) “MySQL - Guia do Programador”. Acessado 3 de Novembro de 2014,

<http://www.novateceditora.com.br/livros/mysqlcompleto/capitulo8575221035.pdf>

Oracle. “What is MySQL?”. Acessado 4 de Novembro de 2014,

<http://dev.mysql.com/doc/refman/5.1/en/what-is-mysql.html>

DB-Engines. "DB-Engines Ranking". Acessado 4 de Novembro de 2014,

<http://db-engines.com/en/ranking>

GARCIA, P. and GORTON, J. (2012) “MySQL para Administradores de Banco de Dados”, Guia do Aluno - Volume I

GARCIA, P. and GORTON, J. (2012) “MySQL para Administradores de Banco de Dados”, Guia do Aluno - Volume II

Universidade do Estado de Santa Catarina - UDESC. “O que é o MySQL”. Acessado 3 de Novembro de 2014,

[http://ceavi.udesc.br/arquivos/id\\_submenu/487/o\\_que\\_e\\_o\\_mysql.pdf](http://ceavi.udesc.br/arquivos/id_submenu/487/o_que_e_o_mysql.pdf)

Oracle. “The History of Java Technology”. Acessado 16 de Novembro de 2014,

<http://www.oracle.com/technetwork/java/javase/overview/javahistory-index-198355.html>

PACIEVITCH, Y. “História do Java”. Acessado 16 de Novembro de 2014,

<http://www.infoescola.com/informatica/historia-do-java/>

Caelum. “Java e Orientação a Objetos”. Acessado 17 de Novembro de 2014,

<http://www.caelum.com.br/apostila-java-orientacao-objetos/>

VisualBuilder. “Java Hibernate Tutorial Home”. Acessado 20 de Novembro de 2014,

<http://www.visualbuilder.com/java/hibernate/tutorial/>

KEITH, M. and SCHINCARIOL, M. “Pro JPA 2 - Mastering the Java™ Persistence API”

Caelum. “Uma introdução prática ao JPA com Hibernate”. Acessado 22 de Novembro de 2014,

<http://www.caelum.com.br/apostila-java-web/uma-introducao-pratica-ao-jpa-com-hibernate/>