



UNIVERSIDADE ESTADUAL PAULISTA
“JÚLIO DE MESQUITA FILHO”
Câmpus de Presidente Prudente

BRUNO RENAN GELAKO SANTOS

**WEBSISLAPAM: SISTEMA DE AVALIAÇÕES DE PESSOAS
BASEADO EM TECNOLOGIAS WEB**



UNIVERSIDADE ESTADUAL PAULISTA
“JÚLIO DE MESQUITA FILHO”
Câmpus de Presidente Prudente

BRUNO RENAN GELAKO SANTOS

**WEBSISLAPAM: SISTEMA DE AVALIAÇÕES DE PESSOAS
BASEADO EM TECNOLOGIAS WEB**

Trabalho de conclusão de curso apresentado ao Departamento de Matemática, Estatística e Computação (DMEC), da Universidade Estadual Paulista “Júlio de Mesquita Filho”, sob o título **Websislapam: sistema de avaliações de pessoas baseado em tecnologias web.**

Orientador: Prof. Dr. Ronaldo Celso Messias Correia

Presidente Prudente

2011

Termo de Aprovação

BRUNO RENAN GELAKO SANTOS

WEBSISLAPAM: SISTEMA DE AVALIAÇÕES DE PESSOAS BASEADO EM TECNOLOGIAS WEB

Monografia sob o título **Websislapam: sistema de avaliações de pessoas baseado em tecnologias web**, defendida por BRUNO RENAN GELAKO SANTOS e aprovada em 23 de Agosto de 2011, em Presidente Prudente, Estado de São Paulo, pela banca examinadora constituída pelos doutores:

Prof. Dr. Ronaldo Celso Messias Correia
Universidade Estadual Paulista

Prof. Dr. Rogério Eduardo Garcia
Universidade Estadual Paulista

Prof. Dr. Marco Antônio Piteri
Universidade Estadual Paulista

Presidente Prudente, 23 de Agosto de 2011

AGRADECIMENTOS

Aos meus pais, pelo apoio incondicional, pela educação e força necessária para superar os obstáculos da vida.

À minha companheira Laís, pelo incentivo nos momentos mais difíceis e por sempre estar ao meu lado.

Aos meus amigos e a minha família, pela compreensão nos momentos em que estive ausente.

Ao meu orientador Prof. Dr. Ronaldo Celso Messias e ao Prof. Dr. Ismael Forte Freitas Junior pela paciência, confiança e atenção durante toda a realização deste trabalho.

RESUMO

Websislapam: sistema de avaliações de pessoas baseado em tecnologias web

O presente trabalho, intitulado Websislapam: Sistema de Avaliações de Pessoas Baseado em Tecnologias Web, possibilita a criação de questionários, e a organização de entidades e de pessoas que participam de avaliações. As entidades coletam dados referentes a pessoas com a ajuda de recursos que diminuem erros de digitação. O Websislapam mantém uma base de dados e fornece relatórios gráficos, que possibilitam a análise das pessoas avaliadas. Desenvolvido com tecnologias Web, tais como, *PHP*, *Javascript*, *CSS*, entre outras. Desenvolvido com o paradigma de programação orientado a objetos e com o *SGBD MySQL*. Para a fundamentação teórica, pesquisas nas áreas de Sistema de Banco de Dados, Tecnologias Web e Engenharia da Web foram realizadas. Conceituou-se processo de avaliação, sistemas e aplicações baseados na Web, Engenharia da Web e Sistema de Banco de Dados. Tecnologias aplicadas na implementação do Websislapam foram descritas. Em capítulo próprio apresentou-se as principais funcionalidades e os artefatos utilizados no desenvolvimento do Websislapam. Um estudo de caso demonstra o uso prático do sistema.

Palavras-chave: Sistema de Banco de Dados, Processo de Avaliação, Tecnologias Web.

ABSTRACT

Websislapam: system of assessments of people based on web techonologies

This work, entitled Websislapam: People Rating System Based on Web Technologies, allows the creation of questionnaires, and the organization of entities and people who participate in evaluations. Entities collect data from people with the help of resources that reduce typing mistakes. The Websislapam maintains a database and provides graphical reporting, which enable the analysis of those tested. Developed using Web technologies such as PHP, Javascript, CSS, and others. Developed with the paradigm of object-oriented programming and the MySQL DBMS. For the theoretical basis, research in the areas of System Database, Web Technologies and Web Engineering were performed. It was determined the evaluation process, systems and Web-based applications, Web and System Engineering Database. Technologies applied in the implementation of Websislapam been described. In a separate chapter presented the main features and artifacts used in the development of Websislapam. A case study demonstrates the practical use of the system.

Keywords: System Database, Assessment Process, Web Technologies.

LISTA DE FIGURAS

Figura 1: Arquitetura de um sistema de banco de dados	27
Figura 2: Documento <i>XML</i> com <i>DTD</i>	31
Figura 3: Diagrama de casos de uso.....	42
Figura 4: Camada de controladores simplificada	43
Figura 5: Camada do modelo simplificada	44
Figura 6: Camada de persistência simplificada	45
Figura 7: Diagrama de sequência cadastrar entidade	45
Figura 8: Modelo relacional das entidades e avaliadores que acessam o sistema ...	47
Figura 9: Modelo relacional dos projetos e subprojetos da entidade.....	48
Figura 10: Modelo relacional das variáveis do projeto e subprojeto	49
Figura 11: Modelo relacional das variáveis quantitativas e qualitativas	50
Figura 12: Diagrama de atividade coletar dados	52
Figura 13: Diagrama de atividade salvar coleta	52
Figura 14: Diagrama de atividade salvar resposta	53
Figura 15: Modelo relacional das coletas de dados	54
Figura 16: Cadastrar entidade.....	58
Figura 17: Cadastrar projeto.....	58
Figura 18: Tela principal do responsável do projeto	59
Figura 19: Cadastrar subprojeto.....	59
Figura 20: Avaliadores do projeto e suas permissões nos subprojetos.....	60
Figura 21: Cadastrar variável quantitativa.....	60
Figura 22: Cadastrar variável qualitativa	61
Figura 23: Variáveis do projeto.....	61
Figura 24: Selecionar grupo de variáveis para coletar	62
Figura 25: Coleta de dados	63

Figura 26: Acompanhamento do avaliado para variável quantitativa	64
Figura 27: Acompanhamento do avaliado para variável qualitativa	64
Figura 28: Grupos de avaliados selecionados para gráfico de variável qualitativa ...	65
Figura 29: Grupos de avaliados para variável qualitativa	65
Figura 30: Grupos de avaliados selecionados para gráfico de variável quantitativa .	66
Figura 31: Grupos de avaliados para variável quantitativa.....	66
Figura 32: Quantidade de coletas por subprojetos.....	67
Figura 33: Dados gerais do projeto	67
Figura 34: Dados coletados exportados.....	68

SUMÁRIO

1	INTRODUÇÃO	10
1.1	MOTIVAÇÃO	10
1.2	ORGANIZAÇÃO	11
2	REVISÃO BIBLIOGRÁFICA.....	12
2.1	PROCESSO DE AVALIAÇÃO.....	12
2.2	SISTEMAS E APLICAÇÕES BASEADOS NA WEB.....	14
2.2.1	Web 2.0	17
2.3	ENGENHARIA DA WEB	18
2.3.1	Formulação e Planejamento	19
2.3.2	Modelagem de Análise	20
2.3.3	Modelagem de Projeto	22
2.3.4	Teste de Aplicações Web	23
2.4	SISTEMA DE BANCO DE DADOS	24
2.4.1	SGBD MySQL	27
2.5	TECNOLOGIAS APLICADAS NA IMPLEMENTAÇÃO DO WEBSISLAPAM	29
2.5.1	XML	29
2.5.2	PHP	32
2.5.3	JpGraph	34
2.5.4	JavaScript e DHTML	34
2.5.5	Cascading Style Sheets (CSS)	36
2.5.6	Ajax	36
2.5.7	JQuery e JQuery UI	37
3	SISTEMA DE AVALIAÇÕES DE PESSOAS WEBSISLAPAM	39
3.1	VISÃO GERAL	41
3.2	CONFIGURAÇÃO DO AMBIENTE DE AVALIAÇÕES	47
3.2.1	Estrutura das Entidades	47
3.2.2	Estrutura das Variáveis Dinâmicas	49
3.3	COLETAS DE DADOS.....	51
3.4	ANÁLISE DOS DADOS COLETADOS	55
4	ESTUDO DE CASO	57
5	CONCLUSÃO	69
	REFERÊNCIAS BIBLIOGRÁFICAS	71

1 INTRODUÇÃO

O presente trabalho, desenvolvido com tecnologias *Web*, tem como objetivo proporcionar a entidades educacionais, comerciais, entre outras, recursos para realizar avaliações de pessoas em áreas de interesse heterogêneas, ou seja, avaliações de pessoas que independem do contexto da pesquisa. Oferece também recursos para a identificação e organização das equipes de pessoas que trabalham nas entidades para o sucesso de suas avaliações.

As avaliações são importantes, pois objetivam as verdadeiras informações, não só sobre as pessoas que são avaliadas, mas também sobre as entidades que realizam as avaliações. Estas informações proporcionam a descoberta de padrões, descoberta de erros, a análise de progressos e objetivos, a tomada de decisões e o aperfeiçoamento.

Para a fundamentação teórica, foi pesquisado sobre avaliações de pessoas, a fim de esclarecer os passos necessários para que um processo de avaliação seja considerado eficiente. Pesquisou-se também sistemas de banco de dados e transferências de dados, as tecnologias *Web* disponíveis, a nova visão da *Web* e o desenvolvimento de sistemas e aplicações baseados na *Web*, a fim da escolha das tecnologias mais adequadas para o desenvolvimento.

Aproveitando os benefícios da *Web 2.0* e considerando a importância de processos de avaliação, o escopo do trabalho é fornecer a entidade independentemente de sua área de pesquisa, a exclusividade sobre seus dados, a definição de um ambiente de avaliações, as coletas de dados e relatórios que possibilitam alguma análise para a descoberta de informações.

1.1 Motivação

Dentre os trabalhos relacionados ao Websislapam que motivam o desenvolvimento, considera o sistema SISLAPAM que realiza coletas de dados através de dispositivos móveis e o sistema de formulários do *Google Docs*.

O SISLAPAM, desenvolvido como trabalho de conclusão de curso por Flávio de Queiroz e Silva, em 2008, na Universidade Estadual Paulista “Júlio de Mesquita Filho”, campus de Presidente Prudente, é um software para coleta e análise de dados sobre as condições físicas de um indivíduo, tais como anamnese, risco coronariano e composição corporal, que utiliza banco de dados em rede com transmissão dos dados via rede sem-fio. Pode ser executado em dispositivos portáteis como PDAs, celulares e *smart-phones* que possuam conexão com Internet (*WiFi*, *Bluetooth* ou *3G*).

O sistema de formulários do *Google Docs* permite ao usuário criar um formulário e disponibilizá-lo na internet, para coletar dados referentes a qualquer área de pesquisa. Possibilita também, o acesso aos dados coletados através de planilhas, ferramentas gráficas, entre outros recursos.

1.2 Organização

O trabalho é organizado da seguinte maneira: O Capítulo 2 apresenta a fundamentação teórica estudada; no Capítulo 3 é apresentado o desenvolvimento e as funcionalidades do Websislapam, o qual foi dividido em Visão Geral, Configuração do Ambiente de Avaliações, Coletas de Dados e Análise dos Dados Coletados; no Capítulo 4 apresenta um estudo de caso que é seguido da conclusão, onde há sugestões de trabalhos futuros.

2 REVISÃO BIBLIOGRÁFICA

Neste capítulo é apresentada a fundamentação teórica para o desenvolvimento e implementação do Websislapam.

2.1 Processo de Avaliação

Segundo Salvia e Ysseldyke (1991, p. 5) “avaliação é o processo de coletar dados com o objetivo de (1) especificar e verificar problemas e (2) tomar decisões”.

Nesse contexto de avaliação, considera o processo de avaliação de pessoas, podendo ser de uma única pessoa ou de um grupo específico.

O processo de avaliação, que é criado por pessoas com o conhecimento adequado na área de interesse, possui objetivos, é amplo e consiste em alguns elementos. Como elementos, podem ser considerados aplicações de testes, onde uma pessoa é exposta a um conjunto específico de questões, as observações, as opiniões e podem ser considerados também fatores externos como o ambiente e as características de uma pessoa.

Todos estes fatores têm como objetivo realizar uma coleta de dados que esteja o mais próximo possível da realidade, ou seja, que identifique as características das pessoas que são interessantes para o objetivo da avaliação.

As avaliações podem ser aplicadas em diferentes áreas de negócios, áreas acadêmicas, áreas na educação, áreas médicas, entre outras, podem ser aplicadas em todas as áreas interessadas em verificar progressos, especificar objetivos que tenham sido atingidos e que desejam tomar as melhores decisões para a evolução e resolução de problemas.

Um dos elementos principais das avaliações são os testes que podem ser aplicados a grupos de pessoas. Por meio dos testes, são realizadas coletas de dados, ou seja, armazenamento das respostas das pessoas aos testes.

Considerando a importância dos testes, eles se definem como:

[...] pode ser considerado um conjunto de tarefas ou questões concebidas para determinar tipos particulares de comportamentos quando apresentados sob condições padronizadas e para fornecer escores que possuam as propriedades psicométricas convenientes. (SALVIA; YSSELDYKE, 1991, p. 4)

Os testes são um conjunto de perguntas ou tarefas para as quais se esperam tipos de respostas. Apresentam utilidades, pois as perguntas ou as tarefas são apresentadas para todas as pessoas entrevistadas exatamente da mesma maneira, assim, os desempenhos das pessoas podem ser comparados, independentemente da pessoa que aplicou o teste.

Após a realização da coleta de dados de acordo com os testes específicos criados, utilizando um conhecimento necessário e com o auxílio das ferramentas adequadas, é possível realizar uma análise dos dados, onde as idéias sobre os dados são esclarecidas e os dados são interpretados para que as melhores decisões sejam tomadas sobre esse grupo de pessoas.

Para obter um processo de avaliação considerado confiável, ou seja, que os resultados sejam válidos é necessário que o processo atenda a alguns pressupostos fundamentais, alguns deles são:

- A pessoa que aplica o teste deve estar devidamente capacitada;
- O teste deve ser amplo o suficiente para garantir seu objetivo;
- Um grau maior de conhecimento é necessário para a pessoa responsável por analisar e interpretar os dados coletados;
- Os dados coletados devem possuir a menor quantidade possível de erros;
- Os dados históricos coletados devem ser utilizados para a tomada de decisões;

Basicamente, os tipos de dados coletados por meio de testes são os dados quantitativos e os dados qualitativos. Considerando a importância dos dados quantitativos e qualitativos, eles se definem como:

[...] os dados quantitativos referem-se aos escores reais obtidos no teste e os dados qualitativos consiste em observações não sistemática feitas enquanto uma criança é testada, e nos diz como essa criança obteve o escore. (SALVIA; YSSELDYKE, 1991, p. 23)

Os dados quantitativos podem ser determinados por meio de instrumentos de medição, por contagens ou estimativas. São subdivididos em dados de medição que possuem um valor numérico e uma unidade de medida, e dados numéricos que são compostos por um valor numérico e uma característica.

Os dados qualitativos podem ser separados em diferentes categorias que se distinguem por alguma característica, sendo expressos por atributos, como exemplo, considera os sexos (masculino / feminino), o estado civil das mulheres no Brasil (casadas, solteiras, divorciadas,...), e muitos outros, todos dependentes do interesse da área em que estão aplicados.

Um processo de avaliação consiste em aplicar técnicas de coleta de dados a pessoas, sendo que estas técnicas devem ser gerenciadas por pessoas com o conhecimento adequado na área de interesse. Após a coleta de dados, os dados devem ser analisados com o objetivo de identificar os progressos, problemas, objetivos ou identificar comportamentos que estejam em destaque das pessoas avaliadas. Com isso, as devidas decisões são tomadas de acordo com o interesse da área e objetivo do processo de avaliação.

2.2 Sistemas e Aplicações baseados na Web

Os sistemas e aplicações baseados na *Web* englobam desde uma simples página *Web* com poucas funcionalidades para o usuário final, até um complexo sistema *Web* com grande processamento de dados, funcionalidades específicas, integrados com outros sistemas, integrados com grandes bancos de dados e integrados com regras de negócios.

Conforme Pressman (2010) podem ser categorizados de acordo com suas funcionalidades que tratam problemas. Entre as categorias mais comuns estão:

- Os sistemas informacionais, que fornecem conteúdo somente de leitura e navegação;
- Os sistemas para baixar, que disponibilizam ao usuário *downloads* de arquivos de um servidor adequado;

- Os sistemas adaptáveis, onde o usuário adapta o conteúdo de acordo com suas necessidades;
- Os sistemas de interação, cujos usuários se comunicam entre si por meio de comunidades, salas de bate-papo, mensagens, entre outras formas;
- Os sistemas de entrada do usuário, onde os formulários são a principal maneira de se comunicar;
- Os sistemas orientados a serviços, que fornecem serviços que resolvem problemas dos usuários, como por exemplo, algum processamento matemático;
- Os sistemas de portal, que orientam os usuários para outros conteúdos ou serviços da *Web*.
- Os sistemas de acesso a banco de dados, que possibilitam ao usuário extrair informações de uma grande base de dados.

Há também sistemas e aplicações baseados na *Web* que se classificam em diversas destas e outras categorias. São mais sofisticados, capazes de fornecerem ao usuário uma ótima experiência combinando soluções adequadas para os seus problemas.

Como características, são consideradas as seguintes:

- São concentrados em redes, podem residir na internet, em intranets ou extranets, estando disponíveis para o mundo todo, somente dentro de uma organização ou comunicando-se entre redes, respectivamente;
- São utilizados de maneira concorrente por diversos usuários;
- Possuem cargas imprevisíveis, ou seja, um número indeterminado de usuários pode variar em momentos indeterminados, podendo atingir milhões de acessos;
- Em se tratando de desempenho, caso um usuário espere muito tempo pela sua resposta, seja ela um processamento, uma exibição ou outra funcionalidade, este usuário pode encaminhar-se para outro sistema em busca de melhor desempenho;
- Os sistemas devem permanecer o maior tempo possível disponíveis. Em casos de sistemas na internet, usuários do mundo

todo possuem acesso, sendo estes acessos em horários locais, ou seja, um horário que poderia ser utilizado para manutenção do sistema e assim tornar a aplicação indisponível na América deve estar disponível, pois, na Ásia muitos outros usuários continuam acessando;

- São voltados a dados, apresentam textos, imagens, gráficos, áudios e vídeos aos usuários e fornecem acessos a bancos de dados;
- São sensíveis aos conteúdos, as estéticas dos conteúdos gerados são fatores determinantes para a qualidade;
- Possuem uma evolução continuada, diferem-se dos *softwares* de aplicação convencional que evoluem com uma série de versões planejadas e em consideráveis intervalos de tempo. Alguns sistemas *Web* atualizam seus conteúdos minuto a minuto;
- Há uma necessidade de rápida colocação no mercado, que se justificam devido às estratégias de negócios dos clientes ou outros fatores. Com o auxílio de ferramentas sofisticadas e metodologias ágeis esses curtos prazos podem ser alcançados;
- Como os sistemas *Web* residem em redes, tornam-se expostos a uma grande quantidade de usuários, então, medidas de segurança devem ser implementadas, limitando o acesso dos usuários, protegendo o conteúdo e a transmissão dos dados;
- A estética pode ser muito importante dependendo da categoria do sistema *Web*. Em casos de sistemas para vendas, a estética se torna fundamental para o sucesso.

Pressman (2010, p. 380) argumenta sobre os sistemas de *softwares* e sistemas *Web*: “Pode-se argumentar que uma aplicação tradicional em qualquer domínio de software, pode exibir essa lista de atributos. No entanto, WebApps quase sempre o fazem”.

2.2.1 Web 2.0

O termo *Web 2.0* foi utilizado pela primeira vez em 2004 pelas empresas *O'Reilly Media* e *MediaLive International* em uma série de conferências onde trataram sobre o tema, e então se popularizou. Este termo divide opiniões entre desenvolvedores e especialistas, alguns discordam do conceito e das idéias.

O precursor do termo *Web 2.0*, O'REILLY define a *Web 2.0*:

Web 2.0 é a mudança para uma internet como plataforma, e um entendimento das regras para obter sucesso nesta nova plataforma. Entre outras, a regra mais importante é desenvolver aplicativos que aproveitem os efeitos de rede para se tornarem melhores quanto mais são usados pelas pessoas, aproveitando a inteligência coletiva.

A principal mudança não se refere a especificações técnicas da *Web*, mas na visão dos usuários e desenvolvedores, no ambiente e nas motivações.

Um conjunto de empresas (entre elas *Google*, *Amazon*, *Yahoo* e *Microsoft*) e estudiosos da *Web* definem uma série de regras que ajudam a definir a *Web 2.0*, entre as regras, estão: a visão das aplicações *Web* como um processo de comprometimento com o usuário; a reutilização dos dados e serviços de outros sistemas; a visão dos dados como sendo o principal elemento e aumento do retorno aos dados gerados pelos usuários.

Houve mudanças na visão dos sistemas, que agora são tratados como serviços, são pagos mensalmente ou pelo seu uso. Tecnologias de desenvolvimento se uniram para proporcionarem maior interatividade e facilidade aos usuários. Esta melhoria fez com que os usuários que antes não tinham conhecimento para manipular as ferramentas existentes, agora possam gerar e organizar conteúdo facilmente, e este conteúdo evoluir os sistemas a ponto de trazer maiores vantagens aos próprios usuários. A *Web 2.0* trouxe novas formas de negócios, alterações no jornalismo, marketing, publicidade, e outros fatores.

Baseando-se nas alterações no ambiente de interação e participação dos usuários com os sistemas e aplicações baseados na *Web*, a *Web 2.0* evolui se tornando cada vez mais dinâmica, fornecendo facilidades para os usuários,

aumentando as respostas sobre os dados gerados e ampliando as possibilidades de negócios.

2.3 Engenharia da *Web*

Pressman define de maneira geral a engenharia da *Web*:

A engenharia da Web (WebE) é o processo usado para criar WebApps de alta qualidade. A WebE não é um clone perfeito da engenharia de software, mas toma emprestados muitos dos conceitos e princípios fundamentais da engenharia de software. Além disso, o processo WebE enfatiza atividades técnicas e de gestão similares. Há diferenças sutis no modo pelo qual essas atividades são conduzidas, mas a filosofia dominante determina uma abordagem disciplinada para o desenvolvimento de um sistema baseado em computador. (PRESSMAN, 2010, p.378)

A engenharia da *Web* divide-se em camadas, sendo estas os processos, tecnologias e métodos da engenharia de software adaptados às características da *Web*.

Os processos são escolhidos de acordo com os atributos dos aplicativos *Web*. Se há um imediatismo e evolução contínua, o modelo de processo escolhido pode ser um modelo de processo ágil, que produz versões em uma rápida sequência, como por exemplo, o modelo de processo *Scrum* e o modelo de processo *Extreme Programming (XP)*, entre outros. No entanto, se há um período de tempo maior para o desenvolvimento da aplicação *Web*, um modelo incremental pode ser uma opção. Como exemplo de outros modelos de processos há o Modelo em Cascata, o Modelo RAD e o Modelo Espiral.

As tecnologias estão disponíveis em uma grande quantidade para facilitar o desenvolvimento, elas englobam a descrição de conteúdo e linguagens de modelagem, linguagens de programação, navegadores, ferramentas de conectividade com banco de dados, ferramentas de segurança, entre outras.

Pressman (2010, p. 382) define de maneira geral o objetivo dos métodos da engenharia da *Web*: “O panorama de métodos da WebE engloba um

conjunto de tarefas técnicas que habilitam um engenheiro *Web* a entender, caracterizar e então construir uma WebApp de alta qualidade.”.

Os métodos se dividem em métodos de comunicação, métodos de análise de requisitos, métodos de projeto e métodos de teste. Estes métodos adotam muitos dos mesmos conceitos e princípios dos métodos da engenharia de software, porém, devem ser adaptados para acomodar as características especiais dos sistemas e aplicações baseados na *Web*.

Nas próximas seções há uma visão geral sobre cada um destes métodos da engenharia da *Web*.

2.3.1 Formulação e Planejamento

As duas primeiras atividades da engenharia da *Web* são Formulação e Planejamento. Estas atividades focalizam a obtenção de um caminho a ser seguido pela equipe de engenharia da *Web*, sendo este caminho, o mais curto e eficiente possível, desenvolvido considerando basicamente as necessidades dos interessados do negócio e os riscos gerados por estas necessidades.

Um princípio fundamental da engenharia de *software* e a base da atividade de Formulação segundo Pressman (2010, p. 389) é: “Entenda o problema antes de começar a resolvê-lo, e certifique-se de que a solução que você venha a conceber seja o que as pessoas realmente desejam.”.

Em busca do entendimento do problema, na atividade de Formulação há a identificação dos interessados e a definição do contexto do negócio, definição das principais metas, ou seja, como o sucesso é medido, e quais tipos de problemas específicos a aplicação soluciona.

Mais especificamente comunica-se com o cliente, para obter as funcionalidades desejadas, as categorias de usuários existentes e as características globais do negócio. Casos de uso também são definidos para cada classe de usuário. Essa coleta de requisitos é realizada de maneira global, o mais importante no momento é a coleta de informação deixando a avaliação dessas informações para a fase de Modelagem de Análise. As atividades da formulação à modelagem de

análise representam uma continuidade. Segundo Pressman (2010), a formulação focaliza um “quadro geral” das necessidades e objetivos de negócio e informação relacionada. A informação adquirida na Formulação serve como base para o Planejamento e Modelagem de Análise.

Mesmo considerando os curtos prazos de entrega, o planejamento de aplicativos para a *Web* é necessário, mas as tarefas são menos formais que em projetos de engenharia de software, elas objetivam uma abordagem ágil.

Como definição geral de planejamento considera-se:

Planejamento estabelece um plano de trabalho, desenvolve estimativas para avaliar a possibilidade das datas de entrega desejadas, considera riscos, define um cronograma e estabelece mecanismos de monitoramento e controle. (PRESSMAN, 2010, p. 390)

2.3.2 Modelagem de Análise

O imediatismo de resultados das aplicações para a *Web* entra em contradição com uma Modelagem de Análise detalhada, mas a realização de uma Modelagem de Análise menos detalhada ajuda a obter um processo de desenvolvimento com objetivos e funcionalidades mais claras e uma aplicação com boa manutenibilidade.

A importância da Modelagem de Análise se dá principalmente pelo esclarecimento dos tópicos que são improváveis de se modificar em curto prazo da aplicação *Web*. Para entrar em conformidade com as características das aplicações *Web*, um modelo mais detalhado e concreto de requisitos não é recomendado, pois, as informações se modificam com frequência. A Modelagem de Análise mais detalhada deve ser adotada quando o aplicativo *Web* é grande ou complexo, ou o número de interessados é grande, ou as metas e objetivos definidos durante a fase de Formulação vão afetar os fundamentos básicos do negócio, ou o sucesso da aplicação *Web* terá um forte efeito no sucesso do negócio. Em outros casos é possível utilizar as informações coletadas na Formulação como base para as próximas fases da engenharia da *Web*.

Tomando como base as informações coletadas pela atividade de Formulação, a Modelagem de Análise inicia com uma coleta mais rigorosa de requisitos. Os casos de uso existentes são expandidos, e a medida do surgimento de novos requisitos, novos casos de uso podem ser desenvolvidos.

Continuando, a Modelagem de Análise foca em quatro tópicos fundamentais da aplicação *Web*. São eles:

O Modelo de Conteúdo que contém todas as classes de análise obtidas através de casos de uso expandidos. Contêm também elementos estruturais que estabelecem uma visão dos objetos de conteúdo, essa visão são os relacionamentos ou hierarquias obtidos de diagramas entidade-relacionamento ou árvore de dados. Um objeto de conteúdo é qualquer item de informação apresentada ao usuário.

O Modelo de Interação que descreve interação entre usuário e aplicativo *Web*. Contêm casos de uso, diagramas de sequência *UML*, diagrama de estados *UML* e um protótipo de interface com o usuário.

O Modelo Funcional cujo objetivo é atender a quaisquer funções de processamento que são iniciadas pelo usuário e as operações com as classes de análise. Usa diagramas de atividade *UML* que ilustra um fluxo de processamento e decisões lógicas.

O Modelo de Configuração que descreve o ambiente que o aplicativo *Web* vai necessitar, do lado do servidor e do cliente. Para arquiteturas de configuração mais complexas, o diagrama de implantação *UML* pode ser utilizado.

Os elementos anteriores representam elementos de conteúdo, funcionais e suas interações com o usuário. Esses elementos de conteúdo podem estar relacionados com outros elementos de conteúdo, o que aumenta a complexidade navegacional. Para relacioná-los da melhor maneira possível há a Análise de Relacionamento-Navegação, que identifica os relacionamentos e define os mecanismos de navegação.

Todos os artefatos são utilizados para que finalmente os requisitos específicos sejam definidos seguindo para a próxima fase de engenharia da *Web*.

2.3.3 Modelagem de Projeto

Baseando-se nas informações da Modelagem de Análise, a Modelagem de Projeto da engenharia *Web* engloba seis atividades que geram um modelo de projeto onde é possível avaliar a qualidade da aplicação. Essa avaliação é realizada em um momento em que a aplicação ainda não foi codificada e nenhum conteúdo foi gerado, ou seja, o modelo de projeto pode ser melhorado no tempo mais adequado causando uma quantidade menor de alterações.

Como descrição geral dos principais objetivos e atividades do projeto consideram-se:

[...] o projeto leva a um modelo que contém a combinação adequada de estética, conteúdo e tecnologia. A combinação varia dependendo da natureza da aplicação web e, como consequência, as atividades do projeto que são enfatizadas também variam. (PRESSMAN, 2010, p. 426)

Este modelo de projeto deve ser detalhado, descrever como que os objetivos da aplicação definidos na Modelagem de Análise serão alcançados e como serão codificados na fase seguinte. Ao final do projeto, os elementos do modelo devem ser revisados com os interessados para descobrir erros e omissões.

As características de qualidade para aplicações baseadas na *Web* são as mesmas de softwares, destacando usabilidade, funcionalidade, confiabilidade, eficiência e manutenibilidade.

As atividades da Modelagem de Projeto são: projeto de interface, projeto estético, projeto de conteúdo, projeto de navegação, projeto arquitetural e projeto de componente, onde:

- O Projeto de interface descreve a interface com o usuário final e define o leiaute da aplicação *Web*;
- O Projeto estético engloba esquemas de cores, tamanho, estilos gerais, o uso de áudio, o uso de vídeo, entre outros. O objetivo é deixar a aplicação *Web* mais atraente na visão dos usuários;
- O Projeto de conteúdo especifica os objetos de conteúdo e seus relacionamentos. Também define o leiaute, a estrutura e o esboço do conteúdo apresentado;

- O Projeto de navegação especifica todo o fluxo de navegação da aplicação *Web*;
- O Projeto arquitetural define a estrutura da aplicação *Web* para interagir com o usuário, manipular o processamento interno, navegar e fornecer conteúdo. Um exemplo de arquitetura é a arquitetura MVC;
- O Projeto de componente segundo Pressman (2010), define as estruturas de dados, algoritmos, características de interface e mecanismos de comunicação alocados a cada componente.

2.3.4 Teste de Aplicações *Web*

O tempo reservado para as atividades da fase de teste muitas vezes não é o suficiente, e isso pode afetar seriamente a qualidade e sucesso da aplicação. Testes devem ocorrer em todas as fases do desenvolvimento para descobrir os erros antes da aplicação *Web* estar disponível para os usuários finais.

Considera a descrição geral e os objetivos:

Teste de aplicativos web é um conjunto de atividades relacionadas com um único objetivo: descobrir erros no conteúdo, na função, na usabilidade, na navegabilidade, no desempenho, na capacidade e na segurança de WebApp. (PRESSMAN, 2010, p. 455)

As aplicações *Web* são mais complexas para serem testadas, pois se localizam em uma rede, estão em contato com diferentes sistemas operacionais, navegadores, plataformas de hardware, protocolos de comunicação, entre outros. As atividades de testes visam às características de qualidade da aplicação *Web*, para cada característica uma estratégia de teste é definida. O planejamento das atividades de teste especifica o que devem ser testados, como devem ser testados, como os resultados serão avaliados e reutilizados, e para isso casos de testes são desenvolvidos.

Basicamente o processo de teste inicia-se com os elementos visíveis pelo usuário, que são o conteúdo e funcionalidade da interface, seguindo com projeto de arquitetura e de navegação. Alguns exemplos de testes são: teste de conteúdo, teste de interface, teste de navegação, teste de componente, teste de configuração, teste de segurança, teste de desempenho.

No teste de desempenho, por exemplo, o objetivo é determinar a maneira que o sistema responde a muitos usuários e muitas transações, para atingir tal objetivo são utilizados os testes de carga e esforço.

Considera a visão geral sobre o teste de desempenho:

[...] é usado para descobrir problemas de desempenho que podem resultar da falta de recursos do lado do servidor, largura de banda de rede inadequada, capacidades de banco de dados inadequadas, falha ou fragilidade das capacidades do sistema operacional, funcionalidade da aplicação web mal projetada e outros tópicos de hardware ou software que podem levar à degradação do desempenho cliente/servidor. (PRESSMAN, 2010, p. 474)

2.4 Sistema de Banco de Dados

Um sistema de banco de dados tem como objetivo suprir a necessidade de armazenamento de informações de maneira organizada.

Considera sobre um sistema de banco de dados:

Um sistema de banco de dados é basicamente um sistema computadorizado de armazenamento de registros; isto é, um sistema computadorizado cujo propósito geral é armazenar informações e permitir ao usuário buscar e atualizar essas informações quando solicitado. (DATE, 2000)

Os dados são armazenados de maneira a se tornarem integrados e compartilhados. A integração se deve ao fato dos dados serem únicos, a redundância ocorre apenas para indicar a localização correta para os devidos acessos, ou seja, a referência aos dados. Compartilhados, pois, os dados podem ser

acessados sem provocar qualquer dano à consistência, de maneira concorrente por diversos usuários, que podem possuir diferentes objetivos.

Um sistema de banco de dados é dividido em algumas camadas. Para garantir a eficiência nas operações solicitadas pelos usuários, há uma camada de software que está entre o banco de dados físico e os usuários, que é chamada do Sistema Gerenciador de Banco de Dados (SGBD). Um SGBD é responsável por gerenciar o acesso, a manipulação e a organização dos dados, fornece uma interface para os usuários que possibilita a inserção, remoção, atualização e visualização, sendo estas atividades realizadas de maneira otimizada.

Segundo Date (2000), o SGBD fornece aos usuários uma visão do banco de dados um tanto elevada acima do nível de *hardware*, e ela admite operações dos usuários que são expressas em termos dessa visão de nível mais alto.

O banco de dados físico ou base de dados é um conjunto de dados persistentes que se relacionam e tratam de um mesmo assunto. Esses dados podem ser de qualquer área de negócio. Os dados são persistentes, pois, após serem inseridos pelo usuário através de um SGBD, só serão removidos com uma solicitação explícita do usuário, ou seja, continuarão existindo mesmo com o término de uma execução de programa por exemplo.

Os sistemas de banco de dados proporcionam um controle centralizado dos dados. Algumas vantagens são:

- **Os dados podem ser compartilhados:** novas aplicações podem utilizar os mesmos dados armazenados;
- **A redundância pode ser reduzida:** dados são armazenados somente uma única vez e referenciados;
- **A inconsistência pode ser evitada:** a redução da redundância dos dados garante a correta atualização, ou seja, a consistência;
- **O suporte a transações pode ser fornecido:** garantia de que uma sequência de operações será efetuada com sucesso;
- **A integridade pode ser mantida:** é a garantia de que não há informações não autorizadas pelas regras de negócio implementadas;

- **A segurança pode ser reforçada:** pelo fato dos dados estarem centralizados, necessita-se de uma maior segurança. Restrições de segurança podem ser aplicadas a determinados tipos de acessos e a cada item de informação no banco de dados.

Os sistemas de banco de dados são divididos em categorias de acordo com a sua estrutura de dados e os operadores que fornecem ao usuário. A categoria dominante no mundo comercial e acadêmico são os sistemas de banco de dados relacionais, esses sistemas se baseiam numa teoria chamada modelo relacional de dados que surgiu em 1969-1970 e se destaca pela simplicidade e eficiência.

O modelo relacional admite a interpretação dos dados de forma muito direta. Os dados são representados por um conjunto de tabelas, onde cada linha de uma tabela pode ser interpretada facilmente. Os operadores oferecidos podem agir sobre diversas tabelas ao mesmo tempo e sobre as linhas e colunas de cada uma.

A maneira com que os usuários se comunicam em um nível alto de abstração com sistemas de banco de dados relacionais é através da linguagem *SQL* (*Structure Query Language*). A *SQL* admite operações de definição e manipulação de dados, ou seja, é possível criar uma base de dados com as devidas regras de negócio da organização, inserir dados e posteriormente consultá-los e atualizá-los,

A Figura 2 apresenta a arquitetura simplificada de um sistema de banco de dados.

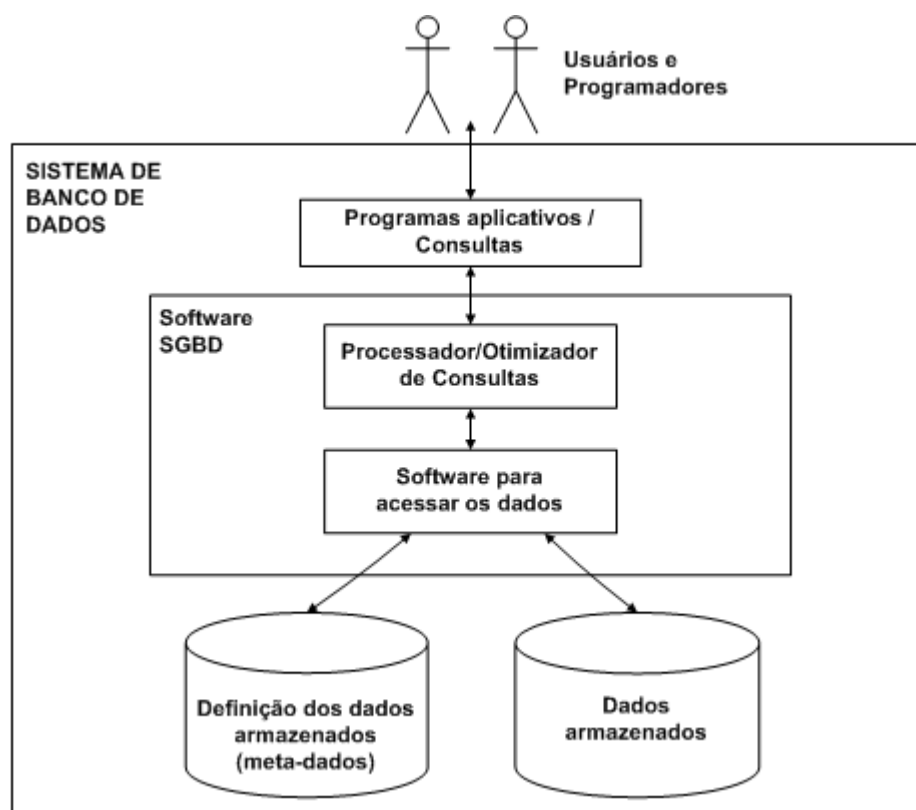


Figura 1: Arquitetura de um sistema de banco de dados
 Fonte: Rodrigo Sanches (2005)

2.4.1 SGBD MySQL

O *MySQL* é um SGBD relacional, de código aberto. Inicialmente foi projetado para aplicações de pequeno e médio porte, mas evoluiu muito e hoje em dia atende a também aplicações de grande porte, realiza operações sobre tabelas com milhões de registros com eficiência. Foi desenvolvido utilizando as linguagens de programação C e C++ e é compatível com mais de 20 plataformas, entre elas, estão as mais conhecidas como *Linux*, *Windows*, *Mac* e *Solaris*.

É baseado em duas licenças, dependendo do tipo de uso. Para situações com suporte diferenciado e pacotes com mais ferramentas a licença é comercial. A segunda licença é a de software livre *GNU-GPL (General Public License)*.

Tornou-se muito popular por ser um *software* livre de muita eficiência, pela facilidade de uso e integração com algumas linguagens de programação.

Algumas características do *MySQL* são:

- **Multithreads:** utiliza a programação de threads, gera um aumento na velocidade de processamento;
- **Velocidade:** possui recursos como cachês em consultas, índices e algoritmos de buscas;
- **Segurança:** utiliza criptografia no tráfego de senhas e ACL (Listas de Controle de Acessos);
- **FullText search:** com tabelas do tipo *MyISAM*, oferece a busca eficiente de texto em uma grande quantidade de dados. Utiliza índices do tipo *FULLTEXT*;
- **Capacidade de armazenamento:** com tabelas do tipo *InnoDB*, é possível armazenar 65.536 TB;
- **Store procedures:** são funções armazenadas no servidor escritas em *SQL*, diminuem o tráfego na rede e melhoram o desempenho;
- **Triggers:** também são funções armazenadas no servidor, mas são ativadas indiretamente, ou seja, antes ou depois de expressões *SQL* de inserção, alteração ou exclusão;
- **Visões:** são nomeações de consultas ao banco de dados definidas pelo usuário que podem reunir diversas tabelas, facilitando um posterior acesso;
- **Cursor:** percorrem os registros de uma tabela, consegue-se acessar uma linha e realizar operações como exclusão, atualização e aritméticas. Basicamente, define-se um conjunto de dados para serem percorridos através de laços de repetição;
- **Transações:** unidade de execução que garante um banco de dados consistente após uma sequência de comandos *SQL*, mesmo ocorrendo falhas;
- **Replicação:** estratégia que utiliza dois ou mais servidores de dados réplicas um do outro, ou seja, todos os servidores existentes são iguais, quando um sofre alguma alteração comunica aos outros para efetuarem a mesma alteração. Alguns benefícios são robustez e menor tempo de resposta às consultas. Para robustez, em caso de problemas com um servidor, pode-se trocá-lo por outro e para

menor tempo de resposta às consultas, as consultas são divididas entre os servidores para diminuir a carga de processamento. A replicação utiliza o tráfego de rede para comunicação, assim, servidores em outros locais físicos também podem ser sincronizados;

- **Clusterização:** Objetiva a melhoria de qualidade dos serviços, também utiliza dois ou mais servidores para aumentar o desempenho. Diferencia da replicação na maneira em que é implementada e nos recursos que oferece.

Este SGBD relacional, é capaz de manipular grandes bancos de dados eficientemente, possui confiabilidade, simplicidade de configuração e operação, velocidade satisfatória comparando-se com seus concorrentes, e é uma ferramenta de código aberto

2.5 Tecnologias aplicadas na implementação do Websislapam

2.5.1 XML

Em 1996, a W3C (*World Wide Web Consortium*) criou a linguagem XML (*Extensible Markup Language*) que objetiva principalmente o compartilhamento de dados entre sistemas. Um documento XML pode ser personalizado para atender a uma necessidade específica de um sistema e estruturado para permitir que os dados sejam trocados por meio de uma linguagem facilmente compreendida por *softwares* e seres humanos.

Segundo Deitel et. al. (2001, p. 152), “A XML é uma tecnologia para criar linguagens de marcação que descrevem dados de praticamente qualquer tipo, de uma forma estruturada.”.

Comparando com a linguagem *HTML*, considera que *XML* surgiu para complementar as funcionalidades, ou seja, se responsabilizar por uma funcionalidade que *HTML* não atende.

Comparando os princípios de *HTML* com *XML* considera:

HTML é uma linguagem de marcação para *apresentação*, legível e exibida por praticamente qualquer navegador *Web* moderno, enquanto XML é uma linguagem de marcação para *conteúdo*, sem nenhum elemento de apresentação inerente ou embutido: apenas elementos de definição de conteúdo. (TITTEL, 2002, p. 13)

Com *XML* é possível realizar transferência de dados com processamento otimizado, devido a sua estrutura de marcações do documento. É garantido à consistência dos dados de acordo com o que foi especificado através da *DTD (Document Type Definition)* e, além do processo ser simplificado, é totalmente extensível às necessidades dos sistemas, ou seja, o autor do sistema pode definir suas próprias marcações.

A estrutura da linguagem *XML* é composta basicamente por um Documento *XML*, *DTD* ou Esquema, *Namespaces* e *Parser XML*, onde:

Um documento *XML* geralmente armazenados em arquivos textos com a extensão ‘. xml’. Neste documento são definidos os componentes utilizados e a estrutura física do documento, ou seja, a estrutura dos dados e informações reais.

A *DTD* especifica os elementos utilizados no Documento *XML*, para cada elemento define um nome, um conteúdo e um contexto.

Conceitualmente, considera sobre a *DTD*:

A Definição de Tipo de Documento (*DTD*) é a chave para o significado de elementos, atributos, estrutura lógica e contexto. Ela é necessária para que um documento seja válido. A *DTD* define os elementos que são utilizados e os atributos que eles precisam, além de permitir que o documento seja intercambiado e compreendido por outro software. (TITTEL, 2002, p.19)

Pode estar definida no documento *XML* ou externamente, é composta por Elementos, Atributos, Entidades, *PCDATA* e *CDATA*.

Um Esquema é uma alternativa às *DTDs* para a validação de um documento *XML*. Possui mais recursos que a *DTD*, suportando a especificação de tipo de dados e *namespaces*. Utiliza a sintaxe *XML* devendo ser válido e podendo

ser manipulado como qualquer outro documento *XML*. O uso de Esquemas é preferível a *DTD* quando é necessário identificar se o conteúdo é de determinado tipo de dado, por exemplo, em interfaces com banco de dados.

Sobre *namespaces*, considera o fato de *XML* permitir a livre criação de marcações, podendo ocorrer colisões de nomes para elementos diferentes. Em caso em que estes elementos são utilizados no mesmo documento *XML*, utiliza *namespaces*. Basicamente atribui um prefixo à marcação que deseja identificar.

De acordo com Tittel (2002) o *parser XML* é o analisador sintático de *XML*. Realiza a leitura do documento *XML* verificando sua sintaxe e relatando os erros encontrados, e ao final, permite acesso ao documento ou não.

Segundo Deitel et. al. (2001, p. 153), “a sintaxe da *XML* requer um único elemento como nó raiz, uma marca de abertura e de finalização para cada elemento, marcas corretamente aninhadas e valores de atributos entre aspas.”.

Os elementos definem regras de formação para um tipo particular de conteúdo. Os atributos adicionam informações aos elementos, estes atributos geralmente auxiliam na identificação dos dados.

A Figura 1 apresenta documento *XML*, cuja validação é garantida por meio da *DTD*.

```

01. <?xml version="1.0" encoding="UTF-8"?>
02. <? DOCTYPE dtd_produto [
03.   <!ELEMENT produto (nome, quantidade_estoque) >
04.   <!ELEMENT nome (#PCDATA) >
05.   <!ELEMENT quantidade_estoque (#PCDATA) >
06. ]>
07. <produto>
08.   <nome>NOME DO PRODUTO</nome>
09.   <quantidade_estoque>100</quantidade_estoque>
10.</produto>

```

Figura 2: Documento *XML* com *DTD*

- Linha 01: declaração *XML*, a qual especifica a versão *XML* e a codificação *UTF-8*.
- Linha 02 a 06: *DTD* que valida o documento *XML*.

- Linha 03: especificação do elemento 'produto'. Garante que o elemento 'produto' tenha na sequência os outros elementos 'nome' e 'quantidade_estoque'.
- Linha 04 e 05: especificação dos elementos 'nome' e 'quantidade_estoque'. '#PCDATA', indica que não podem conter outros elementos, apenas dados.
- Linha 07 e 10: raiz de abertura e fechamento do documento *XML* respectivamente.
- Linha 08 e 09: elementos 'nome' e 'quantidade_estoque' com seus respectivos conteúdos.

A *DTD* pode estar em um arquivo separado e não é obrigatória em um documento *XML*.

2.5.2 *PHP*

O *PHP* (*Hipertext Preprocessor*) teve sua primeira versão desenvolvida em 1994 por Rasmus Lerdorf e ao longo do tempo evoluiu e em 2004 foi lançada a versão cinco.

Como uma visão geral do *PHP*, considera:

O *PHP* é uma linguagem para a criação de scripts para a *Web* do lado servidor embutidos em *HTML*, cujo código-fonte é aberto, e que é compatível com os mais importantes servidores *Web* (especialmente o *Apache*). O *PHP* permite incorporar fragmentos de códigos em páginas de *HTML* normais – código esse que é interpretado à medida que suas páginas são oferecidas aos usuários. O *PHP* também serve como uma linguagem de “cola”, facilitando a conexão de suas páginas *Web* com o banco de dados do lado servidor. (CONVERSE; PARK, 2003)

É uma linguagem de desenvolvimento *Web* interpretada, tem pouco a ver com a aparência de uma página *Web*, pode gerar conteúdo *HTML* e é

responsável pelo processamento, acesso a banco de dados e implementação das regras de negócio dos sistemas e aplicações baseados na *Web*.

Possui o código fonte aberto e é gratuito, pode ser executado em várias plataformas, é destacado por muitos desenvolvedores por sua leveza, capacidade, confiabilidade e facilidade de uso. Possui também funções de manipulação de *arrays*, *strings*, para a comunicação com alguns dos *SGBDs* mais populares, entre eles, o *MySQL* e o *Oracle* e, suporte a alguns protocolos, como o *SNMP*, *HTTP* e *SOAP*. Sua sintaxe é semelhante ao *C/C++*, os tipos de dados são definidos em tempo de execução e possui suporte a funções. Na versão cinco, tem suporte a orientação a objetos com melhor desempenho e mais vantagens que as outras versões.

Em se tratando de sessões de usuários, ou seja, a identificação de um período de tempo em que um usuário visualiza um número de diferentes páginas *Web* em seu navegador, oferece funções para trabalhar com *cookies* e variáveis ocultas no servidor.

Considera uma definição de *cookies*:

Um *cookie* é um pequeno fragmento de informação que é retido na máquina do cliente, quer na memória da aplicação navegadora, quer em um pequeno arquivo gravado no disco rígido do usuário. Esse arquivo contém um par nome/valor – *configurar um cookie* significa associar um valor com um nome armazenar esse par no lado cliente. *Obter o ler* um *cookie* significa utilizar o nome para recuperar o valor. (CONVERSE; PARK, 2003)

O uso de *cookies* é indicado quando armazenar informações do usuário no servidor não é uma opção, e no caso de variáveis ocultas no servidor, o *PHP* fornece o array `$_SESSION` que pode armazenar informações do usuário no servidor e ser acessado em qualquer parte do código *PHP*.

Considerando que o *PHP* oferece orientação a objetos, funções que facilitam o acesso a banco de dados, velocidade de processamento, sintaxe semelhante ao *C/C++*, interação com *HTML* e boa documentação das funcionalidades, esta linguagem apresentou-se como uma boa opção para o desenvolvimento do *Websislapam*.

2.5.3 JpGraph

JpGraph é uma biblioteca escrita totalmente em *PHP* desenvolvida pela empresa *Asial Corporation* que possui licença gratuita para sistemas não comerciais.

Proporciona ao desenvolvedor de sistemas e aplicações *Web* uma maneira acessível de visualizar informações através de gráficos. Alguns dos gráficos disponíveis são: gráfico de linhas, de barras, de setores, entre outros. Disponibiliza também diversas opções de formatação para os gráficos, como por exemplo, gráficos em duas ou três dimensões, opções de cores, opções para formatar o conteúdo, títulos, legendas, entre outras.

Essa biblioteca pode ser vista como uma boa opção para gerar gráficos em *PHP*, pois, fornece uma grande quantidade de recursos que podem ser utilizados de maneira acessível por desenvolvedores com conhecimentos básicos na linguagem.

2.5.4 JavaScript e DHTML

JavaScript é uma linguagem de programação desenvolvida por Brendan Eich da *Netscape* e lançada em 1995.

É uma linguagem para programação *client-side*, ou seja, do lado do cliente, é executada no micro do usuário e não no servidor que hospeda a aplicação. Linguagem de *script*, que é executada no interior de programas ou em outras linguagens. Possui tipagem fraca e dinâmica, os tipos de dados são determinados em tempo de execução. Suporta os tipos primitivos de dados números, *strings*, *booleanos* e tipos especiais de objetos.

Alguns dos tipos especiais de objetos em *JavaScript* são: *arrays*, funções, *Date*, *Math*, *Error*, *RegExp* que representam expressões regulares, entre outros.

DHTML é um termo que descreve páginas *Web* dinâmicas e interativas, é a união de *HTML*, *JavaScript* e *CSS* com o Modelo de Objetos de Documentos (*DOM HTML*).

Considera uma descrição do *DOM HTML*:

[...] interface neutra com relação a linguagens e a plataforma, que permitirá a programas e *scripts* acessar e atualizar dinamicamente o conteúdo, a estrutura e o estilo de documentos. O documento pode ser posteriormente processado e os resultados desse processamento podem ser incorporados de volta à página apresentada. W3C (apud TITTEL, 2003, p. 83)

Considera outra descrição do *DOM HTML*:

[...] é uma interface de programas aplicativos para representar um documento (como um documento *HTML*) e acessar e manipular os vários elementos (como marcas *HTML* e *strings* de texto) que compõem esse documento. [...] ele representa documentos *HTML* (e *XML*) em uma estrutura de árvore e define propriedades e métodos para percorrer a árvore e examinar e modificar seus nodos. (FLANAGAN, 2004)

As janelas do navegador e o conteúdo das páginas são estruturados. Como exemplo de objetos do *DOM HTML* há os objetos: *window*, *document*, *history*, *location*, *forms*, *links*, entre outros. O objeto *window*, por exemplo, representa a janela do navegador e fornece acesso ao objeto *document* que representa o documento *HTML*, que por sua vez fornece acesso ao elemento *body* que pode ser alterado. O objeto *location* contém a *URL* exibida, quando seu conteúdo é alterado, o navegador muda automaticamente para outra página.

Essa combinação de tecnologias permite que a estrutura, conteúdo e estilo de documentos sejam alterados nas máquinas cliente fornecendo alguma interatividade, sendo assim, torna-se uma boa opção para o desenvolvimento de sistemas e aplicações baseados na *Web*.

2.5.5 Cascading Style Sheets (CSS)

Cascading Style Sheets (CSS – “folhas de estilo em cascata”) é uma linguagem de estilos desenvolvida pelo W3C. Segundo (DEITEL, DEITEL e NIETO, 2003) a CSS:

[...] permite especificar a apresentação dos elementos em uma página *Web* (espaçamento, margens, etc.) separadamente da estrutura do documento (cabeçalhos de seção, corpo do texto, links, etc.). Essa separação entre estrutura e apresentação simplifica a manutenção e a modificação do leiaute de um documento.

Uma folha de estilo consiste em uma lista de regras de formatação que definem os estilos. A CSS utiliza seletores para definir os elementos de marcação que um estilo se aplica. Alguns dos seus recursos são: formatar cores, formatar fontes, posicionar elementos, inserir imagens, definir leiaute, entre outros.

É recomendada para sistemas e aplicações baseados na *Web*, pois, tem como vantagem o fato de encapsular a formatação do conteúdo apresentado, o que aumenta a consistência e a qualidade do trabalho, facilitando o desenvolvimento e a manutenção.

2.5.6 Ajax

Ajax (em português “*JavaScript* e *XML* Assíncronos”) foi desenvolvido inicialmente por Jessé James Garret, é o uso combinado de tecnologias e recursos como *JavaScript*, *DOM* e *XML*, tem como objetivo principal diminuir o tráfego de dados na rede proporcionando uma maior interação do usuário com os sistemas e aplicações baseados na *Web*.

Nas aplicações *Web* clássicas, que são baseadas em páginas, o navegador realiza requisições *HTTP* das páginas inteiras desejadas ao servidor *Web*, que é responsável pela busca e processamento dos dados no banco de dados

de acordo com as regras de negócio da aplicação. A cada passo do usuário, é necessário buscar uma nova página no servidor *Web*, isto implica numa menor interatividade do usuário com a aplicação e tráfego de dados desnecessário.

No ambiente de uma aplicação com *Ajax*, um documento mais complexo com capacidade para tomar decisões é enviado ao navegador, este documento permanece durante toda a sessão do usuário, fazendo a requisição de dados e não de conteúdo, sem precisar recarregar toda a página, recarregando apenas as partes importantes. Esse ambiente aproxima as aplicações *Web* das aplicações *desktop*, combinando a interação do usuário com as requisições ao servidor *Web*.

Essa interação se dá por meio do objeto *XMLHttpRequest*. Uma requisição *HTTP* é feita ao servidor *Web* através do objeto *XMLHttpRequest*, o servidor *Web* então, realiza o processamento necessário e devolve a resposta ao objeto *XMLHttpRequest* através de texto ou *XML* que manipulada por *JavaScript* atualiza o *DOM* da página atual.

O interessante é que a interação do usuário não precisa parar quando o servidor é solicitado. Os sistemas e aplicações baseados na *Web* podem utilizar recursos *Ajax* seguindo a metodologia clássica baseada em páginas, ou seja, utilizar *Ajax* somente onde houver necessidade em benefício da interatividade. Pode ser muito útil para auxiliar o usuário no preenchimento de formulários e realizações de buscas, sendo assim, uma boa opção para proporcionar interatividade aos sistemas e aplicações baseados na *Web* na busca de informações no servidor *Web*.

2.5.7 JQuery e JQuery UI

JQuery é uma biblioteca *JavaScript* de código aberto que tem como objetivo simplificar a utilização dos scripts que interagem com o *HTML*. Foi desenvolvida em 2006 por John Resig, mas apesar do pouco tempo é muito utilizada entre os sites mais visitados do mundo.

Oferece recursos para superar a incompatibilidade de navegadores, diminuir a quantidade de linhas de códigos, utilizar plugins, trabalhar com *Ajax*, *DOM* e *CSS*, entre outros.

jQuery UI é uma biblioteca de recursos de interface e interação com efeitos de animação de alto nível. Oferece recursos como janelas, botões, menus, ícones, auto-completar, entre outros.

Entre as principais vantagens do uso dessas bibliotecas estão a grande quantidade de bibliotecas, o acesso aos elementos do *DOM* e uso do *Ajax* com poucas linhas de códigos, além de uma interface com elementos visualmente atraentes.

3 SISTEMA DE AVALIAÇÕES DE PESSOAS WEBSISLAPAM

O sistema denominado Websislapam, desenvolvido com tecnologias *Web*, fornece recursos para que entidades possam realizar processos de avaliação de pessoas em áreas de interesse heterogêneas, ou seja, processos de avaliação independentes da área de pesquisa.

Um processo de avaliação de pessoas envolve organizar um conjunto de perguntas para as quais se esperam tipos predeterminados de respostas, sendo essas respostas adquiridas através de coletas de dados. Uma coleta de dados consiste em uma entrevista feita por uma pessoa capacitada a uma pessoa que se deseja avaliar. Os dados que são coletados podem ser analisados, visando identificar progressos, padrões, problemas ou especificar objetivos que tenham sido atingidos em grupos de pessoas específicos ou em uma única pessoa.

O Websislapam tem a capacidade de criar uma avaliação e organizar a equipe de pessoas que trabalham para o seu sucesso. Entre as principais funcionalidades, destacam as seguintes:

- Gera um conjunto dinâmico de perguntas, ou seja, oferece aos usuários a liberdade de criar e explorar as perguntas que lhes são interessantes, que estão no contexto de suas áreas de pesquisas e entre os objetivos de suas avaliações;
- Permite a realização de coletas de dados, onde oferece interatividade ao usuário, que pode selecionar a pessoa a ser avaliada e obter informações históricas recentes desta pessoa, o que torna o processo mais ágil;
- Oferece recursos que induzem o usuário a realizar uma coleta de dados com uma taxa baixa de erros, erros estes que são especificados e não são aceitos pelo conjunto de perguntas criado anteriormente e aplicado na coleta de dados;
- Mantém a base de dados com cadastros das pessoas que participam das avaliações, sendo estes cadastros das pessoas que são avaliadas e das pessoas que aplicam as entrevistas, ou melhor, as coletas de dados;

- Possibilita diferentes níveis de acessos sobre os dados coletados e outras opções do sistema, garantindo que somente as pessoas com o conhecimento, responsabilidade e confiança adequada manipulem a avaliação;
- Como há um cadastro das pessoas que são avaliadas, estas pessoas podem participar de um número ilimitado de coletas de dados, desta maneira gerando grande quantidade de dados;
- Oferece recursos para a visualização dos dados coletados através de gráficos, que acompanham uma pessoa em seu desenvolvimento ou comparam grupos de pessoas específicos, que são selecionados a critérios do usuário;
- Oferece a opção para importar e exportar dados cadastrais das pessoas que são avaliadas.

Além de oferecer ao usuário a possibilidade de gerar conjuntos dinâmicos de perguntas, coletas de dados com alguma interatividade, cadastro das pessoas que são avaliadas e pessoas que trabalham para o sucesso da avaliação, diferentes níveis de acessos aos dados e a possibilidade de visualizar os dados coletados por meio de gráficos, como brevemente descrito acima, o Websislapam tem a vantagem de estar disponibilizado na *Web* podendo ser utilizado por diversas entidades ao mesmo tempo. Por meio da estrutura do banco de dados e controle de usuários, uma entidade tem exclusividade sobre seus dados, sendo estes visualizados além da própria entidade, somente pelo administrador.

As avaliações são organizadas para melhor entendimento e acompanhamento. As formas de organização são os projetos e subprojetos, que possuem pessoas vinculadas com diferentes níveis de acessos, possuem também pessoas que são avaliadas, conjuntos de perguntas, dados coletados e seus relatórios específicos, como por exemplo, relatórios quantitativos e gráficos, sendo que todos identificam o estado atual dos projetos. Os dados de um projeto dentro de uma entidade não se confundem com os dados de outros projetos, eles são exclusivos.

O escopo do Websislapam está de acordo com os objetivos e funcionalidades descritos acima. Para avaliações em que os recursos do sistema para análise dos dados coletados não são suficientes, o sistema disponibiliza uma opção para a exportação dos dados coletados.

3.1 Visão Geral

O Websislapam foi desenvolvido utilizando o paradigma de programação orientado a objetos, o padrão de arquitetura de software *MVC* e um sistema de banco de dados relacional. A linguagem *PHP* foi utilizada no lado do servidor, com o *SGBD MySQL*. No lado do cliente foi utilizada a linguagem de programação *javascript*, a combinação de tecnologias *DHTML*, e para fornecer interatividade com o usuário foi utilizada a biblioteca *Jquery* com a combinação de tecnologias *Ajax*. Para a importação e exportação dos dados foi utilizada a linguagem *XML*. E para gerar os gráficos disponíveis foi utilizada a biblioteca *JpGraph*.

A Figura 3 apresenta o diagrama de casos de uso com a interação entre o Websislapam e o ambiente, ou seja, os atores e as principais funcionalidades. Os casos de uso são organizados de acordo com as regras de negócios, ou seja, por Entidade, Projeto e Subprojeto.

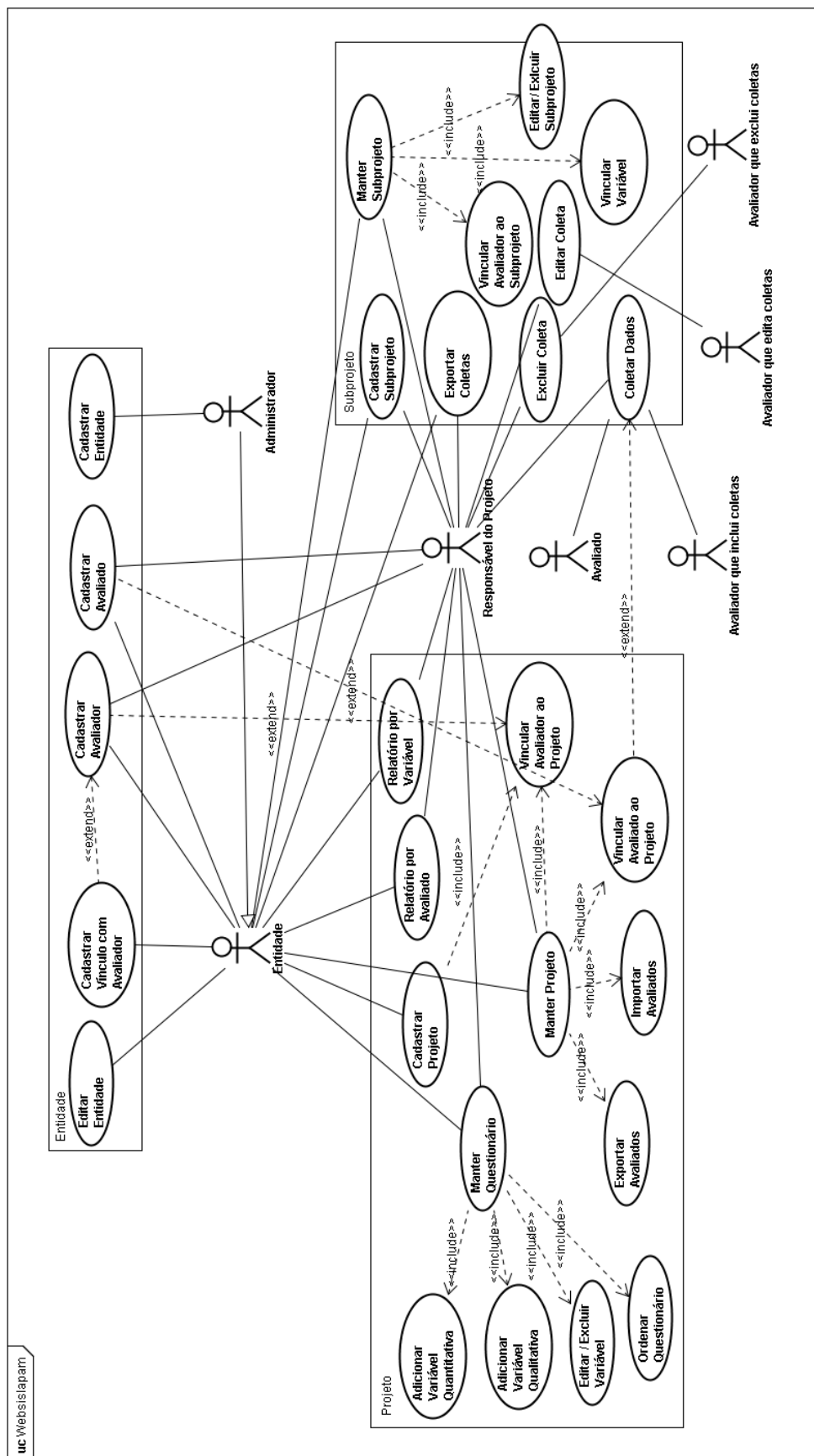


Figura 3: Diagrama de casos de uso

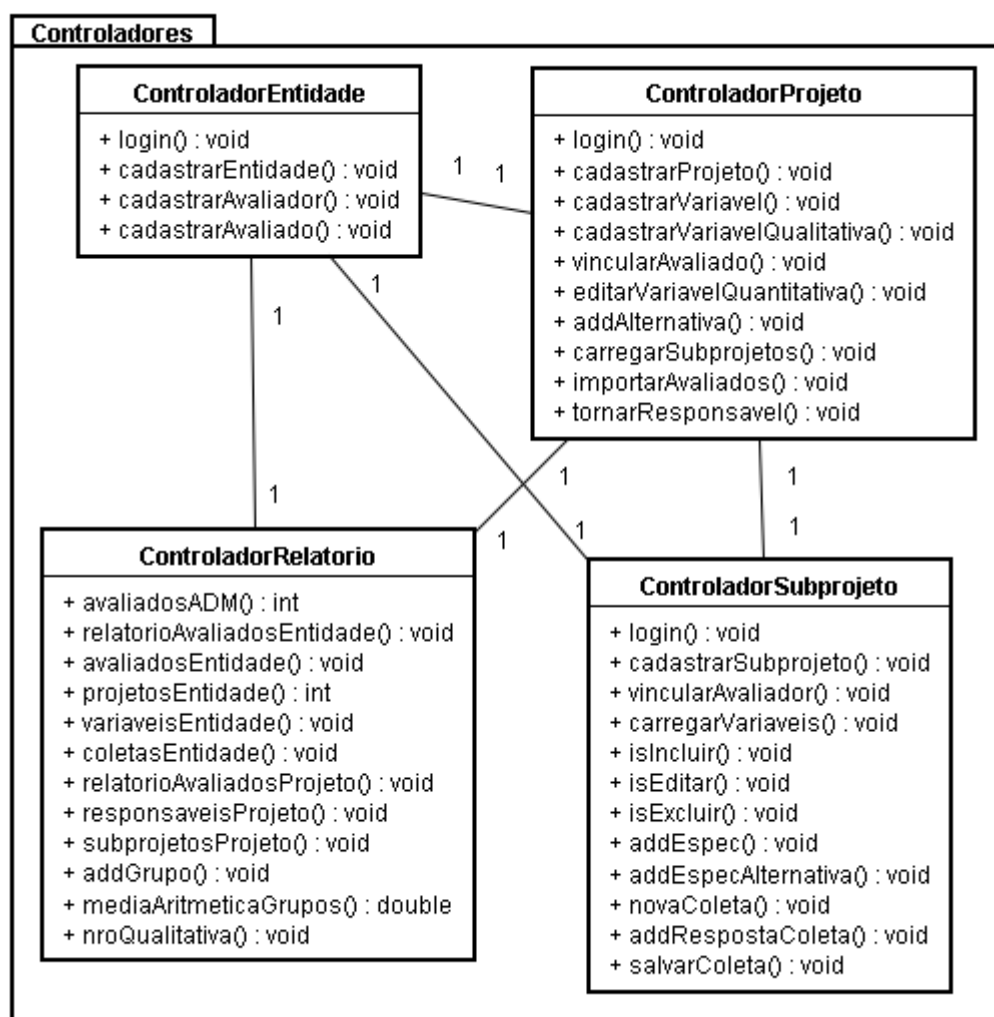


Figura 4: Camada de controladores simplificada

Em relação a modelagem de classes, o sistema é organizado em três camadas, são elas: Controladores, Modelo e Persistência.

A camada controladora é composta por quatro controladores fachadas, ou seja, controladores que recebem mensagens da interface, tratam os eventos do sistema e armazenam o estado dos casos de uso. Os controladores tratam os eventos referentes às entidades, aos projetos, aos subprojetos e aos relatórios, como apresenta a Figura 4.

A Figura 5 apresenta as classes da camada Modelo que também atendem às regras de negócios. E a Figura 6 apresenta as classes que se comunicam com o banco de dados.

Para demonstrar um exemplo de tratamento de eventos, a Figura 7 apresenta um diagrama de sequência para o cadastro de uma entidade, com a interação entre os objetos em função do tempo. A interface comunica a camada dos

controladores, que seguindo às regras de negócios, comunicam a camada modelo e a camada persistência.

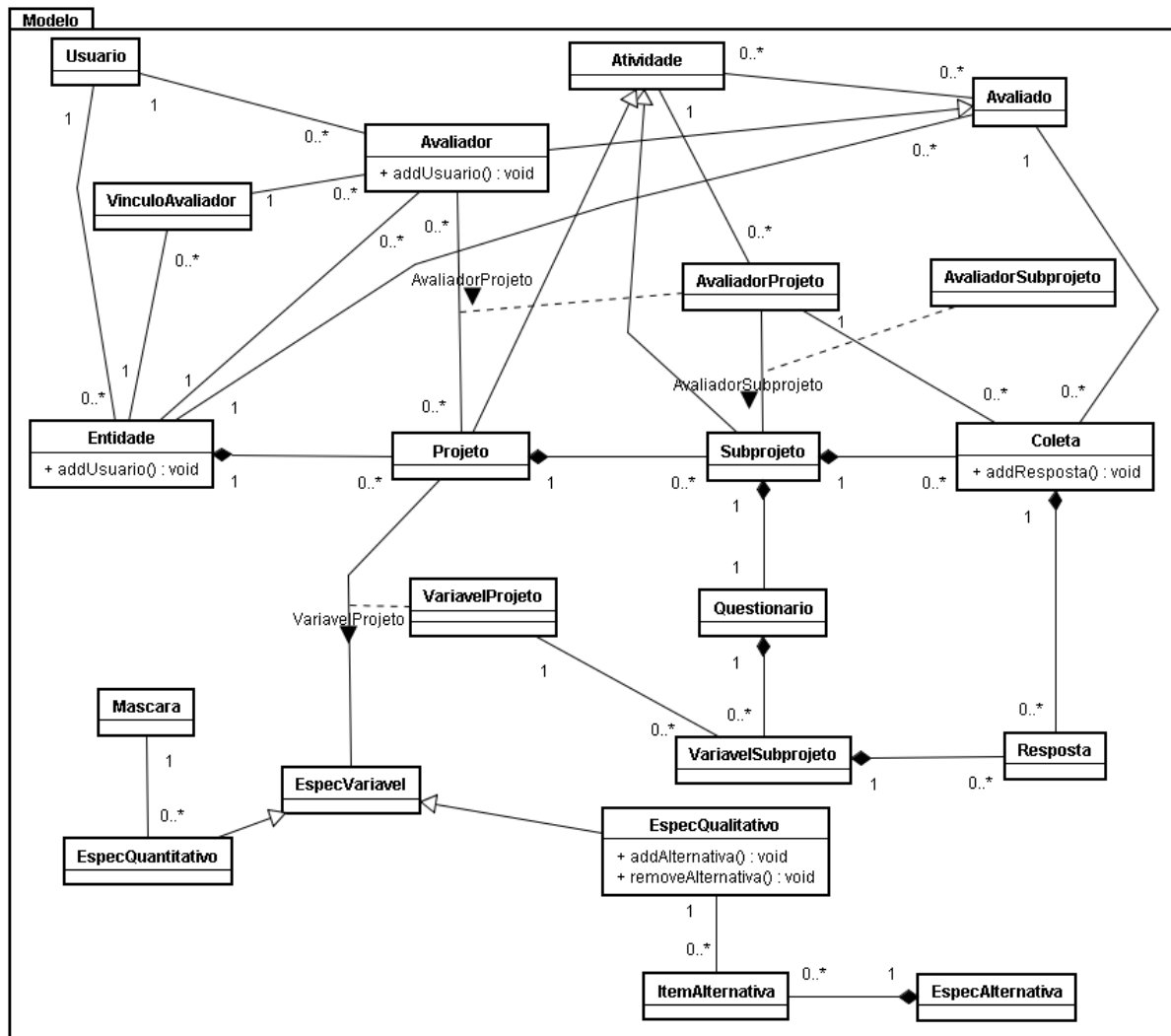


Figura 5: Camada do modelo simplificada

O Websislapam é dividido funcionalmente. Primeiramente, a estrutura do ambiente de trabalho, que identifica a entidade que utiliza o sistema e seus projetos é definida, juntamente com as pessoas que atuam nas avaliações. Após essa primeira definição, é possível gerar e organizar conjuntos de perguntas, o que possibilita o início das coletas de dados. Após as coletas de dados, os dados coletados podem ser analisados através dos relatórios disponíveis ou podem ser exportados para análise em outras ferramentas.

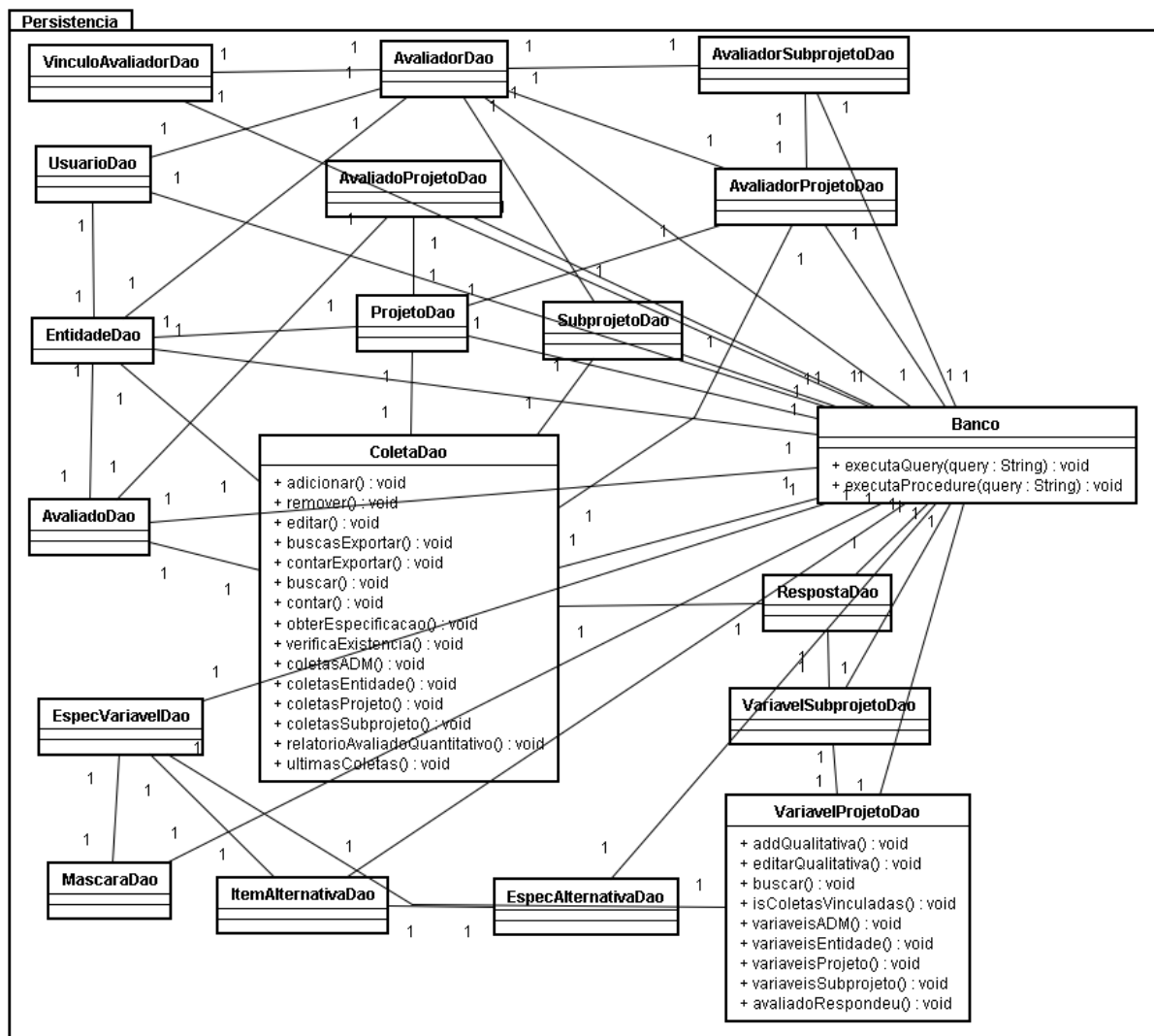


Figura 6: Camada de persistência simplificada

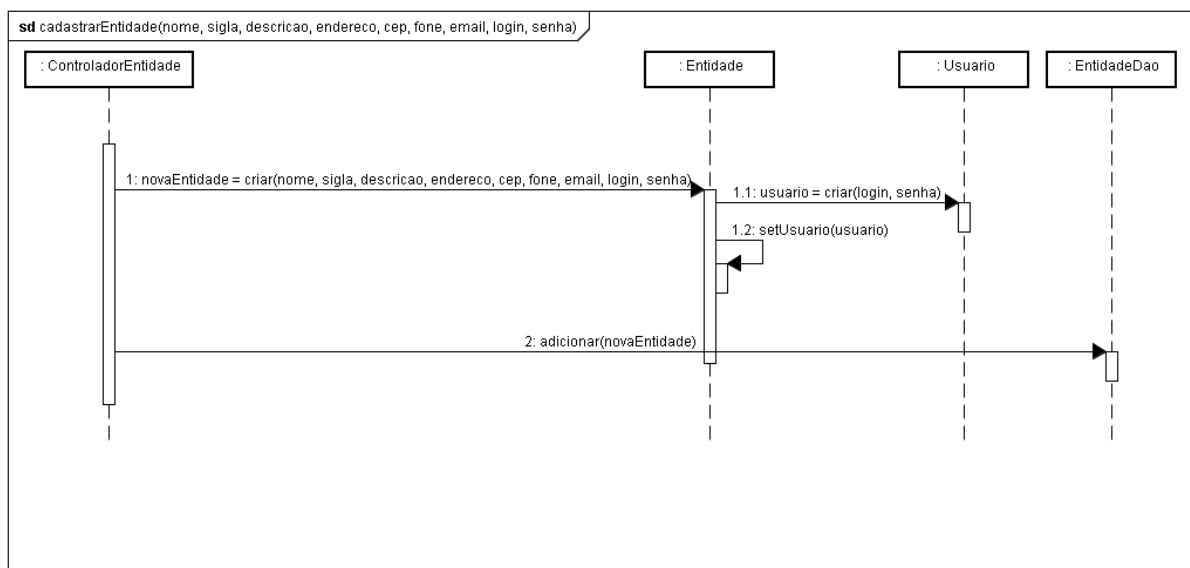


Figura 7: Diagrama de sequência cadastrar entidade

Os projetos e subprojetos são estruturados para fornecerem uma quantidade maior de recursos, pois, as coletas de dados, os avaliadores e as variáveis além de estarem vinculados ao projeto podem também estarem vinculados a um subprojeto específico, e possuir permissões e características próprias, o que proporciona buscas mais específicas e buscas mais globais.

Considera que os avaliadores responsáveis por projetos, tenham o conhecimento necessário e adequado de suas avaliações para configurar o ambiente do sistema, assim, atender aos seus objetivos principais. Espera ao menos que os avaliadores responsáveis, sejam capazes de identificarem os avaliadores necessários e capazes de criarem grupos de variáveis amplos o suficiente para abranger os objetivos de suas avaliações.

Após a estrutura gerada, todos os avaliadores inclusos nos projetos podem acessar o sistema, utilizar as funcionalidades disponíveis que estão de acordo com as suas permissões e realizar as coletas de dados. Para realizar uma coleta de dados, o avaliador deve selecionar o projeto, o conjunto de variáveis e os dados da pessoa a ser avaliada.

Posteriormente à coleta de dados, o sistema disponibiliza somente aos avaliadores responsáveis por projetos, funcionalidades para que possam realizar análises dos dados coletados. É possível selecionar uma pessoa que passou pelo processo de coleta de dados e obter relatórios de acompanhamento de acordo com o tipo específico de variável selecionada. O sistema oferece também a funcionalidade para obter relatórios por variáveis, onde uma variável e grupos de pessoas são selecionados, possibilitando a comparação entre os grupos o que proporciona a real possibilidade da descoberta de padrões, entre outros objetivos.

Para obter uma visão global do projeto, o sistema disponibiliza aos responsáveis pelos projetos, relatórios quantitativos e gráficos que expõem numericamente a atual situação das coletas de dados realizadas, variáveis, avaliadores, avaliados e subprojetos existentes. E para a entidade realizar um acompanhamento global de seus projetos, outros relatórios e gráficos são disponibilizados.

3.2 Configuração do Ambiente de Avaliações

3.2.1 Estrutura das Entidades

A estrutura do Websislapam permite que uma entidade e as pessoas que trabalham com ela, tenham acesso ao sistema e identifiquem claramente a divisão de todos os seus projetos e todas as pessoas que são avaliadas, podendo emitir relatórios para auxiliarem em seu desenvolvimento.

A Figura 8 apresenta a maneira que o sistema mantém os acessos das entidades e avaliadores cadastrados, bem como os vínculos dinâmicos dos avaliadores. Os dados gerados que estão vinculados a uma entidade, são exclusivos não sendo visíveis para outras entidades, permitindo que os dados gerais das entidades sejam visualizados somente pelo administrador do sistema.

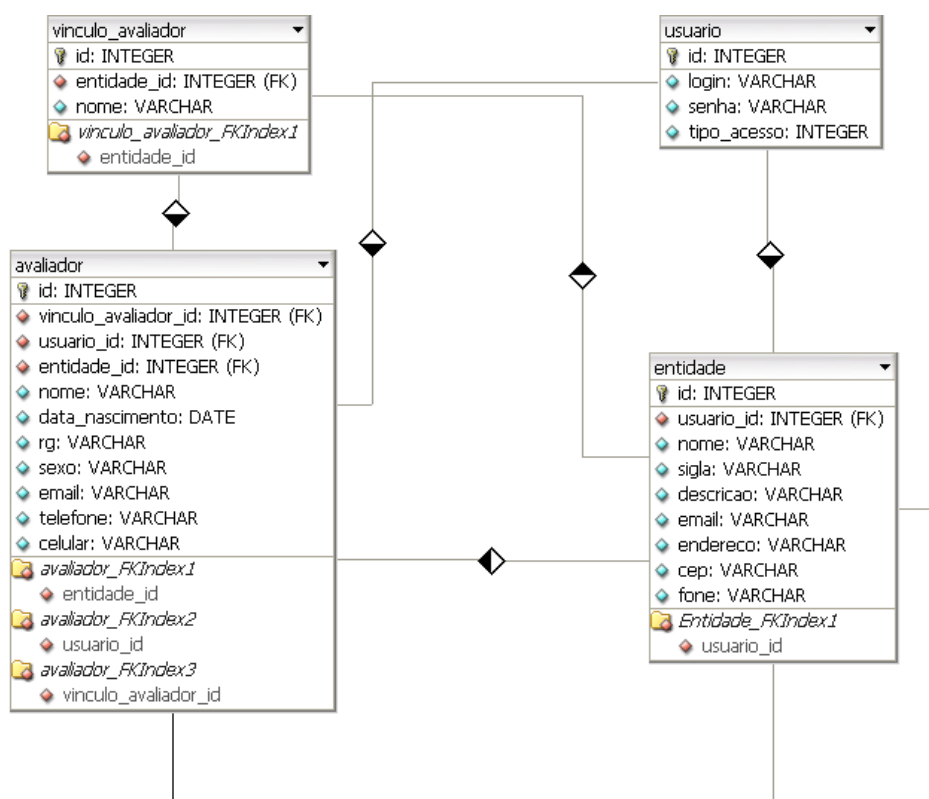


Figura 8: Modelo relacional das entidades e avaliadores que acessam o sistema

A Figura 9 apresenta a estrutura das entidades que podem criar vários projetos e vários subprojetos hierarquicamente. Eles estão relacionados e são dependentes da existência da entidade. É possível vincular avaliadores aos projetos identificando-os como responsáveis ou não. Os avaliadores que são vinculados aos projetos podem ser vinculados aos subprojetos com permissões sobre os dados coletados, esta funcionalidade é garantida pela tabela 'avaliado_subprojeto' e seus atributos incluir, editar e excluir. De maneira parecida com os avaliadores, o sistema permite a vinculação das pessoas que serão avaliadas aos projetos.

Os projetos e subprojetos podem ser identificados pelos seus nomes e siglas, juntamente com todos os avaliadores e avaliados que participam. Podem ser gerados relatórios quantitativos e gráficos comparando resultados entre entidades, projetos ou subprojetos selecionados.

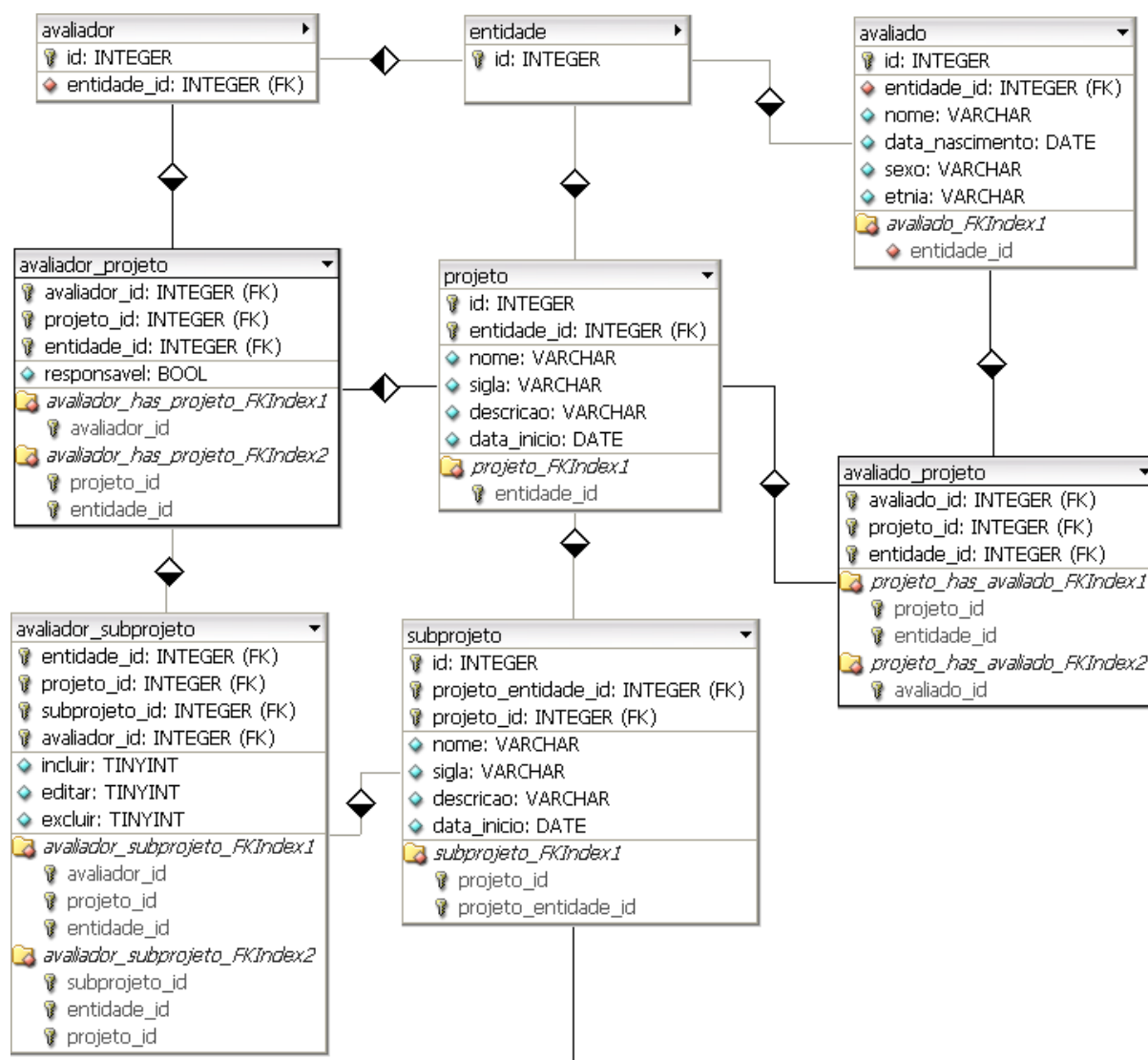


Figura 9: Modelo relacional dos projetos e subprojetos da entidade

3.2.2 Estrutura das Variáveis Dinâmicas

O Websislapam tem a capacidade de gerar perguntas a critérios dos usuários, sendo essas perguntas denominadas no sistema por variáveis. Não há limites para a quantidade de variáveis geradas, nem restrições de contexto.

As variáveis permitidas são quantitativas ou qualitativas. Para variáveis quantitativas, os tipos permitidos de respostas são números inteiros e decimais, uma data ou um horário, e para as variáveis qualitativas, são definidas alternativas.

De acordo com a Figura 10, as variáveis geradas são organizadas em nível de projeto e em nível de subprojeto, o que possibilita especificar subgrupos de variáveis que são coletadas e posteriormente analisadas.

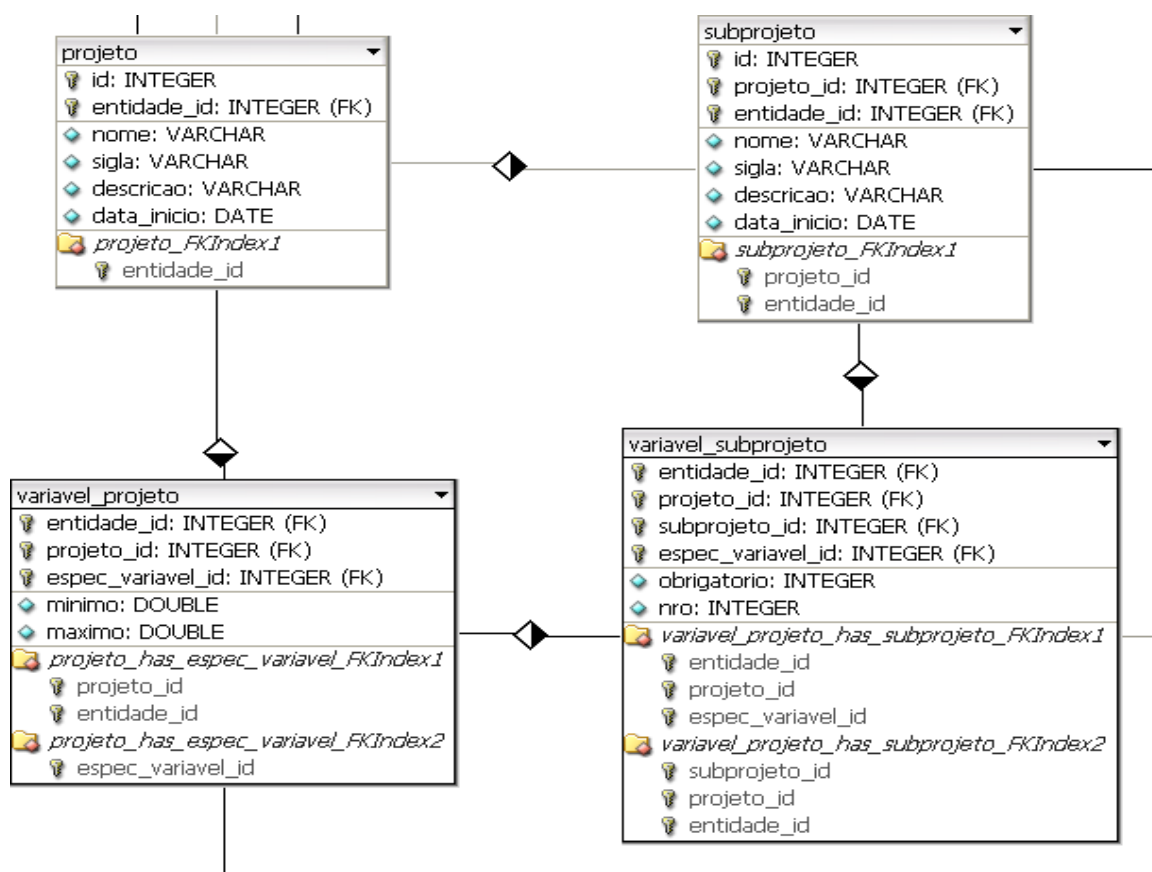


Figura 10: Modelo relacional das variáveis do projeto e subprojeto

De acordo com a Figura 11, o Websislapam trabalha com o reaproveitamento das variáveis e alternativas das variáveis geradas, ou seja, quando um usuário gera uma variável ou alternativa para um projeto, alguns dados

cadastrais podem ser reaproveitados para outros projetos, não havendo a necessidade de um novo cadastramento. As variáveis se encaixam em qualquer contexto de pesquisa devido às tabelas do banco de dados ‘espec_variavel’ e ‘espec_alternativa’ e pelos seus atributos ‘nome’, que permitem ao usuário cadastrar qualquer variável de seu interesse. As variáveis quantitativas possuem seus tipos pré-definidos de respostas devido à tabela do banco de dados ‘mascara’ e as variáveis qualitativas possuem suas alternativas devido à tabela ‘alternativa’. Diferenciam-se variáveis quantitativas das variáveis qualitativas através dos atributos ‘tipo’ nas tabelas ‘espec_variavel’ e ‘mascara’. Os recursos para diminuir a quantidade de erros encontrados nas coletas de dados são definidos através dos atributos ‘mínimo’ e ‘máximo’ da tabela ‘variavel_projeto’, eles identificam o intervalo de resposta permitido para variáveis que possuem respostas dos tipos ‘Inteiro’ e ‘Decimal’. E do atributo ‘obrigatório’ da tabela ‘variavel_subprojeto’ que garante que em uma coleta de dados onde esta variável esteja presente, esta variável não terá resposta vazia. O atributo ‘nro’ das tabelas ‘variavel_subprojeto’ e ‘alternativa’, indicam a ordem em que esta variável está localizada no conjunto de variáveis do subprojeto e a ordem da alternativa no conjunto de alternativas de uma variável, respectivamente.

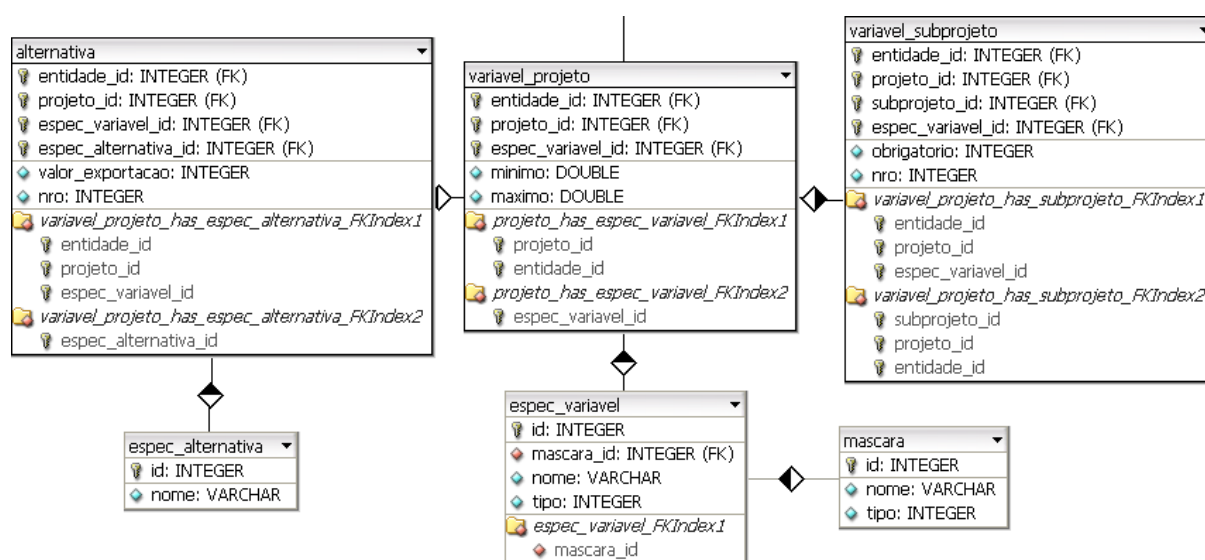


Figura 11: Modelo relacional das variáveis quantitativas e qualitativas

Essa estrutura garante que variáveis quantitativas e qualitativas em diferentes contextos sejam criadas, reaproveitadas e vinculadas a projetos e

subprojetos das entidades. Fornece também alguns recursos para especificar as respostas que serão coletadas, o que tem como objetivo auxiliar os avaliadores para que apliquem corretamente as coletas de dados às pessoas que são avaliadas.

3.3 Coletas de Dados

As coletas de dados iniciam após a configuração do ambiente de avaliação. O avaliador pode escolher entre os conjuntos de variáveis existentes do projeto ou realizar uma coleta reunindo todas as variáveis do projeto.

Durante a coleta de dados, a pessoa avaliada é identificada e os dados das variáveis selecionadas são preenchidos e armazenados. O processo conta com o auxílio de dados históricos recentes do avaliado, e para cada variável, a especificação do tipo de resposta esperada e o intervalo de valores permitidos. Para casos em que o avaliado a ser entrevistado ainda não está cadastrado, é possível cadastrá-lo durante a coleta de dados.

A Figura 12, Figura 13 e Figura 14 apresentam o fluxo de atividades para a realização de uma coleta de dados. Após o preenchimento do formulário de coleta de dados, inicia o fluxo de atividades da Figura 12.

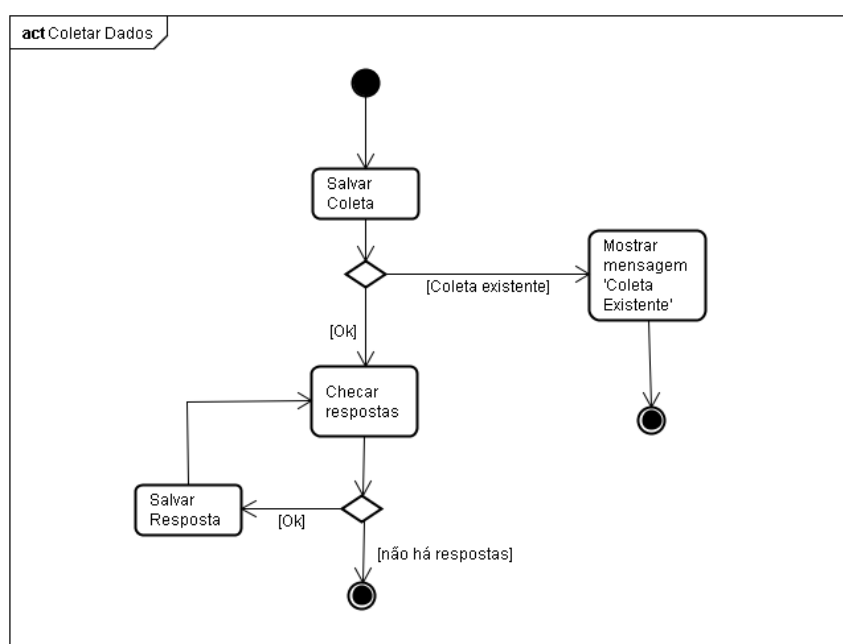


Figura 12: Diagrama de atividade coletar dados

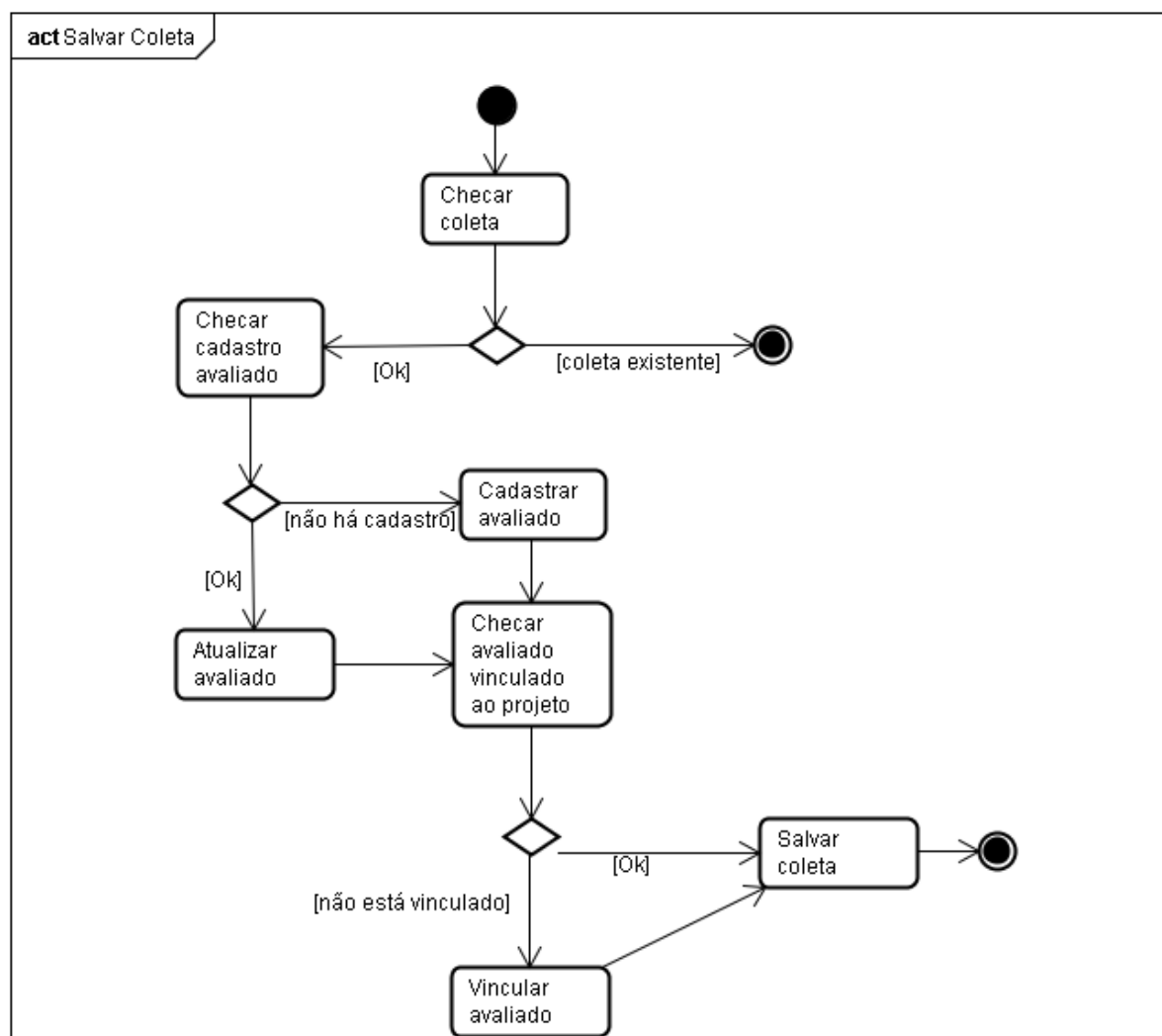


Figura 13: Diagrama de atividade salvar coleta

Para maiores detalhes, as Figuras 13 e 14 apresentam os fluxos de atividades ‘Salvar coleta’ e ‘Salvar resposta’, respectivamente. Garantem que uma coleta de dados seja única, que a pessoa que está sendo avaliada esteja vinculada ao projeto corrente, e que todas as respostas adquiridas sejam armazenadas.

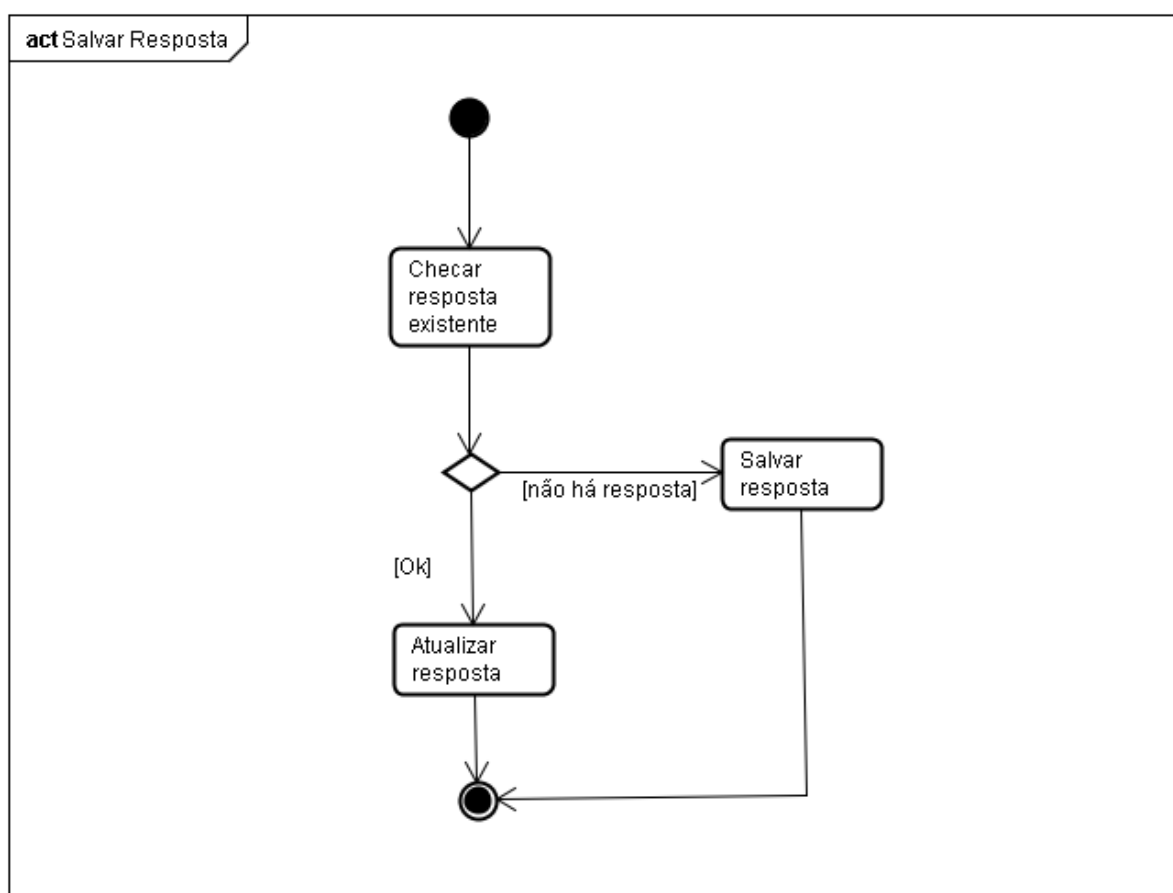


Figura 14: Diagrama de atividade salvar resposta

A Figura 15 apresenta o modelo relacional do banco de dados das coletas de dados.

Em um subprojeto pode haver inúmeras coletas de dados. Uma coleta de dados é identificada pelo seu subprojeto, uma pessoa avaliada e uma data. Mantém também o avaliador que efetuou a coleta para posterior correção de erros encontrados.

As respostas adquiridas durante a coleta de dados estão diretamente vinculadas com a coleta e as variáveis do subprojeto, sendo que o conteúdo é armazenado no atributo 'conteúdo', da tabela 'resposta'. Para variáveis qualitativas o conteúdo armazenado da coleta representa o valor de exportação da alternativa selecionada.

A vantagem de especificar os grupos de variáveis em subprojetos é a melhor organização, para que pessoas avaliadoras possam trabalhar em cada subprojeto separadamente, com suas respectivas permissões de acesso.

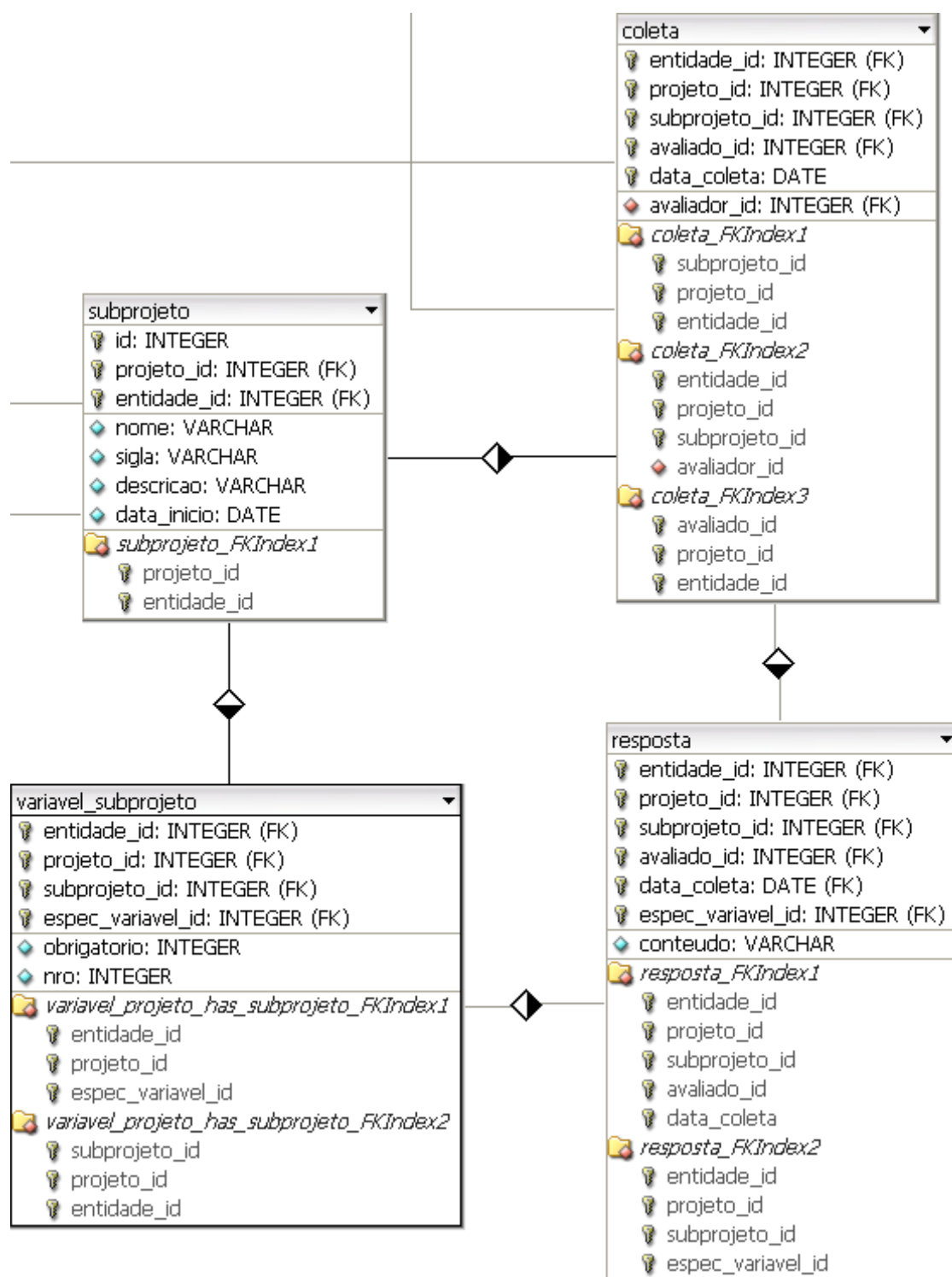


Figura 15: Modelo relacional das coletas de dados

Essa estrutura permite manter e identificar os dados coletados das pessoas avaliadas que estão dentro dos projetos das entidades. Lembrando que os dados gerados desta estrutura são criados a critério do usuário, podendo se encaixar em inúmeras áreas de pesquisas.

3.4 Análise dos Dados Coletados

Sistemas de avaliações se tornam mais completos e eficientes quando fornecem maior quantidade de recursos para a estruturação das avaliações, a realização das coletas de dados e para a análise dos dados coletados.

De acordo com os dados armazenados das avaliações, que são os dados que identificam as pessoas que são avaliadas, os dados das coletas de dados e os dados que identificam cada variável utilizada em coletas de dados, o Websislapam fornece principalmente recursos para realizar um acompanhamento de uma pessoa avaliada e recursos para realizar comparações entre grupos específicos de pessoas avaliadas, onde novos padrões podem ser descobertos ou objetivos podem ser confirmados.

Para realizar o acompanhamento de uma pessoa avaliada, o sistema oferece a opção para visualizar as informações coletadas por meio de gráficos. Ao selecionar uma pessoa para obter relatórios, todas as variáveis do projeto onde há dados coletados referentes a esta pessoa, são listadas.

O acompanhamento da pessoa avaliada é realizado de acordo com os tipos das variáveis existentes. Os relatórios são exibidos para variáveis quantitativas com respostas definidas do tipo inteiro ou decimal, e para variáveis qualitativas.

Para as variáveis quantitativas com respostas do tipo inteiro ou decimal, os dados coletados da pessoa avaliada são visualizados através de gráficos de linha, onde os dados coletados são organizados em uma linha do tempo, o que possibilita uma análise do desenvolvimento deste conteúdo de maneira mais geral.

Para as variáveis qualitativas, os dados coletados são visualizados através de gráficos de setores. O gráfico de setores expõe para o usuário a porcentagem de vezes em que cada alternativa da variável foi selecionada durante as coletas de dados da pessoa avaliada.

Para a comparação entre grupos de avaliados, o usuário tem a liberdade para agrupar as pessoas avaliadas que estão vinculadas a um projeto, baseando-se em intervalos de datas de nascimento, nos seus sexos e em suas etnias. A esses grupos selecionados, podem estar relacionados os relatórios das

variáveis quantitativas, com respostas do tipo inteiro ou decimal e variáveis qualitativas.

Para as variáveis quantitativas com respostas do tipo inteiro ou decimal, as médias aritméticas dos dados coletados referentes aos grupos de avaliados são calculadas e comparadas através de gráficos de barras.

E para as variáveis qualitativas, a quantidade de respostas de cada alternativa e a frequência relativa das respostas dos grupos, são comparadas também por gráficos de barras.

Além do acompanhamento das pessoas avaliadas, o sistema fornece ao administrador e aos usuários a possibilidade de acompanhar com dados numéricos e gráficos o estado atual das entidades, dos projetos e subprojetos existentes.

Ao administrador, há a opção para visualizar a quantidade de entidades cadastradas, projetos, subprojetos, avaliadores, avaliados, variáveis e coletas realizadas em todo o sistema. Também é possível visualizar graficamente a quantidade de avaliados diferenciados pelo sexo em cada entidade, visualizar os avaliadores cadastrados por entidades e a quantidade de coletas de dados realizadas dos avaliados diferenciados também pelo sexo de cada entidade.

Essas funcionalidades também são apresentadas aos usuários entidades e avaliadores responsáveis por projetos, mas visualizam somente os dados que estão abaixo de sua categoria hierárquica na estrutura do sistema. Uma entidade, por exemplo, pode obter um relatório gráfico da quantidade de coletas de dados realizadas em cada um de seus projetos. E um avaliador responsável por um projeto, pode verificar a quantidade de coletas de dados em cada subprojeto pertencente ao projeto.

4 ESTUDO DE CASO

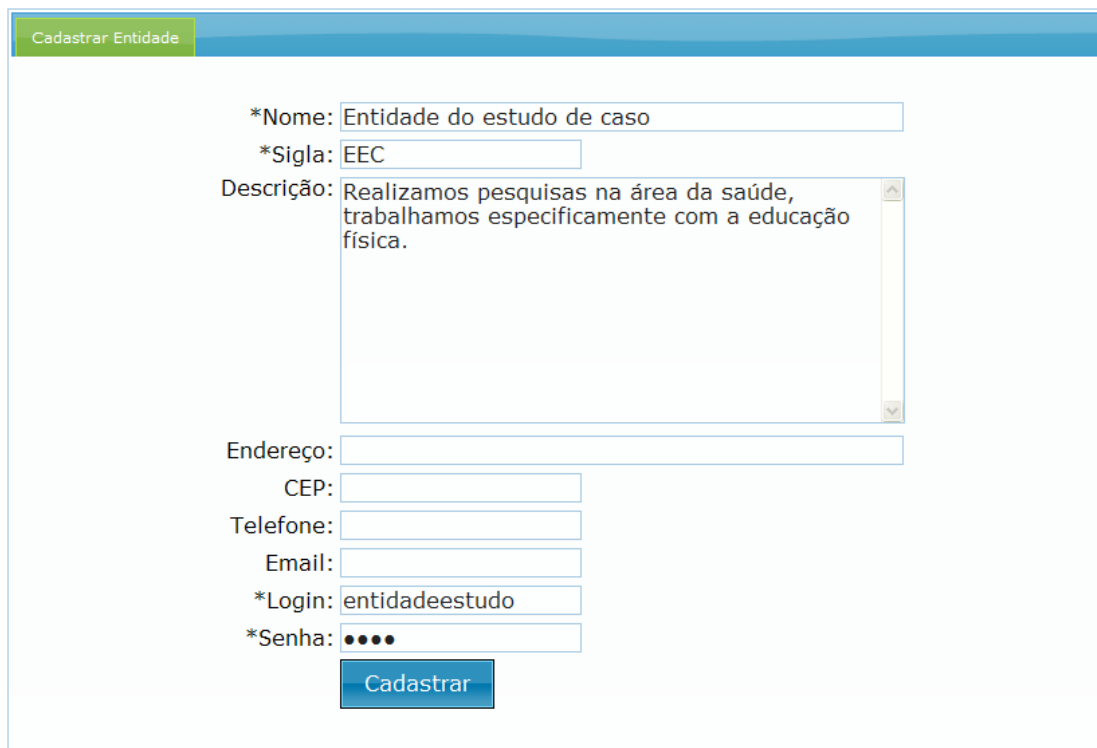
Para o melhor esclarecimento das idéias e funcionalidades do Websislapam, é proposta uma situação com dados fictícios, que realiza um processo de avaliação de pessoas e poderia fazer uso real do sistema.

Considera uma entidade que atua na área da saúde, especificamente com educação física, que realiza avaliações com pessoas de uma cidade. Seus projetos realizados, de maneira geral visam à descoberta de grupos de pessoas que necessitem de mudanças em seus hábitos, além do acompanhamento individual de algumas delas. Essas mudanças têm como objetivo a melhoria da saúde e a qualidade de vida.

Para dar início ao processo de avaliação da entidade, primeiramente, a estrutura da entidade é definida com seus avaliadores e projetos. Em seguida, a estrutura de permissões dos avaliadores dentro dos projetos com as variáveis a serem utilizadas. Logo mais, as coletas de dados e então, os relatórios disponíveis são consultados.

A Figura 16 apresenta o cadastro da entidade. Após a entidade ser cadastrada, ela tem acesso ao sistema podendo cadastrar seus projetos. Para um projeto, deve estar vinculado pelo menos um avaliador responsável, como é apresentado na Figura 17.

Na Figura 18 são apresentadas as opções disponíveis ao usuário avaliador responsável por um projeto. Essas opções são referentes ao projeto criado.



The screenshot shows a web form titled 'Cadastrar Entidade' in a green header bar. The form contains several input fields: '*Nome:' with the value 'Entidade do estudo de caso', '*Sigla:' with 'EEC', 'Descrição:' with a text area containing 'Realizamos pesquisas na área da saúde, trabalhamos especificamente com a educação física.', 'Endereço:', 'CEP:', 'Telefone:', 'Email:', '*Login:' with 'entidadeestudo', and '*Senha:' with four dots. A blue 'Cadastrar' button is at the bottom.

Cadastrar Entidade

*Nome: Entidade do estudo de caso

*Sigla: EEC

Descrição: Realizamos pesquisas na área da saúde, trabalhamos especificamente com a educação física.

Endereço:

CEP:

Telefone:

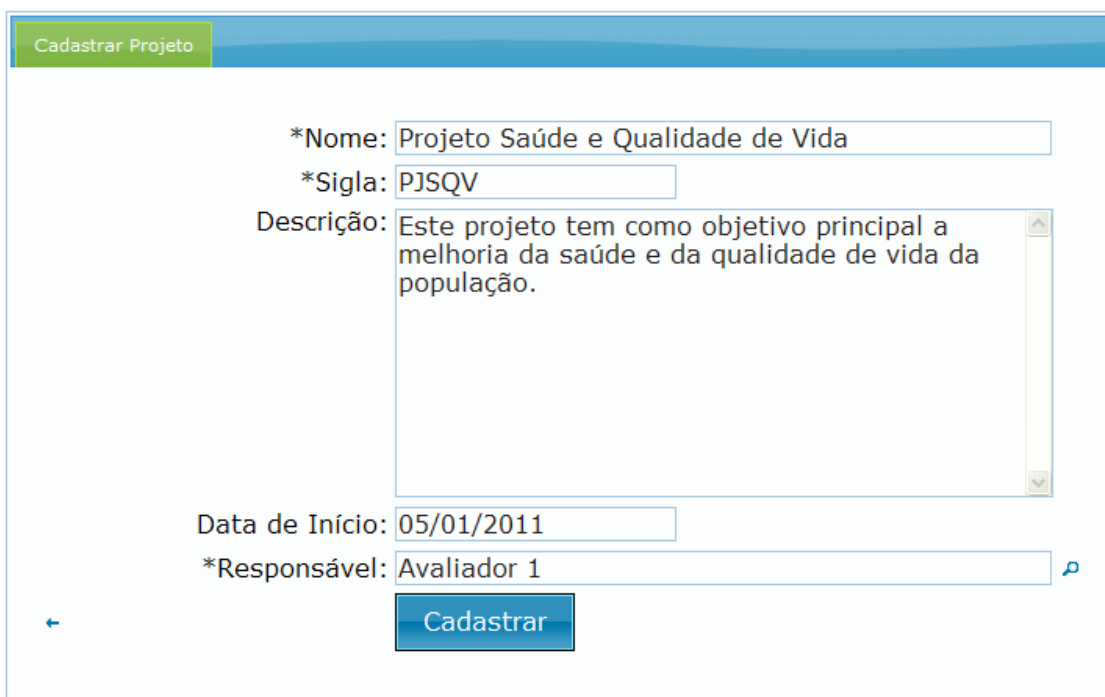
Email:

*Login: entidadeestudo

*Senha: ●●●●

Cadastrar

Figura 16: Cadastrar entidade



The screenshot shows a web form titled 'Cadastrar Projeto' in a green header bar. The form contains several input fields: '*Nome:' with 'Projeto Saúde e Qualidade de Vida', '*Sigla:' with 'PJSQV', 'Descrição:' with a text area containing 'Este projeto tem como objetivo principal a melhoria da saúde e da qualidade de vida da população.', 'Data de Início:' with '05/01/2011', and '*Responsável:' with 'Avaliador 1'. A blue 'Cadastrar' button is at the bottom. There is a back arrow on the left and a search icon on the right of the '*Responsável:' field.

Cadastrar Projeto

*Nome: Projeto Saúde e Qualidade de Vida

*Sigla: PJSQV

Descrição: Este projeto tem como objetivo principal a melhoria da saúde e da qualidade de vida da população.

Data de Início: 05/01/2011

*Responsável: Avaliador 1

Cadastrar

Figura 17: Cadastrar projeto

avaliador1, você está em Projeto :: PJSQV

Nome do Projeto
Projeto Saúde e Qualidade de Vida

Descrição
Este projeto tem como objetivo principal a melhoria da saúde e da qualidade de vida da população.

Sigla	Data de Início	Responsáveis
PJSQV	05/01/2011	1. Avaliador 1

Adicionar

Editar Projeto

Projeto
Coletar
Dados coletados
Subprojetos
Avaliados
Avaliadores
Variáveis
Relatórios

Figura 18: Tela principal do responsável do projeto

Cadastrar Subprojeto

*Nome: Saúde e Evolução

*Sigla: SE

Descrição: Avaliadores que são alunos de pós-graduação.

Data de Início: 05/01/2011

Cadastrar

Figura 19: Cadastrar subprojeto

Na Figura 19 um subprojeto é cadastrado e na Figura 20 são apresentados os avaliadores que estão vinculados ao projeto e suas permissões referentes a cada subprojeto. No exemplo, os avaliadores que são alunos da graduação possuem acesso somente às variáveis do subprojeto 'SA'. Somente os

responsáveis do projeto e os avaliadores do subprojeto 'SA' podem realizar coletas referentes a este subprojeto.

Avaliadores

Gerencie as permissões dos avaliadores do projeto nos subprojetos. Você pode incluir um novo avaliador no projeto, vinculá-lo a um subprojeto com uma determinada permissão (incluir, editar e/ou excluir) sobre os dados.

Adicionar Avaliador

Avaliador				Permissões nos subprojetos	
Opções	ID	Nome	Vínculo	Subprojeto: SA	Subprojeto: SE
	17	Avaliador2	Aluno da Graduação	☒ ☒ ☒	☐ ☐ ☐
	18	Avaliador3	Aluno da Graduação	☒ ☒ ☒	☐ ☐ ☐
	19	Avaliador4	Aluno de Pós-Graduação	☐ ☐ ☐	☒ ☒ ☒
	20	Avaliador5	Aluno de Pós-Graduação	☐ ☐ ☐	☒ ☒ ☒

Salvar Alterações
Permissão para incluir

Figura 20: Avaliadores do projeto e suas permissões nos subprojetos

Adicionar Variável ao Projeto

***Variável**

***Mascara** Decimal v

***Mínimo** Inteiro ou Decimal

***Máximo** Inteiro ou Decimal

Cadastrar

Figura 21: Cadastrar variável quantitativa

A Figura 21 apresenta uma variável quantitativa sendo criada no projeto, para posteriormente ser vinculada a um subprojeto. No campo 'Mascara' é definido o tipo de resposta esperada na coleta de dados. Esses tipos podem ser 'Inteiro', 'Decimal', 'Horas' e 'Data'.

A Figura 22 apresenta um exemplo de variável qualitativa. A variável qualitativa 'Fuma?' e suas alternativas 'A - Sim' e 'B - Não' estão sendo cadastradas.

Adicionar Variável Qualitativa ao Projeto

1 - Variável 2 - Adicionar opção de resposta

Variável:
Fuma?

Opções de resposta:
Sim
Valor de exportação: 1
Excluir

Não
Valor de exportação: 0
Excluir

← Cadastrar

Figura 22: Cadastrar variável qualitativa

Variáveis do Projeto

Gerencie as variáveis do projeto:

- Adicione ou exclua variáveis quantitativas e qualitativas ao projeto;
- Vincule estas variáveis aos subprojetos;
- Selecione uma variável e compare grupos de avaliados

Variável Quantitativa Variável Qualitativa

Variável				Vincular aos subprojetos	
Opções	Variável	Máscara	Intervalo	Subprojeto: SA	Subprojeto: SE
 	% de Gordura Corporal	Decimal	0 - 100	☒ ☐	☐ ☐
 	Está satisfeito com sua condição física?	Qualitativo	.	☒ ☐	☐ ☐
 	Fuma?	Qualitativo	.	☐ ☐	☒ ☒
 	Peso em Kg	Decimal	0 - 500	☒ ☐	☐ ☐

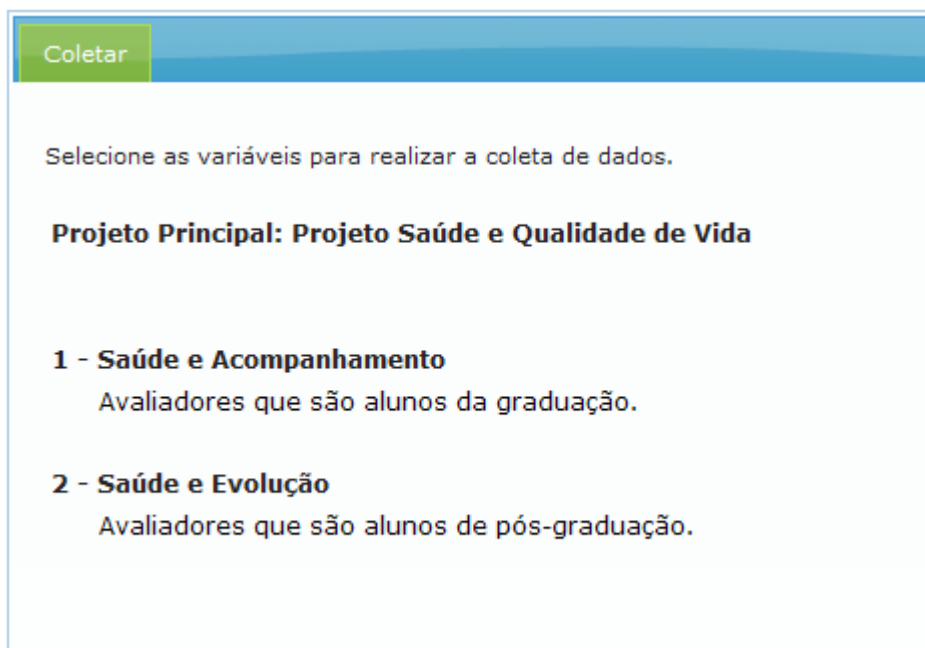
Salvar Alterações

Vincular variável ao subprojeto

Figura 23: Variáveis do projeto

Na Figura 23 estão listadas todas as variáveis vinculadas ao projeto e a cada subprojeto. Essas variáveis podem ser vinculadas especificamente a cada subprojeto existente com a opção da sua resposta ser obrigatória, ou não, nas

coletas realizadas. As opções para incluir e excluir as variáveis também estão presentes, além da opção para visualizar os relatórios referentes a cada variável.



Coletar

Selecione as variáveis para realizar a coleta de dados.

Projeto Principal: Projeto Saúde e Qualidade de Vida

1 - Saúde e Acompanhamento
Avaliadores que são alunos da graduação.

2 - Saúde e Evolução
Avaliadores que são alunos de pós-graduação.

Figura 24: Selecionar grupo de variáveis para coletar

Neste momento a estrutura básica da entidade foi criada, ou seja, o projeto 'Projeto Saúde e Qualidade de Vida' foi cadastrado juntamente com seus subprojetos 'Saúde e Acompanhamento' e 'Saúde e Evolução'. Dois avaliadores possuem permissão para realizar coletas de dados em cada subprojeto. E as variáveis foram cadastradas e vinculadas aos subprojetos.

A Figura 24 apresenta o momento em que os dados serão coletados. Cada opção representa um grupo de variáveis definido anteriormente, ou, cada grupo corresponde a um subprojeto. Há também a opção para realizar coletas referentes ao projeto, onde os responsáveis possuem acesso sobre todas as variáveis e os avaliadores possuem acesso somente aos subprojetos que estão vinculados com a opção 'incluir'.

A Figura 25 apresenta uma coleta de dados sendo realizada. Para cada variável há informações de ajuda referentes à maneira que a resposta deve ser inserida. Para a variável '1', o valor mínimo permitido é '0' e o máximo é '100', sendo que esta resposta deve ser do tipo 'Decimal'. Na opção 'últimas coletas', é listado os dois últimos dados coletados da variável '1' referentes ao avaliado selecionado 'João da Silva'.

Para selecionar um avaliado o sistema oferece o recurso auto-completar, onde um avaliado existente pode ter seus dados cadastrais selecionados. Um novo avaliado também pode ser cadastrado.

Nova Coleta

Avaliador Avaliador 1

Subprojeto: Saúde e Acompanhamento

- * Data da Coleta
06/10/2011 *Data*
- * Nome do Avaliado
João da Silva *Texto*
- * Data de Nascimento
09/01/1989 *Data*
- Sexo
Masculino
- Etnia
Branco

1. % de Gordura Corporal
0 18 *100 Decimal [Últimas coletas](#)*

2. Peso em Kg
0 70 *500 Decimal [Últimas coletas](#)*

3. Está satisfeito com sua condição física?

☒ a. Sim
☐ b. Não

Consulte as últimas coletas do avaliado para esta variável

Figura 25: Coleta de dados

Após as coletas realizadas, os dados podem ser analisados como se observa nas imagens a seguir.

A Figura 26 apresenta o acompanhamento dos dados coletados referentes a um avaliado, sendo estes dados organizados de acordo com as datas das coletas.

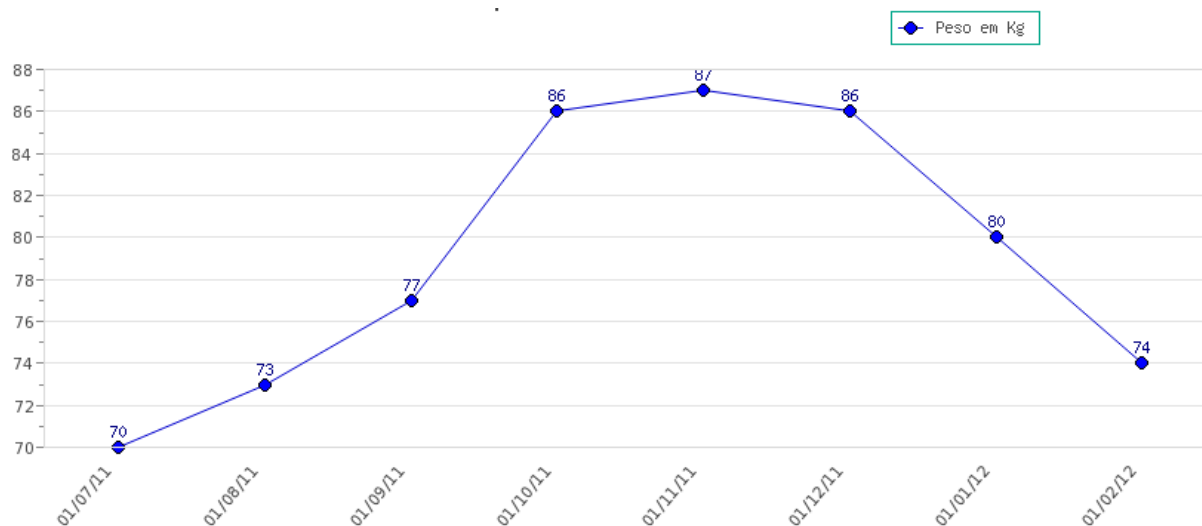


Figura 26: Acompanhamento do avaliado para variável quantitativa

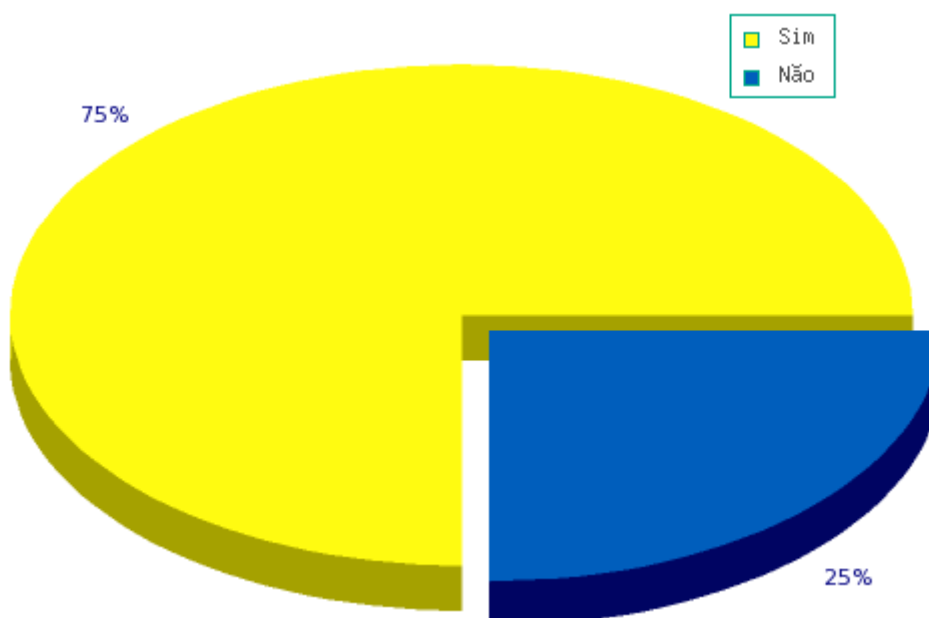


Figura 27: Acompanhamento do avaliado para variável qualitativa

A Figura 27 apresenta o percentual de respostas coletas para a variável 'Está satisfeito com sua condição física?', onde o avaliado esteve mais satisfeito do que insatisfeito, pois, 75% das respostas coletadas foram referentes à alternativa 'Sim' e 25% referentes à alternativa 'Não'.

A Figura 28 apresenta os grupos de pessoas avaliadas selecionados pelo usuário, e em seguida a Figura 29 apresenta um gráfico de barras referente a esses grupos selecionados e para a variável 'Fuma?'. Conclui que entre as pessoas

nascidas na década de 1970, não há fumantes e entre as pessoas com etnia 'Negro', há mais fumantes que não fumantes.

Grupos de avaliados selecionados				
Grupo	Data de nascimento inicial	Data de nascimento final	Sexo	Etnia
1	Geral	Geral	Geral	Branco
2	Geral	Geral	Geral	Negro
3	01/01/1970	31/12/1979	Geral	Geral
4	01/01/1980	31/12/1989	Geral	Geral
5	01/01/1990	31/12/1999	Geral	Geral

Figura 28: Grupos de avaliados selecionados para gráfico de variável qualitativa

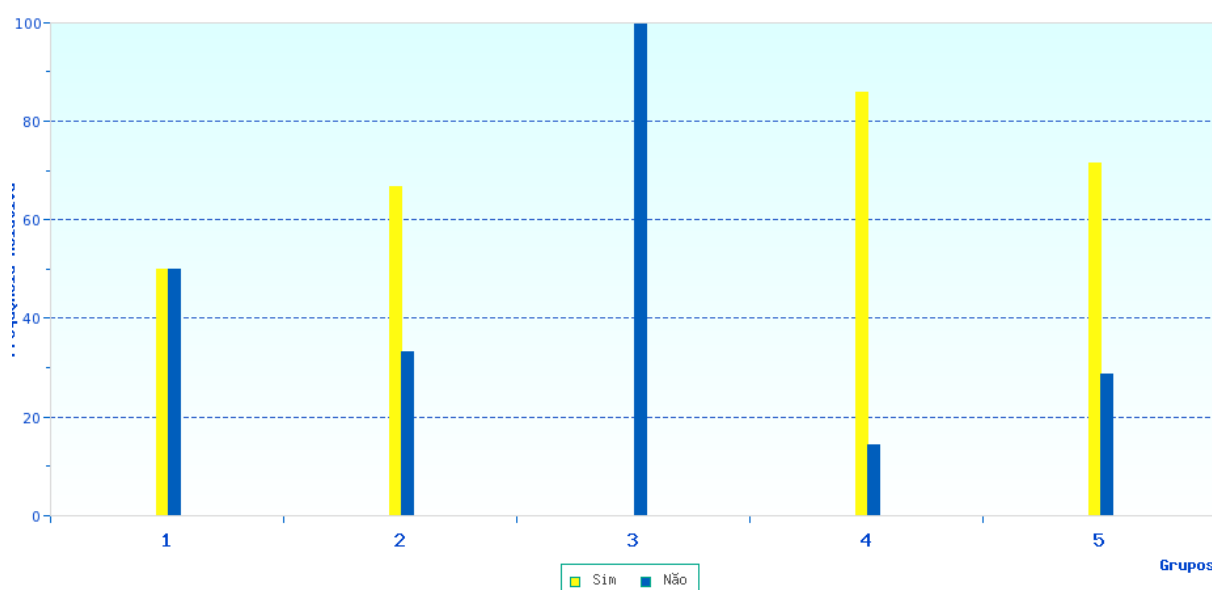


Figura 29: Grupos de avaliados para variável qualitativa

A Figura 30 apresenta os grupos de pessoas avaliadas que são selecionados para análise referente a uma variável quantitativa.

De acordo as médias aritméticas do o gráfico da Figura 31, conclui que o grupo de pessoas com etnia 'Negro' do sexo feminino está com o maior índice de gordura corporal entre os grupos selecionados.

Grupos de avaliados selecionados				
Grupo	Data de nascimento inicial	Data de nascimento final	Sexo	Etnia
1	Geral	Geral	M	Branco
2	Geral	Geral	F	Branco
3	Geral	Geral	M	Negro
4	Geral	Geral	F	Negro

Figura 30: Grupos de avaliados selecionados para gráfico de variável quantitativa

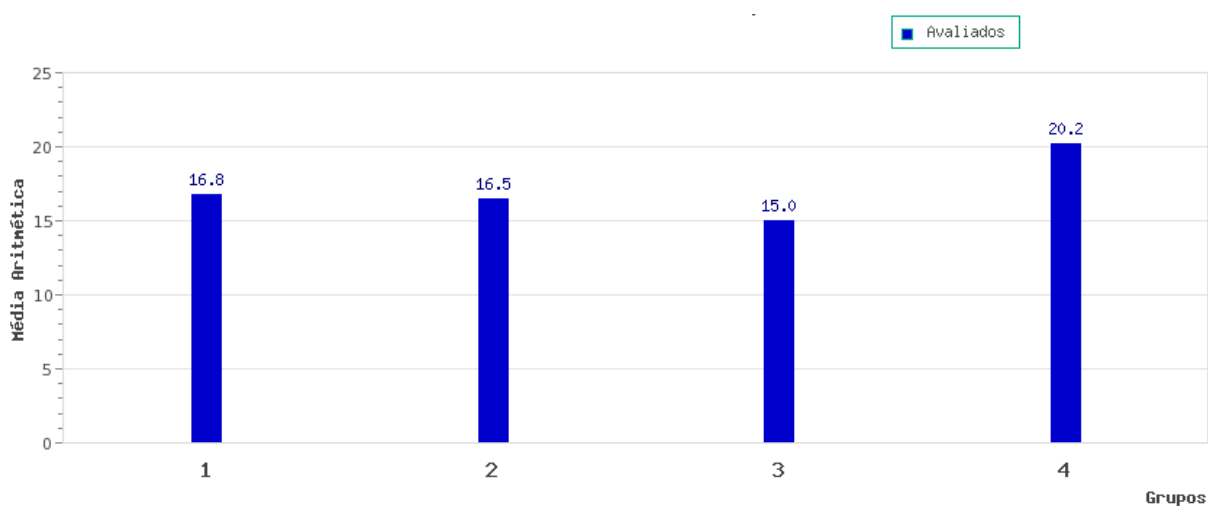


Figura 31: Grupos de avaliados para variável quantitativa

O sistema oferece recursos para o acompanhamento das coletas realizadas nos projetos, como apresenta a Figura 32, assim como dados gerais referentes ao projeto da entidade, como destacado na Figura 33.

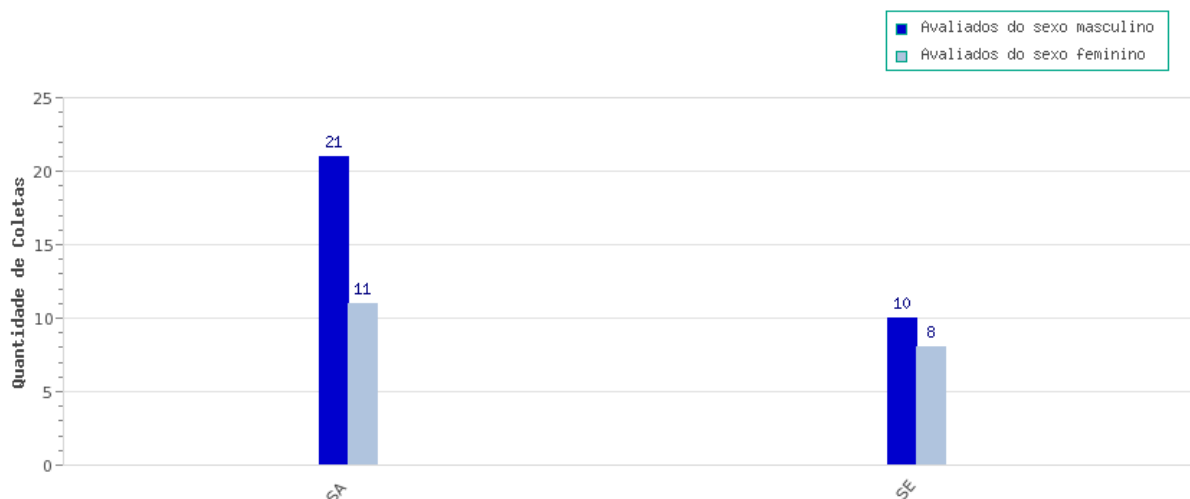


Figura 32: Quantidade de coletas por subprojetos

Dados Gerais - Projeto Saúde e Qualidade de Vida	
	Quantidade
Responsáveis	1
Avaliadores	4
Avaliados	21
Subprojetos	2
Variáveis	4
Coletas	50

Figura 33: Dados gerais do projeto

Além dos relatórios referentes aos dados de um projeto específico, o Websislapam fornece relatórios referentes ao sistema em geral, ou seja, relatórios que acompanham os projetos, as pessoas e as coletas realizadas de todas as entidades cadastradas.

	A	B	C	D	E	F	G
1	ID do avaliado	Nome do avaliado	Data de nascimento	Sexo	Etnia	Data da coleta	Fuma?
2	44	Pessoa1	3/7/1990	M	Branco	1/7/2011	0
3	45	Pessoa2	26/3/1974	M	Negro	1/7/2011	0
4	46	Pessoa3	2/7/1971	M	Branco	1/7/2011	0
5	47	Pessoa4	27/7/1974	F	Branco	1/7/2011	0
6	48	Pessoa5	10/4/1979	F	Oriental	1/7/2011	0
7	49	Pessoa6	16/7/1984	M	Negro	1/7/2011	0
8	50	Pessoa7	23/7/1983	M	Negro	1/7/2011	1
9	51	Pessoa8	12/8/1986	M	Negro	1/7/2011	1
10	52	Pessoa9	3/6/1987	F	Negro	1/7/2011	1
11	53	Pessoa10	10/5/1989	M	Oriental	1/7/2011	1
12	54	Pessoa11	29/9/1984	F	Negro	1/7/2011	1
13	55	Pessoa12	10/7/1988	M	Branco	1/7/2011	1
14	56	Pessoa13	28/7/1990	M	Branco	1/7/2011	1
15	57	peessoa14	19/4/1992	M	Branco	1/7/2011	1
16	58	Pessoa15	18/7/1991	F	Oriental	1/7/2011	1
17	59	Pessoa16	16/7/1991	F	Negro	1/7/2011	1
18	60	Pessoa17	1/4/1992	F	Negro	1/7/2011	1
19	61	Pessoa18	31/7/1992	F	Negro	1/7/2011	0

Figura 34: Dados coletados exportados

Para casos em que as funcionalidades para análise dos dados coletados não são suficientes, o sistema disponibiliza a opção de exportação. A Figura 34 apresenta a exportação das coletas realizadas do subprojeto 'Saúde e Evolução', que possui somente a variável 'Fuma?'. Os dados coletados e exportados são os valores referentes às alternativas da variável, ou seja, valor '1' refere-se à alternativa 'Sim' e valor '0' refere-se à alternativa 'Não'. As coletas dos avaliados exportadas, podem ser filtradas com os mesmos critérios dos grupos de avaliados selecionados para análise.

5 CONCLUSÃO

A *Web*, as tecnologias e processos de desenvolvimento de sistemas e aplicações baseados na *Web*, possibilitam que muitas pessoas em diferentes lugares, tenham à sua disposição facilidades e estejam trabalhando para a resolução de seus problemas e alcance de seus objetivos, com o auxílio de ferramentas apropriadas.

Um processo de avaliação de pessoas consiste na coleta e análise de dados que objetiva a descoberta do conhecimento, para a tomada de decisões.

O Websislapam, desenvolvido com tecnologias *Web*, fornece entre outras funcionalidades, as principais:

- Criar questionários amplos o suficiente para atingir os objetivos de uma avaliação proposta por uma entidade cadastrada, independentemente da área de interesse de pesquisa;
- Realizar o processo de coleta de dados com interatividade com o usuário;
- Realizar análise dos dados coletados referentes às pessoas que passaram pelo processo de coleta de dados, por meio de gráficos comparativos;
- Manter uma base de dados com as pessoas avaliadas e os dados coletados;
- Estruturar e organizar as entidades e as pessoas que trabalham para o sucesso das avaliações, com diferentes níveis de acessos;
- Importar e exportar os dados cadastrais referentes aos avaliados e exportar os dados coletados das avaliações.

Considerando o conceito e as características de um processo de avaliação, a interatividade dos sistemas e aplicações baseados na *Web* com o usuário, e as funcionalidades disponibilizadas pelo Websislapam, ele possibilita a qualquer entidade que tenha interesse na descoberta de conhecimento e que desejam tomar as melhores decisões, realizar avaliações de pessoas com eficiência e consistência.

Para a continuação do Websislapam, novas funcionalidades podem ser adicionadas, o que aumentará a integração com outros sistemas, a facilidade de

acesso, a interatividade com o usuário e a capacidade de análise dos dados coletados. Como exemplos de novas funcionalidades destacam as seguintes:

- Na criação do conjunto de perguntas, podem ser adicionados recursos para dividir o conjunto de perguntas em seções, e vincular a resposta de uma variável com a garantia de resposta de outra;
- Recursos mais próximos da *Web 2.0* podem ser incrementados, como por exemplo, a comunicação entre os avaliadores de uma entidade e comunicação da entidade com seus avaliadores;
- Novas opções para a filtragem dos dados durante a análise podem ser adicionadas, como por exemplo, dentre os dados coletados referentes a um avaliado, analisar somente os dados de um período selecionado. Adicionar novos gráficos de desempenho relacionados aos avaliadores;
- Adicionar a opção para importar dados referentes a coletas de dados, o que traria a vantagem de utilizar os recursos existentes para análise de dados;
- Adicionar a opção para exportar os dados coletados em nível de projeto, o que possibilitaria mais uma forma de comparação dos dados coletados;
- Adaptar a interface do sistema para o acesso por meio de dispositivos móveis, trazendo deste modo, toda a facilidade de acesso para o Websislapam.

REFERÊNCIAS BIBLIOGRÁFICAS

CONVERSE, TIM; PARK, Joyce; **PHP A Bíblia**; trad. Edson Furmankiewicz – 2.ed – Rio de Janeiro: Campus:2003.

DATE, C. J; **Introdução a Sistemas de Banco de Dados**. trad. Vanderberg Dantas de Souza - Rio de Janeiro: Campus, 2000.

DEITEL, H. M; DEITEL, P.J; e NIETO T. R; **Internet & World Wide Web Como Programar**; trad. Edson Furmankiewicz. – 2ed. – Porto Alegre: Bookman, 2003.

DEITEL, Harvey M; **XML Como Programar**. Porto Alegre: Bookman, 2001.

FLANAGAN, David, **Javascript - O Guia Definitivo** – 4ed. Porto Alegre: Bookman, 2004.

JPGGRAPH. **Site oficial da biblioteca para gráficos em PHP**. Disponível em: <<http://jpgraph.net>>. Acesso em 16 junho 2011.

JQUERY. **Site oficial da biblioteca Javascript**. Disponível em: <<http://jquery.org>> Acesso em 16 junho 2011.

JQUERYUI. **Site oficial da biblioteca de recursos de interface**. Disponível em <<http://jqueryui.com>>. Acesso em 16 junho 2011.

MILANI, André; **MySQL: Guia do Programador**. São Paulo: Novatec, 2006.

PHP Group. PHP: hypertext preprocessor. [S.l.], 2004. Disponível em: <<http://www.php.net>>. Acesso em: 20 junho 2011.

PRESSMAN, Roger S; **Engenharia de Software**; trad. Rosângela Ap. D. Penteado – 6ed – Porto Alegre: AMGH, 2010.

RODRIGO SANCHES, André. **Fundamentos de Armazenamento e Manipulação de Dados, 2005**. Disponível em: < <http://www.ime.usp.br/~andrers/aulas/bd2005-1/aula5.html>> Acesso em 10 de junho de 2011.

SALVIA, John; YSSELDYKE, James E; **Avaliação em educação especial e corretiva**; trad. Doris Sanches Pinheiro – 4ed. – São Paulo: Manole, 1991.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN S; **Sistema de banco de dados**. tradução Daniel Vieira - Rio de Janeiro : Elsevier : Campus, 2006.

TITTEL, Ed; **XML Coleção Schaum**. Porto Alegre: Bookman, 2002.

WEB 2.0. O'REILLY Tim. **Wikipédia A enciclopédia livre**. Disponível em <http://pt.wikipedia.org/wiki/Web_2.0> Acesso em 17 junho 2011.