



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis - IMESA

GUILHERME VINICIUS MUNHOZ

**COMPUTAÇÃO EM NUVEM, UTILIZANDO COMO ESTUDO DE CASO
UM SISTEMA PARA CLUBE DE LAZER.**

2012
Assis-SP

GUILHERME VINICIUS MUNHOZ

**COMPUTAÇÃO EM NUVEM, UTILIZANDO COMO ESTUDO DE CASO
UM SISTEMA PARA CLUBE DE LAZER.**

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, Como requisito do curso de Graduação de Ciência da Computação.

Orientadora: Esp. Prof^a. Diomara Martins Reigato Barros

Área de Concentração: Informática

Assis
2012

FICHA CATALOGRÁFICA

MUNHOZ, Guilherme Vinicius

Computação nas Nuvens , utilizando como estudo de caso um Sistema de Clube de Lazer/ Guilherme Vinicius Munhoz. Fundação Educacional do Município de Assis – FEMA – Assis, 2011.

70 p.

Orientadora: Esp. Prof^a. Diomara Martins Reigato Barros

Trabalho de Conclusão de Curso – Instituto Municipal de Ensino Superior de Assis – IMESA

1. Computação em Nuvens 2. Aplicativo comercial 3. Google App Engine

CDD: 001.6
Biblioteca da FEMA

COMPUTAÇÃO EM NUVEM, UTILIZANDO COMO ESTUDO DE CASO UM SISTEMA PARA CLUBE DE LAZER.

GUILHERME VINICIUS MUNHOZ

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, Como requisito do curso de Graduação, analisado pela seguinte comissão examinadora:

Orientadora: Esp. Diomara Martins Reigato Barros

Analizador: Dr. Almir Rogério Camolesi

Assis

2012

DEDICATÓRIA

Dedico esse trabalho a minha família, em especial aos meus pais biológicos Leonor Souza e Floriano Munhoz, e aos meus pais de coração Aparecida Gonçalves e Vitor Gonçalves.

AGRADECIMENTOS

Agradeço a Deus pela força de vontade e inspiração que sempre me deu, e pela fé por nunca desistir!

Aos professores que sempre nos acompanharam com muita dedicação! Em especial professor Almir e professora Diomara, que tive a honra de tê-los como orientadores em períodos diferentes.

A família que sempre me apoiou. Minha tia Aparecida Gonçalves, minha mãe Leonor Souza, ao meu super irmão Rodrigo Gonçalves e minha cunhada Luciana Guskuma! Que com muito carinho e amor me incentivaram em todo momento.

Aos amigos Maisa Santos, Karina Duarte, que sempre escutaram meus lamentos e nunca me deixaram desistir! Ao amigo Paulo Henrique Oliveira, que mesmo com seus projetos diferentes, esteve ao meu lado em todo momento!

RESUMO

A evolução tecnológica tem avançado e crescido desenfreadamente com o passar do tempo. Conhecido como “Cloud computing” (Computação em Nuvens), com uso de servidores físicos ou virtuais. Esta tecnologia consiste na utilização de memórias de armazenamento de computadores interligados pela internet. A ideia de armazenamento de dados por meio de interligação de computadores através de uma rede se teve entre as décadas de 60 e 70, mas foi logo abandonada por se tratar de tecnologias muito avançadas para a época. Porém o resurgimento deste conceito tem crescido e ganhado a popularidade.

A cada dia novas tecnologias aparecem ganhando espaço e concorrência para a abordagem do tema. A tecnologia Amazon foi a primeira a lançar uma plataforma de computação em Nuvens, conhecida como EC² (Elastic Cloud Computing). A plataforma de serviços Azure da Microsoft é um grupo de tecnologia em nuvem que pode ser usado tanto para aplicativos em nuvem quanto para aplicativos locais. O Google Application Engine ou Google GAE é um serviço que diferente do conceito original, onde todo o ambiente responsável pelo ciclo de desenvolvimento e publicação do software é local. Por meio de um ambiente desktop completo através de plugins para o Eclipse IDE chamado Googlipse para criar as aplicações. O GWT pode ser utilizado para qualquer aplicativo web sendo escrita unicamente em Java.

O objetivo deste trabalho foi fazer uma pesquisa sobre as diferentes modalidades dentro do termo computação nas nuvens. E aprofundar no tema desenvolvendo um pequeno sistema para clube de lazer com a plataforma Google Application Engine.

ABSTRACT

Technological evolution has advanced and grown wildly over time. "Cloud computing" is known for using physical or virtual servers. This technology consists in the use of memory storage of computers interconnected by the Internet. The idea of storing data through the interconnection of computers over a network started between the 60 and 70 decades, but it was soon abandoned for need of using very advanced technology for the time. However the resurgence of this concept has grown and gained popularity.

Every day new technologies and competition are developed to approach the subject. Amazon Technology was the first to launch a Cloud Computing platform, known as EC² (Elastic Cloud Computing). The Azure Services Platform is a Microsoft cloud technology group that can be used both for cloud and for local applications.

Google Application Engine or Google GAE is a service different from the original concept because the entire environment responsible for the development and promotion cycle of the software is local; using a desktop environment via Eclipse IDE plugins called Googlipse to create applications. GWT can be used for any web application because it is written in Java itself.

The objective of this study is to research on different modalities within the term Cloud Computing and go further developing a small system for the Leisure Centre with Google Application Engine platform.

LISTA DE ILUSTRAÇÕES

Figura 1 – Nogueira, 2010 – Cenário Atual de Computação em Nuvem.....	16
Figura 2 - Desafio de Segurança da nuvem - O Papel da Segurança na Nuvem Confiável	19
Figura 3 - Tabela Periódica de API para produtos Google (Junior 2011, pg. 28).....	23
Figura 4 - O Custo dos Recursos da Computação do App Engine da Google (Junior 2011, pg 28).....	24
Figura 5 - Console de Administração do APP Engine da Google (Junior, 2011 pg.29).	25
Figura 6 - Logo da Linguagem de Programação Eclipse.....	27
Figura 7 - Folha de Cadastro para novos Associados Paraguaçu Tênis Clube.....	32
Figura 8 - Diagrama de Caso de uso	35
Figura 9 - Diagrama de Classes Clube de Lazer	36
Figura 10 - Diagrama de Caso de Uso Efetuar Login.....	37
Figura 11 – Diagrama de Caso de Uso Manter Título	38
Figura 12 – Caso de Uso Manter Titular	39
Figura 13 - Caso de Uso Manter Tipo de dependente	40
Figura 14 - Manter Dependente	41
Figura 15 - Manter Tipo de Usuário	42
Figura 16 - Manter Usuário	43
Figura 17 - Manter Mensalidade	44
Figura 18 - Manter Lançamento	45
Figura 19 - Caso de Uso Consultar/Emitir Relatório	46
Figura 20 – Work Break Struture.....	46
Figura 21 – Modelagem Proposta.....	49
Figura 22 – Ambiente de programação Eclipse Helio 3.6 com ícones do GWT e App Engine.....	50
Figura 23 – Estrutura do Projeto	51
Figura 24 – Development Mode	52
Figura 25 – Development Mode: Compilação e Servidor	53
Figura 26 - Interfaces de Usuários Google código.....	54
Figura 27 - Página principal do aplicativo	54

Figura 28 - Tela de Login da Google	55
Figura 29 - Saudação ao usuário e desbloqueio do sistema	55
Figura 30 - Persistência dos dados	56
Figura 31 - Variáveis classe Clube	57
Figura 32 - Classe PMF	58
Figura 33 - Classe CadastrarClubeServlet	59
Figura 34 - Web.xml	60
Figura 35 - Formulário de cadastro de clube	61
Figura 36 - Tela de Cadastro de Clube	61
Figura 37 - Arquivo jdoconfig.xml	63
Figura 38 - Pesquisa de clube presente no arquivo pesquiseseuclube.jsp	64
Figura 39 - Tela de Pesquisa de Clubes	65

SUMÁRIO

1. INTRODUÇÃO.....	13
1.1 OBJETIVO.....	14
1.2 MOTIVAÇÃO	14
2. FUNDAMENTAÇÃO TEÓRICA.....	15
2.1. COMPUTAÇÃO EM NUVENS	15
2.2. COMPUTAÇÃO NAS NUVENS VANTAGENS E DESVANTAGENS.....	17
2.3. APLICATIVOS PARA DESENVOLVER COMPUTAÇÃO EM NUVENS	20
2.3.1. APLICATIVO AMAZON	21
2.3.2. APLICATIVO AZURE.....	21
2.4. GOOGLE APPLICATION ENGINE	22
2.5. SANDBOX	25
2.6. DHTML.....	26
2.7. TECNOLOGIA JAVA/ECLIPSE.....	26
3. ESTUDO DE CASO.....	28
3.1. VISÃO GERAL DE UM SISTEMA PARA CLUBE	28
3.2. REQUISITOS DO SISTEMA	28
3.3. PESQUISA DA EMPRESA	29
3.4. FUNÇÕES PRINCIPAIS	30
3.5. REQUISITOS DE SAIDA	30
3.6. ESTATUTOS PESQUISADOS.....	30
3.6.1 PARAGUAÇU TÊNIS CLUBE	31
3.6.2 ASSIS TÊNIS CLUBE.....	33
3.7 ANÁLISE ORIENTADA A OBJETOS.....	33
3.7.1 DIAGRAMA DE CASO DE USO.....	34
3.7.2 DIAGRAMA DE CLASSE	35
3.9 EXPLOSÃO DO CASO DE USO.....	37

3.9.1	CASO DE USO EFETUAR LOGIN	37
3.9.2	CASO DE USO MANTER TÍTULO	38
3.9.3	CASO DE USO MANTER TITULAR	39
3.9.4	CASO DE USO MANTER TIPO DE DEPENDENTE.....	40
3.9.5	CASO DE USO MANTER DEPENDENTE	41
3.9.6	CASO DE USO MANTER TIPO DE USUÁRIO.....	42
3.9.7	CASO DE USO MANTER USUÁRIO.....	43
3.9.8	CASO DE USO MANTER MENSALIDADE	44
3.9.9	CASO DE USO MANTER LANÇAMENTO	45
3.9.10	CASO DE USO CONSULTAR/EMITIR RELATÓRIO DE CAIXA 46	
3.10.	RECURSOS.....	47
3.10.1.	RECURSOS FÍSICOS	47
3.10.2.	PESSOAL	47
3.10.3.	EQUIPAMENTO.....	48
4.	DESENVOLVIMENTO DO TRABALHO E IMPLEMENTAÇÃO	49
4.10.	CASO DE USO	49
4.11.	AMBIENTE DE PROGRAMAÇÃO.....	50
4.12.	ESTRUTURA DO PROJETO	51
4.13.	DEVELOPMENT MODE	52
4.14.	USUÁRIO DO APLICATIVO.....	54
4.15.	PERSISTÊNCIA DOS DADOS	56
4.15.1.	CLASSE CLUBE.JAVA	57
4.16.	CLASSE PMF.JAVA.....	58
4.17.	CLASSE CADASTRARCLUBESERVLET.....	59
4.18.	WEB.XML	60
4.19.	CADASTRESEUCLUBE.JSP	61
4.20.	BUSCA DE INFORMAÇÃO DE DADOS	62
4.21.	PESQUISE SEU CLUBE.JSP.....	63
5.	REFERÊNCIAS	66

5.1 REFERÊNCIAS DE ARTIGOS E TRABALHOS.....	66
5.2 REFERÊNCIAS BIBLIOGRÁFICAS.....	67
5.3 REFERÊNCIAS ELETRÔNICAS.....	67

1. INTRODUÇÃO

Com o passar do tempo a evolução tecnológica quanto à informação tem crescido a cada minuto, sua disseminação tem sido tão gigantesca quanto seus benefícios. Segundo Nogueira 2010, uns dos recursos mais utilizados para a disseminação da informação é a internet. Por ela se tem um acesso rápido e fácil.

Durante muito tempo, pesquisas acadêmicas, indústrias, programadores entre outros, tem se aprimorado mais na construção/desenvolvimento de sistema em diferentes modalidades para simplificar o acesso à internet. Dentre muitas tecnologias encontradas voltadas para simplificar sistemas de informação, uma que vem ganhando espaço e se destacando é “Computação nas nuvens”.

Conhecido como “Cloud computing” (Computação nas nuvens), tem se tornado popular e está associada à utilização da rede mundial de computadores com uso massivo de servidores físicos ou virtuais. Contudo Mello 2010, a computação em Nuvem tem cada vez mais ganho espaço e força nos debates e nas mídias. Esta tecnologia consiste na utilização de memória e nas capacidades de armazenamento de computadores interligados pela internet. Ao conectar a rede, o usuário pode acessar seus arquivos armazenados na internet sem a necessidade de instalar nenhum dispositivo. Muitas ferramentas existente estão disponíveis para a utilização de usuários comuns, como por exemplo e-mails, armazenamento de fotos em redes sociais entre outros.

Um grande exemplo de computação nas nuvens, está presente nas empresas Google. Entretanto para Oliveira et all, a Google é vista como a maior e mais utilizada base de buscas da atualidade, pois possui uma forma de fácil manuseio e a cada dia se inova. Contendo inúmeras fontes de inovação, de diferentes temas, atrai cada vez mais usuários. E com toda essa popularidade, a Google aposta em várias tecnologias atualmente. Computação em nuvens está diretamente ligada ao nome Google. Todavia Mello 2010, completa que a Google acredita muito nas nuvens, como uma realidade futura da computação. Dentre muitos aplicativos desenvolvidos pela Google, os mais conhecidos são: G - Mail, Google Talk, Google Maps, Google Docs, iGoogle, entre outros que executam diretamente no navegador. Alguns

aplicativos são oferecidos pela Google para usuários comuns, programadores ou empresas, para serem adeptos a esta nova tecnologia, como por exemplo o Google Web Toolkit e Google App Engine. Para este trabalho o aplicativo usado será o Google App Engine ou também conhecido como Google GAE. Esta grande movimentação e popularização dos aplicativos nas nuvens levam as empresas a apostarem cada vez mais nesta tecnologia. Levando a uma dúvida: Será o fim dos softwares desktop?

1.1 OBJETIVO

O objetivo deste trabalho foi fazer uma pesquisa sobre as diferentes modalidades dentro do termo computação nas nuvens. E aprofundar no tema desenvolvendo um pequeno sistema para clube de lazer com a plataforma Google Application Engine com auxílio de pesquisas, professores, palestras e aulas obtidas durante o curso.

1.2 MOTIVAÇÃO

Entre suas motivações as que mais se destacam é o enriquecimento ao aprendizado adquirido na pesquisa sobre Computação nas Nuvens. E poder colocar em prática este estudo desenvolvendo um aplicativo. Além do mais independente de sua plataforma de criação, o sistema executará em qualquer sistema operacional, se sua criação foi Windows naturalmente sua execução poderá ser em Linux, Apple, Sollaris, Android entre outros. Já que se trata de um aplicativo usando a tecnologia Computação nas Nuvens.

2. FUNDAMENTAÇÃO TEÓRICA

2.1. COMPUTAÇÃO EM NUVENS

Computação nas Nuvens sem dúvida hoje passa a ser o marco histórico mais importante que se dá desde a primeira geração dos computadores, segundo Martins 2011, Computação em Nuvem, usado como metáfora é para representar uma descrição derivada de diagrama de redes. Este conceito é datado do início do ano de 1961, pelo professor John McCarthy, ele sugeriu que a computação por meio de compartilhamento de tempo poderia levar a um futuro onde o poder computacional seria vendido por meios de negócios utilitários. Esta idéia foi muito popular no meio da década de 60 e no meio da década de 70 foi basicamente abandonada, ao se tornar claro que as tecnologias da época não eram autossuficientes para a agregação da idéia. Entretanto o resurgimento deste conceito com a virada do milênio tem crescido e ganhado popularidade. Algumas empresas que estão entrando no ramo da computação utilitária usam principalmente para necessidade não críticas, pois ainda existe, um certo receio quanto a confiabilidade de dados. A computação nas nuvens está ligada a três camadas distintas sobre tecnologia da informação, infraestrutura, plataforma e software, o que pode ser considerado também como formas, seguimentos, estilos, tipos níveis ou mesmo camadas em Computação em Nuvens.

Todavia Nogueira 2010, o termo “Cloud Computing” ou Computação nas Nuvens, está cada vez mais vinculado à utilização da rede mundial ou computadores com uso em massa de servidores físicos ou virtuais. Em um termo geral nos últimos anos foram fortalecidos a ideia de utilização de serviços por meio de um conjunto. A rápida transmissão de dados possibilita que uma empresa possa, de forma ágil e em tempo real, manipular seus dados via uma rede integrada de aplicações serviços e dispositivos. Pode se dizer que a Computação em Nuvens se incorpora ao paradigma de arquitetura orientada a serviços (Service Oriented Architecture - SOA), onde todas as funções dos sistemas são vistas como serviços de software independentes e auto-contidos. As indústrias da computação são altamente conhecidas pelos seus “jargões” de curta vida útil. Com o avanço da tecnologia, e

pelo seu poder de englobar serviços de hardware e software, ganham formas mais complexas e geram um desafio questionável: Todos estes recursos trabalham de formas integradas?

Em nossa nova geração de Computação em Nuvens, os pioneiros tem avançado de forma suave, quase como imperceptíveis, Desde 2002, tem desenvolvido aplicativos como: editor de texto, planilha eletrônica, correio eletrônico agenda, entre outros aplicativos muito usados para escritórios e pessoas físicas.

Dentre tantos pioneiros os que mais se destacam são: Yahoo mail – ferramenta para armazenagem de mensagens eletrônicas da empresa Yahoo. Amazon que já disponibiliza grandes softwares e hardwares como Web Service. Icloud sistema operacional disponibilizado gratuitamente pela empresa Xcerion AB. E por meio de um navegador o usuário pode acessar recursos tradicionais de um sistema operacional, como Microsoft Windows, com o diferencial de que o sistema está sendo executado remotamente em um servidor em algum lugar do mundo.



Figura 1 – Nogueira, 2010 – Cenário Atual de Computação em Nuvem.

2.2. COMPUTAÇÃO NAS NUVENS VANTAGENS E DESVANTAGENS

De acordo com Souza et al 2009, Tecnologia nas nuvens é uma tendência recente onde seu objetivo é proporcionar serviços de tecnologia da informação (TI). Tendências anteriores foram limitadas a uma determinada classe de usuários ou focadas para um determinado público. Esta tecnologia pretende ser global e abranger serviços entre as massas que vão dentre hospedar seus dados, documentos pessoais ou empresas que terceirizam toda sua infraestrutura de TI. Com a migração dos dados para a nuvem, ficou fácil para o usuário comum acessar seus dados de qualquer lugar ou grandes corporações controlarem seus acessos. Este cenário de compartilhamento teve uma ocorrência semelhante a 50 anos: Um servidor de tempo, compartilhado e acessado por vários usuários. Entretanto quando os computadores pessoais começaram a ganhar mercado, seus dados e aplicações começaram a ser alocados localmente.

Naturalmente os recursos de computação e hardwares estão propensos a ficarem ultrapassados rapidamente e a utilização de plataformas de terceiros seria uma forma rápida e inteligente para os usuários lidarem com a infraestrutura.

Estes serviços de TI são fornecidos para todos usuários e permitido que desfrutem sem a necessidade de obter o conhecimento sobre a tecnologia utilizada. Para os usuários comuns, utilizarem serviços ou aplicativos nas nuvens, há a necessidade de terem em suas máquinas ou dispositivos um sistema operacional, com acesso a internet. A infraestrutura de um ambiente de computação em nuvem é composta por um grande número de centelhas ou máquinas físicas, conectados por meio de uma rede, estas máquinas físicas, possuem as mesmas configurações de software, podendo ter variações de hardware. Cada máquina física possui a capacidade de executar máquinas virtuais, ou seja, ela simula várias outras máquinas em funcionamento, de acordo com a capacidade de execução de cada máquina física.

Pode se destacar que Computação nas Nuvens traz três benefícios visíveis:

1º. Reduzir o custo em infraestrutura, podendo assim construir propostas sob demanda e recursos heterogenia.

2º. Flexibilidade escalar tanto em nível de hardware quando em nível de software, atendendo as necessidades de empresas e usuários.

3º. Oferecer uma abstração e facilidade de acesso, fazendo com que os usuários não tenham a necessidade de saber aspectos de localização física e de entrega dos resultados de serviços.

Contudo Ruschel et all 2008, complementa que Computação em Nuvem é um recurso bastante eficiente de maximizar e flexibilizar os recursos computacionais. Uma nuvem é por “natureza” um ambiente redundante e resiliente (conceito físico que significa capaz de armazenar energia quando estimulado ou exigido), ou seja, pode ser definido como a capacidade de um sistema de informação e continuar a executar corretamente apesar de mau funcionamento de um ou mais componentes. Entretanto um projeto em nuvem tem a necessidade de possuir uma infraestrutura de TI complexa, onde o usuário, não tenha que se preocupar com instalação, manutenção ou atualização de softwares. Além disso, recursos de computação e hardware rapidamente ficam ultrapassados e com uma vida útil curta. Computação em Nuvem pretende ser global, e fornecer serviços para qualquer publico, desde grandes empresas até usuários finais. Não apenas recursos de armazenamentos serão aplicados, mas toda linha pode ser aproveitado na nuvem.

Porém RSA 2012, complementa com novos paradigmas em questão da segurança, prestadores de serviços terceirizados. Para que a computação em nuvem tenha uma alternativa bem sucedida, precisará ter um alto nível de segurança similar aos atuais modelos centrados em controle, em um modo em que permitam em que as empresas tenham confiança para estender partes de seus datacenters a uma nuvem publica.

Para que a computação em nuvem possa cumprir sua promessa, ela precisa de alguns aprimoramentos, principalmente na área da segurança. Pois os aplicativos e dados presentes nas nuvens são processados de modo não confidencial, os serviços oferecem segurança mínima ou somente disponível no mercado. O crescimento das informações só tornará um problema mais urgente para as empresas, fazendo com que elas tornem responsáveis pela confidencialidade dos dados de seus usuários de ao menos 85% do universo digital em rápida expansão.

E fica claro que a computação em nuvem precisa tornar mais segura, para obter uma melhor aceitação pelas empresas. Investidores em nuvem (usuários, corporações, redes, prestadores de serviços, etc.), precisam ser cautelosamente considerados à medida que a computação em nuvem cresce a um ponto onde possa haver gerencia de dados corporativos e confidenciais.

Os datacenters convencionais tem sua segurança baseada em estruturas reforçadas que protegem os dados em subestruturas físicas, de hardware e software. Sua segurança é baseada principalmente em controle de acesso dos usuários e dos dados da infraestrutura. E seus datacenters estão em algum lugar, onde não se tem a localização exata, e sem saber quem os controla. A computação em nuvem dispersa os tradicionais limites de segurança e substitui cadeias de custodia temporária dos dados com importantes implicações para a confiança e confidencias corporativas.

Um ponto importante é fazer o tratamento dos dados antes que eles sejam migrados para a nuvem, esse tratamento é uma questão relativa aos padrões de segurança e compatibilidade, que abrangem a autenticação solida, autorizada e delegada para gerenciamento de chaves para os dados criptografados. Os servidores atuais de segurança nas nuvens podem definir os modelos verdadeiros a partir dos quais a segurança e a inovação dos controles poderão ser criadas.

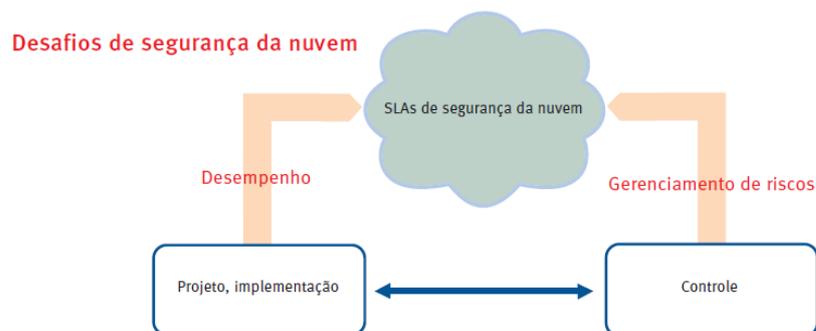


Figura 2 - Desafio de Segurança da nuvem - O Papel da Segurança na Nuvem Confiável

Embora a computação em nuvem ofereça uma promessa de arquitetura aberta e integração facilitada, as primeiras demandas de aplicativos foram criados em modelos fechados de segurança. Tanto a Amazon quanto a Google app engine, para serem acessados, os usuários precisavam ter uma conta de acesso cadastrada

a empresa, para acessarem seus arquivos. As empresas precisarão de portabilidade e de identidade de informação entre varias nuvens para que haja uma miscigenação e combinar seus serviços em um ambiente aberto e baseado em padrões que facilitem a interoperabilidade. As empresas estão encarregando seus departamentos de TI para proteger seus dados nas nuvens, tanto privada quanto publica, com a expectativa de que os dados deixarão de ser confidenciais ou passarão a ser implementadas com autorização de acesso verificável, ou seja, implementadas criptografadamente, para proteger a confidencialidade e a privacidade de seus dados. E quem imagina que o perigo esta do lado de fora, pode se surpreender, pois próprios funcionários, colaboradores ou prestadores de serviços podem obter informações de caráter especial onde os prestadores de serviço terão de entender e considerar essa responsabilidade como caráter fundamental.

Outro problema encontrado seria a facilidade de acesso, pois os requisitos de controle das informações precisarão ser controlados de acordo com o desejo dos usuários. As empresas terão participação transparente junto ao usuário, como uma forma de segurança as empresas podem fornecer uma abordagem com base em token, entretanto pode não ser aceita pelo usuário, outra forma seria com utilização de documentação, ou autorização delegada, que tornaria um outro problema, pois, embora alguns serviços forneçam de forma solida delegada, poucos ou nenhum usuário sejam adeptos a este recurso.

2.3. APLICATIVOS PARA DESENVOLVER COMPUTAÇÃO EM NUVENS

A computação em nuvens traz novamente a tona uma ideia de centralização onde vários datacenters são criados e mantidos por grandes empresas e corporações e empresas por todo o mundo. Contudo Neto e Freitas 2011, destaca que os últimos 40 anos mudaram um cenário centralizado com mainframes, onde todas as aplicações e dados eram locais e distribuídos por meios de rede internas quando aplicativos desktops utilizavam a mesma base de dados. Após as aplicações passaram a ser utilizadas via browser, de uma forma também interna até chegarmos ao nosso cenário atual.

Dentre as varias formas de computação em Nuvens e seus datacenters alguns são destaques por suas popularidades. São estas as mais populares: Google, Amazon, e Microsoft. Onde seus hardwares “feitos sob - demanda” central de energia e resfriamento de alta tecnologia, juntamente a softwares de controle, provem aos clientes formas rápidas de adicionar ou subtrair maquinas do seu meio computacional.

2.3.1. APLICATIVO AMAZON

A tecnologia Amazon foi a primeira a lançar uma plataforma de computação em Nuvens, conhecida como EC² (Elastic Cloud Computing), a tecnologia EC² é a mais heterogenia em termos de sistemas operacionais e softwares, abrangendo de uma forma geral diversos cenários. Segundo Neto e Freitas 2011, A plataforma é composta por uma série de funcionalidades que prezam o desenvolvimento e a facilitação das gerencias das instancias disponibilizadas, as principais funcionalidades são: *CloudWatch*: Webservice que permite aos clientes monitorarem em tempo real os recursos das instancias como hardware e rede. *Escalonamento automático*: permite a mudança nos recursos de hardware para adaptar a instância como hardware e rede e *Endereço IP elástico*: permite o mapeamento e rastreamento de um IP para uma ou várias instancias de forma fácil e imediata, sem a necessidade de espera da propagação de DNS.

Por padrão as instancias contem um serviço não relacionado de caráter temporário onde mantém sua memória salva mesmo com a reinicialização de maquinas.

2.3.2. APLICATIVO AZURE

A plataforma de serviços Azure da Microsoft é um grupo de tecnologia em nuvem que pode ser usado tanto para aplicativos em nuvem quanto para aplicativos locais. Seus aplicativos podem ser executados em uma variedade de sistemas. De acordo com Chapell 2008, cada componente da tecnologia azure tem sua própria função, ou seja: Windows Azure, plataforma baseada em ambiente Windows para executar

aplicativos e armazenar dados remotamente nos servidores da Microsoft. Microsoft .Net Service, responsável por disponibilizar aplicativos baseados em nuvens e locais. Microsoft SQL Services, fornece serviços de dados em nuvens baseado em SQL Server. Live Service, através do live framework, fornece acesso aos dados a partir de aplicativos live da Microsoft e outros. Esta é uma plataforma simples de se entender, pois seu objetivo é executar aplicativos Windows e armazenar seus dados nas nuvens. Seus maiores benefícios são de ao invés de comprar diferentes máquinas físicas e instalar seus aplicativos, uma organização na nuvem pode fazer isto pra ela. E elas pagarão apenas pelos aplicativos e dados armazenados ao invés de servidores.

2.4. GOOGLE APPLICATION ENGINE

O Google foi uma das empresas pioneiras a fornecerem a plataforma computação em nuvens para desenvolvimento de aplicativos. Todavia Junior 2011, demonstra que o Google Application Engine ou Google GAE é um serviço que diferente do conceito original, onde todo o ambiente responsável pelo ciclo de desenvolvimento e publicação do software é local, neste caso se obtém suas ferramentas disponibilizadas via web. Por meio de um ambiente desktop completo através de plugins para o Eclipse IDE chamado Googlipse para criar as aplicações.

Com o ambiente Google App Engine pode se criar aplicativos web nos mesmos moldes dos sistemas escalonáveis dos aplicativos da Google. Um grande diferencial é que esses aplicativos são fáceis de serem criados, manter e escalar a medida que seu tráfego e armazenamento de dados precisa para crescer. Uma vez implantado o aplicativo no App Engine os clientes já poderão acessar e utilizar os aplicativos. A Google é mundialmente conhecida por ser altamente confiável e por ter uma infraestrutura de alto desempenho. Com a tecnologia Google App Engine são garantidos 10 anos de experiência e conhecimento da Google. O mais importante para os desenvolvedores e principalmente para os usuários, é que a Google garante a mesma segurança, privacidade e escalabilidade e proteção de dados para seus aplicativos. Ela ainda oferece um leque de aplicativos e APIs para acessos aos seus

serviços. Pois sua ideia é oferecer estas AP'Is para fornecer a nuvem da Google em sistemas, para cada recurso necessário são utilizados AP'Is diferentes, ou seja, para executar um vídeo em sua aplicação basta usar o You-Tube api. Os diferentes tipos de AP'Is podem ser vistos na

Figura 3 - Tabela Periódica de API para produtos Google (Junior 2011, pg. 28).

The figure is a periodic table titled "Google APIs & Developer Products - January 2011". It is organized into columns representing different product categories and rows representing different API types. The categories include Mobile, Search, Analytics, Data APIs, Social, Maps, Ads, Gms, Tools, and Chrome. Each cell in the table contains a small icon representing a specific API, such as Google Analytics API, YouTube API, or Google Maps API.

Figura 3 - Tabela Periódica de API para produtos Google (Junior 2011, pg. 28).

Esses recursos serão sempre gratuitos garante a Google, até que atinja o limite de 500 MG de armazenamento 5 milhões de acesso de paginas mensais. E ao atingir esse limite basta ativar o plano de faturamento e pagar apenas pelo que for usado. Com os seguintes preços baseados em dólar americanos. Como mostra a Figura 4 - O Custo dos Recursos da Computação do App Engine da Google (Junior 2011, pg 28).

Recurso	Unidade	Custo unitário
Largura de banda de saída	gigabytes	\$0,12
Largura de banda de entrada	gigabytes	\$0,10
Tempo de CPU	Horas da CPU	\$0,10
Dados armazenados	gigabytes por mês	\$0,15
Destinatários de e-mail	destinatários	\$0,0001

Figura 4 - O Custo dos Recursos da Computação do App Engine da Google (Junior 2011, pg 28).

O Google Application Engine possui suportes a aplicativos criados com os usos das linguagens de programação Python e Java, ele ainda permite o uso das bibliotecas padrões da linguagem Java. O Application Engine apresenta de forma inovadora o paradigma de computação em nuvem em que o processamento e armazenamento são distribuídos entre os clusters da Google, onde necessita de um ambiente virtual seguro para cada aplicativo chamado de Sandbox.

O Google Web Toolkit (GWT) é uma ferramenta elaborada para a utilização da plataforma GAE, pois com a padronização do projeto facilita a escala de aplicações desenvolvidas. A ferramenta trabalha com duas partes distintas, ou seja, o código a ser executado pelo cliente e outra com o código a ser executado pelo servidor. O código Java executado pelo cliente é compilado pelo Google Web Toolkit e traduzido para o JavaScript, capaz de ser executado por navegadores fazendo uma otimização do navegador deixando o cargo do compilador executar a compatibilidade entre navegadores. Aplicativos escritos neste conceito tem uma maior usabilidade por não fazer uso excessivo de solicitações e evitando frequentes carregamentos de

paginas. A própria Google utiliza esta ferramenta para vários aplicativos tais como: g-mail, Orkut, entre outras. O GWT pode ser utilizado para qualquer aplicativo web sendo escrita unicamente em Java. Sem ter a obrigação de utilizar tecnologias dinâmicas como JSP. Seu meio de compilação gera código JavaScript cross-browser, garantindo que suas aplicações sejam construídas através de uma forma de programação, que utilize tecnologias implementadas em qualquer navegador.

2.5. SANDBOX

O Sanbox é uma ferramenta virtual, que disponibiliza de uma forma limitada de acesso ao sistema operacional. Esta limitação possibilita que o App Engine distribua uniformemente as solicitações web da aplicação entre diversos clusters, podendo iniciar ou interromper os servidores para atender a demanda de chamados e trafico de dados. Assim o GAE permite que cada aplicação possua uma área isolada de (conhecida como caixa de areia), confiável e segura independente de hardware, sistema operacional e localização física do servidor. Garantindo que o funcionamento das aplicações não interfira no funcionamento das outras. Esta tecnologia de virtualização além de possibilitar a distribuição na execução do aplicativo, evita um ambiente compartilhado, e uso abusivo de recursos fazendo com que outras aplicações ativas de alguma forma afetem o desempenho das demais.

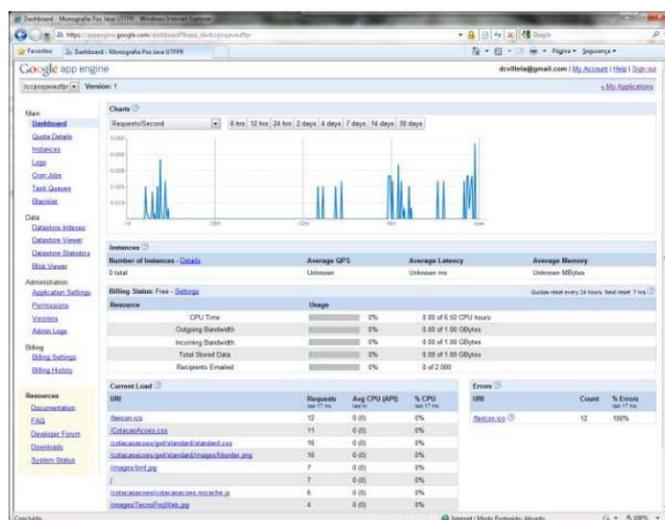


Figura 5 - Console de Administração do APP Engine da Google (Junior, 2011 pg.29).

2.6. DHTML

O DHTML ou Dynamic HTML é a união das tecnologias javascript e HTML. Uma linguagem de apresentação como as folhas de estilo Cascading Style Sheets (CSS) que seu funcionamento em conjunto com o modelo de objeto de documentos (DOM), permite que uma página web seja modificada dinamicamente, no próprio navegador do usuário sem a necessidade de novos acessos ao servidor web.

2.7. TECNOLOGIA JAVA/ECLIPSE

Uma grande vantagem de se programar em Java é que seu código fonte pode ser compilado e interpretado, ou seja, após escrever um código fonte em um editor de texto qualquer, salva se o programa como código fonte e a seguir pode ser compilado a fim de se resultar em um arquivo binário chamado de bytecodes, sendo assim este programa pode ser executado em qualquer sistema operacional por meio de um interpretador Java, com isso os códigos são escritos uma única vez, pois os bytecodes são executados independentes da plataforma de hardware e software.

Contudo Ricarte 2001, indaga que por se tratar de uma linguagem de programação Orientada a Objetos, Java possui suas classes e seus inter-relacionamentos, essa demonstração pode ser visualizada com o uso de uma linguagem de modelagem UML. Uma classe é o esboço para a definição de um corpo de objetos, que se descreve em propriedades ou atributos que um objeto terá. Com isso além da especificação de atributos sua definição descreve o comportamento de seus objetos, ou seja, quais as funcionalidades podem ser aplicadas aos objetos da classe, chamados de métodos.

Estes métodos são equivalentes a procedimentos ou funções com a restrição de que ele apenas manipula suas variáveis locais e os atributos que forem definidos para a classe.

Para o desenvolvimento deste trabalho será utilizado o ambiente de edição eclipse, pois além de se tratar de um ambiente livre de código aberto, possui facilidade de rápida visualização presente no projeto e até ferramentas de edição de acesso

coletivo. De acordo com Lupo e Cerderial, o Eclipse não precisa de instalação, basta descompactar e executar de acordo com sua plataforma, eclipse.exe na plataforma Windows ou eclipse na plataforma Linux. Em sua primeira execução ele pedirá um caminho para a pasta “Workspace”, ou seja, espaço de trabalho onde ficará armazenado os projetos de criação e edição, este diretório poderá ser alterado posteriormente utilizando o próprio eclipse. O eclipse também é muito conhecido pela vasta compatibilidade e variedade de plugins existentes, pois estes que estendem suas funcionalidades, os plugins para o eclipse pode se estender de examinadores de aspecto de qualidade de código, refactoring, UML, construtores de interface gráfica, tocadores de MP3 entre outros.

A instalação de plugins em geral costuma ser bastante simples, bastando apenas copiá-lo para a pasta “plugins” dentro do diretório e reiniciado o eclipse. Alguns mais complexo podem exigir um pouco mais de “esforço”, tendo a necessidade de copiar pastas arquivos para o diretório “features” também dentro do diretório onde o eclipse foi descompactado.



Figura 6 - Logo da Linguagem de Programação Eclipse

3. ESTUDO DE CASO

3.1. VISÃO GERAL DE UM SISTEMA PARA CLUBE

Com o avanço da tecnologia e facilidade de acesso à internet, para obter informações, muitas empresas optaram por deixar seus softwares cada vez mais preparados para o dia a dia. Em sua publicação Dias 2004, relata que o avanço da tecnologia e os sistemas de informação levaram empresas a movimentar ou mesmo reestruturar todos os seus conceitos em tarefas administrativas e/ou linha de produção industrial.

As empresas estão cada vez mais buscando sistemas robustos e complexos para sua evolução e isto faz com que elas repensem suas formas de produção, considerando o forte impacto dos sistemas de informação como um dos principais fatores de estratégia competitiva. Até mesmo pequenos empreendimentos devem possuir sistemas que além de satisfazerem suas necessidades, lhes ofereçam algo a mais. Hoje em dia até mesmo pequenos clubes de lazer necessitam de uma boa tecnologia. Segundo Belokurov, 2011, a competitividade existente nos esportes profissionais, está se tornando frequente também nos esportes amadores. No decorrer das últimas décadas, os jogos desportivos em geral ganharam um grande aliado para a perfeição e resultados melhores em desempenho, estes são softwares específicos para cada modalidade de esporte, entretanto algumas destas modalidades estão livres para serem acessadas por esportes amadores.

3.2. REQUISITOS DO SISTEMA

Em um levantamento de Requisitos, segundo Araujo et al, 2004, existem diversas abordagens para realização de processos de negócio. Cada uma delas utiliza uma determinada notação e linguagem, porém todas com o mesmo fim em que abrangem o mesmo conjunto de passos/atividade. Estes passos estão divididos em etapas:

Definir o escopo da ação é considerado uma das mais importantes ações nas divisões de processos, pois seu principal objetivo é a identificação e a priorização dos processos de softwares, que devem ser implantados para a melhoria dos processos. O projeto toma base nas diretrizes e sugestões estimuladas pelo modelo e com isso seu desenvolvimento é trabalhar dentro deste padrão.

Documentar a Missão, nesta etapa são relacionadas suas estratégias, metas e objetivos da organização, ou seja, os recursos necessários quanto a padrões de projeto, aplicativos a ser usados e recursos são obtidos nesta etapa.

Compreensão do processo/organização, esta etapa é conhecida com o termo “como esta?” em que o engenheiro de software analisa o projeto de acordo com seu cronograma, e faz sua análise técnica da evolução.

Avaliação do modelo obtido, este modelo se dá basicamente através da realização de entrevistas com profissionais representantes da unidade. Esta modalidade costuma ser uma iniciativa de definição de processo de software, através de questionários ou chek-list para avaliar o estado atual da organização.

Discutir e gerar ideias, nesta modalidade rege a criação de grupos de processos, onde os fazem reuniões agendadas para a discussão construtiva sobre modificações sugeridas e incorporadas aos padrões de projeto.

Projetar os processos, durante a realização do projeto piloto, novas necessidades de alteração vão surgindo e se tornando por consequência necessária aos padrões.

3.3. PESQUISA DA EMPRESA

Ramos dos clubes

R: Em comum os clubes de lazer controlam seus associados e dependentes como pagamentos, inadimplências, débitos e tipos de sócios.

Descrição do problema

R: A maioria dos clubes controlam seus associados por meio de fixas ou softwares que não atendem os requisitos necessários, dessa forma não conseguem obter

controle preciso para agilizar e controlar periodicamente seus associados. E isso pode ocasionar problemas para a empresa.

Resultado Esperado

R: É esperado que o sistema atenda os requisitos com as informações necessárias agilizando o cadastro de dados importantes para o controle de dados . Futuramente será integrado ao software o módulo de contas a pagar, a receber e web, para que diretores e associados possam acessar suas contas e visualizar sua área.

3.4. FUNÇÕES PRINCIPAIS

- Login no sistema;
- Manter Títulos;
- Manter Titular;
- Manter Dependente;
- Manter Tipo de Dependente;
- Manter Mensalidade;

3.5. REQUISITOS DE SAIDA

- Relatório de sócio e dependente;
- Comprovante de pagamento;
- Relatório de pagamentos.

3.6. ESTATUTOS PESQUISADOS

Como todas as empresas o clube de lazer tem suas regras com associados e dependentes. Suas regras são divididas em seu estatuto que é atualizado e disponibilizado aos seus associados, pretendentes à associação e o publico em geral. De acordo com a pesquisa realizada para a construção do software.

3.6.1 PARAGUAÇU TÊNIS CLUBE

Tendo como base no Estatuto Social do Paraguaçu Tênis Clube PTC, disponível em seu site o clube possui diferentes tipos de associados.

Proprietário, aqueles que adquirirem o título patrimonial do clube possuindo assim amplos direitos dentro da associação e de acordo com o estatuto.

Remido, os sócios proprietários que contribuíram para o patrimônio social com quantia fixada, o qual fica isento de pagamento de qualquer mensalidade possuindo ainda amplos direitos dentro da associação dentro da associação e de acordo com o estatuto.

Usuário, associados que após ser aprovado pelo quadro social, passam a frequentar o clube sem adquirir o título patrimonial da associação, mediante o pagamento de mensalidades no valor igual ao sócio proprietário, entretanto sem o direito de votos e outros benefícios aos demais associados.

Flutuante, os que em razão de seu ofício ou profissão vierem a residir na cidade sede do clube no pelo prazo não superior a 1 (um) ano e que após este período sua permanência no clube deve ser com aquisição de título sócio proprietário ou usuário.

Ausente, este sócio recebe o benefício de 50% (cinquenta por cento) de desconto na mensalidade do clube, é o mesmo sócio proprietário com a diferença no pagamento da mensalidade por conta de sua localização de moradia.

Além das diferenças nos associados seus dependentes também passam por classificações de acordo com o estatuto. Os filhos homens maiores de 18 (dezoito) anos matriculados e cursando ensino superior passam a pagar 1/5 da mensalidade, do contrario seu pagamento será de 50% (cinquenta por cento) da mensalidade do sócio, filhas são isentas de qualquer mensalidade. Estes ainda que na qualidade de solteiros.

Os sócios e seus dependentes que atrasar 2 (dois) meses consecutivos em suas mensalidades ficarão privados de todos os direitos sociais e proibidos de frequentar o clube. Os sócios e dependentes que atrasar 4 (quatro) meses consecutivos em suas mensalidades serão eliminados do quadro social, mediante processo que

constituirá de previa notificação por escrito e posterior deliberação da diretoria administrativa.

O primeiro contato de cadastro realizado cliente/empresa (PTC) é com o preenchimento da ficha de cadastro para proposta para novos associados como mostra Figura 7 - Folha de Cadastro para novos Associados.

PARAGUAÇU TÊNIS CLUBE
Fundado em 19/06/1932

Folha de cadastro para proposta de Associado

Nome: _____ sexo: _____
 RG: _____ CPF: _____ Nacionalidade: _____
 Estado civil: _____ Data de Nascimento: _____
 Residência: _____ Nº: _____
 Bairro: _____ CEP: _____
 Cidade: _____ Profissão: _____
 Onde exerce a profissão: _____
 Telefone Res.: _____ Telefone Com.: _____
 Celular: _____ e-mail: _____

Cole aqui
Sua foto
3 X 4

Nome do Dependente: _____
 Parentesco: _____ Data de Nascimento: _____ sexo: _____

Cole aqui
Sua foto
3 X 4

Nome do Dependente: _____
 Parentesco: _____ Data de Nascimento: _____ sexo: _____

Cole aqui
Sua foto
3 X 4

Nome do Dependente: _____
 Parentesco: _____ Data de Nascimento: _____ sexo: _____

Cole aqui
Sua foto
3 X 4

Sócio Apresentante: _____ Título: _____

Proponente _____ Proposto: Sócio _____

Av. Paraguaçu, nº 1203 – Centro
Paraguaçu Paulista – 19700-000

Figura 7 - Folha de Cadastro para novos Associados Paraguaçu Tênis Clube

3.6.2 ASSIS TÊNIS CLUBE

De acordo com o estatuto social do Assis Tênis Clube ATC, disponível em seu site.

Alguns itens são semelhantes, porém não idênticos, mas seguem as mesmas características. Como exemplo os tipos de sócios:

Sócios Beneméritos, os que receberam esse título em função atenção a relevantes serviços prestados ao clube.

Sócios Honorários, os que receberam este título em virtude a serviços excepcionais prestados ao clube ou ao esporte amador do país.

Remidos, os sócios cotistas que hajam adquirido direito sociais atribuídos a esta categoria.

Sócio Contribuinte, os que apenas pagando regularmente as mensalidades, não possuem o direito de votar e ser votado.

Sócio Transitório, semelhante ao sócio cotista, entretanto deve ser indicado e aprovado para ingresso ao quadro social e seu período é por tempo determinado.

Sócio Visitante, A pessoa que reside à localidade distante a no mínimo 30 km em visita a cidade, poderá sob a responsabilidade de um sócio, frequentar o clube num prazo de até 30 dias, com um valor a ser estabelecido pela diretoria administrativa do clube.

Sócio Militante, convidados pela sociedade para exercer atividades esportivas específicas, que representem efetiva colaboração na área de sua militância.

3.7 ANÁLISE ORIENTADA A OBJETOS

A UML (Unified Modeling Language) é uma linguagem visual, gráfica que auxilia a especificação, construção e documentação de artefatos de um sistema. Antes do processo de desenvolvimento é necessário a documentação e modelos que representem de uma forma abstrata como será o funcionamento do sistema em

execução. A UML é conhecida pela capacidade de modelar uma grande variedade de sistema e poder satisfazer qualquer requisito de modelação.

Um grande apoio que a UML se dá é a liberdade para modelar independente da linguagem de programação a ser usada. Na pratica a UML nasceu para que se pudessem inserir elementos de outros métodos que se mostrem mais eficaz e incorporar elementos novos ao processo de desenvolvimento.

Antigamente não existia uma forma padrão para modelar e documentar sistemas orientado a objetos, diferentes metodologias causavam um desentendimento e confusão por parte de alguns desenvolvedores, por suas diferentes características elementos conceituais e notações. Alguma metodologia tem características que as tornam boas em determinados assuntos, porem em uma modelagem abrangente que muitas vezes tornava-se falha ou dispensável para outras aplicações.

Para solucionar este problema de modelagem orientada a objetos, Grady Booch, James Rumbaugh e Ivar Jacobson reuniram seus estudos e unificaram metodologias criando a UML, pegando o melhor de cada uma e melhorando com suporte da comunidade.

Por meio de seus diagramas é possível realizar a representação de sistemas de software sob diversas perspectiva de visualização. Facilita a comunicação de todos envolvidos no projeto e identificação de terceiros quando necessário.

3.7.1 DIAGRAMA DE CASO DE USO

O diagrama de caso de uso ou Use Case, representam uma unidade funcional do sistema, sua utilização é para organizar a visualização e os relacionamentos entre use case e atores, esse modelo fornece uma perspectiva do sistema através de um ponto de vista externo. Graficamente sua representação é feita sob a representação de use case, atores, relacionamento e o sistema. Sua identificação:

Caso de Uso é representado por uma elipse incluindo somente o nome. Ele define uma grande função do sistema, que pode ser uma função estruturada em outras funções, ou seja um caso de uso pode ser estruturado.

Ator é representado por um boneco, e um rótulo com o nome do ator que se refere a um usuário externo ao sistema, que pode ser do mundo real ou mesmo outro sistema.

Relacionamento ajuda a descrever a ligação entre caso de uso, ator e caso de uso.

Sistema representado por um retângulo envolvendo os casos de uso que compõem o sistema em seu interior.

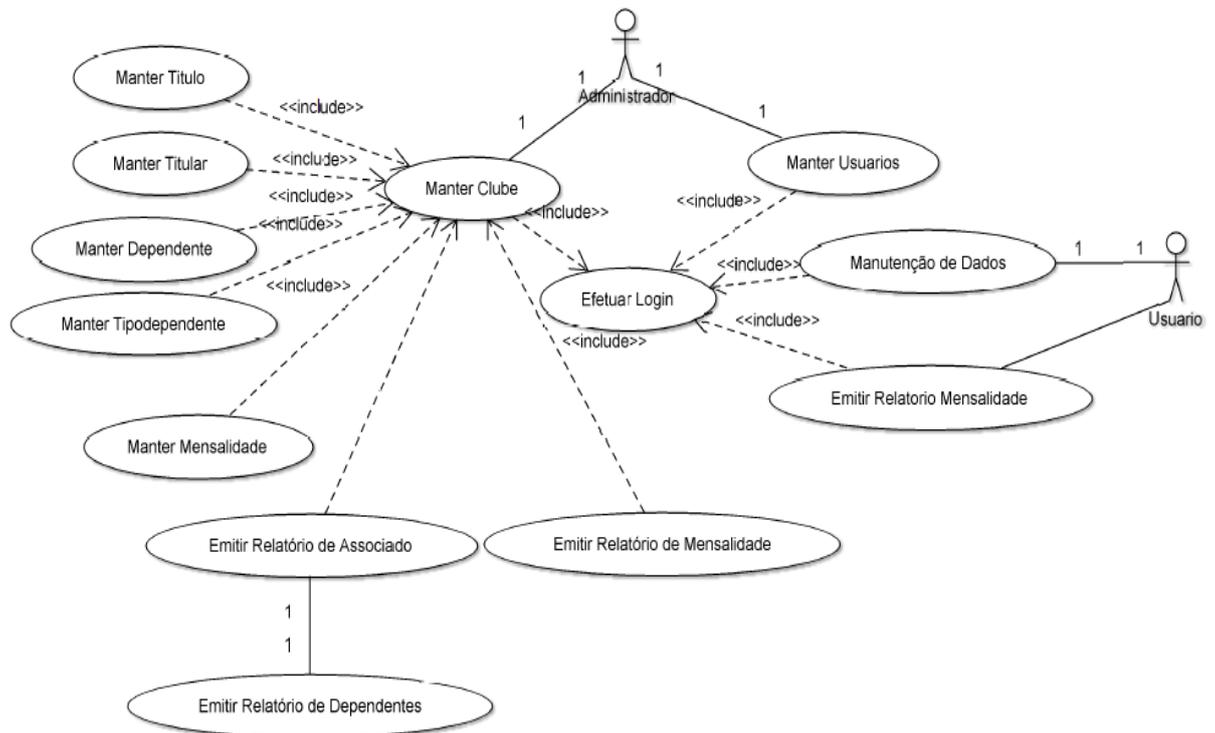


Figura 8 - Diagrama de Caso de uso

3.7.2 DIAGRAMA DE CLASSE

Segundo Mota et all 2005, diagrama de classes são representados por partes e conectores, as partes são as classes representadas no modelo e não representam instancias particulares, mas podem demonstrar notas que objetos ou instancias desempenham. Elas são inscritas de forma semelhante aos objetos.

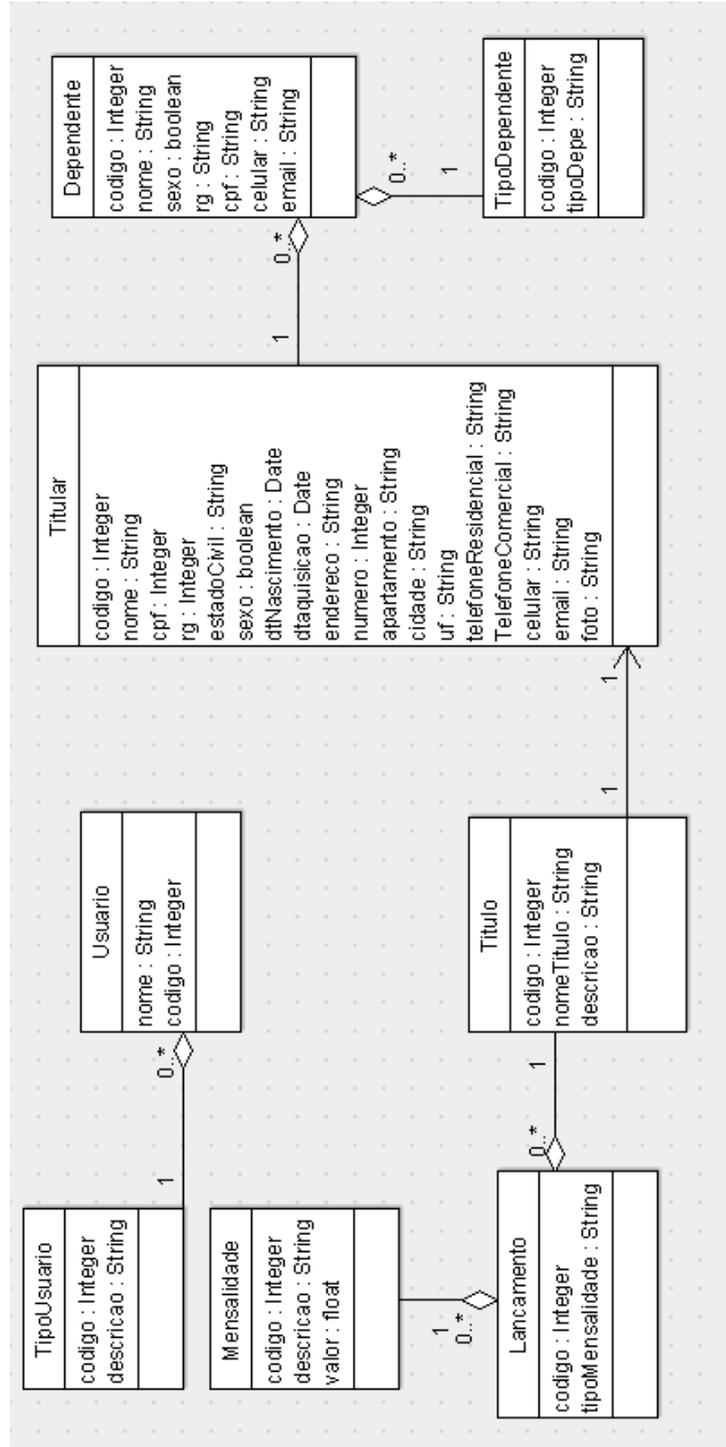


Figura 9 - Diagrama de Classes Clube de Lazer

3.9 EXPLOSÃO DO CASO DE USO

3.9.1 CASO DE USO EFETUAR LOGIN

Nome do Caso de Uso: Efetuar login.

Objetivo: Realizar o controle de acesso ao sistema através de login e senha do usuário cadastrado.

Atores: Administrador, usuário.

Cenário Principal

1. O usuário realiza acesso ao sistema Controle de Clube de Lazer, através do navegador por meio de nome de domínio;
2. É apresentado ao usuário tela de acesso onde obrigatoriamente ele deve fornecer seu login e senha;
3. O usuário digita seus dados (login e senha) e clica no botão login;
4. O sistema realiza a validação de usuário e senha;
5. O sistema é aberto e libera exercícios que estão relacionados a este usuário;

Cenário Alternativo

1. O usuário não está cadastrado no sistema ou senha e login não correspondem;

O sistema exibe uma mensagem de erro e retorna a tela de logon após o usuário clicar em OK.

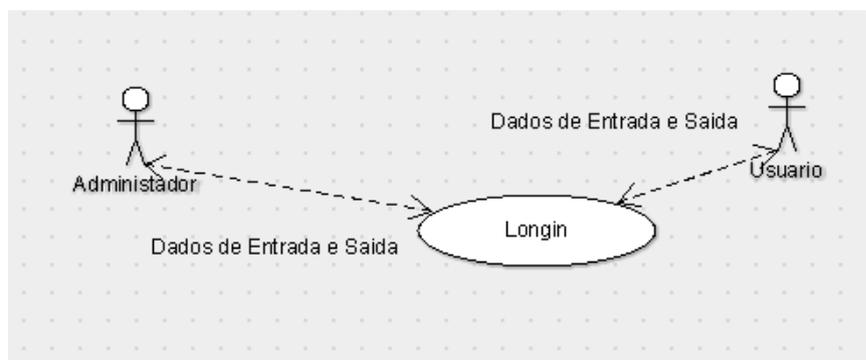


Figura 10 - Diagrama de Caso de Uso Efetuar Login

3.9.2 CASO DE USO MANTER TÍTULO

Nome do Caso de Uso: Manter Título.

Objetivo: Realizar cadastro de títulos.

Atores: Administrador.

Cenário Principal

1. O administrador seleciona a opção cadastro de títulos;
2. O sistema abre a tela com campos necessários para efetuar cadastro;
3. O administrador seleciona a opção gravar após todos os campos serem preenchidos corretamente;

Cenário Alternativo

1. O título que esta sendo gravado já existe;
- O sistema mostra em outra tela uma mensagem de erro e voltará à tela de cadastro.

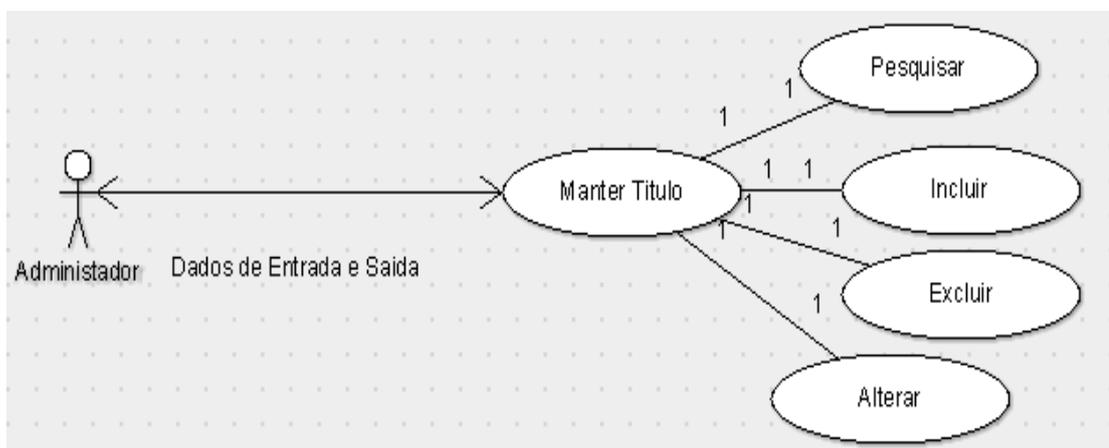


Figura 11 – Diagrama de Caso de Uso Manter Título

3.9.3 CASO DE USO MANTER TITULAR

Nome do Caso de Uso: Manter Titular.

Objetivo: Realizar cadastro de titulares.

Atores: Administrador.

Cenário Principal

1. O administrador seleciona a opção cadastro de titulares;
2. O sistema abre a tela com campos necessários para efetuar cadastro;
3. O administrador seleciona a opção gravar após todos os campos serem preenchidos corretamente;

Cenário Alternativo

1. O titular que esta sendo gravado já esta cadastrado;
- O sistema mostrará em outra tela o erro e voltará à tela de cadastro;

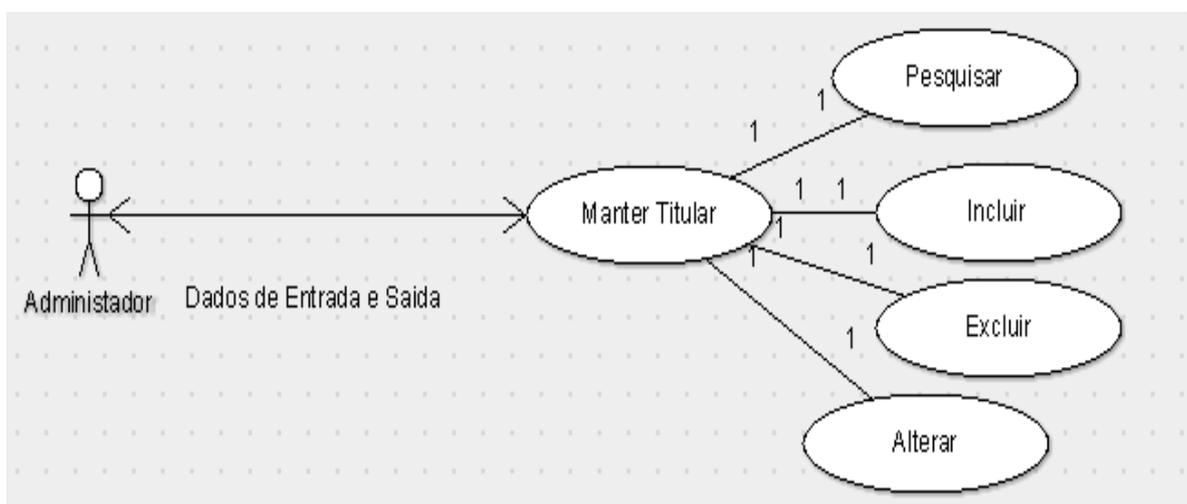


Figura 12 – Caso de Uso Manter Titular

3.9.4 CASO DE USO MANTER TIPO DE DEPENDENTE

Nome do Caso de Uso: Manter Tipo de dependente.

Objetivo: Realizar cadastro de diferentes tipos de dependentes.

Atores: Administrador.

Cenário Principal

1. O administrador seleciona a opção cadastro de tipo de dependentes; Neste cadastro é possível realizar o cadastro de conjugue, pais, filhos, entre outros.
2. O sistema abre a tela com campos necessários para efetuar cadastro; O administrador seleciona a opção gravar após todos os campos serem preenchidos corretamente

Cenário Alternativo

1. O tipo de dependente que esta sendo gravado já esta cadastrado; O sistema mostrará em outra tela o erro e voltará à tela de cadastro;

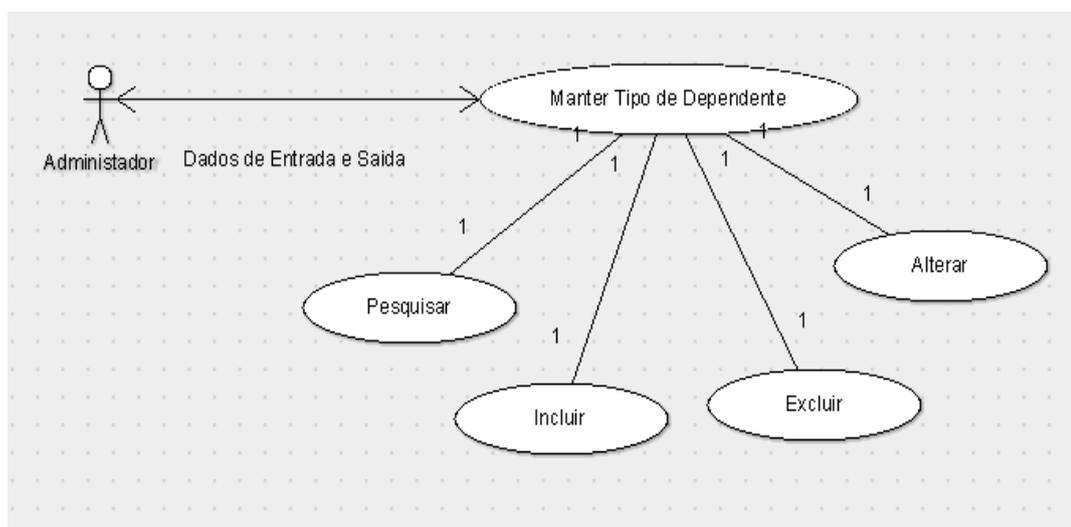


Figura 13 - Caso de Uso Manter Tipo de dependente

3.9.5 CASO DE USO MANTER DEPENDENTE

Nome do Caso de Uso: Manter Dependente.

Objetivo: Realizar cadastro de Dependentes.

Atores: Administrador, usuário.

Cenário Principal

1. Este módulo está ligado ao cadastro de Titular, só é possível cadastrar um dependente se houver um titular.
2. O sistema abre a tela com campos necessários para efetuar cadastro;
O administrador seleciona a opção gravar após todos os campos serem preenchidos corretamente

Cenário Alternativo

1. O dependente a ser cadastrado não pode permanecer no registro por critérios declarados ao tipo de dependente ou devido aos critérios do título; O sistema abrirá uma tela informando o erro e voltará a tela de cadastro.

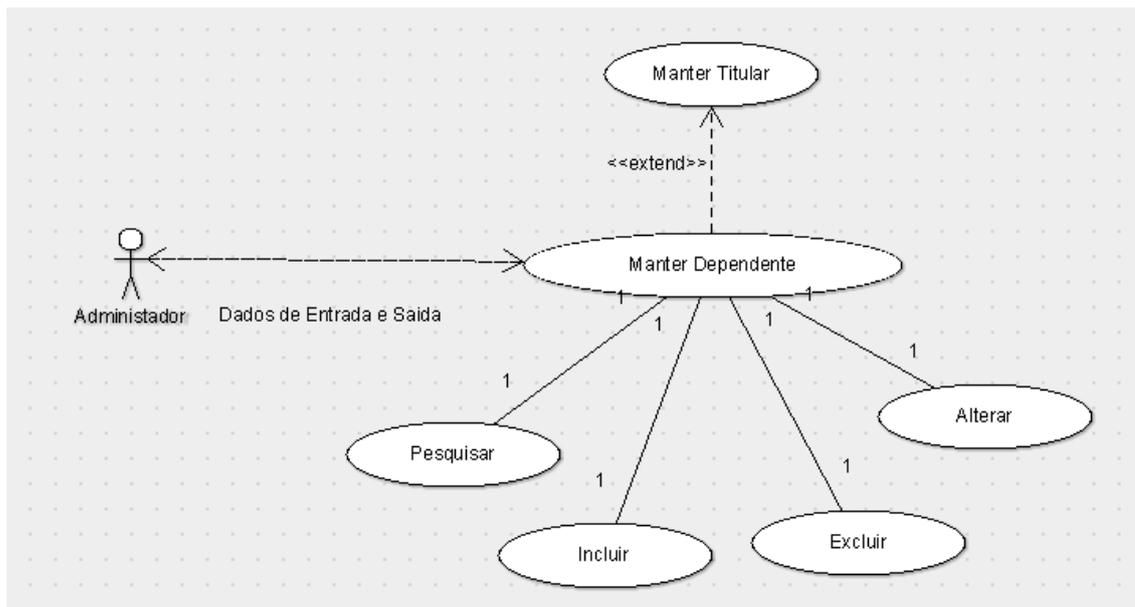


Figura 14 - Manter Dependente

3.9.6 CASO DE USO MANTER TIPO DE USUÁRIO

Nome do Caso de Uso: Manter Tipo de Usuário.

Objetivo: Realizar cadastro de tipos diferentes de usuário (Titular, dependente, administrador).

Atores: Administrador.

Cenário Principal

1. Este módulo é responsável para realizar cadastro de diferentes tipos de usuários.
2. Ele permitirá que cada usuário seja responsável pela sua área de acesso e restrições.

Cenário Alternativo

1. Caso não haja tipo de usuários cadastrados, não será possível realizar consultas.

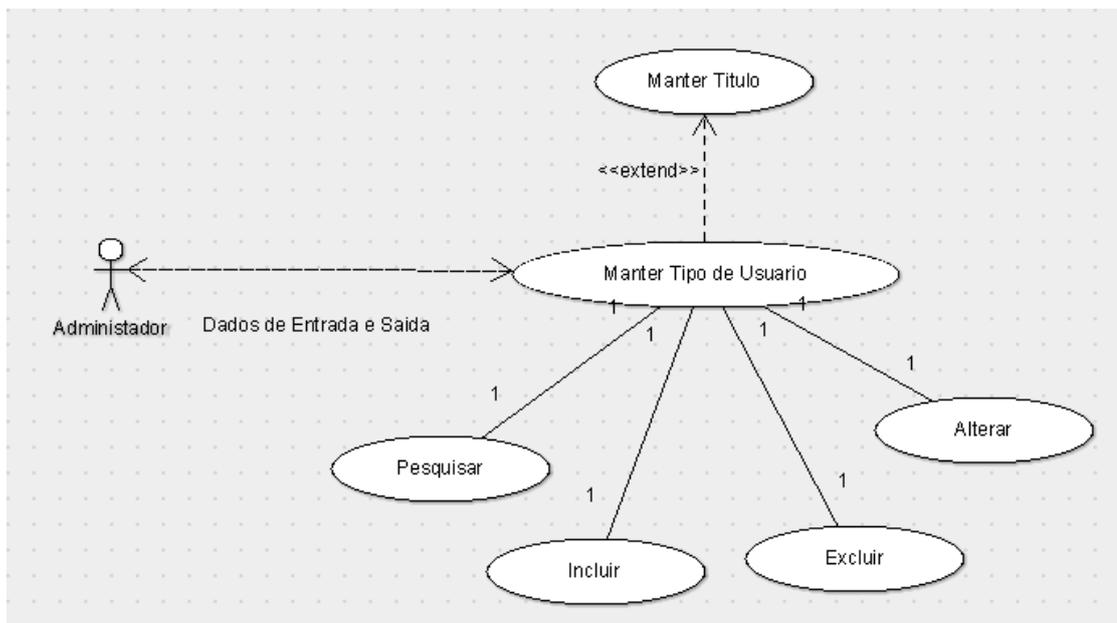


Figura 15 - Manter Tipo de Usuário

3.9.7 CASO DE USO MANTER USUÁRIO

Nome do Caso de Uso: Manter Usuário.

Objetivo: Realizar cadastro dos diferentes tipos de usuários e permitir acesso a suas devidas áreas.

Atores: Administrador.

Cenário Principal

1. Este módulo é responsável por informar quais usuários tem permissão para leitura, gravação e alteração dos dados no sistema.

Cenário Alternativo

1. Caso o usuário não tenha nenhuma permissão uma mensagem o informará de suas restrições.

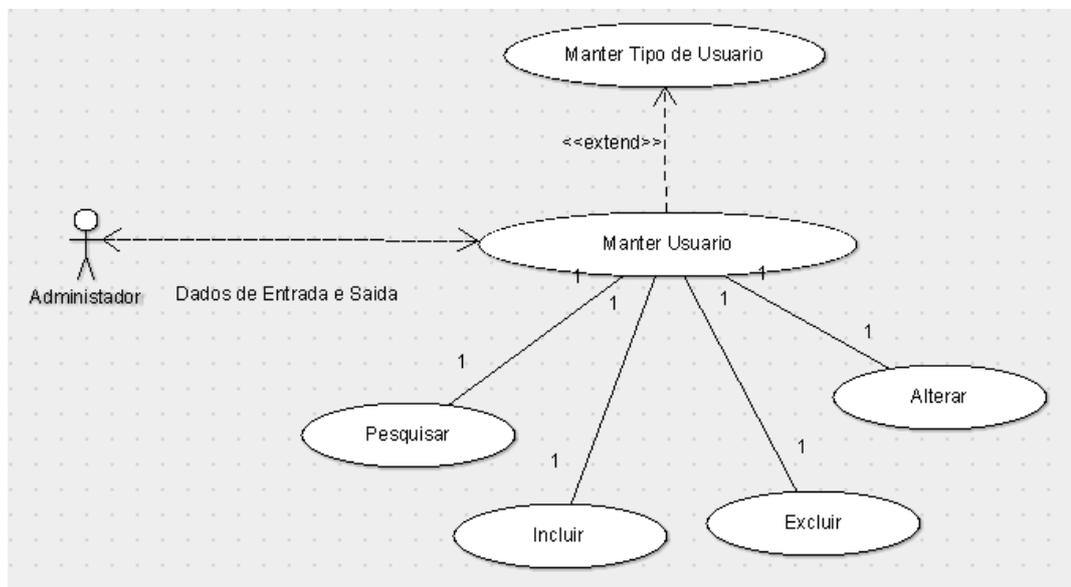


Figura 16 - Manter Usuário

3.9.8 CASO DE USO MANTER MENSALIDADE

Nome do Caso de Uso: Manter Mensalidade.

Objetivo: Realizar cadastro de diferentes tipos de mensalidade.

Atores: Administrador.

Cenário Principal

1. Este módulo é responsável para realizar cadastro de mensalidades ligando-as ao título.
2. Um título pode ter vários tipos de mensalidade.

Cenário Alternativo

1. Caso o título não tenha nenhuma mensalidade cadastrada, uma mensagem informará o motivo deste benefício.

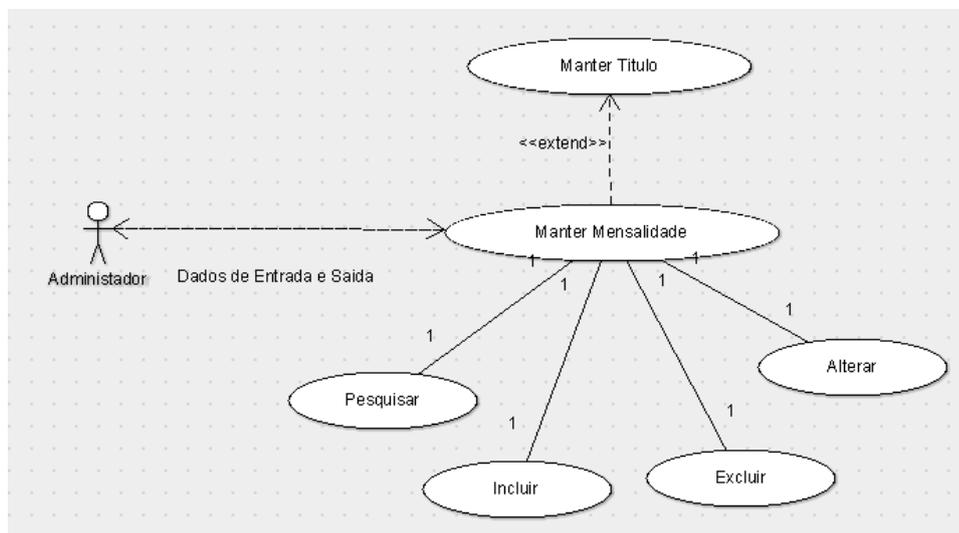


Figura 17 - Manter Mensalidade

3.9.9 CASO DE USO MANTER LANÇAMENTO

Nome do Caso de Uso: Manter Lançamento

Objetivo: Realizar alteração nas mensalidades cadastradas para cada título.

Atores: Administrador.

Cenário Principal

1. Este módulo é responsável para realizar consultas/ alterações nas mensalidades cadastradas para cada título. Ele funcionará como um caixa informando mensalidade em aberto, em dia ou as liquidando.

Cenário Alternativo

1. Caso o título não tiver mensalidades cadastradas, também não haverá lançamentos. Uma mensagem informará o motivo deste benefício.

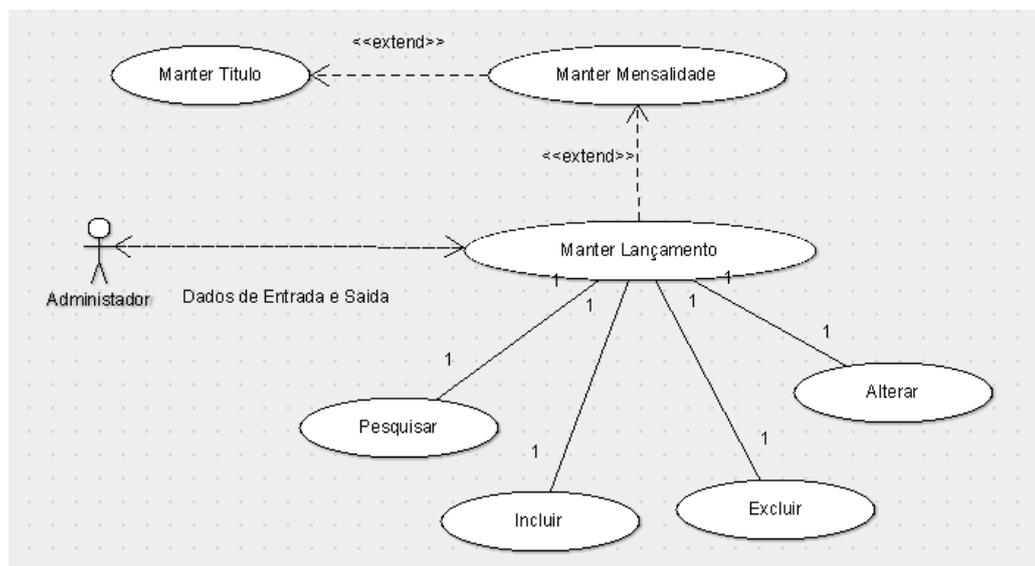


Figura 18 - Manter Lançamento

3.9.10 CASO DE USO CONSULTAR/EMITIR RELATÓRIO DE CAIXA

Nome do Caso de Uso: Consultar/Emitir Relatórios de Caixa.

Objetivo: Realizar consulta de contas a receber cadastrados ao caixa

Atores: Administrador, usuário.

Cenário Principal

1. Este modulo é responsável para realizar consultas sobre contas a receber sobre associado do clube e seus dependentes.
2. O objetivo é mostrar quais associados e/ou dependentes estão em atraso com suas mensalidades.
3. Sua consulta será realizada a partir de informações fornecidas pelo usuário..

Cenário Alternativo

1. Caso não haja mensalidades em atraso nenhum sócio será apontado.
2. O sócio que ultrapassar o limite estabelecido para estar em atraso será modificado seu quadro de situação.

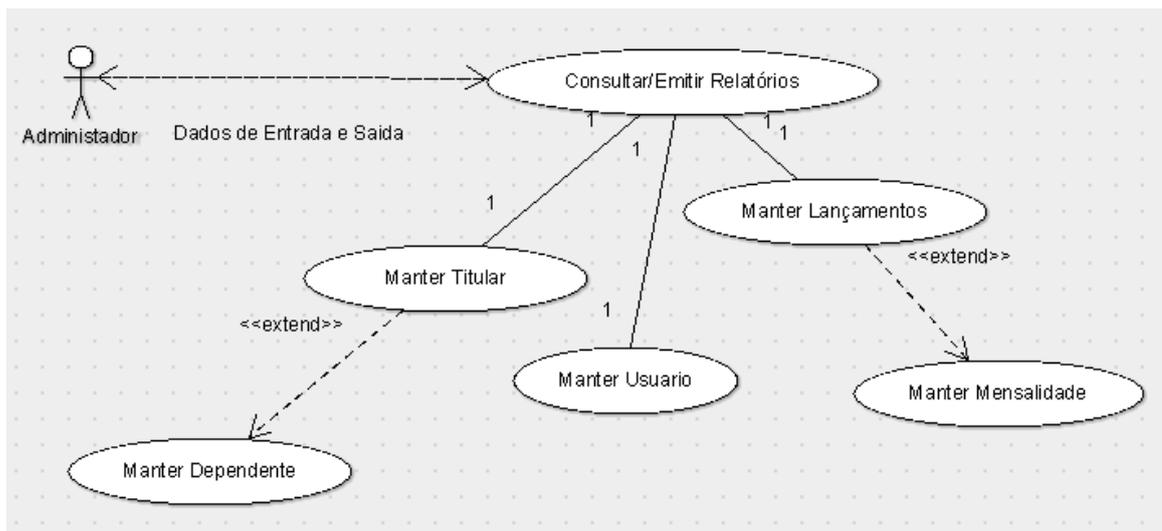


Figura 19 - Caso de Uso Consultar/Emitir Relatório

3.10. RECURSOS

3.10.1. RECURSOS FÍSICOS

Número de pessoas: 01 Analista/Programador.

Equipamentos:

- 01 notebook;
- 01 impressora jato de tinta.

Softwares

- Linguagem: Eclipse/Java
- Modelagem: Argo UML, DB Designer 4, DIA, Grantt Project;
- Banco de Dados: My SQL;
- Ferramenta de Manutenção em Banco: MY SQL;

Orçamento do projeto

- Previsão de custos para as atividades, somado com previsão de custos para os recursos.

Estimativa de Custos para as atividades

- Período de tempo para desenvolvimento;

3.10.2. PESSOAL

Analista/Programador	Quantidade	Dias	Custo (R\$)	Total (R\$)
Guilherme Vinicius Munhoz	Análise	20	44,00	880,00
	Desenvolvimento	60	44,00	2.640,00
Custo Total: 3.520,00				

3.10.3. EQUIPAMENTO

01 Computador notebook

- Valor Unitário: R\$ 2.100,00;
- Depreciação: (2 anos) / 24 meses = R\$ 87,50 por mês;
- Custo Por dia: R\$ 87,50 / 26 dias = R\$ 3,37 por dia;
- Custo do computador: R\$ 3,37 *

01 impressora jato de tinta

- Valor Unitário: R\$ 300,00;
- Depreciação: (2 anos) / 24 meses = R\$ 12,50 por mês;
- Custo Por dia: R\$ 12,50 / 26 dias = R\$ 0,50 por dia;
- Custo da impressora: R\$ 0,50 *

4. DESENVOLVIMENTO DO TRABALHO E IMPLEMENTAÇÃO

4.10. CASO DE USO

Este trabalho tem como caso de uso o desenvolvimento de um aplicativo que sirva para a gestão de qualquer clube de lazer. Entretanto como seu foco é estudar a tecnologia Computação em Nuvens seu foco ficará restrito a três tabelas como mostra a Figura 21 – Modelagem Proposta:

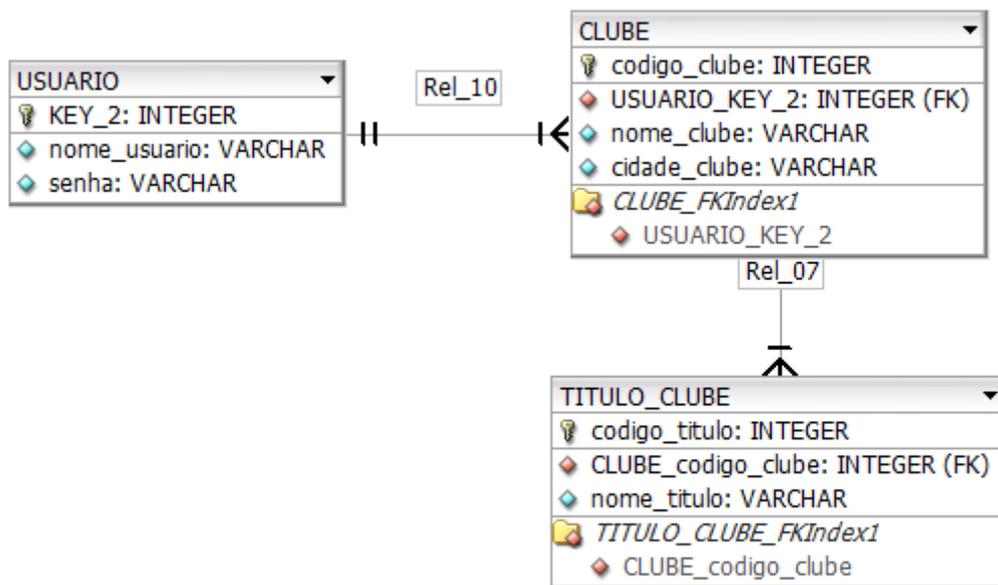


Figura 21 – Modelagem Proposta

4.11. AMBIENTE DE PROGRAMAÇÃO

Para iniciar seu projeto, a forma mais rápida é com a instalação dos plugins da Google Web Toolkit e do Google App Engine, com esta instalação é possível implementar um “Hello World” sem grandes dificuldades.

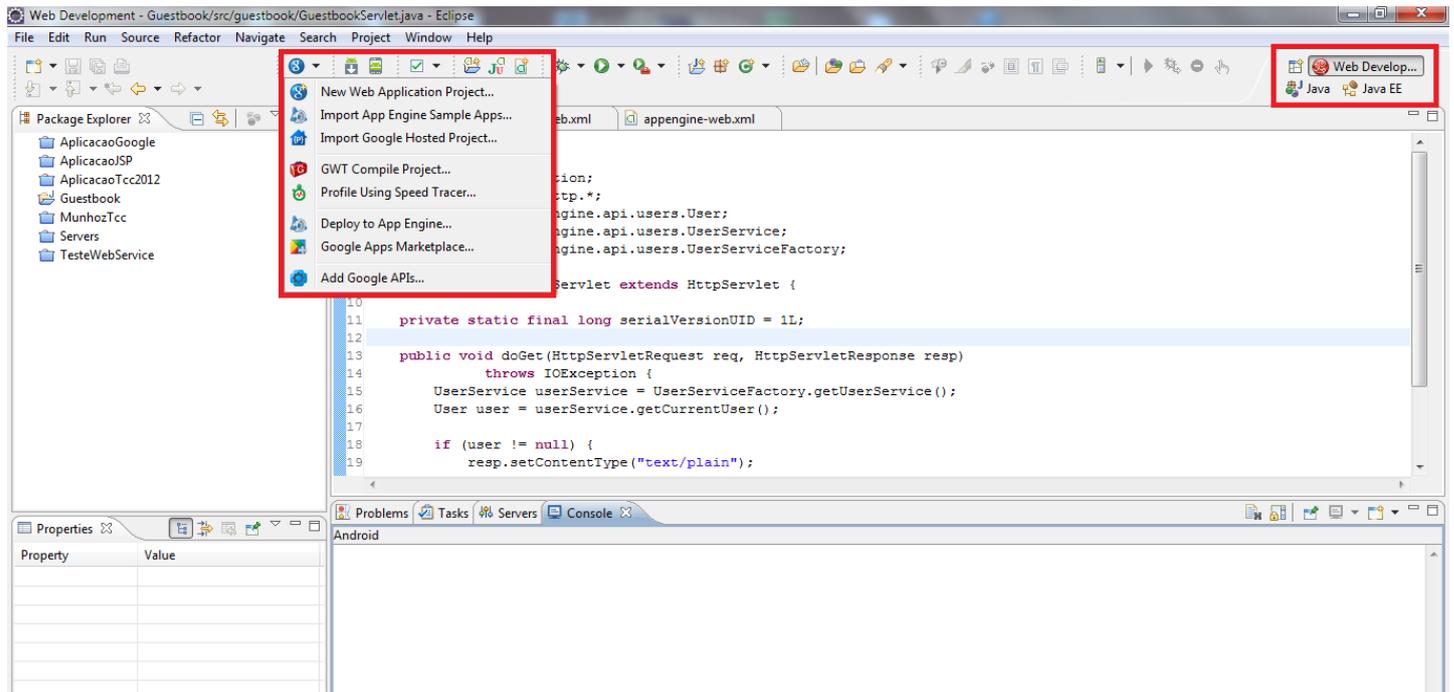


Figura 22 – Ambiente de programação Eclipse Helio 3.6 com ícones do GWT e App Engine

4.12. ESTRUTURA DO PROJETO

Ao iniciar uma aplicação a ferramenta GWT cria uma estrutura de diretórios. Todo código Java desenvolvido como classes e servlets fica residente no diretório src (source) que é dividido pelos seus subdiretórios cliente onde é executado pelo browser do usuário, shared código que pode ser executado tanto cliente como servidor e o código Server que é executado exclusivamente pelo lado do servidor.

Dentro do Diretório src , foram criados pacotes para separa suas classes por tipo, os pacotes são: beans, dão, servlet e util.

No pacote META-INF, ficam as configurações XML para as persistências no pacote APP Engine SDK, ficam o SDK para a execução do Cloud e no pacote JRE System Library ficam as runtimes.

E no pacote WEB-INF ficam os arquivos libs para agregar drivers e frame works do projeto.

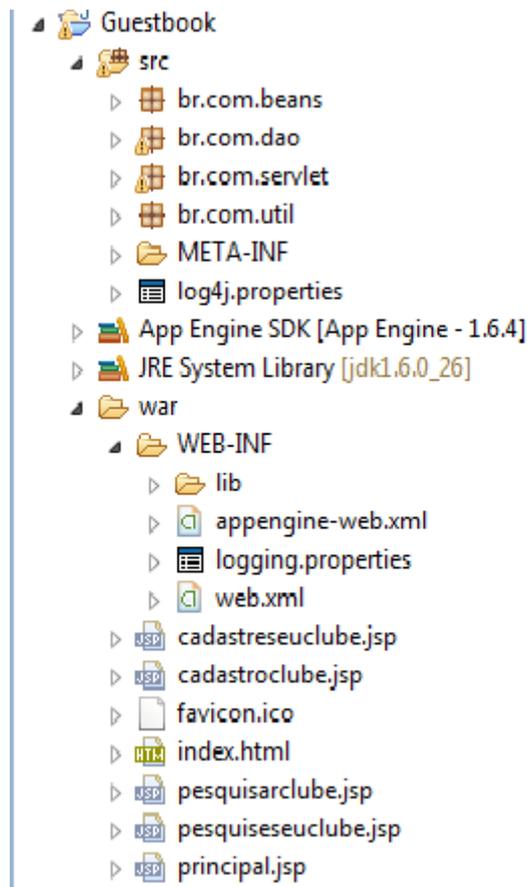


Figura 23 – Estrutura do Projeto

4.13. DEVELOPMENT MODE

Development Mode é a compilação e execução do projeto local utilizando um navegador padrão para visualização.

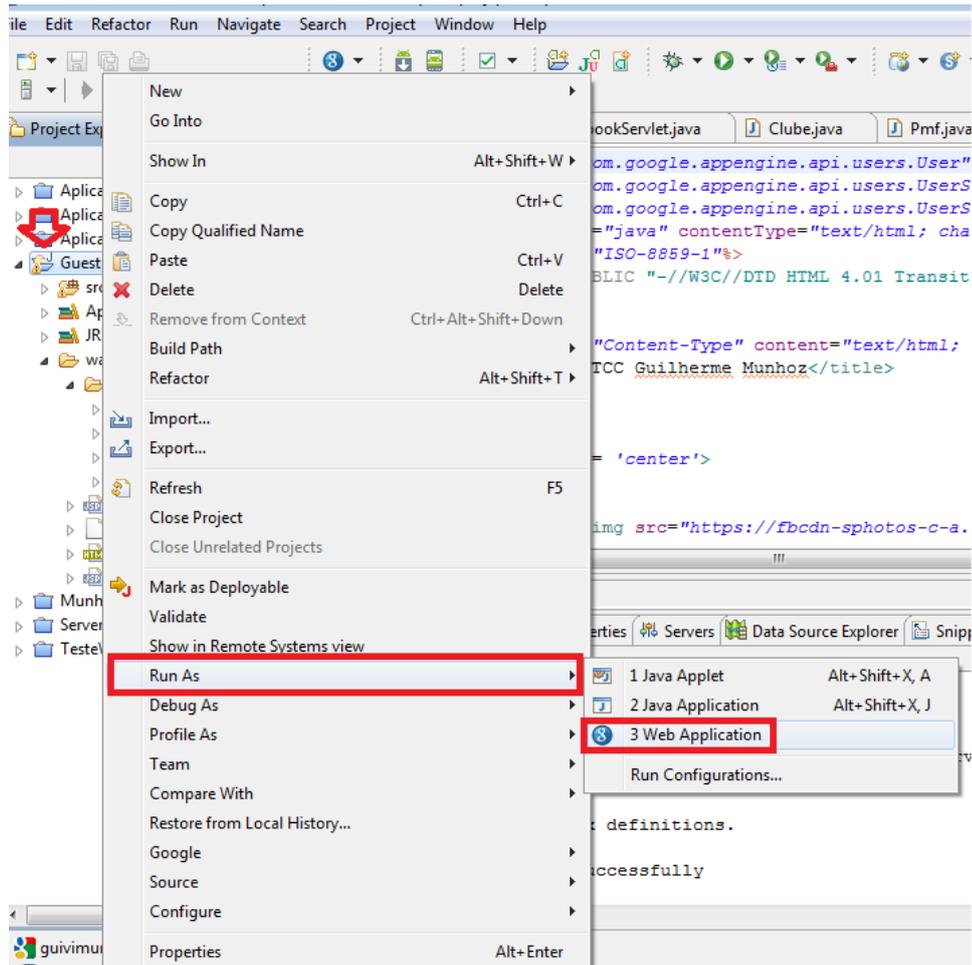


Figura 24 – Development Mode

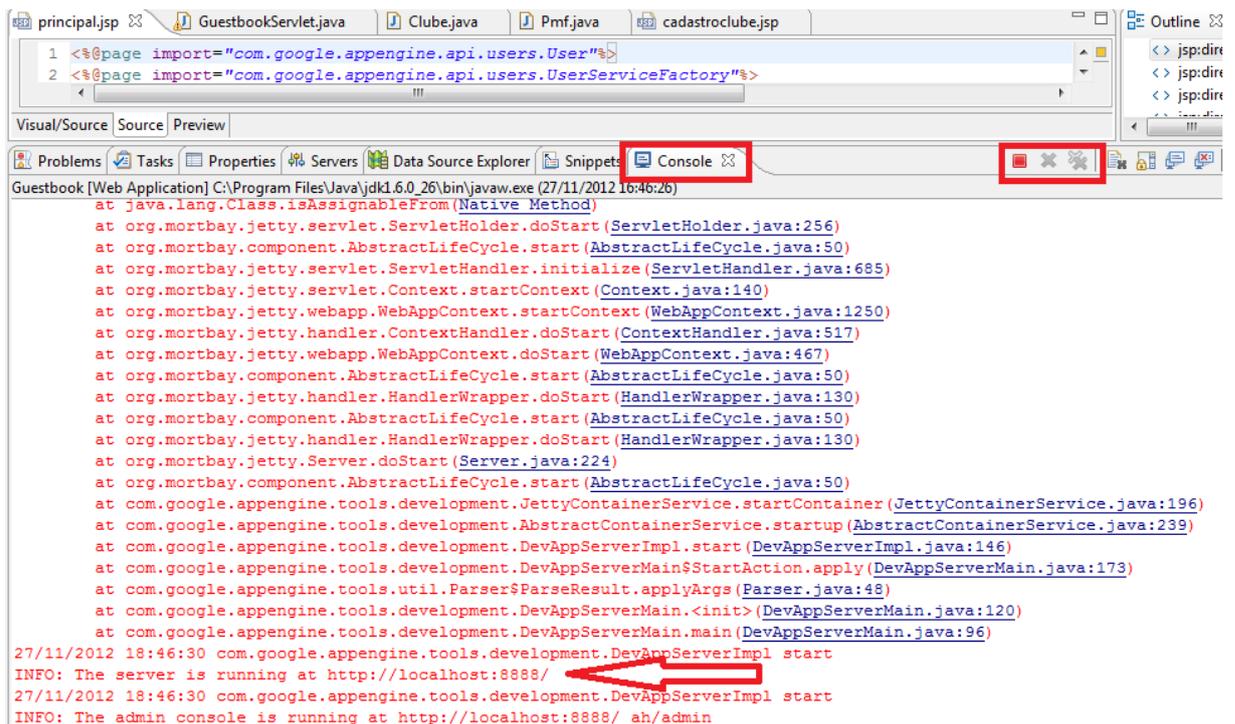


Figura 25 – Development Mode: Compilação e Servidor

Após realizado o Development Mode, na aba console mostrará o processo de compilação e um código URL para visualizar localmente sua aplicação, normalmente como: <http://localhost:8888/>

4.14. USUÁRIO DO APLICATIVO

Por ser um aplicativo web e para que não haja distorções dos dados, é essencial que para cada usuário do aplicativo, tenha seu próprio login ou área. Para resolver esse problema de forma clara a Google.appEngine fornece serviços de usuários. Basta ter um cadastro de e-mail Google (g-mail) para fazer login no projeto.

```

principal.jsp | cadastroclube.jsp | pesquisapais.jsp | Clube.java
15      <tr>
16      <td>
17      
18      </td>
19      <td>
20      <h1 align = 'center'>Aplicativo para gerenciar Clube de Laser com APP Engine</h1>
21      </td>
22      </tr>
23      </table>
24      <%
25      UserService userService = UserServiceFactory.getUserService();
26      User user = userService.getCurrentUser();
27
28      if (user != null) {
29      <%
30      <h2>Olá, <%= user.getNickname() %>! Você pode fazer seu
31      <a href="<%= userService.createLogoutURL(request.getRequestURI()) %>">logout</a> aqui.</h2><br>
32      <h2>Cadastro de Clubes <a href = "cadastroclube.jsp">aqui</a>.</h2>
33      <%
34      }else{
35      <%
36      <h2>Olá! Você pode fazer seu <a href="<%= userService.createLoginURL(request.getRequestURI()) %>">login</a> aqui.</h2>
37      <%
38      }
39      <%
40      >

```

Figura 26 - Interfaces de Usuários Google código

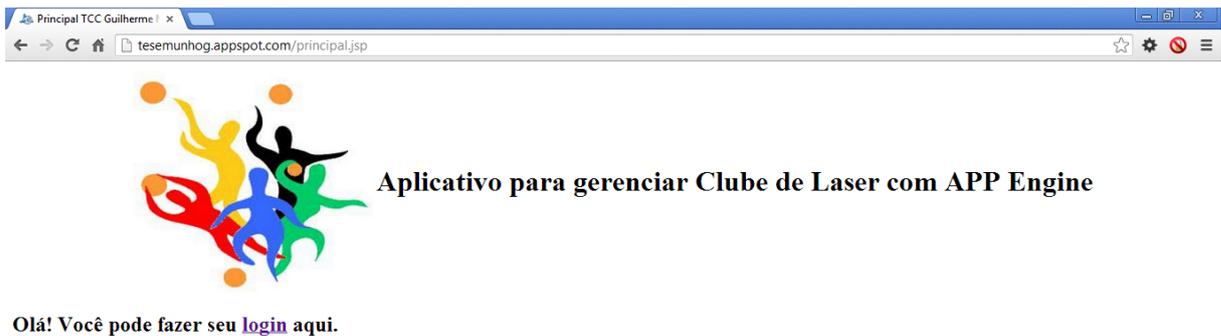


Figura 27 - Página principal do aplicativo

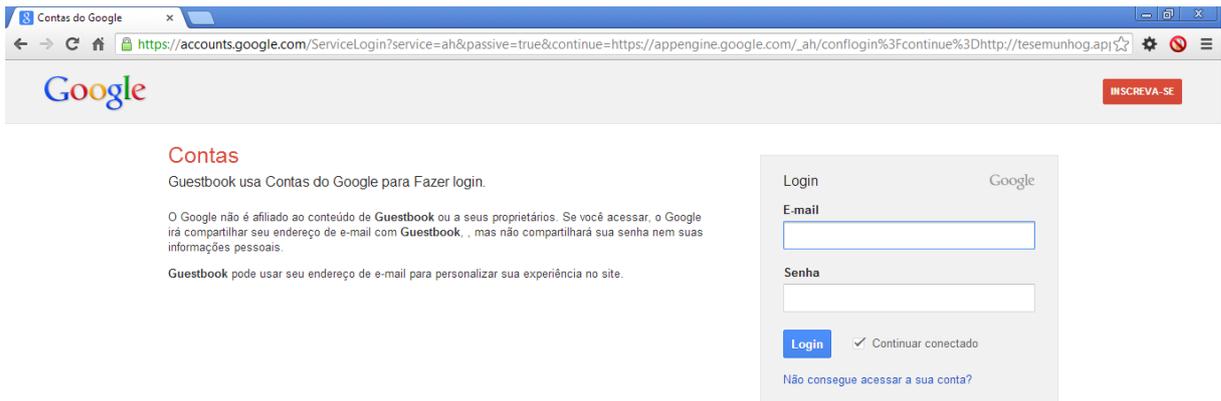


Figura 28 - Tela de Login da Google

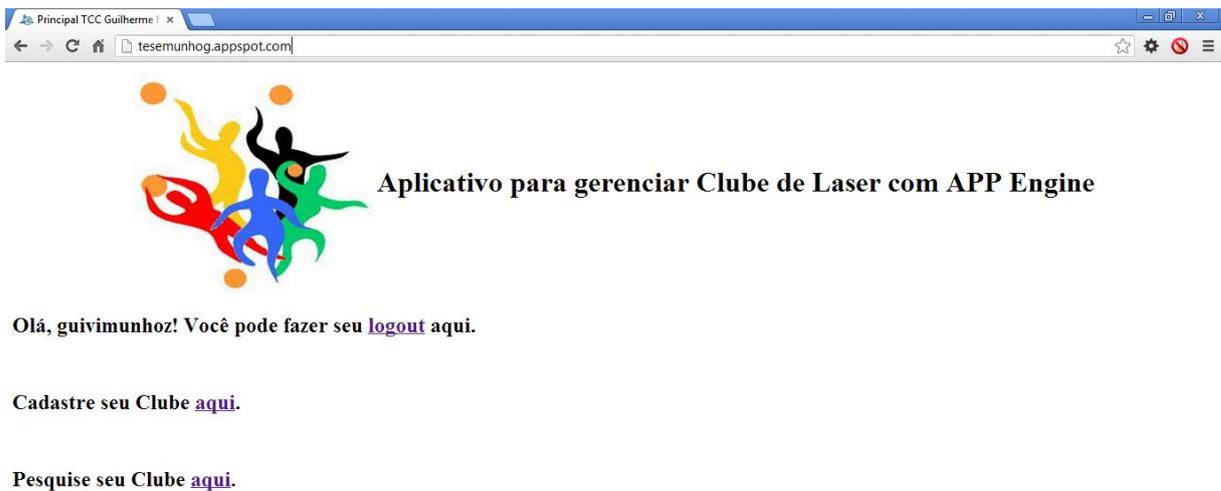


Figura 29 - Saudação ao usuário e desbloqueio do sistema

4.15. PERSISTÊNCIA DOS DADOS

Para criar uma persistência será necessário Utilizarmos as seguintes estruturas e suas modificações:

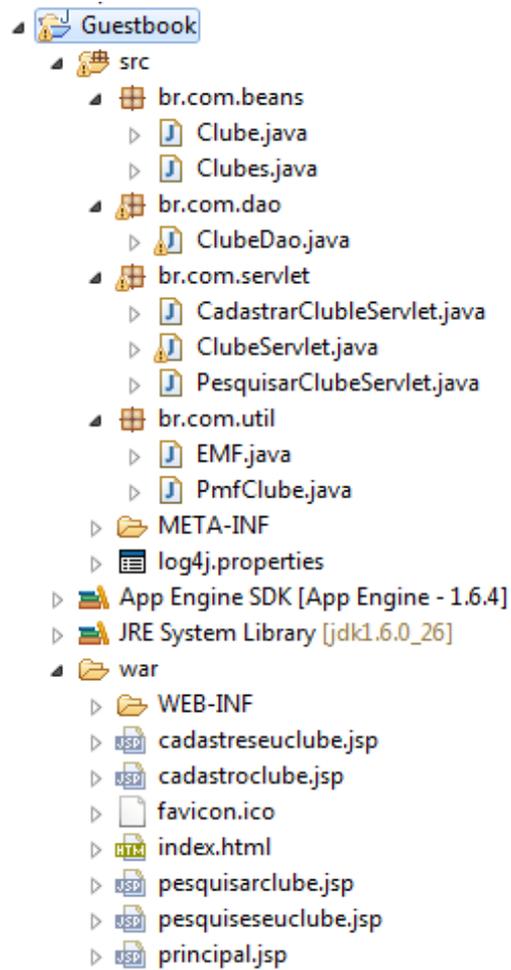


Figura 30 - Persistência dos dados

4.15.1. CLASSE CLUBE.JAVA

A classe Clube.Java que se encontra do pacote br.com.beans, contem variáveis que são responsáveis pelos dados que serão gravados no banco.

```
@PersistenceCapable
public class Clubes {
    @PrimaryKey
    @Persistent(valueStrategy = IdGeneratorStrategy.IDENTITY)
    private Key key;

    @Persistent
    private User usuarioLogado;

    @Persistent
    private String nomeClube;

    @Persistent
    private String cidadeClube;

    @Persistent
    private Date data;

    public void Clube(User usuarioLogado, String nomeClube, Date data, String cidadeClube) {
        this.usuarioLogado = usuarioLogado;
        this.nomeClube = nomeClube;
        this.data = data;
    }
}
```

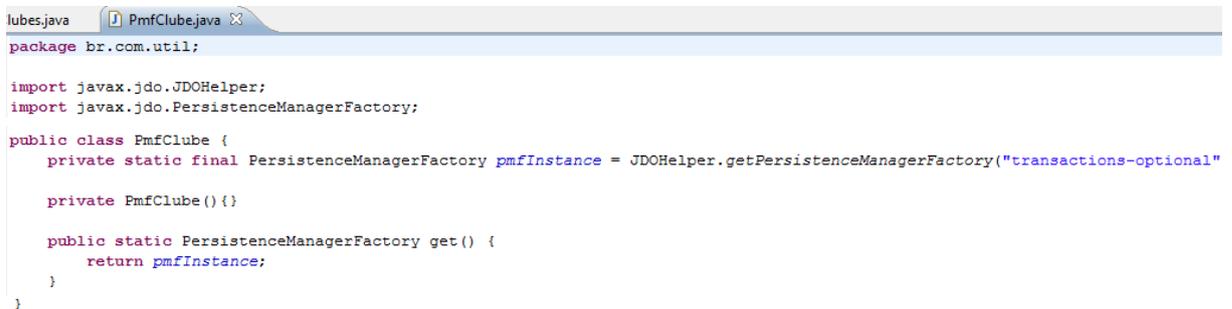
Figura 31 - Variáveis classe Clube

Esta classe Clubes.java define cinco campos privados marcados como @Persistent para determinar que o DataNucleus os armazene como propriedades do objetos de dados do App Engine. Ela ainda define as funções de obter e configurar as propriedades que são usadas somente pelo aplicativo. Contudo a Google Developers afirma que a modificação direta dos campos ignora um recurso do JDO que salva os campos atualizados automaticamente.

A classe define um campo chamado Key que é do tipo Key anotado como @Persistent e @PrimaryKey. O armazenamento de dados do Google App Engine tem uma responsabilidade com chaves e entidade e pode representar as chaves de diversas maneiras do objeto. A classe key determina todos os aspectos de armazenamento de dados, incluindo um ID numérico que é definido para um valor exclusivo quando o objeto é salvo.

4.16. CLASSE PMF.JAVA

A classe Pmf.java que se encontra no pacote br.com.util, é responsável por gravar os dados do projeto no banco, para realizar esta gravação, o App Engine usa a classe PersistenceManagerFactory. Cada solicitação de armazenamento de dados cria uma nova instancia de classes PersistenceManagerFactory. Esta instancia demora para ser realizada, mas ela pode ser carregada em uma variável estática para ser usada por varias solicitações durante a execução do aplicativo.



```
lubes.java | PmfClube.java X
package br.com.util;

import javax.jdo.JDOHelper;
import javax.jdo.PersistenceManagerFactory;

public class PmfClube {
    private static final PersistenceManagerFactory pmfInstance = JDOHelper.getPersistenceManagerFactory("transactions-optional");

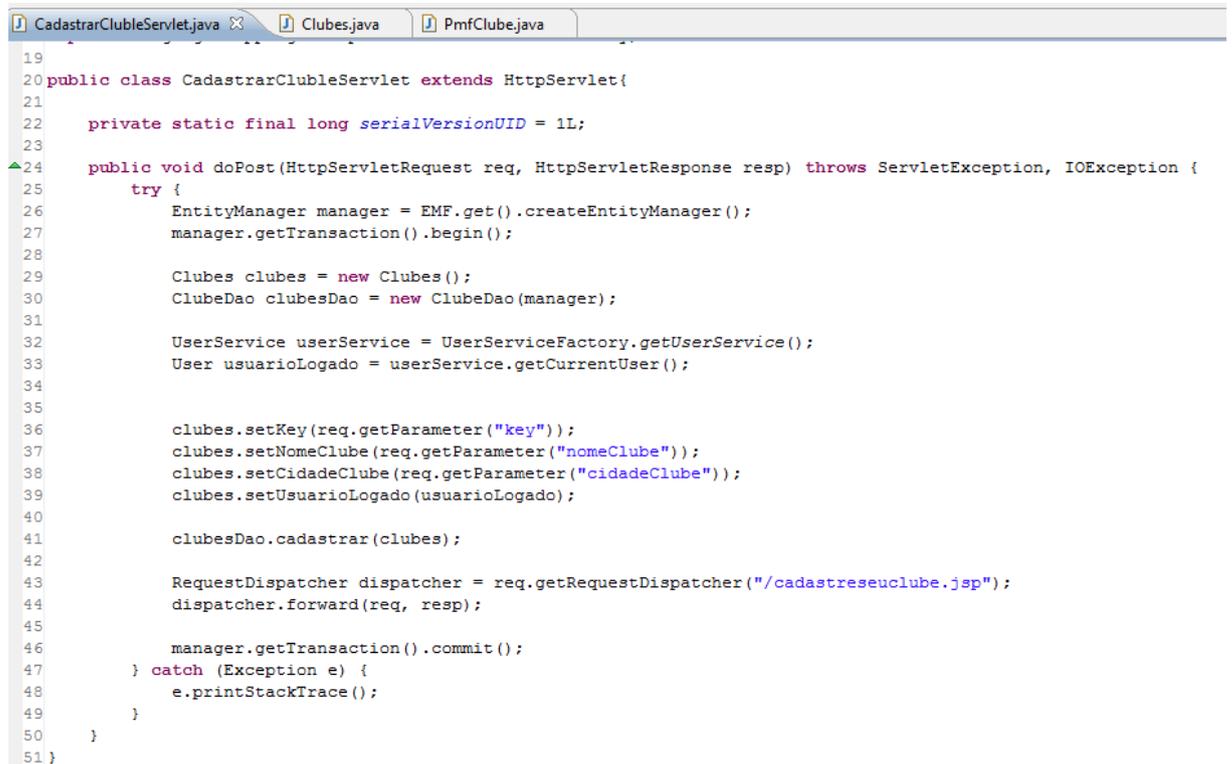
    private PmfClube() {}

    public static PersistenceManagerFactory get() {
        return pmfInstance;
    }
}
```

Figura 32 - Classe PMF

4.17. CLASSE CADASTRARCLUBESERVLET

Esta classe se encontra no pacote br.com.servlet



```
19
20 public class CadastrarClubeServlet extends HttpServlet{
21
22     private static final long serialVersionUID = 1L;
23
24     public void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
25         try {
26             EntityManager manager = EMF.get().createEntityManager();
27             manager.getTransaction().begin();
28
29             Clubes clubes = new Clubes();
30             ClubeDao clubesDao = new ClubeDao(manager);
31
32             UserService userService = UserServiceFactory.getUserService();
33             User usuarioLogado = userService.getCurrentUser();
34
35
36             clubes.setKey(req.getParameter("key"));
37             clubes.setNomeClube(req.getParameter("nomeClube"));
38             clubes.setCidadeClube(req.getParameter("cidadeClube"));
39             clubes.setUsuarioLogado(usuarioLogado);
40
41             clubesDao.cadastrar(clubes);
42
43             RequestDispatcher dispatcher = req.getRequestDispatcher("/cadastreseuclube.jsp");
44             dispatcher.forward(req, resp);
45
46             manager.getTransaction().commit();
47         } catch (Exception e) {
48             e.printStackTrace();
49         }
50     }
51 }
```

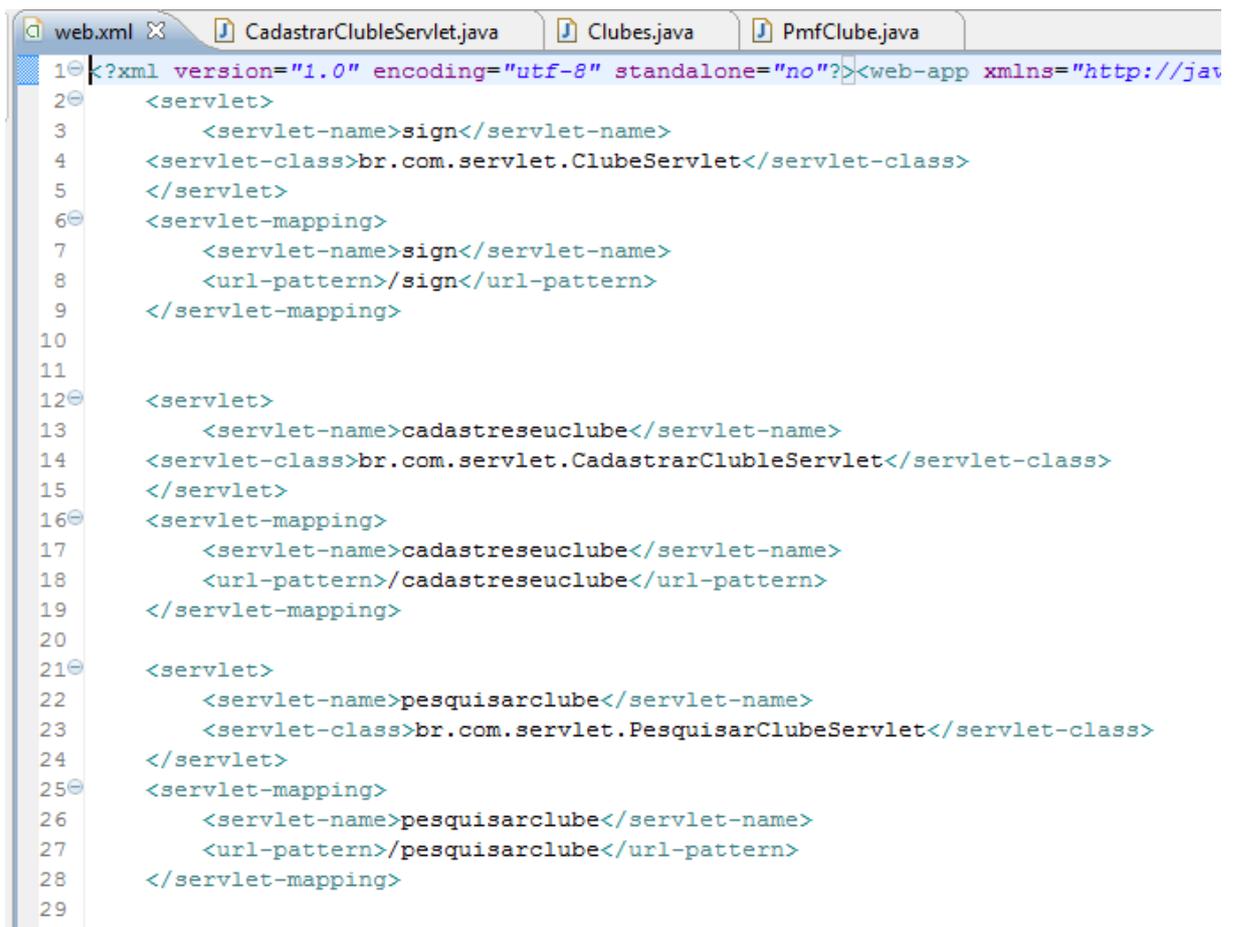
Figura 33 - Classe CadastrarClubeServlet

Esta classe cria uma nova instancia ao chamar o construtor. Para que essa instancia seja armazenada no banco, ela cria um EntityManager usando um EntityManagerFactory e em seguida passa a instancia para o método manager.getTransaction(). O aprimoramento do bytecode e anotações partem desse ponto.

4.18. WEB.XML

Este arquivo .xml, se encontra em war.WEB-INF.

Web.xml serve para fazer o mapeamento entre as classes, o aplicativo precisa de um formulário para receber as informações que serão gravadas no banco. Esse processo consiste em receber o conteúdo HTML do formulário e armazená-lo na JSP e o destino do formulário será um novo URL, "/sign" que será processado por uma nova classe de servlet (GuestbookServlet.Guestbook.servlet), este servlet apenas gravará as mensagens postadas no registro.



```
1 <?xml version="1.0" encoding="utf-8" standalone="no"?><web-app xmlns="http://jav
2 <servlet>
3     <servlet-name>sign</servlet-name>
4     <servlet-class>br.com.servlet.ClubeServlet</servlet-class>
5 </servlet>
6 <servlet-mapping>
7     <servlet-name>sign</servlet-name>
8     <url-pattern>/sign</url-pattern>
9 </servlet-mapping>
10
11
12 <servlet>
13     <servlet-name>cadastreseuclube</servlet-name>
14     <servlet-class>br.com.servlet.CadastrarClubeServlet</servlet-class>
15 </servlet>
16 <servlet-mapping>
17     <servlet-name>cadastreseuclube</servlet-name>
18     <url-pattern>/cadastreseuclube</url-pattern>
19 </servlet-mapping>
20
21 <servlet>
22     <servlet-name>pesquisarclube</servlet-name>
23     <servlet-class>br.com.servlet.PesquisarClubeServlet</servlet-class>
24 </servlet>
25 <servlet-mapping>
26     <servlet-name>pesquisarclube</servlet-name>
27     <url-pattern>/pesquisarclube</url-pattern>
28 </servlet-mapping>
29
...
```

Figura 34 - Web.xml

4.19. CADASTRESEUCLUBE.JSP

Este aplicativo para gerenciamento de clubes, precisará de um formulário web para que o usuário possa comunicar com o banco. O conteúdo do formulário será armazenado na JSP e o destino deste formulário será um novo URL “/sign” declarado dentro de Web.xml, que será processado pela classe servlet “CadastrarClubeServlet” que processará e retornará de volta para “/cadastreseuclube.jsp”

```

1 <%@page import="java.util.*"%>
2 <%@page import="java.util.List"%>
3 <%@page import="java.util.List"%>
4 <%@page import="javax.jdo.PersistenceManager"%>
5 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
6   pageEncoding="ISO-8859-1"%>
7 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
8 <html>
9 <head>
10 <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
11 <title>Cadastro de Clube</title>
12 </head>
13 <body>
14 <!-- Form para cadastrar nome do clube -->
15 <form action = "/cadastreseuclube" method = "post">
16 <table>
17 <tr>
18 <td>Nome do Clube:</td>
19 <td><input type="text" name="nomeClube"></td>
20 </tr>
21 <tr>
22 <td>Cidade do Clube:</td>
23 <td><input type="text" name="cidadeClube"></td>
24 </tr>
25 <tr>
26 <td><div><input type="submit" value="Enviar"/></div></td>
27 </tr>
28 </table>
29 <a href="/principal.jsp">Voltar</a>
30 </form>
31 </body>
32 </html>
33

```

Figura 35 - Formulário de cadastro de clube

Figura 36 - Tela de Cadastro de Clube

4.20. BUSCA DE INFORMAÇÃO DE DADOS

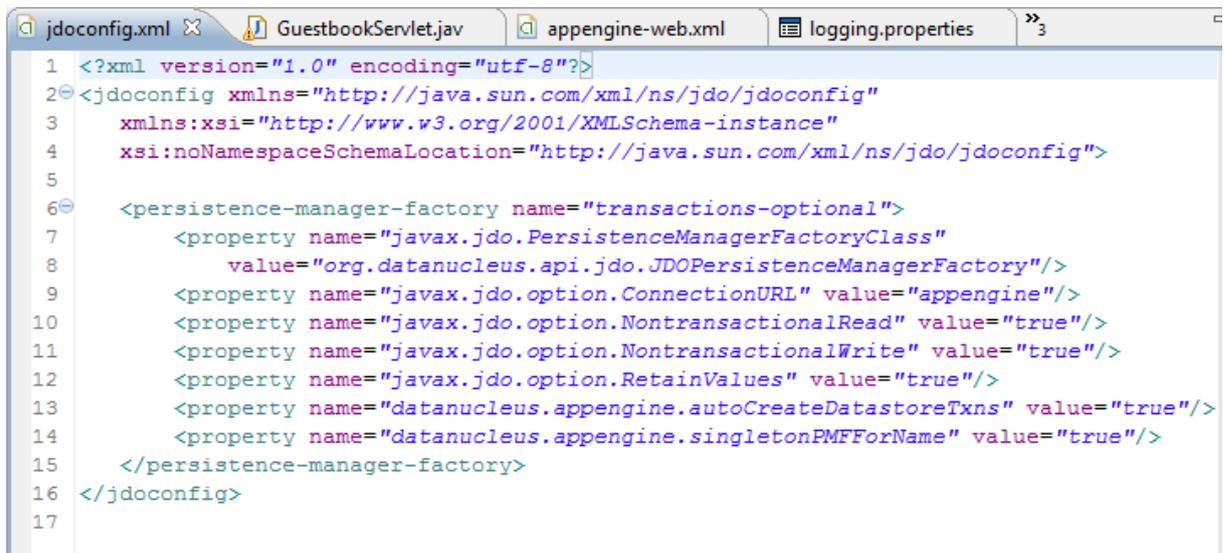
Para realizar o teste e conferir se os dados foram gravados no banco com sucesso, deve ser feito uma busca simples. O recomendado para realizar buscas é criar classes e arquivos .jsp separados. Segundo Google Developers, escalonamento de dados web pode ser complicado pois o usuário pode em determinados momentos estar reagindo com servidores diferentes, um exemplo é que o servidor que esta processando a resposta pode não ser o mesmo que executou na solicitação anterior. E isso faz com que todos os servidores web interajam com muitas maquinas que estão espalhadas ao redor do mundo.

Entretanto com o app engine não a necessidade de se preocupar com esse problema. Sua infraestrutura se encarrega de toda distribuição, replicação, balanceamento de dados usando uma simples API. O app engine possui formas diferentes de armazenamento de dados, JDO (Objetos de Dados Java) e JPA (API persistence Java), estas interfaces pertencem ao DataNucleos Access Platform, uma implementação de código free com adaptador de fornecimento de dados com app engine.

Para recuperar os registros de clubes no arquivo pesquiseseuclube.jsp será usado JDO, será feito uma busca simples.

O arquivo jdoconfig.xml, deve conter um código fonte padrão onde faz suas referencias de extensões de classes e interfaces da Google.appEngine.

4.21. PESQUISE SEU CLUBE.JSP



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <jdoconfig xmlns="http://java.sun.com/xml/ns/jdo/jdoconfig"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:noNamespaceSchemaLocation="http://java.sun.com/xml/ns/jdo/jdoconfig">
5
6   <persistence-manager-factory name="transactions-optional">
7     <property name="javax.jdo.PersistenceManagerFactoryClass"
8       value="org.datanucleus.api.jdo.JDOPersistenceManagerFactory"/>
9     <property name="javax.jdo.option.ConnectionURL" value="appengine"/>
10    <property name="javax.jdo.option.NontransactionalRead" value="true"/>
11    <property name="javax.jdo.option.NontransactionalWrite" value="true"/>
12    <property name="javax.jdo.option.RetainValues" value="true"/>
13    <property name="datanucleus.appengine.autoCreateDatastoreTxns" value="true"/>
14    <property name="datanucleus.appengine.singletonPMFForName" value="true"/>
15  </persistence-manager-factory>
16 </jdoconfig>
17
```

Figura 37 - Arquivo jdoconfig.xml

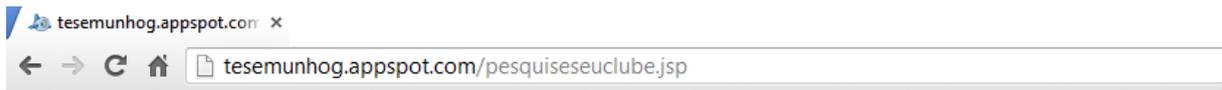
```

19  List<Clubes> clubes = (List<Clubes>) pm.newQuery(query).execute();
20  %>
21
22  <!-- Tabala para carregar os Clubes Cadastrados -->
23  <table border=1;>
24  <tr>
25      <th>Código</th>
26      <th>Nome do Clube</th>
27      <th>Cidade do Clube</th>
28      <th>Usuário</th>
29  </tr>
30  <%
31      for (Clubes clu : clubes) {
32
33      %>
34  <tr>
35      <td> <%= clu.getKey() %></td>
36      <td> <%= clu.getNomeClube() %></td>
37      <td> <%= clu.getCidadeClube() %></td>
38      <td> <%= clu.getUsuarioLogado() %></td>
39  </tr>
40  <%
41      }
42      pm.close();
43  %>
44
45  </table>
46
47  <br>
48  <a href="/principal.jsp">Voltar</a>
49 </body>
50 </html>

```

Figura 38 - Pesquisa de clube presente no arquivo pesquiseseuclube.jsp

Para realizar uma consulta é utilizado o método `newQuery()` instanciado de `PersistenceManeger()`. Esse método retorna uma consulta dentro de uma lista armazenando todos os dados encontrados.



Pesquisa de Clubes

Código	Nome do Clube	Cidade do Clube	Usuário
Clubes(5002)	Praieiros	Santos	guivimunhoz@gmail.com
Clubes(8001)	Leo Clube	null	null
Clubes(10001)	null	null	guivimunhoz@gmail.com
Clubes(11001)	Clube da Sogra	Cascavel	guivimunhoz@gmail.com

[Voltar](#)

Figura 39 - Tela de Pesquisa de Clubes

5. REFERÊNCIAS

5.1 REFERÊNCIAS DE ARTIGOS E TRABALHOS

ARAUJO, Renato, CAPELLI Claudia, JUNIOR, Gomes Augusto, PEREIRA, Marluce Iendrike, SANTOS, Hedeliane dos, IELPO, Daniel, TOVAR José Augusto: **A definição de Processo de Software sob o ponto de vista da Gestão de Processo de Negócio**: São Paulo – Brasil, 2004

BELOKUROWS, Rafael Antonio, **Aplicativo Móvel Para Controle de Dados de Time de Futebol Americano**. In: Trabalho de Conclusão de Curso, Universidade de Tecnologia Federal do Paraná, 2011, Pato Branco - Paraná.

CHAPELL, Davis & Associates: **Uma breve análise do Windows Azure, .Net Services, SQL Services e Live Services**. In: Patrocinado pela Microsoft Corporation, 2008, São Francisco - California.

DIAS, Fernando Skackauskas: **Análise da Interseção e Alinhamento entre os Sistemas de Informação e o Planejamento**. In: I Simpósio Mineiro de Sistemas de Informação, 2004, Belo Horizonte. I Simpósio Mineiro de Sistemas de Informação, 2004, Minas Gerais.

FOWLER, Martin, KENDAL, Scott, tradução PEZERICO, Vera, PRICE, Critians Thomas: **UML essencial: um breve guia para a linguagem padrão de modelagem de objetos**. Bookman, 2000

JÚNIOR, Domingos de Carvalho Villela: **Computação em Nuvem: Desenvolvimento de Aplicações Empresariais ricas na internet, na Arquitetura Java, Google Web Toolkit e Google Application Engine**: Universidade Federal do Paraná, 2011, Cornélio Procópio - PR.

MARTINS, Adriano: **Fundamentos de Computação Nuvem para Governos**. In: Serviço Federal em processamento de Dados, 2011, Brasília – DF - Brasil.

MOTA, Antonio; VALENTE Hugo; NAVEGA Ivo; PACHECO Pedro; SILVA Tiago; PACHECO José: **UML 2.0 – Modeling Language 2.0**. In: Faculdade de Engenharia da universidade do Porto, 2011, Porto – Portugal,2005.

NETO, Olavo Oliveira; FREITAS, Rejane Cunha: **Computação em Nuvens, Visão Comparativa entre as Principais Plataformas de Mercado**: Faculdade Estácio do Ceará UFPE, 2011.

NOGUEIRA, Matheus Cadori; PEZZI, Daniel da Cunha: **A Computação Agora É Nas Nuvens**: X Escola Regional de Alto Desempenho: UPF – Passo Fundo, 2010.

RICARTE, Ivan Luiz Marques: **Programação Orientada a Objetos Uma abordagem Java**. In: **Departamentos de Engenharia de Computação e Automação Industrial**. 2001, Universidade Estadual de Campinas. Campinas – SP.

RUSCHEL, Henrique; ZANOTTO, Mariana Suzan; MOTA, Welton Costa da Mota: **Especialização em Redes e Segurança de Sistemas**: Pontifica Universidade Católica do Paraná: Curitiba – PR, 2008.

SOUZA, Flavio R. C.; MOREIRA, Leonardo O.; MACHADO, Javam C.: **Computação em Nuvem: Conceitos, Tecnologias, Aplicações e desafios**^{1 2}: Universidade Federal do Ceará (UFC): Ceará – CE, 2009.

5.2 REFERENCIAS BIBLIOGRÁFICAS

DEITEL. **Como programar em Java**. 6ª Edição.USA: Parson, 2001.

5.3 REFERÊNCIAS ELETRÔNICAS

*APOSTILA DE DBDESIGNER4 MARCELO MARÇULA Disponível em: <
<https://skydrive.live.com/?cid=84f8791b3b3dd5b4&id=84F8791B3B3DD5B4%21264>
> Acessado em: 15 de Julho de 2012.*

CIÊNCIAS HUMANAS E SOCIAIS APLICADAS Uma Pesquisa Sobre Ferramentas CASE para engenharia Reversa Estática Disponível em: <http://www.sumarios.org/sites/default/files/pdfs/uma_pesquisa_sobre.pdf> Acesso em: 16 de Julho de 2012.

COMPUTAÇÃO EM NUVEM: CONCEITOS, TECNOLOGIAS, APLICAÇÕES E DESAFIOS. Disponível em: <> Acesso em: 20 de Agosto de 2012.

DIA 0.92.2 Disponível em: <[http://www.vivaolinux.com.br/artigo/Dia-O-Editor-de-diagrama-\(Microsoft-Visio\)-para-Linux/](http://www.vivaolinux.com.br/artigo/Dia-O-Editor-de-diagrama-(Microsoft-Visio)-para-Linux/)> Acesso em: 15 de Julho de 2012.

ECLIPSE POR LUPO, PETER P, CERDERIAL, CRISTINA T. Disponível em:

< <http://www.dcc.ufrj.br/~comp2/TextosJava/Eclipse-PeterLupo-30abr06.pdf>> Acesso em: 21 de Junho de 2012.

ESTATUTO SOCIAL ASSIS TENIS CLUBE. Disponível em:

<<http://assistenisclube.com.br/index.php/utilitarios/estatuto.html>> Acesso em: 10 de Abril de 2012.

ESTATUTO SOCIAL PARAGUAÇU TENIS CLUBE. Disponível em:

<<http://ptcparaguacu.com.br/conteudo/conteudo.asp?id=12>> Acesso em: 9 de Abril de 2012.

GOOGLE DEVELOPERS. Disponível em: <<https://developers.google.com>> Acesso em 12 de outubro de 2012

GOOGLE: UM FENÔMENO INFORMACIONAL?. Disponível em: <<http://revista.crb8.org.br/index.php/crb8digital/article/viewFile/70/72>> Acesso em: 10 de Agosto de 2012.

MAGNO, C Xavier Como Criar uma WBS. Disponível em:

<<http://www.beware.com.br/arquivos/comoCriarUmaWBS-Magno.pdf>> Acesso em 21 de Junho de 2012.

METRÔ - COMPANHIA DO METROPOLITANO DE SÃO PAULO Disponível em:

<http://www.etcfran.com.br/alvaro/wp-content/uploads/2012/04/ganttproject_2.0.51.pdf> Acesso em: 02 de Julho de 2012.

RSA – THE SECURITY DIVISION OF EMC – O PAPEL DA SEGURANÇA NA COMPUTAÇÃO EM NUVEM CONFIÁVEL Disponível em: <> Acesso em 26 de Agosto de 2012.