

Universidade Federal do Pampa

Rafhael Rodrigues Cunha

**SmartACG - Um aplicativo de Suporte ao
Gerenciamento de Atividades Complementares
de Graduação**

Alegrete

2014

Rafhael Rodrigues Cunha

SmartACG - Um aplicativo de Suporte ao Gerenciamento de Atividades Complementares de Graduação

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de software da Universidade Federal do Pampa como requisito parcial para a obtenção do título de Bacharel em Engenharia de software.

Orientador: Prof^o. Me. João Pablo Silva da Silva

Coorientador: Prof^o. Dr. Fábio Natanael Kepler

Alegrete

2014

Ficha catalográfica elaborada automaticamente com os dados fornecidos
pelo(a) autor(a) através do Módulo de Biblioteca do
Sistema GURI (Gestão Unificada de Recursos Institucionais) .

R972s Rodrigues Cunha, Rafael
SmartACG - Um aplicativo de Suporte ao Gerenciamento de
Atividades Complementares de Graduação / Rafael Rodrigues
Cunha.

116 p.

Trabalho de Conclusão de Curso(Graduação)-- Universidade
Federal do Pampa, ENGENHARIA DE SOFTWARE, 2014.

"Orientação: João Pablo Silva da Silva".

1. Computação Móvel. 2. Sistemas Smarts. I. Título.

Rafhael Rodrigues Cunha

SmartACG - Um aplicativo de Suporte ao Gerenciamento de Atividades Complementares de Graduação

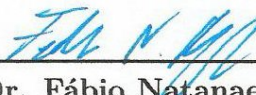
Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de software da Universidade Federal do Pampa como requisito parcial para a obtenção do título de Bacharel em Engenharia de software.

Trabalho de Conclusão de Curso defendido e aprovado em 23 de março de 2014

Banca examinadora:



Prof.º. Me. João Pablo Silva da Silva
Orientador



Prof.º. Dr. Fábio Natanael Kepler
Coorientador
UNIPAMPA



Prof.º. Dr. Alessandro Gonçalves
Girardi
UNIPAMPA



Prof.º. Ma. Aline Vieira de Mello
UNIPAMPA

*Este trabalho é dedicado a minha mãe,
Maria Aparecida Rodrigues Berriel, que com sabedoria e dignidade
nunca deixou faltar nada de que eu realmente precisasse.*

Agradecimentos

A Deus por ter me dado saúde, força e oportunidade para a realização deste trabalho.

A minha mãe, Maria Aparecida Rodrigues Berriel, que com muito carinho e apoio, não mediu esforços para que eu chegasse até essa etapa da minha vida. Sem o teu apoio eu não teria conseguido, te amo minha rainha.

Ao meu pai, José Gustavo de Leon Cunha, pela colaboração prestada no decorrer da minha formação.

A minha irmã, Andressa Rodrigues Cunha, por ter destinado inúmeras horas de sua vida para a realização de melhorias nos textos deste trabalho. Nunca esqueça que estaremos sempre juntos, mesmo que distantes.

A minha namorada, Adrianni Fioravante Manzoni, pela constante presença em minha vida, ajudando-me por meio de carinho, amor, respeito, inclusive nos momentos difíceis. Obrigado também por ter me aturado nas minhas horas de estresse (várias inclusive), e por tornar a minha vida cada dia mais feliz.

Ao meu orientador, prof^o Me. João Pablo Silva da Silva que, com muita paciência e atenção, dedicou do seu tempo para me orientar em cada passo deste trabalho. Agradeço em especial aos "puxões de orelha", fundamentais para que acreditasse em meu potencial.

A professora, Aline Vieira de Mello, por estar sempre disposta a ajudar nas horas em que necessitei, tirando dúvidas mesmo que sem conhecer o trabalho a fundo. Permita-me dizer que o seu apoio foi crucial para o sucesso deste trabalho.

Ao meu tio, Luis Antonio Ribeiro Rodrigues, por sempre se fazer presente em minha vida e demonstrar-me através de seus passos, o comportamento de um verdadeiro grande homem.

Ao meu colega, Thiago Cassio Krug, que esteve presente comigo nesses quatro anos de caminhada, ajudando-me nas dificuldades e incentivando-me a sempre buscar os meus objetivos, muito obrigado.

Ao meu amigo Vitor André da Silva Duarte, que através de bons conselhos no decorrer das madrugadas, fez com que a carga do trabalho se tornasse menos exaustiva.

A todos aqueles que contribuíram de forma direta ou indireta para a realização deste trabalho.

*Eu não posso mudar a direção do vento,
mas eu posso ajustar as minhas velas
para sempre alcançar o meu destino
(Jimmy Dean)*

Resumo

A popularização das tecnologias móveis potencializou o mercado de aplicativos, os quais proveem aos seus usuários os mais variados recursos. Os sistemas intitulados *smart* são capazes de realizar tarefas inesperadas pelo usuário, porém condizentes as suas expectativas. O contexto universitário é um cenário promissor para esse tipo de aplicação, pois possui um público adepto à inovação tecnológica. Atualmente na Universidade Federal do Pampa (UNIPAMPA) o processo de aproveitamento de Atividades Complementares de Graduação (ACG) é extenso, pois é fundamentado em diversas etapas executadas manualmente. Dessa forma, o trabalho tem como objetivo gerar um aplicativo *smart* que funcione por intermédio de dispositivos móveis e apoie os alunos na coleta e classificação de suas ACGs. Para a concretização do objetivo pretendido, foram executados os seguintes passos: Análise, teve como propósito delimitar o problema; Projeto, elaborou-se diversos diagramas *Unified Modeling Language* para especificar as funcionalidades a serem desenvolvidas; Implementação, etapa de desenvolvimento do aplicativo; Testes, teve como propósito comprovar a eficiência do aplicativo construído. Os resultados obtidos após a realização dos testes foram motivadores, devido ao mecanismo de inferência utilizado para realizar as classificações de atividades no aplicativo, apresentando média de acertos duas vezes maior se comparada a de erros. Por fim, conclui-se que o aplicativo desenvolvido está em um bom nível de maturidade e, caso seja disponibilizado um sistema para o coordenador, que receba o *eXtensible Markup Language* (XML) gerado pelo aplicativo, viabilizará a informatização total do processo de aproveitamento de ACGs na UNIPAMPA.

Palavras-chave: Computação Móvel. Aplicativo *smart*. Atividade Complementar de Graduação.

Abstract

The mobile technology popularization enhances the application markets, the ones which provide to their users the most varied resources. The application called smart is a specific type of application is able to make unexpected tasks of the user, however linked to his expectations. The academic context is a promising scenario for this type of application due to the public that is adept to the technological renewing. Currently the Federal University of Pampa (UNIPAMPA) the process of exploitation of Undergraduate Enrichment Activities (ACG) is extensive, as it is based on several steps performed manually. Thus, the study aims to generate a smart application that works through mobile devices and supports students in collecting and classifying their ACGs. To achieve the desired goal, the following steps were performed: Analysis, aimed to define the problem. Project, we prepared several diagrams Unified Modeling Language to specify the features to be developed. Implementation stage of the application development and testing. Tests, aimed to prove the efficiency of the application built. The results obtained after the tests were motivators, due to the inference engine used to make classifications of activities in the application, submit batting average twice as high compared to errors. Finally, we conclude that the system is in a good maturity level and, if available a system to the coordinator, which receives the eXtensible Markup Language (XML) generated by the application, will enable the full computerization of the recovery process ACGs in UNIPAMPA.

Key-words: Mobile Computation. Smart Application. Graduation Complement Activities.

Lista de ilustrações

Figura 1 – Processo atual de aproveitamento de ACGs.	20
Figura 2 – Atores no processo de aproveitamento de ACGs.	20
Figura 3 – Arquitetura do Android.	28
Figura 4 – Visão Geral da Solução.	38
Figura 5 – Diagrama de Casos de Uso.	41
Figura 6 – Diagrama de Pacotes - SmartACG.	46
Figura 7 – Diagrama de Classes - Activities.	47
Figura 8 – Diagrama de Classes - Model.	49
Figura 9 – Diagrama de Classes - Entrada Saída.	51
Figura 10 – Diagrama de Classes - DAOs.	52
Figura 11 – Diagrama de Sequência - Cadastrar Aluno.	53
Figura 12 – Diagrama de Sequência - Visualizar Aluno.	54
Figura 13 – Diagrama de Sequência - Coletar informações da ACG.	54
Figura 14 – Diagrama de Sequência - Classificar ACG.	56
Figura 15 – Diagrama de Sequência - Visualizar informações da ACG.	57
Figura 16 – Diagrama de Sequência - Editar as Informações da ACG.	58
Figura 17 – Diagrama de Sequência - Excluir ACG.	59
Figura 18 – Diagrama de Sequência - Aproveitar ACG.	60
Figura 19 – Diagrama de Entidade-Relacionamento.	61
Figura 20 – Telas do aplicativo <i>SmartACG</i>	63
Figura 21 – Saída de console - Pedido de aproveitamento de ACG.	64
Figura 22 – Diagrama de Atividades - Método <i>verificarCapacidadeDeInferir</i>	65
Figura 23 – Print Screen - Tabela do banco de dados chamada <i>base_conhecimento</i>	65
Figura 24 – Diagrama de Atividades - Método <i>inferirAtividade</i>	67
Figura 25 – Amostra da base de ACGs.	70
Figura 26 – Diagrama de Sequência - modelo de Teste 1.	72
Figura 27 – Resultados obtidos na execução do <i>script</i> de testes 1.	74
Figura 28 – Resultados obtidos na execução do modelo de testes 2 variando o tamanho da base para as atividades de Ensino.	75
Figura 29 – Resultados obtidos na execução do modelo de testes 2 variando o tamanho da base para as atividades de Pesquisa.	75
Figura 30 – Resultados obtidos na execução do modelo de testes 2 variando o tamanho da base para as atividades de Extensão.	76
Figura 31 – Resultados obtidos na execução do modelo de testes 2 variando o tamanho da base para as atividades de Gestão.	76

- Figura 32 – Resultados obtidos na execução do modelo de testes 2 variando a porcentagem de corte para inferência sob o as atividades do grupo de ensino. 78
- Figura 33 – Resultados obtidos na execução do modelo de testes 2 variando a porcentagem de corte para inferência das atividades do grupo de pesquisa. 79
- Figura 34 – Resultados obtidos na execução do modelo de testes 2 variando a porcentagem de corte para inferência das atividades do grupo de extensão. 79
- Figura 35 – Resultados obtidos na execução do modelo de testes 2 variando a porcentagem de corte para inferência das atividades do grupo de gestão. . 80

Lista de tabelas

Tabela 1 – Caso de uso detalhado - Manter informações básicas do usuário.	42
Tabela 2 – Caso de uso detalhado - Coletar informações da ACG.	42
Tabela 3 – Caso de uso detalhado - Classificar ACG.	43
Tabela 4 – Caso de uso detalhado - Visualizar informações da ACG.	43
Tabela 5 – Caso de uso detalhado - Editar informações da ACG.	44
Tabela 6 – Caso de uso detalhado - Excluir informações da ACG.	45
Tabela 7 – Caso de uso detalhado - Aproveitar ACG.	45

Sumário

1	INTRODUÇÃO	19
1.1	Objetivos gerais e específicos	21
1.2	Principais contribuições	21
1.3	Organização do Documento	21
2	FUNDAMENTAÇÃO TEÓRICA-TECNOLÓGICA	23
2.1	Visão Geral de sistemas <i>Smart</i>	23
2.1.1	Exemplos de Aplicação	24
2.2	Visão Geral de Computação Móvel	25
2.2.1	Perspectiva da Engenharia de Software para Dispositivos Móveis	25
2.2.2	Sistemas <i>smart</i> relacionados a Computação Móvel	26
2.2.3	Plataformas para Computação Móvel	27
2.2.4	SO Android	27
2.2.5	Ambiente de Desenvolvimento	29
2.3	Recuperação de Informação	30
2.4	Considerações Sobre o Capítulo	31
3	TRABALHOS RELACIONADOS	33
3.1	Processo de Revisão Sistemática	33
3.2	Resultados da Revisão	33
3.2.1	Trabalhos Relacionados ao Aprendizado do Aluno	33
3.2.2	Trabalhos Relacionados aos Serviços Universitários	34
3.3	Análise dos Resultados	35
3.4	Considerações Sobre o Capítulo	36
4	PROCESSO DE DESENVOLVIMENTO DO APLICATIVO	37
4.1	Concepção	37
4.1.1	Finalidade do Aplicativo	38
4.1.2	Por que <i>smart</i> ?	38
4.1.3	Visão Geral da Solução	38
4.2	Análise	39
4.2.1	Casos de Uso	40
4.3	Projeto	44
4.3.1	Diagrama de Pacotes	46
4.3.2	Diagrama de Classes	47
4.3.3	Diagramas de Sequência	52

4.3.4	Diagrama de Entidade-Relacionamento	59
4.4	Implementação	62
4.4.1	Mecanismo de Inferência	64
4.5	Considerações Sobre o Capítulo	66
5	VALIDAÇÃO	69
5.1	Amostra de Teste	69
5.2	Estratégia de Teste	71
5.3	Resultado dos Testes	73
5.4	Considerações Sobre o Capítulo	81
6	CONSIDERAÇÕES FINAIS	83
6.1	Trabalhos Futuros	84
	Referências	85
	APÊNDICES	89
	APÊNDICE A – FORMULÁRIO DE PESQUISA DE OPINIÃO	91
	APÊNDICE B – RESPOSTAS OBTIDAS NA PESQUISA DE OPINIÃO REALIZADA	93
	ANEXOS	103
	ANEXO A – TABELA DE ACGS - CURSO ENGENHARIA DE SOFTWARE	105
	ANEXO B – NORMAS ACADÊMICAS - DAS ATIVIDADES COMPLEMENTARES DE GRADUAÇÃO	111

1 Introdução

O advento das tecnologias interativas promoveu uma revolução na sociedade, principalmente na maneira como as pessoas interagem entre si. Essa revolução atingiu também as universidades, em razão de seu público evoluir de maneira a se tornarem pessoas tecnologicamente ativas, ou seja, passarem a utilizar da diversidade de meios eletrônicos encontrados no mercado para otimizar o seu cotidiano.

A evolução da sociedade e a modernização de diversos recursos encontrados nos meios eletrônicos proporcionou a criação de um novo conceito em sistemas computacionais capaz de surpreender os usuários de maneira proativa. Esses sistemas são intitulados *smarts*. Segundo [Herczeg \(2010\)](#), O uso do termo *smart* no contexto de sistemas que envolvem a tecnologia homem-máquina é muitas vezes acompanhada da forma de pensar: A tecnologia é capaz de realizar pelos usuários, o que eles desejam que ela faça.

[Kloch, Kristensen e Bilstrup \(2010\)](#) retrataram que sistemas *smart* devem ser construídos por meio da integração de tecnologias. O objetivo é ampliar a interconectividade entre os seres humanos e a Tecnologia de Informação e Comunicação (TIC) através da utilização de dispositivos capazes de satisfazer as necessidades do usuário. A interconectividade citada pode ser explorada por intermédio de dispositivos móveis, em razão de hoje a maioria do público frequentador das universidades possuir pelo menos um exemplar desses aparelhos.

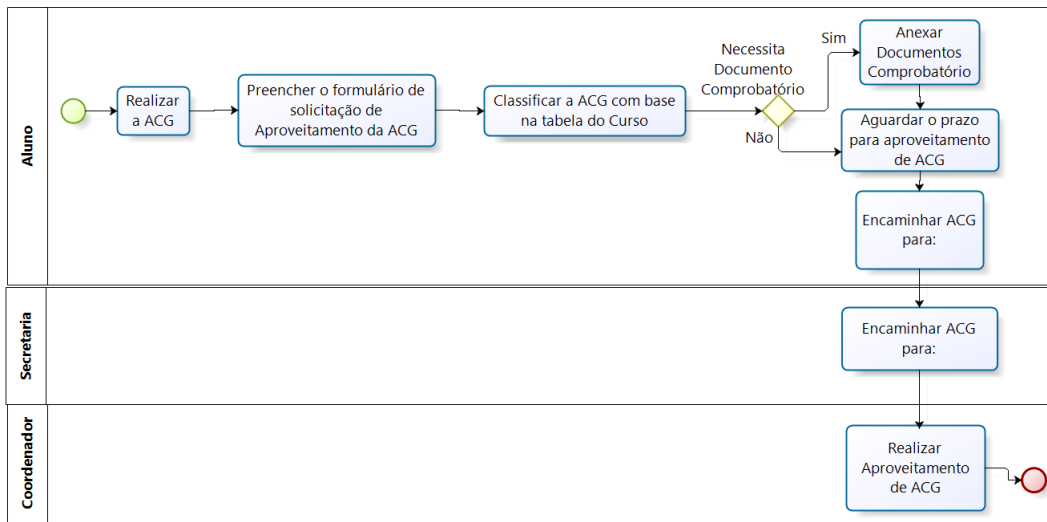
A utilização de dispositivos portáteis dentro do ambiente universitário possibilita a exploração do conceito *smart* explorado nesse trabalho. Por intermédio de softwares instalados nesses dispositivos, é cabível proporcionar a seus usuários à informatização de diversos serviços que atualmente são realizados de maneira manual, dentre eles o de aproveitamento de Atividades Complementares de Graduação (ACGs).

Com a inovação do currículo dos cursos superiores no Brasil, agregou-se como requisito para obtenção do diploma a realização de atividades extracurriculares que objetivam complementar o conhecimento adquirido pelos alunos no decorrer da graduação. Existem muitas possibilidades para realização dessas atividades, desde bolsas acadêmicas, até publicações de trabalhos em eventos científicos. Com isso, torna-se necessário uma forma de classificar as atividades, uma vez que é desejável que os alunos realizem grupos de atividades distintos.

Atualmente na Universidade Federal do Pampa (UNIPAMPA) - Campus Alegrete, o processo de aproveitamento de ACG é trabalhoso, pois é composto de várias etapas, as quais são vistas na [Figura 21](#). Entretanto, no momento que o aluno participa de uma atividade extra-curricular, este se preocupa em aproveitar o conhecimento proporcionado

e acaba por ignorar a coleta das informações necessárias para posteriormente pedir a sua validação. Adicionalmente, a época de pedido de aproveitamento de ACGs coincide com os dias que antecedem o final do período letivo, época essa que os alunos estão atarefados com a realização de tarefas providas das disciplinas cursadas, resultando no adiamento das requisições de validação das atividades.

Figura 1 – Processo atual de aproveitamento de ACGs.



A necessidade atual dos formulários de solicitações de aproveitamento de ACGs circularem diversas vezes entre os coordenadores de cursos e a secretária acadêmica resultam também no prolongamento desse processo. Sendo assim, na [Figura 2](#) é apresentado os atores que estão envolvidos com o procedimento atual de validação das ACGs. A solução proposta para este trabalho explora os problemas encontrados nesse processo sobre a perspectiva do aluno.

Figura 2 – Atores no processo de aproveitamento de ACGs.



Em suma, a união dos recursos encontrados nos dispositivos móveis com a infraestrutura da Universidade tornam exequível a construção de um aplicativo *smart* no contexto acadêmico da UNIPAMPA - Campus Alegrete. Propõe-se então o desenvolvimento de um aplicativo *smart*, utilizando-se dos recursos disponíveis na plataforma Android, em prol da melhoria dos serviços disponibilizados para o público universitário, mais precisamente o aproveitamento das Atividades Complementares de Graduação (ACG).

1.1 Objetivos gerais e específicos

Este trabalho teve como objetivo gerar um aplicativo *smart* que funciona através de dispositivos móveis e apoia os alunos no gerenciamento de suas ACGs. Este auxílio ocorre de forma automatizada, sendo necessário apenas que o aluno colete as informações da ACG. Complementarmente, definiu-se os seguintes objetivos específicos:

- Definir a tecnologia que será utilizada para disponibilizar as informações da ACG em tempo de sua execução.
- Definir uma solução de classificação da ACG que respeite as normas acadêmicas presentes no [Apêndice B](#) e se aplique a qualquer um dos cursos atuais da UNIPAMPA - Campus Alegrete.
- Definir uma tecnologia para exportar a saída do sistema de modo a ser aproveitada por intermédio de outro software.

1.2 Principais contribuições

As principais contribuições deste trabalho são:

- A modelagem de uma solução que represente as ACGs dos cursos da UNIPAMPA respeitando as regras definidas no [Apêndice B](#).
- A construção de um aplicativo que contemple parte das solicitações relatadas pelos alunos através da pesquisa de opinião realizada juntamente com este trabalho.
- O desenvolvimento de uma funcionalidade que por meio de inferências consegue deduzir classificações para algumas atividades ingressadas no sistema.
- A elaboração de uma forma de saída do sistema que, após existir outra solução que otimize as funções de coordenador e secretária acadêmica, acabe por informatizar totalmente o processo de aproveitamento de ACGs na UNIPAMPA - Campus Alegrete.

1.3 Organização do Documento

O restante deste documento está organizado como segue:

- [Capítulo 2](#): Apresenta uma fundamentação sobre os conceitos e tecnologias necessárias para a compreensão da solução desenvolvida e também descreve de forma sucinta alguns trabalhos que foram construídos baseados na definição *smart* que está sendo explorado neste trabalho.

- **Capítulo 3:** Apresenta a revisão sistemática utilizada para realização da pesquisa e seleção de trabalhos relacionados, após é descrito o resumo de ambos e por fim é feita uma análise dos trabalhos.
- **Capítulo 4:** Apresenta o aplicativo SmartACG, todo o processo de análise, modelagem e implementação do mesmo.
- **Capítulo 5:** Apresenta os testes executados no aplicativo e os resultados obtidos sobre os mesmos. Após é feita uma análise desses resultados.
- **Capítulo 6:** Apresenta a conclusão do trabalho desenvolvido e sugestões de trabalhos futuros para a otimização deste.

2 Fundamentação Teórica-Tecnológica

Neste capítulo é apresentada uma fundamentação teórica a fim de introduzir conceitos e tecnologias utilizadas sobre aplicativos *smart*, além do ambiente desenvolvimento sugerido pela *Android development* (ANDROID, 2013b) para a construção de aplicativos Android. A seção 2.1 apresenta a visão geral de sistemas *smart*, algumas de suas características e a sua utilização para diferentes contextos. Na seção 2.2 é exibida uma visão geral de computação móvel. É feita a conexão do desenvolvimento de aplicações *smarts* com dispositivos móveis, exibindo as características gerais do SO Android e o ambiente que será utilizado para desenvolver o aplicativo proposto neste trabalho. A seção 2.3 introduz o conteúdo de recuperação de informação. Concluindo, na seção 2.4 é feito o fechamento do capítulo.

2.1 Visão Geral de sistemas *Smart*

A palavra *smart* traduzida de forma livre pode ser definida como inteligente, esperto, ativo. O uso dessa no contexto de aplicações computacionais é geralmente acompanhada pela premissa que a tecnologia pode realizar pelos usuários o que eles esperam que esta faça (HERCZEG, 2010).

Crutzen (2006) retratou que a tecnologia *Smart* deve desaparecer no ambiente para trazer aos seres humanos uma vida fácil e divertida. Cook, Augusto e Jakkula (2009) mencionou que a ideia de aplicações *Smart* não é exatamente nova, a novidade é a capacidade de se poder pensar seriamente sobre isso como uma realidade e disciplina com um conjunto único de contribuições. No seu livro, Vasilakos e Pedrycz (2006) citou que pessoas estão cercadas por ambientes e aplicações que podem sentir o seu estado, se anteciparem, e talvez até se adaptarem às suas necessidades. (WEISER, 1991) afirmou: "As tecnologias mais profundas são aquelas que desaparecem. Escondem-se na vida cotidiana até que se tornam indistinguíveis a partir desta".

Cook, Augusto e Jakkula (2009) caracterizou aplicações *smart* como sendo adaptáveis pelo fato de conseguirem interagir com diferentes contextos, transparentes por funcionarem de forma silenciosa, onipresentes devido a poderem estar em diferentes lugares ao mesmo tempo através de sensores, e sobretudo inteligentes por anteciparem tarefas que o usuário precisaria realizar em um futuro próximo. A maioria das aplicações também tem a finalidade de simplificação ou automatização de processos, atributo que pode ser visualizado no trabalho de Perozzo e Pereira (2007).

Com base nessas definições se conclui que aplicações *smart* apoiam de maneira

proativa as pessoas em suas vidas, de uma forma não intrusiva e sensata, como antecipar-se ligando as luzes das casas antes da chegada de seus moradores, pedir a sua comida favorita do supermercado, orientar os cuidados de pacientes alocados no quarto de um hospital, entre outras coisas que facilitam o cotidiano. A tendência é usar o predicado *smart* quando um sistema é capaz de realizar inesperadamente o que realmente se espera de uma aplicação "bem comportada" ou "ideal", ou seja, a aplicação em questão deve responder ao usuário de forma que surpreenda-o (HERCZEG, 2010).

Diante dessa perspectiva, há fortes razões para acreditar-se que a vida humana sofrerá fortes mudanças nas próximas décadas com a introdução de uma infinidade de dispositivos equipados com aplicações caracterizadas como *smarts*, nas quais através do poder computacional, poderá de fato melhorar a forma de viver (HERCZEG, 2010).

2.1.1 Exemplos de Aplicação

No trabalho de Yim e Woo (2013), foi proposta a construção de uma aplicação de *Video On Demand* (VOD). A ideia do aplicativo é reconhecer os interesses do usuário no momento que ele reproduz um vídeo e sugerir para ele vídeos relacionados de acordo com o perfil traçado a ele. O perfil do usuário é traçado de forma silenciosa, assim como a sugestão de vídeos, caracterizando assim uma aplicação *smart*.

A demanda por melhores serviços de saúde resultaram na melhoria da saúde eletrônica (*eHealth*). Em Najafi et al. (2011) foi proposto um *Virtual Remote Nurse* (VRN), o qual fornece um agente de enfermeira virtual instalado no computador pessoal do cliente ou *smartphone* para ajudar a gerenciar o estado de saúde do cliente continuamente. Com essa abordagem, os médicos podem atribuir tarefas diferentes para uma enfermeira virtual usando um mecanismo de definição de tarefa genérica, onde uma tarefa é definida como uma combinação de fluxo de trabalho médico, diretrizes operacionais e dados associados. VRN é controlado por profissionais que decidem sobre o tratamento do paciente. Isso permite que o VRN atue simulando uma enfermeira em tempo integral para o seu cliente.

Na visão de aplicações *smart* para melhoria/automatização de casas, pode-se citar o trabalho do Das et al. (2002) que tem como objetivo formar um ambiente para proporcionar o máximo de conforto aos seus habitantes, com custos operacionais mínimos. A ideia *smart* desse projeto é o sistema que roda nele, este monitora as atividades com auxílio de um sensor e guarda em um banco de dados para referências futuras; Através disso, o sistema prevê com base em padrões de movimento diário, onde o morador vai e automaticamente fazer coisas que antes eram realizados manualmente.

Abowd e Mynatt (2004) construiu uma casa inteligente equipada com tecnologias para rastreamento de posição humana através de sensores, Identificação por Radiofrequência (RFID), entre outros. A construção *smart* da casa é um sistema para encontrar objetos

perdidos, como chaves, carteiras, óculos, e controles remotos. O sistema utiliza tags de RFID ligados a cada objeto que o usuário gostaria de rastrear e um sistema de posicionamento interno de longo alcance para controlar esses objetos. O usuário através de um painel led equipado em vários lugares da casa solicita a localização do objeto perdido, o sistema busca a sua localização e após emite um sinal auditivo informando em que ambiente da casa se encontra este objeto.

2.2 Visão Geral de Computação Móvel

A computação móvel surgiu devido à miniaturização de dispositivos e da conectividade sem fio. O aumento da necessidade das pessoas acessarem as informações pessoais e corporativas em qualquer momento e lugar, estimulou a indústria da tecnologia computacional a construir equipamentos que condizem com as necessidades de seus usuários. Hoje, estes dispositivos fazem parte do cotidiano da maioria das pessoas, permitindo para essas a possibilidade de deslocamento em conjunto com o seu ambiente computacional. Dessa forma se proporciona uma maneira rápida e eficaz de permanecer em ligação com suas fontes de informação (GONÇALVES, 2012).

2.2.1 Perspectiva da Engenharia de Software para Dispositivos Móveis

O mercado de desenvolvimento de software para dispositivos móveis tem pelo menos 10 anos de existência, entretanto, houve um crescimento exponencial no desenvolvimento para essa plataforma desde o *iPhone AppStore* inaugurado em 2008 (WASSERMAN, 2010). Segundo o mesmo autor, atualmente existem mais de 250 mil aplicações disponíveis na *AppStore*.

Com o aumento da complexidade de aplicações para dispositivos móveis, sentiu-se a necessidade de aplicar processos da engenharia de *software* para assegurar o desenvolvimento de aplicações móveis seguras e de alta qualidade (WASSERMAN, 2010). Dessa forma, acredita-se que os processos clássicos da engenharia de *software* se aplicam facilmente no desenvolvimento de aplicativos para dispositivos móveis, porém existem áreas que devem explorar novas formas de desenvolvimento.

Diversas empresas detentoras das plataformas encontradas atualmente nos dispositivos móveis publicaram guias de programação de aplicativos para suas plataformas. O assunto mais debatido nos guias encontrados é a forma de fazer engenharia para esses dispositivos, preocupando-se com a compatibilidade deles com outros aplicativos, as diretrizes para realizar interfaces de fácil manuseio ao usuário e o seu desempenho (ANDROID, 2013a).

O trabalho de (WASSERMAN, 2010) cita as áreas mais promissoras para explorar a engenharia de *software* relacionada com o desenvolvimento de aplicações móveis. A

primeira delas é a experiência com o usuário, visto que a tela menor encontrada nesses aparelhos e a forma como o usuário interage nela, resultam em um grande impacto no design das interações que o usuário vai realizar com o dispositivo, os quais influenciam no desenvolvimento do aplicativo.

Outra característica que deve ser explorada na engenharia de software para *mobile*, segundo o trabalho do Wasserman (2010), é os requisitos não funcionais. Segundo Wasserman (2010), os requisitos não funcionais mais relevantes para dispositivos móveis são desempenho, confiabilidade, qualidade e segurança. Todos esses aspectos são pensados devido a limitação do *hardware* desses dispositivos. Outro fator que contribui para essa lista de requisitos é a interconexão dos dispositivos móveis com diferentes redes, pois o comportamento do dispositivo varia dependendo da sua conexão, além da segurança que a rede proporciona.

Teste é outra área importante para a pesquisa de engenharia de *software* para dispositivos móveis, pois é insuficiente testar um aplicativo apenas com o emulador disponibilizado pelas plataformas de desenvolvimento. É efetivamente correto testar o aplicativo desenvolvido em vários dispositivos, objetivando a visualização das exigências necessárias para a distribuição do aplicativo com qualidade.

Realizar projetos de engenharia de *software* portáteis para outras plataformas é outra prática que deve ser pensada segundo Wasserman (2010). Esse relata que atualmente os desenvolvedores escolhem uma plataforma de desenvolvimento e criam o software baseando nessa. Entretanto, com a evolução desse mercado, objetiva-se em um futuro próximo desenvolver aplicativos móveis independente de plataforma. Atualmente, existem alguns *frameworks* que tendem a facilitar o processo, como é o caso do *Phonegap*¹, um framework que utiliza de *HyperText Markup Language* (HTML), *Cascading Style Sheets* (CSS) e *javascript* para desenvolver aplicações móveis, necessitando apenas que no final do projeto o desenvolvedor exporte para a plataforma desejada.

2.2.2 Sistemas *smart* relacionados a Computação Móvel

Pelos estudos realizados até o presente momento, pode-se perceber uma predisposição na utilização de dispositivos móveis para empregar sistemas caracterizados como *smart*. O fato é justificado em consequência desses dispositivos conseguirem satisfazer algumas características encontradas nesse tipo de sistemas.

A conectividade constante é um dos atributos encontrados na computação móvel e que satisfaz a uma das características de sistemas *smart*, a interconectividade. A grande maioria dos dispositivos portáteis possui alguma forma de conectividade com a rede sem fio. As conexões variam entre *bluetooth*, Wi-fi, *General packet radio service* (GPRS).

¹ <http://phonegap.com/>

A ciência de contexto também pode ser destacado nessa ligação. Uma vez que é possível utilizar funcionalidades encontradas no dispositivo, como exemplo pode ser citado a câmera, para funcionar como uma forma de sensor e assim captar e reconhecer o ambiente.

Outro fator destacável na fusão entre computação móvel e sistemas *smart* é a mobilidade. Dispositivos móveis geralmente possuem uma infraestrutura reduzida, isto propicia o seu deslocamento, tornando possível a sua conexão com diversas redes existentes em variados ambientes.

2.2.3 Plataformas para Computação Móvel

No mercado de telefonia móvel existe diversos SOs que são utilizados para gerenciamento dos recursos dos dispositivos. Dentre eles pode-se citar: *Iphone OS*, *Android*, *Windows Phone*, *BlackBerry*, entre outros.

Dentre os sistemas operacionais citados, o único que possui o código totalmente aberto é o *Android*. O *Iphone OS* necessita de hardware específico (computadores que rodem o *mAC OS*) da *Apple* para ser possível desenvolver para o mesmo.

O SO da *Microsoft*, o *Windows Phone*, atualmente é um SO bem reconhecido e está cada vez mais ganhando mercado, entretanto, em questões de desempenho e utilização, o *Android* ainda está a sua frente. Esse fato pode ser visualizado pela diversidade de fabricantes que usam do SO *Android* em seus aparelhos.

O SO da *BlackBerry* é voltado ao público empresário. É um SO que se difundiu melhor nos Estados Unidos, mas é encontrado também no mercado brasileiro. Este SO necessita de um teclado específico para ser utilizado. Essa especialidade é fornecida nos aparelhos produzidos pela própria *BlackBerry*. O fato de necessitar ter um teclado específico já descartou a escolha desse SO para esse trabalho.

As pesquisas realizadas sobre a utilização dos SOs para dispositivos móveis, ([RASMUSSEN, 2013](#)) e ([TROYACK, 2013](#)), também contribuíram para a escolha da plataforma a ser utilizada no trabalho. Em consequência de todos os fatores citados, optou-se pelo SO *Android* para gerenciar o aplicativo desenvolvido.

2.2.4 SO Android

O *Android* inicialmente era um projeto mantido por uma *Startup* americana localizada no Vale do Silício chamada *Android Inc*. A empresa foi adquirida pelo grupo *Google* no ano de 2005, que aperfeiçoou o projeto e o tornou público no ano de 2007, com a finalidade de ser a primeira plataforma de código aberto (*open source*) de desenvolvimento para dispositivos móveis. O *Android* foi lançado e é mantido pela empresa *Open*

Handset Alliance (OHA)², que é um grupo formado por gigantes do mercado de telefonia de celulares sendo liderada pela Google³.

O SO da Google foi baseado no *kernel* 2.6 do Linux, o qual é responsável pelo gerenciamento de memória, processos, entradas e saídas, e sistemas instalados (LECHETA, 2009). Todas as aplicações que rodam no SO, quando abertas, disparam um novo processo.

Figura 3 – Arquitetura do Android.



Fonte: PEREIRA e SILVA (2009, p.5)

A arquitetura do SO Android pode ser visualizada na Figura 3; A camada do Núcleo do Linux (*Kernel*) atua como responsável pelo gerenciamento de energia, operando como uma camada de abstração entre o *hardware* e o resto da pilha de *software*. Acima do *kernel*, encontra-se as bibliotecas e o Android em si, ambos trabalham em paralelo. As bibliotecas do núcleo do Android contêm a maioria das funcionalidades disponíveis na linguagem de programação Java e algumas bibliotecas desenvolvidas em C e C++, ambas utilizadas em vários recursos do sistema.

² <http://www.openhandsetalliance.com>

³ www.google.com.br

O nível do ambiente de execução (*Android Runtime*) é uma instância da máquina virtual Dalvik, criada para a cada aplicação executada no sistema Android (PEREIRA; SILVA, 2009). Acima se encontra a camada arcabouço de aplicações, onde é possível encontrar todas as APIs e os recursos utilizados pelos aplicativos, como classes visuais nas quais incluem listas, caixas de textos, botões e até um navegador web embutido (PEREIRA; SILVA, 2009). Na extremidade da arquitetura em questão, localiza-se a camada de aplicativos, a qual contém todos os aplicativos fundamentais (escritos em Java) do Android.

A maioria das aplicações desenvolvidas para o Android são escritas na linguagem de programação Java. Entretanto, a máquina virtual utilizada para a execução não é a *Java Virtual Machine*(JVM). O Android utiliza outra máquina denominada *Dalvik* que otimiza a execução de aplicações em dispositivos móveis. Os aplicativos programados para o Android são escritos em Java e compilados para *bytecode* (.class), após são convertidos para o formato *Dalvik Executable* (.dex), que representa a aplicação do Android compilada. Concluindo a conversão, todos os arquivos .dex e outros recursos utilizados são compactados em um único arquivo denominado *Android Package File* (.apk), o qual representa a aplicação pronta para ser distribuída e instalada nos dispositivos que rodam a plataforma Android (LECHETA, 2009).

2.2.5 Ambiente de Desenvolvimento

Para a implementação de aplicações para a plataforma Android é necessário configurar um ambiente contendo de preferência a versão mais recente do *Java Development Kit* (JDK) juntamente com o ambiente de desenvolvimento do Android (SDK), o qual fornece as bibliotecas da API e ferramentas de desenvolvimento necessárias para construir, testar e depurar aplicativos para o Android (ANDROID, 2013b). Essas ferramentas incluem um depurador, bibliotecas, emulador de *Smartphone*, documentação, código de exemplo e tutoriais. O kit está disponível para os S.O *Windows*, *Linux* e *MacOS*, podendo ser baixado diretamente do site da *development* Android (ANDROID, 2013b).

Para a construção do aplicativo se optou pela utilização do *Integrated Development Environment* (IDE) *Eclipse*⁴ que juntamente com o *plugin* *Android Development Tools*(ADT) possibilita o desenvolvimento de aplicativos para o Android, pois o ADT permite criar projetos do tipo Android no ambiente de desenvolvimento *Eclipse* e testá-los no emulador encontrado no SDK de forma trivial (LECHETA, 2009).

⁴ Eclipse é uma plataforma tecnológica de propósito geral e pode ser utilizada na criação de IDEs para diferentes linguagens, além de poder ser utilizada na criação de IDEs personalizadas por muitos SDKs particulares (LECHETA, 2009)

2.3 Recuperação de Informação

Recuperação de informação ou *information Retrieval* (RI ou IR) é entendida como representação, armazenamento, organização e acesso a itens de informação (CARDOSO, 2000). Seu objetivo é proporcionar ao usuário uma forma fácil de acesso a informações de seu interesse.

Uma estrutura de dados bastante antiga que foi utilizada por muitos anos para a rápida RI são os índices. Esses contêm uma coleção de palavras selecionadas ou conceitos, os quais estão diretamente relacionados as informações relevantes para o contexto. Entretanto, com o advento dos computadores tornou-se possível a construção automatizada de índices.

Os índices construídos por computadores resultam em problemas de RI que são mais relacionados a sistema do que a necessidade do usuário. Sendo assim, existem duas visões para os problemas de RI. A primeira é relacionada aos computadores e a seguinte aos homens. O segundo problema não é abordado neste trabalho.

Os problemas encontrados decorrentes do computador em RI consistem principalmente na construção de índices eficientes, no processamento de consultas de usuários com desempenho otimizado e no desenvolvimento algoritmos de ordenação que tragam qualidade à resposta do usuário.

Nesse sentido, um sistema de RI em fase de indexação, constrói um índice que representa a coleção de documentos. Tal índice é constituído por unidades de indexação (descritores) e seus respectivos pesos. Esses pesos não foram adotados neste trabalho.

Na fase de busca, após o usuário formular a consulta, os descritores são pesquisados. Entretanto, para se fazer uma consulta é necessário que ocorra a seleção, normalização e o cálculo da representatividade dos termos e os relacionamentos entre eles.

Desconsiderar a variação linguística das palavras na seleção dos termos pode resultar em prejuízo de identificação de ocorrências desses termos na base de dados. Desse modo, sempre é uma boa prática tratar os textos antes dos mesmos serem analisados.

Em relação ao cálculo de representatividade, a indexação automática esta fundamentada na frequência de ocorrência das palavras nos documentos, (SALTON, 1986). Ambas as questões citadas tem sido tratadas por três abordagens clássicas: A booleana, a vetorial e a probabilística. As estratégias presentes nessas abordagens podem ser divididas em vários grupos, entretanto, para esse trabalho foi adotado o modelo com unigramas ⁵.

Nesse sentido, a abordagem com unigramas e bigramas ⁶ assumem como premissa que toda palavra encontrada no texto é estatisticamente independente de todas as outras

⁵ Representação de uma letra, palavra ou numero

⁶ Sequência de duas letras ou palavras ou números consecutivos.

palavras.

2.4 Considerações Sobre o Capítulo

Neste capítulo foram apresentados alguns exemplos de aplicações *smarts*, a visão geral da computação móvel, bem como suas perspectivas, relacionamentos com o conceito *smart*, plataforma e ambiente de desenvolvimento. Finalizou-se com uma breve introdução sobre o conteúdo de recuperação de informações.

Através do estudo realizado foi possível aprofundar os conhecimentos com a temática envolvida, além de, por intermédio dos exemplos, conseguir visualizar aplicativos que exploram o conceito desse trabalho sendo empregadas em diferentes contextos. Percebeu-se na [seção 2.3](#), que a temática de recuperação de informação é bastante extensa, possuindo diversos conceitos até então desconhecidos. Entretanto, se esses conceitos forem bem explorados, podem resultar em uma expressiva melhoria no produto final do trabalho.

3 Trabalhos Relacionados

Neste capítulo são apresentados trabalhos relacionados que tem como propósito tornar válido o tema dessa monografia. A [seção 3.1](#) apresenta a metodologia utilizada para a realização da revisão literal. Na [seção 3.2](#) é apresentada os resultados obtidos, citando vantagens e desvantagens dos trabalhos selecionados. Na [seção 3.3](#) é exibida uma análise sob os trabalhos. Por fim, na [seção 3.4](#) é efetuado o fechamento do capítulo.

3.1 Processo de Revisão Sistemática

No intuito de entender o estado da arte sobre o uso das aplicações *smart*, realizou-se uma avaliação sistemática que tem por objetivo a busca de trabalhos que atendam o conceito *smart* definido na [seção 2.1](#). A busca foi realizada nas bases *IEEE Explorer* ¹ e *Elsevier* ². As palavras chaves utilizadas para a realização da busca foram: *Smart Application* e *university*, *Smart Application* e *college*, *Smart Application* e *education*. A partir das bases mencionadas foram extraídos vários artigos, entretanto, foram selecionados apenas os que possuíam mínimo de seis páginas, em formato pdf, disponíveis para download, com resultados, com data de publicação inferior a 5 anos, e a solução desenvolvida no trabalho respeitar a definição *smart* proposta na [seção 2.1](#).

3.2 Resultados da Revisão

O resultado da revisão foi organizado em dois grupos. No primeiro grupo se encontram os trabalhos que contribuem com melhorias relacionadas diretamente ao aprendizado do aluno. No segundo grupo se encontram os trabalhos que visam o aprimoramento de serviços disponíveis ao público universitário.

3.2.1 Trabalhos Relacionados ao Aprendizado do Aluno

Em [Schon, Kopf e Effelsberg \(2012\)](#) foi proposto um ambiente virtual para que professores possam lançar perguntas aos discentes e estes responderem através de seu *Smartphone*, automatizando o processo de aplicações de *Quiz* de perguntas aos alunos. Para tal, dividiu-se a solução em 2 etapas: Um *backend* fornece a visão para o professor, e um *frontend* para os participantes do *Quiz*. O *backend* é implementado como um *plugin* para o sistema *e-learning* da universidade, o *frontend* foi construído com as tecnologias web existentes entre estas, o *jquery*. Para simplificação do acesso ao formulário, foi utilizado

¹ <http://ieeexplore.ieee.org/>

² <http://www.elsevier.com/>

um *QRCode* dinâmico que exibe um *link* no quadro para que os participantes possam acessar o *Quiz* de maneira mais rápida. Na conclusão foi constatado que o *Mobile Quiz* é eficiente para sua finalidade, entretanto, para realizar parte de suas funcionalidades é necessário a implementação de *HTML 5* nos navegadores *mobile*. A funcionalidade *smart* deste trabalho consiste na organização das respostas obtidas através do *quiz* de perguntas, gerando gráficos que tendem a facilitar a compreensão da pesquisa pelo professor.

Em Wang e Wu (2011) foi desenvolvido um sistema *e-learning* para ajudar os alunos em seu aprendizado de forma sensível ao contexto. O aluno se conecta à internet e através do sistema de identificação por radiofrequência (RFID), pode aplicá-lo para detectar a localização de objetos no mundo real. Após a detecção, o sistema transmite os dados do objeto e associa-os ao seu utilizador. O módulo de recomendação de material didático também pode fornecer material adaptativo com base no comportamento de aprendizagem do aluno e preferências pessoais. Os testes demonstraram que havendo sensibilidade de contexto e recomendação didática fez-se diferença significativa para a precisão de assuntos ao responder as perguntas de tarefas de aprendizagem. Indivíduos com apoio à aprendizagem, concluíram suas tarefas de aprendizagem em menos tempo e acertaram mais respostas. A contribuição *smart* deste trabalho consiste na funcionalidade que gera perfis de usuário com base nos objetos detectados pelos mesmo, dessa forma o sistema consegue recomendar materiais ao utilizador com maior precisão.

Em Brus et al. (2010) foi elaborado uma aplicação que pode criar mapa automaticamente ou com atividades interativas do usuário. A pretensão do trabalho é eliminar o problema da insuficiência de conhecimento de regras cartográficas para elaboração de diferentes tipos de mapas. No processo de construção de um sistema *smart* em cartografia é necessário construir uma base de conhecimento, que incluirá termos e normas cartográficas. Para tal foi utilizado ontologias, podendo assim compartilhar o conhecimento cartográfico comum e tornar as regras cartográficas explícitas e reutilizáveis. Foi utilizado o algoritmo proposto pelo *Czech Science Foundation*, tendo como pré-requisito um ambiente de internet/intranet. Necessita-se de um sistema esperto como base, tendo sido escolhido o *Drools*. Este integra a lógica do negócio. O experimento resultou no aumento da qualidade do mapa e das habilidades do usuário. Aplicação *smart* vai trazer uma melhor utilização de mapa de decisões, principalmente com o software GIS. A contribuição *smart* desse trabalho consiste na funcionalidade que valida o mapa criado para o usuário através da utilização do sistema *Drools*. Dessa forma os mapas se tornam mais precisos e evita que o usuário gere na maioria dos casos mapas equivocados.

3.2.2 Trabalhos Relacionados aos Serviços Universitários

Em (KBAR; ALY; ALHANAFI, 2012) foi projetado uma aplicação que permite aos usuários realizar pesquisas no campus da universidade com base na aplicação intitulada

university Geographic Information System (GIS) e a utilização da *wi-fi*, aprimorando a obtenção de informações referentes aos pesquisadores e seus departamentos em uma universidade. Esta aplicação se respalda em três opções de busca; A pesquisa baseada em palavras é realizada sob um banco de dados normal; A busca baseada em localização identifica automaticamente o local da busca com base no *Internet Protocol* (IP) da rede e por fim, a busca baseada em perfil que monta dinamicamente o perfil de cada usuário levando em consideração as pesquisas realizadas nas outras modalidades anteriormente. Após a seleção do modo de pesquisa, a aplicação utiliza de um algoritmo de busca que leva como premissa a escolha feita pelo usuário sobre a pesquisa desejada e após exibe o resultado da mesma. Esse aplicativo resultou em uma busca flexível na universidade e otimizou o ambiente de trabalho dos pesquisadores, estudantes e gestores da universidade. A contribuição *smart* desse trabalho consiste na funcionalidade que gera o perfil do usuário com base nas pesquisas realizadas pelo mesmo anteriormente, conseguindo assim inferir pesquisas para o usuário de maneira que o surpreenda-o.

Em Geng e Cassandras (2011) foi sugerido um sistema *smart* que otimiza o uso dos estacionamentos através da internet. Este embasa-se em coletar as informações do estacionamento e enviar para um servidor na internet, a onde é possível reservar vagas para posterior utilização. A ideia principal do sistema é a comodidade que ele oferece, visto que informando o local de destino e o preço limite que se deseja pagar, este procura a melhor vaga disponível para a alocação de seu veículo. Os testes foram realizados no estacionamento do campus universitário da cidade de Boston, onde foi atestado que as vagas mais distantes e baratas em termos financeiro foram preenchidas rapidamente. Fina-se que a utilização do sistema trouxe melhorias significativas na utilização do estacionamento, todavia, é necessário rever algumas melhorias para aprimorar o seu funcionamento. A contribuição *smart* desse trabalho consiste na funcionalidade de ficar buscando por novas vagas de estacionamento mesmo após o sistema ter alocado o usuário a uma vaga. Caso seja disponibilizada uma vaga que se adeque melhor a sua necessidade e o usuário ainda não tenha ocupado a vaga que foi estabelecida ao mesmo anteriormente, o sistema notifica o usuário para que ele ocupe a nova vaga disponível.

3.3 Análise dos Resultados

Os trabalhos (SCHON; KOPF; EFFELBERG, 2012), (WANG; WU, 2011) e (BRUS et al., 2010) são focados diretamente na melhoria da aprendizagem do aluno. Esses possuem como contribuição *smart* respectivamente a organização das respostas dos alunos em gráficos melhorando a visualização destes sob a visão do professor (SCHON; KOPF; EFFELBERG, 2012), a sugestão de materiais específicos ao perfil do aluno com base em objetos identificados por ele anteriormente (WANG; WU, 2011) e a melhoria na criação de mapas com a utilização de ontologias e algoritmos que sugerem formas alternativas

para tal função (BRUS et al., 2010). Esses trabalhos têm como características comuns: o aprimoramento de serviços que favorecem o aluno, a utilização de dispositivos móveis e a necessidade de conexão com a internet para seu funcionamento.

Trabalhos de (GENG; CASSANDRAS, 2011), (KBAR; ALY; ALHANAFI, 2012) se concentram em aperfeiçoar os serviços oferecidos ao público universitário, detendo como colaboração *smart* respectivamente a sugestão de vagas de estacionamento com base na sua localização atual, valor máximo pagável e ponto final de chegada (GENG; CASSANDRAS, 2011), realização de pesquisas departamental dentro da universidade com base no perfil do utilizador (KBAR; ALY; ALHANAFI, 2012). Estes tem como pontos comuns a melhoria de tarefas que são utilizadas pelos frequentadores da universidade e a utilização de servidor para colaborar com as aplicações.

Todos os trabalhos citados possuem uma relação com esse trabalho pois foram desenvolvidos para a comunidade acadêmica. Entretanto, os trabalhos que focam na melhoria de serviços prestados pela universidade possuem mais afinidade com esse trabalho, em razão desses deterem o mesmo objetivo. Percebe-se que esse grupo de trabalhos tem como propósito a melhoria de serviços providos a discentes das instituições. Para conseguir os resultados esperados, todos os trabalhos utilizam da infraestrutura disponibilizada pela universidade. Desses, alguns necessitaram da utilização de dispositivos móveis para conceder mobilidade a seus usuários.

3.4 Considerações Sobre o Capítulo

O capítulo em questão proporcionou o estudo de trabalhos que objetivam a melhoria de processos no contexto acadêmico. Os trabalhos escolhidos foram separados em dois grupos e através de uma análise foi possível perceber que existe uma predisposição na utilização de dispositivos móveis para a automatização de processos existentes nas universidades, além de todos os trabalhos usufruírem a infraestrutura da instituição. Por intermédio da análise, ficou decidido a utilização de dispositivos portáteis para a realização do presente trabalho.

4 Processo de Desenvolvimento do Aplicativo

O propósito geral do capítulo é demonstrar os passos percorridos para o desenvolvimento do aplicativo SmartACG. Na [seção 4.1](#) é feita a retomada do problema, enumerando os motivos que vieram a ocasionar o desenvolvimento desse trabalho, além de sua finalidade, o porquê de ser intitulado *smart* e também é apresentada uma figura geral que demonstra a visão da solução. A [seção 4.2](#) contém o estudo da viabilidade do aplicativo, requisitos a serem desenvolvidos, diagrama de casos de uso e casos de uso detalhados. A [seção 4.3](#) possui diagrama de pacotes, diagrama de classes, diversos diagramas de sequência e diagrama de entidade-relacionamento. Na [seção 4.4](#) é constatado todas as decisões de projeto tomadas, além de demonstração de *print screens* de todas as telas existentes no aplicativo. Por fim, na [seção 4.5](#) é feito o fechamento do capítulo.

4.1 Concepção

Na UNIPAMPA - Campus Alegrete, a maioria dos cursos requer trezentas horas de ACGs como parte dos requisitos para obtenção do diploma de graduação. Essas são classificadas em modalidades, conforme [Apêndice B](#). Hoje existem quatro modalidades: Atividades de Ensino, Atividades de Pesquisa, Atividades de Extensão e Atividades Culturais e de Gestão, podendo existir variações em alguns cursos.

Atualmente o processo de validação das atividades na UNIPAMPA - Campus Alegrete é feito todo de forma manual. O aluno após realizar a atividade, precisa preencher o formulário de solicitação para posteriormente ocorrer a validação do seu pedido. É fundamental que o aluno através do formulário classifique a atividade realizada, dispondo como subsídio para a realização desse processo, a tabela de ACGs de seu respectivo curso. Posterior a classificação, é necessário que o aluno anexe os documentos que comprovem a realização da atividade.

A etapa final do processo de aproveitamento de ACGs, por parte do aluno, ocorre quando o mesmo encaminha a documentação referente à todas as ACGs realizadas à secretária acadêmica do campus. Entretanto, é necessário que o aluno aguarde pelo prazo de realização dessa atividade, o qual é aberto uma vez por semestre e definido sempre no calendário acadêmico da instituição.

4.1.1 Finalidade do Aplicativo

A finalidade do aplicativo é facilitar o cotidiano dos alunos, através do auxílio na coleta e classificação de suas ACGs. Agregado a finalidade principal, pretende-se iniciar uma caminhada em busca da informatização do processo atual de aproveitamento de ACGs, realizado hoje de forma manual.

4.1.2 Por que *smart*?

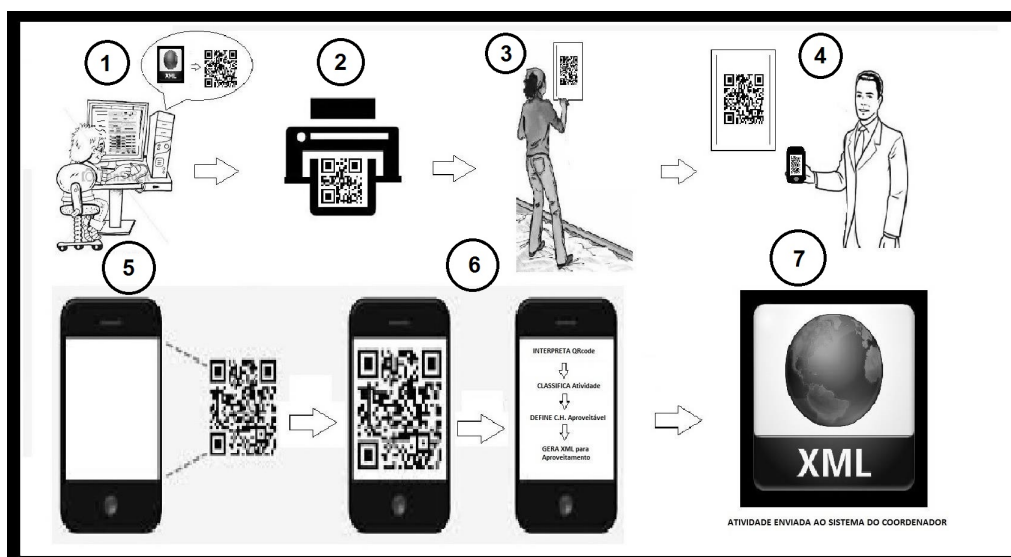
A universidade é um local propício para inovações tecnológicas, pois possui um público tecnologicamente ativo que busca através da diversidade de meios eletrônicos maneiras de facilitar o seu cotidiano. Esse cenário se torna promissor para a aplicabilidade do conceito *smart*, o qual tem em sua essência surpreender o usuário oferecendo soluções inteligentes.

Respaldado no processo de aproveitamento de ACGs, elaborou-se uma solução que viabiliza ao usuário coletar as informações da atividade realizada no seu próprio local de realização. Posteriormente, o aplicativo se encarrega de realizar o processamento necessário para classificar corretamente a atividade realizada.

4.1.3 Visão Geral da Solução

Na [Figura 4](#) é apresentada a visão geral da solução proposta. Seu funcionamento ocorre da seguinte maneira:

Figura 4 – Visão Geral da Solução.



1. Assume-se a premissa de que as atividades são descritas pelo *promoter*¹ através de uma estrutura *XML* e codificadas em *QRCode*. Em razão disso, os três primeiros passos não serão exemplificados.
2. O escopo do trabalho inicia a partir do momento que o usuário lê o *QRCode*, perceptível na [Figura 4](#) através do passo 4.
3. Após, o aplicativo decodifica o *QRCode*, resultando em um *XML*. O aplicativo interpreta as informações do *XML*, notificando o usuário assim que concluído essa etapa.
4. É armazenado na base de dados do software todas as informações providas do *XML* interpretado, Caso a interpretação tenha sido executada com êxito e o *XML* contenha todas as informações necessárias para o aplicativo prosseguir com o seu processamento.
5. A partir da persistência no banco, o aplicativo desencadeia uma série de processamentos a fim de inferir uma classificação para a atividade realizada.
6. O último passo realizado pelo aplicativo é compilar em um *XML* as informações do aluno com as da atividade realizada pelo mesmo, para que posteriormente ocorra o envio desses dados ao coordenador. Entretanto, em virtude de atualmente ainda não existir o sistema que receberá o *XML* em questão, o processo se encerra com a geração do *XML* pelo aplicativo.

4.2 Análise

Para definir o escopo do aplicativo desenvolvimento no trabalho, foi realizada uma pesquisa de opinião com os alunos da UNIPAMPA - Campus Alegrete. As respostas podem ser visualizadas no [Apêndice A](#). A pesquisa se justifica em razão da necessidade de se definir um escopo de funcionalidades que atendam total ou parcialmente as solicitações observadas através das respostas obtidas.

A pesquisa realizada ficou disponível durante uma semana, entre os dias 02/09/2013 à 09/09/2013, alcançando ao seu final 41 respostas, as quais podem ser visualizadas na [Apêndice B](#). Analisando as respostas obtidas, pode-se aproveitar 83%, pois o restante das respostas fugiam do escopo do trabalho. Houve uma disparidade significativa nas respostas obtidas, o que é normal em razão do formulário ter sido preenchido por alunos de diferentes cursos.

¹ Entende-se como *promoter* neste trabalho qualquer indivíduo capaz de propor uma atividade que se enquadre na tabela de ACGs de um dos respectivos cursos da UNIPAMPA.

Em suma, o objetivo principal da pesquisa foi descobrir as necessidades encontradas atualmente pelos discentes no momento de aproveitamento de suas ACGs. Com base nas respostas obtidas, foram escritas *user stories* (US) ² que podem ser visualizadas a seguir:

- a) O aplicativo deverá ser capaz de ler as informações da atividade no ato de sua realização, para que dessa forma consiga guardar as informações necessárias para simplificar o processo atual de aproveitamento da ACG.
- b) O aplicativo deverá ser capaz de classificar a atividade coletada, disponibilizando ao usuário a modalidade da ACG realizada e o tempo máximo de horas aproveitáveis nesta durante o semestre vigente.
- c) O aplicativo deverá ser capaz de informar ao usuário os documentos necessários para comprovação da atividade realizada.
- d) O aplicativo deverá ser capaz de após a classificação da ACG, gerar um produto final que traga melhorias para o aluno, simplificando o processo atual.

As US foram escritas com base em padrões de comportamento identificados nos resultados obtidos pela pesquisa. Sendo assim, uma US pode contemplar inúmeras respostas. Respalhando-se nas US descritas, foi constatado que para o aplicativo funcionar de forma correta, este precisa conter além das US já citadas outra funcionalidade, sendo-a:

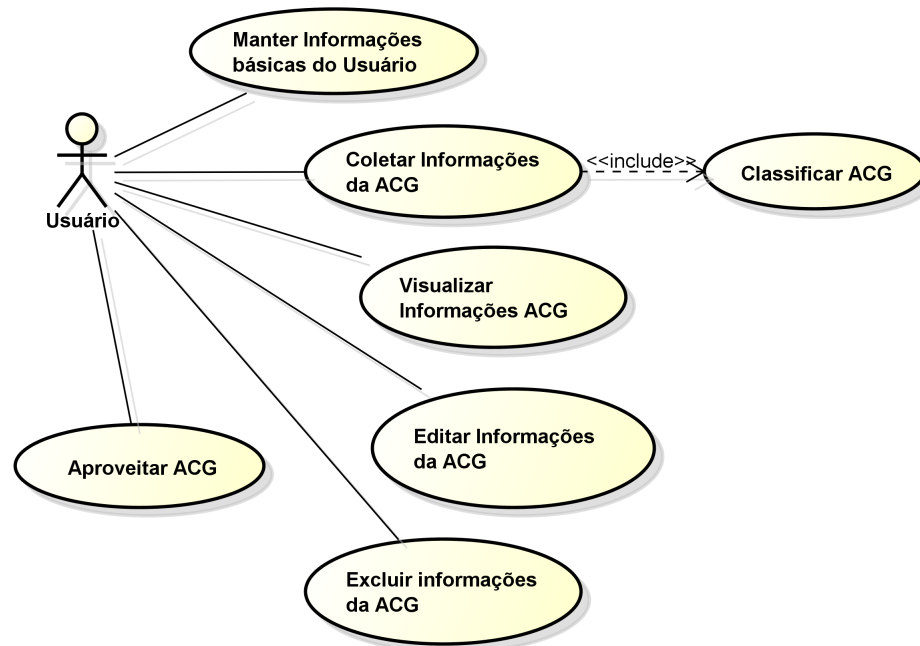
- a) O aplicativo deverá ser capaz de cadastrar as informações básicas do usuário em seu primeiro uso, em razão de que para emitir a saída de seu processamento, é necessário existir algumas informações pessoais deste.

Considerando que a etapa mais longa do processo de aproveitamento de ACGs é a sua classificação, explorou-se o conceito *smart*, referenciado no trabalho, para o desenvolvimento dessa funcionalidade. Desse modo, após o usuário coletar as informações do ambiente, a sensação para o usuário será de que o aplicativo apenas coletou seus dados. Porém, após coletá-los o programa desencadeia automaticamente as funções de classificar e definir carga horária passível de aproveitamento, fazendo com que os dados cruzados ofereçam uma solução completa ao usuário, sem envolvê-lo no processo de classificação.

4.2.1 Casos de Uso

Para uma melhor compreensão e visualização das funcionalidades que foram automatizadas através do aplicativo, desenhou-se um diagrama de casos de uso que pode ser visualizado na [Figura 5](#). Os casos de uso demonstram as interações que o usuário necessita fazer com o sistema, em consequência disso a funcionalidade que explora o conceito

Figura 5 – Diagrama de Casos de Uso.



smart referenciado neste trabalho não aparece ligada diretamente ao usuário.

Os cenários dos casos de uso foram descritos no intuito de estabelecer as interações entre ator e sistema. O primeiro caso de uso detalhado é o manter informações básicas do usuário, visualizado na [Tabela 1](#). O caso de uso em questão ocorre na primeira interação do usuário com o aplicativo e tem como intuito coletar algumas informações pessoais que serão utilizadas posteriormente por outra funcionalidade.

O próximo caso de uso detalhado é o de coletar as informações da ACG, o qual pode ser visualizado na [Tabela 2](#). O caso de uso em evidência é acionado preferencialmente no momento que o discente está realizando a atividade.

Após coletar as informações da ACG, é acionada a chamada Classificar ACG. A mesma é um caso de uso do tipo *include* por ser executado sempre ao término do caso de uso anterior. O caso de uso em questão é mostrado na [Tabela 3](#). A funcionalidade referenciada a partir do caso de uso explora o conceito *smart*.

Sucessivamente a classificação da ACG, o usuário pode visualizá-la através da funcionalidade de Visualizar informações da ACG, vista em detalhes na [Tabela 4](#). Caso o usuário sinta a necessidade, pode editar a ACG realizada por meio da funcionalidade editar informações da ACG, detalhada na [Tabela 5](#).

Na hipótese do usuário necessitar remover a atividade realizada do aplicativo,

² *User Story* é uma história contada pelo usuário que através de uma ou mais frases na linguagem cotidiana consegue contar o que ele faz ou precisa fazer, como parte de sua função.

Tabela 1 – Caso de uso detalhado - Manter informações básicas do usuário.

Item	Valor
Caso de Uso:	Manter informações básicas do usuário.
Ator:	Usuário.
Fluxo Principal:	<p>1 - O caso de uso inicia quando o usuário abre o aplicativo pela primeira vez.</p> <p>2 - O usuário seleciona a funcionalidade de se cadastrar.</p> <p>3 - O sistema solicita que o usuário informe seu nome, curso e matrícula para prosseguir.</p> <p>4 - O usuário entra com o nome, curso e matrícula.</p> <p>5 - O sistema armazena os dados em sua base de dados e o caso de uso é encerrado.</p>
Fluxo Alternativo:	5 - Caso o sistema por algum motivo não consiga guardar as informações de cadastro do usuário, o mesmo notifica o usuário e o caso de uso é encerrado.

Tabela 2 – Caso de uso detalhado - Coletar informações da ACG.

Item	Valor
Caso de Uso:	Coletar informações da ACG.
Ator:	Usuário.
Pré-Condições:	Usuário já ter se cadastrado no sistema.
Fluxo Principal:	<p>1 - O caso de uso é iniciado quando o usuário seleciona a funcionalidade que coleta as informações da atividade.</p> <p>2 - O usuário coleta as informações da atividade.</p> <p>3 - O sistema captura essas informações.</p> <p>4 - O sistema avisa para o usuário que a coleta das informações da atividade ocorreu com sucesso.</p> <p>5 - O sistema inicia o processamento para conseguir inferir uma classificação para a atividade coletada.</p> <p>6 - O caso de uso é encerrado.</p>
Fluxo Alternativo:	4 - Caso o sistema por algum motivo não consiga guardar as informações da atividade coletada, este avisa o usuário e o caso de uso é encerrado.

Tabela 3 – Caso de uso detalhado - Classificar ACG.

Item	Valor
Caso de Uso:	Classificar ACG.
Ator:	Sistema
Pré-Condições:	Usuário já ter se cadastrado no sistema.
Fluxo Principal:	<p>1 - Após as informações serem coletadas pelo aplicativo, este caso de uso é iniciado.</p> <p>2 - O aplicativo processa a atividade coletada.</p> <p>3 - O aplicativo classifica a atividade realizada em alguma das atividades da tabela de ACGs do aluno, usuário do mesmo. Após alimenta sua base de conhecimento.</p> <p>4 - O aplicativo calcula a carga horária plausível de aproveitamento com base na classificação da atividade.</p> <p>5 - O sistema guarda a ACG classificada na sua base de dados.</p> <p>6 - O caso de uso é encerrado.</p>
Fluxo Alternativo:	<p>3 - Se o sistema não conseguiu classificar a ACG, o caso de uso é encerrado.</p> <p>4 - Se o sistema não conseguir salvar em sua base de dados a ACG classificada, ele emite um aviso ao usuário que deu erro em gravar a ACG realizada.</p>

Tabela 4 – Caso de uso detalhado - Visualizar informações da ACG.

Item	Valor
Caso de Uso:	Visualizar informações da ACG.
Ator:	Usuário.
Pré-Condições:	Usuário já ter se cadastrado no sistema.
Fluxo Principal:	<p>1 - Após o usuário selecionar a ACG que deseja visualizar o caso de uso é iniciado.</p> <p>2 - O usuário visualiza as informações da ACG realizada.</p> <p>3 - O caso de uso é encerrado.</p>
Fluxo Alternativo:	2 - Caso o sistema não consiga encontrar as informações da ACG selecionada pelo usuário, o sistema notifica o usuário que ocorreu erro na visualização da atividade e o caso de uso é encerrado.

Tabela 5 – Caso de uso detalhado - Editar informações da ACG.

Item	Valor
Caso de Uso:	Editar informações da ACG.
Ator:	Usuário.
Pré-Condições:	Usuário já ter se cadastrado no sistema.
Fluxo Principal:	1 - Após o usuário selecionar a ACG que deseja alterar e a opção de editá-la, o caso de uso é iniciado. 2 - O usuário visualiza as informações da ACG realizada. 3 - O usuário altera o local de realização da ACG. 4 - O usuário aciona a funcionalidade de salvar a alteração. 5 - O sistema guarda na sua base de dados a alteração realizada pelo usuário. 6 - O sistema notifica o usuário que a alteração ocorreu com sucesso. 7 - O caso de uso é encerrado.
Fluxo Alternativo:	5 - Se o sistema não conseguir armazenar em sua base de dados a alteração realizada, o mesmo notifica o usuário que deu erro na alteração realizada e o sistema retorna ao passo 2.

o procedimento pode ser feito por intermédio da funcionalidade excluir informações da ACG. O caso de uso da atividade citada pode ser visualizado na [Tabela 6](#).

Para concluir o ciclo de funcionalidades do sistema, o usuário deve solicitar a execução da funcionalidade Aproveitar ACG, disponível em detalhes na [Tabela 7](#). A funcionalidade em questão compila todos os dados da atividade juntamente com os dados do aluno e os envia para o servidor que, posteriormente, deve se comunicar com o sistema de ACGs da coordenação, inexistente até o momento.

4.3 Projeto

Nessa etapa do desenvolvimento do aplicativo, fez-se a elaboração de diagramas da *Unified Modeling Language* (UML). Esses, posteriormente, ajudaram o programador a desenvolver o aplicativo, além de serem fontes de documentação essenciais para futuras manutenções no software. Segundo (BRAUDE, 2005), o projeto de uma aplicação específica como ela deve ser construída, descrevendo as partes envolvidas e a forma como devem ser montadas. O referido autor complementa sua afirmação alegando que um projeto é construído a partir de requisitos, sendo assim, uma boa análise é essencial para a elaboração de um projeto de software eficaz.

Tabela 6 – Caso de uso detalhado - Excluir informações da ACG.

Caso de Uso:	Excluir informações da ACG
Ator:	Usuário
Pré-Condições:	Usuário já ter se cadastrado no sistema.
Fluxo Principal:	<p>1 - Após o usuário selecionar a ACG que deseja alterar e a opção de editá-la o caso de uso é iniciado.</p> <p>2 - O usuário visualiza as informações da ACG realizada.</p> <p>3 - O usuário seleciona a funcionalidade de excluir a ACG realizada.</p> <p>4 - O usuário confirma a exclusão da ACG selecionada.</p> <p>5 - O sistema exclui a ACG selecionada da sua base de dados e informa ao usuário que a exclusão ocorreu com sucesso.</p> <p>6 - O caso de uso é encerrado.</p>
Fluxo Alternativo:	<p>4 - Se o usuário não confirmar a exclusão da ACG realizada o sistema retorna ao passo 2.</p> <p>5 - Se o sistema não conseguir eliminar a ACG selecionada para exclusão, o sistema notifica o usuário que ocorreu um erro e retorna ao passo 2.</p>

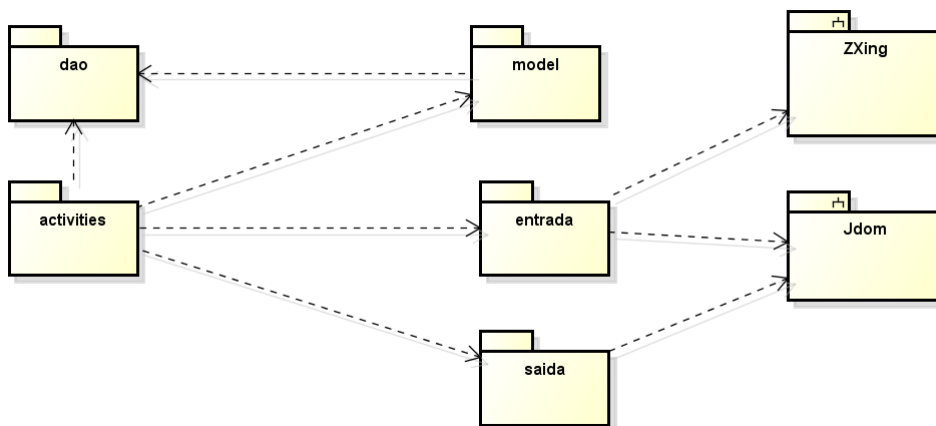
Tabela 7 – Caso de uso detalhado - Aproveitar ACG.

Item	Valor
Caso de Uso:	Aproveitar ACG.
Ator:	Usuário.
Pré-Condições:	Usuário já ter se cadastrado no sistema.
Fluxo Principal:	<p>1 - Após o usuário selecionar a ACG que deseja visualizar, o caso de uso é iniciado.</p> <p>2 - O usuário visualiza as informações da ACG realizada.</p> <p>3 - O usuário seleciona a funcionalidade de aproveitar a ACG realizada.</p> <p>4 - O sistema notifica o usuário que a ACG foi enviada para aproveitamento e o caso de uso é encerrado.</p>
Fluxo Alternativo:	4 - Caso o sistema por algum motivo não consiga executar o passo de número 4, este notifica o usuário que não foi possível a execução e retorna ao passo de número 2.

4.3.1 Diagrama de Pacotes

O primeiro diagrama a ser apresentado é o de pacotes. Esse descreve os pacotes ou partes do sistema dividido em seqüências lógicas, além de exibir as dependências existentes entre os pacotes e subsistemas. A finalidade básica desse diagrama é exibir a arquitetura de um sistema e o agrupamento de suas respectivas classes. Deste modo, um pacote representa um conjunto de classes. Esses pacotes se relacionam com outros através de relações do tipo "dependência". O diagrama de pacotes do aplicativo em questão é visualizado na [Figura 6](#).

Figura 6 – Diagrama de Pacotes - SmartACG.



A arquitetura é organizada em cinco pacotes. O pacote *activities* abrange todas as classes do tipo *Activity*. A arquitetura do Android isola as telas estáticas dentro de arquivos *XMLs*, delegando para as classes *activities* somente tarefas que envolvam a manipulação de componentes gráficos dinâmicos e interação com os demais pacotes.

O pacote *entrada* engloba todas as classes que possuem as funcionalidades de leitura e interpretação do *QRCode*. Esse pacote depende de um subsistema chamado *ZXing*. O subsistema citado é responsável pela leitura do *QRCode*. Além deste subsistema, o pacote *entrada* necessita do subsistema chamado *Jdom*, o qual é responsável pela manipulação do *XML* codificado pelo *QRCode*. No presente pacote a função do *Jdom* é interpretar as *tags* encontradas no *XML* para que as demais classes consigam utilizar das informações contidas no arquivo.

O pacote *model* contém todas as classes que representam as entidades presentes no escopo da aplicação e as demais classes que possuem regras de negócio ou servem como auxílio para execução de algumas funcionalidades.

Dentro do pacote *saída*, estão todas as classes que oportunizam algum meio de saída para funcionalidades do sistema. Este pacote depende do subsistema *Jdom* para gerar *XMLs* que podem ser enviados a um outro sistema.

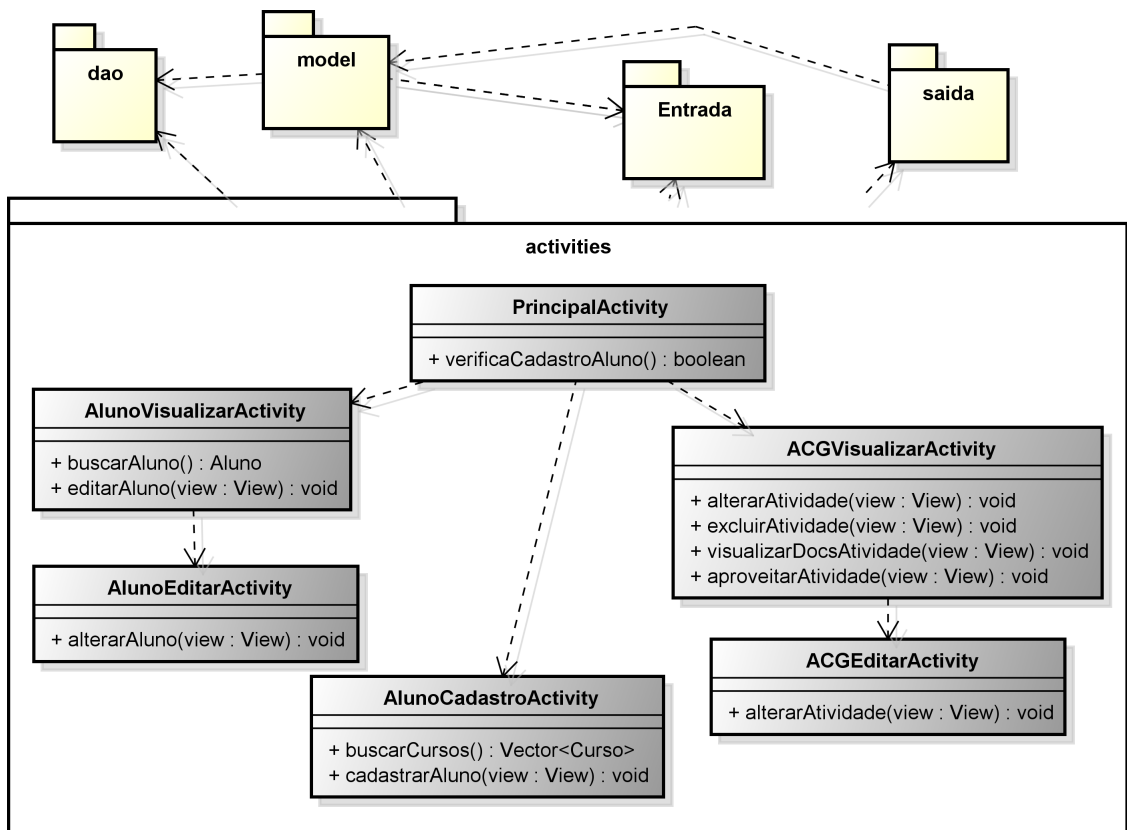
Por último é visualizado o pacote DAO, o qual agrupa todas as classes que manipulam dados da aplicação no banco de dados.

4.3.2 Diagrama de Classes

Conforme (O'NEILL, 2004), o diagrama de classes é uma das técnicas mais utilizadas no desenvolvimento orientado a objetos, pois representa uma descrição formal dos objetos de um sistema. Esse diagrama descreve para cada classe a sua identidade, os seus relacionamentos com outras classes, seus atributos e suas operações.

A primeira parte explicada é a fração que contém o pacote das *activities*, visível na Figura 7. No diagrama, as classes do tipo *activity* foram destacadas com um tom acinzentado. Fundamentalmente, cada classe desse pacote gerencia uma tela do sistema, conseqüentemente todas estão interligadas a algum *XML* representante de alguma tela.

Figura 7 – Diagrama de Classes - Activities.



A classe *PrincipalActivity* é a principal do sistema, por possuir o menu geral deste e funções que chamam as demais *activities* do pacote. Por esse motivo, ela é ligada as demais *activities* por meio de uma relação de dependência. O método *verificaCadastroAluno* tem a responsabilidade de verificar se o aluno já realizou seu cadastro no aplicativo. Ele é acionado sempre que o aplicativo é iniciado. O método é justificado em razão do usuário

não poder usufruir das demais funcionalidades presentes no aplicativo caso não tenha cadastro no mesmo.

A classe *AlunoVisualizarActivity* é relacionada a tela de visualizar as informações do aluno. Esta possui o método *buscarAluno* que consiste em trazer as informações básicas do aluno cadastrado no banco para exibição. Esse método retorna somente um aluno, devido a se assumir que o aplicativo terá somente um usuário, posto que habitualmente cada pessoa detém de um dispositivo móvel particular.

A classe *ACGVisualizarActivity* é responsável pelo gerenciamento da tela de visualização das informações de ACGs realizadas. Ela possui o método *alterarAtividade* que chama a *activity* responsável por essa função. O método *excluirAtividade* tem como papel acionar a classe responsável por executar a exclusão da atividade presente no banco de dados. O método *visualizarDocsAtividades* tem por finalidade exibir uma lista de documentos necessários para comprovar a realização da atividade em questão. Por fim, o método *aproveitarACG* aciona as classes que compilam as informações da atividade realizada com os dados do aluno para, posteriormente, ocorrer o envio dos mesmos ao coordenador.

As classes *AlunoEditarActivity* e *ACGEditarActivity* são relacionadas a tela de editar informações do aluno e editar informações da ACG respectivamente. Ambas contém o método *alterar* que tem como função coletar as novas informações do aluno ou ACG em questão e acionar classes executoras da alteração dessas informações na base de dados.

Por fim existe a classe *AlunoCadastroActivity* que é relacionada a tela de cadastro do aluno. Essa classe mantém o método de *buscarCursos* que possui como retorno um vetor de cursos, além do método *cadastrar* que é utilizado para coletar as informações da tela e solicitar a classe responsável por persistir esses dados no banco.

A próxima fração do diagrama de classes expandido explica o pacote *model*. O diagrama reduzido em questão é visualizado na [Figura 8](#). O pacote *model* possui as classes de domínio da aplicação, como por exemplo: *Alunos*, *Atividade*, *Curso*, *TipoAtividade*, entre outras.

De acordo com o artigo 104 das normas acadêmicas disponíveis no [Apêndice B](#), todas as atividades complementares de graduação são classificadas em grupos, os quais são representados pela classe *Modalidade* no diagrama em questão. A classe *TipoAtividade* representa as atividades elucidadas nos artigos 106, 107, 108 e 109 da mesma norma.

Segundo a mesma norma referida anteriormente, no artigo 111 é compreendido que todas as atividades necessitam de cópias de documentos comprobatórios. Em conformidade a essa exigência, modelou-se a classe *Documento*. Por intermédio da tabela de Cômputo de Horas de ACGs do curso de Engenharia de software, a qual está disponível no [Apêndice A](#), pode-se verificar que existe uma coluna chamada observações. A

coluna em questão é representada através da classe *Restrição*. Conseguiu-se por meio dessa modelagem não replicar os objetos, visto que segundo o diagrama é possível que um *TipoAtividade* tenha relacionamento com 1 ou N restrições.

A tabela de ACGs dos cursos estão representadas através das classes *Documento*, *Modalidade*, *Restrição*, sendo que todas estão ligadas à classe *TipoAtividade* que finda a representação. Esta classe por sua vez possui ligação com a classe *Curso*, garantindo que cada curso de graduação possa ter uma tabela de ACGs distinta.

A classe *Atividade* representa uma atividade realizada pelo aluno. Em um primeiro momento ela pode não ter uma relação com a *TipoAtividade*, além de nenhuma atividade pode conter mais de um *TipoAtividade*. A afirmação é garantida por meio do relacionamento 0 ou 1 presente entre as duas classes. Além dessa relação, a classe *Atividade* é relacionada também à classe *BaseConhecimento*. Essa classe representa um conjunto de duas palavras relacionadas a uma atividade, as quais são utilizadas para inferenciar a classificação de novas atividades introduzidas no sistema. Detalhes de funcionamento dessa inferência são demonstrados na [seção 4.4](#).

O aluno é representado na classe *Aluno*. Em conformidade com as especificações, o aluno é obrigado a ter um pré cadastro e se relacionar a um curso. Isso se deve ao fato de que o aplicativo necessita saber qual tabela de ACGs realizará consultas para classificar as novas ACGs ingressadas no sistema.

A classe *Processador* é fundamental para o êxito do sistema. Seu funcionamento acontece através do recebimento da atividade realizada vinda do pacote *entrada*, utilizando-se do padrão de projeto *Observer*³. Após, é acionado o método *processarAtividade* que utiliza da classe *Classificador* para tentar inferir uma classificação para a atividade realizada. Por esse motivo existe a relação de dependência com a classe *Classificador*.

A classe *Classificador* possui peculiaridades explicadas na [seção 4.4](#). Concluído a execução do método *inferirAtividade*, presente na referida classe, a *Processador* através do padrão de projeto *Strategy*⁴ utiliza de uma entre as classes que implementam a interface *RestritorACG*. A escolha não é aleatória, ou seja, ela é baseada no curso do usuário do aplicativo. Através da classe implementadora da *RestritorACG*, é possível calcular as horas aproveitáveis da ACG classificada, em razão desse cálculo ter variações conforme a classificação da ACG.

A classe *GerenciadorBaseConhecimento* é utilizada para alimentar a tabela da

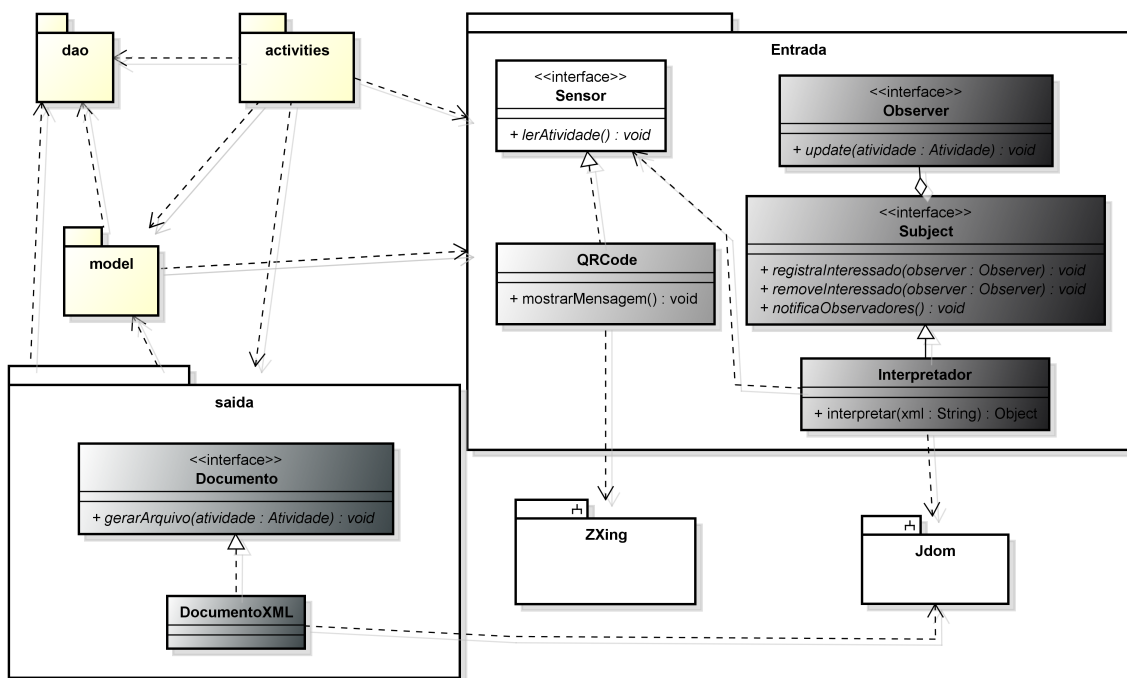
³ O padrão de projeto *Observer* define uma dependência um-para-muitos entre objetos. Seu funcionamento se dá quando um objeto muda de estado, pois todos os seus dependentes são notificados e atualizados automaticamente.

⁴ O padrão de projeto *Strategy* permite estabelecer novas operações sem alterar as classes dos elementos sobre os quais ela opera. É possível definir um conjunto de algoritmos e encapsular cada algoritmo como uma classe, permitindo que o algoritmo possa variar independente das classes que o utilizam.

base de dados chamada *base_conhecimento*. Essa classe utiliza de outra classe chamada *TratadorTexto* para manipular a *string* descrição provida juntamente com o objeto *atividade*. Com a utilização da classe *TratadorTexto* é eliminado diversas palavras e expressões dessa *string*, deixando ao final do processo somente um conjunto de palavras que servem para alimentar a tabela da base de dados citada.

A subsequente fração esclarecida é a que contém os pacotes *saída* e *entrada* do sistema, visualizáveis na Figura 9. O pacote *entrada* engloba a classe *QRCode* que utiliza do aplicativo *ZXing* para ler *QRCode*s do ambiente. Por essa razão, existe a dependência ao subsistema citado. A classe em pauta está destacada devido a também ser uma *activity*, em consequência do aplicativo externo necessitar de métodos implementáveis somente em classes desse gênero para exibir a sua saída. As classes com o tom de cinza mais forte também utilizam do padrão de projetos *Observer*.

Figura 9 – Diagrama de Classes - Entrada | Saída.

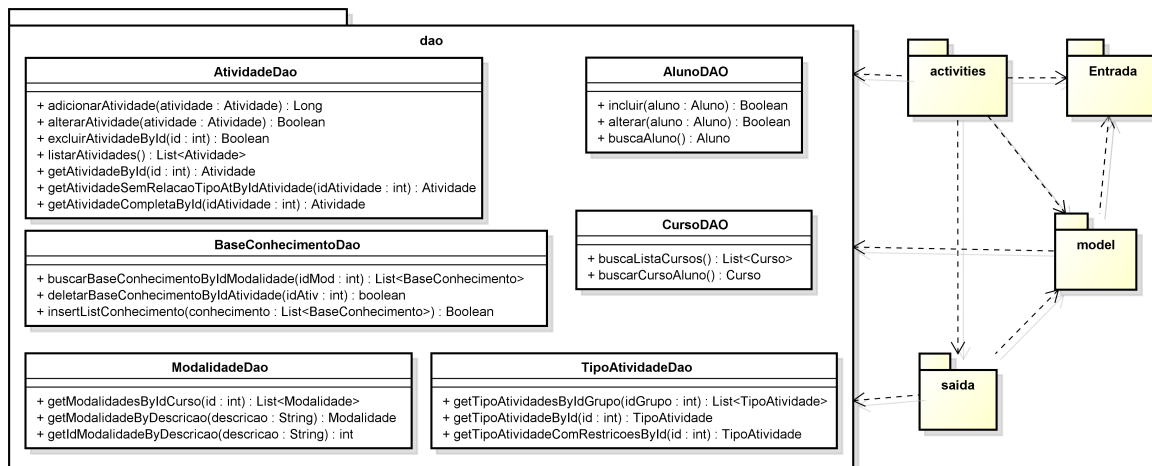


Posterior a decodificação do *QRcode*, A classe *Interpretador* cria um objeto *atividade* contendo as informações processadas pelo método e registra observadores interessados. Sucessivamente a esse processo ocorre a notificação das classes observadoras e é passado o objeto *atividade* para as interessadas como parâmetro no método *update*. A classe *Interpretador* utiliza do subsistema *Jdom* para interpretar o *XML*.

No pacote *saída* existe a classe *DocumentoXML* que implementa o método *gerarArquivo* da interface *Documento*. A classe *DocumentoXML* também é utilizadora do subsistema *Jdom* para compilação de informações e envio para um sistema externo, inexistente até o momento, como já foi explicado na seção anterior.

A última fração extraída do diagrama de classes expandido desse projeto é a que contém o pacote com as classes *DAOs*, as quais podem ser visualizadas na [Figura 10](#). Neste pacote, encontra-se as classes que possuem métodos para persistir dados no banco de dados. Ela é baseada no padrão de projeto *DAO*, em razão de isolar em camada distinta as classes que manipulam dados do banco.

Figura 10 – Diagrama de Classes - DAOs.



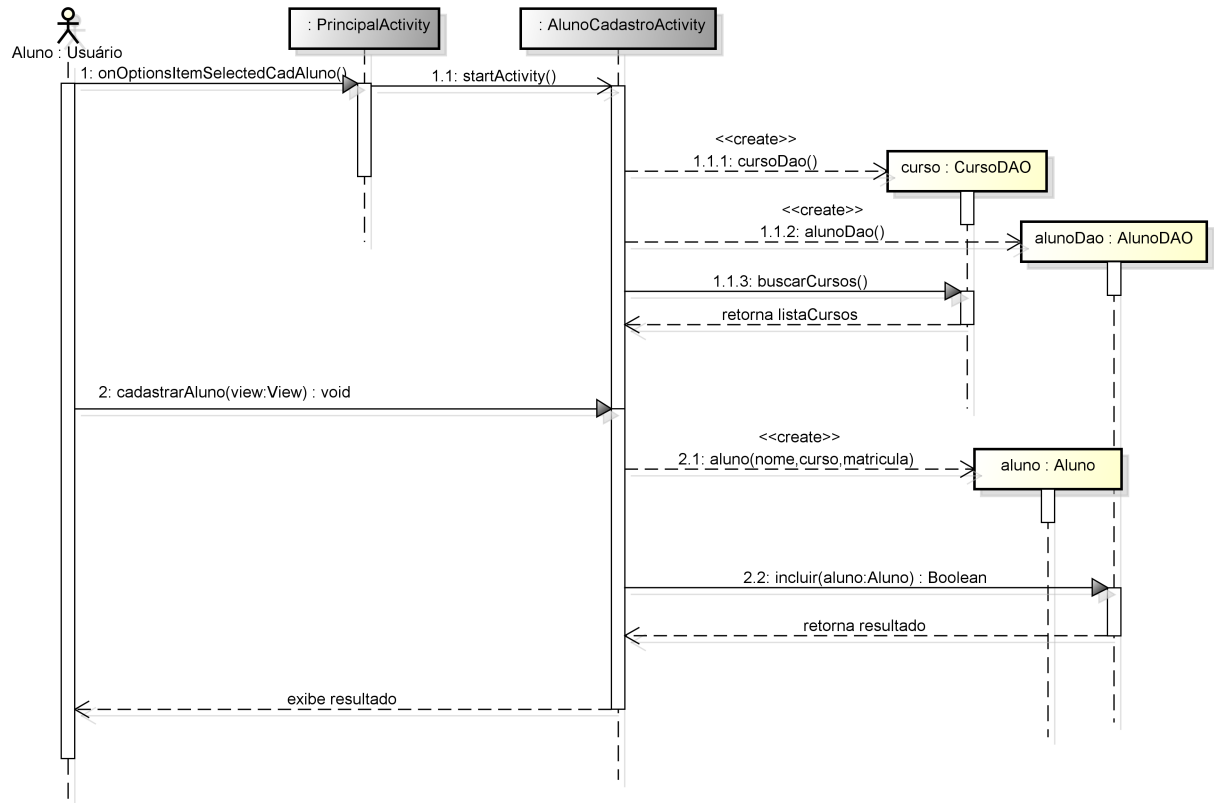
4.3.3 Diagramas de Sequência

O diagrama de sequência serve para apresentar a colaboração dinâmica entre os vários objetos de um sistema (SILVA, 2013). Através dele é possível perceber a sequência de mensagens trocadas entre os objetos. Seu objetivo é demonstrar os métodos executados para a realização de uma funcionalidade.

Para auxiliar no desenvolvimento do aplicativo desse projeto, foram criados diagramas de sequência para cada funcionalidade a ser desenvolvida. A primeira função do sistema é o *Cadastro do Aluno* e seu diagrama é exposto na [Figura 11](#). Com exceção da funcionalidade de *classificar ACG*, todas as demais iniciam através da interação do usuário clicando em uma das opções presentes no menu principal do aplicativo. Essa afirmação pode ser visualizada na primeira interação do diagrama, onde o usuário seleciona a funcionalidade de *cadasturar um aluno*. Após é chamada a *activity* responsável por essa funcionalidade que através da classe *CursoDao* elenca os cursos a serem escolhidos pelo aluno. Por fim, o usuário aciona o método *cadasturarAluno* que cria o objeto *Aluno* e persiste-o no banco por meio da classe *AlunoDao*, encerrando a funcionalidade.

Posterior ao cadastro do aluno no sistema, é possível que o mesmo queira visualizar seus dados. A funcionalidade que contempla essa solicitação é a de *Visualizar Aluno* e seu funcionamento pode ser esclarecido através da [Figura 12](#). O ciclo inicial é idêntico ao da

Figura 11 – Diagrama de Sequência - Cadastrar Aluno.



funcionalidade anterior. As mudanças ocorrem a partir da chegada do processamento na *activity* responsável por gerenciar a exibição desses dados. Esta cria um objeto *AlunoDao* e busca seus dados utilizando o método *buscaAluno* que retorna um objeto *Aluno*. O retorno serve para alimentar um novo objeto *Aluno* na *activity* em questão. Seguinte à criação do objeto *Aluno*, são realizadas consultas via métodos *gets* neste objeto, afim de exibir os dados encontrados nos componentes gráficos da tela.

A próxima funcionalidade explanada é a de *Coletar Informações da ACG*. O seu funcionamento é representado pela [Figura 13](#). O ciclo inicial segue sendo idêntico aos demais, ou seja, a iteração inicia com a escolha da opção de menu por parte do usuário. Posterior a escolha, é iniciado a *activity* responsável pela leitura do *QRCode* do ambiente. Essa, conforme já citado anteriormente, é uma classe do tipo *activity* pois necessita de métodos específicos da mesma para receber o retorno do aplicativo externo *Zxing*.

Com a criação da *activity QRCode*, é chamado em seu construtor o método *lerAtividade*, o qual cria uma instância do subsistema *Zxing*. O método utiliza do subsistema instanciado para fazer a leitura do *QRCode* do ambiente e decodificá-lo, retornando uma *string* correspondente ao *XML* que contém as informações da atividade. Após o retorno é criado uma instância da classe *Interpretador* e chamado o método *interpretar* passando a *string* correspondente ao *XML* como parâmetro. Este método ao final de seu proces-

Figura 12 – Diagrama de Sequência - Visualizar Aluno.

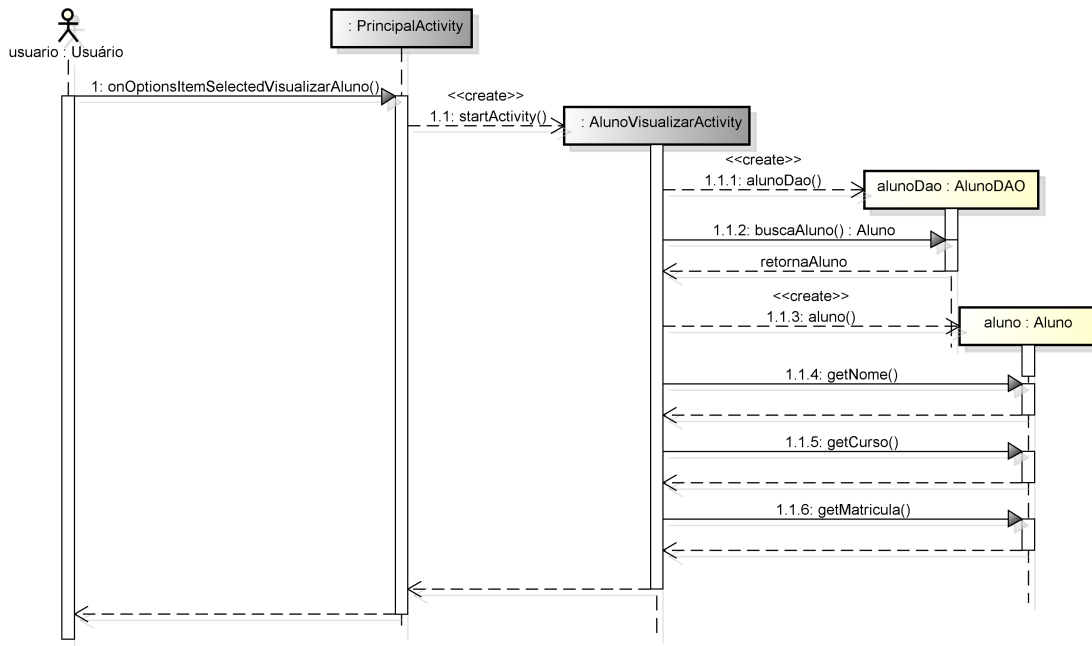
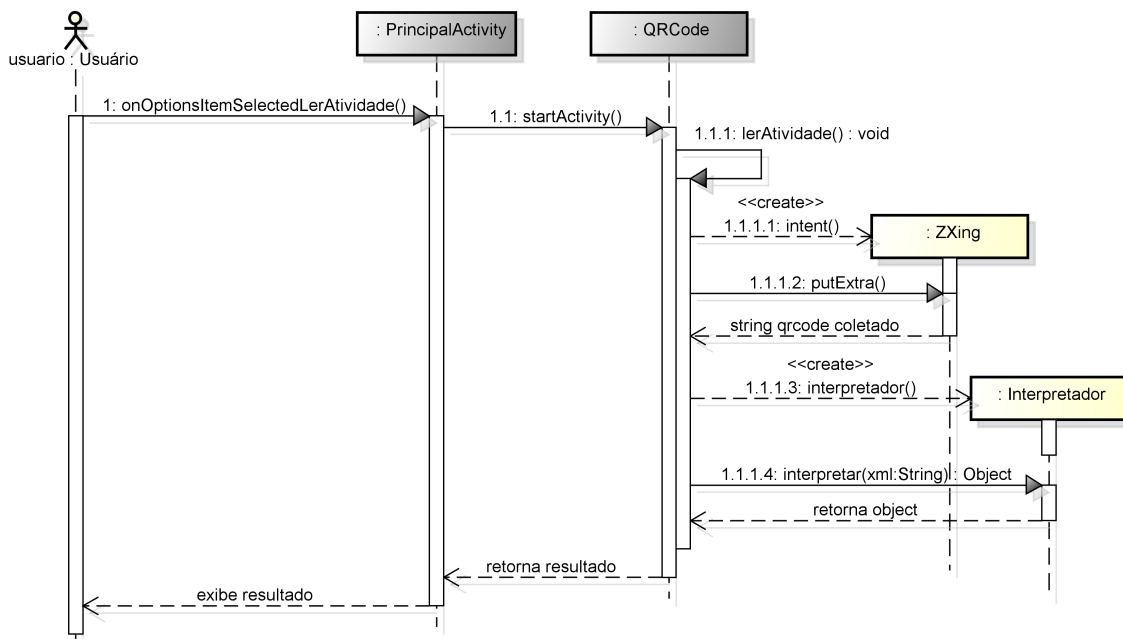


Figura 13 – Diagrama de Sequência - Coletar informações da ACG.



samento, retorna um *object* ou *null* para a classe `QRCode`, a qual exibe o resultado ao usuário.

Recapitulando os casos de uso citados anteriormente, existe uma relação do tipo *include* entre a funcionalidade de *Coletar as Informações da ACG* e *Classificar ACG*, ou seja, ao término da primeira é executada a próxima, conforme Figura 5. O diagrama de

sequência que detalha a segunda funcionalidade é apresentado na [Figura 14](#). A função em destaque é a mais complexa do sistema pois envolve uma série de classes e consultas ao banco de dados.

A funcionalidade citada anteriormente, começa assim que a classe *Intrepretador* registrar como interessada em seu resultado, a classe *Processador*, o que caracteriza o padrão *Observer*. A classe *Processador* ao ser criada, gera por intermédio de seu construtor uma instância da classe *AtividadeDAO*. Após a classe *Interpretador* notifica os interessados e com o método *update* envia o objeto *Atividade* à classe *Processador*. Essa, assim que receber o objeto, chama o método *processarAtividade* iniciando um longo ciclo de processamento.

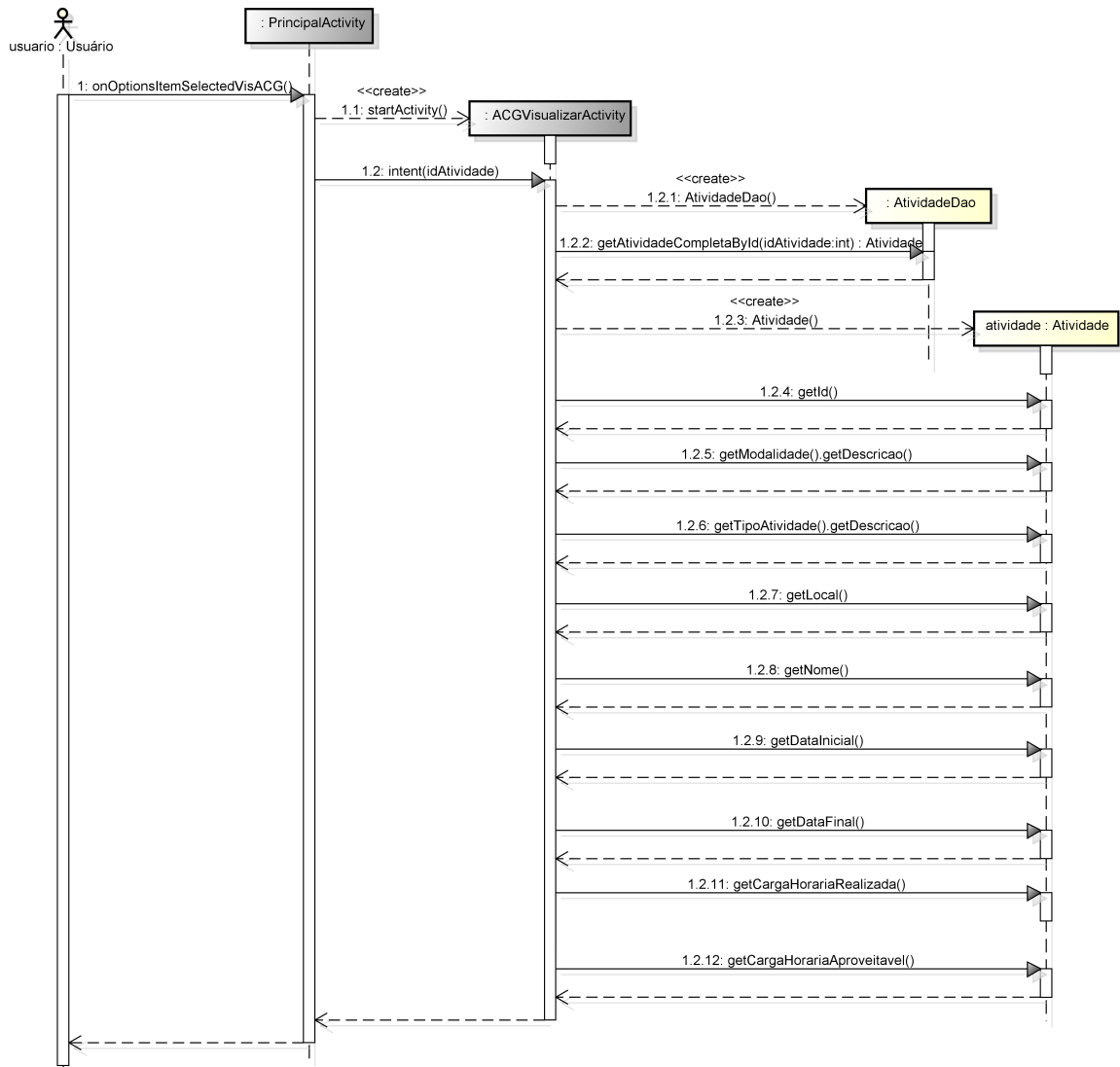
Posterior ao início do método *processarAtividade*, o objeto *atividade* é salvo na base de dados através do método *adicionarAtividade*. O retorno desse método é o *id* adicionado, o qual é setado no objeto *atividade*. É criado um *service* chamado *Classificador* e passado como parâmetro ao mesmo, através do método *putExtra*, o *id* do objeto *atividade* recém salva no banco de dados. Esse procedimento ocorre em razão do do método *putExtra* não aceitar como parâmetro objetos. O *service* *Classificador* cria instâncias das classes *TratadorTexto*, *ModalidadeDao*, *TipoAtividadeDao* e *AtividadeDao*. Seguente ao término de criação dos objetos, o *service* por meio do método *getAtividadeSemRelacaoTipoAtByIdAtividade* recebe um objeto *Atividade*.

Seguido ao recebimento do objeto *Atividade*, o *service* *Classificador* chama o método *verificarCapacidadeDeInferir*. Caso o mesmo retorne *true*, é invocado o método *inferirAtividade*, *alterarAtividade* e por último passado o *idAtividade* como parâmetro ao método *sendReceiver*, retornando o processamento a classe *Processador*. Na circunstância do método *verificarCapacidadeDeInferir* retornar *false* é acionado da mesma forma o método *sendReceiver*, porém, passando 0 como parâmetro.

A seguir, verificasse se o retorno do parâmetro *resultReceiver* é maior que o numeral 0. Na hipótese positiva, a classe *Processador* usufrui do método *getAtividadeById* para recuperar o objeto *Atividade*. Posteriormente, é criado o objeto *Restritor* e passado por intermédio de parâmetro o objeto atividade para o método *RestringirACG*. Ao final, é verificado se o mesmo é diferente de *null*, caso sim, é alterada a atividade no banco de dados através do método *alterarAtividade*.

O próximo diagrama de sequência exibido é o *Visualizar Informações da ACG*, aparente na [Figura 15](#). O início da função é igual aos demais. Criado a *activity* responsável pela funcionalidade, a mesma em seu construtor cria o objeto *AtividadeDao* e utiliza deste para buscar as informações da ACG por meio do método *getAtividadeCompletaById*, passando como parâmetro o *id* da atividade. Posterior ao retorno do método, é inicializado o objeto *Atividade* e executada diversas consultas a este através de métodos *gets*.

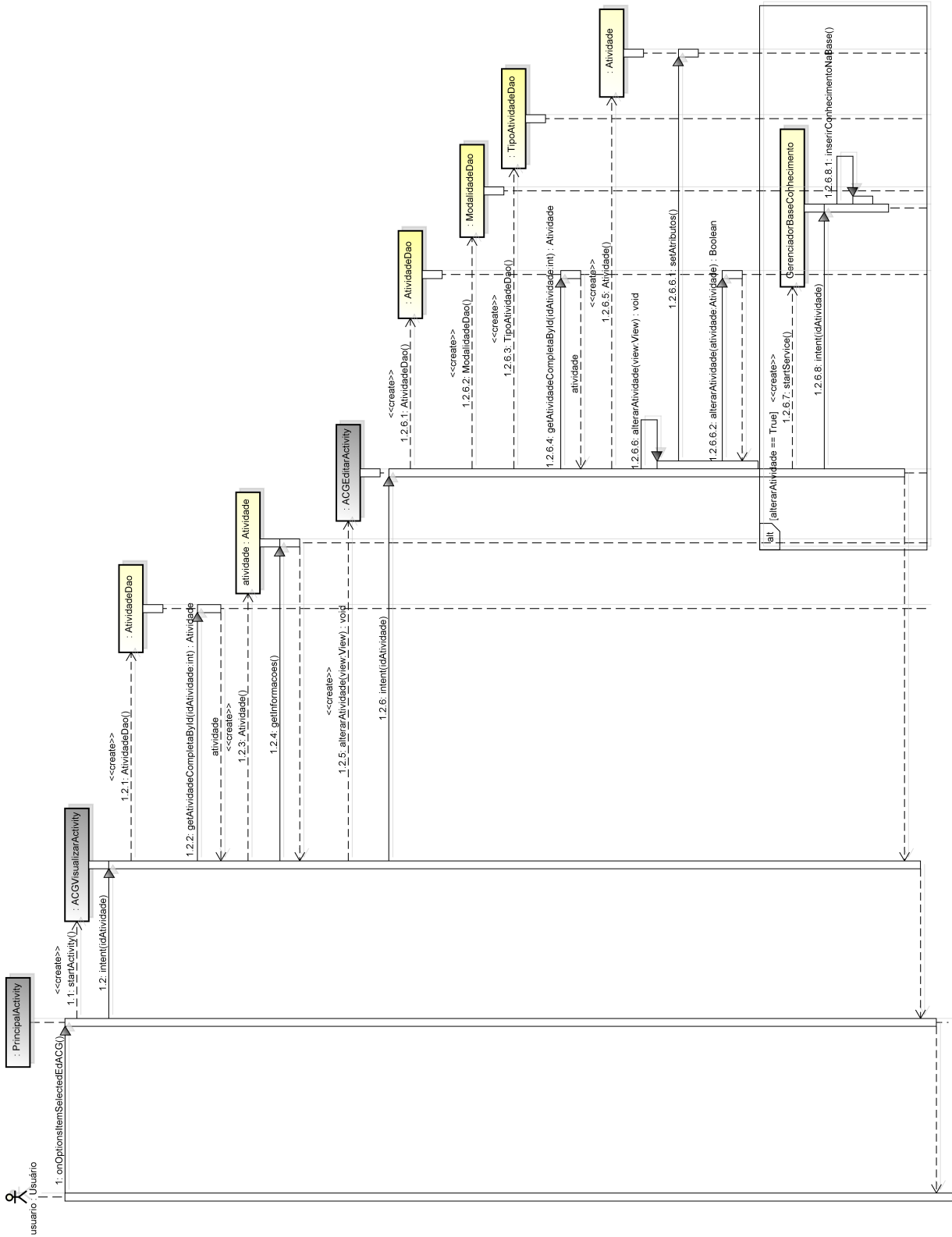
Figura 15 – Diagrama de Sequência - Visualizar informações da ACG.



o usuário aciona o método *alterarAtividade*, resultando na criação da *activity* *ACGEditarActivity*, a qual recebe o *id* da Atividade através do método *intent*. Posteriormente, a *activity* em questão cria os objetos *AtividadeDao*, *ModalidadeDao* e *TipoAtividadeDao*. Ao utilizar do método *getAtividadeCompletaById* e passar o *idAtividade* vinda da *Intent* como parâmetro, a *activity* consegue recuperar a atividade. Após, a mesma cria o objeto *Atividade* para receber esse resultado. Posterior ao usuário requisitar o método *alterarAtividade* da *activity*, é setado novos valores aos atributos do objeto *Atividade* e este é persistido no banco de dados por intermédio do método *alterarAtividade*.

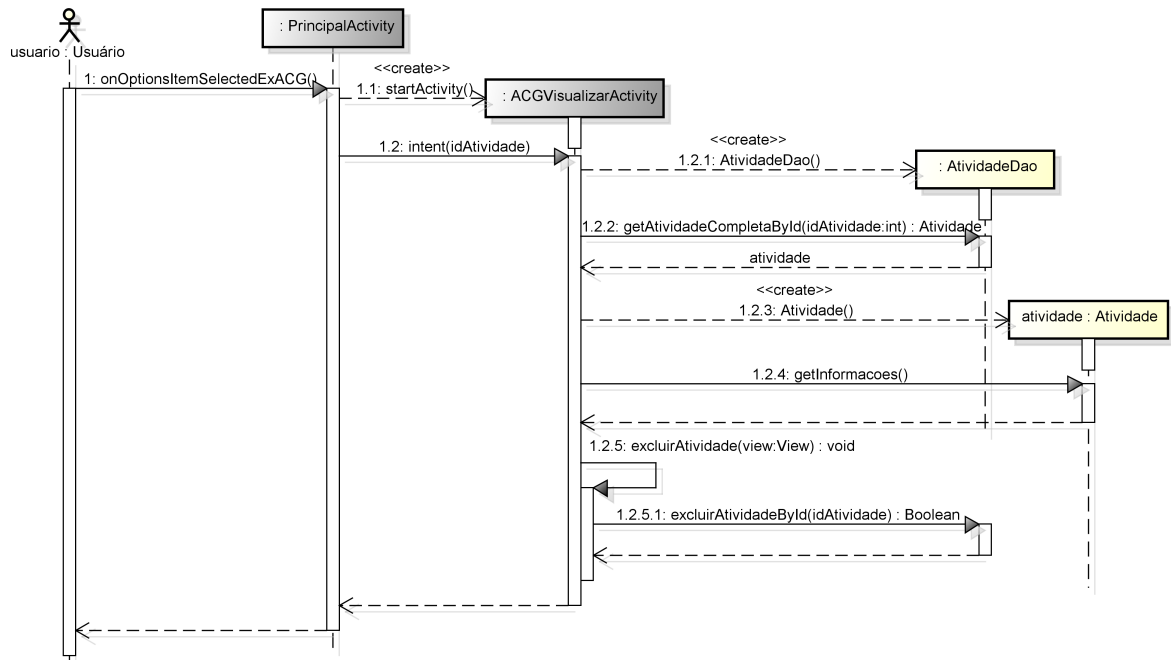
A função de *Excluir ACG* é demonstrada na [Figura 17](#). Seu funcionamento inicial é semelhante ao de visualização de informações da ACG. A diferenciação ocorre a partir do instante que o usuário aciona o método *exclurAtividade*. Este, utiliza o método *ExcluirAtividadeById* da classe *AtividadeDAO*, resultando na exclusão da ACG na base de

Figura 16 – Diagrama de Sequência - Editar as Informações da ACG.



dados.

Figura 17 – Diagrama de Sequência - Excluir ACG.



A última função projetada por meio de diagramas de sequência é a *Aproveitar ACG*, visível na [Figura 18](#). Seu funcionamento permanece sendo semelhante aos demais. A modificação inicia a partir do momento que o usuário aciona o método *aproveitarAtividade*, o qual cria o objeto *DocumentoXML*. A sequência ideal para essa funcionalidade seria a comunicação com um sistema externo para envio do *XML* gerado. Entretanto, pela inexistência do sistema, a funcionalidade termina com a geração do *XML*.

4.3.4 Diagrama de Entidade-Relacionamento

O Diagrama de Entidade Relacionamento (DER) descreve o modelo de dados de um sistema com alto nível de abstração. Ele é o resultado da modelagem de dados executada pelo projetista. No projeto em questão, o DER gerado pode ser visualizado na [Figura 19](#).

Figura 18 – Diagrama de Sequência - Aproveitar ACG.

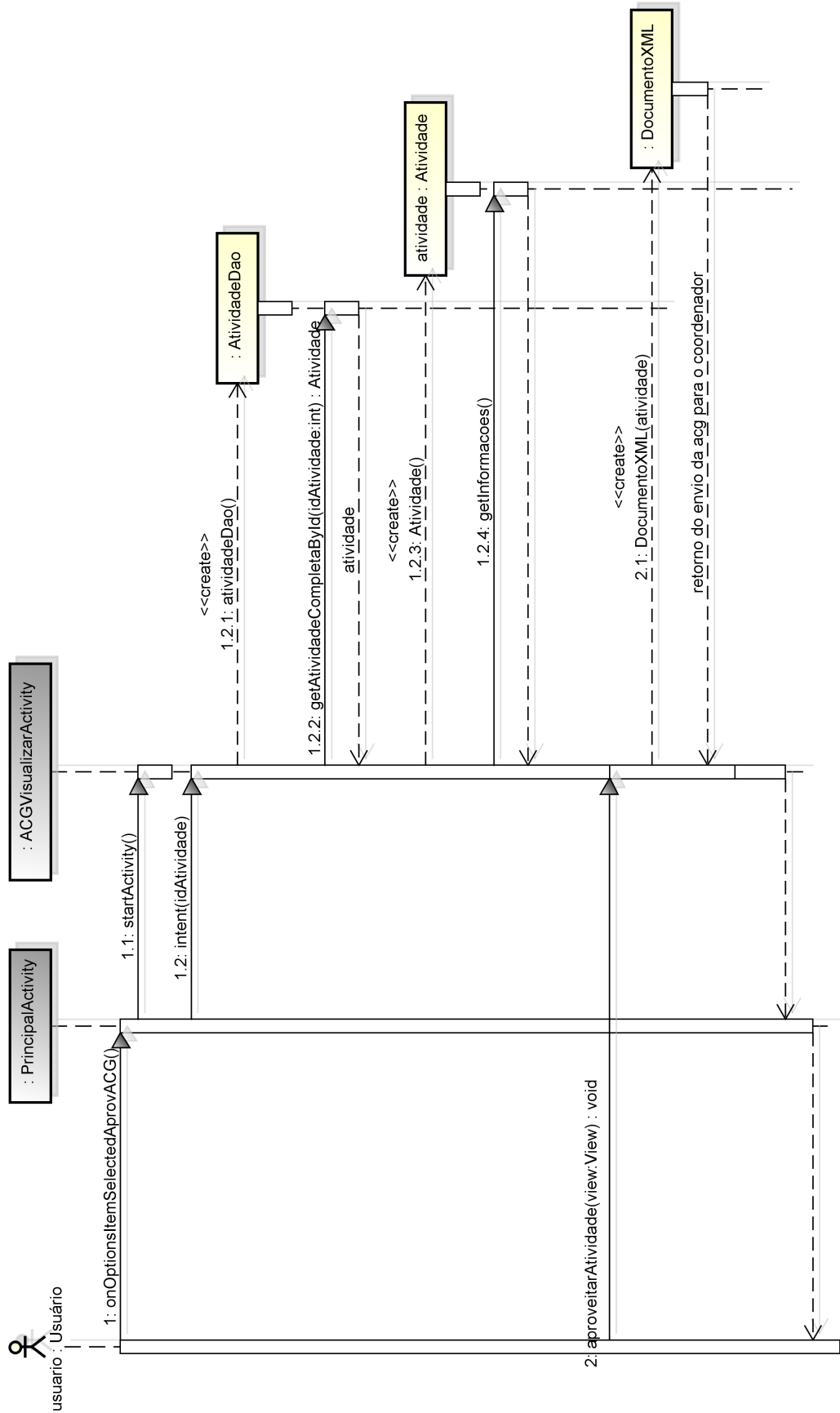
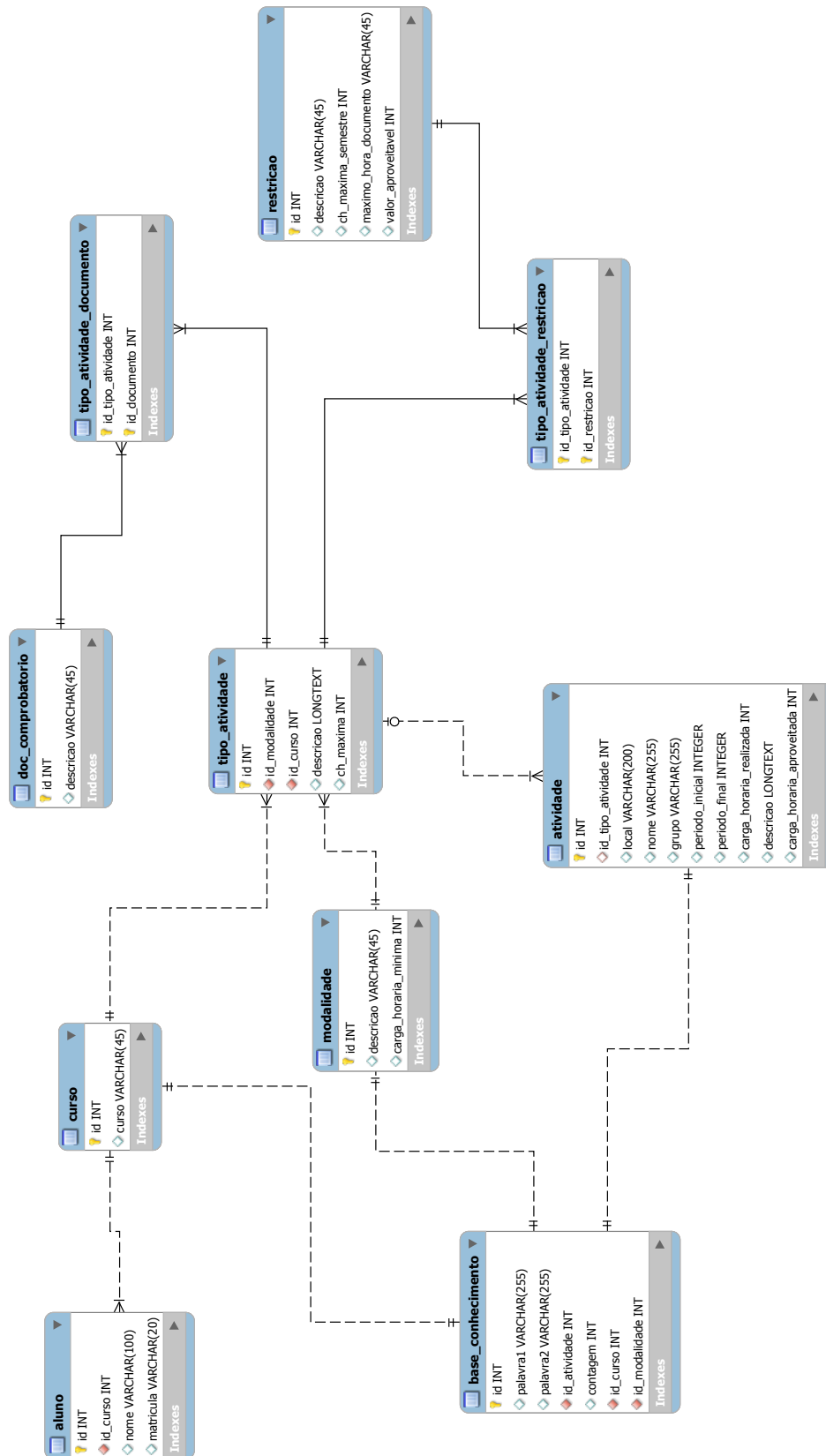


Figura 19 – Diagrama de Entidade-Relacionamento.



No diagrama em questão cada quadrado representa uma tabela na base de dados. Os relacionamentos ocorrem por meio das chaves primárias, ou *primary keys* (PK). A tabela *atividade* pode ou não ter uma relação com a tabela *TipoAtividade*, por essa razão o relacionamento entre ambas é classificado como fraco. Na tabela *base_conhecimento* existe a relação com a tabela *modalidade* em razão da aplicação utilizar do *id Modalidade* para otimizar alguns métodos executados no processamento da função de inferência, explicados na seção 4.4.

Para evitar redundância de dados, modelou-se as tabelas *tipo_atividade_documento* e *tipo_atividade_restricao*, ficando sob responsabilidade de ambas apenas guardar os *ids* das respectivas ligações. Na tabela *atividade* foi definido os campos *periodo_inicial* e *periodo_final* como *integer* em consequência do *sqlite* na sua versão para Android não aceitar campos do tipo *date*.

4.4 Implementação

Para a construção do aplicativo denominado SmartACG foi configurado o ambiente de desenvolvimento *Eclipse* na sua versão 3.6 juntamente com a última versão do SDK do Android. Foi necessário também a integração do *Eclipse* com o SDK por meio do *plugin* chamado ADT. O ambiente na integra pode ser baixado através do link <<http://developer.android.com/sdk/index.html/win-bundle>>.

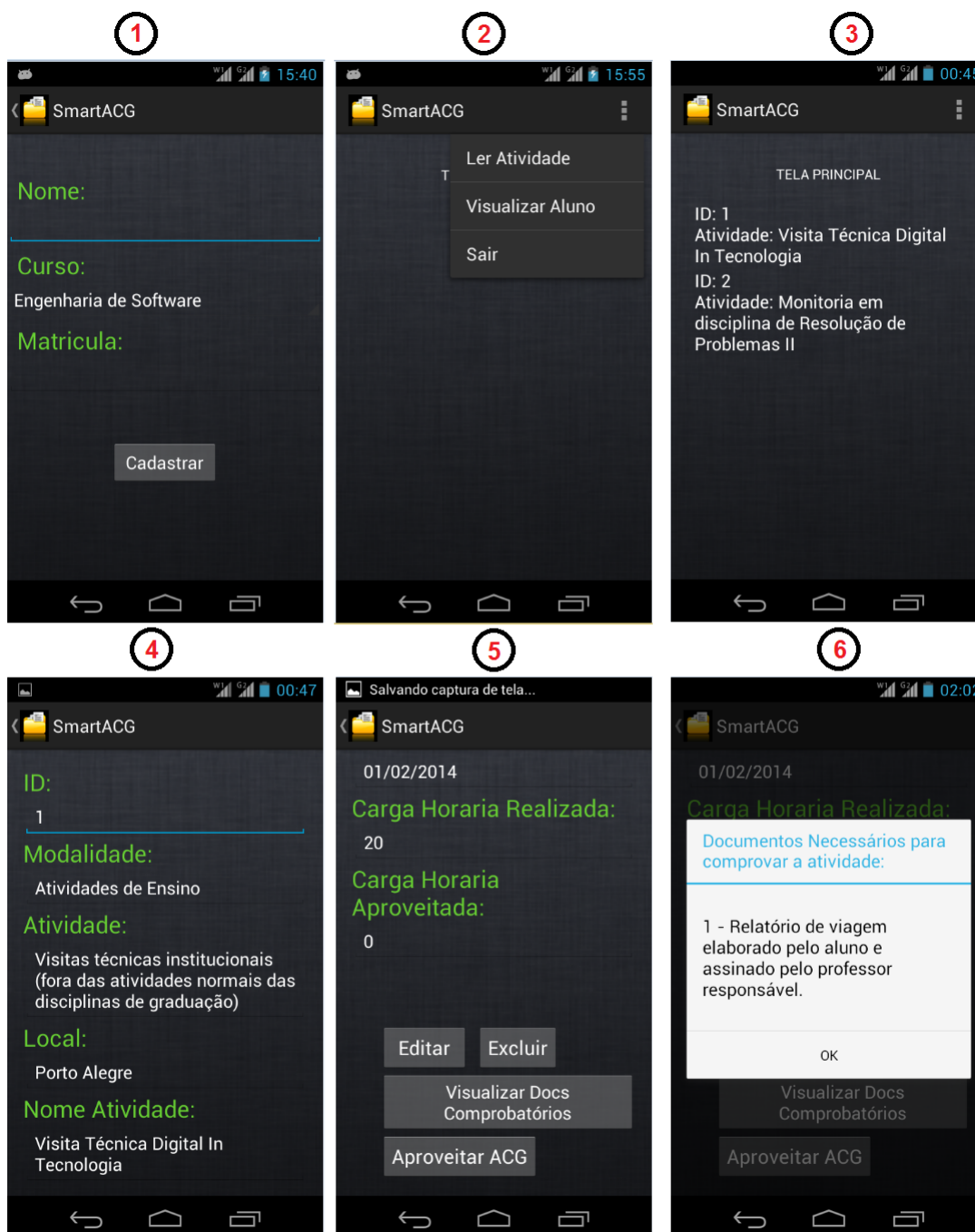
A API de desenvolvimento foi a 4.2.2 *Jelly Bean*. Esta contém a API do Android e o banco de dados *sqlite* integrado. O aplicativo desenvolvido é apto a funcionar em dispositivos móveis com a plataforma Android na versão 4.1 ou superior.

Com o intuito de apresentar o sistema desenvolvido, fez-se uma montagem, visível na Figura 20, onde contém algumas telas do sistema. Todos os *print screens* foram tirados por intermédio do *smartphone* Motorola *Razr D3*, o qual possui tela de quatro polegadas.

A tela número um corresponde a funcionalidade de *Cadastrar Aluno*, onde é necessário que o mesmo informe seu nome, curso e matrícula. É obrigatório o aluno realizar o seu cadastramento na primeira vez que acessar o aplicativo, caso contrário, o aluno fica inibido de utilizar quaisquer das funcionalidades presentes no aplicativo.

A tela número dois corresponde a tela principal. Nessa imagem o aplicativo ainda não contém nenhuma atividade em sua base de dados, por esse motivo a tela está vazia. Na parte superior direita dessa tela é possível ver o menu com as opções que o usuário dispõe na utilização do aplicativo.

A tela número três exibe a tela principal já com duas atividades fictícias. Caso o usuário desejar, é possível visualizar as informações da atividade realizada clicando na mesma.

Figura 20 – Telas do aplicativo *SmartACG*.

Na tela de número quatro é exibida a primeira parte das informações da atividade de id número 1. Caso o aplicativo não tenha conseguido inferir a classificação para a atividade realizada, os campos Modalidade e Atividade aparecerem em branco, sendo necessário que o aluno realize a classificação da atividade.

Na tela de número cinco é possível ver as demais informações da atividade realizada e as opções de *Editar*, *Excluir*, *Visualizar Documentos Comprobatórios* e *Aproveitar ACG*. A opção editar, redireciona o aluno para a tela de edição da atividade, possibilitando a

alteração de seus dados.

O aluno pode desejar visualizar os documentos necessários para comprovar a atividade realizada. Para isso, é preciso clicar no botão *Visualizar Docs Comprobatórios*. Ao clicar, é exibida a tela de número seis para o aluno. Essa tela basicamente consiste em um *pop-up* que contém uma lista com todos os documentos necessários para comprovar a veracidade da atividade realizada.

Por fim, caso o aluno pretender, é possível pedir o aproveitamento da ACG, clicando no botão *Aproveitar ACG*. Acionada essa funcionalidade, o sistema compila as informações do aluno com as da atividade realizada, gerando no final do processo um *XML*. O fluxo ideal dessa funcionalidade seria após gerar o *XML*, o mesmo ser enviado ao coordenador por intermédio da internet, entretanto, como atualmente não existe o sistema para receber o *XML*, a funcionalidade termina com a geração do mesmo, visível na [Figura 21](#).

Figura 21 – Saída de console - Pedido de aproveitamento de ACG.

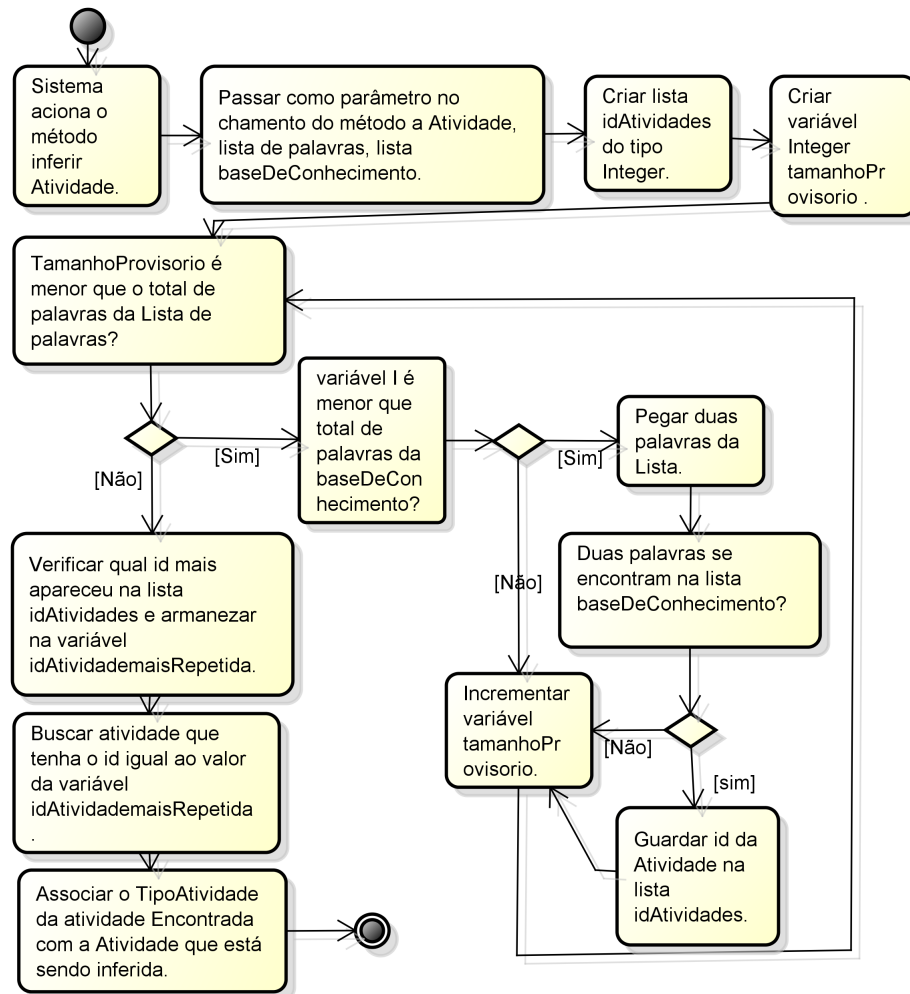
```
br.unipampa.smartacg      System.out  <?xml version="1.0" encoding="UTF-8"?>
br.unipampa.smartacg      System.out  <acg><aluno>Rafhael</aluno><curso>Engenharia de Software</curso><modalidade>Atividade
es de Ensino</modalidade><atividade>Visitas técnicas institucionais (fora das ativid
ades normais das disciplinas de graduação)</atividade><local>Porto Alegre</local><no
me>Visita Técnica Digital In Tecnologia</nome><datainicio>Sat Feb 01 00:00:00 BRT 20
14</datainicio><datafim>Sat Feb 01 00:00:00 BRT 2014</datafim><chrealizada>0</chreal
izada</acg>
```

O código desse projeto está disponível no *Bitbucket* e pode ser clonado por intermédio do link <<https://rafaelrc@bitbucket.org/rafaelrc/smartacg.git>>.

4.4.1 Mecanismo de Inferência

O mecanismo de inferência é responsável por executar a funcionalidade que explora o conceito *smart* deste trabalho. Após o usuário coletar as informações da atividade no ambiente, é acionada a funcionalidade de *verificarCapacidadeDeInferir* uma classificação, seu fluxo é apresentado na [Figura 22](#). Para a execução dos métodos dessa função, é utilizada uma classe do tipo *IntentService* que no Android funciona rodando tarefas em *background*.

A funcionalidade de *verificarCapacidadeDeInferir* objetiva verificar se na tabela *base_conhecimento*, presente no banco de dados, existe a porcentagem mínima de bigramas do texto analisado. Essa porcentagem é passada como parâmetro ao acionar o método. A tabela *base_conhecimento* está estruturada conforme apresentado na [Figura 23](#).

Figura 22 – Diagrama de Atividades - Método *verificarCapacidadeDeInferir*.Figura 23 – Print Screen - Tabela do banco de dados chamada *base_conhecimento*.

RecNo	id	palavra1	palavra2	id_atividade	contagem	id_curso	id_modalidade
Click here to define a filter							
1	1	visita	tecnica	1	1	6	1
2	2	empresa	digital	1	1	6	1
3	3	in	tecnologia	1	1	6	1
4	4	localizada	cidade	1	1	6	1
5	5	porto	alegre	1	1	6	1
6	6	rs	empresa	1	1	6	1
7	7	especializada	solucoes	1	1	6	1
8	8	infraestrutura	servidores	1	1	6	1
9	9	rede	seguranca	1	1	6	1
10	10	suporte	hardware	1	1	6	1
11	11	desenvolvimento	sistemas	1	1	6	1

A tabela *base_conhecimento* guarda o conjunto de bigramas, o *id* da atividade já realizada e a modalidade a qual essa atividade pertence. Sendo assim, ao acionar o

método *verificarCapacidadeDeInferir* é feita uma busca na base de dados para recuperar todo o conjunto de bigramas que contém a relação com a modalidade da atividade realizada. O procedimento explicado é realizado em razão de que caso fosse buscado todo o conjunto de bigramas existente na base de dados, o aplicativo demoraria muito para realizar o processamento, visto que os dispositivos móveis ainda possuem certa limitação de hardware.

Posterior ao resgate, é verificado quantos bigramas da atividade realizada estão presentes no *array* de bigramas resgatado. Por fim, assume-se que o conjunto total de bigramas da atividade realizada corresponde ao total de 100% e, a partir disso, é feita uma regra de três objetivando descobrir qual foi a porcentagem de bigramas encontrados na base de dados. Se a porcentagem for maior ou igual a informada no chamamento do método *verificarCapacidadeDeInferir* é retornado *true*, caso contrário *false*.

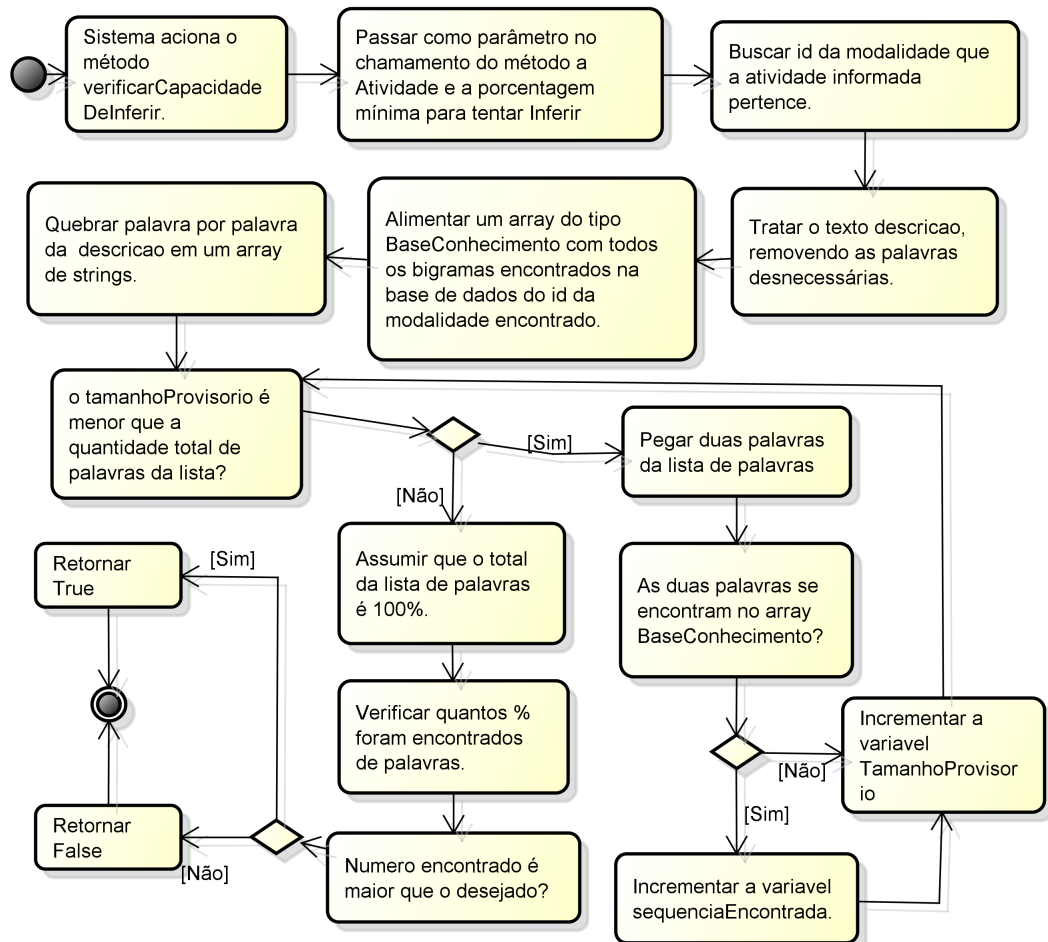
Caso o método *verificarCapacidadeDeInferir* retorne *false*, são salvas as informações da atividade realizada e a funcionalidade termina sem conseguir inferir uma classificação para a mesma. Se o retorno do método for *true*, é acionada a funcionalidade *inferirAtividade*, que tem seu fluxo apresentado na [Figura 24](#).

A funcionalidade *inferirAtividade* objetiva encontrar o *id* da atividade que mais esteve relacionado aos bigramas da atividade realizada e relacionar sua classificação a atividade. Seu funcionamento é similar ao do método anterior, a diferenciação ocorre a partir do momento do encontro dos bigramas. Nesse caso, é guardado o *id* da atividade em um *array* de *ids*. Ao final do processo é verificado qual *id* apareceu mais dentro do *array* alimentado no decorrer do método.

O próximo passo da funcionalidade *inferirAtividade* é descobrir qual é a atividade na qual o *id* mais se repetiu nesse processamento. Após essa descoberta, o *TipoAtividade*, o qual representa a atividade na tabela de ACGs, é relacionado a nova atividade em questão. Posteriormente, é chamado um outro *intentService* que pega o conjunto de bigramas da atividade ingressada no sistema e alimenta a tabela *base_conhecimento* fazendo com que o aplicativo aprenda.

4.5 Considerações Sobre o Capítulo

Neste capítulo foi apresentada a visão da solução elaborada para o problema de aproveitamento de ACGs, detalhando todo o processo de construção do aplicativo e seu mecanismo de inferência. A pesquisa realizada foi muito importante para a coleta das necessidades enfrentadas pelos alunos no processo de aproveitamento de ACGs. Na fase de projeto, a criação dos diagramas UML serviram para que se pudesse especificar todo o aplicativo, possibilitando encontrar problemas de inconsistência antes de seu desenvolvimento. A etapa de desenvolvimento proporcionou o aprendizado da linguagem de

Figura 24 – Diagrama de Atividades - Método *inferirAtividade*.

programação utilizada para a criação de aplicativos para a plataforma Android, além de suas peculiaridades. Por fim, a construção do mecanismo de inferência viabilizou a percepção dos conhecimentos adquiridos na [seção 2.3](#).

5 Validação





O capítulo a seguir expõe os objetivos buscados na realização dos testes do aplicativo desenvolvido. Exibe-se o processo utilizado para a criação dos modelos de teste, posteriormente, utilizados para a realização dos testes no aplicativo. Por fim, apresenta-se a análise dos resultados obtidos após a conclusão da execução dos testes. Na [seção 5.1](#) é exemplificado a base de ACGs criada para servir de subsídio para a realização dos testes. A [seção 5.2](#) ilustra detalhadamente os modelos criados explicando o seu funcionamento. Na [seção 5.3](#) é exibido os resultados dos testes após a execução dos modelos desenvolvidos na seção anterior e a análise dos resultados obtidos através dessa execução. Finalizando, na [seção 5.4](#) é realizado o fechamento do capítulo.

5.1 Amostra de Teste

Para a realização dos testes do aplicativo desenvolvido, primeiramente criou-se uma planilha contendo dez exemplos de cada uma das 28 atividades existente na tabela do curso de Engenharia de Software, disponível no [Apêndice A](#). Entretanto, todos os grupos que compõem essa planilha possuem a atividade: "Outras, a critério da comissão de Curso". Essa atividade foi excluída devido a possuir um processo diferente de validação, o qual envolve a decisão do colegiado do curso. O grupo cinco intitulado avaliação de desempenho também foi excluído do processo de validação, em razão de deter somente um exemplar de atividade e não possuir um comprovante de realização da mesma até o presente momento.

Uma pequena amostra da base de ACGs é visível na [Figura 25](#). Com exceção da coluna de código da atividade que foi criada com o intuito de orientar o testador a saber qual é a classificação correta da atividade em questão, as demais são todas usuais para preenchimento do formulário de geração de *QRCode* já exemplificado anteriormente. A coluna *QRCode* contém a imagem do mesmo já gerado para facilitar a leitura do aplicativo. O final do processo de criação resultou em duzentos e oitenta exemplares de atividades, os quais não serão disponibilizados no trabalho por questões de dimensão.

Figura 25 – Amostra da base de ACGs.

Código Atividade	Local	Nome	Data Inicio	Data Fim	Carga Horaria	Descrição	QR Code
1.1	Alegrete	Disciplina de circuitos digitais	05/11/2013	04/04/2014	60	Realização da disciplina de Circuitos digitais juntamente com a turma do 2º semestre do curso de Engenharia Elétrica da Unipampa. O objetivo da disciplina é servir de subsídio para o aluno após seu término ser capaz de analisar, simplificar e sintetizar sistemas à base de circuitos digitais. Pedido de aproveitamento da disciplina Arquitetura e Organização de computadores cursada com a turma da engenharia elétrica. A disciplina deverá ao seu término propiciar que o aluno consiga e descrever os elementos constituintes de uma unidade central de processamento, analisar o fluxo elementar de seus dados e programá-la.	
1.1	Alegrete	Disciplina de arquitetura e organização de computadores 1	05/11/2013	04/04/2014	60		
1.1	Alegrete	Disciplina de Redes de Computadores	05/11/2013	04/04/2014	90	Aproveitamento da cadeira de Redes de computadores feita com a turma do 5º semestre da Engenharia da Computação em Bagé RS. Disciplina tem por finalidade introduzir o assunto de redes de computadores, estrutura e topologias de redes, arquiteturas de camadas, 90 camadas de aplicação, entre outros assuntos.	
1.1	Alegrete	Disciplina Sistemas Operacionais	05/11/2013	04/04/2014	90	Solicitação de computo de horas para ACG da disciplina Sistemas Operacionais cursada com a turma da Engenharia da computação da Unipampa Campus Bagé, onde está tem por intuito introduzir uma definição para sistemas operacionais, serviços providos por um sistema operacional, estrutura e organização típicas, gerência de processos, gerência de memória, gerência de entrada e saída e sistemas de arquivos.	

Depois de concluída a base de ACGs, a mesma foi replicada em um outro arquivo *Excel*. Posteriormente, se fez uso do método *random* do mesmo software para embaralhar as atividades. Objetiva-se com esse procedimento dificultar os testes, em virtude da base original conter todos os exemplares de atividades ordenados, conforme demonstrado na figura anterior.

5.2 Estratégia de Teste

Para testar o aplicativo desenvolvido no decorrer desse trabalho, criou-se duas maneiras distintas para validá-lo. A necessidade de automatização dos testes fez com que fossem criados também dois modelos diferentes para executá-los mais rapidamente.

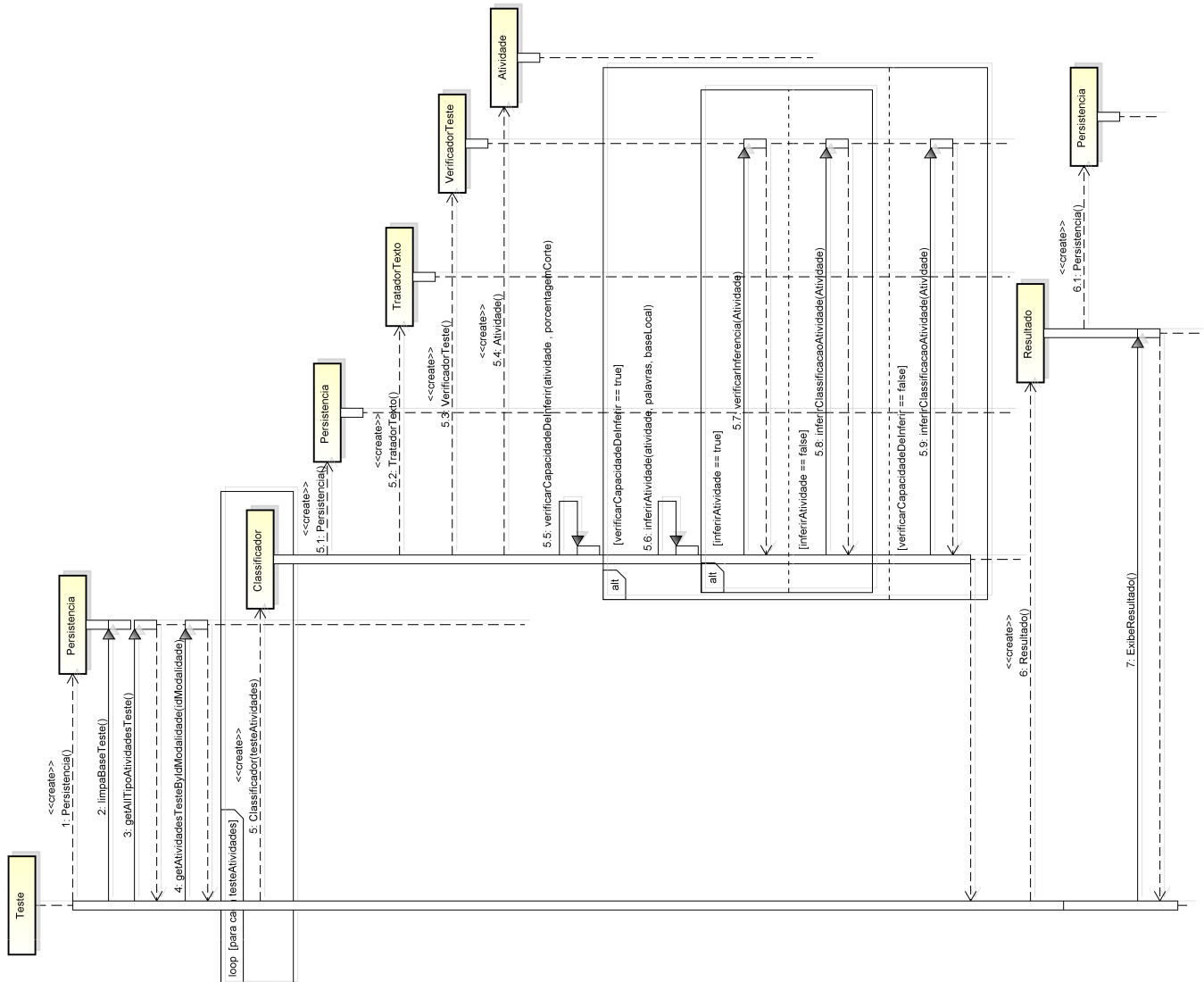
O primeiro modelo de testes criado tem como propósito simular o uso do aplicativo de maneira real. O segundo modelo tem por finalidade testar a capacidade do mecanismo de inferência, adotado nesse trabalho para realizar a classificação automática das atividades ingressadas no sistema. A fins de explicação, adotou-se como nomenclatura chamar o primeiro exemplo de modelo 1 e o segundo como modelo 2.

Os modelos de teste foram criados na linguagem de programação Java SE com a utilização da IDE de programação Netbeans em sua versão 7.4. Escolheu-se a referida linguagem em razão do SO Android utilizar da mesma como referência em sua API, possibilitando o reuso das classes já implementadas para o uso do aplicativo.

O funcionamento do primeiro modelo pode ser visto na [Figura 26](#). Suas classes possuem as seguintes finalidades:

- **Classe *Teste***: Utilizada para a realização dos testes. Possui o método *main* para acionar as demais classes e executar os testes.
- **Classe *Persistencia***: Utilizada para persistir dados no banco.
- **Classe *Classificador***: Utilizada para deduzir a classificação de uma atividade.
- **Classe *TratadorTexto***: Utilizada para tratar a *string* a ser analisada, eliminando as palavras desnecessárias.
- **Classe *VerificadorTeste***: Utilizada para verificar se uma atividade foi inferida de maneira correta e inferir uma atividade caso o ponto de corte de bigramas encontrados no sistema seja inferior ao desejado.
- **Classe *Atividade***: Utilizada para representar uma atividade.
- **Classe *Resultado***: Utilizada para exibir os resultados após a realização dos testes.

Figura 26 – Diagrama de Sequência - modelo de Teste 1.



O modelo de teste 1 é responsável por tentar inferir a classificação de todas as atividades do grupo a ser testado. Caso os métodos tradicionais de classificação da atividade presentes no aplicativo não consigam realizar a classificação da atividade ingressada, o modelo realiza a classificação da atividade, entretanto, é informado na tabela *log*, presente na base de dados, que a atividade não conseguiu ser inferida. Todas as atividades testadas servem de subsídio para alimentar a base de conhecimento. Além das peculiaridades citadas, o modelo também é capaz de corrigir a classificação da atividade, caso essa tenha sido inferida de maneira errônea.

O modelo de teste 2 possui seu funcionamento semelhante ao anterior. Entretanto, por objetivar testar o mecanismo de inferência presente no aplicativo em questão, são executados testes diferenciados. A distinção entre os modelos ocorre em razão do modelo

de teste 2 não alimentar a base de conhecimento a cada atividade testada. Nesse modelo, a base de conhecimento é alimentada apenas uma vez, com todas as atividades separadas para essa finalidade. Esse modelo também não realiza a classificação para a atividade caso os métodos responsáveis por essa finalidade não tenham obtido êxito.

Os dois modelos exemplificados alimentam a tabela *log* presente na base de dados. Ao final da execução dos testes é lido o resultado presente nessa tabela por intermédio do método *exibeResultado*, o qual tem como responsabilidade apresentar o total de atividades testadas inferidas corretamente, erroneamente e não inferidas.

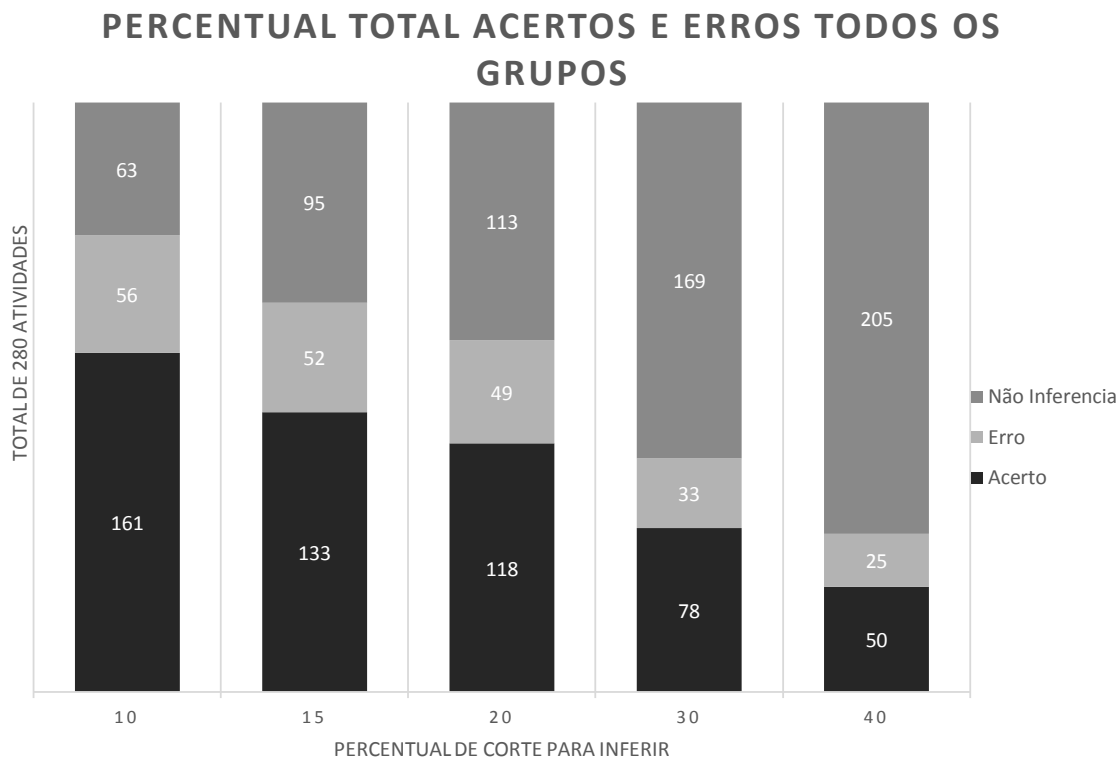
5.3 Resultado dos Testes

A execução do modelo de teste 1, o qual foi explicado seu funcionamento na [seção 5.2](#), é demonstrado na [Figura 27](#). A imagem propicia o entendimento de que todos os testes foram realizados com o total de 280 atividades, correspondentes a soma de exemplares de todos os grupos. Por meio da imagem citada também é compreensível que nos testes realizados o percentual de inferência ¹ é mudado, sendo executados testes nos percentuais de 10, 15, 20, 30 e 40%. A área da coluna pintada na cor preta corresponde as atividades que foram deduzidas de maneira correta. Sob a cor cinza claro ficaram as inferidas de forma errada. Recorda-se que as inferências erradas são corrigidas e salvas novamente. Na cor cinza escuro, encontram-se as atividades não inferidas pelo modelo executado.

O objetivo inicial ao realizar o modelo de testes 1 era simular o uso real do aplicativo, em virtude da base estar sempre se modificando através da ação de classificação das atividades realizadas por parte do usuário. A [Figura 27](#) demonstra que quanto mais se exigir de percentual de bigramas na base de conhecimento, menos o aplicativo consegue realizar inferências. Este resultado é esperado, visto que existe uma grande diferenciação no modo de escrever da linguagem portuguesa, ocasionando em inúmeras maneiras de expressão. Dado que os bigramas são gerados com base nos textos de descrições das atividades escritos pelos *promoters* destas, essa diversificação de palavras é totalmente normal e em razão dessa variável o resultado se torna diversificado.

Os pontos positivos da realização deste teste que foram: A porcentagem de acertos sempre ser superior a de erros e a de erros decrementar quando ha o aumento do percentual de corte para inferência de classificações. Em todos os casos exemplificados na [Figura 27](#), o número de acertos é pelo menos o dobro se comparado ao de erros, podendo assim ser concluído que o mecanismo de inferência utilizado nesse trabalho é eficaz. Contudo, o mesmo necessita de melhorias para se tornar totalmente satisfatório.

¹ Porcentagem informada ao aplicativo para que ele tente ou não realizar a inferência de uma atividade. Para a decisão, o sistema basea-se no número de bigramas existentes na base de conhecimento que são idênticos ao da atividade sendo testada.

Figura 27 – Resultados obtidos na execução do *script* de testes 1.

Os pontos negativos que puderam ser averiguados na realização do mesmo teste foram: O percentual de erros se comparado ao total de inferências em todos os casos foi igual ou superior a 25%. O total de atividades não inferidas sempre foi superior a 60, o que equivale a mais de 20% do total de atividades testadas. Os valores exemplificados não estão fora da média aceitável, entretanto, tornam-se um ponto negativo devido a ocasionar reais possibilidades de induzir o usuário ao erro.

O modelo de testes 2 foi executado de duas formas distintas. Primeiramente executou-se o mesmo variando as atividades classificadas para compor a base de conhecimento. Os resultados obtidos de cada grupo podem ser vistos nas [Figura 28](#), [Figura 29](#), [Figura 30](#), [Figura 31](#) que correspondem respectivamente as atividades de ensino, pesquisa, extensão e gestão.

Todos os testes foram executados ao ponto de corte de 20% e separou-se para todos os grupos 30 atividades. As atividades separadas pertenciam sempre ao grupo a ser testado. Subsequente a separação, incrementou-se o número de atividades alimentadoras da base de conhecimento até chegar ao limite máximo disponível, considerando o número restante de atividades existentes para testes do referido grupo. O procedimento para as atividades do grupo de ensino é demonstrado na [Figura 28](#). Repara-se que o número de atividades da base foi incrementado de 8 em 8, em razão do grupo de Ensino conter 8 atividades e o modelo estar programado sempre para colocar um exemplo de cada, como

Figura 28 – Resultados obtidos na execução do modelo de testes 2 variando o tamanho da base para as atividades de Ensino.

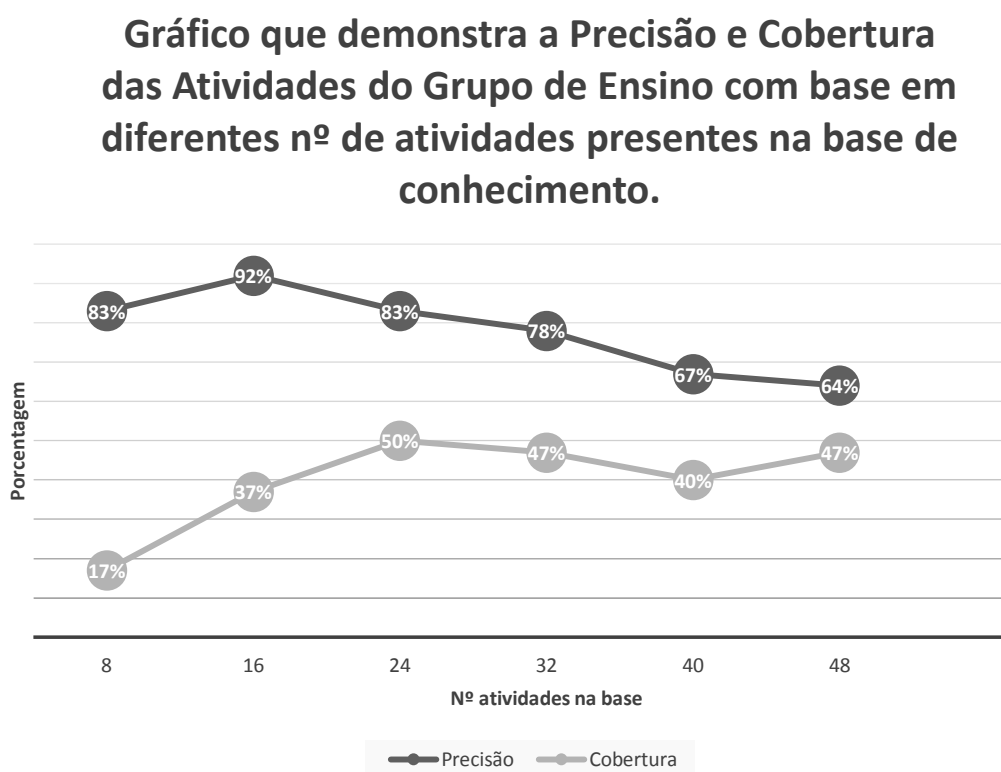


Figura 29 – Resultados obtidos na execução do modelo de testes 2 variando o tamanho da base para as atividades de Pesquisa.

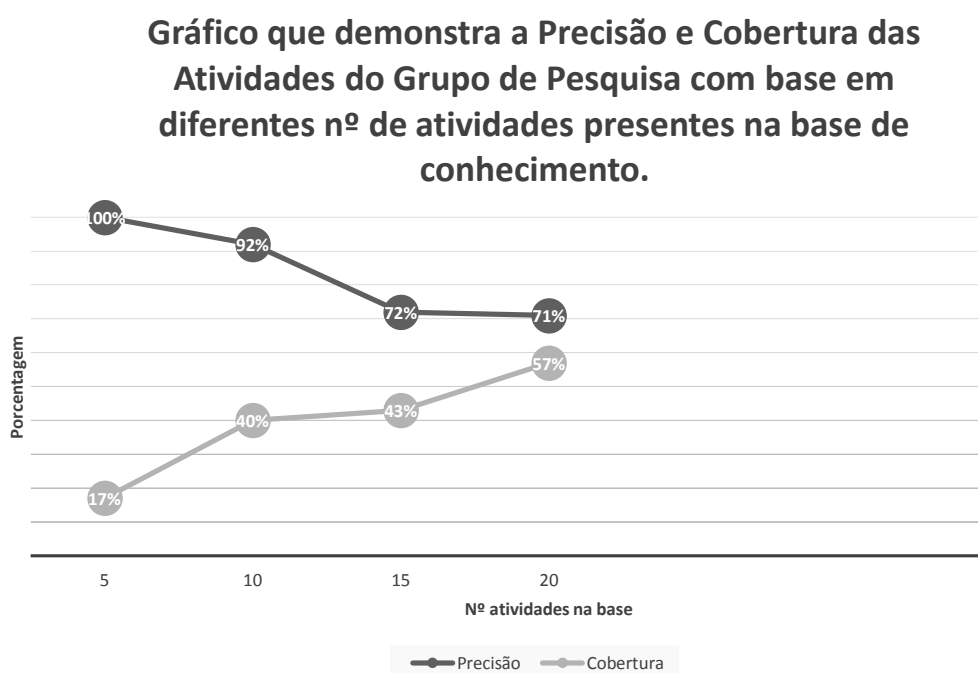


Figura 30 – Resultados obtidos na execução do modelo de testes 2 variando o tamanho da base para as atividades de Extensão.

Gráfico que demonstra a Precisão e Cobertura das Atividades do Grupo de Extensão com base em diferentes nº de atividades presentes na base de conhecimento.

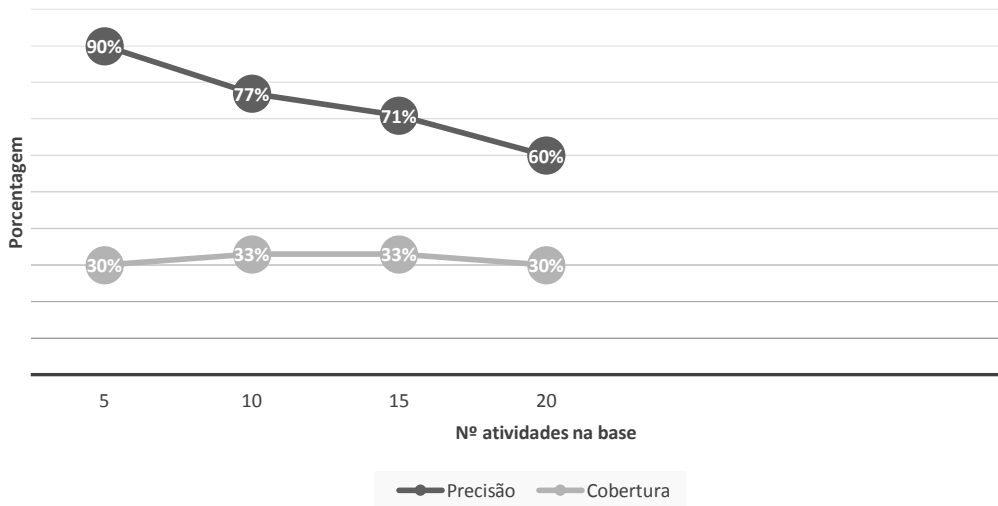
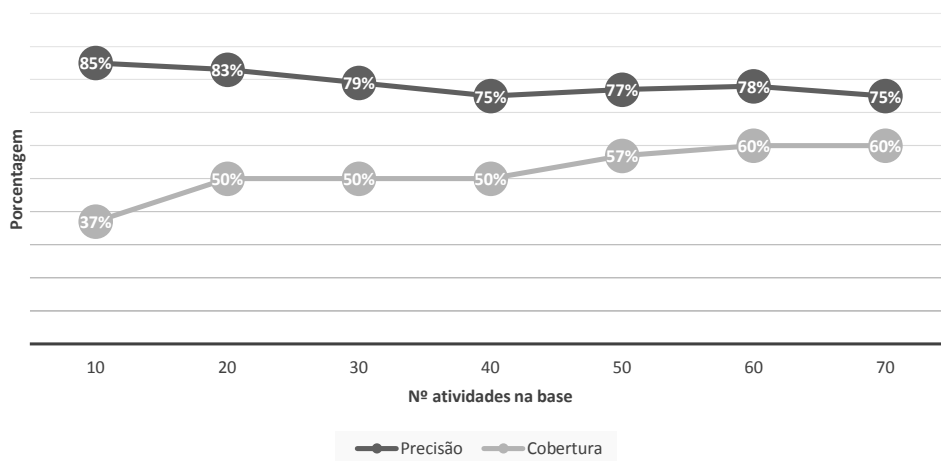


Figura 31 – Resultados obtidos na execução do modelo de testes 2 variando o tamanho da base para as atividades de Gestão.

Gráfico que demonstra a Precisão e Cobertura das Atividades do Grupo de Gestão com base em diferentes nº de atividades presentes na base de conhecimento.



já foi demonstrado na [seção 5.2](#). O valor máximo de atividades possíveis de alimentar a base foi 48, em razão de $48 + 30$ resultar em 78 atividades e, na base de ACGs separada a esse grupo, conter o total de 80 atividades.

Os grupos de pesquisa e extensão tiverem incrementos idênticos, como pode ser visto nas [Figura 29](#) e [Figura 30](#). Esse fato ocorreu em razão de ambos os grupos possuírem o mesmo número de atividades. Pode-se aumentar a base de conhecimento para até 20 atividades, em virtude de antes do início dos testes ocorrer a separação de 30 para serem testadas, como já foi mencionado anteriormente.

O grupo de gestão destacou-se dos demais por conseguir ter mais atividades classificadas na base de conhecimento. Isto ocorre devido ao grupo de gestão possuir o maior número de exemplares de atividades passíveis a teste.

A primeira variação de teste do modelo 2 foi executado com êxito conforme visto nas figuras anteriores. O objetivo desse teste é demonstrar que quanto maior for a base de atividades disponibilizadas para o aplicativo tentar novas inferências, melhor é a cobertura² e precisão³ dos testes realizados. Entretanto, devido ao conjunto de bigramas disponíveis na base aumentar, é possível que este fator ocasione a inferência equivocada de algumas atividades. Esse resultado negativo se deve ao fator da linguagem ser bastante diversa, como já explicado anteriormente.

O resultado do grupo das atividades de ensino, conforme [Figura 28](#), se iniciou como o esperado, todavia, após o terceiro aumento na base de conhecimento, a precisão começou a cair. Acredita-se que este resultado é ocasionado devido a conter muitos textos com escritas semelhantes, o que acarreta em aumento no número de erros e conseqüentemente no decréscimo da linha de precisão. Porém, a cobertura se mostrou estável, visto que ocorreu pouca variação no gráfico.

As atividades do grupo de pesquisa obtiveram um excelente resultado, pois a linha de cobertura só aumentou conforme o acréscimo de atividades na base de conhecimento e a coluna de precisão, embora ter decaído um pouco em relação aos valores iniciais, acabou se estabilizando e se aproximou bastante da linha de cobertura, o que caracterizaria o comportamento ideal. Crê-se que caso houvessem mais atividades a serem testadas, os resultados seriam ainda mais satisfatórios.

Os resultados obtidos através do teste das atividades do grupo de Extensão foram os piores dessa bateria. A linha de cobertura se manteve estabilizada mesmo com o aumento das atividades na base de conhecimento e a linha de precisão só decaiu em relação ao seu valor inicial. Embora a linha de precisão ter só decaído, a mesma se manteve acima dos 50%, o que não caracteriza um resultado insatisfatório. Para justificar os resultados obtidos nesse grupo, cita-se a semelhança dos textos utilizados para a execução desses

² Representa quantos acertos foram obtidos sobre o total de atividades.

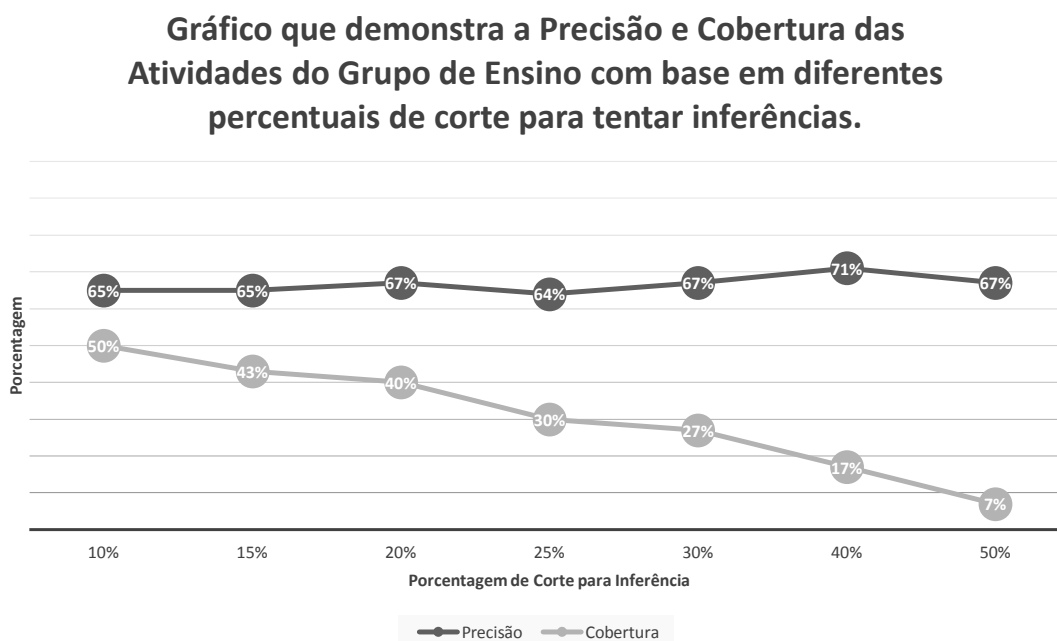
³ Representa quantos acertos foram obtidos sobre o total de atividades inferidas.

testes e a pequena quantidade de atividades presentes na base para auxiliar na inferência de classificações. Acredita-se que caso a base aumentasse seria possível perceber um avanço positivo em relação aos resultados atuais.

As últimas atividades testadas nessa bateria de testes foram as do grupo de Gestão. As mesmas obtiveram os melhores resultados da bateria. A porcentagem de cobertura de acertos das atividades somente aumentou conforme a base de conhecimento fosse crescendo. A linha de precisão manteve-se estável e com a maior média de aproximação em relação a coluna de cobertura, o que seria o comportamento ideal. Possivelmente esse grupo obteve o melhor resultado pois contém mais atividades para a alimentação da base, ou seja, mais possibilidades de inferência. Subentende-se também que os textos das atividades testadas ficaram bem diversificados, mesmo que criados pela mesma pessoa.

A segunda maneira de execução do modelo de testes 2 se baseou em alterar o ponto de corte para tentativa de inferência das atividades. A separação de 30 atividades a serem testadas foi mantida, entretanto, a base de dados sempre foi alimentada com o restante das atividades que sobravam de cada grupo. Os resultados obtidos nesse teste são vistos nas [Figura 32](#), [Figura 33](#), [Figura 34](#) e [Figura 35](#).

Figura 32 – Resultados obtidos na execução do modelo de testes 2 variando a porcentagem de corte para inferência sob o as atividades do grupo de ensino.



A segunda variação do modelo 2 de teste tinha por intuito demonstrar que quanto maior fosse a porcentagem de corte para tentar inferir uma classificação para a atividade, mais aumentaria a precisão das inferências realizadas.

Os testes realizados com o grupo de ensino demonstraram quase o resultado es-

Figura 33 – Resultados obtidos na execução do modelo de testes 2 variando a porcentagem de corte para inferência das atividades do grupo de pesquisa.

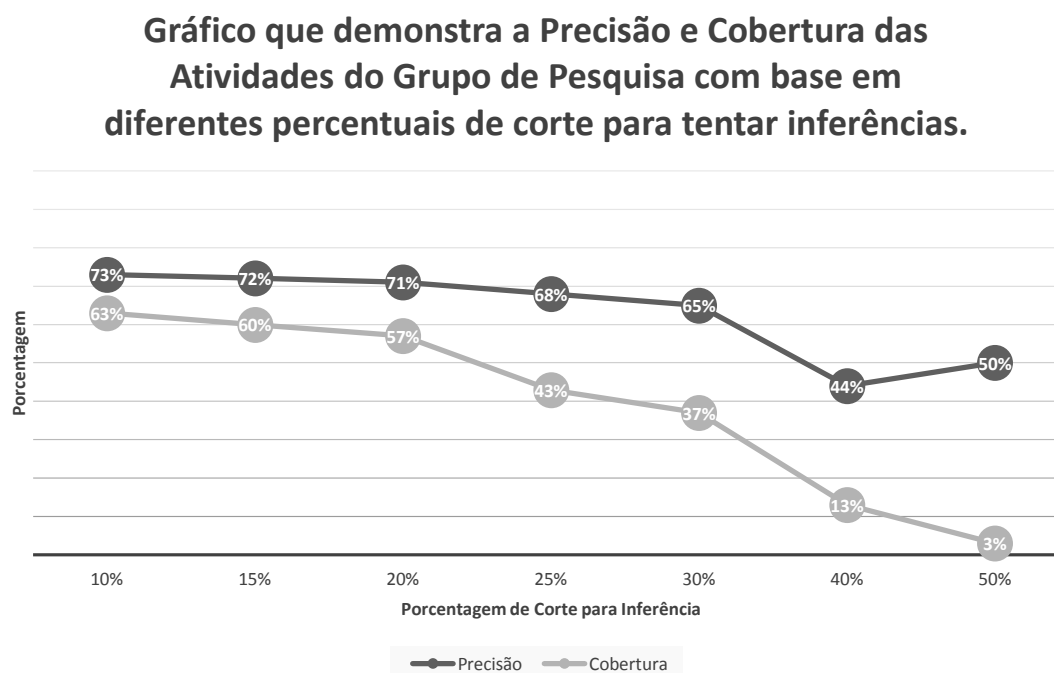
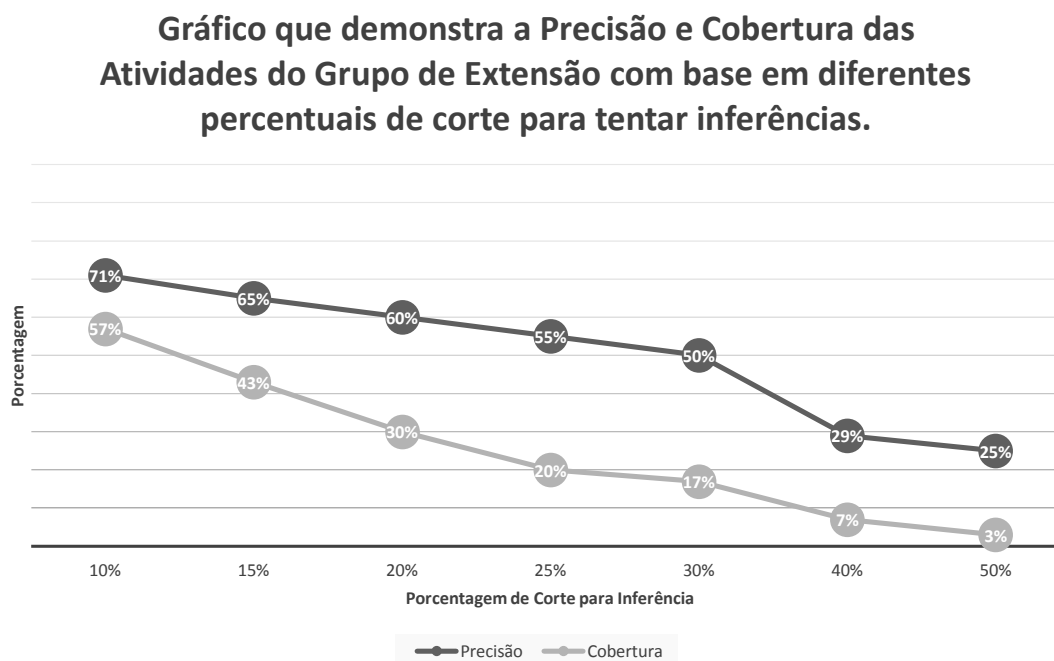
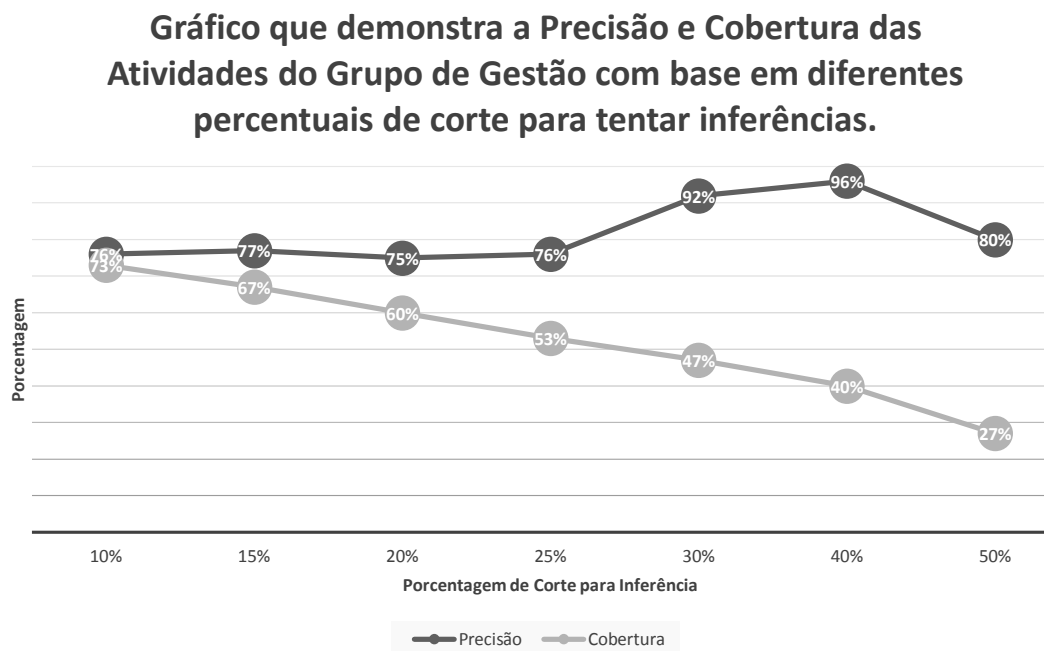


Figura 34 – Resultados obtidos na execução do modelo de testes 2 variando a porcentagem de corte para inferência das atividades do grupo de extensão.



perado. O comportamento ideal seria a cobertura ir diminuindo, conforme aconteceu, porém a precisão deveria ir aumentando ou estabilizando, e é perceptível pela [Figura 32](#)

Figura 35 – Resultados obtidos na execução do modelo de testes 2 variando a porcentagem de corte para inferência das atividades do grupo de gestão.



que a precisão teve alguns picos. Entretanto o resultado não pode ser considerado ruim, pois em nenhum momento a coluna demonstrou variações superiores a 10%.

As atividades de pesquisa não obtiveram um bom resultado, conforme [Figura 33](#). A linha de precisão se manteve estabilizada até o percentual de corte de 20% e após começou a decrementar, o que não deveria ocorrer. Possivelmente essa variação de precisão é resultado do número limitado de atividades existentes na base de teste para auxiliar nas inferências, o qual é somente 20 neste caso.

O grupo de extensão seguiu com o pior resultado também nessa bateria. A medida que aumentava a porcentagem de corte para inferência, o percentual de precisão só decaía, o que não é esperado nesse teste. Acredita-se que esse resultado ocorreu pelo mesmo motivo do anterior, ou seja, excesso de expressões parecidas nos textos analisados, o que acaba por resultar em maior probabilidade de inferências erradas.

As atividades de gestão obtiveram novamente os melhores resultados. Através da [Figura 35](#) é possível visualizar que ao percentual de corte de 40%, a precisão das atividades inferidas atingiu o valor de 95%. É um resultado muito satisfatório. Novamente é demonstrado que quanto mais atividades tiverem disponíveis na base para auxiliar nas inferências, melhor será o resultado obtido.

5.4 Considerações Sobre o Capítulo

Neste capítulo foi apresentado o processo de validação utilizado para testar a eficiência do aplicativo. Através de duas modalidades distintas de teste, foi possível perceber que o aplicativo em questão está apresentando resultados satisfatórios em seu mecanismo de inferência, visto que a média de acertos se comparada a de erros é pelo menos duas vezes maior. Complementarmente, tratando-se de um protótipo desenvolvido em poucos meses, pode-se dizer que o mesmo já apresenta um bom nível de maturidade. Por intermédio dos testes realizados também ficou evidente que quanto mais atividades ingressarem no aplicativo, melhor será o seu resultado. Entretanto, algumas melhorias no mecanismo de inferência do aplicativo certamente aumentariam a precisão dos resultados.

6 Considerações Finais

Este trabalho teve como objetivo desenvolver um aplicativo *smart* usual através de dispositivos móveis para apoio aos alunos na coleta e classificação de suas ACGs. Pretende-se com a utilização do aplicativo *smartACG* facilitar o cotidiano dos alunos além de introduzir uma caminhada para a informatização do processo de aproveitamento de ACGs, o qual atualmente é todo feito de forma manual.

A pesquisa realizada na etapa de análise do projeto foi muito importante para esse trabalho, pois através da mesma se conseguiu descobrir as dificuldades que os alunos vinham enfrentando no processo de aproveitamento de suas ACGs. Posteriormente, foram desenvolvidas algumas funcionalidades, detalhadas no [Capítulo 4](#), que objetivaram atender a maioria das solicitações relatadas na pesquisa.

A construção do aplicativo SmartACG deixou claro a viabilidade de melhorias no processo atual de aproveitamento de ACGs na UNIPAMPA. Apesar de resultados falso positivos e não deduções, a quantidade de inferências corretas, em relação as classificações automáticas de atividades ingressantes no sistema, é relativamente alta e, diante disso, é possível afirmar que o aplicativo é eficaz em realizar as tarefas que foram designadas ao mesmo, atingindo assim o objetivo principal deste trabalho.

A escolha da linguagem de marcação *XML* proporcionou a conclusão de um dos objetivos específicos constatados neste trabalho, devido a essa ser uma linguagem genérica e viável de ser interpretada por diferentes sistemas computacionais. O *XML* também proporcionou uma melhor forma de descrever as atividades disponibilizadas, por parte dos *promoters*.

Para a realização desse trabalho era necessário escolher uma tecnologia que viabilizasse que as informações da atividade estivessem descritas no seu ambiente de realização. Com isso, escolheu-se o *QRCode*, visto que o mesmo é uma boa tecnologia para codificar informações, além de ser plausível a sua exposição em contextos distintos. A capacidade da maioria dos dispositivos móveis existentes conseguirem decodificar *QRCodes*, foi fator decisivo para essa preferência.

O impasse maior na realização do projeto era modelar uma solução que se adequasse a todos os cursos do campus Alegrete. Entretanto, no ano de 2009 a UNIPAMPA disponibilizou as normas acadêmicas institucionais, disponível no ([UNIPAMPA, 2014a](#)), que prevê algumas regras por meio de seus artigos para que sejam criadas as tabelas de ACGs dos cursos da UNIPAMPA. Consequentemente a esse fato, a solução modelada foi baseada nas regras presentes nessa norma e com isso outro objetivo do trabalho foi atingido com êxito.

Foram encontradas algumas dificuldades no desenvolvimento do trabalho, relacionadas à tecnologia *smart* e a quantidade reduzida de materiais disponíveis sobre a mesma. O processo de inferência das atividades necessitou ser estudado em detalhes, visto que era preciso encontrar um mecanismo de inferência que fosse ao menos satisfatório para que não houvesse dúvidas sobre o benefício do trabalho. Entretanto, como é perceptível nos testes realizados na [seção 5.3](#), o mecanismo escolhido se encontra em um bom nível de maturidade.

6.1 Trabalhos Futuros

O aplicativo desenvolvido para este trabalho cobre o domínio do processo de aproveitamento de ACGs referente ao aluno. Entretanto, atualmente existe a carência de um sistema que informatize o processo na visão do coordenador de curso e da secretária acadêmica.

O desenvolvimento do SmartACG pode ser considerado como um passo inicial para a caminhada em busca da informatização do processo de aproveitamento de ACGs na UNIPAMPA. Contudo, é imprescindível que ocorra mudanças em sua forma de inferência para que o aplicativo apresente resultados mais satisfatórios se utilizados pelos alunos. Além da mudança sugerida, é aconselhado também que ocorra a inserção de um mecanismo capaz de interpretar uma ontologia, eliminando a necessidade da atividade ser descrita através de um formulário.

Acredita-se que a construção de um sistema web poderá cobrir tanto o domínio do *promoter* da atividade, o qual precisa descrevê-la, quanto o do coordenador do curso. O mesmo precisa receber o pedido de aproveitamento da atividade realizada pelo aluno para posteriormente analisá-la a fim de validar a mesma. A parte do sistema que cobre o contexto do *promoter* poderia ser transformado em um portal semântico que visa auxiliar os descritores das atividades a compartilharem ontologias.

O sistema sugerido para o domínio do coordenador e da secretaria acadêmica poderá trazer melhorias nos processos atuais, os quais são executados de forma manual em ambos os casos. Na visão do coordenador, o sistema poderá trazer as atividades realizadas pelos alunos para que o mesmo aprove ou não essas atividades. Na perspectiva da secretaria acadêmica, o sistema poderá enviar as ACGs já aproveitadas pelo coordenador para que os funcionários da secretaria só vinculem as horas aproveitadas no currículo escolar do aluno solicitante.

Referências

- ABOWD, G. D.; MYNATT, E. D. Designing for the human experience in smart environments. *Smart environments: technologies, protocols, and applications*, Wiley Online Library, p. 151–174, 2004. Citado na página 24.
- ANDROID, G. *App Components*. 2013. Disponível em: <<http://developer.android.com/guide/components/index.html>>. Citado na página 25.
- ANDROID, G. *Get the Android SDK*. 2013. Disponível em: <<http://developer.android.com/sdk/index.html>>. [Acesso em: 02 set.2013]. Citado 2 vezes nas páginas 23 e 29.
- BRAUDE, E. *Projeto de software: da programação à arquitetura: uma abordagem baseada em Java*. [S.l.]: Bookman, 2005. Citado na página 44.
- BRUS, J. et al. Design of intelligent system in cartography. In: IEEE. *Roedunet International Conference (RoEduNet), 2010 9th*. [S.l.], 2010. p. 112–117. Citado 3 vezes nas páginas 34, 35 e 36.
- CARDOSO, O. N. P. Recuperação de informação. *Lavras, [sd]*, 2000. Citado na página 30.
- COOK, D. J.; AUGUSTO, J. C.; JAKKULA, V. R. Ambient intelligence: Technologies, applications, and opportunities. *Pervasive and Mobile Computing*, Elsevier, v. 5, n. 4, p. 277–298, 2009. Citado na página 23.
- CRUTZEN, C. K. Invisibility and the meaning of ambient intelligence. *International Review of Information Ethics*, Citeseer, v. 6, n. 12, p. 52–62, 2006. Citado na página 23.
- DAS, S. K. et al. The role of prediction algorithms in the mavhome smart home architecture. *Wireless Communications, IEEE, IEEE*, v. 9, n. 6, p. 77–84, 2002. Citado na página 24.
- GENG, Y.; CASSANDRAS, C. G. A new “smart parking” system based on optimal resource allocation and reservations. In: IEEE. *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*. [S.l.], 2011. p. 979–984. Citado 2 vezes nas páginas 35 e 36.
- GONÇALVES, J. C. Uso da plataforma android em um protótipo de aplicativo coletor de consumo de gás natural. Curitiba, 2012. Citado na página 25.
- HERCZEG, M. The smart, the intelligent and the wise: roles and values of interactive technologies. In: ACM. *Proceedings of the First International Conference on Intelligent Interactive Technologies and Multimedia*. [S.l.], 2010. p. 17–26. Citado 3 vezes nas páginas 19, 23 e 24.
- KBAR, G.; ALY, S.; ALHANAFI, B. Smart and context aware search for university campus (scasuc). In: IEEE. *Multimedia Computing and Systems (ICMCS), 2012 International Conference on*. [S.l.], 2012. p. 809–814. Citado 2 vezes nas páginas 34 e 36.

- KLOCH, C.; KRISTENSEN, J. E.; BILSTRUP, B. Future scenarios: What are the future services and applications? *Wireless personal communications*, Springer, v. 53, n. 3, p. 315–327, 2010. Citado na página 19.
- LECHETA, R. R. *Google Android - Aprenda a criar aplicações para dispositivos Móveis com o Android SDK*. [S.l.]: Editora Novatec, 2009. Citado 2 vezes nas páginas 28 e 29.
- MIYAZAKI, E. et al. Development of an automatic health screening system for student health education of university. In: IEEE. *Digital Information and Communication Technology and it's Applications (DICTAP), 2012 Second International Conference on*. [S.l.], 2012. p. 421–427. Nenhuma citação no texto.
- NAJAFI, M. et al. Virtual remote nursing system. In: IEEE. *Consumer Communications and Networking Conference (CCNC), 2011 IEEE*. [S.l.], 2011. p. 13–17. Citado na página 24.
- O'NEILL, M. N. H. *Fundamental de uml*. Lisboa: FCA-Editora de Informática, 2004. Citado na página 47.
- PEREIRA, L. C. O.; SILVA, M. L. D. *Android para desenvolvedores*. [S.l.]: Brasport, 2009. Citado 2 vezes nas páginas 28 e 29.
- PEROZZO, R. F.; PEREIRA, C. E. Ambientes inteligentes: Uma arquitetura para cenários de automação predial/residencial baseada em experiências. *Encontro de TI e Comunicação na Construção Civil, POA, 7f*, 2007. Citado na página 23.
- RASMUSSEN, B. *Os números não mentem: Android ou iOS, qual é o melhor?* 2013. Disponível em: <<http://corporate.canaltech.com.br/analise/mobile/Os-numeros-nao-mentem-Android-ou-iOS-qual-e-o-melhor/>>. [Acesso em: 04 mar.2014]. Citado na página 27.
- SALTON, G. On the use of term associations in automatic information retrieval. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of the 11th conference on Computational linguistics*. [S.l.], 1986. p. 380–386. Citado na página 30.
- SCHON, D.; KOPF, S.; EFFELSBURG, W. A lightweight mobile quiz application with support for multimedia content. In: IEEE. *e-Learning and e-Technologies in Education (ICEEE), 2012 International Conference on*. [S.l.], 2012. p. 134–139. Citado 2 vezes nas páginas 33 e 35.
- SILVA, A. C. *Unified Modeling Language (UML)*. 2013. Disponível em: <http://www.deinf.ufma.br/~acmo/MOO_Seq.pdf>. [Acesso em: 09 mar.2014]. Citado na página 52.
- TROYACK, L. *Android bate recorde de uso e Windows Phone se estabiliza em terceiro lugar*. 2013. Disponível em: <<http://codigofonte.uol.com.br/noticias/android-bate-recorde-de-uso-e-windows-phone-se-estabiliza-em-terceiro-lugar>>. [Acesso em: 04 mar.2014]. Citado na página 27.
- UNIPAMPA. *Normas Acadêmicas*. 2014. Disponível em: <http://porteiros.unipampa.edu.br/livramento/index.php?option=com_docman&task=doc_download&gid=51&Itemid=64>. [Acesso em: 23 fev.2014]. Citado na página 83.

- UNIPAMPA. *Tabela para Cômputo de Horas de Atividades Complementares de Graduação do curso de Engenharia de Software*. 2014. Disponível em: <<http://cursos.unipampa.edu.br/cursos/engenhariadesoftware/files/2013/04/ES-TabelaACG.pdf>>. [Acesso em: 23 fev.2014]. Nenhuma citação no texto.
- VASILAKOS, A.; PEDRYCZ, W. *Ambient intelligence, wireless networking, and ubiquitous computing*. [S.l.]: Artech House Norwood, 2006. Citado na página 23.
- WANG, S.-L.; WU, C.-Y. Application of context-aware and personalized recommendation to implement an adaptive ubiquitous learning system. *Expert Systems with applications*, Elsevier, v. 38, n. 9, p. 10831–10838, 2011. Citado 2 vezes nas páginas 34 e 35.
- WASSERMAN, T. Software engineering issues for mobile application development. *FoSER 2010*, 2010. Citado 2 vezes nas páginas 25 e 26.
- WEISER, M. The computer for the twenty-first century (pp. 94–100). *Scientific American, September Issue*, 1991. Citado na página 23.
- YIM, J.; WOO, J. Design of a smart vod system. 2013. Citado na página 24.

Apêndices

APÊNDICE A – Formulário de Pesquisa de Opinião

Formulário de Pesquisa de Opção

Este formulário tem por finalidade coletar sugestões de funcionalidades que servirão de apoio para o desenvolvimento de um aplicativo que auxiliará o aluno no gerenciamento de suas Atividades Complementares de Graduação (ACGs).
**Obrigatório*

Nome Completo: *
Digite seu nome completo.

Esta pergunta é obrigatória

Curso: *
Marque o Curso que você está cursando.

Qual semestre você está cursando atualmente? *
Marque o semestre que você está cursando.


Você já realizou em algum momento o aproveitamento de suas ACGs? *
Marque sim caso já tenha aproveitado alguma vez suas ACGs e não caso contrário.
 Sim
 Não

Caso tenha aproveitado, quais foram as dificuldades encontradas no processo?
Responda essa pergunta somente se você já aproveitou suas ACGs pelo menos uma vez.

Definindo o Contexto.
Imagine-se em um cenário perfeito, onde através da tecnologia é possível interligar aparelhos eletrônicos de forma que estes otimizem as suas atividades cotidianas. No contexto acadêmico, que funcionalidade você gostaria que um sistema de informação realize por você de maneira que facilitasse o gerenciamento de suas ACGs?

Sugestão: *
Digite sua sugestão de funcionalidade com base no paragrafo de definição do contexto.

Nunca envie senhas em formulários do Google.

Powered by  Este conteúdo não foi criado nem aprovado pelo Google.
[Denunciar abuso](#) - [Termos de Serviço](#) - [Termos Adicionais](#)

APÊNDICE B – Respostas Obtidas na pesquisa de Opinião Realizada

Indicação de data e hora	Curso:	Qual semestre você está cursando atualmente?	Você já realizou em algum momento o aproveitamento de suas ACGs?	Caso tenha aproveitado, quais foram as dificuldades encontradas no processo?	Sugestão:
05/09/2013 11:02:35	Engenharia Civil	4º Semestre	Não	Tenha quase 120 vinte horas de ACGs e 1 ano de pesquisa, mas vim de outra instituição e não pode até o momento aproveitar tudo que fiz!	Deveriam aceitar, pois afinal faz parte da vida acadêmica do aluno independente de ter sido de transferência! No meu caso, caso minha sugestão não seja aceita vou contestar junto ao MEC, pois é meu direito! Fazer novas normas é aceitável desde que não aflijam os direitos do aluno segundo a constituição federal!!!
05/09/2013 15:02:48	Engenharia Elétrica	10º Semestre	Sim	Nenhuma dificuldade.	Banco de horas.
05/09/2013 15:05:52	Engenharia Elétrica	7º Semestre	Sim	Encontrar os formulários e normas online com facilidade e falta de opção por preenchimento e submissão online com posterior acompanhamento no mesmo modo.	Submissão online, com upload dos devidos comprovantes necessários. Recibo de entrega e extrato de situação do deferimento e aproveitamento das ACGs submetidas (como no histórico do portal do aluno). Interligação com o sistema para que a secretaria acadêmica pudesse acompanhar e lançar sem dificuldades as horas/ créditos aproveitados.
05/09/2013 15:06:48	Ciências da Computação	3º Semestre	Sim	No primeiro aproveitamento não tive dificuldades, entretanto, no segundo houve mudança na coordenação do curso e no fim o pedido se perdeu e precisei pedir novamente no semestre atual...	Uma ferramenta web na qual é possível enviar o formulário já seria de grande ajuda.
05/09/2013 15:11:16	Engenharia de Software	5º Semestre	Sim	Encontrei dificuldades em levantar toda a documentação necessária para entregar ao meu coordenador.	Sim, em minha opinião seria interessante um ambiente web que fosse possível enviar documentos comprobatórios de acgs, o que agilizaria o processo de análise da coordenação do curso.

Indicação de data e hora	Curso:	Qual semestre você está cursando atualmente?	Você já realizou em algum momento o aproveitamento de suas ACGs?	Caso tenha aproveitado, quais foram as dificuldades encontradas no processo?	Sugestão:
05/09/2013 15:11:48	Engenharia de Software	7º Semestre		<p>A primeira delas é burocrática: poder entregar a solicitação apenas em uma semana no semestre (impresa e em mãos na secretaria acadêmica), sendo que poderia ser entregue durante o semestre e haver uma data limite para avaliação (as que forem, por exemplo, entregues em data posterior seriam avaliadas no semestre seguinte).</p> <p>A segunda delas é tecnológica: a solicitação poderia ser realizada através de um software, o que geraria economia em impressão (a maioria dos certificados de participação em eventos já são emitidos digitalmente) e também de espaço (porque as solicitações necessitam ser arquivadas).</p> <p>A terceira delas é relacionada ao feedback: ao solicitar o aproveitamento de ACG a única informação que aparece no histórico escolar é o total de horas computadas, não tem como saber se alguma das solicitações foi negada e o porquê.</p>	<p>Existe uma tabela para o cômputo de horas para cada atividade e uma norma informando os comprovantes necessários. Essas informações poderiam ser cruzadas e em um sistema de solicitação, de acordo com a atividade poderiam aparecer campos específicos (e obrigatórios) para a submissão de comprovantes de acordo com a atividade, evitando que solicitações sejam negadas por falta de comprovantes.</p>
05/09/2013 15:14:07	Engenharia de Software	5º Semestre	Sim	<p>Não tenho uma idéia formalizada</p>	
05/09/2013 15:22:11	Engenharia Agrícola	7º Semestre	Sim	<p>demora aprovação do coordenador e ser lançado no sistema</p>	<p>Não faço a minima ideia.</p>

Indicação de data e hora	Curso:	Qual semestre você está cursando atualmente?	Você já realizou em algum momento o aproveitamento de suas ACGs?	Caso tenha aproveitado, quais foram as dificuldades encontradas no processo?	Sugestão:
05/09/2013 15:47:09	Engenharia de Software	7º Semestre	Sim	<p>1- Definir com certeza a que categoria de atividade a solicitação pertence.</p> <p>2- Em algumas situações que documento seria o mais ideal para comprovar que tal atividade foi realizada. Principalmente atividades do Grupo IV - Atividades Culturais, Artísticas, Sociais e Gestão.</p> <p>3- A burocracia que sempre é um problema de difícil solução.</p>	<p>1- Que o sistema separasse as solicitações de acordo com a sua classificação e calculasse a carga horária atingida na solicitação, para comparar depois com o que foi aceito ou não.</p> <p>2- Gostaria que o sistema classificasse de forma automática a solicitação de acordo com a tabela de ACGs.</p> <p>3- Com os dados da solicitação informados, o sistema poderia fazer uma triagem antes mesmo do envio da solicitação, identificando possíveis problemas ou inconsistências (número de horas máximo por exemplo).</p> <p>4- O sistema poderia avisar os prazos de submissão de ACGs com antecedência, porque sempre nos esquecemos dos prazos, ou acabamos deixando para a última hora.</p>
05/09/2013 16:00:44	Engenharia Elétrica	9º Semestre	Sim	<p>Nenhuma dificuldade.</p> <p>O processo de solicitação de ACG não é muito claro. O formulário disponibilizado é difícil de ser preenchido pelas atividades realizadas. Não é possível saber o andamento da solicitação de maneira clara. Há informação insuficiente para colocar as atividades em seus grupos de ACG corretos.</p>	<p>Contabilização das horas das ACGs.</p>
05/09/2013 16:03:56	Engenharia de Software	7º Semestre	Sim		<p>Que a ferramenta descubra o evento que estou participando, preencha automaticamente o formulário de solicitação de ACG, e envie pra secretaria acadêmica.</p> <p>Gostaria que quando participarmos de algo, seja palestra, cursos, pesquisa, projetos, o certificado ser gerado no ato e consequentemente já inserir no sistema. Se cada orientador tivesse acesso ao sistema para colocar a ACG, ficaria mais fácil e não teria que precisar de tantos documentos.</p>
05/09/2013 16:13:19	Engenharia Civil	7º Semestre	Não		<p>Um sistema, onde eu entraria com dados sobre as ACGs que já tenho, então este mostraria em que área estas atividades se encaixam e qual a equivalência de horas, também poderia mostrar as atividades que estão faltando com sugestão para completá-las.</p>
05/09/2013 16:26:23	Engenharia Mecânica	7º Semestre	Não		

Indicação de data e hora	Curso:	Qual semestre você está cursando atualmente?	Você já realizou em algum momento o aproveitamento de suas ACGs?	Caso tenha aproveitado, quais foram as dificuldades encontradas no processo?	Sugestão:
05/09/2013 16:33:00	Engenharia Civil	7º Semestre	Não		Sabendo a instituição na qual voce faz parte, os congressos e outras instituições ao invés de te entregar o certificado, poderia enviar direito para o sistema da universidade, que faria a analise e contabilização das horas.
05/09/2013 16:40:01	Engenharia Civil	9º Semestre	Não		No momento da confirmação do código de barras do crachá da semana acadêmica, automaticamente as horas da palestra/curso serem adicionadas ao histórico.
05/09/2013 16:52:00	Engenharia de Software	7º Semestre	Não		Em um cenário perfeito, minhas ACGs se atualizariam automaticamente quando eu participasse de aulas e conferências à distância. O sistema também poderia ser notificado pela ETS quando eu fizesse o TOEFL ou o GRE.
05/09/2013 17:03:34	Engenharia Civil	7º Semestre	Não		tabela de equivalência ou conversão das atividades/control de horas/registro das atividades já realizadas
05/09/2013 17:20:44	Engenharia de Software	7º Semestre	Sim	O maior problema é ter que juntar vários documentos, sendo que, por algum motivo perca algum, teria inúmeros problemas para conseguir novos comprovantes.	Seria interessante, no momento do evento coletar as informações do moesmo, após isso o sistema, deveria possuir um local onde o usuario teria um ambiente onde ele iria visualizar todos eventos que coletou no evento e poderia imprimir um relatório
05/09/2013 17:42:53	Engenharia Mecânica	7º Semestre	Sim		Fossem contabilizadas em tempo real as cargas horárias das ACGs realizadas.
05/09/2013 18:10:56	Engenharia Civil	7º Semestre	Não		Acredito que seria interessantes se todos os certificados fossem online e a partir de algum código gerado por eles, calcularia horas de ACG e ligaria ao nome da pessoa. Caso o certificado não seja online criasse alguma aba para acrescenta-los.
05/09/2013 18:19:31	Engenharia Mecânica	5º Semestre	Não		poderia ter um programa ou um sistema junto ao moodle, que todo aluno poderia administrar sua graduação toda não só as ACGs, mas sim tbm as cadeiras que faltam e as que "trancam" outras e DCGs, que tenha todo o cronograma do aluno, assim o aluno terá mais noção de o que faltam de ACGs, DCGs e cadeiras a cursar.

Indicação de data e hora	Curso:	Qual semestre você está cursando atualmente?	Você já realizou em algum momento o aproveitamento de suas ACGs?	Caso tenha aproveitado, quais foram as dificuldades encontradas no processo?	Sugestão:
05/09/2013 18:53:02	Engenharia de Software	3º Semestre	Não		<p>O sistema poderia prover uma forma de verificar o status do progresso das ACGs realizadas em forma de gráficos. Esses gráficos poderiam mostrar as ACGs realizadas, em horas, e mostrar a quantidade de horas restantes necessárias para concluir o curso.</p> <p>O sistema poderia ter uma forma de visualizar as ACGs realizadas de acordo com o semestre. Seria como um histórico, onde o aluno verificaria as ACGs que realizou em algum semestre passado.</p>
05/09/2013 19:09:22	Engenharia de Software	1º Semestre	Sim		<p>Não entendi a pergunta, sei lá</p> <p>Um sistema simples, em que o aluno preencha os dados da sua atividades, envie através deste sistema o comprovante (certificado) destas atividades, sem necessidade de papel, e que estas atividades sejam avaliada pelo responsável das ACGs do curso em qualquer momento e sendo aproveitadas no período do calendário acadêmico.</p> <p>Tudo sem necessidade de papel, e com maior rapidez! Um sistema totalmente interligado, em que no momento em que o aluno envia no sistema o comprovante e preencha o formulário, o docente responsável analisa e encaminha o deferimento/deferimento destas atividades em um prazo determinado de tempo para a Secretaria Acadêmica. Mas para maior eficiência deste aplicativo/sistema as normas de ACGs dos cursos de graduação deverão estar atualizadas e seus respectivos PPCs.</p>
05/09/2013 21:55:14	Engenharia Elétrica	5º Semestre	Não	<p>Não aproveitei, pois devido a demora no aproveitamento, vou deixar para o último semestre da graduação.</p>	
05/09/2013 22:53:47	Engenharia Civil	5º Semestre	Não		<p>Indicasse em qual grupo (ensino, pesquisa, etc.) e modalidade (participação, monitoria, publicação, etc) se encontra o certificado de horas que possui.</p>

Indicação de data e hora	Curso:	Qual semestre você está cursando atualmente?	Você já realizou em algum momento o aproveitamento de suas ACGs?	Caso tenha aproveitado, quais foram as dificuldades encontradas no processo?	Sugestão:
05/09/2013 23:21:06	Ciências da Computação	5º Semestre	Sim	Não digo que foi dificuldade, mas tive que escrever com letra mais legível o possível, preencher o código e fazer a equivalência de horas conforme a tabela de ACGs.	Poderia o campo de tipo de atividades ser uma caixa de seleção que possibilitasse escolher dentre um rol de atividades na listas. Informar a carga horária da atividade conforme certificado/atestado e realizar a equivalência de horas de maneira automática e realizar o controle de restrições caso o número máximo de horas fosse utilizado. Caso por alguma questão legal não pudesse ser informatizado a solicitação de ACGs, poderia ter a possibilidade de imprimir um relatório. Uma tecnologia com este fim seria muito boa, porém não faço idéia das dificuldades que a instituição tem para validar as ACGs. Eu gostaria mesmo é que essa pergunta fosse feita à coordenação e secretaria acadêmica... Abraço, boa sorte.
05/09/2013 23:43:01	Engenharia de Telecomunicações	3º Semestre	Não		Um sistema que mostre o quanto de carga horária de ACG possui e quanto falta para completar a carga mínima, e que auxiliasse no registro da carga horária da participação de eventos.
05/09/2013 23:56:49	Engenharia de Software	5º Semestre	Não		Seria interessante um sistema que informasse quando houvesse uma ACG que fosse compatível com minha área de interesse; que conseguisse dizer quantas horas faltam para completar as ACGs.
05/09/2013 23:59:18	Engenharia Mecânica	5º Semestre	Não		Talvez um simples formulário que tenha a opção de realizar um upload com o certificado, em um sistema parecido com o moodle talvez já otimize bastante o gerenciamento das ACGs.
06/09/2013 00:04:37	Ciências da Computação	5º Semestre	Sim	Não tive dificuldades, o coordenador do curso me explicou como funcionava o processo e foi tranquilo.	

Indicação de data e hora	Curso:	Qual semestre você está cursando atualmente?	Você já realizou em algum momento o aproveitamento de suas ACGs?	Caso tenha aproveitado, quais foram as dificuldades encontradas no processo?	Sugestão:
06/09/2013 08:15:54	Engenharia Elétrica	6º Semestre	Sim	<p>Inicialmente temos que imprimir uma cópia de solicitação para cada ACG, mesmo sendo para a mesma coisa. EX: Participação em palestras. Processo demorado.</p> <p>Quando entreguei a documentação demorou mais de seis meses para lançarem as horas no sistema.</p> <p>Quando foram lançadas alguém da secretaria acadêmica não digitou todas horas contidas nos documentos entregues. Somente após eu pedir o motivo das horas não aparecem perceberam o erro.</p>	<p>Gostaria que os documentos entregues pudessem ser visualizados e que eu conseguisse visualizar quais aproveitamentos tive, em quais áreas, além de um gerenciamento indicando quais áreas já estão completas ou quais áreas posso fazer novos aproveitamentos.</p> <p>Entrar no sistema GURI e anexar os documentos digitalizados para serem avaliados por um responsável, o qual irá validar as horas.</p> <p>Sim. Uma espécie de leitor (scanner) onde os certificados fossem lidos, avaliados pelo coordenador do curso e já entrassem no histórico do aluno. Tudo pela ferramenta, sem precisar marcar horário para avaliar as ACGs.</p>
06/09/2013 08:51:20	Engenharia Elétrica	6º Semestre	Não		
06/09/2013 12:19:51	Engenharia Mecânica	5º Semestre	Não		
06/09/2013 16:13:26	Engenharia de Software	3º Semestre	Não		Registrar as ACGs
06/09/2013 21:39:35	Engenharia Civil	7º Semestre	Não		<p>Um programa que eu pudesse validar organizar de maneira simples minhas acg's.</p> <p>Seria bom ter um assistente mais inteligente para ajudar na localização do tipo de ACG e tempo aproveitável assim como sugerir opções para obter ACGs</p>
06/09/2013 22:35:30	Engenharia de Software	7º Semestre	Não	<p>cada curso possui sua tabela de equivalencia de acgs, a demora para contabilizar a carga horaria, a má vontade da secretaria e o preenchimento de artefatos. se fossem formulários digitais com campo para anexo facilitaria muito!</p>	<p>o desenvolvimento de um formulário com campo para anexo, onde o anexo é o comprovante do evento ou da área de atuação para o ganho da acg, a tabela deveria ser carregada automaticamente de acordo com o curso do aluno, sabendo que os cursos possuem valores diferentes para cada hora de determinado tipo de evento.</p>
07/09/2013 00:01:04	Ciências da Computação	8º Semestre	Sim		

Indicação de data e hora	Curso:	Qual semestre você está cursando atualmente?	Você já realizou em algum momento o aproveitamento de suas ACGs?	Caso tenha aproveitado, quais foram as dificuldades encontradas no processo?	Sugestão:
08/09/2013 16:19:08	Engenharia de Software	3º Semestre	Não		Informações de o porque da minha participação naquela ACG, um calendário com o máximo de tempo prolongado de informações das ACGs e para qual curso se aproveitaria, horas etc.
08/09/2013 17:23:13	Engenharia de Software	5º Semestre	Não		Um aplicativo que opinasse sobre eventos que tem maior carga de horário
08/09/2013 17:23:51	Engenharia de Software	7º Semestre	Não		Seria interessante se ACG's de eventos que acontecem na UNIPAMPA fossem adicionadas automaticamente ao meu currículo, por exemplo as acgs da semana acadêmica, do SIEPE e de visitas técnicas. Ao emitir o certificado da ACG a mesma poderia ser adicionada ao componente curricular ou enviada para quem possa fazer isso, sem a necessidade de ir lá na secretária preencher milhares de formulários.
08/09/2013 21:58:08	Engenharia de Software	7º Semestre	Não	Ainda não realizei aproveitamento, mas por causa de toda burocracia envolvida.	Acredito que um sistema de fácil utilização onde fosse possível o envio dos dados necessários para as solicitações de ACGs assim como a possibilidade de envio de anexos, como os certificados PDF de eventos iriam facilitar bastante.

Anexos

ANEXO A – Tabela de ACGs - Curso Engenharia de Software

**UNIVERSIDADE FEDERAL DO PAMPA
CAMPUS ALEGRETE
CURSO DE ENGENHARIA DE SOFTWARE**

TABELA DE ATIVIDADES COMPLEMENTARES DE GRADUAÇÃO (ACG)

ACG	CH min	Atividades	Carga-horária	Observações	Código
Grupo I - Atividades Ensino	30 h	Disciplinas, desde que aprovado pela Comissão de curso	1h	para cada 1h da disciplina	1.1
		Cursos na área de interesse em função do perfil do egresso	2h/1h	2h para cada 1h de curso presencial e 1h para cada 1h de curso a distância, no máximo 75h por semestre e por curso	1.2
		Monitorias em disciplinas na UNIPAMPA	1h	para cada 1h de atividade, no máximo 75h por semestre	1.3
		Participação em projetos de ensino ou em grupos de ensino/estudo (Por exemplo: PET, GEINFOEDU)	1h	para cada 1h de atividade, no máximo 75h por semestre	1.4
		Estágio não obrigatório ligados a atividades de ensino	1h	para cada 1h de estágio, máximo de 75h por semestre	1.5
		Organização de eventos de ensino	10h	por dia de evento	1.6
		Participação como ouvinte em eventos de ensino, pesquisa e extensão	1h	para cada 1h de evento local	1.7.1
			2h	para cada 1h de evento regional	1.7.2
			4h	para cada 1h de evento nacional	1.7.3
			5h	para cada 1h de evento internacional	1.7.4
			4h	por visita no município	1.8.1
		Visitas técnicas institucionais (fora das atividades normais das disciplinas de graduação)	15h	por visita na região (< 250 Km)	1.8.2
			30h	por visita no estado (> 250Km)	1.8.3
			50h	por visita fora do estado	1.8.4
Outras a critério da Comissão de Curso			1.9		

Grupo II - Atividades de Pesquisa	30 h	Participação em projeto de pesquisa na Unipampa ou em outras IES ou em espaço de pesquisa reconhecido como tal	1h	para cada 1h de atividade, no máximo 75h por semestre	2.1
		Publicação de pesquisa em evento científico ou publicação em fontes de referência acadêmica, impressa ou de acesso online, na forma de livros, capítulos de livros, periódicos, anais, jornais, revistas, vídeos ou outro material de referência acadêmica;	250h	Livro	2.2.1
			150h	periódico científico indexado pela Capes ou Capítulo de livro	2.2.2
			100h	periódico científico não indexado pela Capes	2.2.3
			10h	por página de trabalhos completos publicados em anais de eventos (máximo de 80 horas por trabalho)	2.2.4
			10h	por página de resumos publicados em anais de eventos (máximo de 40 horas por documento)	2.2.5
			20h	jornais, revistas, vídeos ou outro material de referência acadêmica	2.2.6
			40h	Relatórios Técnicos	2.2.7
		Participação na condição de conferencista, ou painalista, ou debatedor, ou com apresentação de trabalho em eventos que tratam de pesquisa, tais como grupos de pesquisa, seminários, congressos, simpósios, semanas acadêmicas, entre outros;	10h	por apresentação em evento local	2.3.1
			20h	por apresentação em evento regional	2.3.2
			30h	por apresentação em evento nacional	2.3.3
			40h	por apresentação em evento internacional	2.3.4
		Estágios ou práticas não obrigatórios em atividades de pesquisa	1h	para cada 1h de estágio, máximo de 75h por semestre	2.4
		Organização de eventos de pesquisa	10h	por dia de evento	2.5
Outras a critério da Comissão de Curso			2.6		

Grupo III - Atividades de Extensão	30 h	Participação em projetos e/ou atividades de extensão desenvolvidos na UNIPAMPA ou outra IES, ou em instituição governamental ou em organizações da sociedade civil com fim educativo, de promoção da saúde, da qualidade de vida ou da cidadania, do desenvolvimento social, cultural ou artístico.	1h	para cada 1h de atividade, no máximo 75h por semestre	3.1
		Estágios e práticas não obrigatórias em atividades de extensão (Por exemplo: PampaTec, Empresa Jr).	1h	para cada 1h de atividade, no máximo 75h por semestre	3.2
		Organização de eventos de extensão.	10h	por dia de evento	3.3
		Publicação de atividade de extensão ou publicação de material pertinente à extensão em fontes de referência acadêmica, impressa ou de acesso online, na forma de livros, capítulos de livros, periódicos, anais, jornais, revistas, vídeos ou outro material de referência acadêmica.	250h	Livro	3.4.1
			150h	periódico científico indexado pela Capes ou Capítulo de livro	3.4.2
			100h	periódico científico não indexado pela Capes	3.4.3
			10h	por página de trabalhos completos publicados em anais de eventos (máximo de 80 horas por trabalho)	3.4.4
			10h	por página de resumos publicados em anais de eventos (máximo de 40 horas por resumo)	3.4.5
			20h	jornais, revistas, vídeos ou outro material de referência acadêmica	3.4.6
		Participação na condição de conferencista, ou painalista, ou debatedor, ou com apresentação de trabalho em eventos que tratam de extensão, como grupos de estudos, seminários, congressos, simpósios, semana acadêmica, entre outros.	10h	por apresentação em evento local	3.5.1
			20h	por apresentação em evento regional	3.5.2
			30h	por apresentação em evento nacional	3.5.3
			40h	por apresentação em evento internacional	3.5.4
Outras a critério da Comissão de Curso.			3.6		

Grupo IV - Atividades Culturais, Artísticas, Sociais e Gestão	30 h	Participação em atividades de cunho cultural, social ou artística	1h	por dia de atividade comprovada, no máximo 15h por semestre	4.1
		Organização de campanhas beneficentes, educativas, ambientais ou de publicidade e outras atividades de caráter cultural, social ou artístico	1h	por 1h de atividade comprovada, no máximo até 75h	4.2
		Premiação referente a trabalho acadêmico de ensino, de pesquisa, de extensão ou de cultura	30h	por distinção ou mérito	4.3
		Representação discente em órgãos colegiados	25h	por portaria ao término da gestão, no máximo 50 horas	4.4
		Representação discente em diretórios acadêmicos	25h	por ano de gestão, no máximo 50 horas	4.5
		Participação, como bolsista, em atividades de iniciação ao trabalho técnico-profissional e de gestão acadêmica	1h	para cada 1h de atividade, no máximo 75h por semestre	4.6
		Participação em estágios não obrigatórios com atividades na área cultural, social, artística e de gestão administrativa e acadêmica	1h	para cada 1h de atividade, no máximo 75h por semestre	4.7
		Cursos de Línguas estrangeiras (presenciais)	2h	para cada 1h de atividade, máximo 75h por semestre	4.8
		Aprovação em exame de proficiência em idioma estrangeiro	50h	por idioma	4.9
		Palestras fora de eventos	10h	por palestra proferida	4.10.1
			1h	por palestra assistida	4.10.2
		Outros a critério da Comissão de Curso	4.11		
Grupo V - Avaliação de Desempenho	60 h	Realização da avaliação de desempenho (ADes).	30h	para cada avaliação de desempenho realizada com desempenho maior ou igual ao estipulado pela comissão de curso. O desempenho mínimo será divulgado anualmente, antes da realização da avaliação.	5.1
		Outros a critério da Comissão de Curso			5.2

ANEXO B – Normas Acadêmicas - Das Atividades Complementares de Graduação

TÍTULO IX DAS ATIVIDADES COMPLEMENTARES DE GRADUAÇÃO

Art. 103 Atividade Complementar de Graduação (ACG) é definida como atividade desenvolvida pelo discente, no âmbito de sua formação humana e acadêmica, com o objetivo de atender ao perfil do egresso da UNIPAMPA e do respectivo curso de graduação, bem como a legislação pertinente.

Art. 104 As atividades complementares classificam-se em 4 (quatro) grupos:

- I. Grupo I: Atividades de Ensino;
- II. Grupo II: Atividades de Pesquisa;
- III. Grupo III: Atividades de Extensão;
- IV. Grupo IV: Atividades Culturais e Artísticas, Sociais e de Gestão.

Art. 105 Cabe à Comissão de Curso analisar e definir no respectivo Projeto Pedagógico do Curso (PPC) a carga horária mínima a ser cumprida pelo discente em ACG, como requisito obrigatório para a integralização curricular e para a colação de grau, considerando-se as diretrizes curriculares nacionais para cada curso e a carga horária mínima de 10% (dez por cento) em cada um dos grupos previstos no artigo 104, incisos I, II, III e IV.

Art. 106 As atividades do GRUPO I – Atividades de Ensino – incluem, entre outras, as seguintes modalidades:

- I. componente curricular de graduação, desde que aprovado pela Comissão do Curso;
- II. cursos nas áreas de interesse em função do perfil de egresso;
- III. monitorias em componentes curriculares de cursos da UNIPAMPA;
- IV. participação em projetos de ensino;
- V. estágios não obrigatórios ligados a atividades de ensino;
- VI. organização de eventos de ensino;
- VII. participação como ouvinte em eventos de ensino, pesquisa e extensão.

Art. 107 As atividades do GRUPO II – Atividades de Pesquisa – incluem, entre outras, as seguintes modalidades:

- I. participação em projetos de pesquisa desenvolvidos na UNIPAMPA, ou em outra IES ou em espaço de pesquisa reconhecido legalmente como tal;
- II. publicação de pesquisa em evento científico ou publicação em fontes de referência acadêmica, impressa ou de acesso online, na forma de livros, capítulos de livros, periódicos, anais, jornais, revistas, vídeos ou outro material de referência acadêmica;
- III. participação na condição de conferencista, ou painalista, ou debatedor, ou com apresentação de trabalho em eventos que tratam de pesquisa, tais como grupos de pesquisa, seminários, congressos, simpósios, semanas acadêmicas, entre outros;
- IV. estágios ou práticas não obrigatórios em atividades de pesquisa.

Art. 108 As atividades do GRUPO III – Atividades de Extensão – incluem, entre outras, as seguintes modalidades:

- I. participação em projetos e/ou atividades de extensão desenvolvidos na UNIPAMPA ou outra IES, ou em instituição governamental ou em organizações da sociedade civil com fim educativo, de promoção da saúde, da qualidade de vida ou da cidadania, do desenvolvimento social, cultural ou artístico;
- II. estágios e práticas não obrigatórios, em atividades de extensão;
- III. organização e/ou participação em eventos de extensão;
- IV. publicação de atividade de extensão ou publicação de material pertinente à extensão em fontes de referência acadêmica, impressa ou de acesso online, na forma de livros, capítulos de livros, periódicos, anais, jornais, revistas, vídeos ou outro material de referência acadêmica;
- V. participação na condição de conferencista, ou painalista, ou debatedor, ou com apresentação de trabalho em eventos que tratam de extensão, como grupos de estudos, seminários, congressos, simpósios, semana acadêmica, entre outros.

Art. 109 As atividades do GRUPO IV – Atividades Culturais e Artísticas, Sociais e de Gestão - incluem, entre outras, as seguintes modalidades:

- I. organização ou participação ou premiação em atividades de cunho cultural, social ou artístico;
- II. participação na organização de campanhas beneficentes, educativas, ambientais ou de publicidade e outras atividades de caráter cultural, social ou artístico;
- III. premiação referente a trabalho acadêmico de ensino, de pesquisa, de extensão ou de cultura;
- IV. representação discente em órgãos colegiados;
- V. representação discente em diretórios acadêmicos;
- VI. participação, como bolsista, em atividades de iniciação ao trabalho técnico-profissional e de gestão acadêmica;
- VII. participação em estágios não obrigatórios com atividades na área cultural, social, artística e de gestão administrativa e acadêmica.

Art. 110 Os critérios de aproveitamento e as equivalências da carga horária nas ACG são definidos pela Comissão de Curso, considerando o perfil do egresso definido em seu PPC.

Art. 111 É de responsabilidade do discente solicitar, na Secretaria Acadêmica, no período informado no Calendário Acadêmico da UNIPAMPA, o aproveitamento das atividades complementares realizadas.

I. o discente deve anexar ao seu requerimento cópia dos documentos comprobatórios, com indicação da carga horária da atividade, autenticados por técnico-administrativo mediante apresentação dos originais.

II. o requerimento é protocolado na Secretaria Acadêmica, em 2 (duas) vias, assinadas pelo discente e pelo técnico-administrativo, onde estão listadas todas as cópias de documentos entregues; uma via é arquivada na Secretaria Acadêmica e a outra entregue ao discente como comprovante de entrega das cópias.

Art. 112 Cabe à Coordenação de Curso de Graduação validar ou não o aproveitamento da ACG requerida pelo discente, de acordo com documentos comprobatórios e os critérios estabelecidos pela Comissão de Curso.

Art. 113 Fica a cargo da Secretaria Acadêmica o registro do aproveitamento da ACG no Histórico Escolar do discente conforme deferido pela Coordenação do Curso, respeitando os prazos estabelecidos.

Art. 114 As atividades complementares somente são analisadas se realizadas nos períodos enquanto o discente estiver regularmente matriculado na UNIPAMPA, inclusive no período de férias.

Art. 115 Os casos omissos são apreciados e deliberados pela Comissão de Curso.