

Protótipo de uma Plataforma Web de Desenvolvimento Colaborativo de Software

Trabalho de Conclusão de Curso
apresentado à UTFPR como requisito
parcial para obtenção do título de Tecnólogo
em Sistemas para Internet.

**Curitiba
2011**

Guilherme Esplugues Sanches Calegari

**Protótipo de uma Plataforma Web de
Desenvolvimento Colaborativo de Software**

Trabalho de Conclusão de Curso
apresentado à UTFPR como requisito
parcial para obtenção do título de Tecnólogo
em Sistemas para Internet.

Orientador:

Professor Dr. Paulo César Stadzisz

**Curitiba
2011**

SUMÁRIO

1	INTRODUÇÃO.....	8
1.1	Contexto do Trabalho.....	8
1.2	Tecnologia da Informação no Mundo Contemporâneo.....	9
1.3	Trabalho Colaborativo.....	11
1.4	Desenvolvimento Distribuído de Software.....	13
1.5	Objetivo do Trabalho Apresentado.....	16
2	PLANEJAMENTO DO DESENVOLVIMENTO DO PROJETO.....	18
2.1	Caracterização e Objetivo do Projeto.....	18
2.2	Plano de Projeto.....	18
2.2.1	Escopo.....	19
2.2.2	Etapas de Desenvolvimento do Protótipo.....	20
2.2.3	Cronograma.....	21
2.2.3.1	Primeira Iteração: Rede Social.....	21
2.2.3.2	Segunda Iteração: Mecanismo de Projetos de Software...23	
2.2.3.3	Terceira Iteração: Ferramentas de Gerenciamento de Atividades e Rastreamento de Falhas.....	24
2.4	Riscos do Projeto.....	24
2.4.1	Riscos de Tecnologia.....	25
2.4.2	Riscos de Requisitos.....	25
2.4.3	Riscos de Estimativas.....	26
2.5	Tecnologias Adotadas.....	26
2.5.1	Interface (<i>front-end</i>).....	26
2.5.2	Comunicação Cliente-Servidor.....	27
2.5.3	Servidor (<i>back-end</i>).....	27
2.5.4	<i>Framework</i> de Persistência.....	27
2.5.5	Banco de Dados.....	28
2.6	Estudo de Viabilidade.....	28

2.6.1	Aspectos Organizacionais	28
2.6.2	Aspectos Econômicos.....	29
2.6.3	Custos de Desenvolvimento do <i>Developer Linker</i>	29
2.6.4	Conclusão do Estudo de Viabilidade	31
3	DETALHAMENTO DO PROJETO DE SOFTWARE	32
3.1	Requisitos do Sistema	32
3.1.1	Requisitos Funcionais.....	32
3.1.1.1	RF01 – Cadastrar Usuário	32
3.1.1.2	RF02 – Efetuar Entrada.....	32
3.1.1.3	RF03 – Personalizar Perfil.....	33
3.1.1.4	RF04 – Rede de Contatos.....	33
3.1.1.5	RF05 – Pesquisar Usuário(s)	33
3.1.1.6	RF06 – Adicionar Contato	34
3.1.1.7	RF07 – Mecanismo de Projetos de Software.....	34
3.1.1.8	RF08 – Criar Projeto.....	34
3.1.1.9	RF09 – Visualizar Detalhes de Projeto	35
3.1.1.10	RF10 – Adicionar Projeto.....	35
3.1.1.11	RF11 – Pesquisar Projeto(s).....	35
3.1.1.12	RF12 – Ferramenta de Gestão de Atividades	35
3.1.1.13	RF13 – Visualizar Atividade(s)	36
3.1.1.14	RF14 – Criar Atividade	36
3.1.1.15	RF15 – Visualizar Detalhes das Atividades.....	36
3.1.1.16	RF16 – Delegar Atividade	37
3.1.1.17	RF17 – Ferramenta de Rastreamento de Falhas	37
3.1.1.18	RF18 – Visualizar Falha(s).....	37
3.1.1.19	RF19 – Visualizar Detalhes da Falha	37
3.1.1.20	RF20 – Criar Falha.....	38
3.1.1.21	RF21 – Atribuir Falha	38
3.1.2	Requisitos Não Funcionais	38
3.1.2.1	RNF01 – Eficiência.....	38
3.1.2.2	RNF02 – Confiabilidade	39

3.1.2.3	RNF03 – Portabilidade	39
3.1.2.4	RNF04 – Usabilidade.....	39
3.1.2.5	RNF05 – Implementação.....	39
3.2	Esquema da Arquitetura.....	40
3.3	Modelagem do Software.....	40
3.3.1	Diagramas de Casos de Uso.....	40
3.3.1.1	Rede Social – Geral	41
3.3.1.2	Rede Social – Projetos	42
3.3.1.3	Ferramentas – Gestão de Atividades	43
3.3.2	Diagramas de Classes.....	44
3.3.2.1	Rede Social – Geral (<i>front-end</i>)	44
3.3.2.2	Rede Social – Geral (<i>back-end</i>)	47
3.3.2.3	Rede Social – Projetos (<i>front-end</i>)	50
3.3.2.4	Rede Social – Projetos (<i>back-end</i>)	51
3.3.2.5	Ferramentas – Gestão de Atividades (<i>front-end</i>)	53
3.3.2.6	Ferramentas – Gestão de Atividades (<i>back-end</i>)	54
3.3.3	Diagramas de Sequência	54
3.3.3.1	Adicionar Contato	55
3.3.3.2	Cadastrar Nova Conta	56
3.3.3.3	Efetuar Entrada	57
3.3.3.4	Personalizar Perfil.....	58
3.3.3.5	Visualizar Detalhes do Contato	59
3.3.3.6	Pesquisar Usuários	60
3.3.3.7	Visualizar Projetos	61
3.3.3.8	Criar Projeto	62
3.3.3.9	Adicionar Projeto	63
3.3.3.10	Pesquisar Projetos.....	64
3.3.3.11	Visualizar Detalhes Projeto	65
3.3.3.12	Criar Atividade.....	66
3.3.3.13	Visualizar Atividades	67
3.3.3.14	Visualizar Detalhes Atividades.....	68

3.3.3.15	Delegar / Editar Atividade	69
3.4	Interfaces com o Usuário (Telas)	70
3.4.1	Tela de Cadastro	70
3.4.2	Tela de <i>Login</i>	70
3.4.3	Tela Inicial do Perfil	71
3.4.4	Tela de Alteração de Dados do Perfil.....	71
3.4.5	Tela de Visualização dos Detalhes dos Contatos	72
3.4.6	Tela de Pesquisa de Usuários	72
3.4.7	Tela Inicial do Mecanismo de Projetos de Software.....	73
3.4.8	Tela de Pesquisa de Projetos	73
3.4.9	Tela de Criação de Projetos	74
3.4.10	Tela de Visualização de Detalhes do Projeto	74
3.4.11	Tela Inicial de Projeto	75
3.4.12	Tela de Gestão de Atividades do Projeto.....	76
3.4.13	Tela de Criação e Edição de Atividade do Projeto	76
3.4.14	Tela de Visualização dos Detalhes da Atividade.....	77
3.5	Sequência de Navegação das Telas	77
4	CONCLUSÃO	78
4.1	Trabalhos Futuros	80
5	REFERÊNCIAS	81

RESUMO

Este trabalho de conclusão de curso apresenta os resultados do estudo sobre a utilização de sistemas colaborativos para o desenvolvimento de software por equipes geograficamente distantes. Tal prática se tornou uma importante estratégia de negócios, utilizada por governos e corporações globais, que visam minimizar seus custos de desenvolvimento de software, por meio da terceirização de mão de obra. Além da apresentação sobre os estudos realizados, este trabalho também contempla o desenvolvimento de um protótipo de uma ferramenta para o desenvolvimento de software colaborativo. Esta ferramenta chama-se *Developer Linker* e integra funções de rede social com funções de ferramentas de gestão de equipes.

Palavras chaves: Desenvolvimento, Software, Web, Colaborativo, Distribuído.

1 INTRODUÇÃO

1.1 Contexto do Trabalho

Dos tempos mais remotos aos dias atuais, pode-se considerar que a capacidade de transmissão de informação tem sido uma das principais características que permitiram a evolução da humanidade e de suas sociedades.

Em um primeiro instante, a informação era transmitida somente de forma oral. Desta maneira, o que era passado de geração para geração eram informações baseadas em relatos e interpretações subjetivas, resultando em possíveis distorções da realidade.

Em um segundo instante, a descoberta e o aperfeiçoamento da escrita por meio da utilização de símbolos e alfabetos, proporcionou que o conhecimento de fatos presenciados ou relatos feitos por pessoas que viveram em outras épocas fossem passados para as próximas gerações. Mais tarde, há pouco mais de 500 anos, Gutenberg inventou a prensa, tecnologia que possibilitou a disseminação da informação para um número cada vez maior de pessoas.

Em um terceiro instante, surgiu a era eletrônica, guiada por tecnologias informacionais, baseadas em sistemas computacionais e redes de telecomunicações. A característica marcante desta era foi a velocidade pela qual a informação era transmitida (em tempo real ao acontecimento), fazendo com que a sociedade priorizasse a informação e a tecnologia, a qual acabou influenciando estilos de vida e padrões de comportamento (lazer, trabalho, consumo) (Lima et al., 2001).

Nos dias atuais, muitas vezes a sociedade é classificada como a “Sociedade da Informação”, porém, pode-se dizer que tal sociedade existe desde os tempos em que a informação era transmitida somente de forma oral. O que diferencia estas sociedades são a forma e a velocidade pela qual a informação é transmitida. Assim, Lima (2001) define que a sociedade atual pode ser chamada de “Sociedade da Informação Eletrônica”.

Partindo do contexto exposto, observa-se na atualidade a consolidação das sociedades da informação a nível global, ou seja, a velocidade da geração da informação e sua transmissão, assim como o conceito de distância, foram alteradas de tal forma que as sociedades antes “isoladas” em termos sociais, políticos, culturais e financeiros, agora, mesmo que virtualmente, formam uma só sociedade. Tal fenômeno pode ser interpretado como o efeito da globalização que a sociedade pós-industrial (década de 80) sofreu durante os últimos 20-30 anos. O avanço das tecnologias resultou em uma maior dispersão da informação e, conseqüentemente, do conhecimento humano, fato que têm guiado a maneira de viver destas sociedades globais (Werthein, 2000).

1.2 Tecnologia da Informação no Mundo Contemporâneo

Nos últimos anos tem-se observado o amadurecimento e expansão do software de maneira geral, o qual se tornou um componente vital para governos, empresas, indústrias, universidades, etc.. Criou-se uma dependência tal que o sucesso das instituições muitas vezes depende diretamente da eficácia dos sistemas utilizados por elas.

O advento e utilização de sistemas computacionais para fins não militares datam de aproximadamente cinquenta anos, quando segundo Audy (2007), algumas poucas empresas começaram a adquirir os primeiros computadores de grande porte para uso comercial.

Da década de sessenta do século passado aos dias de hoje, o amadurecimento dos sistemas computacionais passou por três importantes fases. A primeira fase é conhecida como Processamento de Dados, baseada no uso de *Mainframes*, que segundo Audy (2007), vigorou durante as décadas de 50 e 60. Nesta fase, os sistemas tinham como foco apoiar internamente as organizações (principalmente a área financeira e administrativa), sendo que tal apoio objetivava o ganho de produtividade e eficiência.

Durante as décadas de 70 e 80 vigorou a fase conhecida por Informática. Esta fase continuava sendo baseada no uso de *Mainframes*,

mas surgia naquela época um novo paradigma, a introdução dos Microcomputadores no ambiente corporativo. O alvo desta fase continuava sendo apoiar os departamentos internos das instituições, porém o objetivo organizacional havia mudado para a eficácia dos indivíduos e grupos de trabalho. Foi nesta fase que ocorreu a integração dos sistemas existentes, caracterizando a topologia de sistemas de informação que serviria como alicerce para a terceira fase (Audy, 2007).

Finalmente, a terceira fase, também conhecida como a fase de Tecnologias de Informação, veio para alterar o papel do software nas instituições de mero coadjuvante para protagonista. A área de atuação do software nesta época era centrada no comercial e produção, sendo assim, o novo objetivo era a criação de valor e agregação de valor (desenvolvimento de produtos) por meio dos sistemas de informação. Foi durante esta fase que o uso de redes de computadores, sistemas integrados, orientação a objetos, ferramentas colaborativas, etc., se consolidaram. Tal fase teve início na década de 90 até meados do início do século XXI.

Nos dias atuais, alguns poucos anos após o “término” da terceira fase, o cenário mundial relativo ao uso de sistemas de informação ganhou dimensões nunca antes pensadas. O fenômeno das redes sociais, o colaborativismo e a dispersão da informação fazem parte da realidade de uma parcela bastante significativa da população mundial. Por meio da Internet e dos sistemas de informação, empresas locais tornaram-se empresas globais, governos tornaram seus processos burocráticos em rápidos e úteis sistemas, a maneira de se comprar e vender produtos por meio dos sistemas computacionais aliados à Internet virou uma confortável realidade (*e-commerce*), etc. Pode-se assumir que, na atualidade, a vida humana depende direta ou indiretamente dos sistemas de informação, pois, eles se tornaram base para praticamente todos os ramos de atuação humana.

1.3 Trabalho Colaborativo

Os seres humanos, por natureza, são seres sociais. Só evoluíram graças ao comportamento social, o qual prevê a colaboração no desenvolvimento das mais diferentes atividades. No contexto atual, o corporativismo exige a colaboração entre os indivíduos de uma forma muito peculiar, a transmissão da informação para a execução do trabalho (Jiménez et al., 2008). Assim, no final da década de oitenta, surgiu um novo campo de estudo focado na colaboração/cooperação dos indivíduos que utilizam o computador como meio de trabalho, o Trabalho Cooperativo Apoiado por Computador – CSCW (*Computer Supported Cooperative Work*).

Por definição, o CSCW consiste de uma atividade coordenada assistida por computador, obtida a partir da colaboração de um grupo de pessoas. Este campo de *estudo concentra seu foco em duas áreas, a distância entre os indivíduos (co-localização e dispersão geográfica) e os métodos de comunicação utilizados (síncrono e assíncrono)*. Sendo ele um objeto de estudo de diversas áreas do conhecimento como a ciência da computação, *design*, engenharia, administração, matemática, psicologia, e sociologia.

Ambientes computacionais voltados para o trabalho colaborativo propiciam grandes vantagens no auxílio ao trabalho individual, em grupo e corporativo. No entanto, os sistemas informatizados precisam prover benefícios tangíveis para cada um desses domínios de colaboração e permitir que o indivíduo possa executar seu trabalho de maneira eficiente e criativa (Araújo & Campanário, 2004).

O CSCW é o campo de estudo que pesquisa e expande os conhecimentos humanos sobre o trabalho colaborativo assistido por computador, porém, é um campo teórico. Por sua vez, a utilização dos conceitos do CSCW de modo prático é conhecida como *Groupware*. A definição de Araújo & Campanário para o termo Groupware é “uma família de aplicações, baseadas em computador, que dá suporte a grupos de pessoas, engajadas em uma tarefa comum e que prevê uma interface para

compartilhar o ambiente, especialmente em termos de comunicação, colaboração e suporte à decisão”. Desta forma, pode-se resumir a definição de *Groupware* como sendo qualquer sistema baseado em computador que permita que grupos de pessoas trabalhem de forma cooperativa objetivando alcançar uma mesma finalidade.

No cenário atual, tanto no individualismo quanto no corporativismo, sistemas colaborativos são utilizados no dia a dia, e, de tão corriqueiros, a maioria dos seus usuários nem se dá conta de que estão utilizando sistemas baseados em *Groupware*. Alguns *Groupwares* de uso frequente são os sistemas de comunicação (síncrono/assíncrono), tais como e-mail, *chats* (ICQ, MSN Messenger, Skype, etc.), sistemas de informação compartilhada como fóruns, wikis e listas de discussão, sistemas de *e-learning* como o Moodle, editores de grupo como o Google Docs, sistemas de redes sociais como Facebook, Orkut e MySpace, sistemas empresariais tais como o Microsoft SharePoint, IBM Lotus Notes, entre tantos outros.

O trabalho colaborativo agrega pessoas em um modo de trabalho diferente e dinâmico. A implementação de um ambiente de trabalho colaborativo deverá levar em conta que o grau de interação de seus usuários não se restrinja a diálogos simplórios por meio das telas fornecidas pelo sistema. O objetivo das aplicações baseadas em *groupware* não é somente o ganho de produtividade, mas também a melhoria da comunicação e a colaboração entre os funcionários da organização, o que pode-se chamar de melhoria nos relacionamentos profissionais.

O *Awareness* ou, percepção, é assim um fator determinante para o sucesso de uma equipe que trabalha baseada em *groupware*. Tal percepção recai sob os indivíduos pertencentes à equipe em relação ao conhecimento do que seus colegas estão fazendo, onde suas atividades se encaixam no trabalho como um todo, qual é a situação desse trabalho, etc.. Por vezes, a literatura cita esta característica como sendo um dos pontos falhos ou insuficientes da utilização de *Groupware*.

Para que o colaborativismo flua de maneira eficaz, o modelo organizacional tradicional deverá ser substituído, não quanto à hierarquia,

mas sim quanto à forma pela qual a informação é transmitida dentro da organização visto que quanto mais ágil e menos burocrática a organização for, maiores serão as chances de adaptação às rápidas mudanças que o mercado global exige.

1.4 Desenvolvimento Distribuído de Software

A massificação das empresas rumo à globalização transformou a visão do software no mundo corporativo, tornando-o uma peça importante e, em alguns casos, vital para o processo de expansão de negócios, possibilitando empresas antes locais a se tornarem empresas globais.

O avanço da economia, a sofisticação dos meios de comunicação e a pressão para a diminuição dos custos forçaram as empresas a buscar alternativas para suprir a falta de mão de obra qualificada, orçamentos e prazos cada vez menores. Uma das maneiras que foi encontrada para esta finalidade, foi optar pela terceirização de mão de obra, prática muito comum em outras áreas como, por exemplo, a manufatura.

A terceirização de mão de obra no contexto de desenvolvimento de software é classificada baseada na distância entre os envolvidos (*stakeholders*, usuários, equipe de desenvolvimento, etc.) da seguinte maneira (Audy, 2007).

- **Mesma Localização Física:** ocorre quando todos os envolvidos estão co-localizados, possibilitando encontros face a face, sem a diferença de fuso horário e diferenças culturais significativas;
- **Distância Nacional:** reuniões em curto intervalo de tempo, ocorrência de fuso horário e possibilidade de diferenças culturais maiores;
- **Distância Continental:** envolvidos em países diferentes, mas dentro do mesmo continente. As reuniões face a face são comprometidas, e há maior interferência de fuso horário
- **Distância Global:** atores em países e continentes diferentes. Todas as outras características.

Os diferentes níveis de dispersão geográfica são chamados de *outsourcing*, *insourcing*, *inshore*, *onshore*, etc. e, mais comumente usado, o termo *Offshore* ou, ainda, *Offshoring* (*Offshore Sourcing*).

Algumas classificações, segundo Audy (2007), sobre as estratégias de terceirização de mão de obra são:

- *Outsourcing*: terceirizar processos para uma entidade externa (mesmo país).
- *Nearshoring*: estratégia *Offshore*, ocorre quando a distância entre os atores envolvidos não ultrapassa os limites continentais, porém estes estando em países diferentes.
- *Offshore Sourcing* (*Offshoring*): terceirizar processos, não necessariamente para uma entidade externa, porém necessariamente em um país diferente de onde está situada a matriz da empresa / cliente.

Por fim, a partir da necessidade da diminuição dos custos, tempo de desenvolvimento, qualidade etc., surgiu, há alguns anos, uma nova área da Engenharia de Software, voltada a possibilitar o desenvolvimento de software de maneira distribuída. Essa área é conhecida por DDS (Desenvolvimento Distribuído de Software) ou do inglês DSD (*Distributed Software Development*).

Para Audy (2007), algumas razões que levam as empresas a buscar soluções de desenvolvimento distribuído de software são:

- **Demanda e Custos**: a crescente demanda por serviços de software contribuiu para a valorização dos profissionais da área. Assim, países desenvolvidos passaram a competir por profissionais e por custos, porém, profissionais destes países custam muito mais do que de países em desenvolvimento, ocasionando a procura por profissionais de menor custo. Além do fator de mão de obra mais barata e qualificada, os países origem desta mão de obra, geralmente oferecem benefícios fiscais para empresas estrangeiras se estabelecerem.

- Rapidez de resposta ao mercado: no cenário atual, colocar um produto, o quanto antes, no mercado pode ser um fator decisivo na realidade das empresas. Assim, o desenvolvimento distribuído possui uma característica interessante para a redução do tempo de desenvolvimento, a característica *follow-the-sun* (seguir o sol), onde há a dispersão geográfica e temporal das equipes ao redor do globo, possibilitando o desenvolvimento do produto 24 horas por dia.
- Mercado e presença global: atualmente a demanda por serviços de software não compreende somente a América do Norte e Europa. Por meio do aumento do poder computacional e do barateamento da informatização, praticamente todo o globo necessita de serviços de desenvolvimento de software.
- Sinergia Cultural: contando com equipes distribuídas em cidades, estados, países e, até mesmo, continentes diferentes, a convivência de culturas diferentes pode ser benéfica para o projeto, pois promove a ampliação da criatividade e a maneira de abordar e resolver problemas.

Ao se optar por uma estratégia *Offshore*, deve-se estar ciente de que, apesar dos benefícios como, por exemplo, economia de gastos em média de 25% a 50%, o desenvolvimento de software continua sendo uma tarefa complexa. Ao adicionar distância física, temporal e cultural, a complexidade indubitavelmente aumentará, requerendo especial atenção para fatores como: confiança entre as equipes, consciência sobre o trabalho alheio (*awareness*), gerenciamento de conflitos, diferenças culturais, espírito de equipe e liderança (Lanubile, 2009).

Para que a execução de um projeto de desenvolvimento de software em modelo distribuído possa ocorrer, é necessário também que a arquitetura do software seja modular, pois esta é a única maneira de resolver tarefas complexas de forma distribuída. Um projeto modular reduz a complexidade e permite o desenvolvimento em paralelo. O gerenciamento do projeto requer especial atenção visto que o modelo de gerência co-localizado perde efeito quando em projetos com equipes dispersas.

1.5 Objetivo do Trabalho Apresentado

Este trabalho de conclusão de curso foi motivado por observações e experiências com desenvolvimento de software no âmbito distribuído (*offshoring*) e, também, pela carência de ferramentas e sistemas específicos para o desenvolvimento distribuído de software no mercado. Tais observações motivaram a pesquisa sobre o assunto e, desta forma, o sistema a ser desenvolvido envolve uma gama de conceitos, sendo os mais importantes o trabalho colaborativo (CSCW) e o desenvolvimento distribuído de software (DDS).

O nome proposto neste trabalho, Plataforma Web de Desenvolvimento Colaborativo de Software, vem da idéia da construção de um sistema distribuído (baseado na Internet) no qual desenvolvedores de software, independentemente de localização, pudessem se organizar em times a fim de colaborar uns com os outros buscando um objetivo em comum.

O termo “Colaborativo” no nome define como o sistema em si é classificado. Por se tratar de um conjunto de ferramentas voltadas à colaboração, pode-se, então, classificar formalmente que o sistema futuro é um *Groupware*.

O Desenvolvimento Distribuído de Software (DDS), por sua vez, vem a ser o objetivo do *Groupware* neste contexto, ou seja, aquilo que os usuários do *Groupware* almejam ao utilizá-lo.

O objetivo geral do trabalho de conclusão de curso pode, então, ser definido como desenvolver um Protótipo de uma Plataforma Web de Colaboração para auxiliar o desenvolvimento distribuído de software. O trabalho propõe um protótipo uma vez que o desenvolvimento de uma ferramenta completa para este propósito ultrapassaria as possibilidades de tempo e custo de um trabalho de conclusão de curso.

Existem, atualmente, algumas poucas soluções voltadas à colaboração corporativa disponíveis no mercado e também na Internet (sistemas distribuídos). Estas ferramentas são, na sua maioria, genéricas, servindo para qualquer tipo de negócio. A Plataforma Web de Desenvolvimento Colaborativo de Software proposta será desenhada para atender

exclusivamente o desenvolvimento distribuído de software e também será uma ferramenta baseada em rede social, ou seja, incluirá algumas das vantagens de uma rede de usuários. Estas vantagens podem ser entendidas como o alto grau de interação entre os participantes, formação de redes de contatos (*networking*), alto grau de comunicação, entre outros.

Além da rede social, a Plataforma oferecerá a seus usuários um conjunto de ferramentas pensadas para a solução de problemas relacionados ao gerenciamento de equipes de software, em especial as fisicamente distantes. Estas ferramentas deverão permitir que usuários da rede social formem equipes, se comuniquem, designem papéis e atividades, monitorem atividades, rastreiem falhas, entre outros.

2 PLANEJAMENTO DO DESENVOLVIMENTO DO PROJETO

Este capítulo apresenta os resultados relativos ao planejamento do desenvolvimento do projeto, incluindo o plano de projeto, os riscos envolvidos no desenvolvimento, as tecnologias adotadas e estudo de viabilidade do projeto.

2.1 Caracterização e Objetivo do Projeto

Como indicado no capítulo de introdução, este trabalho de conclusão de curso tem por objetivo o estudo e desenvolvimento de um protótipo de uma plataforma Web, que será, a partir deste ponto, denominado *Developer Linker*. Este nome é uma abreviação para “Plataforma Web de Desenvolvimento Colaborativo de Software”, pois, se faz compreensível por meio de apenas duas palavras, explicitando o objetivo do protótipo.

O objetivo do *Developer Linker* é auxiliar desenvolvedores e/ou equipes de desenvolvimento de software geograficamente distantes na realização de suas atividades. Este auxílio compreende a formação de equipes, comunicação, divisão das atividades e acompanhamento. Portanto, o *Developer Linker* é uma ferramenta de colaboração para o desenvolvimento distribuído de software.

2.2 Plano de Projeto

Este projeto é resultado de uma atual demanda da área da engenharia de software, o DDS (Desenvolvimento Distribuído de Software), que tem como principal objetivo a redução dos custos de desenvolvimento de software, ganho de qualidade e de produtividade. Os resultados esperados do projeto são a expansão do tema e a construção de um protótipo de uma plataforma web de desenvolvimento colaborativo de software. Tal plataforma proverá meios para que o gerenciamento de projetos de software distribuídos seja realizado de maneira controlada e otimizada.

2.2.1 Escopo

O escopo do projeto foi definido partindo do princípio de que o resultado final não deverá caracterizar um sistema completo, mas sim, como diz o próprio título, um protótipo funcional de uma ferramenta de colaboração. Portanto, o presente projeto, incluindo todos os seus produtos, pode ser considerado uma prova de conceito para um projeto futuro, sendo este projeto futuro uma ferramenta totalmente funcional.

O protótipo proposto implementará um conjunto de funcionalidades consideradas essenciais para o funcionamento básico da ferramenta como um todo, sendo estas funcionalidades descritas por:

- Rede Social: uma estrutura básica de rede social, na qual o sistema será baseado. Esta rede social básica será responsável pelo *networking* entre os participantes.
- Mecanismo de Projetos de Software: um mecanismo capaz de definir projetos de software e reunir participantes em prol de um objetivo comum.
- Ferramenta de Gerenciamento de Atividades: uma ferramenta de gestão de atividades, capaz de gerenciar as atividades geradas pelos projetos de software.
- Ferramenta de Rastreamento de Falhas: uma ferramenta capaz de auxiliar o registro e acompanhamento de erros e inconsistências ocorridos nos projetos de software.

No capítulo 3, as funcionalidades descritas acima serão expandidas e especificadas em maior detalhe.

2.2.2 Etapas de Desenvolvimento do Protótipo

As etapas do desenvolvimento do protótipo *Developer Linker* são baseadas em uma visão evolutiva da engenharia de software, que envolve a produção de protótipos funcionais de um sistema futuro, nos quais são realizadas verificações a fim de validar as funcionalidades desenvolvidas e descobrir outras mais. Este processo evolutivo funciona por meio de iterações, ou seja, o período total definido para o desenvolvimento do projeto é dividido em porções menores, sendo que cada porção tem a responsabilidade de desenvolver um protótipo funcional, a partir de um conjunto de requisitos funcionais e não funcionais previamente estabelecidos. Assim, o desenvolvimento se torna linear, a cada iteração novas funcionalidades são adicionadas ao protótipo e, ao final das iterações, o resultado será o sistema previsto como um todo. Esta visão ou ciclo de vida é chamado de prototipação.

O *Developer Linker* foi planejado para ser desenvolvido em três grandes iterações, sendo elas:

- Desenvolvimento da Rede Social: compreendendo o desenvolvimento do protótipo que conterà as características relacionadas à Rede Social.
- Desenvolvimento do Mecanismo de Projetos de Software: compreendendo o desenvolvimento do protótipo que conterà as características relacionadas à Rede Social e Projetos de Software.
- Desenvolvimento das Ferramentas de Gerenciamento de Atividades e Rastreamento de Falhas: compreendendo o desenvolvimento do protótipo que conterà as características relacionadas à Rede Social, Projetos de Software e Ferramentas.

2.2.3 Cronograma

O cronograma do projeto tem o objetivo de organizar as atividades e o tempo despendido para a sua realização. Por se tratar de um projeto com base na evolução contínua, o gerenciamento das atividades será dividido por iteração.

O planejamento da execução do projeto foi elaborado com base na disponibilidade de 4 horas diárias de desenvolvimento, sete dias por semana, sendo este desenvolvimento dividido entre as categorias: requisitos, modelagem e codificação.

As dependências entre as atividades não são necessariamente reais, visto que muitas delas poderiam ser realizadas em paralelo, porém, devido à dinâmica e prazo de desenvolvimento, muitas atividades foram sequenciadas visando um melhor andamento do projeto.

O projeto em questão foi desenvolvido em um prazo de 112 (cento e doze) dias, tendo seu início no dia 26 de Abril de 2011 e seu término no dia 15 de Agosto de 2011.

2.2.3.1 Primeira Iteração: Rede Social

A primeira iteração corresponde às atividades voltadas a Rede Social e teve duração de sessenta e dois dias, de 26 de Abril de 2011 a 15 de Julho de 2011, sendo a maior das três iterações.

Foram previstas para esta iteração a identificação e análise dos requisitos, modelagem *UML* e codificação relativos à Rede Social. Tais requisitos são descritos em detalhes no capítulo de requisitos.

A Identificação e Análise dos Requisitos relativos à Rede Social foram previstas para ocorrer do dia 26 de Abril a 13 de Maio, compreendendo os requisitos funcionais e não funcionais do *Developer Linker*. A figura 1 mostra a iteração de identificação e análise dos requisitos.

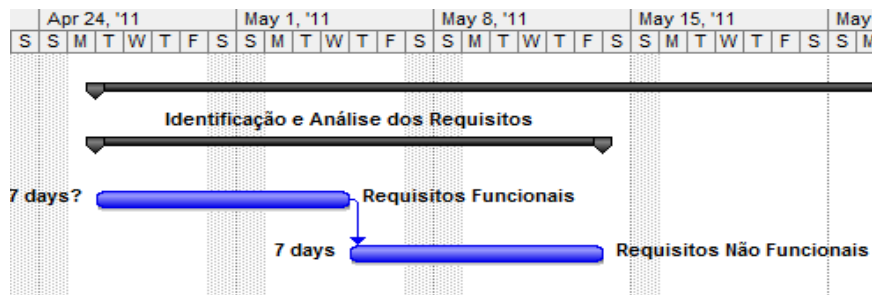


Figura 1. Primeira Iteração – Identificação e Análise dos Requisitos

A modelagem *UML* foi prevista para ocorrer entre os dias 16 de Maio a 26 de Junho, compreendendo os diagramas de casos de uso, sequência e classes. A figura 2 mostra a iteração da modelagem *UML*.

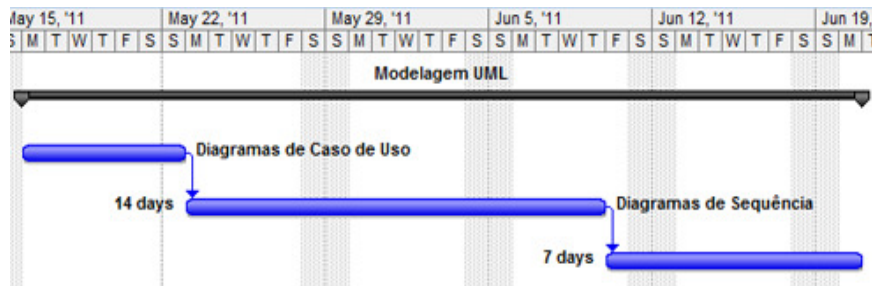


Figura 2. Primeira Iteração – Modelagem *UML*

A codificação (implementação) relativa à Rede Social ocorrerá durante os dias 21 de Junho e 15 de Julho, compreendendo a codificação das telas e das regras de negócio (*server side*). A figura 3 mostra a iteração de codificação.

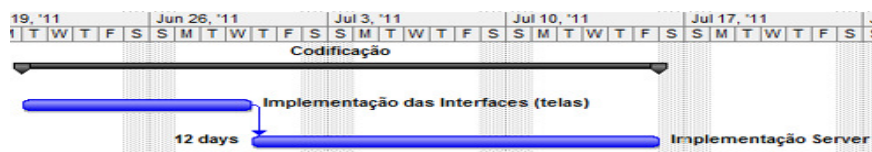


Figura 3. Primeira Iteração – Codificação

2.2.3.2 Segunda Iteração: Mecanismo de Projetos de Software

A segunda iteração terá duração de apenas doze dias, ocorrendo do dia 18 a 31 de Julho. Esta iteração ocorrerá em um menor período de tempo por se tratar de uma funcionalidade da Rede Social. As atividades realizadas na primeira iteração foram repetidas, porém, relacionadas ao Mecanismo de Projetos de Software. A figura 4 mostra a iteração do mecanismo de projetos de software.

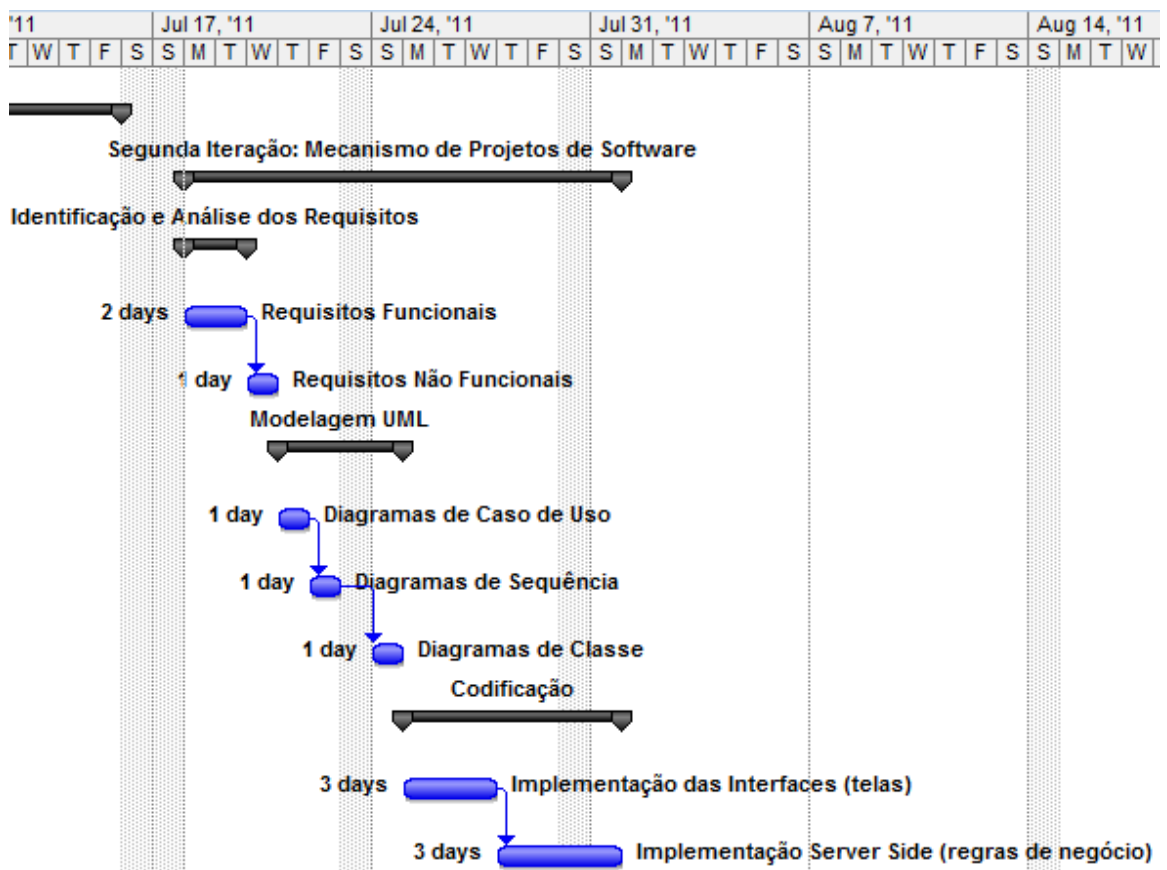


Figura 4. Segunda Iteração – Mecanismo de Projetos de Software

2.2.3.3 Terceira Iteração: Ferramentas de Gerenciamento de Atividades e Rastreamento de Falhas

A terceira e última iteração ocorreu em um curto período de tempo, exatos quinze dias, tendo início no dia 1º de Agosto e término no dia 15 do mesmo mês.

As atividades desenvolvidas neste período foram o desenvolvimento das ferramentas de Gerenciamento de Atividades e Rastreamento de Falhas. A figura 5 mostra a iteração das ferramentas de gerenciamento de atividades e rastreamento de falhas.

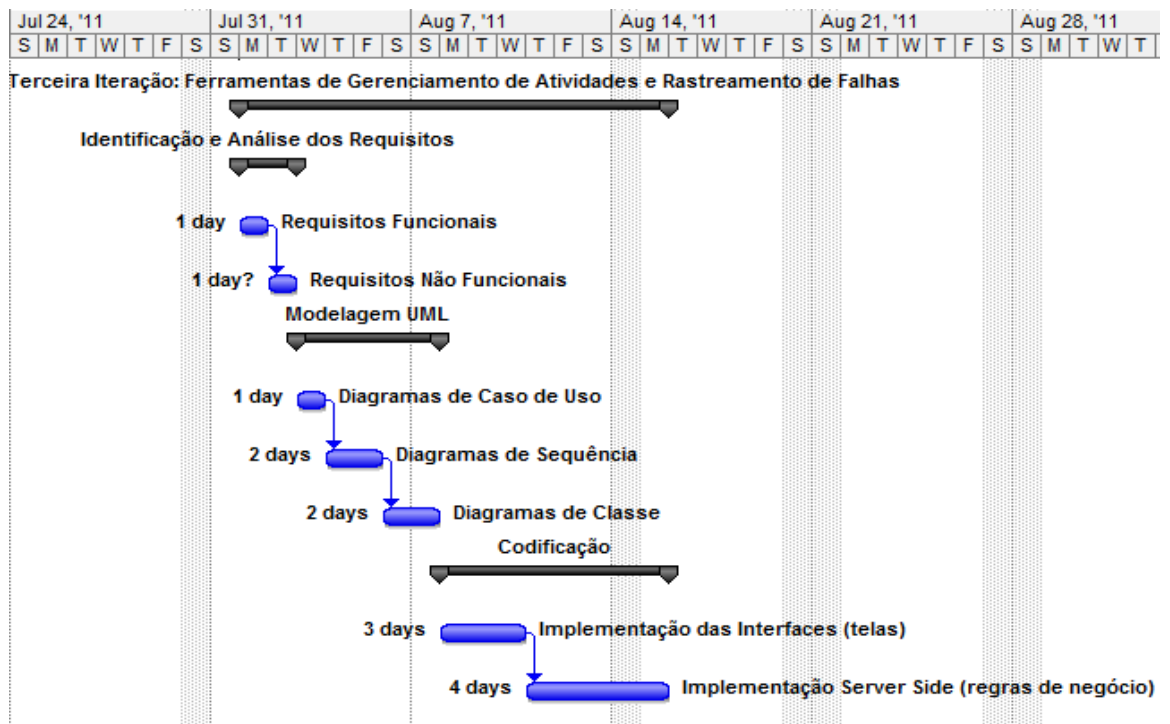


Figura 5. Terceira Iteração - Ferramentas de Gerenciamento de Atividades e Rastreamento de Falhas

2.4 Riscos do Projeto

Esta seção prevê a identificação dos riscos inerentes ao desenvolvimento do *Developer Linker*, que representam real probabilidade

de ocorrência cujos efeitos possam ser danosos para a continuidade do projeto.

2.4.1 Riscos de Tecnologia

Os riscos de tecnologia são aqueles que derivam de tecnologias de software ou hardware utilizadas para desenvolver o sistema [Sommerville, 2007]. Os principais riscos de tecnologia considerados neste projeto são:

- **Container de Aplicação:** o sistema utilizará como *container* de aplicação o produto da *Apache Software Foundation*, o *Tomcat*, o qual não é recomendado para sistemas de médio/grande porte, por ser limitado no atendimento de múltiplas requisições (escalabilidade), entre outros problemas.
- **Sistema Gerenciador de Banco de Dados:** o sistema utilizará o SGDB (sistema gerenciador de banco de dados) da Oracle, o *Mysql*, o qual não comporta altas taxas de tráfego de dados e transações.
- **Tecnologia de Interface:** o sistema utilizará o *framework* da Adobe, o *Flex*, o qual utiliza o *Adobe Flash Player* para a exibição das telas. Dois fatores que classificam o *Flex* como um risco são o desenvolvimento do *HTML 5*, o qual prevê conteúdos ricos sem a necessidade de *pluguins* adicionais e as vulnerabilidades a ataques que o *Flash Player* expõe seus usuários .

2.4.2 Riscos de Requisitos

Os riscos de requisitos são aqueles que derivam de mudanças de requisitos do cliente e do processo de gerenciamento de mudança de requisitos (Sommerville, 2007).

- **Mudanças de Requisitos:** requisitos que foram analisados e modelados em determinada fase, podem sofrer mudanças durante o desenvolvimento do projeto. Estas mudanças requererão um maior esforço, que será refletido por todas as etapas envolvidas no processo de desenvolvimento do projeto (análise, modelagem e

codificação), acarretando em um possível atraso da entrega do projeto.

- Identificação de Novos Requisitos: é comum que a identificação de novos requisitos ocorra durante o projeto, porém estes, se não forem gerenciados, podem acarretar no atraso da entrega.

2.4.3 Riscos de Estimativas

Os riscos de estimativas são aqueles que derivam de estimativas de gerenciamento das características de sistema e estimativas de recursos necessários para construir o sistema [Sommerville, 2007].

- Prazo de Desenvolvimento: o prazo estimado para o desenvolvimento total do *Developer Linker* é de cento e doze dias, podendo vir a ser estendido devido à ocorrência de contratempos.
- Tamanho do Projeto: existe a possibilidade do tamanho e o esforço necessário para o desenvolvimento do corrente projeto ter sido subestimado, acarretando em um maior esforço de desenvolvimento, o que geraria atrasos.

2.5 Tecnologias Adotadas

Esta seção descreve as tecnologias utilizadas para a construção do *Developer Linker*.

2.5.1 Interface (*front-end*)

A camada de apresentação do *Developer Linker* foi construída a partir do *framework* da Adobe Systems, o Adobe Flex, versão 4. O Adobe Flex é um conjunto de tecnologias voltadas a conteúdos ricos, ou seja, conteúdos que possibilitem uma maior experiência do usuário com o sistema, daí a classificação de *RIA* (*Rich Internet Application*). Baseado em duas “linguagens” distintas, utiliza para a representação visual dos elementos e telas uma linguagem de marcação baseada em XML (*Extensible Markup Language*), o MXML. Para o controle dos elementos da interface escritos em

MXML, é utilizado o *ActionScript*, o qual é baseado em *JavaScript*. Tanto o MXML quanto o *ActionScript* são de propriedade da Adobe Systems e de uso não licenciado.

2.5.2 Comunicação Cliente-Servidor

Para a realização da comunicação entre o cliente (*front-end*) e o servidor (*back-end*), foi utilizado o Adobe BlazeDS, versão 4. O BlazeDS é um sistema de *messengeria* baseado em Java, o qual trabalha “acoplado” ao servidor, sendo o responsável pela troca de mensagens entre *front-end* e *back-end*, sendo o *front-end* a interface com o usuário do sistema e o *back-end* o servidor. Apesar de existirem outras maneiras de promover a comunicação em sistemas baseados em *Flex*, o *BlazeDS*, é o *data source* (fonte de dados) recomendado pela Adobe para trabalhar com o Adobe Flex.

2.5.3 Servidor (*back-end*)

O servidor da aplicação será baseado na tecnologia Java, sexta versão. A escolha pela tecnologia Java, deve-se principalmente por ser uma tecnologia aberta, de alta escalabilidade e que possui uma comunidade abundante de desenvolvedores (oferta de mão de obra). O aqui chamado de servidor será o centro de tomada de decisões ou controlador da aplicação. O servidor recebe as requisições do lado cliente, as processa e responde de acordo com as regras de negócio.

2.5.4 Framework de Persistência

Para a execução das operações relacionadas a banco de dados (leitura, escrita, etc.), será utilizado o *framework* de persistência da JBoss, o *Hibernate*. Este framework faz a “ponte” entre a aplicação e o banco de dados. A escolha pelo *Hibernate se deve*, pelo fato de que através dele, se torna desnecessária a utilização explícita de SQL (*Structured Query Language*).

2.5.5 Banco de Dados

O sistema de banco de dados a ser adotado é o Oracle *Mysql*. Sistema de banco de dados relacional rápido e de fácil manutenção. Apesar de não contar com um bom suporte a transações e escalabilidade, o *Mysql* foi considerado a opção mais adequada, visto que, é um sistema gerenciador de banco de dados de uso aberto e que as demandas do protótipo por escalabilidade e transações não são relevantes.

2.6 Estudo de Viabilidade

O estudo de viabilidade tem o objetivo de verificar se o sistema proposto atinge os propósitos da organização para a qual está sendo desenvolvido [SOMMERVILLE, 2007].

O estudo de viabilidade do *Developer Linker* é classificado em três aspectos comuns às instituições que trabalham com o desenvolvimento de software: organizacionais, econômicos e técnicos, conforme apresentado nas próximas subseções.

2.6.1 Aspectos Organizacionais

Os aspectos organizacionais podem ser descritos como a eficácia com que o sistema proposto apóia os objetivos estratégicos da organização utilizadora. Desta maneira, o *Developer Linker* tem como objetivo organizacional, promover apoio gerencial às equipes de desenvolvimento de software.

O sistema proposto não visa a substituição das ferramentas convencionais de gerenciamento, mas sim, complementar tais ferramentas para que o desenvolvimento distribuído de software tenha suas peculiaridades gerenciadas sob uma ótica detalhista, além das demais dificuldades encontradas no desenvolvimento de software.

2.6.2 Aspectos Econômicos

Os aspectos econômicos que o sistema desenvolvido oferece às instituições e/ou grupos de desenvolvedores de software estão centrados em duas principais áreas:

- **Economia de Custos:** o desenvolvimento de software de maneira distribuída, isto é, geograficamente distante, prima por benefícios financeiros, pois, países fornecedores de *Offshore Outsourcing* (Índia, China, Brasil, México, etc.) geralmente oferecem incentivos fiscais, custos inferiores de desenvolvimento, entre outros. Desta maneira, o *Developer Linker* é uma ferramenta que apóia o gerenciamento de equipes distribuídas a fim de otimizar a economia de custos das instituições / grupos de desenvolvedores que o utilizam.
- **Redução de Investimentos:** quando em modelo de desenvolvimento de software distribuído geograficamente, os investimentos necessários para a viabilização de projetos de software são minimizados, pois, investimentos em estrutura (maiores escritórios, mobília, etc.), hardware, software, despesas em geral, tendem a sofrer diminuição visto que tais investimentos ficam sob responsabilidade da instituição/grupo de desenvolvedores contratados.

2.6.3 Custos de Desenvolvimento do *Developer Linker*

Os custos de desenvolvimento aproximados da plataforma *Developer Linker* seguem listados a seguir.

- **Hardware:** dois computadores domésticos, sendo um deles um notebook (máquina de desenvolvimento) e o segundo um desktop (servidor). Valor aproximado: R\$ 2000.00,00
- **Software:** Microsoft Windows7, Microsoft Project e Microsoft Office. Valor Aproximado: R\$ 800.00,00

- Salários: a tabela a seguir apresenta uma estimativa dos custos de mão de obra técnica para o desenvolvimento do Developer Linker, tendo como base os valores médios praticados em Curitiba, Paraná.

Função	Val. Aprox.	Horas Aprox.	Total Individual
Analista de Requisitos	R\$ 35,00	96	R\$ 3.360,00
Analista de Sistemas	R\$ 30,00	120	R\$ 3.600,00
Desenvolvedor Adobe Flex	R\$ 25,00	140	R\$ 3.500,00
Desenvolvedor Java	R\$ 25,00	160	R\$ 4.000,00
Total Aproximado:			R\$ 14.760,00

Tabela 1. Estimativa dos custos salariais

- Estrutura Física: assumindo que a execução do projeto é de cento e doze (112) dias, a tabela a seguir demonstra os gastos aproximados com estrutura.

Item	Val. Aprox.	Total Individual
Aluguel Escritório	R\$ 750.00,00/mês	R\$ 3.000,00
Energia Elétrica	R\$ 120.00,00/mês	R\$ 480.00,00
Acesso Rápido a Internet	R\$ 80.00/mês	R\$ 320.00,00
Telefone	R\$ 50.00,00/mês	R\$ 200.00,00

Item	Val. Aprox.	Total Individual
Mobilha	R\$ 2000.00,00	R\$ 2000.00,00
Total Aproximado: R\$ 6.000,00		

Tabela 2. Estimativa dos custos estruturais

O custo total aproximado de desenvolvimento do *Developer Linker*, sem o acréscimo de impostos é de **R\$ 20.760,00**.

2.6.4 Conclusão do Estudo de Viabilidade

O *Developer Linker* é um sistema para apoiar o processo de gerenciamento de projetos geograficamente distribuídos. Além disso, sua utilização requer apenas conexão rápida com a Internet e um navegador de Internet, que possua o *Adobe Flash Player* instalado. Partindo destas premissas, para as instituições/grupos de desenvolvedores, o *Developer Linker* se mostra uma opção viável para atender a uma demanda específica, o gerenciamento de projetos de desenvolvimento de software realizado por equipes distribuídas. Tanto o prazo de desenvolvimento quanto os custos envolvidos estão dentro de um patamar aceitável para um produto desta natureza. Os custos referem-se ao desenvolvimento de um protótipo. Assim, para um produto final, custos adicionais deveriam ser incluídos.

3 DETALHAMENTO DO PROJETO DE SOFTWARE

Este capítulo apresenta os documentos que definem o *Developer Linker* em nível de implementação de software. Os documentos presentes neste capítulo são os requisitos do sistema, diagramas de casos de uso, diagramas de sequência, diagramas de classes, esquema da arquitetura, diagramas entidade relacionamento e a definição das telas do sistema.

3.1 Requisitos do Sistema

A sessão de requisitos tem como objetivo documentar os requisitos classificados como fundamentais para o desenvolvimento de uma versão funcional do protótipo.

3.1.1 Requisitos Funcionais

Os requisitos funcionais são aqueles que descrevem quais as funcionalidades que o sistema deverá ter.

3.1.1.1 RF01 – Cadastrar Usuário

O *Developer Linker* deverá fornecer uma funcionalidade voltada para o cadastramento de novos usuários.

- Pré-condição: Nenhum.
- Entradas: nome, e-mail, nome de usuário do sistema e senha.
- Saídas: Confirmação de criação de conta; Não confirmação de criação de conta.

3.1.1.2 RF02 – Efetuar Entrada

O *Developer Linker* deverá identificar seus usuários no ato de entrada (*login*).

- Pré-condição: Possuir cadastro.
- Entradas: Nome de usuário e senha.
- Saídas: Permissão de acesso; Erro de usuário / senha incorreto ou inexistente.

3.1.1.3 RF03 – Personalizar Perfil

O *Developer Linker* deverá oferecer a funcionalidade de personalização de perfil, aonde serão inseridos dados de caráter pessoal, profissional, escolar, etc..

- Pré-condição: Possuir cadastro; Efetuar Entrada.
- Entradas: Dados pessoais, dados profissionais, foto, etc..
- Saídas: Nenhum.

3.1.1.4 RF04 – Rede de Contatos

O *Developer Linker* deverá agrupar e exibir os contatos que façam parte da rede de contatos de determinado usuário. Tal exibição se valerá da apresentação da foto e nome do(s) contatos.

- Pré-condição: Possuir cadastro; Possuir contato(s).
- Entradas: Nenhum.
- Saídas: Ao interagir com outro usuário listado na rede de contatos, deverá ser mostrada a tela do perfil do usuário selecionado.

3.1.1.5 RF05 – Pesquisar Usuário(s)

O *Developer Linker* deverá oferecer aos seus usuários uma ferramenta de pesquisa que possibilite a pesquisa por demais usuários. Esta ferramenta deverá estar disponível a partir do perfil de usuário. Os resultados das pesquisas serão exibidos de acordo com os critérios fornecidos pelo usuário.

- Pré-condição: Possuir cadastro; Efetuar Entrada.
- Entradas: Critério(s) de busca.
- Saídas: Redirecionamento para a tela de visualização dos resultados da pesquisa.

3.1.1.6 RF06 – Adicionar Contato

Ao visualizar o perfil de determinado usuário, deverá ser possível adicionar o mesmo a rede de contatos.

- Pré-condição: Possuir cadastro; Efetuar entrada.
- Entradas: Nenhum.
- Saídas: Confirmação de envio de requisição; Não Confirmação de envio de requisição.

3.1.1.7 RF07 – Mecanismo de Projetos de Software

O *Developer Linker* deverá conter uma sessão denominada “Projetos”, a qual exibirá os projetos do usuário. A exibição deste mecanismo deverá ocorrer quando o usuário selecionar a opção no menu de navegação. Em caso de prévia participação de projetos, os mesmos deverão ser exibidos nesta sessão.

- Pré-condição: Possuir cadastro; Ter efetuado a entrada.
- Entradas: Nenhum.
- Saídas: Nenhum.

3.1.1.8 RF08 – Criar Projeto

O *Developer Linker* deverá oferecer aos seus usuários a opção de criar novos projetos de software. A criação se dará por meio de uma tela interna a sessão de Projetos, a qual deverá conter campos para a inserção dos dados pertinentes ao projeto (nome, categoria, descrição, etc.).

- Pré-condição: Possuir cadastro; Ter efetuado a entrada.
- Entradas: nome do projeto, tecnologia, descrição, etc..
- Saídas: Redirecionamento para tela de visualização do projeto criado; Erro na criação de projeto.

3.1.1.9 RF09 – Visualizar Detalhes de Projeto

O Mecanismo de Projetos de Software deverá fornecer uma visão dos detalhes dos projetos de software.

- Pré-condição: Possuir cadastro; Ter efetuado a entrada.
- Entradas: Nenhum.
- Saídas: Exibição dos detalhes do projeto selecionado.

3.1.1.10 RF10 – Adicionar Projeto

O Mecanismo de Projetos de Software deverá fornecer ao usuário a opção de adicionar (ingressar) projeto(s).

- Pré-condição: Possuir cadastro; Ter efetuado a entrada.
- Entradas: Nenhum.
- Saídas: Confirmação de adição de projeto.

3.1.1.11 RF11 – Pesquisar Projeto(s)

O *Developer Linker* deverá oferecer aos seus usuários uma ferramenta de projetos. Esta ferramenta deverá estar disponível a partir do Mecanismo de Projetos de Software. Os resultados das pesquisas serão exibidos de acordo com os critérios fornecidos pelo usuário.

- Pré-condição: Possuir cadastro; Efetuar Entrada.
- Entradas: Critério(s) de busca.
- Saídas: Redirecionamento para a tela de visualização dos resultados da pesquisa.

3.1.1.12 RF12 – Ferramenta de Gestão de Atividades

A ferramenta de gestão de atividades deverá apoiar a delegação e acompanhamento das atividades do projeto de software. Tal ferramenta estará acessível a partir da tela inicial de cada projeto.

3.1.1.13 RF13 – Visualizar Atividade(s)

A partir da tela inicial da ferramenta de Gestão de Atividades, deverão estar listadas as atividades pertencentes ao projeto que está sendo visualizado no momento.

- Pré-condição: Possuir cadastro; Ter efetuado a entrada; Possuir/Participar de projeto(s).
- Entradas: Nenhum.
- Saídas: Nenhum.

3.1.1.14 RF14 – Criar Atividade

A ferramenta de gestão de atividades deverá permitir que sejam criadas atividades relacionadas ao projeto de software que está sendo visualizado no momento.

- Pré-condição: Possuir cadastro; Ter efetuado a entrada; Participar de Projeto.
- Entradas: nome da atividade, data de início da atividade, data de encerramento da atividade, prioridade, recursos necessários, responsável pelo desenvolvimento da atividade, status, descrição e comentários.
- Saídas: Redirecionamento a tela inicial da sessão de gestão de atividades.

3.1.1.15 RF15 – Visualizar Detalhes das Atividades

A ferramenta de gestão de atividades deverá permitir a visualização dos detalhes das atividades pertencentes ao(s) projeto(s) de software.

- Pré-condição: Possuir cadastro; Ter efetuado a entrada; Participar de projeto.
- Entradas: Nenhum.
- Saídas: Exibição da tela de detalhes da atividade selecionada.

3.1.1.16 RF16 – Delegar Atividade

A ferramenta de gestão de atividades deverá fornecer aos usuários do *Developer Linker*, a opção de atribuir (delegar) atividades a outros membros pertencentes ao projeto sendo visualizado.

- Pré-condição: Possuir cadastro; Ter efetuado a entrada; Participar de projeto.
- Entradas: nome do membro a ser delegada a atividade.
- Saídas: Nenhum.

3.1.1.17 RF17 – Ferramenta de Rastreamento de Falhas

O *Developer Linker* conterá uma ferramenta de rastreamento de falhas, a qual estará disponível a partir da tela inicial de cada projeto. Esta ferramenta deverá auxiliar os usuários na documentação dos erros encontrados nos artefatos de software gerados pelos projetos.

3.1.1.18 RF18 – Visualizar Falha(s)

A tela principal da ferramenta de levantamento de falhas deverá apresentar a lista de falhas corrente. Esta lista será dividida em duas categorias: falhas ativas e falhas concluídas.

- Pré-condição: Possuir cadastro; Ter efetuado a entrada; Participar de projeto.
- Entradas: Nenhum.
- Saídas: Lista de falhas.

3.1.1.19 RF19 – Visualizar Detalhes da Falha

A ferramenta de rastreamento de falhas deverá permitir a visualização dos detalhes das falhas pertencentes ao(s) projeto(s) de software.

- Pré-condição: Possuir cadastro; Ter efetuado a entrada; Participar de projeto.
- Entradas: Nenhum.
- Saídas: Exibição da tela de detalhes da falha selecionada

3.1.1.20 RF20 – Criar Falha

A ferramenta de rastreamento de falhas deverá permitir que sejam criadas falhas relacionadas ao projeto de software que está sendo visualizado no momento.

- Pré-condição: Possuir cadastro; Ter efetuado a entrada; Participar de projeto.
- Entradas: nome da falha, prioridade, responsável, status, descrição e comentários.
- Saídas: Nenhum.

3.1.1.21 RF21 – Atribuir Falha

A ferramenta de rastreamento de falhas deverá permitir ao usuário atribuir a correção da falha a outro usuário. Caso a falha não possua responsável pela sua correção no momento de sua criação, posteriormente um usuário poderá ser nomeado a sua correção.

- Pré-condição: Possuir cadastro; Ter efetuado a entrada; Participar de projeto.
- Entradas: nome do responsável pela correção da falha.
- Saídas: Nenhum.

3.1.2 Requisitos Não Funcionais

Os requisitos não funcionais são aqueles que descrevem as qualidades do sistema.

3.1.2.1 RNF01 – Eficiência

O *Developer Linker* deverá apresentar respostas rápidas, pois, o longo tempo de carregamento poderá acarretar em descontentamento por parte dos utilizadores, fazendo com que os mesmos parem de utilizar o sistema. Requisições não deverão tardar mais que dez (10) segundos para serem atendidas.

3.1.2.2 RNF02 – Confiabilidade

O *Developer Linker* deverá apresentar um bom nível de confiabilidade, tendo aparatos que garantam a integridade de dados e informações fornecidas por seus usuários.

3.1.2.3 RNF03 – Portabilidade

O *Developer Linker* deverá ser independente de sistema operacional e navegador, tendo como única restrição de uso a instalação do *Adobe Flash Player*.

3.1.2.4 RNF04 – Usabilidade

O sistema devera conter telas objetivas e organizadas, fazendo com que a experiência do usuário seja valorizada. Principais itens:

- Textos para leitura em tamanho 12pt;
- Fundos brancos a fim de proporcionar contraste;
- Barra de navegação sempre disponível;
- Execução de funcionalidades em menos de oito cliques;

3.1.2.5 RNF05 – Implementação

O sistema deverá ser implementado nas seguintes tecnologias, caracterizando-se como um RIA (*Rich Internet Application*).

- *Front-end* (Interface): *Adobe Flex 4*
- Comunicação cliente- servidor: *Adobe BlazeDS*
- *Back-end* (servidor): *Oracle Java 6*
- Framework de Persistência: *JBoss Hibernate*
- Sistema Gerenciador de Banco de Dados: *Oracle Mysql*

3.2 Esquema da Arquitetura

O esquema ilustrado na figura 6 representa a arquitetura e o fluxo dos dados do *Developer Linker* do ponto de vista macro, ou seja, não atentando aos detalhes. O sistema será dividido em camadas, sendo elas: camada de apresentação (*front-end*), camada de comunicação (*BlazeDS*), camada de controle (*back-end*), camada de persistência (*Hibernate*). A figura 6 indica, também, as tecnologias que foram adotadas em cada uma das camadas.

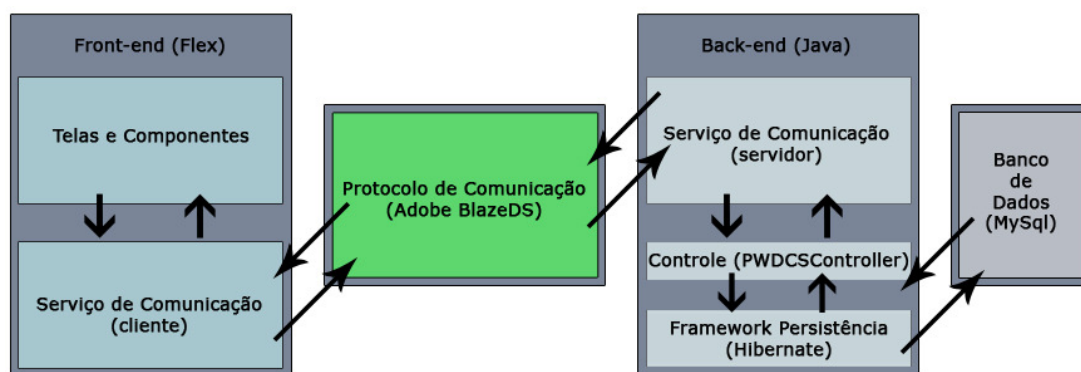


Figura 6. Esquema da Arquitetura

Modelagem do Software

Esta seção tem o objetivo de apresentar a modelagem do software, sendo esta, constituída de diagramas UML (*Unified Modeling Language*).

3.2.1 Diagramas de Casos de Uso

Os diagramas de casos de uso representam as interações entre os atores (agentes externos ao sistema – usuários, banco de dados, outros sistemas, etc.) e o sistema com o qual eles interagem (Pressman, 2005), *Developer Linker*.

3.2.1.1 Rede Social – Geral

A figura 7 apresenta as interações relacionadas à Rede Social - Geral.

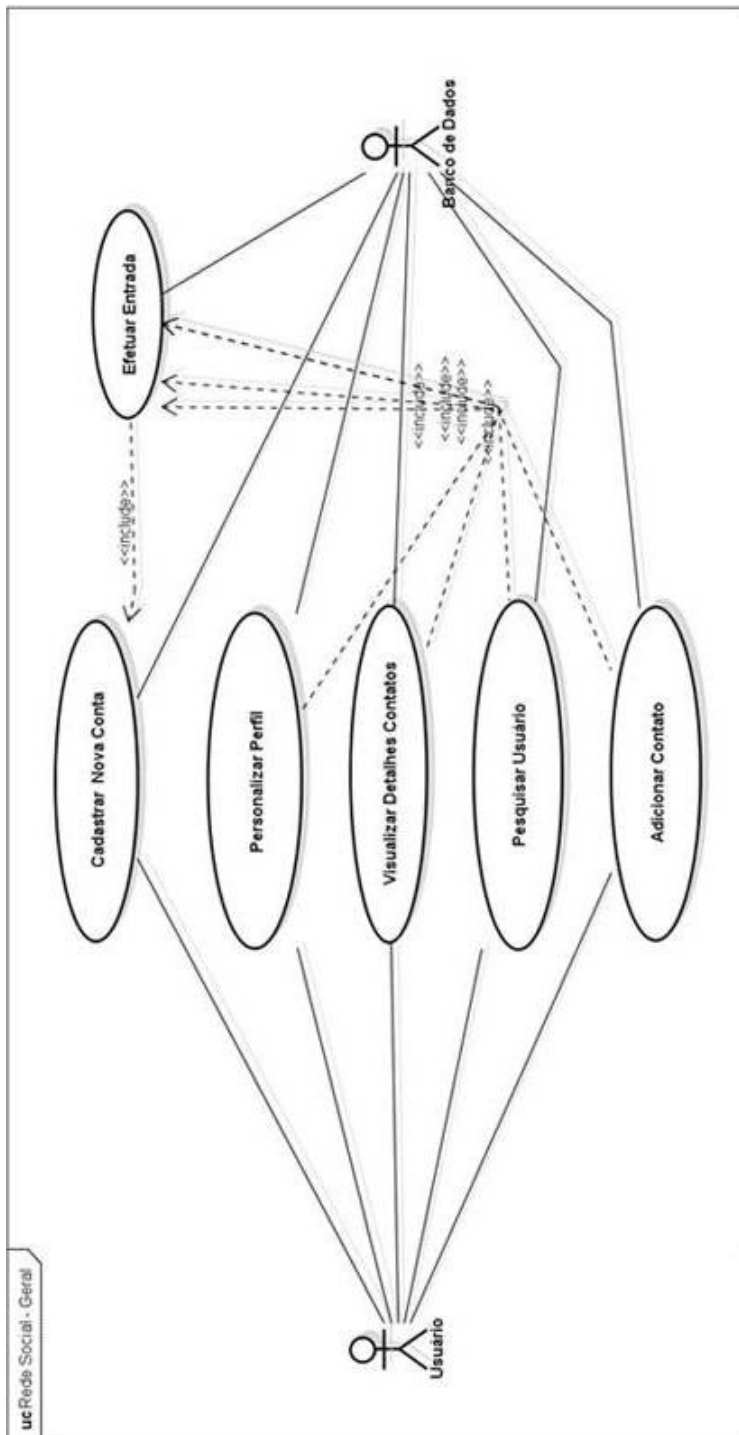


Figura 7. Diagrama de Casos de Uso: Rede Social - Geral

3.2.1.2 Rede Social – Projetos

A figura 8 apresenta as interações relacionadas com a Rede Social - Projetos.

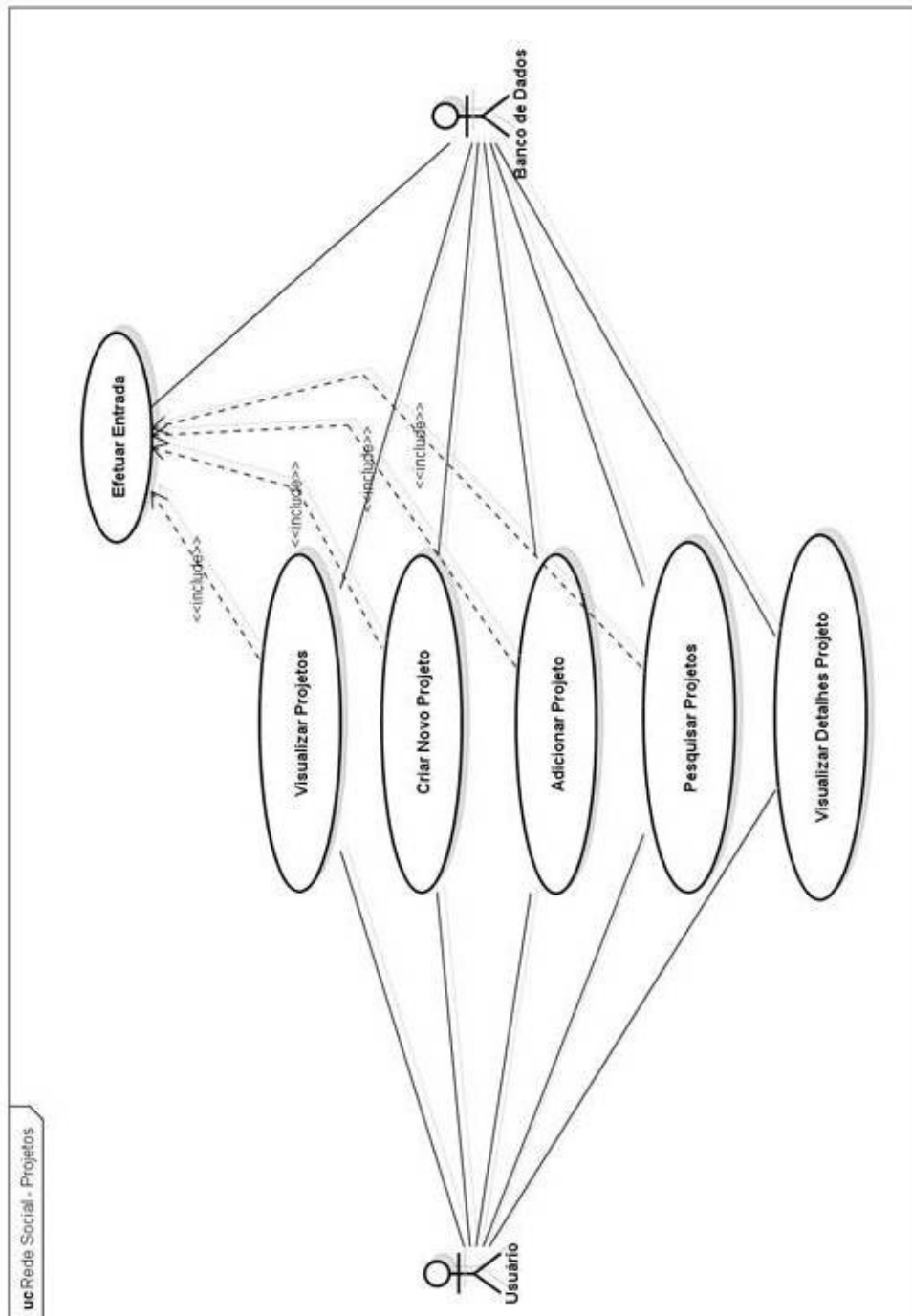


Figura 8. Diagrama de Casos de Uso: Rede Social - Projetos

3.2.1.3 Ferramentas – Gestão de Atividades

A figura 9 apresenta as interações relacionadas com a Ferramenta – Gestão de Atividades.

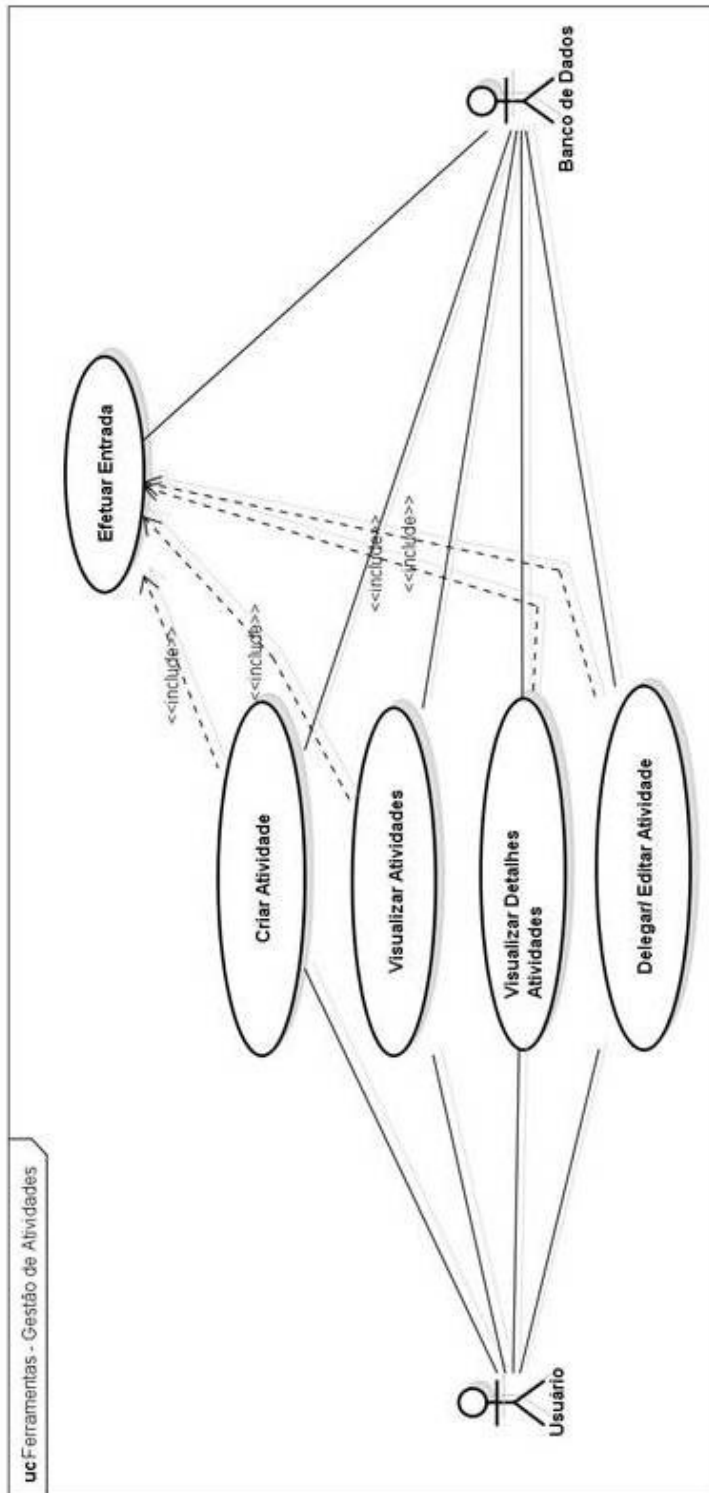


Figura 9. Diagrama de Casos de Uso: Ferramentas – Gestão de Atividades

3.2.2 Diagramas de Classes

Os diagramas de classes são representações das estruturas e suas relações (Pressman, 2005), sendo estas estruturas o modelo para os objetos que compõem o *Developer Linker*.

Dada a arquitetura utilizada pelo *Developer Linker*, os diagramas de classes foram divididos em diagramas que, representam a interface (*front-end*) em azul e diagramas que representam o servidor (*back-end*) em branco.

3.2.2.1 Rede Social – Geral (*front-end*)

A figura 10 apresenta as estruturas e seus relacionamentos com a Rede Social – Geral (*front-end*).

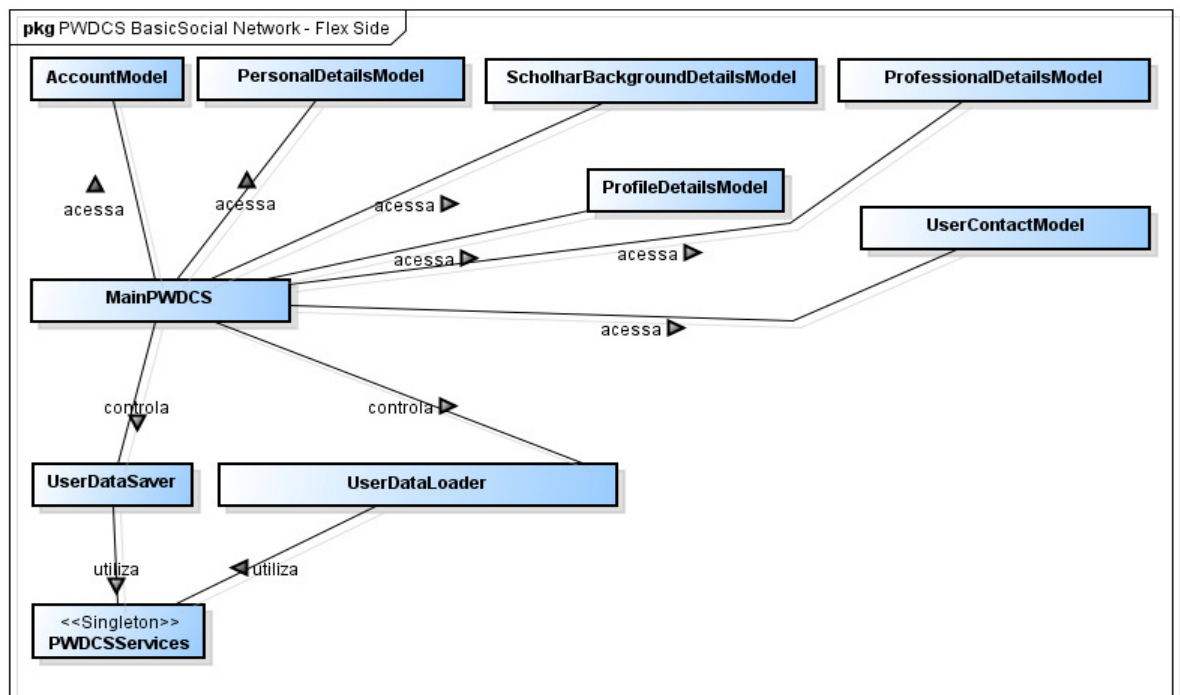


Figura 10. Diagrama de Classes: Rede Social – Geral (*front-end*)

A figura 11 apresenta os atributos e métodos pertencentes ao diagrama de classes Rede Social – Geral (*front-end*) parte 1.

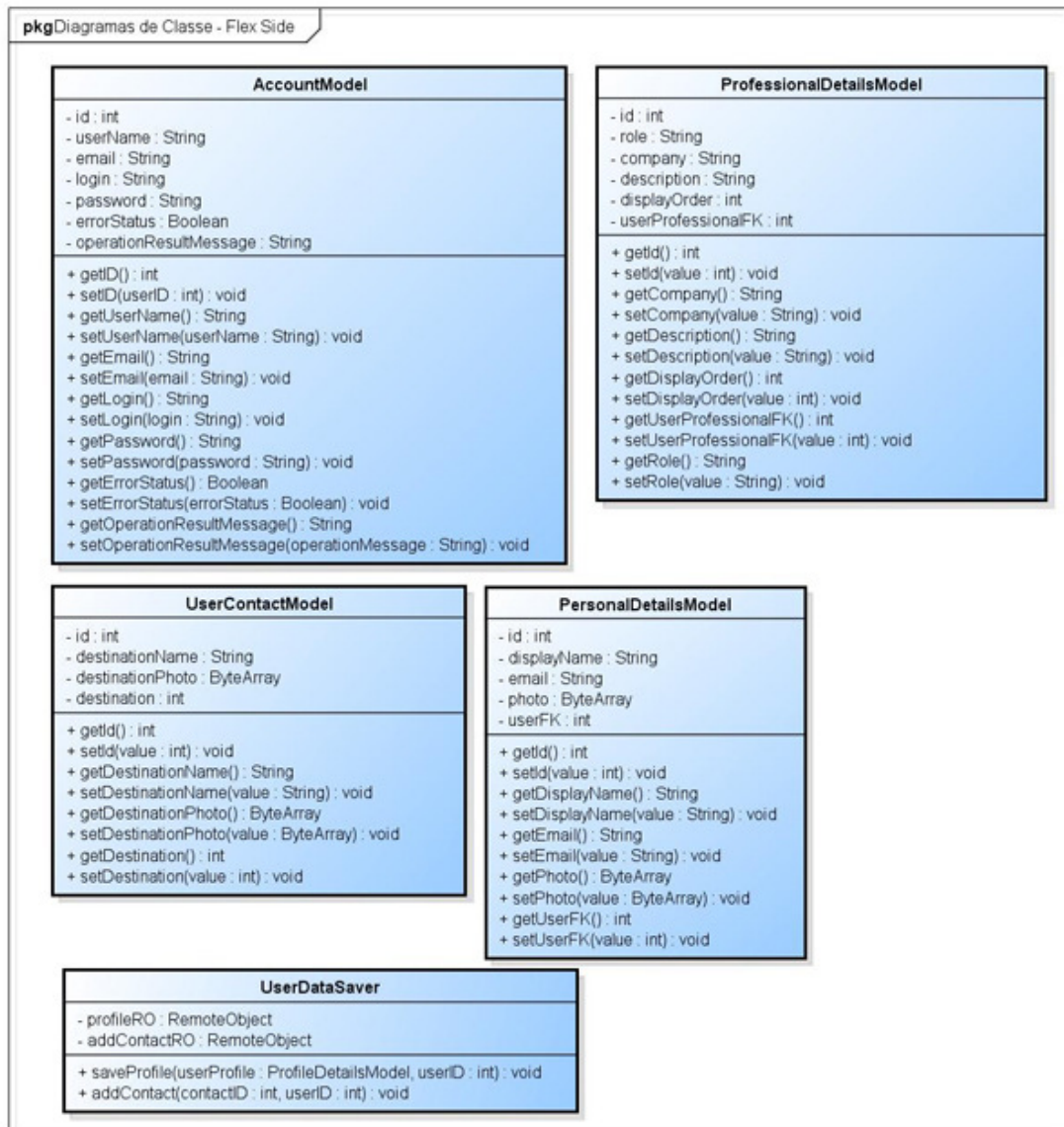


Figura 11. Atributos e Métodos: Rede Social – Geral parte 1

A figura 12 apresenta os atributos e métodos pertencentes ao diagrama de classes Rede Social – Geral (*front-end*) parte 2.

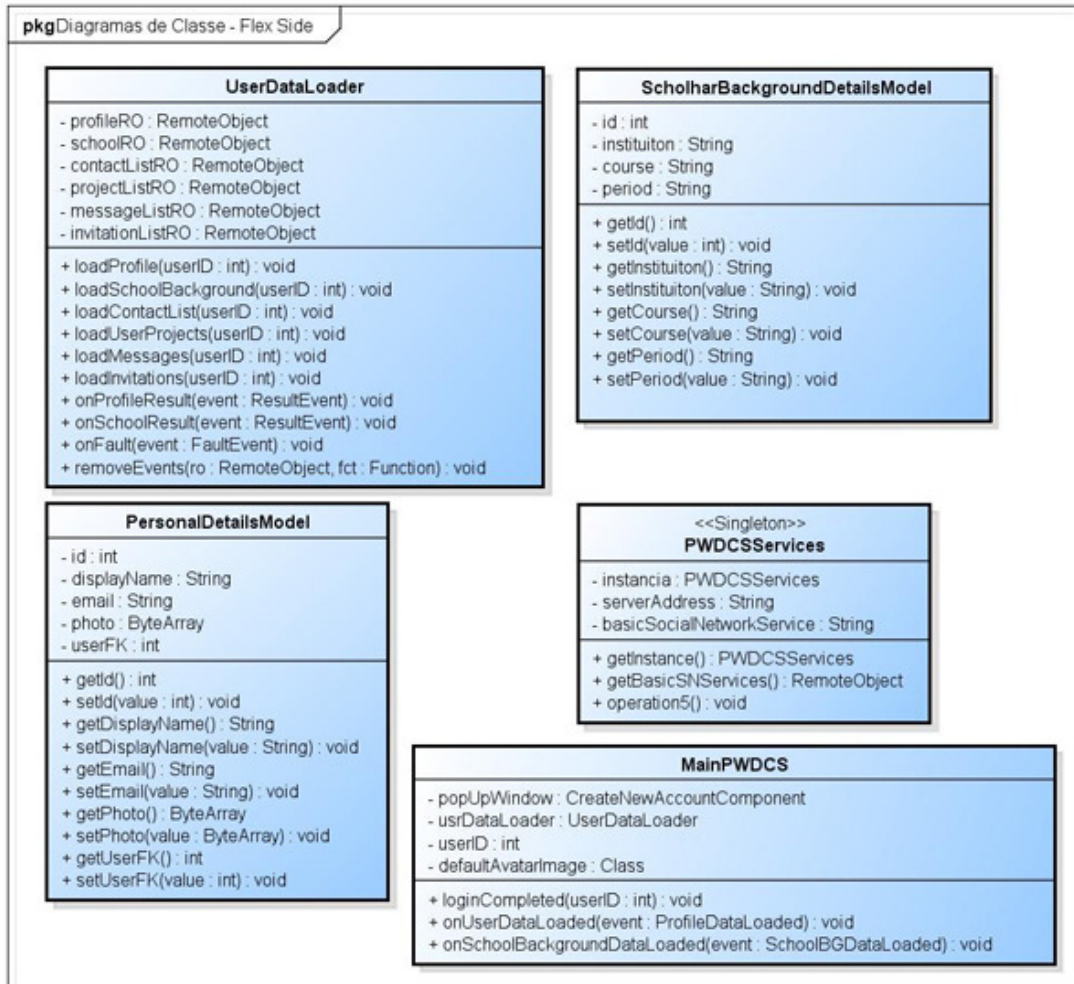


Figura 12. Atributos e Métodos: Rede Social – Geral parte 2

3.2.2.2 Rede Social – Geral (*back-end*)

A figura 13 apresenta as estruturas e seus relacionamentos com a Rede Social – Geral (*back-end*).

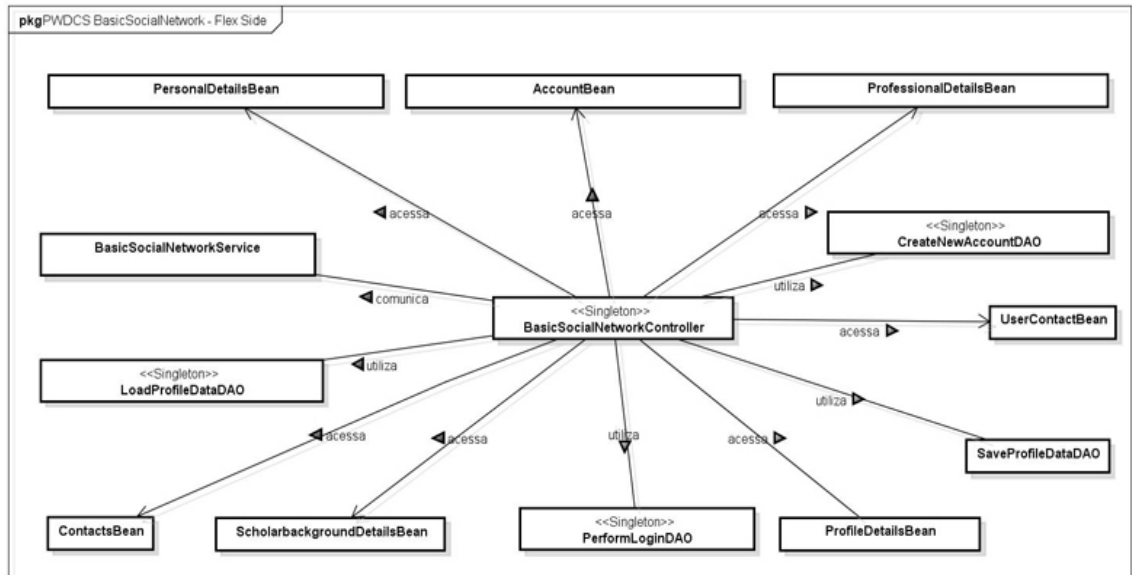


Figura 13. Diagrama de Classes: Rede Social – Geral (*back-end*)

A figura 14 apresenta os atributos e métodos pertencentes ao diagrama de classes Rede Social – Geral (*back-end*) parte 1.

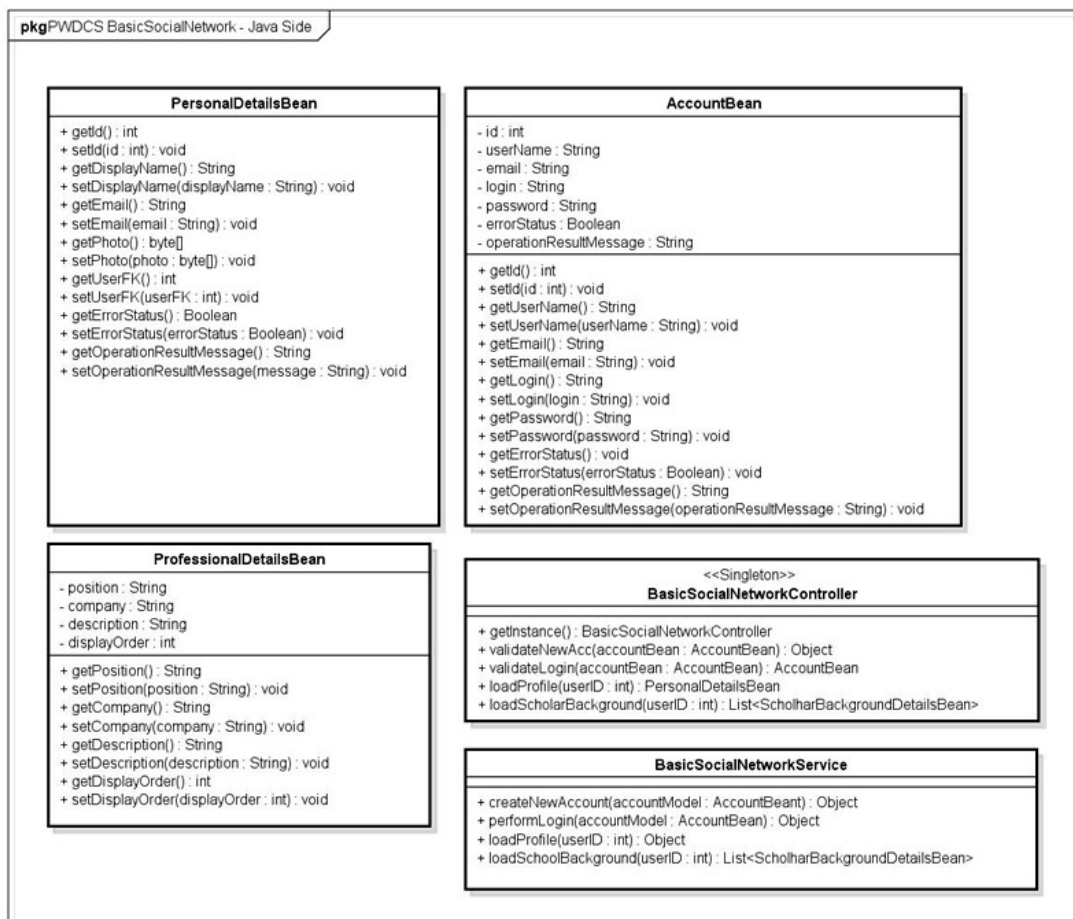


Figura 14. Atributos e Métodos: Rede Social – Geral parte 1

A figura 15 apresenta os atributos e métodos pertencentes ao diagrama de classes Rede Social – Geral (*back-end*) parte 2.

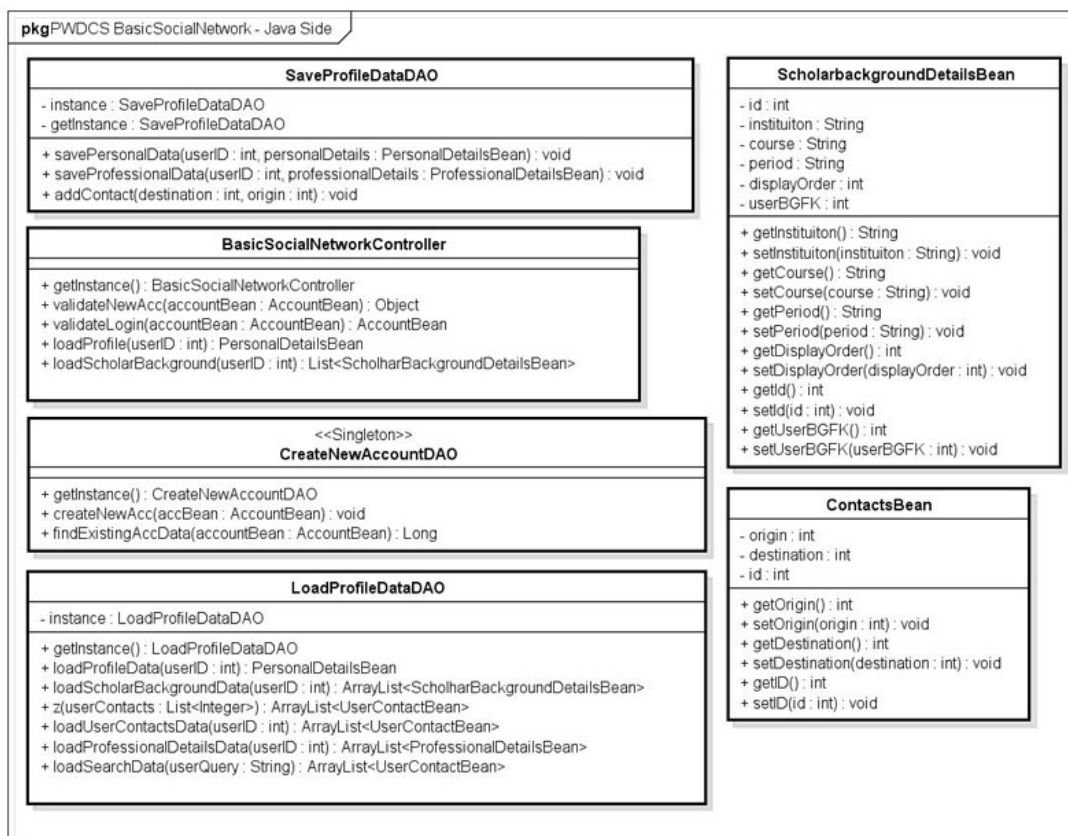


Figura 15. Atributos e Métodos: Rede Social – Geral parte 2

A figura 16 apresenta os atributos e métodos pertencentes ao diagrama de classes Rede Social – Geral (*back-end*) parte 3.

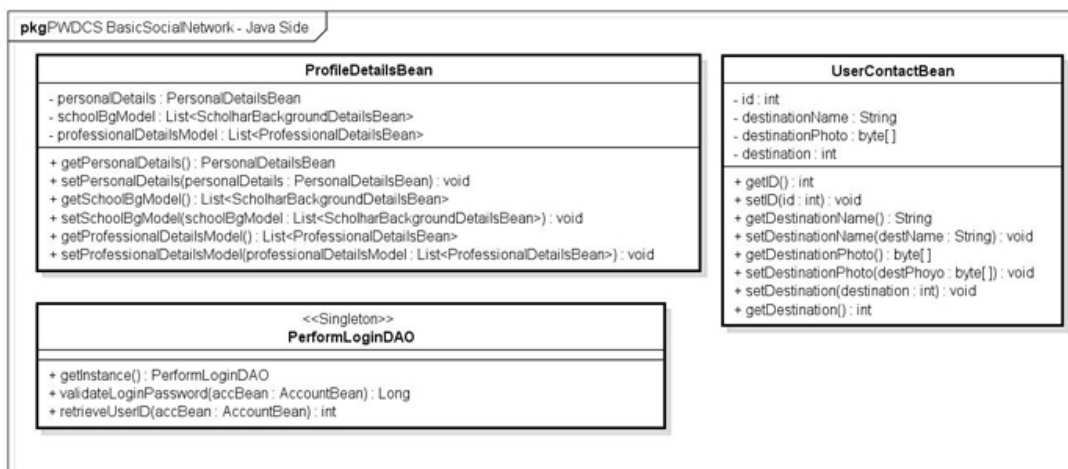


Figura 16. Atributos e Métodos: Rede Social – Geral parte 3

3.2.2.3 Rede Social – Projetos (*front-end*)

A figura 17 apresenta as estruturas e seus relacionamentos com a Rede Social – Projetos (*front-end*).

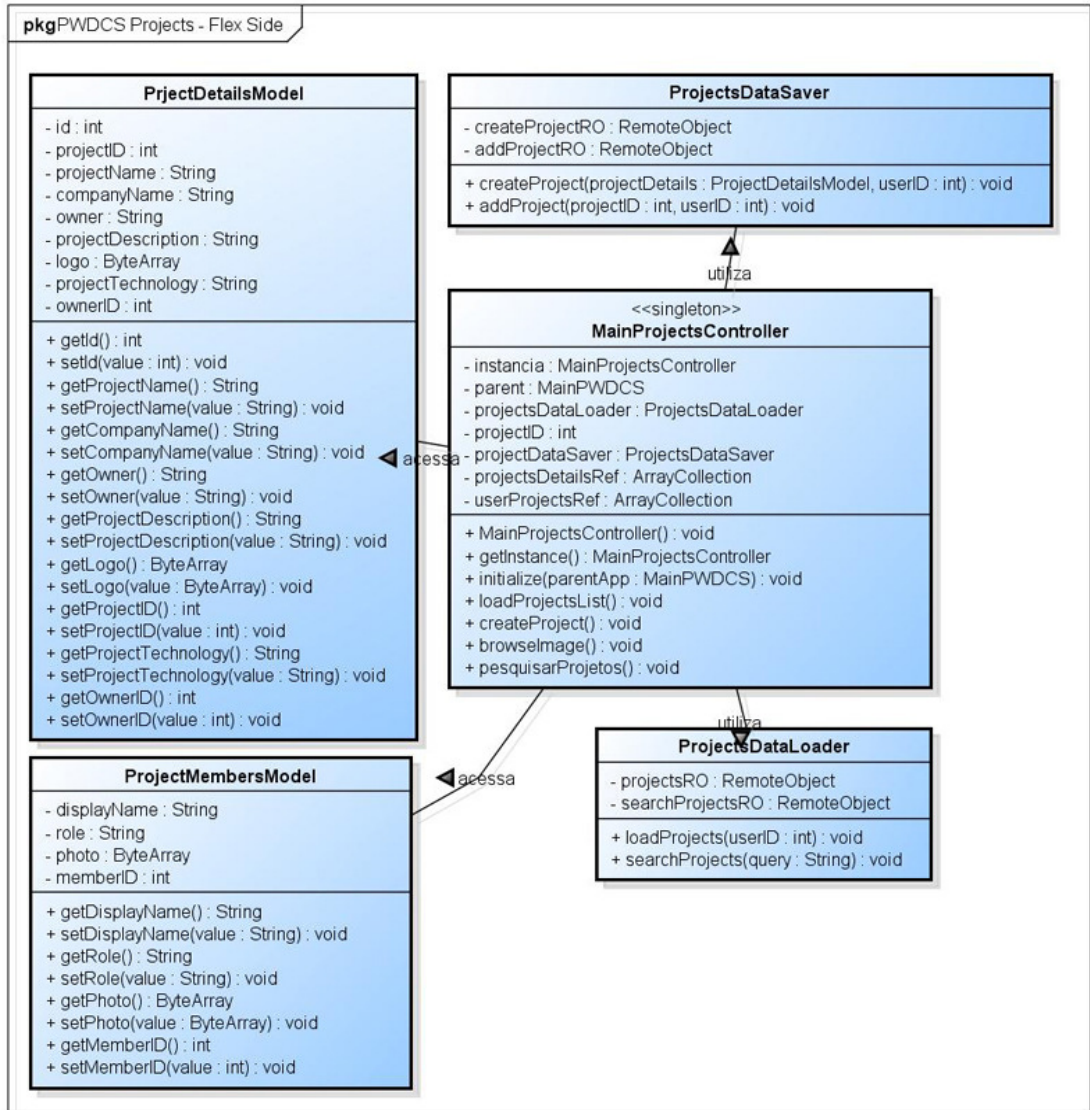


Figura 17. Diagrama de Classes: Rede Social – Geral (*front-end*)

3.2.2.4 Rede Social – Projetos (*back-end*)

A figura 18 apresenta as estruturas e seus relacionamentos com a Rede Social – Projetos (*back-end*) parte 1.

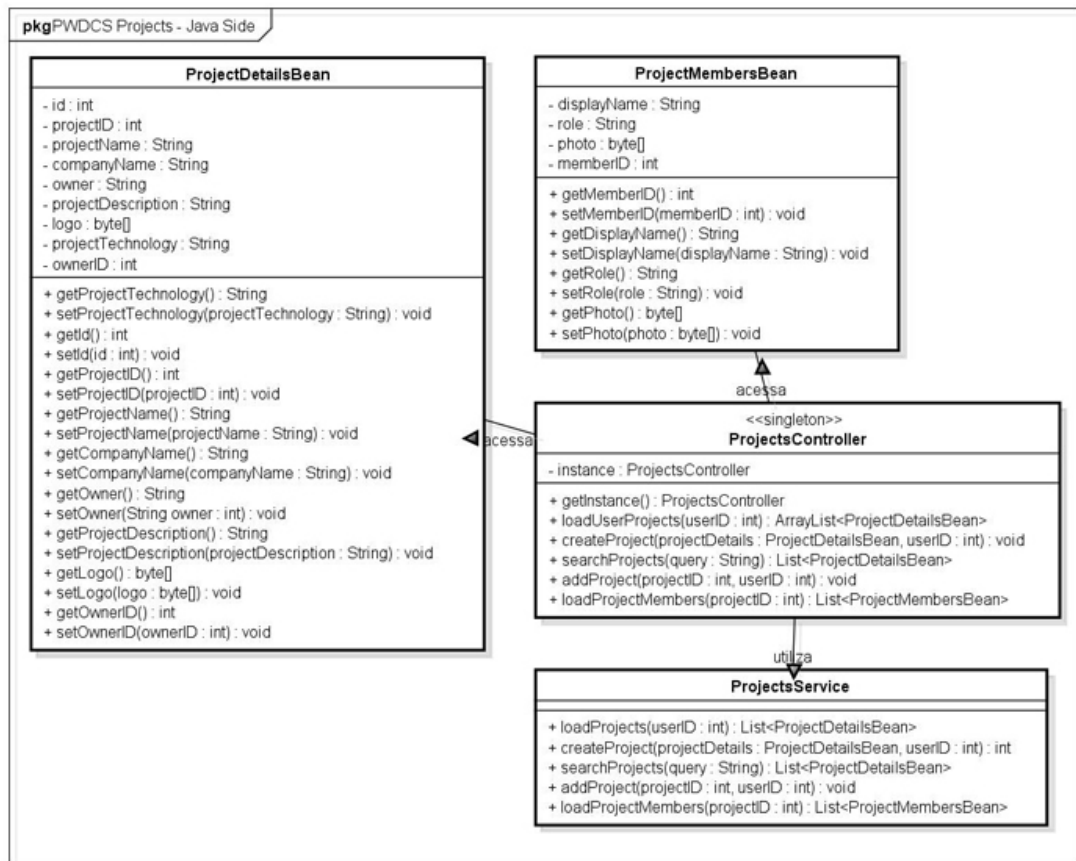


Figura 18. Diagrama de Classes: Rede Social – Projetos parte 1

A figura 19 apresenta as estruturas e seus relacionamentos com a Rede Social – Projetos (*back-end*) parte 2.

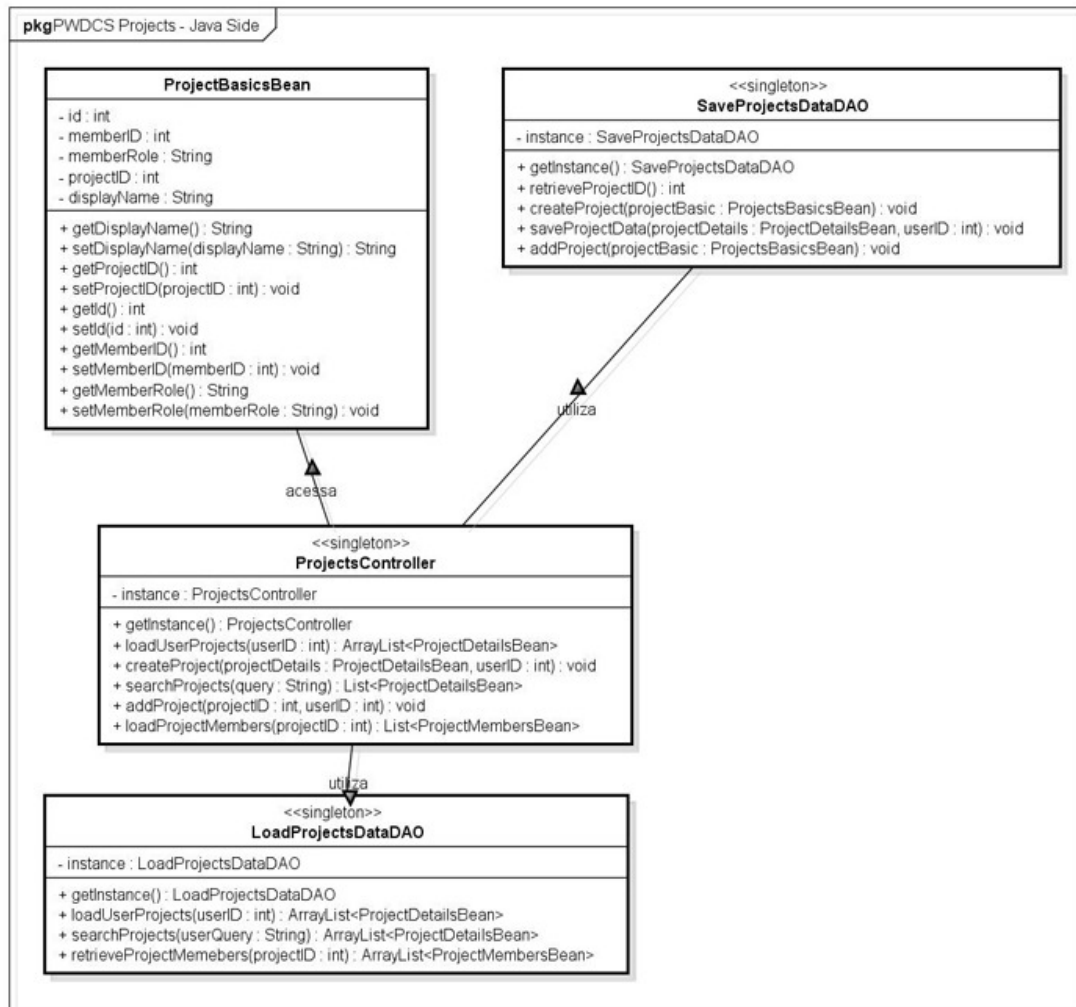


Figura 19. Diagrama de Classes: Rede Social – Projetos parte 2

3.2.2.5 Ferramentas – Gestão de Atividades (*front-end*)

A figura 20 apresenta as estruturas e seus relacionamentos com a sessão de Ferramentas – Gestão de Atividades (*front-end*).

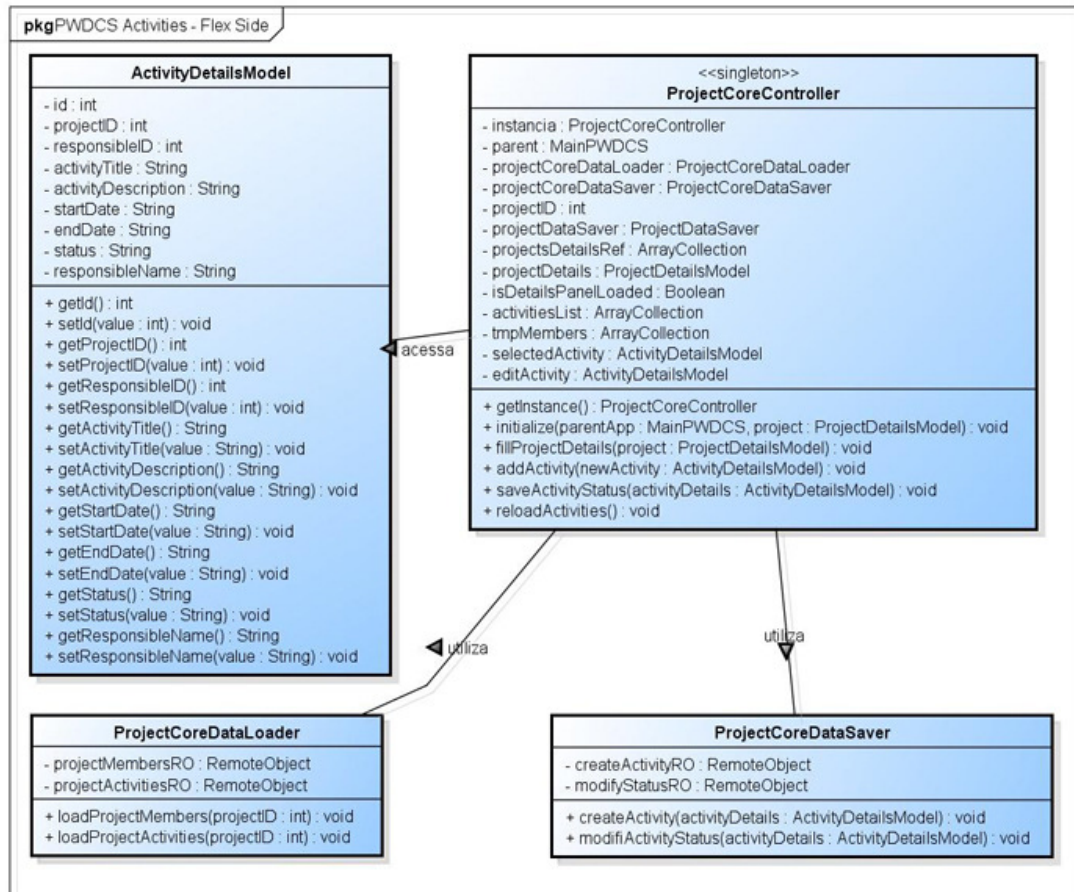


Figura 20. Diagrama de Classes: Rede Social – Gestão de Atividades

3.2.2.6 Ferramentas – Gestão de Atividades (*back-end*)

A figura 21 apresenta as estruturas e seus relacionamentos com a sessão de Ferramentas – Gestão de Atividades (*back-end*).

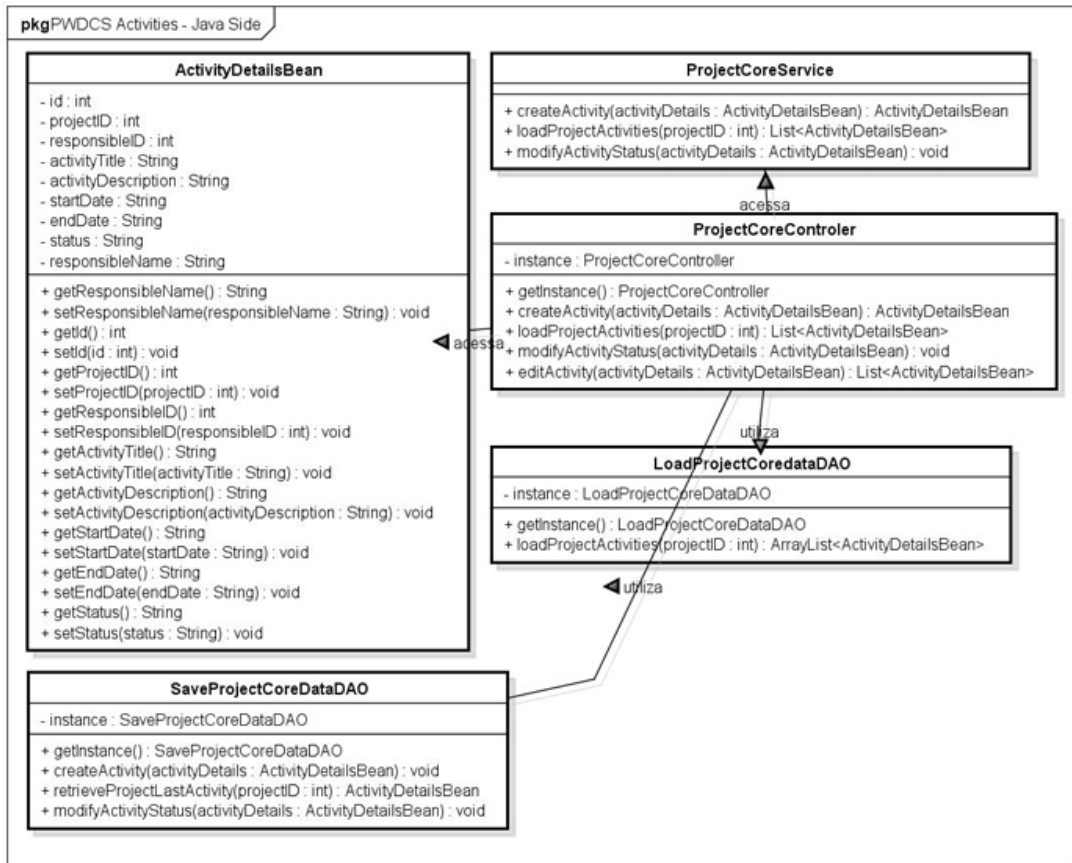


Figura 21. Diagrama de Classes: Rede Social – Gestão de Atividades

3.2.3 Diagramas de Sequência

Os diagramas de sequência representam as mensagens trocadas entre os objetos de um sistema, e na ordem temporal que tais trocas ocorrem (Pressman, 2005).

3.2.3.1 Adicionar Contato

A figura 22 apresenta a sequência das mensagens trocadas entre os objetos do *Developer Linker*, relativas ao caso de uso - Adicionar Contato.

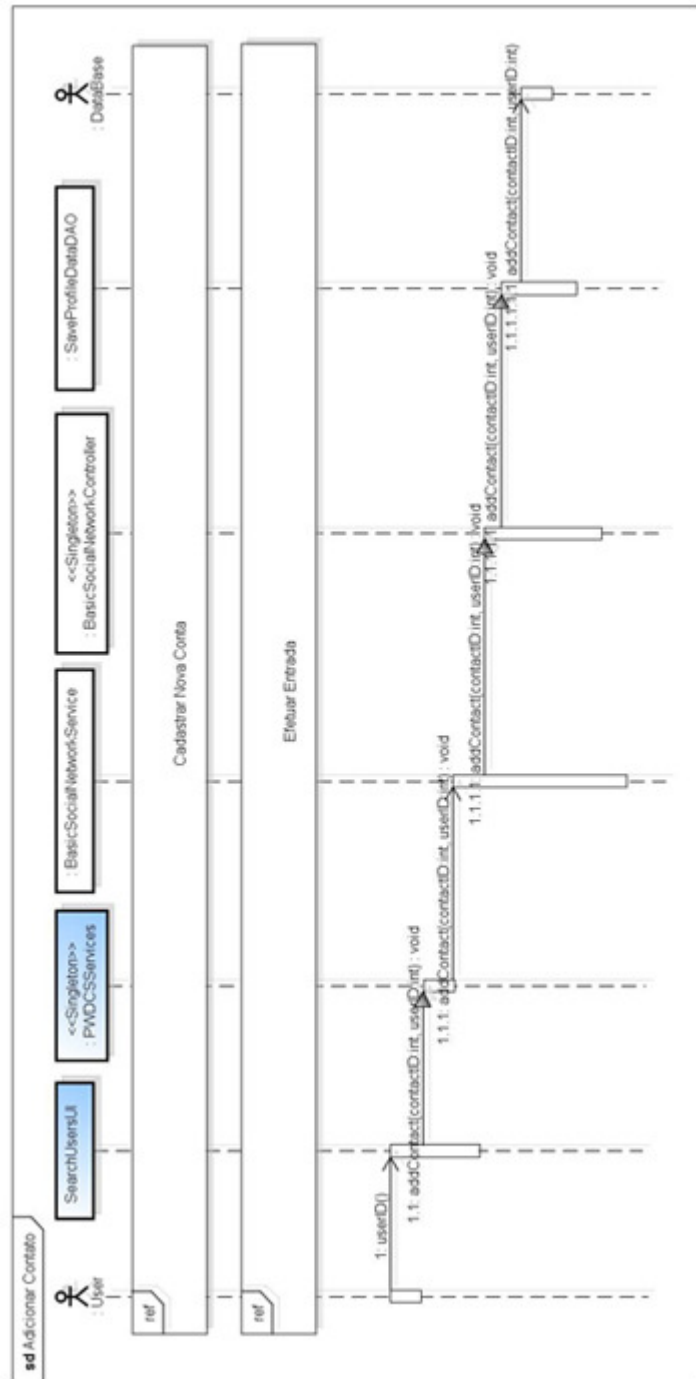


Figura 22. Diagrama de Sequência: Adicionar Contato

3.2.3.2 Cadastrar Nova Conta

A figura 23 apresenta a sequência das mensagens trocadas entre os objetos do *Developer Linker*, relativas ao caso de uso - Cadastrar Nova Conta.

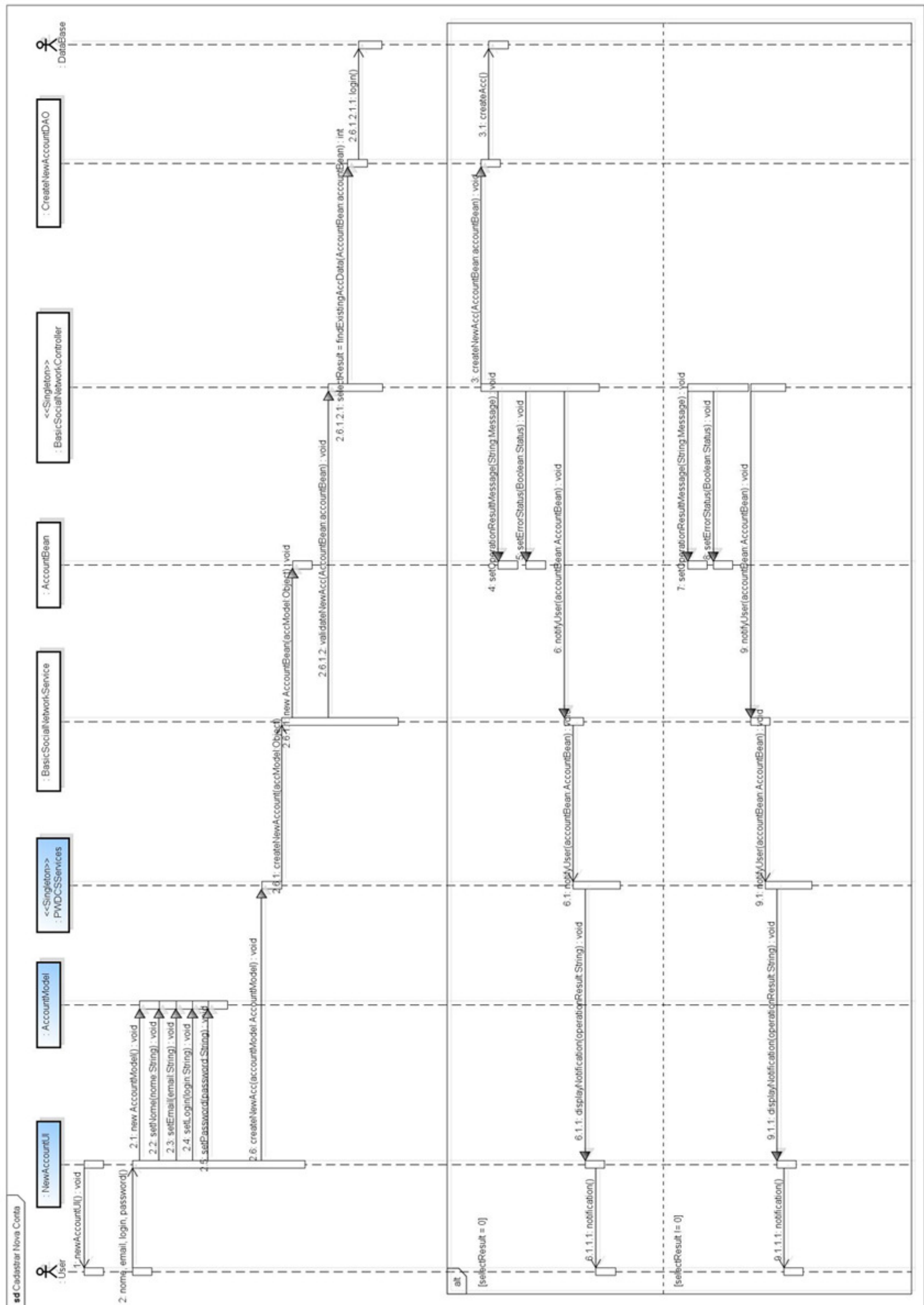


Figura 23. Diagrama de Sequência – Cadastrar Nova Conta

3.2.3.3 Efetuar Entrada

A figura 24 apresenta a sequência das mensagens trocadas entre os objetos do *Developer Linker*, relativas ao caso de uso – Efetuar Entrada.

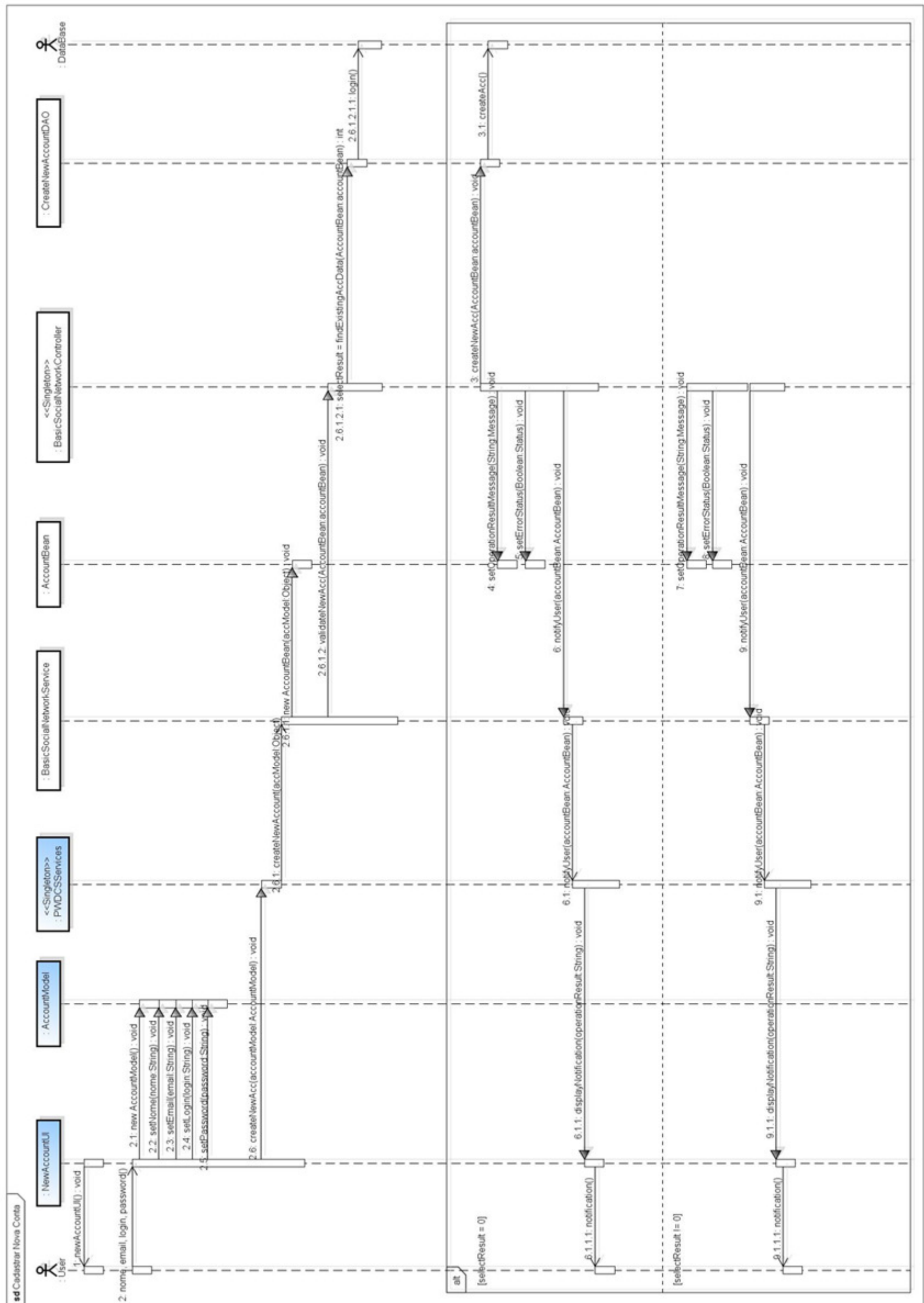


Figura 24. Diagrama de Sequência – Efetuar Entrada

3.2.3.4 Personalizar Perfil

A figura 25 apresenta a sequência das mensagens trocadas entre os objetos do *Developer Linker*, relativas ao caso de uso – Personalizar Perfil.

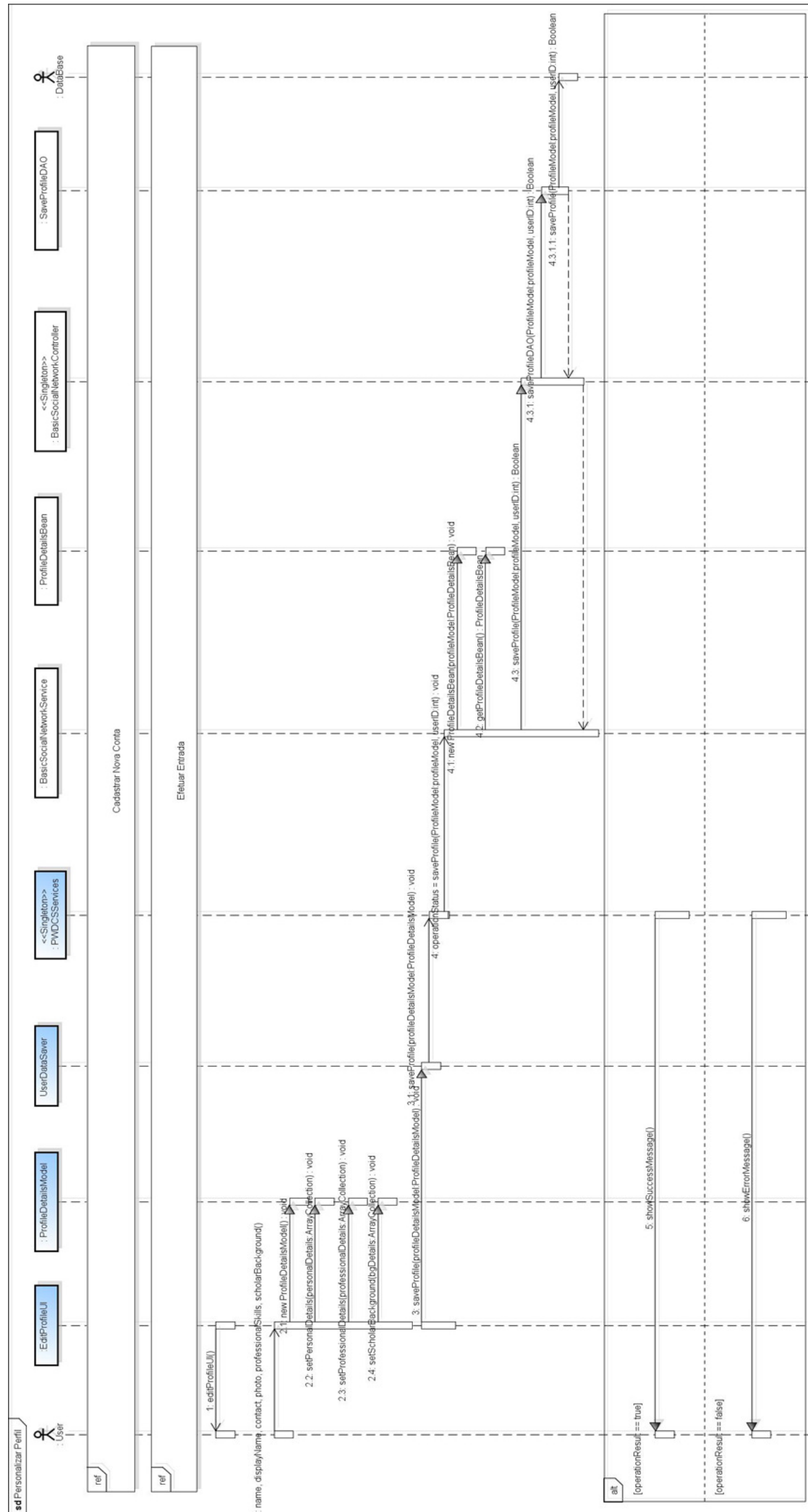


Figura 25. Diagrama de Sequência – Personalizar Perfil

3.2.3.5 Visualizar Detalhes do Contato

A figura 26 apresenta a sequência das mensagens trocadas entre os objetos do *Developer Linker*, relativas ao caso de uso – Visualizar Detalhes do Contato.

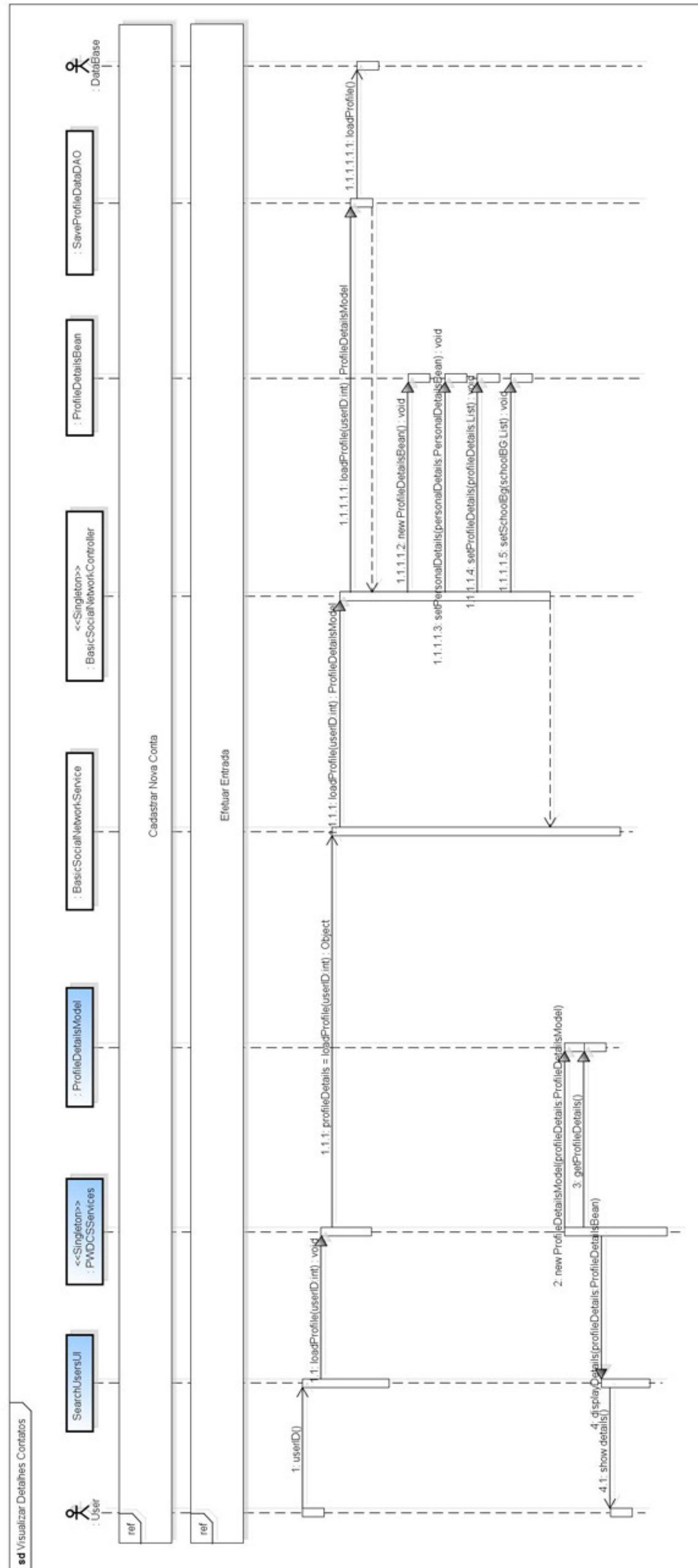


Figura 26. Diagrama de Sequência – Visualizar Detalhes do Contato

3.2.3.6 Pesquisar Usuários

A figura 27 apresenta a sequência das mensagens trocadas entre os objetos do *Developer Linker*, relativas ao caso de uso – Pesquisar Usuários.

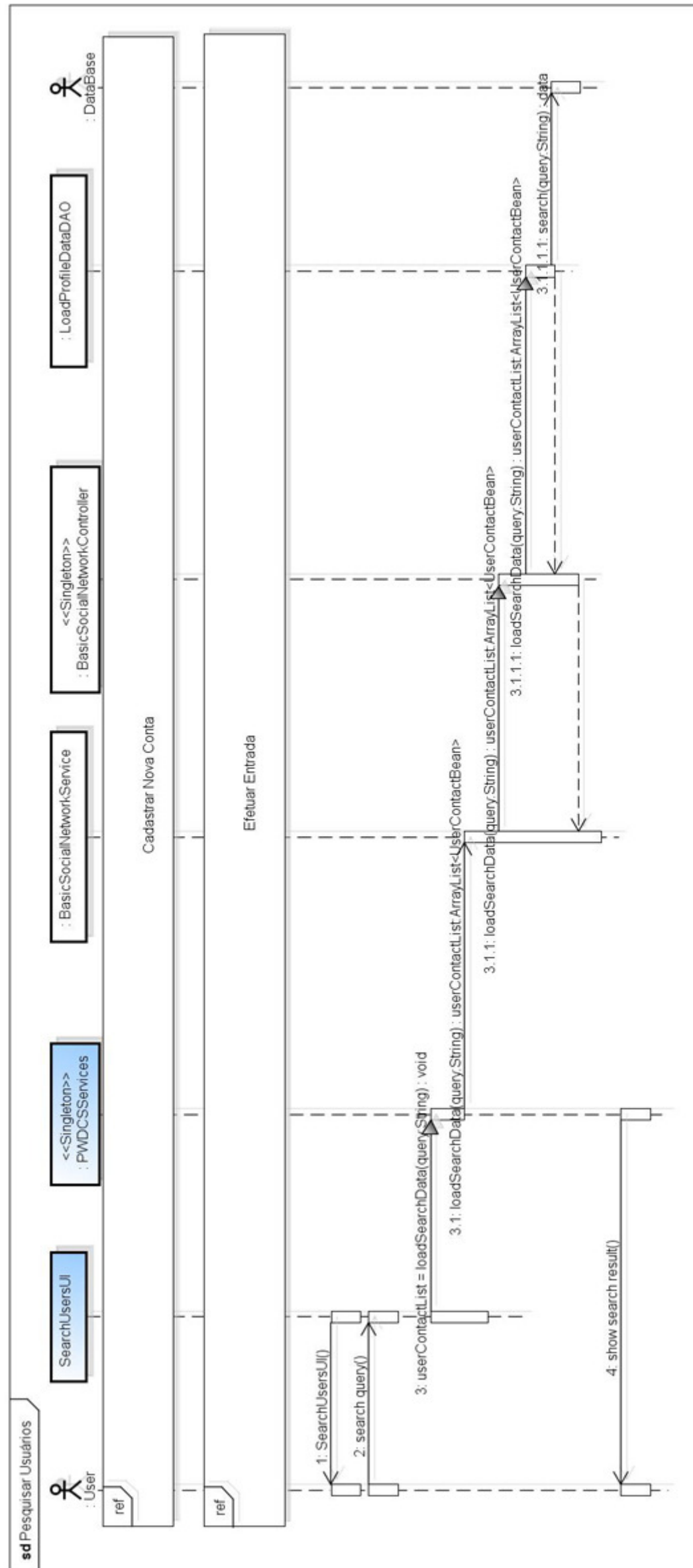


Figura 27. Diagrama de Sequência – Pesquisar Usuários

3.2.3.7 Visualizar Projetos

A figura 28 apresenta a sequência das mensagens trocadas entre os objetos do *Developer Linker*, relativas ao caso de uso – Visualizar Projetos.

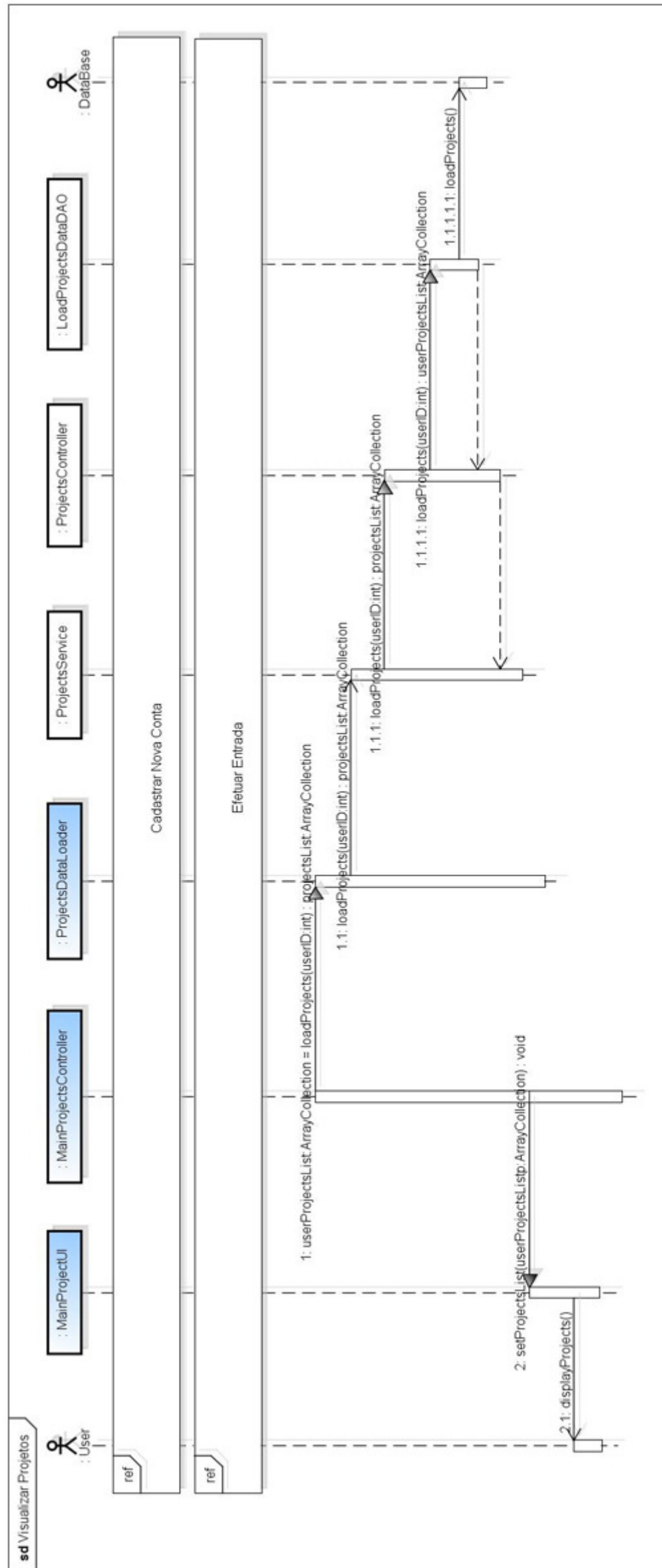


Figura 28. Diagrama de Sequência – Visualizar Projetos

3.2.3.8 Criar Projeto

A figura 29 apresenta a sequência das mensagens trocadas entre os objetos do *Developer Linker*, relativas ao caso de uso – Criar Projeto.

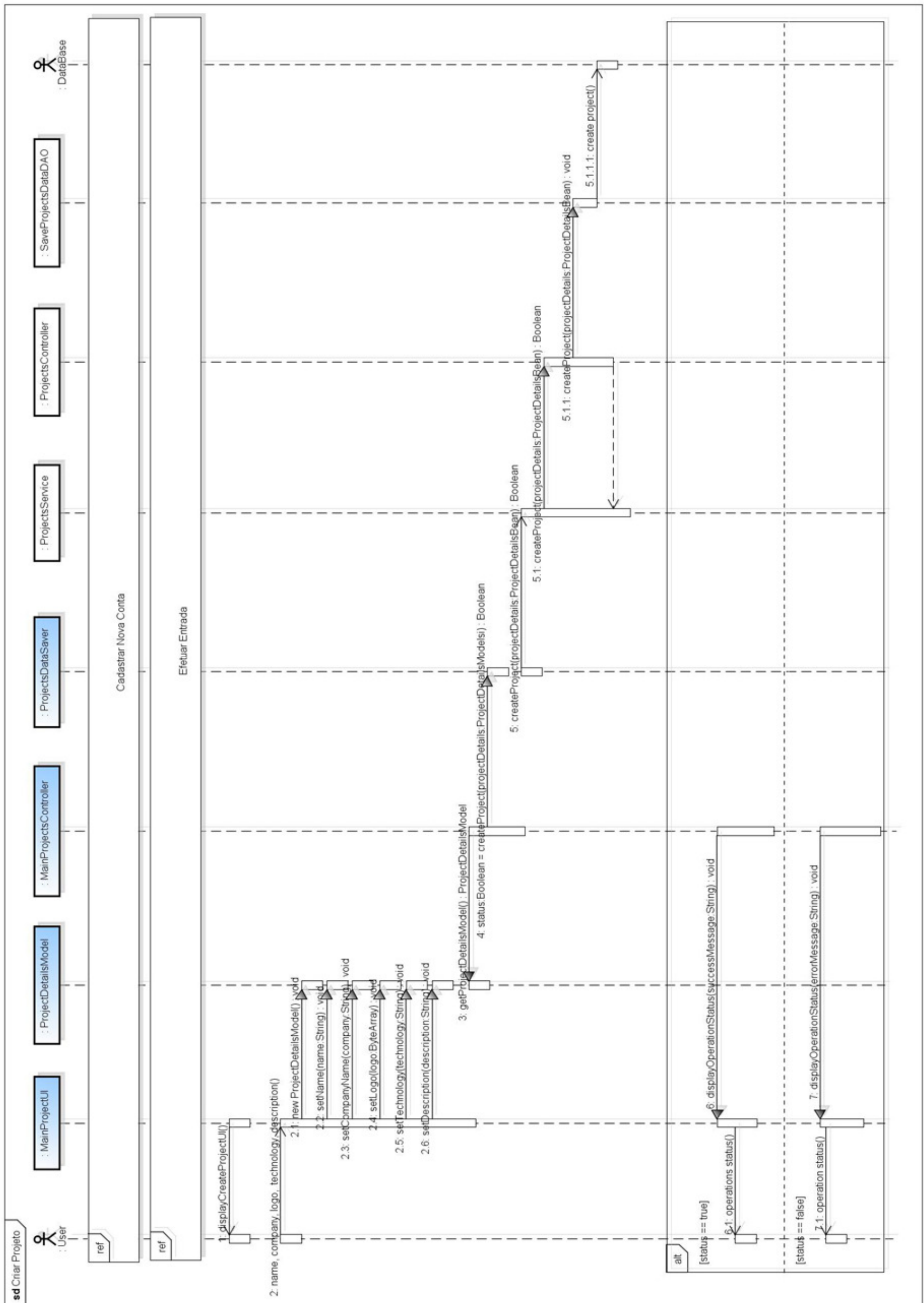


Figura 29. Diagrama de Sequência – Criar Projeto

3.2.3.9 Adicionar Projeto

A figura 30 apresenta a sequência das mensagens trocadas entre os objetos do *Developer Linker*, relativas ao caso de uso – Adicionar Projeto.

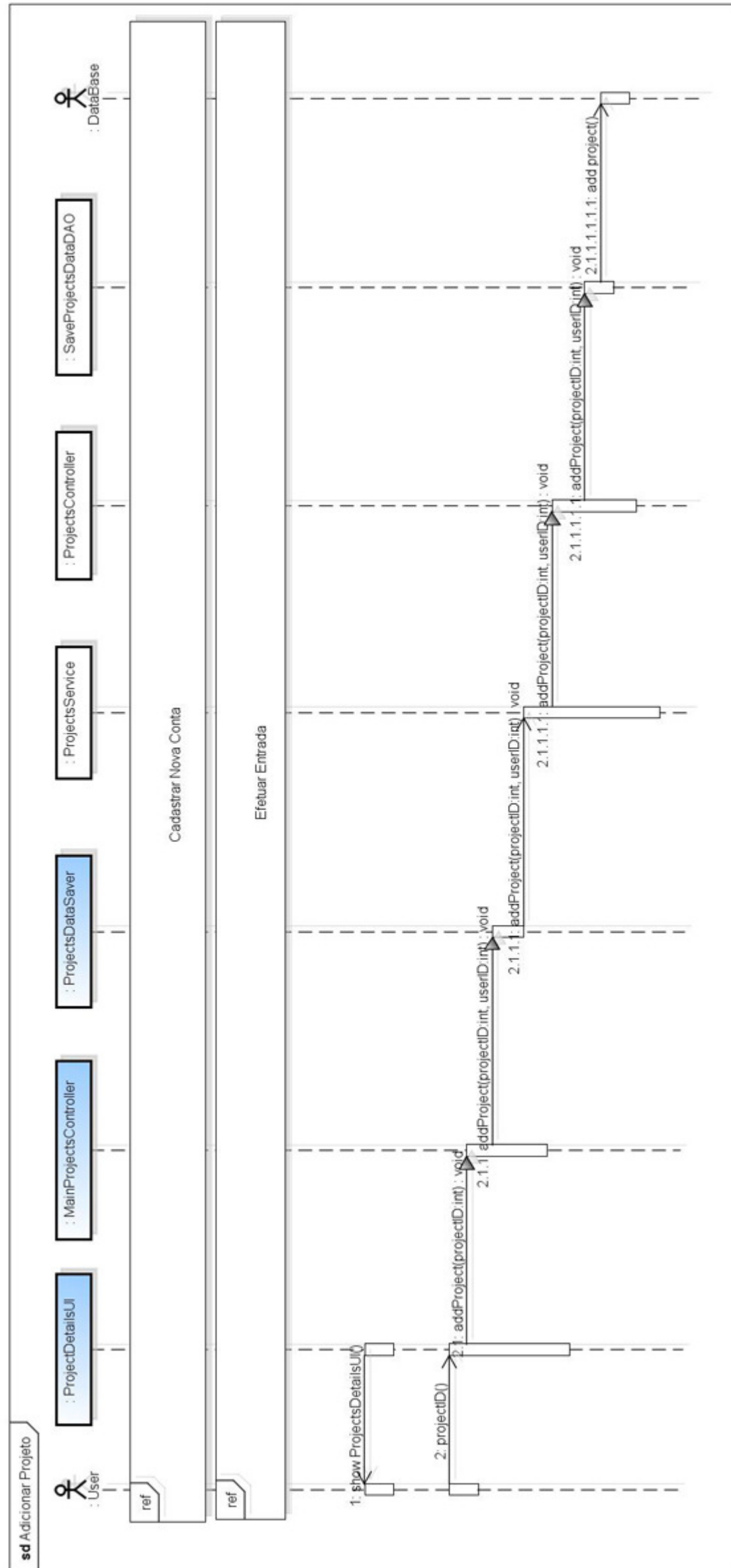


Figura 30. Diagrama de Sequência – Adicionar Projeto

3.2.3.10 Pesquisar Projetos

A figura 31 apresenta a sequência das mensagens trocadas entre os objetos do *Developer Linker*, relativas ao caso de uso – Pesquisar Projetos.

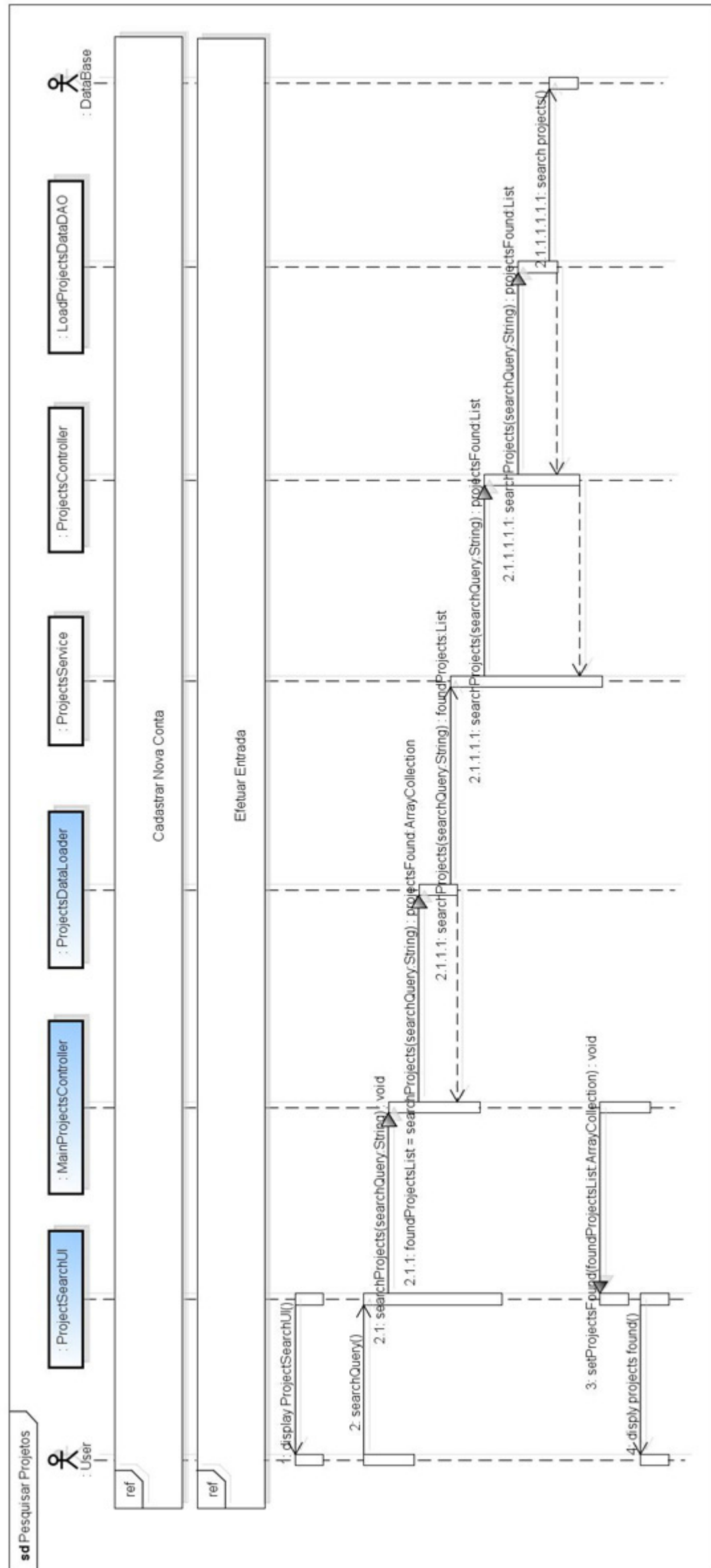


Figura 31. Diagrama de Sequência – Pesquisar Projetos

3.2.3.11 Visualizar Detalhes Projeto

A figura 32 apresenta a seqüência das mensagens trocadas entre os objetos do *Developer Linker*, relativas ao caso de uso – Visualizar Detalhes Projeto.

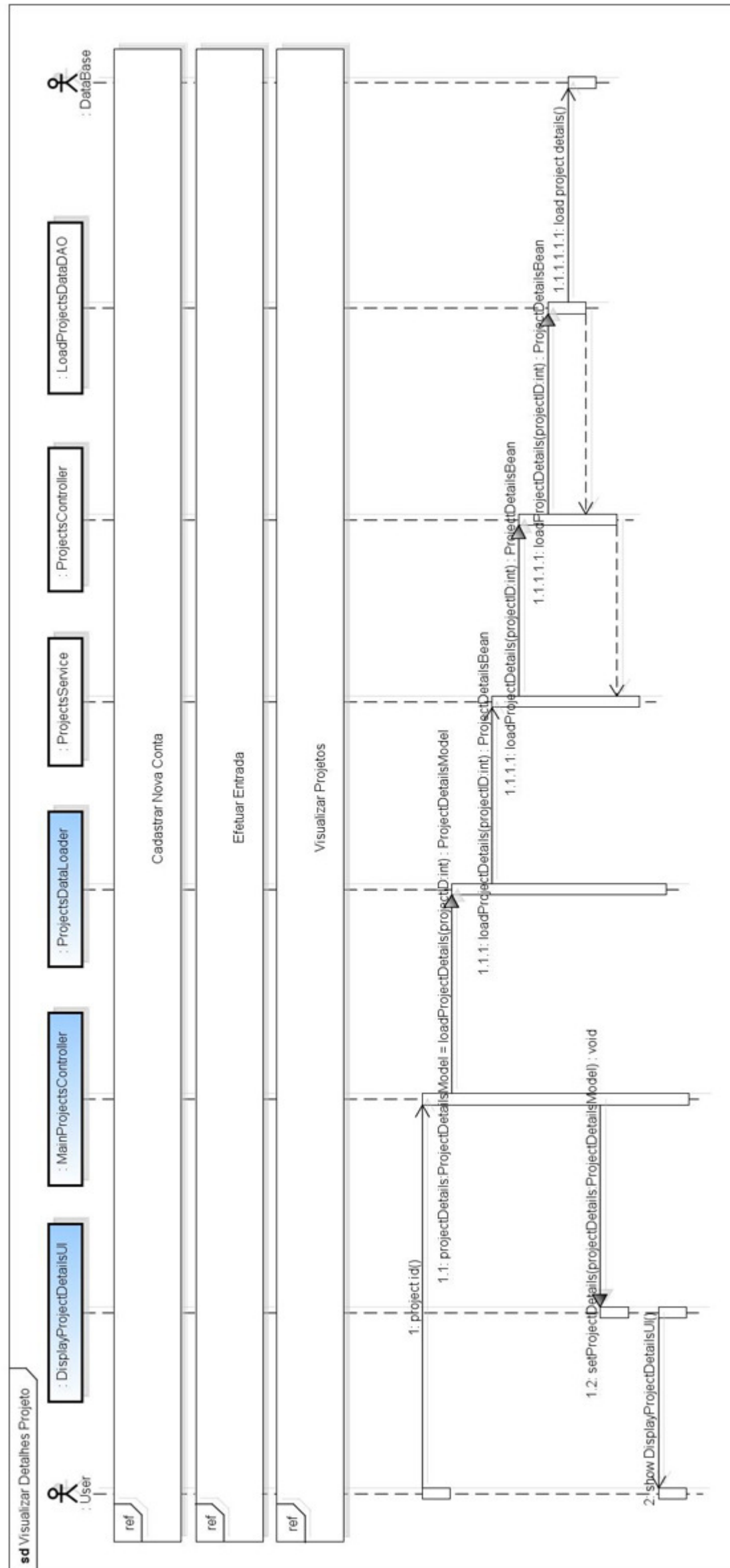


Figura 32. Diagrama de Sequência – Visualizar Detalhes Projeto

3.2.3.12 Criar Atividade

A figura 33 apresenta a sequência das mensagens trocadas entre os objetos do *Developer Linker*, relativas ao caso de uso – Criar Atividade.

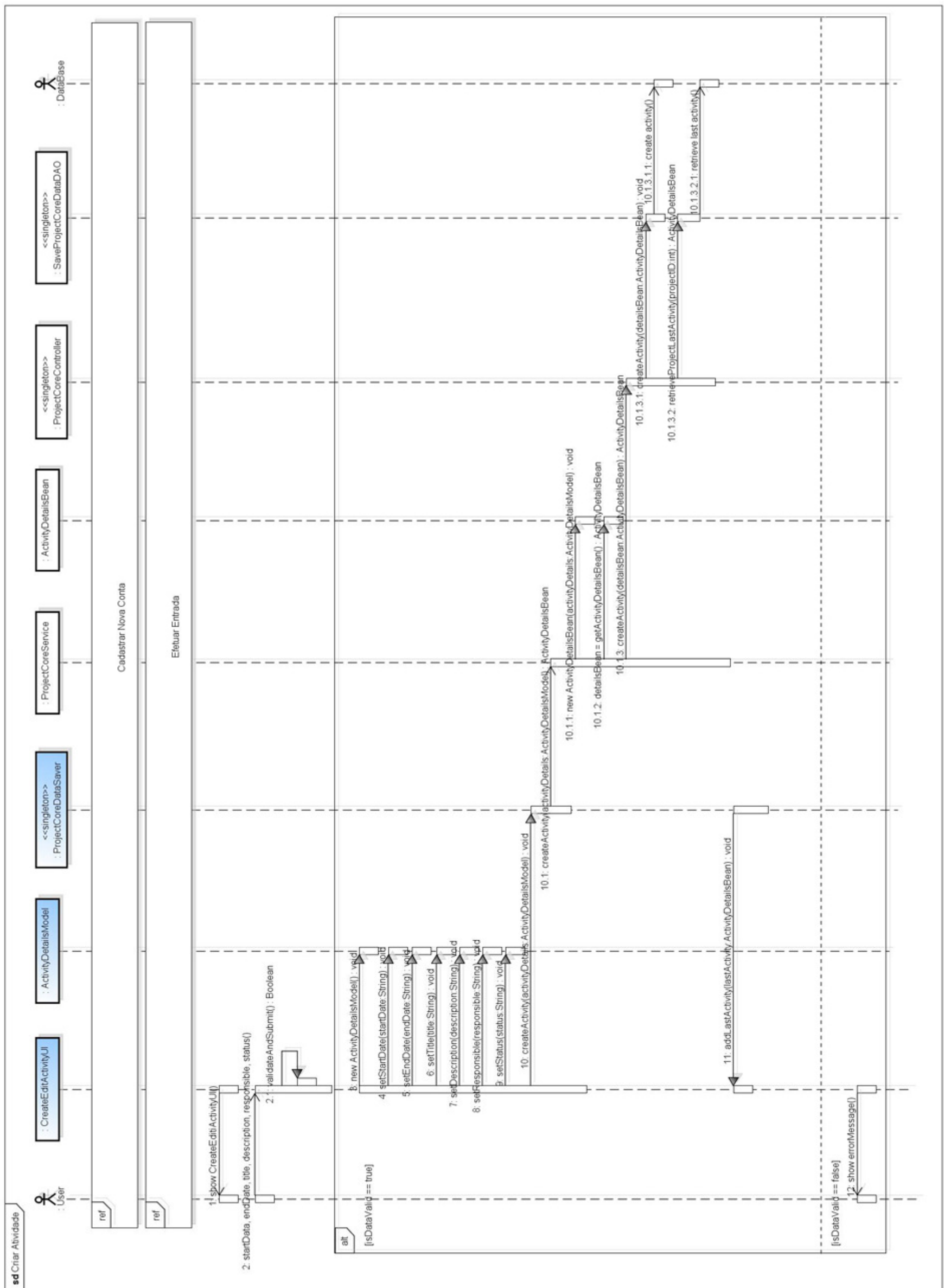


Figura 33. Diagrama de Sequência – Criar Atividade

3.2.3.13 Visualizar Atividades

A figura 34 apresenta a seqüência das mensagens trocadas entre os objetos do *Developer Linker*, relativas ao caso de uso – Visualizar Atividades.

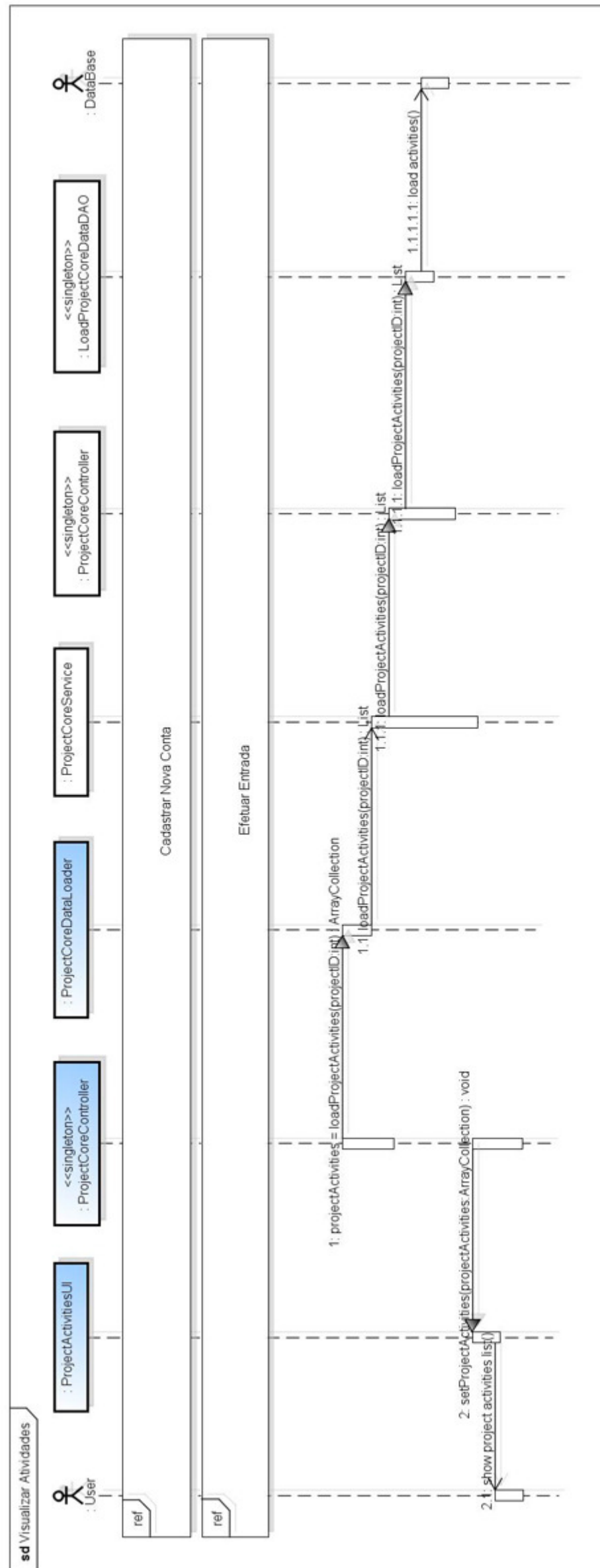


Figura 34. Diagrama de Sequência – Visualizar Atividades

3.2.3.14 Visualizar Detalhes Atividades

A figura 35 apresenta a seqüência das mensagens trocadas entre os objetos do *Developer Linker*, relativas ao caso de uso – Visualizar Detalhes Atividades.

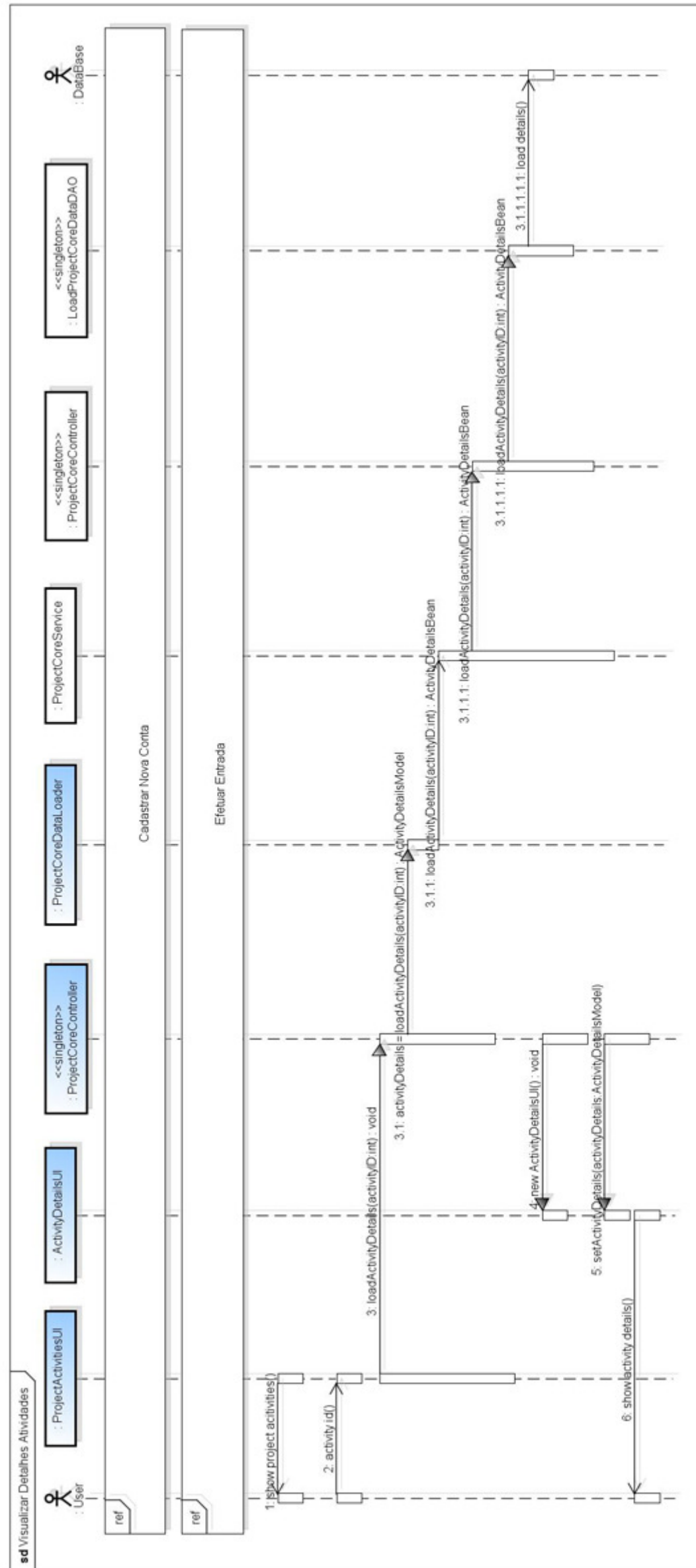


Figura 35. Diagrama de Sequência – Visualizar Detalhes Atividades

3.2.3.15 Delegar / Editar Atividade

A figura 36 apresenta a seqüência das mensagens trocadas entre os objetos do *Developer Linker*, relativas ao caso de uso – Delegar / Editar Atividade

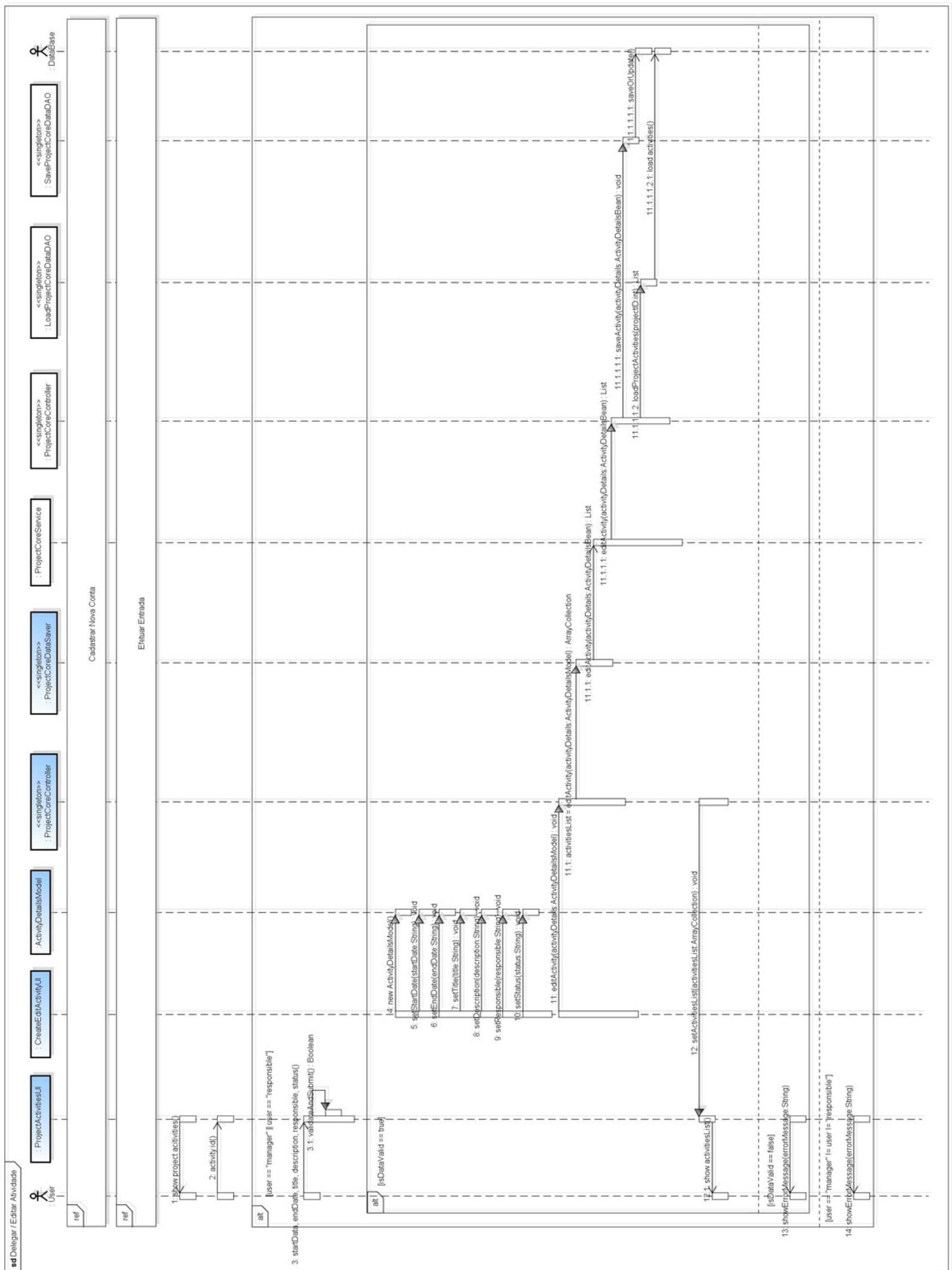


Figura 36. Diagrama de Sequência – Delegar / Editar Atividade

3.3 Interfaces com o Usuário (Telas)

Esta seção tem o objetivo de explicar o objetivo e o funcionamento das telas do *Developer Linker*.

3.3.1 Tela de Cadastro

A tela de cadastro tem o intuito de permitir que novas contas sejam criadas. Para tal, algumas informações são necessárias: nome, e-mail, *login* e senha.

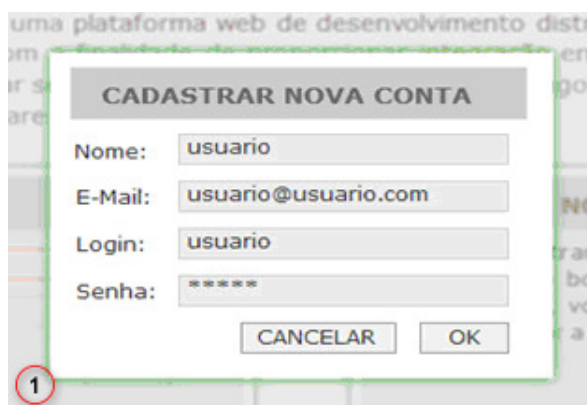


Figura 37. Tela – Cadastrar Nova Conta

3.3.2 Tela de Login

A tela de *login* é a primeira tela a ser exibida ao usuário. Por meio da tela de *login*, é possível realizar a entrada no sistema, assim como, o cadastro de uma nova conta.



Figura 38. Tela - Login

3.3.3 Tela Inicial do Perfil

A tela inicial do perfil é uma das principais telas do *Developer Linker*. Nela são exibidos os dados pessoais, dados profissionais, a lista de contatos, foto e o menu principal.

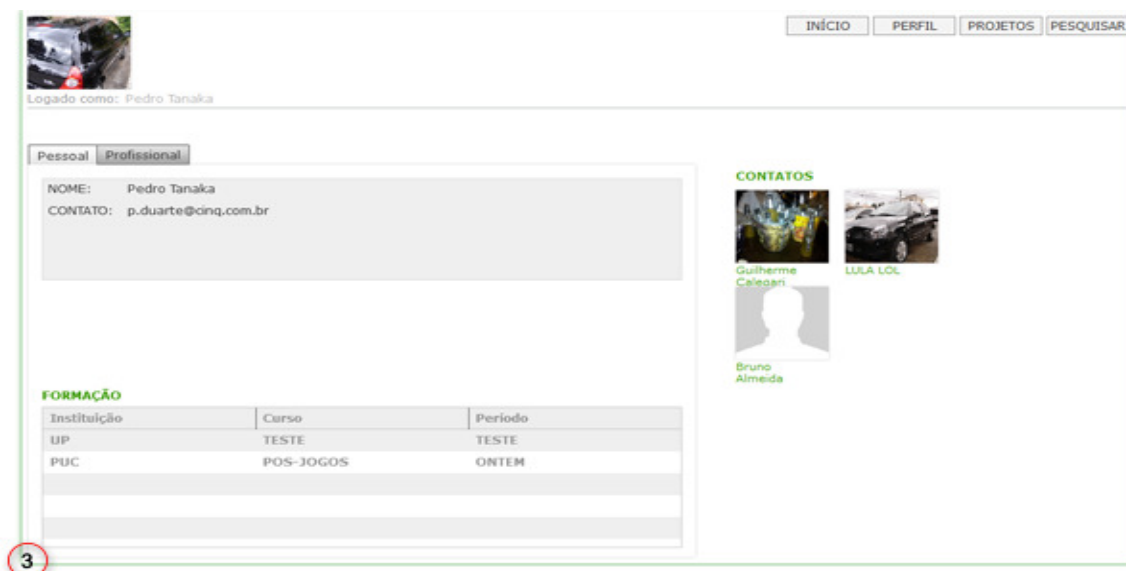


Figura 39. Tela – Inicial do Perfil

3.3.4 Tela de Alteração de Dados do Perfil

A tela de alteração dos dados do perfil tem o intuito de permitir ao usuário a personalização do perfil, podendo alterar ou adicionar dados, como: nome, e-mail, foto, histórico escolar e experiência profissional.



Figura 40. Tela – Inicial do Perfil

3.3.5 Tela de Visualização dos Detalhes dos Contatos

A tela de visualização dos detalhes dos contatos permite ao usuário, visualizar os detalhes do perfil de um contato pertencente a sua rede de contatos.

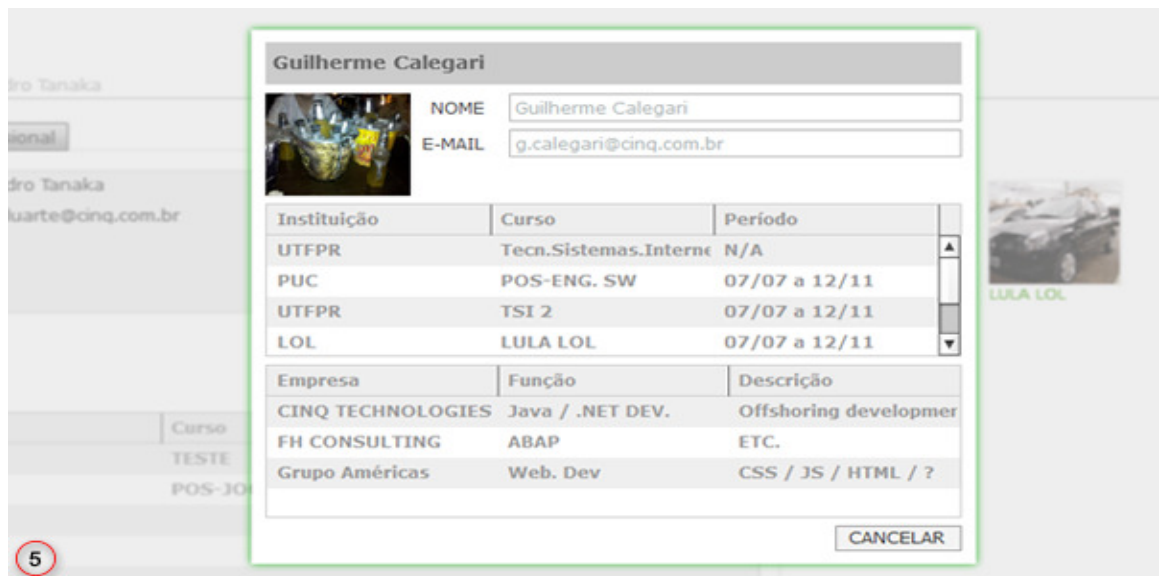


Figura 41. Tela – Inicial do Perfil

3.3.6 Tela de Pesquisa de Usuários

A tela de pesquisa de usuários fornece uma ferramenta de busca de usuários. Os critérios da busca são o nome ou parte do nome do usuário.



Figura 42. Tela – Pesquisa de Usuários

3.3.7 Tela Inicial do Mecanismo de Projetos de Software

A tela inicial do Mecanismo de Projetos de Software fornece uma visão dos projetos que o usuário participa, assim como, a opção de criar novos projetos e pesquisar projetos existentes.



Figura 43. Tela – Mecanismo de Projetos de Software

3.3.8 Tela de Pesquisa de Projetos

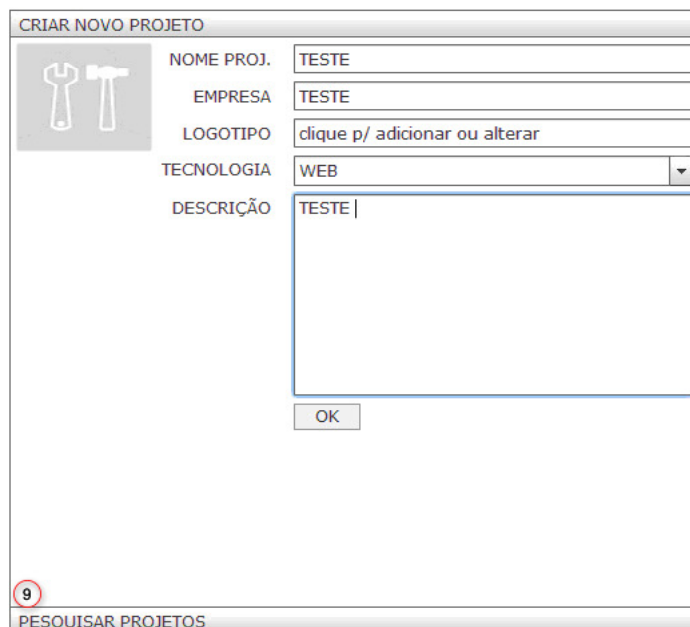
A tela de pesquisa de projetos fornece uma ferramenta de busca por projetos. Os critérios da busca são o nome ou parte do nome do projeto.



Figura 44. Tela – Pesquisa de Projetos

3.3.9 Tela de Criação de Projetos

A tela de criação de projetos fornece, aos usuários, a opção de criarem seus próprios projetos de software.

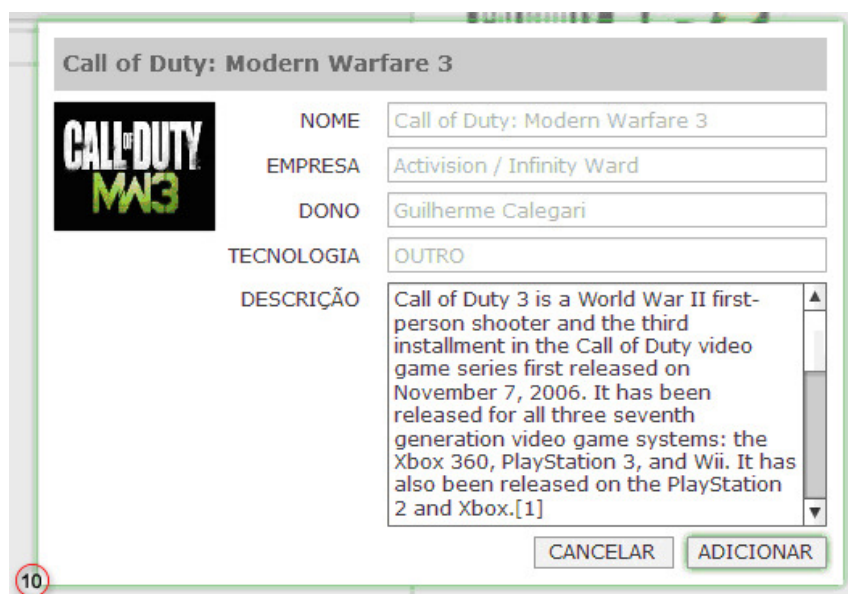


The screenshot shows a web form titled "CRIAR NOVO PROJETO". On the left, there is an icon of a wrench and a hammer. The form fields are: "NOME PROJ." with the value "TESTE"; "EMPRESA" with the value "TESTE"; "LOGOTIPO" with the text "clique p/ adicionar ou alterar"; "TECNOLOGIA" with a dropdown menu showing "WEB"; and "DESCRIÇÃO" with a text area containing "TESTE |". Below the description field is an "OK" button. At the bottom left, there is a small red circle with the number "9" and a link labeled "PESQUISAR PROJETOS".

Figura 45. Tela – Criação de Projetos

3.3.10 Tela de Visualização de Detalhes do Projeto

A tela de Visualização de Detalhes do fornece, aos usuários, uma listagem de características relacionadas a determinado projeto.



The screenshot shows a project details page for "Call of Duty: Modern Warfare 3". On the left is the game's logo. The details are: "NOME" (Call of Duty: Modern Warfare 3), "EMPRESA" (Activision / Infinity Ward), "DONO" (Guilherme Calegari), and "TECNOLOGIA" (OUTRO). The "DESCRIÇÃO" field contains a paragraph of text: "Call of Duty 3 is a World War II first-person shooter and the third installment in the Call of Duty video game series first released on November 7, 2006. It has been released for all three seventh generation video game systems: the Xbox 360, PlayStation 3, and Wii. It has also been released on the PlayStation 2 and Xbox.[1]". At the bottom right, there are "CANCELAR" and "ADICIONAR" buttons. At the bottom left, there is a small red circle with the number "10".

Figura 46. Tela – Visualização de Detalhes do Projeto

3.3.11 Tela Inicial de Projeto

A tela inicial de projeto fornece, aos usuários, opções de visualização dos membros do projeto, visualização de características gerais e opções de menu.

Battlefield 3 INICIO PROJETOS

DETALHES ATIVIDADES FALHAS

BATTLEFIELD 3

NOME PROJETO: Battlefield 3
EMPRESA: EA / Dice
DONO: Guilherme Calegari
TECNOLOGIA: OUTRO
DESCRIÇÃO:
In Battlefield 3, players step into the role of the elite U.S. Marines. As the first boots on the ground, players will experience heart-pounding missions across diverse locations including Paris, Tehran and New York. As a U.S. Marine in the field, periods of tension and anticipation are punctuated by moments of complete chaos. As bullets whiz by, as walls crumble, as explosions force players to the ground, the battlefield feels more alive and interactive than ever before.

MEMBROS

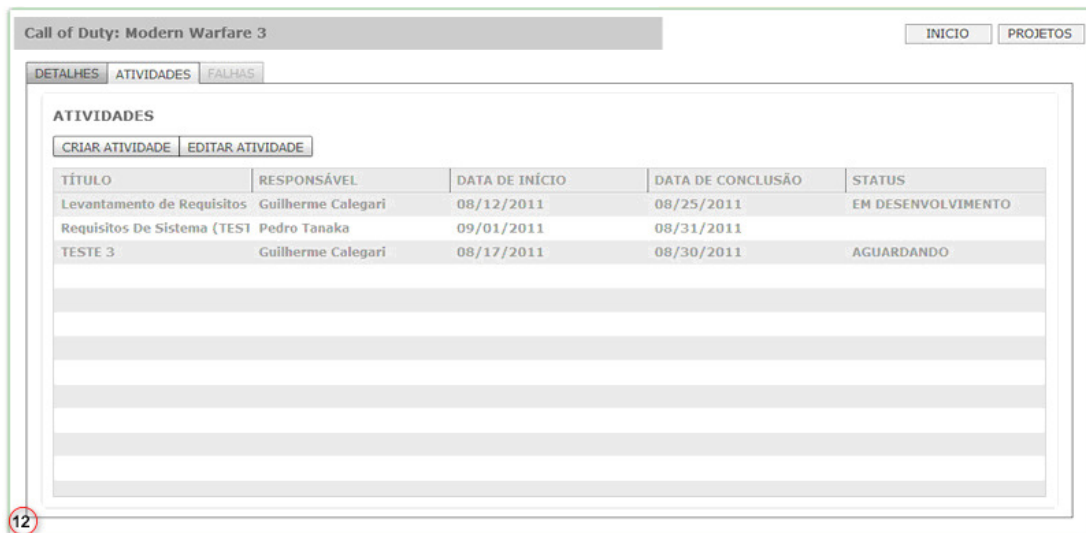
FOTO	NOME	PAPEL
	Guilherme Calegari	Manager
	Pedro Tanaka	Developer
	Unknow Name	Developer

11

Figura 47. Tela – Inicial de Projeto

3.3.12 Tela de Gestão de Atividades do Projeto

A tela de Gestão de Atividades do Projeto fornece, aos usuários, a listagem das atividades do projeto, além disso, também permite aos usuários a criação, deleção e edição de atividades.



TÍTULO	RESPONSÁVEL	DATA DE INÍCIO	DATA DE CONCLUSÃO	STATUS
Levantamento de Requisitos	Guilherme Colegari	08/12/2011	08/25/2011	EM DESENVOLVIMENTO
Requisitos De Sistema (TESTE 3	Pedro Tanaka	09/01/2011	08/31/2011	
TESTE 3	Guilherme Colegari	08/17/2011	08/30/2011	AGUARDANDO

Figura 48. Tela – Gestão de Atividades do Projeto

3.3.13 Tela de Criação e Edição de Atividade do Projeto

A tela de Criação e Edição de Atividades do Projeto fornece, aos usuários, a opção de criar ou alterar determinada atividade fornecendo os dados necessários: título, início, término, responsável, status e descrição.

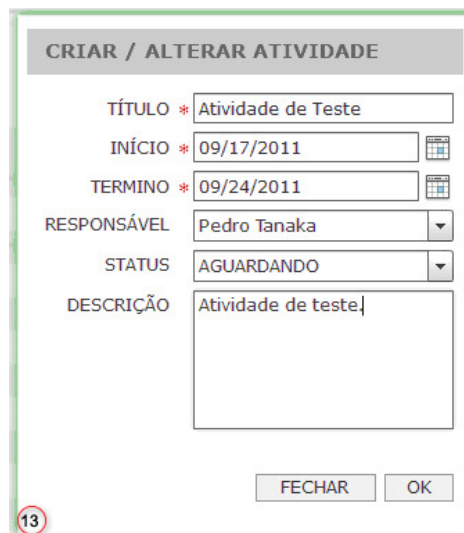


Figura 49. Tela – Gestão Criação e Edição de Atividades do Projeto

3.3.14 Tela de Visualização dos Detalhes da Atividade

A tela de Visualização dos Detalhes da Atividade fornece, aos usuários, uma visão expandida dos detalhes de determinada atividade, sendo possível também, a alteração do status da atividade.



Figura 50. Tela – Visualização dos Detalhes da Atividade

3.4 Sequência de Navegação das Telas

O esquema a seguir, ilustra a ordem pelas quais, as tela dos *Developer Linker* são exibidas. O esquema reflete a sequência da numeração contida nas figuras relativas as telas.

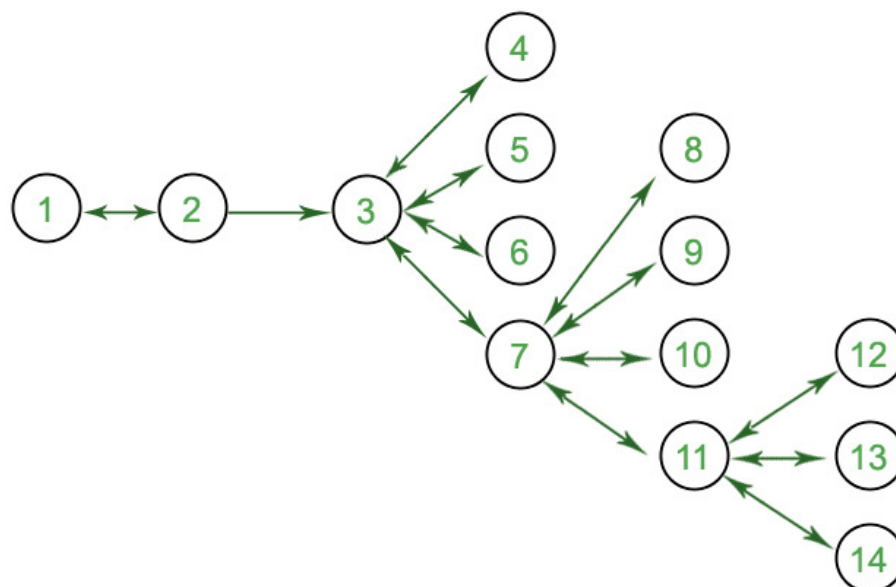


Figura 51. Esquema– Sequência de Navegação de Telas

4 CONCLUSÃO

Apesar de ser uma área teoricamente nova no ramo da informática, o desenvolvimento de software geograficamente distribuído, vem se mostrando uma estratégia viável tanto para grandes empresas, quanto para empresas terceirizadoras de mão de obra (intermediárias). Em especial para aquelas que trabalham com o grupo dos países emergentes, BRICS (Brasil, Rússia, Índia, China e África do Sul), pois, a abundância de mão de obra qualificada aliada ao baixo custo (salários, impostos, etc.), têm permitido que parte da demanda mundial por sistemas de informação seja suprida.

Partindo do pressuposto de que, a demanda mundial por sistemas de informação, é superior a capacidade de desenvolvimento e que o valor da mão de obra, oriunda de países considerados desenvolvidos, é superior a de países considerados emergentes, o que impossibilitaria financeiramente grande parte dos sistemas desenvolvidos no mundo hoje, pode-se constatar que, o *Offshoring* continuará sendo praticado. Desta forma, a expansão do assunto e o desenvolvimento de boas práticas e ferramentas se mostram cabíveis para a continuidade e melhoria dos processos e ferramentas já existentes.

Pode-se, então, concluir que a proposta deste trabalho de conclusão de curso (*Developer Linker*), enquadra-se nesta atual demanda, sendo a proposta, constituída por um conjunto de ferramentas colaborativas, interligadas por uma rede social construída especificamente para instituições e pessoas diretamente associadas ao desenvolvimento de software e que sua viabilidade tanto tecnológica quanto organizacional se demonstraram pouco custosas em relação aos benefícios oferecidos.

Em linhas gerais, o desenvolvimento do protótipo, demonstrou-se mais árduo do que o esperado inicialmente, visto que, a interface com o usuário demandou esforços adicionais, principalmente no manuseio das respostas vindas do servidor. Entretanto, o lado do servidor foi desenvolvido da maneira esperada, sem grandes surpresas, resultando em um *back-end* simples e eficaz. Embora o protótipo tenha apenas uma quantia limitada de funcionalidades, poderia ser usado para a gestão das atividades reais de um

projeto de software sem maiores dificuldades, pois, as funcionalidades tidas como essenciais para o seu objetivo, foram implementadas com sucesso (rede social, projetos de software e ferramenta de gestão de atividades).

Por último, há de ser mencionado, o conhecimento adquirido pelo autor do trabalho sobre assuntos bastante recentes na engenharia de software, sendo eles, o trabalho colaborativo apoiado por computador e o desenvolvimento distribuído de software. Além disso, o conhecimento e experiência nas tecnologias utilizadas para no desenvolvimento do *Developer Linker* (*Adobe Flex*, *Oracle Java*, *JBoss Hibernate* e *Oracle MySqI*).

4.1 Trabalhos Futuros

De acordo com o título, este trabalho, trata-se de um protótipo, sendo este, o resultado de uma pesquisa sobre o trabalho colaborativo apoiado por computador, voltado ao desenvolvimento distribuído de software. Contudo, tal pesquisa, foi impulsionada pela necessidade do desenvolvimento do presente trabalho de diplomação (graduação), deste modo, uma futura expansão da pesquisa, em maior nível de detalhamento, fica em aberto.

Além de um maior aprofundamento da pesquisa sobre o tema, a implementação (protótipo – *Developer Linker*) abrangeu apenas requisitos básicos de funcionamento, assim, uma implementação mais detalhista seria requerida para que o *Developer Linker* pudesse vir a se tornar um software completamente funcional (produto).

As seguintes propostas foram classificadas como alternativas para possíveis trabalhos futuros, a partir do que foi feito neste trabalho de conclusão de curso:

- Aprofundamento da pesquisa.
- Desenvolvimento de um produto a partir do protótipo.

5 Referências

LIMA, GERCINA ÂNGELA B.; PINTO, LÍLIAM PACHECO; LAIA, MARCONI MARTINS. Tecnologia da Informação: Impactos na Sociedade. Brasil, 2001.

AUDY, JORGE; PRIKLADNICKI, RAFAEL. Desenvolvimento Distribuído de Software: Desenvolvimento de Software com Equipes Distribuídas. Brasil, 2007.

ARAÚJO, JOSÉ BRAZ; CAMPANÁRIO, MILTON DE ABREU. Trabalho Colaborativo e *Groupware*. Brasil, 2004.

SOMMERVILLE, IAN. Engenharia de Software. Brasil, 2007.

WERTHEIN, JORGE. A sociedade da informação e seus desafios. Brasil, 2000.

JIMÉNEZ, MIGUEL; PIATTINI, MARIO; VIZCÁMO AURORA. *Challenges and Improvements in Distributed Software*. Espanha, 2008.

LANUBILE, FILLIPO. *Collaboration in Distributed Software Development*. Italia, 2009.

PRESSMAN, ROGER S. *Software Engineering: A Practitioner's Approach*. Estados Unidos, 2005.