

## **4**

### **Resultados Finais**

Neste capítulo, são apresentados os resultados do trabalho, onde é apresentado, como primeiro ponto, uma seção de cenários de uso, a qual permite ver como o framework pode ser instanciado em aplicações orientadas ao serviço médico. Como segundo ponto, é apresentada uma avaliação do framework, que é feita aplicando o modelo GQM que permite ver as virtudes do framework e os pontos que deveriam ser melhorados a partir da perspectiva de desenvolvedores.

#### **4.1.**

##### **Cenários de uso**

Esta seção apresenta exemplos de cenários de uso onde pode ser instanciado o framework proposto nesta dissertação. Inicialmente, apresentam-se os cenários de monitoramento fetal e de administração de medicamentos, os quais são explicados de forma detalhada. Em seguida, apresentam-se alguns outros exemplos que não foram implementados, mas servem para ilustrar as possibilidades de instanciação que o framework oferece.

##### **4.1.1.**

###### **Monitoramento Fetal**

No trabalho (SU et al., 2014) é apresentada uma aplicação que permite realizar monitoramento fetal de forma remota, a partir de agentes de software. Aproveitando esta ideia e com o propósito de mostrar que é possível criar uma instância similar à proposta nesse trabalho, nesta seção descreveremos como é possível usar o framework para implementar uma aplicação colaborativa que permita realizar o monitoramento de uma gestante, a fim de saber o estado de saúde do feto.

O primeiro passo será apresentar a ideia principal, seguida da explicação de como foi realizada a implementação.

#### 4.1.1.1.

##### **Ideia principal**

A ideia da implementação deste cenário é que se possa contar com um aplicativo móvel que tenha a possibilidade de obter dados sobre o feto a partir de um dispositivo externo via *bluetooth*. Depois de obter esta informação, ela precisa ser enviada a um especialista, de modo que o especialista possa analisar a informação e fornecer um diagnóstico preventivo, para que sejam tomadas as medidas adequadas. Já que os dados são quantificáveis e assume-se que se conhece o perfil clínico da gestante, o diagnóstico será feito de forma automática pelo computador, no caso específico, por um agente que represente um médico obstetra que, a partir de uma base de conhecimento parametrizada, poderá fazer os diagnósticos.

A não detecção de problemas por parte do especialista implica no término do fluxo, devendo-se, neste caso, esperar até que aconteça um novo ciclo de monitoramento. Entretanto, caso o especialista se depare com algum problema, é preciso que alerte o médico titular da gestante, para que este possa tomar uma ação. Para este cenário de uso e com o propósito de mostrar a possibilidade de colaboração que qualquer agente que represente um membro da equipe médica poderia ter ao detectar um problema no estado de saúde do feto, o médico responsável irá comunicar-se automaticamente com uma ambulância disponível, solicitando que a paciente seja assistida.

#### 4.1.1.2.

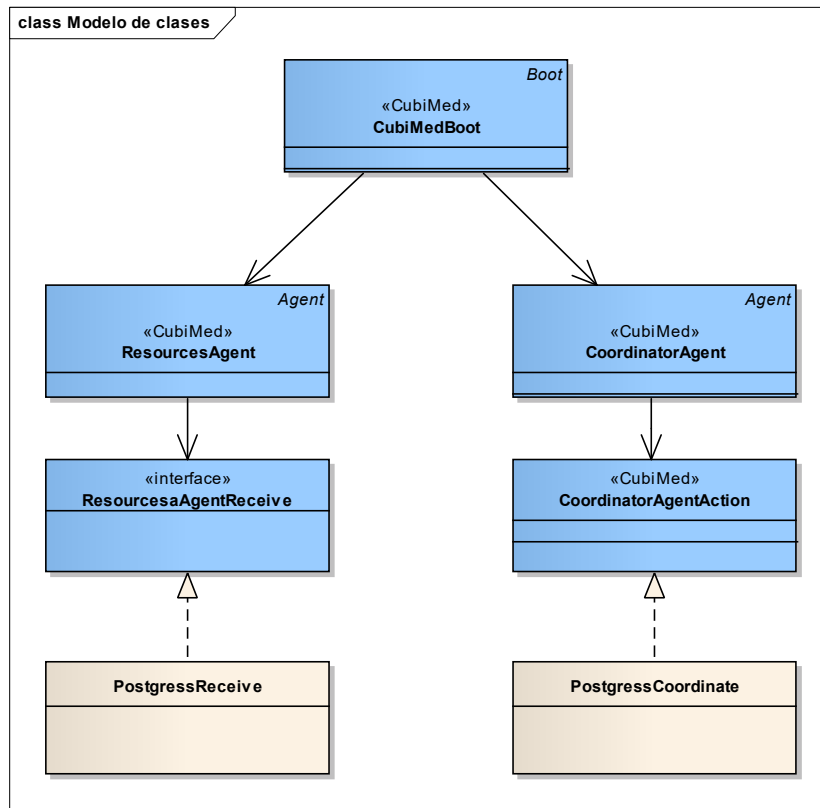
##### **Implementação**

A implementação deste cenário de uso está composta pelo servidor principal e por quatro aplicativos colaborativos, os quais são descritos a seguir.

*Servidor Principal:* No servidor principal, são instanciados o agente coordenador e o agente de recursos. O primeiro terá a tarefa de informar aos participantes sobre quem ingressou e quem saiu da aplicação. Neste sentido, este agente cumpre com as seguintes regras:

- O Agente Gestante pode visualizar apenas os agentes de tipo obstetra e o agente médico, que é seu médico titular.
- O Agente Obstetra pode ver todas as gestantes e os médicos.
- O Agente Médico Titular pode ver os seus pacientes e todos os membros do staff que estão ativos.

- O Agente Ambulância pode ver os médicos e os pacientes.
- O agente de recursos tem acesso a um banco de dados de tipo PostgreSQL, que conterà registros sobre os perfis das gestantes, sendo possível fornecer esta informação ao Agente Obstetra, quando for preciso.

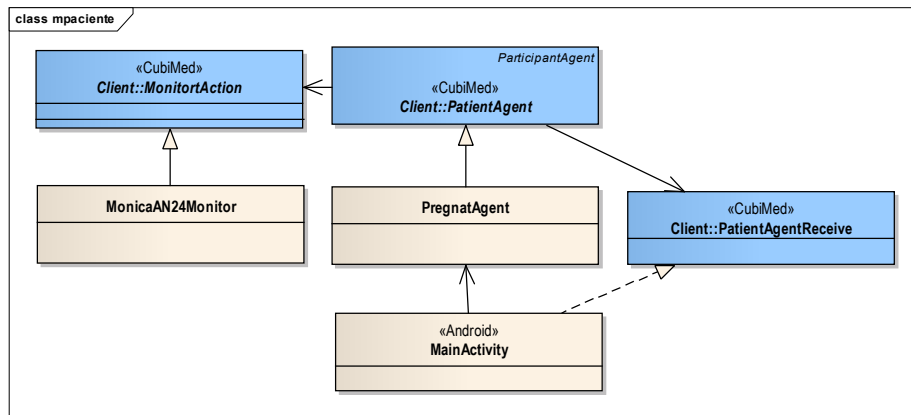


**Figura 19:** Diagrama de classes Servidor Principal – Cenário Monitoramento Fetal

*Aplicativo Gestante:* O aplicativo da gestante será criado na plataforma Android para que possa ser executado em um *smartphone* com conexão *bluetooth*. Neste dispositivo, será instanciado um Agente Gestante, que é uma extensão do Agente Paciente, o qual tem como tarefa recolher informações sobre a saúde do feto, a partir de um dispositivo externo. Neste caso específico, estamos nos referindo a dispositivos como o MonicaAN24 que é um cardiotocógrafo digital, fruto de uma pesquisa de 15 anos feita na Universidade de Nottingham, Reino Unido (MONICAA24, 2015). Com este dispositivo, a traves de eletrodos que são aderidos ao abdômen materno, poderão ser obtidos os seguintes dados:

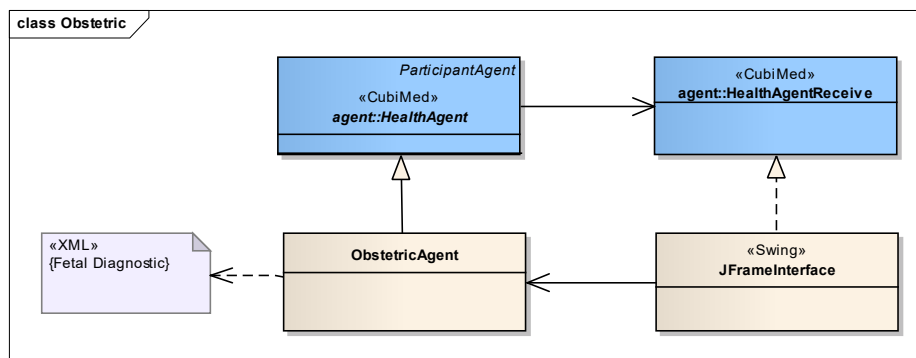
- Batimento Cardíaco Fetal (FHR);
- Batimento Cardíaco Materno (MHR);
- Contrações Uterinas (UA);

Após a coleta destas informações, estas são enviadas para um agente obstetra.



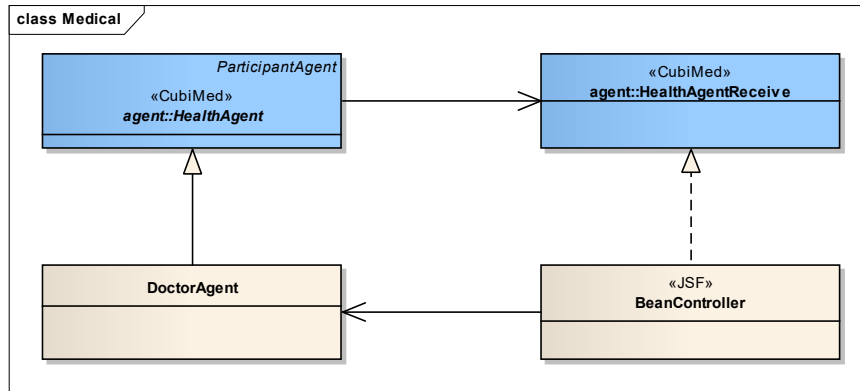
**Figura 20:** Diagrama de classes Aplicativo Gestante – Cenário Monitoramento Fetal

*Aplicativo Obstetra:* Foi instanciado em um computador como aplicação *desktop*, na qual se instancia um Agente Obstetra que é uma extensão do Agente de Saúde. Este agente possui uma base de conhecimento armazenada em arquivos XML contendo regras que permitam que se realize um diagnóstico preventivo para alertar o médico. Para entender melhor como este agente foi implementado, na seção 4.1.1.2.1, é explicado como foi estruturada a base de conhecimento utilizada para



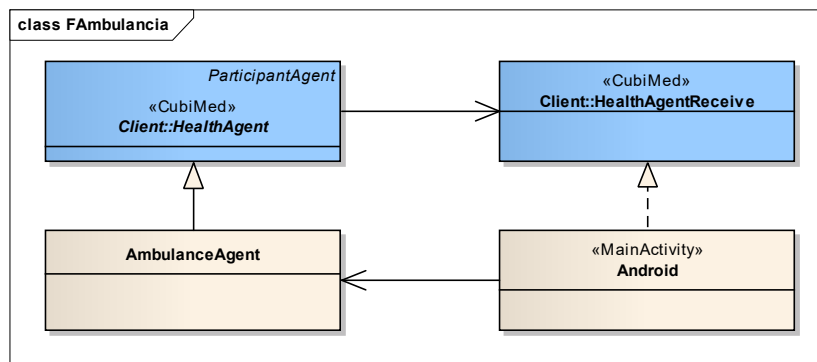
**Figura 21:** Diagrama de classes Aplicativo Obstetra – Cenário Monitoramento Fetal

*Aplicativo Médico:* Trata-se de uma aplicação web na qual é instanciado um Agente Médico, que é uma extensão do Agente de Saúde. Este agente espera continuamente por informações que possam vir do Agente Obstetra e, quando as recebe, informa ao médico, através da interface web, para que este possa realizar uma ação. Depois de analisar as informações, o médico titular tem a possibilidade de solicitar uma ambulância para atender o paciente.



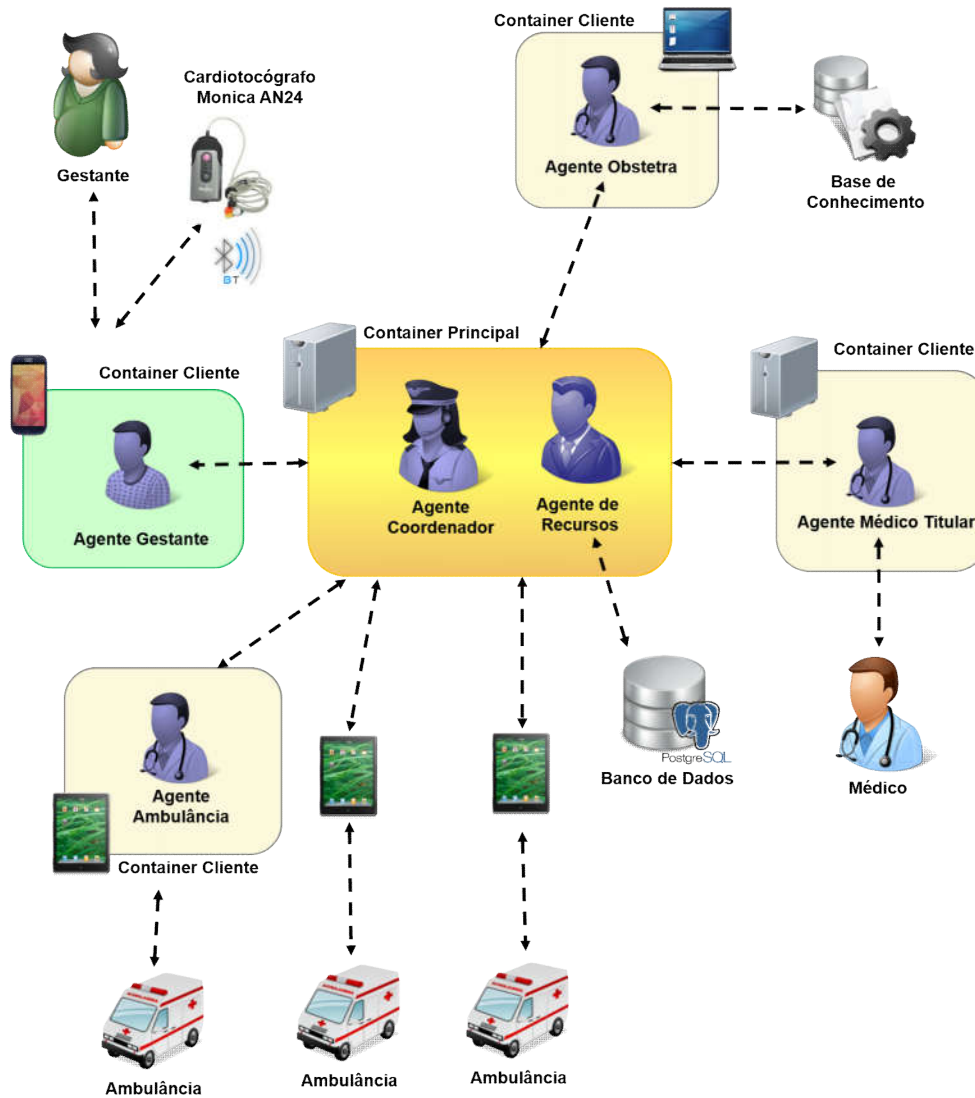
**Figura 22:** Diagrama de classes Aplicativo Médico – Cenário Monitoramento Fetal

*Aplicativo Ambulância:* Este aplicativo será usado por um enfermeiro ou paramédico que trabalhe prestando o serviço de ambulâncias. Além disso, será uma aplicação móvel onde estará instanciado um Agente Ambulância que é uma extensão do Agente de Saúde. Este agente receberá instruções por parte dos médicos e alertará o enfermeiro, para que possa prestar os serviços de saúde para a gestante o quanto antes.



**Figura 23:** Diagrama de classes Aplicativo Ambulância – Cenário Monitoramento Fetal

A seguir, é apresentada uma imagem ilustrando o esquema descrito.



**Figura 24:** Especificação de implementação do cenário de monitoramento fetal

#### 4.1.1.2.1. Base de Conhecimento

Para ter um melhor entendimento de como o Agente Obstetra funciona, explicamos como foi estruturada a base de conhecimento utilizada pelo agente para fornecer um pré-diagnóstico sobre a saúde do feto.

Os exames de histórico médico, juntamente com a observação de mudanças na frequência cardíaca fetal ou a atividade uterina da mãe, apresentam cinco tipos de sintomas de risco: taquicardia, bradicardia,

desaceleração precoce, desaceleração tardia e desaceleração variável. As características destes sintomas são descritas abaixo (MILLER & MILLER, 2012):

- *Taquicardia*: Taquicardia refere-se a uma frequência cardíaca fetal acima da média (acima de 160 bpm). Pode ocorrer por várias razões, mas geralmente não está diretamente relacionada à contração uterina.
- *Bradycardia*: Bradycardia refere-se a uma frequência cardíaca fetal muito lento (abaixo de 110 bpm). Isto pode ter um impacto significativo na saúde do feto e pode causar a morte, requerendo, portanto, mais atenção no acompanhamento.
- *Desaceleração precoce*: desaceleração precoce originando uma curva consistente nas contrações uterinas, com quase nenhum atraso.
- *Desaceleração tardia*: desaceleração tardia desenha uma curva regular, espelhando as contrações uterinas, que é o oposto de desaceleração precoce, apresentando uma curva de sino invertido. A defasagem consistentemente ocorre no início, pico e final de cada contração.
- *Desaceleração variável*: Como o nome sugere, a desaceleração variável pode assumir várias formas (U ou V), e não tem relação com as contrações uterinas. Esta condição é observada em 80% dos procedimentos de nascimento. Alguns casos de desaceleração irregular estão relacionados aos movimentos fetais, ao invés de contrações uterinas.

As características destes cinco sintomas são principalmente de natureza qualitativa, de modo que um diagnóstico preciso depende amplamente da experiência do obstetra. Nesse sentido, tentando reduzir o potencial de erro humano e pretendendo realizar um diagnóstico automático, apresenta-se um desenvolvimento de diretrizes quantitativas. As notações são definidas como segue e as regras de diagnóstico são apresentadas abaixo (SU et al., 2014):

- RHs: ciclo de mudança da frequência cardíaca hora de início.
- HRE: alteração de taxa tempo do fim do ciclo do coração.
- HRP: ciclo de mudança de frequência cardíaca.
- UCs: ciclo de contração uterina, hora de início.
- UCE: a contração uterina hora de fim de ciclo.
- UCP: a contração uterina pico do ciclo.

- HSP: Pico do ciclo de aceleração da frequência cardíaca (chamado de "região dos ombros" e aparecendo apenas em desaceleração Variável).
- HSe: Hora de fim do ciclo de aceleração da frequência cardíaca (chamado de "região dos ombros" e aparecendo apenas em desaceleração Variável).
- B: linha de base da frequência cardíaca.

**Tabela 3:** Regras para o diagnóstico de doenças no feto

Sintoma	Crítérios
Taquicardia	Taxa de batimento cardíaco $\geq 160$ bpm. A condição deve durar pelo menos três minutos.
Bradycardia	Taxa de batimento cardíaco $\leq 110$ bpm. A condição deve durar pelo menos três minutos.
Desaceleração precoce	HRp <UCe, a frequência cardíaca atinge seu ponto mais baixo antes do útero, ciclo de contração está completo. HRp - HRs $\geq 30$ , mais de 30 segundos de atraso entre o ponto em que frequência cardíaca começa a cair e o ponto mais baixo. RHs $\geq$ UCs + 5, mais de cinco segundos de atraso entre o momento em que a frequência cardíaca começa a cair e o início da contração uterina. B - HRP $\geq 15$ , ou seja, o ponto mais baixo de um ciclo de mudança de frequência cardíaca deve ser, pelo menos, 15 batimentos abaixo da linha de base.
Desaceleração tarde	O ponto mais baixo da frequência cardíaca fetal ocorreu antes da conclusão da contração uterina (violando a regra HRp $\geq$ UCe), então a desaceleração tardia pode ser descartada.
Desaceleração Variável Tipo A	Mais de 30 segs passaram-se entre o início da diminuição da taxa e o seu ponto mais baixo, violando assim a regra HRp - HRs $\leq 30$ , de modo que a desaceleração variável tipo A pode ser descartada.
Desaceleração Variável Tipo B	O intervalo entre o início da queda no ritmo cardíaco fetal e o início da contração uterina é mais de 10 pontos, o que significa que a contração uterina tinha começado mais de cinco segundos antes do momento em que a taxa da frequência cardíaca



	começou a cair. Este fato satisfaz os critérios da regra RHs $\geq$ UCs + 5. Em segundo lugar, a maior diferença entre a linha de base e o ponto de frequência cardíaca fetal é de aproximadamente 45 batidas, que também cumpre o requisito da regra B - HRP $\geq$ 15. No entanto, nenhuma região de ombro pode ser encontrada no ciclo de mudança de batida do coração, sendo possível descartar a desaceleração variável de tipo B.
--	---

Este procedimento permite fazer um diagnóstico inicial que pode ser aproveitado pelos médicos para tomar ações pertinentes no atendimento da gestante.

#### **4.1.2. Assistente de Administração de Medicamentos**

De acordo com um estudo de referência sobre erros médicos, realizado pelo Instituto de Medicina dos EUA em 1999, erros de interpretação na receita de medicamentos e reações adversas a medicamentos são os casos mais comuns entre todos os erros médicos (ZAO et al., 2010). Este cenário apresenta a implementação de uma aplicação que permitirá ao paciente dispor de um assistente de administração de medicamentos. Este assistente manterá contato com um farmacêutico e com o médico titular do paciente para garantir que os remédios que o paciente irá tomar sejam adequados.

A seguir, são apresentadas a ideia principal e a implementação do cenário.

##### **4.1.2.1. Ideia principal**

A ideia principal desta implementação é permitir a criação de um aplicativo que fornece um assistente de administração de medicamentos para o paciente. Este assistente terá as seguintes funcionalidades:

- Receber e salvar informações das prescrições médicas feitas pelo médico;
- Lembrar o paciente de que precisa tomar um remédio em determinado horário;
- Informar as doses exatas que o paciente precisa tomar;
- Receber as novas prescrições do médico e notificá-las ao paciente;

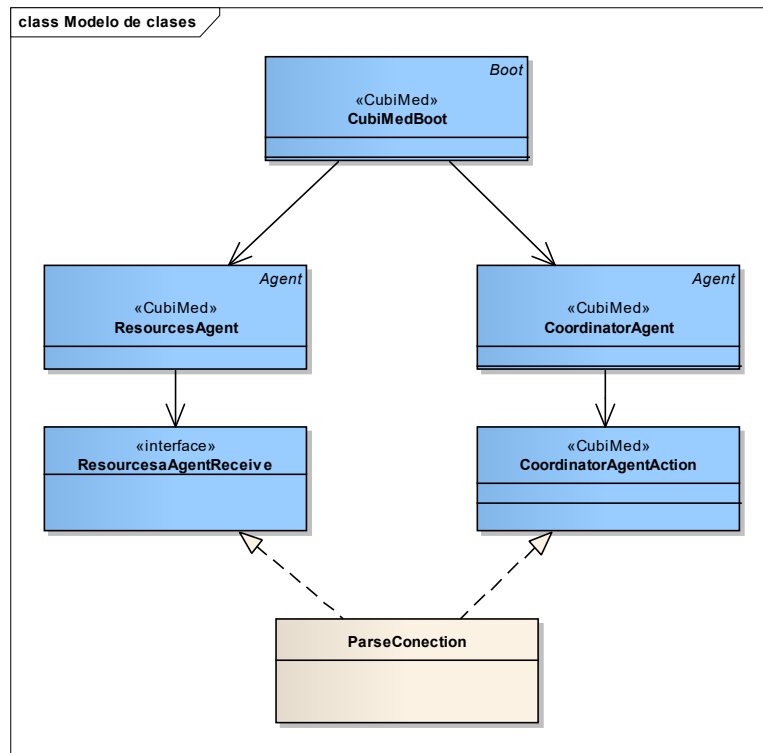
- Obter dados de um medicamento por meio da tecnologia NFC;
- Consultar o médico sobre um determinado remédio;

Para conseguir executar as funcionalidades listadas, o assistente poderá comunicar-se com um farmacêutico, que terá conhecimento suficiente para responder as consultas do paciente. Caso o farmacêutico detecte algum problema, irá alertar o médico, para que este possa tomar as ações necessárias.

#### **4.1.2.2. Implementação**

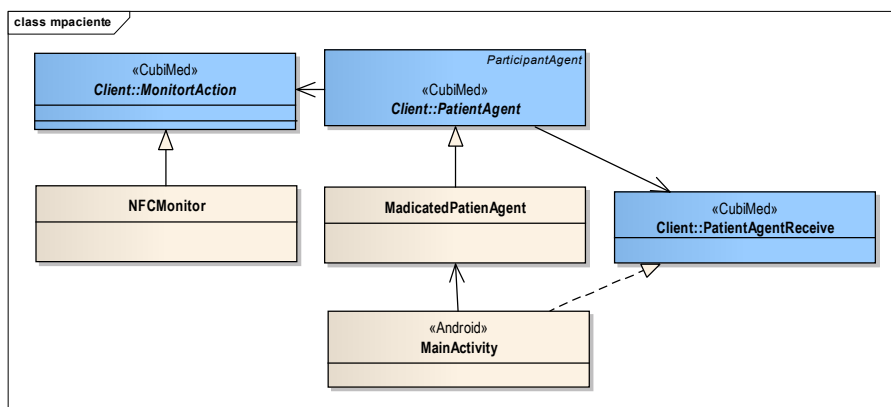
A implementação deste cenário de uso é composta pelo servidor principal e por três aplicativos colaborativos que são descritos a seguir.

*Servidor Principal:* No servidor principal, são instanciados o Agente Coordenador e o Agente de Recursos. O Agente Coordenador será responsável por informar aos participantes sobre quem ingressou e quem saiu da aplicação. Além disso, fará uso dos métodos de autenticação que irão permitir que haja uma conexão com o Parse (PARSE, 2015), que irá servir como banco de dados na nuvem, que permitirá o acesso do usuário à aplicação. O agente de recursos terá acesso ao Parse para obter informações sobre os perfis dos usuários e assim colaborar com o farmacêutico para que possa analisar a informação sobre os remédios que o paciente precisa tomar.



**Figura 25:** Diagrama de classes Servidor Principal – Cenário Administração de Medicamentos

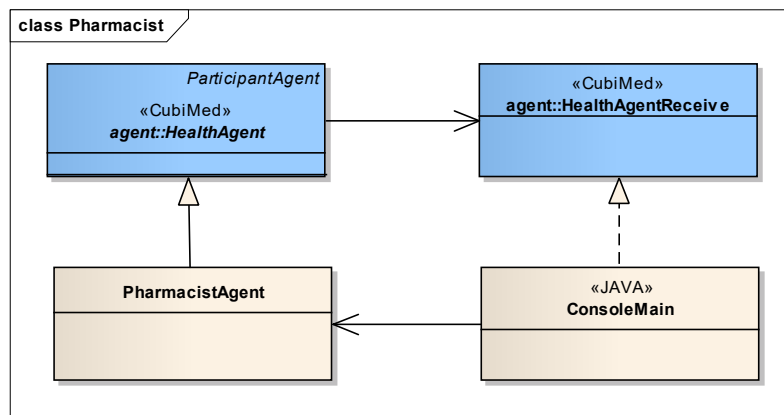
*Aplicativo Paciente:* O aplicativo do paciente é criado em um dispositivo móvel e contém uma instância do tipo Agente Paciente, o qual tem a tarefa de coletar informações sobre os medicamentos que o paciente vai tomar e enviá-la ao Agente Farmacêutico. Esta informação pode ser capturada por meio da interface do aplicativo ou por meio de leitura NFC.



**Figura 26:** Diagrama de classes Aplicativo Paciente – Cenário Administração de Medicamentos

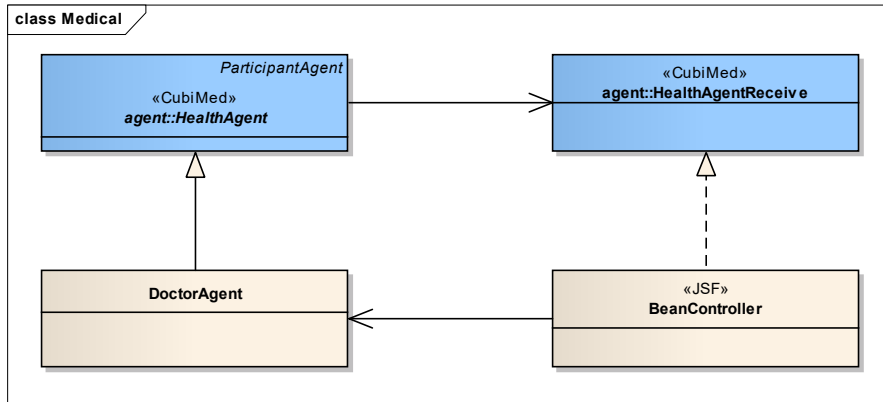
*Aplicativo Farmacêutico:* O aplicativo do farmacêutico é criado como uma aplicação desktop. Esta aplicação instancia um Agente Farmacêutico, que é uma extensão do Agente de Saúde. O agente farmacêutico terá a possibilidade de dialogar com agente de recursos para obter informações do perfil do paciente. Com esse conhecimento, o agente farmacêutico pode receber do paciente os seguintes tipos de solicitações:

- Lista de remédios que pode comprar, caso não consiga encontrar um determinado remédio.
- Questionamento sobre a possibilidade de tomar um determinado remédio.

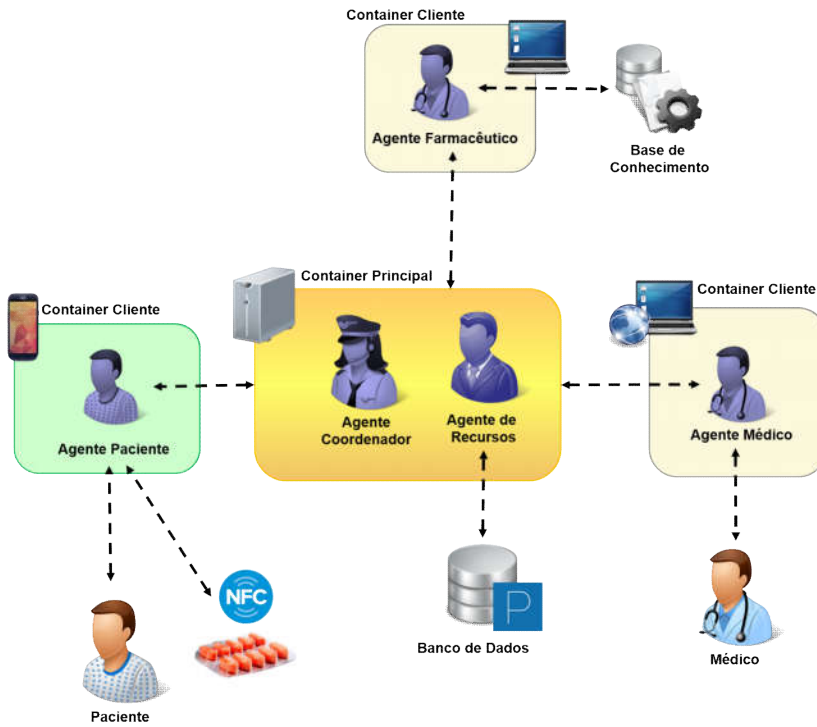


**Figura 27:** Diagrama de classes Aplicativo Farmacêutico – Cenário Administração de Medicamentos

*Aplicativo Médico:* O aplicativo médico consiste em uma aplicação web que instancia um Agente Médico que estende um Agente de Saúde. Este agente recebe os alertas do Agente Farmacêutico e os comunica ao médico, para que ele possa tomar uma ação. O aplicativo contém uma interface que permite ao médico preencher e enviar instruções para o paciente.



**Figura 28:** Diagrama de classes Aplicativo Médico – Cenário Administração de Medicamentos



**Figura 29:** Especificação de implementação do cenário de Assistente de

## 4.2. Avaliação

Nesta seção, é apresentada uma avaliação do framework desenvolvido neste trabalho. Já que esta proposta apresenta uma abordagem orientada a sistemas multi-agentes para a criação de aplicações colaborativas voltadas para u-Healthcare, e o framework está construído sobre a plataforma JADE, esta

avaliação pretende verificar se o uso deste framework é recomendável para a criação deste tipo de aplicações, e se agiliza o trabalho do desenvolvedor, em comparação com o uso direto do JADE.

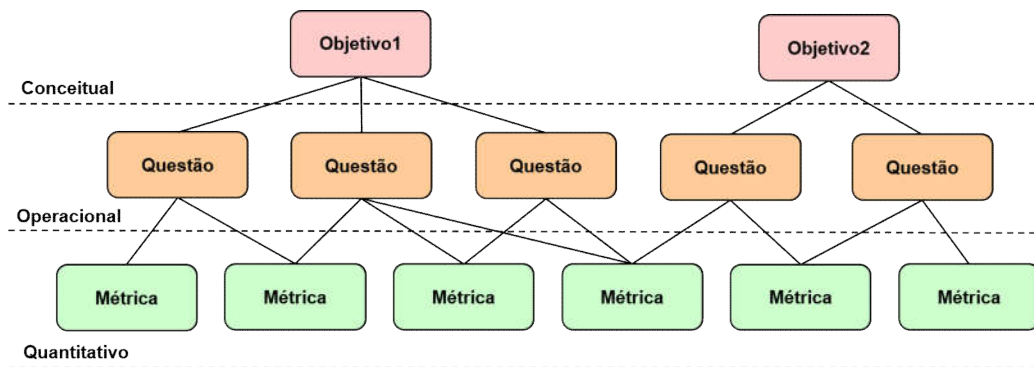
A seguir, será apresentado brevemente o modelo GQM, para então explicar de forma detalhada como foi realizada a avaliação e quais resultados foram obtidos.

#### 4.2.1. Modelo GQM

O método GQM foi originalmente proposto por Basili (BASILI et al., 2002), com o propósito de avaliar os defeitos de um conjunto de projetos da NASA. Posteriormente, o uso do GQM foi expandido e tem sido adotado para medir e melhorar a qualidade do software.

O modelo de medida proposto contém três níveis (Figura 30):

- Conceitual - Objetivos;
- Operacional - Questões
- Quantitativo - Métricas.



**Figura 30:** Estrutura de hierarquia do modelo GQM

Para aplicar o modelo GQM como método de avaliação para um software, deve-se seguir os seguintes passos:

- Definir um conjunto de metas que se pretende atingir quanto ao processo e aos produtos com as atividades de medição. As metas estão diretamente relacionadas a questões de relevância fundamental para o

interessado na avaliação, como a satisfação dos usuários, a qualidade dos produtos e a produtividade.

- Gerar uma série de perguntas que traduzam essas metas em aspectos quantitativos, e que possam ser alvos de medição. A ideia é gerar perguntas que, se respondidas, ajudariam a atingir as metas estabelecidas.
- Especificar as métricas que precisariam ser coletadas de forma a obter as informações necessárias para responder às perguntas geradas.
- Depois de ter seguido estes passos, a avaliação pode ser aplicada ao grupo de usuários ao qual está dirigido o software.

#### 4.2.2. Execução da Avaliação

Segundo (MARKIEWICZ & LUCENA, 2001) os três principais estágios no desenvolvimento de um framework são a análise de domínio, design da estrutura e instanciação do framework. Nesse sentido, para realizar a avaliação do framework usando o modelo GQM foram propostos três objetivos relacionados aos principais estágios de desenvolvimento de um framework.

Após o estabelecimento dos objetivos, são definidas questões em torno as quais são criadas as métricas de avaliação do framework. A seguir, são apresentados cada um dos objetivos com a questão correspondente e as métricas que foram estabelecidas para cada uma delas.

Para avaliar a análise de domínio, é estabelecido o primeiro objetivo, o qual é apresentado na tabela 4.

**Tabela 4:** Objetivo para a avaliação de análise de domínio

Objetivo 1	Avaliar a generalidade do framework
Analisar a:	Framework CubiMed
Com o propósito de	Verificar se o framework é genérico suficiente para ser usado em várias aplicações
Com respeito a	Criação de aplicações colaborativas na área da saúde que permitam atender ao paciente de forma remota
Do ponto de vista	Desenvolvedor de aplicações
No contexto de	Desenvolvimento de aplicações colaborativas voltadas para u-healthcare

Após ter sido estabelecido o objetivo, são propostas a questão e as métricas que permitirão cumprir este ponto.

**Tabela 5:** Questões e métricas para a avaliação de análise de domínio

Questão	Qual o grau de generalidade que o framework apresenta, para criar aplicações colaborativas no domínio de u-Health?
Métrica 1	Qual o grau de cobertura dos pontos fixos do framework?
Métrica 2	Qual o grau de cobertura dos pontos flexíveis do framework?

Finalmente, são apresentados os valores que serão usados para as métricas estabelecidas.

**Tabela 6:** Valores usados para as métricas da avaliação de análise de domínio

Etiqueta	Valor
Muito baixo	1
Baixo	2
Meio	3
Alto	4
Muito Alto	5

Como segundo ponto de avaliação, apresenta-se o design da estrutura para o qual é estabelecido o seguinte objetivo:

**Tabela 7:** Objetivo para a avaliação do design da estrutura

Objetivo 2	Avaliar a simplicidade de instanciação do framework, comparado com o JADE
Analisar o:	Framework CubiMed
Com o propósito de	Verificar se a arquitetura do framework fornece características simples e fáceis de usar
Com respeito a	Facilidade para usar as classes e interfaces do framework com relação ao JADE
Do ponto de vista	Desenvolvedor de aplicações
No contexto de	Desenvolvimento de aplicações colaborativas voltadas para u-healthcare

A seguir, são apresentadas a questão e métricas deste objetivo.

**Tabela 8:** Questões e métricas para a avaliação da instanciação do framework

Questão	Qual o grau de dificuldade para usar o framework, comparado com o JADE?
Métrica 1	Qual o grau de dificuldade para instanciar um agente com o framework?
Métrica 2	Qual o grau de dificuldade para estabelecer uma comunicação entre os diferentes agentes do framework?



Métrica 3	Qual o grau de dificuldade para estabelecer comunicação entre o agente e o usuário final da aplicação a traves de uma interface de usuário?
-----------	---

Finalmente são listados também os valores das métricas pertencentes ao objetivo 2.

**Tabela 9:** Valores usados para as métricas da avaliação do design da estrutura

Métricas	Valor
Muito difícil	1
Difícil	2
Dificuldade media	3
Fácil	4
Muito fácil	5

Finalmente, é estabelecido o ultimo objetivo que corresponde a avaliação da facilidade de instanciação do framework para diferentes dispositivos.

**Tabela 10:** Objetivo para a avaliação da instanciação do framework

Objetivo 3	Avaliar portabilidade do framework
Analisar o	Framework CubiMed
Com o propósito de	Verificar se o framework pode ser considerado portátil
Com respeito a	A portabilidade que apresenta para ser instanciado em diferentes dispositivos
Do ponto de vista	Desenvolvedor de aplicações
No contexto de	Desenvolvimento de aplicações colaborativas voltadas para u-healthcare

A questão e as métricas são estabelecidas.

**Tabela 11:** Objetivo para a avaliação do design da estrutura

Questão	Qual o grau de portabilidade que o framework apresenta para ser instanciado em dispositivos diferentes?
Métrica 1	Qual o grau de dificuldade para instanciar o framework em um dispositivo diferente?
Métrica 2	Qual o grau de dificuldade para configurar o framework em uma plataforma?

Da mesma forma que nos pontos anteriores, é apresentada também a escala de valores para as métricas deste objetivo.

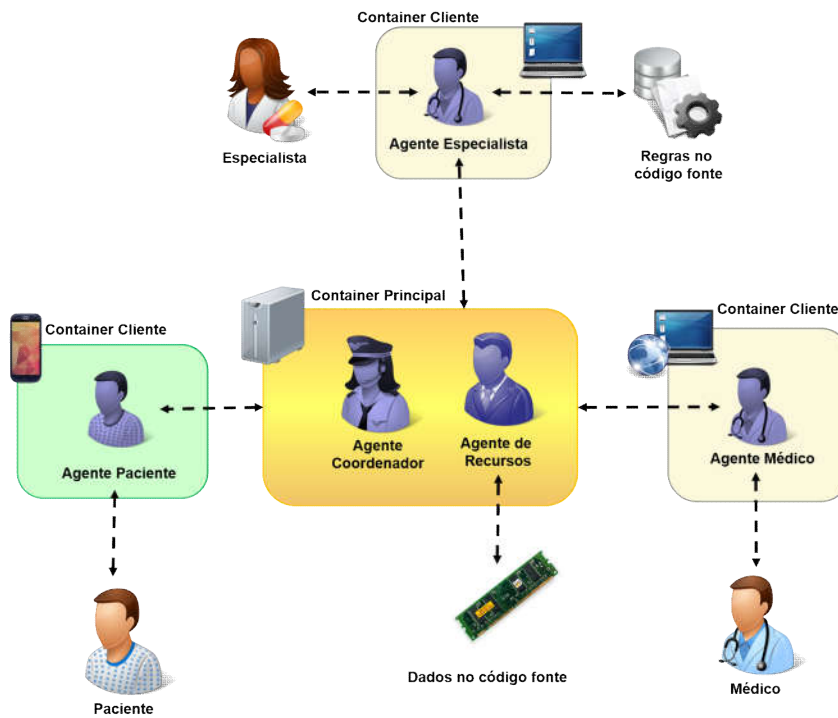
**Tabela 12:** Valores usados para as métricas da avaliação da instanciação do framework

Métricas	Valor
Muito difícil	1
Difícil	2
Dificuldade media	3
Fácil	4
Muito fácil	5

Depois que foram estabelecidos os objetivos de avaliação, foi realizada a seguinte atividade.

Foram entrevistadas 6 pessoas do Laboratório de Engenharia de Software da PUC-RIO. As pessoas entrevistadas tinham conhecimentos sobre o paradigma de sistemas multi-agentes e sobre a plataforma JADE. A entrevista com cada pessoa durou aproximadamente três horas. Dentro deste período, de tempo foi realizado o seguinte procedimento.

1. Foi explicada a motivação deste trabalho e o objetivo para o qual foi desenvolvido este trabalho.
2. De forma breve, foi apresentada a estrutura geral do JADE, para que as pessoas entrevistadas possam lembrar o uso desta tecnologia.
3. Foi apresentado o framework CubiMed e explicada a arquitetura do framework.
4. Para que as pessoas entrevistadas pudessem avaliar o framework, foi feita uma pequena capacitação, onde foi apresentada uma pequena aplicação que tinha a seguinte estrutura.



**Figura 31:** Especificação de implementação de um cenário de interação paciente – equipe médica

- Foi criado um servidor que apresentava os agentes Coordenador e de Recursos e que não possuía conexão com um banco de dados. O agente recursos devolvia apenas dados fixados no código.
- Foi criada uma aplicação móvel para o paciente. A aplicação instanciava um agente paciente e possuía uma interface que permitia enviar um dado que representava a temperatura.
- Foi criada uma aplicação desktop que representava o especialista. Foi instanciado um Agente de Saúde, com o papel de especialista. Este agente recebia a informação de temperatura do paciente e, se a temperatura fosse maior que 37, era enviada uma mensagem para o médico indicando que o paciente estava apresentando problemas de saúde. Se a temperatura fosse menor que 37, enviava uma mensagem para o paciente indicando que estava tudo bem.
- Finalmente foi criada uma aplicação web para o médico. A aplicação foi executada no servidor glassfish e tinha um Agente de Saude instanciado que representava o médico. A interface web permitia observar quando chegava uma mensagem do especialista.

Após a apresentação da aplicação e a explicação de como foi implementado o seu código, cada entrevistado respondeu às perguntas do modelo GQM, fornecendo algumas observações. Os resultados são apresentados na seção seguinte.

### 4.2.3. Resultados da Avaliação

**Tabela 13:** Resultados da avaliação de análise de domínio.

Objetivo 1: Avaliar a generalidade do framework							
Qual o grau de generalidade que o framework apresenta, considerando a criação de aplicações colaborativas no domínio de u-Health?	P1	P2	P3	P4	P5	P6	M
Qual o grau de cobertura dos pontos fixos do framework?	4	4	4	3	3	3	3,5
Qual o grau de cobertura dos pontos flexíveis do framework?	5	3	5	4	3	4	4

O resultado da avaliação sobre o análise do domínio indica que o framework tem um bom grau de generalidade. Foi possível identificar devido ao fato de que, tanto os pontos fixos como os flexíveis, têm uma boa cobertura dos mecanismos necessários para a criação de uma aplicação colaborativa.

*“Toda a estrutura de comunicação já está criada e isso é muito bom. Porém, senti falta de uma estrutura já desenvolvida que dê suporte à interpretação de dados oriundos de diferentes mecanismos (sensores, arquivo pdf) ”*

Porém, conforme mostra o comentário acima, os entrevistados sentiram falta de ter mais métodos implementados que permitam pegar dados de fontes externas de uma forma padrão.

**Tabela 14:** Resultados avaliação do design da estrutura

<b>Objetivo 2: Avaliar a simplicidade de instanciação do framework comparado com JADE</b>							
<b>Qual o grau de dificuldade para usar o framework comparado com o JADE?</b>	P1	P2	P3	P4	P5	P6	M
Qual o grau de dificuldade para instanciar um agente com o framework?	5	4	5	4	5	5	4,6
Qual o grau de dificuldade para poder estabelecer comunicação entre os distintos agentes do framework?	5	3	4	4	4	4	4
Qual o grau de dificuldade para estabelecer comunicação entre o agente e o usuário final da aplicação a traves de uma interface de usuário?	5	3	4	3	5	4	4

Observando-se os resultados que a avaliação do design da estrutura apontou, percebe-se que a instanciação do framework Cubimed é considerada mais simples que a instanciação do JADE nativo, significando que o grau de dificuldade para usar o framework é baixo.

*“Com a opção de herança, ficou bem intuitivo a criação de novos agentes, e a comunicação e feita de forma trivial”*

Conforme o comentário reflete, a implementação das características do framework é considerada simples e fácil de usar.

**Tabela 15:** Resultados da avaliação da instanciação do framework

<b>Objetivo 3: Avaliar a portabilidade do framework</b>							
<b>Qual o grau de portabilidade que o framework apresenta para ser instanciado em dispositivos diferentes?</b>	P1	P2	P3	P4	P5	P6	M
Qual o grau de dificuldade para instanciar o framework em um dispositivo diferente?	4	3	5	4	3	4	3,8
Qual o grau de dificuldade para configurar o framework em uma plataforma?	4	4	4	3	3	4	3,6

Considerando a instanciação do framework, é possível ver que o grau de portabilidade é alto, já que o framework consegue ser instanciado em plataformas web, desktop e móvel.

*“Foi demonstrado claramente que podemos instanciar em várias plataformas, porém na plataforma Android achei um pouco complexa a implementação dos agentes”*

Conforme demonstra o comentário, existe uma certa dificuldade da instanciação nos dispositivos móveis, o que ocorre devido às restrições que o JADE-LEAP apresenta, sendo um pouco complexo entender o motivo pelo qual precisa-se usar interfaces extras para executar os métodos dos agentes.

Como conclusão, pode-se perceber que o framework proposto é bem avaliado, de modo que poderia ser utilizado pelos desenvolvedores para agilizar o trabalho e implementar aplicações colaborativas que precisem ser executadas em diferentes plataformas.