



INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA (ISEL)

DEPARTAMENTO DE ENGENHARIA ELETRÓNICA E DE  
TELECOMUNICAÇÕES E COMPUTADORES (DEETC)

---

MESTRADO EM ENGENHARIA DE INFORMÁTICA E MULTIMÉDIA (MEIM)

UNIDADE CURRICULAR DE ENGENHARIA DE SOFTWARE

---

# Relatório Final

Docente

---

Professor	Jorge Branco
-----------	--------------

---

Ana Sofia Oliveira (A39275)

Janeiro, 2024



# Índice

Introdução .....	5
Enquadramento teórico .....	6
Análise de requisitos .....	8
Casos de Utilização .....	8
Requisitos .....	15
Arquitetura .....	17
Diagramas de interação .....	17
Modelo de domínio .....	17
Implementação .....	20
Código Fonte .....	20
Testes de desenvolvimento .....	21
Modelo de implantação .....	21
Análise crítica .....	22
Conclusão .....	23
Anexos .....	24

# Índice de Tabelas

Tabela 1 - Requisitos funcionais .....	15
--	----

## Índice de figuras

Figura 1 - Diagrama macro de casos de utilização .....	8
Figura 2 - Casos de utilização da gestão de consultas .....	9
Figura 3 - Casos de utilização da gestão de receitas .....	11
Figura 4 - Casos de utilização da gestão de exames .....	13
Figura 5 - Modelo de domínio .....	17
Figura 6 - Mecanismo de gestão de consultas .....	18
Figura 7 - Mecanismo de gestão de receitas .....	18
Figura 8 - Mecanismo de gestão de exames .....	19
Figura 9 - Modelo de implantação .....	21
Figura 10 - Diagrama de interação adicionarConsulta .....	24
Figura 11 - Diagrama de interação confirmarConsulta .....	25
Figura 12 - Diagrama de interação cancelarConsulta .....	26
Figura 13 - Diagrama de interação reagendarConsulta .....	27
Figura 14 - Diagrama de interação submeterReceitas .....	28
Figura 15 - Diagrama de interação consultarReceitas .....	29
Figura 16 - Diagrama de interação exportarReceitas .....	30
Figura 17 - Diagrama de interação submeterExames .....	31
Figura 18 - Diagrama de interação consultarExames .....	32
Figura 19 - Diagrama de interação submeterResultadoExame .....	33
Figura 20 - Diagrama de interação exportarExames .....	34

# Introdução

A clínica HealthCare, com atividade no setor médico, disponibiliza aos seus pacientes serviços de acompanhamento médico nas mais diversas áreas, como medicina cardiovascular, medicina neurológica e psiquiatria.

Quando um paciente realiza uma consulta é habitual fazer uma breve descrição do seu histórico médico, caso não esteja a ser acompanhado na especialidade da atual consulta. No entanto, a informação fornecida pelos pacientes nem sempre é a mais assertiva por diversos motivos – falta de conhecimento médico, dificuldade em indicar períodos exatos em que determinados atos médicos ocorreram, entre outros. Isto provoca inconsistências e atrasos nos diagnósticos médicos. É sobre esta problemática que a clínica sugeriu a criação de uma aplicação para dispositivos móveis com integração com o seu sistema médico, que representasse os atos médicos no percurso do paciente na clínica.

A apresentação dos exames previamente pedidos pelo médico em formato físico constitui um desafio para o corpo médico por levantar algumas questões ao nível do acesso, nomeadamente ausência da apresentação dos exames por esquecimento do paciente, tempo de análise mais reduzido pois o corpo médico apenas tem acesso aos exames no momento da consulta, podendo não ter tempo suficiente para os analisar com a devida atenção.

Assim, para além do propósito anteriormente indicado, esta aplicação atua igualmente no sentido de minimizar este impacto, com o carregamento dos resultados exames médicos automaticamente (caso tenham sido realizados dentro da clínica) ou manualmente (caso tenham sido realizados noutra clínica). Desta forma, o médico poderá consultar o resultado dos exames pedidos atempadamente antes da realização da consulta ou, caso veja necessário, pedir que a consulta seja remarcada por serem necessários exames adicionais.

A ambição da clínica, após a criação da aplicação proposta, é a de escalar esta metodologia para outras entidades de saúde como, por exemplo, o serviço nacional de saúde centralizando o histórico médico dos pacientes entre diferentes entidades de saúde à escala nacional.

No decorrer deste relatório iremos abordar as várias fases de desenvolvimento do projeto durante a primeira iteração do modelo de desenvolvimento incremental e iterativo.

# Enquadramento teórico

O desenvolvimento do projeto HealthCenter está envolvido no âmbito da unidade curricular de engenharia de software.

A engenharia de software define a aplicação de princípios de engenharia para o desenvolvimento, manutenção e testes de produtos de software, com o objetivo de garantir a satisfação dos requisitos definidos e a qualidade de experiência dos utilizadores do produto desenvolvido.

A aplicação da engenharia de software no processo de desenvolvimento de um produto promove um desenvolvimento sistemático, organizado e previsível, minimizando a possibilidade de erros no processo e, consequentemente, o esforço de manutenção do produto. Para isto, existem diversas metodologias, técnicas e ferramentas que podem ser aplicadas para o desenvolvimento de cada produto e que visam a criação de um sistema informático de qualidade dentro dos limites orçamentais e temporais definidos.

O desenvolvimento sistemático é alcançado através de duas principais componentes de trabalho – conhecer e concretizar. Para o conhecimento devem considerar-se dois domínios, domínio do problema – âmbito do problema a resolver, envolvendo os utilizadores e outras partes envolvidas - e o domínio da solução – âmbito de conhecimento técnico, essencial à conceção e produção da solução. A concretização é conseguida através da conceção e construção da solução com base nos conhecimentos obtidos no âmbito do domínio do problema e da solução.

Os processos de desenvolvimento definem atividades a realizar, artefactos a criar e as formas de organização das equipas e do trabalho. As várias etapas dos processos são organizadas temporalmente, variando de acordo com a metodologia utilizada, mas usualmente incluem as seguintes atividades:

1. Análise: identificação e definição das necessidades dos utilizadores e especificação dos requisitos do software a produzir.
2. Conceção: especificação das partes que compõem o software e da interação entre cada uma das partes, tendo como resultado a arquitetura do software.
3. Implementação: concretização da arquitetura definida sob a forma de código fonte.
4. Verificação: validação e teste do software, de forma a garantir os requisitos previamente definidos.
5. Implantação: instalação, configuração e disponibilização do software.
6. Manutenção: manutenção do software em operação e a sua eventual evolução.

Existem diversos modelos de processos de desenvolvimento, entre os quais o modelo de desenvolvimento em cascata, incremental, ágil. Cada modelo de processos de desenvolvimento é

organizado de diferentes formas – linear, cíclico, incremental -, e concretiza os tipos de atividade de acordo com princípios e formas de organização do próprio modelo.

No decorrer do projeto foi utilizado um modelo de desenvolvimento incremental e iterativo, baseado no modelo cíclico, onde as atividades anteriores são realizadas em diferentes ciclos (iterações), com o aumento sucessivo do conhecimento obtido e da funcionalidade entre versões produzidas. A conclusão do projeto é conseguida quando os conhecimentos e funcionalidades convergem na versão final.

O modelo cíclico permite que os utilizadores interajam no processo de desenvolvimento de forma precoce através do contacto com versões anteriormente desenvolvidas. Desta forma, é possível recolher o feedback dos utilizadores e considerar as suas necessidades no desenvolvimento de próximas versões. Este modelo permite manter foco de áreas de especialização consoante a versão em desenvolvimento.

De modo geral, o modelo está organizado entre fases, atividades e iterações.

- Fase: diferentes etapas de desenvolvimento. Cada fase tem um âmbito de atividade específico: análise, conceção, construção ou verificação.
- Atividade: Conjunto de ações com propósito específico no âmbito da solução: análise de requisitos, conceção da arquitetura, construção do código, verificação da solução.
- Iteração: Conjunto de atividades com o propósito de produzir uma versão incremental da solução.

O modelo incremental e iterativo consiste no modelo cíclico orientado para a produção de versões incrementais da solução, com a repetição das mesmas etapas, organizadas por iterações, até que o resultado pretendido seja obtido. Este modelo possibilita a obtenção de conhecimento de forma incremental ao longo da realização de um projeto, reduzindo progressivamente a incerteza e, conseqüentemente, o risco, até à concretização da versão final da solução. Este modelo conta ainda com as propriedades do modelo adaptativo como a aceitação do imprevisto e da mudança, permitindo a adaptação à mudança.

Apesar do desenvolvimento incremental e iterativo apresentar características relevantes, para lidar com aspetos como a incerteza e a redução do tempo de desenvolvimento, também pode apresentar limitações, nomeadamente, quando os sistemas disponibilizam funcionalidades críticas para os seus utilizadores na produção dos quais o foco não é tanto a rapidez, mas o controlo do desenvolvimento.

# Análise de requisitos

O projeto proposto consiste numa aplicação informática que permite gerir a marcação de consultas e dos atos médicos de um paciente. A autenticação na aplicação deverá ser a mesma que a usada pelos restantes sistemas da clínica HealthCare, pelo que não será explorada neste documento.

Durante capítulo iremos documentar a análise de requisitos suportada em duas componentes: identificação dos casos de utilização e levantamento dos requisitos.

## Casos de Utilização

No modelo proposto, existem três atores: o paciente, o médico e o administrativo. Numa visão macro, o paciente deve conseguir gerir consultas, receitas e exames, o médico deve conseguir gerir receitas e exames, e o administrativo deve conseguir gerir consultas. A gestão de exames está dependente do sistema de gestão de exames existente na clínica HealthCare.

No diagrama de casos de utilização na figura 1 apresentamos uma visão macro das tarefas que cada utilizador deverá conseguir realizar no sistema.

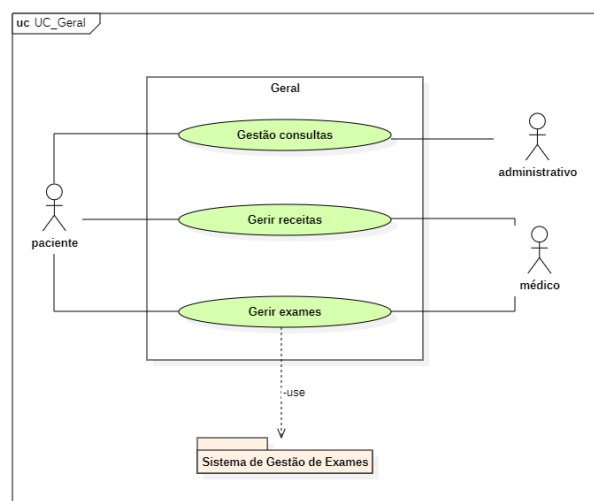


Figura 1 - Diagrama macro de casos de utilização



## Gestão de consultas

A gestão das consultas é composta por quatro tarefas: agendamento, confirmação do agendamento, cancelamento ou reagendamento da consulta. Na figura 2 apresentamos o diagrama de casos de utilização com foco na gestão de consultas.

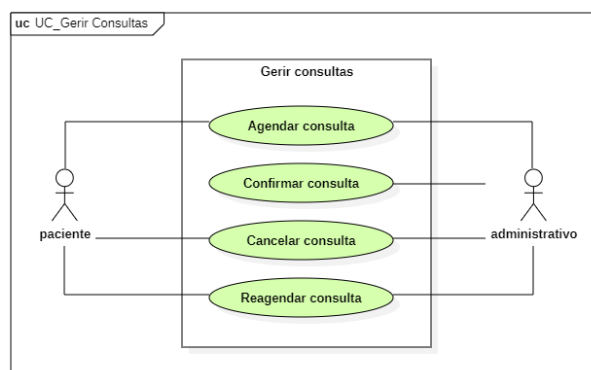


Figura 2 - Casos de utilização da gestão de consultas

Abaixo apresentamos o detalhe de cada caso de utilização identificado, bem como a sua relação com os requisitos identificados mais à frente.

### Agendar consulta

**Resumo:** Este caso de utilização permite ao paciente ou administrativo agendar uma consulta.

**Atores:** Paciente ou administrativo

**Cenário principal:**

1. Este cenário inicia-se quando o paciente ou o administrativo quer agendar uma consulta.
2. O utilizador selecciona uma especialidade.
3. O utilizador selecciona um dia e hora com, pelo menos, 72 horas de antecedência.
4. O utilizador observa a confirmação do agendamento da consulta e o caso de utilização termina.

**Requisitos:** ver requisitos R1, R2, R20.1 e R21.1.

### Confirmar consulta

<p><b>Resumo:</b> Este caso de utilização permite que o administrativo confirme o agendamento de uma consulta em sistema.</p> <p><b>Atores:</b> Administrativo</p> <p><b>Pré-condições:</b></p> <ul style="list-style-type: none"><li>• A confirmação de consulta depende da existência de agendamentos de consultas que não tenham sido confirmadas ou canceladas.</li><li>• As consultas pendentes de confirmação devem ser confirmadas até 5h antes da hora do agendamento da consulta.</li></ul>
<p><b>Cenário principal:</b></p> <ol style="list-style-type: none"><li>1. Este cenário inicia-se quando o administrativo quer confirmar as consultas agendadas.</li><li>2. O administrativo seleciona uma consulta.</li><li>3. O administrativo confirma a consulta e o caso de utilização termina.</li></ol>
<p><b>Requisitos:</b> ver requisitos R3, R4 e R21.2.</p>

### Cancelar consulta

<p><b>Resumo:</b> Este caso de utilização permite que o paciente ou o administrativo cancelem uma consulta agendada.</p> <p><b>Atores:</b> Paciente ou administrativo</p> <p><b>Pré-condições:</b></p> <ul style="list-style-type: none"><li>• A consulta apenas pode ser cancelada com, pelo menos, 5 horas de antecedência.</li></ul>
<p><b>Cenário principal 1:</b></p> <ol style="list-style-type: none"><li>1. Este cenário inicia-se quando o paciente quer cancelar uma consulta previamente agendada.</li><li>2. O paciente seleciona a consulta.</li><li>3. O paciente cancela a consulta e o caso de utilização termina.</li></ol> <p><b>Cenário principal 2:</b></p> <ol style="list-style-type: none"><li>1. Este cenário inicia-se quando o administrativo quer cancelar uma consulta previamente agendada.</li><li>2. O administrativo seleciona a consulta.</li><li>3. O administrativo identifica o motivo do cancelamento.</li><li>4. O administrativo cancela a consulta e o caso de utilização termina.</li></ol> <p><b>Cenário alternativo 1:</b></p> <ol style="list-style-type: none"><li>1. Este cenário inicia-se quando a consulta não foi confirmada ou cancelada até 5h antes da realização da mesma.</li><li>2. O sistema cancela a consulta e o caso de utilização termina.</li></ol>
<p><b>Requisitos:</b> ver requisitos R5, R6, R20.3 e R21.2.</p>

## Reagendar consulta

<b>Resumo:</b> Este caso de utilização permite que o paciente ou o administrativo reagendem uma consulta.
<b>Atores:</b> Paciente ou administrativo
<b>Pré-condições:</b> <ul style="list-style-type: none"><li>• A consulta apenas pode ser reagendada com, pelo menos, 5 horas de antecedência.</li></ul>
<b>Cenário principal:</b> <ol style="list-style-type: none"><li>1. O caso de utilização inicia-se quando o paciente ou o administrativo pretende reagendar uma consulta previamente marcada.</li><li>2. O utilizador seleciona a consulta.</li><li>3. O utilizador seleciona a opção para remarcar consulta.</li><li>4. Aplica-se o cenário de agendar consulta, a partir do ponto 3.</li></ol>
<b>Requisitos:</b> ver requisitos R7, R20.3 e R21.2.

## Gestão de receitas

A gestão das receitas é composta por três tarefas: submissão, consulta e exportação. Na figura 3 apresentamos o diagrama de casos de utilização com foco na gestão de receitas.

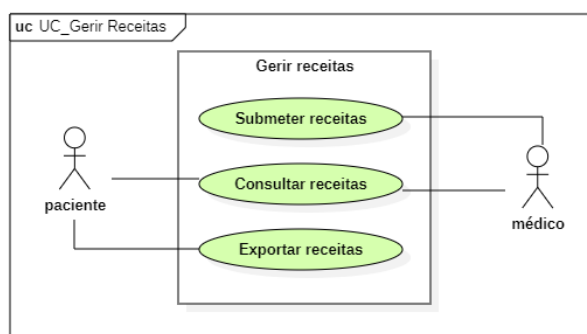


Figura 3 - Casos de utilização da gestão de receitas

Abaixo apresentamos o detalhe de cada caso de utilização identificado, bem como a sua relação com os requisitos.

### Submeter receitas

<b>Resumo:</b> Este caso de utilização permite que o médico submeta receitas
<b>Atores:</b> Médico
<b>Pré-condições:</b> <ul style="list-style-type: none"><li>• O médico precisa de ter os documentos das receitas para submeter nos formatos JPEG, PNG ou PDF.</li></ul>
<b>Cenário principal:</b> <ol style="list-style-type: none"><li>1. O caso de utilização inicia-se quando o médico emite uma receita ao paciente.</li><li>2. O médico submete a receita no sistema e o caso de utilização termina.</li></ol>

**Requisitos:** ver requisitos R8, R9 e R22.1.

### *Consultar receitas*

**Resumo:** Este caso de utilização permite que o paciente e o médico consultem receitas emitidas.

**Atores:** Paciente e médico

**Pré-condições:**

- Existência de receita emitida no contexto de uma consulta.

**Cenário principal:**

1. O caso de utilização inicia-se quando o paciente ou o médico pretendem consultar uma receita emitida.
2. O utilizador selecciona uma consulta.
3. O utilizador selecciona uma receita emitida no contexto da consulta e o caso de utilização termina.

**Requisitos:** ver requisitos R10, R20.2, R20.4 e R22.2.

### *Exportar receitas*

**Resumo:** Este caso de utilização permite que o paciente descarregue a receitas médicas emitidas.

**Atores:** Paciente

**Pré-condições:**

- Existência de receita emitida no contexto de uma consulta.

**Cenário principal:**

1. O caso de utilização inicia-se quando o paciente pretende descarregar o documento com a receita médica.
2. Aplica-se o cenário principal do caso de utilização consultar receitas.
3. O paciente descarrega o documento correspondente à receita seleccionada e o caso de utilização termina.

**Requisitos:** ver requisitos R11, R12, R20.2 e R20.5.

## Gestão de Exames

A gestão de exames é composta por três tarefas: submissão, consulta dos exames e consulta do resultado dos exames. A tarefa de consulta do resultado dos exames pode ser repartida em duas tarefas: extração do resultado dos exames ou submissão do resultado dos exames. Na figura 4 apresentamos o diagrama de casos de utilização com foco na gestão de exames.

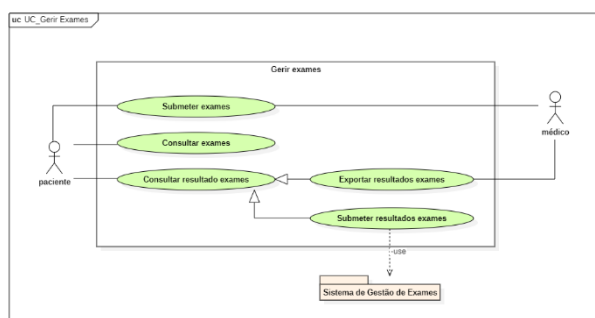


Figura 4 - Casos de utilização da gestão de exames

Abaixo apresentamos o detalhe de cada caso de utilização identificado, bem como a sua relação com os requisitos.

### Submeter exames

**Resumo:** Este caso de utilização permite que o médico submeta exames

**Atores:** Médico

**Pré-condições:**

- O médico precisa de ter os documentos dos exames para submeter nos formatos JPEG, PNG ou PDF.

**Cenário principal:**

1. O caso de utilização inicia-se quando o médico emite exames médicos ao paciente.
2. O médico carrega para o sistema o(s) exame(s) emitidos e o caso de utilização termina.

**Requisitos:** ver requisitos R13, R14 e R22.3.

### Consultar exames

**Resumo:** Este caso de utilização permite que o paciente consulte exames emitidos.

**Atores:** Paciente

**Pré-condições:**

- Existência de, pelo menos, um exame emitido no contexto de uma consulta.

**Cenário principal:**

1. O caso de utilização inicia-se quando o paciente pretende consultar exames emitidos.
2. O paciente seleciona uma consulta.
3. O paciente seleciona um exame emitido no contexto da consulta e o caso de

utilização termina.
<b>Requisitos:</b> ver requisitos R15, R20.2 e R20.6.

### Submeter resultados exames

<p><b>Resumo:</b> Este caso de utilização permite que o paciente ou o Sistema de Gestão de Exames da HealthCare submeta resultados de exames realizados noutras clínicas</p> <p><b>Atores:</b> Paciente e Sistema de Gestão de Exames da HealthCare</p> <p><b>Pré-condições:</b></p> <ul style="list-style-type: none"> <li>• Existência de exame pedido no contexto de uma consulta.</li> <li>• O paciente tem de ter acesso aos resultados dos exames realizados noutras clínicas.</li> <li>• O paciente precisa de ter os documentos dos resultados dos exames para submeter nos formatos JPEG, PNG ou PDF.</li> </ul>
<p><b>Cenário principal 1:</b></p> <ol style="list-style-type: none"> <li>1. O caso de utilização inicia-se quando o paciente recebe os resultados de exames médicos realizados noutra clínica.</li> <li>2. O paciente seleciona a consulta.</li> <li>3. O paciente seleciona o exame.</li> <li>4. O paciente submete os documentos com os resultados dos exames no sistema e o caso de utilização termina.</li> </ol> <p><b>Cenário principal 2:</b></p> <ol style="list-style-type: none"> <li>1. O caso de utilização inicia-se quando os resultados de exames médicos realizados na clínica estão disponíveis no Sistema de Gestão de Exames.</li> <li>2. O sistema de gestão submete os documentos com os resultados dos exames no sistema, identificando o paciente e o exame médico, e o caso de utilização termina.</li> </ol>
<b>Requisitos:</b> ver requisitos R18, R19 e R20.7

### Exportar exames

<p><b>Resumo:</b> Este caso de utilização permite que o paciente e o médico extraiam exames médicos emitidos.</p> <p><b>Atores:</b> Paciente e médico</p> <p><b>Pré-condições:</b></p> <ul style="list-style-type: none"> <li>• Existência de exame emitido no contexto de uma consulta.</li> <li>• O sistema de gestão de exames deverá ter carregado o resultado dos exames ou paciente deverá tê-los carregado no sistema.</li> </ul>
<p><b>Cenário principal:</b></p> <ol style="list-style-type: none"> <li>1. O caso de utilização inicia-se quando o paciente ou o médico pretendem descarregar o documento com o exame médico.</li> <li>2. Aplica-se o cenário principal do caso de utilização consultar exames.</li> <li>3. O utilizador descarrega o documento correspondente ao exame selecionado e o caso de utilização termina.</li> </ol>
<b>Requisitos:</b> ver requisitos R16, R17, R20.8 e R22.4.

## Requisitos

As funções do sistema para o projeto proposto podem ser verificadas na tabela 1.

Tabela 1 - Requisitos funcionais

Ref.	Descrição	Categoria
R1	Agendamento de consultas	Obrigatório
R2	Os campos de agendamento de consulta admitem as seguintes gamas de valores:	Obrigatório
R2.1	Especialidade: clínica geral, dentista, obstetrícia, cardiologia, pediatria, neurologia, oftalmologia	Obrigatório
R2.2	Data agendamento consulta $\geq$ Data corrente + 72h	Obrigatório
R2.3	Hora: 10-20	Obrigatório
R2.4	Minutos: 0-59	Obrigatório
R3	Confirmação de consultas	Obrigatório
R4	Os campos de confirmação de consulta admitem as seguintes gamas de valores:	Obrigatório
R4.1	Data agendamento consulta $\geq$ Data corrente	Obrigatório
R4.2	Hora agendamento consulta $\geq$ Hora atual + 5h	Obrigatório
R5	Cancelamento de consultas	Obrigatório
R6	Os campos de cancelamento de consulta admitem as seguintes gamas de valores:	Obrigatório
R6.1	Motivo de cancelamento: ausência médica, pedido médico, greve.	Obrigatório
R6.2	Cancelamento Manual: Data agendamento consulta $\geq$ Data corrente	Obrigatório
R6.3	Cancelamento Manual: Hora agendamento consulta $\geq$ Hora atual + 5h	Obrigatório
R6.4	Cancelamento Automático: Data agendamento consulta = Data corrente	
R6.5	Cancelamento Automático: Hora agendamento consulta < Hora atual	
R7	Reagendamento de consultas	Obrigatório
R7	Os campos de reagendamento de consulta admitem as seguintes gamas de valores:	Obrigatório
R7.1	Hora agendamento consulta $\geq$ Hora atual + 5h	Obrigatório
R7.2	Data reagendamento consulta $\geq$ Data corrente + 72h	Obrigatório
R7.3	Hora: 10-20	Obrigatório
R7.4	Minutos: 0-59	Obrigatório
R8	Submeter receitas	Obrigatório
R9	Os campos de submissão de receitas admitem as seguintes gamas de valores:	Obrigatório

R9.1	Tipo de documento: JPEG, PNG, PDF	Obrigatório
R10	Consultar receitas	Obrigatório
R11	Exportar receitas	Obrigatório
R12	Os campos de exportar receitas admitem as seguintes gamas de valores:	Obrigatório
R12.1	Tipo de documento: PDF	Obrigatório
R13	Submeter exames	Obrigatório
R14	Os campos de submissão de exames admitem as seguintes gamas de valores:	Obrigatório
R14.1	Tipo de documento: JPEG, PNG, PDF	Obrigatório
R15	Consultar exames	Obrigatório
R16	Exportar resultados exames	Obrigatório
R17	Os campos de exportar resultados exames admitem as seguintes gamas de valores:	Obrigatório
R17.1	Tipo de documento: JPEG, PNG, PDF	Obrigatório
R18	Carregar resultados de exames	Obrigatório
R19	Os campos de carregar resultados exames admitem as seguintes gamas de valores:	Obrigatório
R19.1	Tipo de documento: JPEG, PNG, PDF	Obrigatório
R20	gUI paciente apresenta as seguintes opções:	Obrigatório
R20.1	Agendar consulta	Obrigatório
R20.2	Visualizar consulta	Opcional
R20.3	Cancelar ou reagendar consulta	Opcional
R20.4	Visualizar receitas	Opcional
R20.5	Exportar receitas	Opcional
R20.6	Visualizar exames	Opcional
R20.7	Submeter resultados exames	Opcional
R20.8	Exportar resultados exames	Opcional
R21	gUI administrativo apresenta as seguintes opções:	Obrigatório
R21.1	Agendar consulta	Obrigatório
R21.2	Gerir consulta(s)	Obrigatório
R22	gUI médico apresenta as seguintes opções:	Obrigatório
R22.1	Carregar receitas	Obrigatório
R22.2	Consultar receitas emitidas	Opcional
R22.3	Carregar exames	Obrigatório
R22.4	Exportar resultados exames	Opcional



# Arquitetura

Com o objetivo de projetar o desenvolvimento dos casos de utilização mencionados, e com a intenção de tornar o HealthCenter funcional, começamos o processo de desenvolvimento da arquitetura. Este processo foi iniciado com a identificação das partes do sistema e a interação entre as mesmas. Com essa informação foram criados diagramas de interação e modelo de domínio. Este processo é iterativo sendo as partes do sistema e a relação entre estas, identificada à medida que o desenho da arquitetura do projeto avança. Por este motivo, os diagramas refletem a fase final do desenho dos diagramas durante a primeira iteração do projeto.

## Diagramas de interação

Os diagramas de interação para cada um dos casos de utilização, identificados no capítulo anterior, estão ilustrados nas figuras 10 a 20, nos anexos.

## Modelo de domínio

A arquitetura foi criada com base em padrões de desenvolvimento de software: arquitetura model, view, controller (MVC) e arquitetura por camadas.

Com o padrão MVC, a aplicação foi dividida em três componentes principais:

- Model: lógica de negócio da aplicação;
- View: interfaces gráficas da aplicação;
- Controller: controlo e coordenação das componentes anteriores, estabelece a ligação entre a interação do utilizador com o processamento de dados do sistema.

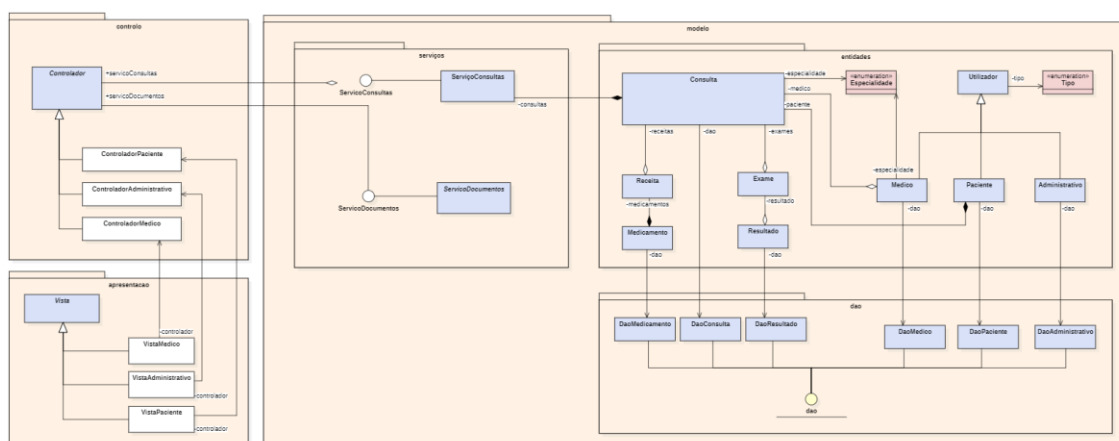


Figura 5 - Modelo de domínio

A arquitetura por camadas consiste em atribuir um propósito único para cada camada, permitindo que apenas as camadas superiores tenham acesso aos recursos das camadas inferiores.

O pacote model é o mais complexo, pois contém toda a informação lógica da aplicação, e subdivide-se em três pacotes:

- Serviços: serviços disponíveis na aplicação;
- Entidades: mapeia as entidades do sistema de gestão de base de dados para entidades no domínio da aplicação;
- Dao: interage com o sistema de gestão de base de dados.

O pacote controlo contém todos os mecanismos necessários para manter o controlo da aplicação. E o pacote apresentação contém as vistas da aplicação.

As figuras 6, 7 e 8 materializam o diagrama de mecanismos para os casos de utilização da gestão de consultas, gestão de receitas e gestão de exames, respetivamente.

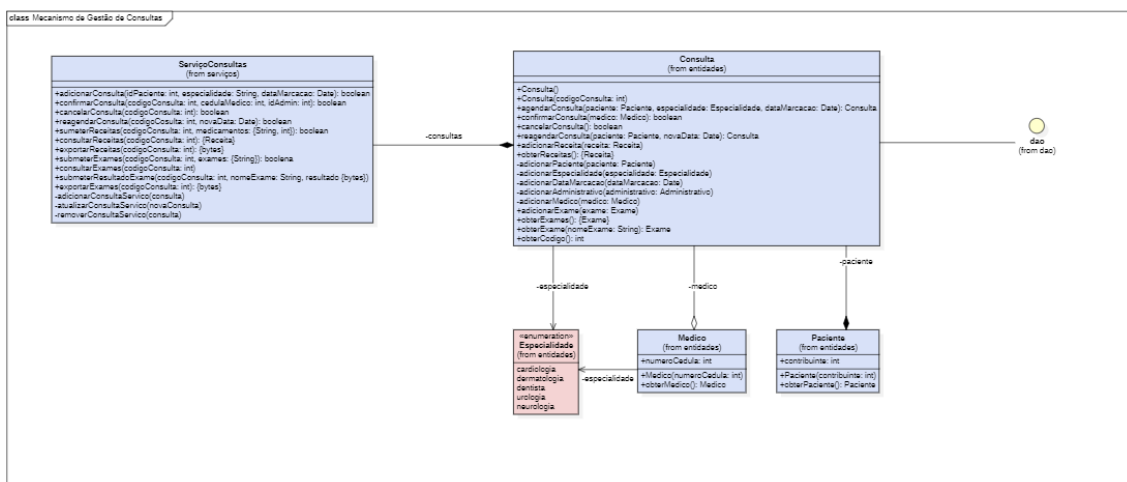


Figura 6 - Mecanismo de gestão de consultas

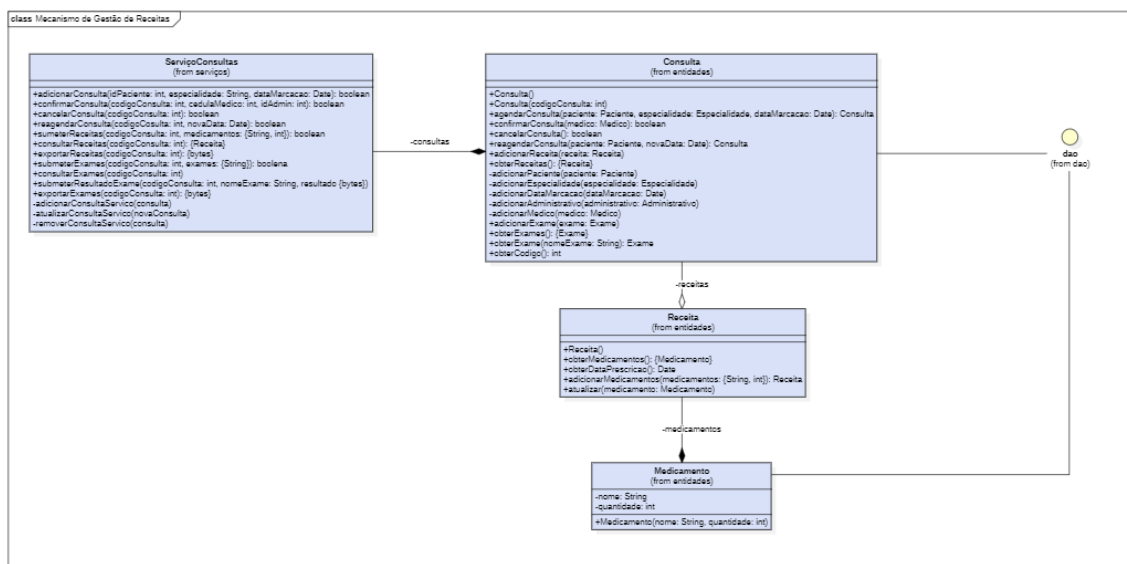


Figura 7 - Mecanismo de gestão de receitas

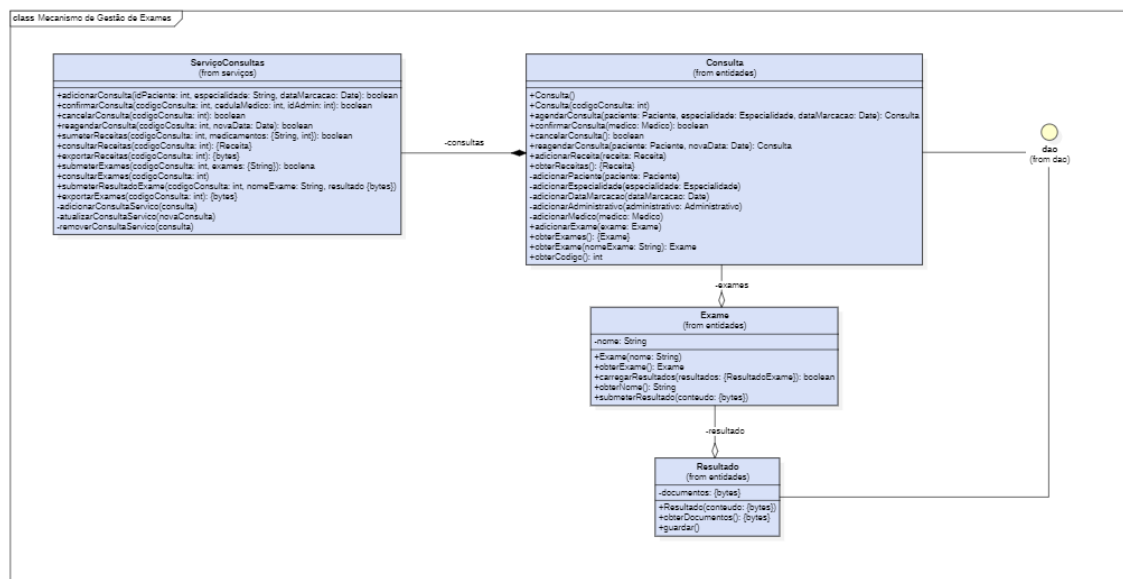


Figura 8 - Mecanismo de gestão de exames

# Implementação

## Código Fonte

Após a definição da arquitetura do projeto, avançamos para a implementação do mesmo. Nesta fase, foi criado o código fonte para um caso de utilização explorado – agendamento de consulta do paciente.

```
class VistaPaciente(Vista):
    def VistaPaciente(self):
        self.__consultas = None

    def mostrarEcraConsultas(self, consultas):
        print("Consultas:")
        for consulta in consultas:
            print(consulta)
        (...)
```

```
class ControladorPaciente(Controlador):
    def ControladorPaciente(self, vista):
        self.__servicoConsultas = ServicoConsultas()
        self.__vista = vista

    def adicionarConsulta(self, idPaciente, especialidade, dataMarcacao):
        consultas = self.__servicoConsultas.adicionarConsulta(idPaciente, especialidade, dataMarcacao)
        self.__vista.mostrarEcraConsultas(consultas)
        (...)
```

```
class ServicoConsultas(IServicoConsultas):
    def ServicoConsultas(self):
        self.__consultas = []

    def adicionarConsulta(self, idPaciente, especialidade, dataMarcacao):
        self.__consulta = Consulta().agendarConsulta(idPaciente, especialidade, dataMarcacao)
        if isinstance(self.__consulta, Consulta):
            self.__consultas.append(self.__consulta)
            return self.__consultas
        return None
        (...)
```

```
class Consulta:

    def Consulta(self):
        self.__dao = DaoConsulta()

    def Consulta(self, codigoConsulta):
        self.__codigoConsulta = codigoConsulta
        self.__dao = DaoConsulta()

    def agendarConsulta(self, paciente, especialidade, dataMarcacao):
        self.__paciente = paciente
        self.__especialidade = especialidade
        self.__dataMarcacao = dataMarcacao
        return (self.__dao.criar(self))
        (...)
```

```

class DaoConsulta(iDao):
...
    def criar(self, entidade):
        with open(".\codigo\consultas.txt", 'w') as f:
            f.write(str(entidade))

    def ler(self, entidade):
        with open(".\codigo\consultas.txt", 'r') as f:
            for line in f:
                if line.startswith(entidade.id):
                    return line
            (...)

```

## Testes de desenvolvimento

Para além da implementação do código fonte para o caso de utilização de agendamento de consulta por parte de um paciente, foi também desenvolvido o código de testes para o caso de utilização.

```

class TestesConsulta:

    def main(self):
        print("Testes de Consulta")
        print("1. Teste de Consulta de um Paciente")

        vista = VistaPaciente()
        controlador = ControladorPaciente(vista)
        listaConsultas = controlador.adicionarConsulta(1, "Cardiologia", "2019-12-12")
        vista.mostrarConsultas(listaConsultas)

    (...)

```

## Modelo de implantação

Para a implementação do projeto foi necessário definir inicialmente quais as tecnologias a utilizar. Para isso, foi desenvolvido o modelo de implantação apresentado na figura ZZ.

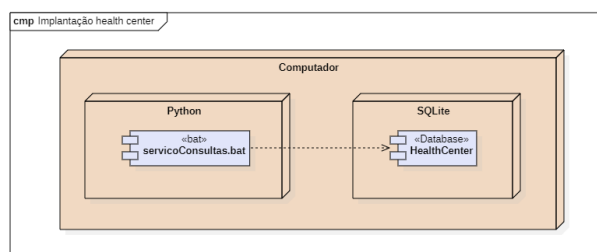


Figura 9 - Modelo de implantação

## Análise crítica

A metodologia proposta para o desenvolvimento do projeto HealthCenter está alinhada com princípios fundamentais da Engenharia de Software, seguindo um modelo incremental e iterativo baseado no modelo cíclico. A escolha desse modelo é justificada pela necessidade de interação com os usuários à medida que o projeto vai evoluindo, permitindo melhorias contínuas do sistema de acordo com o feedback recebido. A abordagem de dividir o desenvolvimento em fases, atividades e iterações é adequada para garantir uma evolução gradual e controlada do sistema.

A descrição dos requisitos e casos de uso é clara e detalhada, facilitando a compreensão do escopo do projeto. A estrutura dos requisitos funcionais é bem organizada, especificando claramente as funcionalidades esperadas.

Este projeto foi desenvolvido durante a primeira iteração do modelo e, por esse motivo, existem diversas componentes que carecem de melhorias que transitam para a próxima iteração do modelo de desenvolvimento.

À medida que existam incrementos ao projeto desenvolvido também a complexidade do sistema aumentará, exigindo uma gestão cuidadosa para garantir que as integrações de novas funcionalidades não causem problemas em partes existentes do sistema. O modelo MVC e arquitetura por camadas usado foi utilizado de forma a promover o desacoplamento e redução da complexidade através da modularização e distinção de responsabilidades por cada módulo. Desta forma, a implementação do projeto torna-se menos complexa e intuitiva permitindo simplificar eventuais alterações necessárias.

## Conclusão

O projeto HealthCenter representa uma iniciativa inovadora da clínica HealthCare para melhorar a gestão e o acompanhamento médico de seus pacientes. A proposta de uma aplicação para dispositivos móveis, integrada com o sistema médico da clínica, visa solucionar problemas relacionados à inconsistência nas informações fornecidas pelos pacientes e aos desafios na apresentação física de exames.

Ao longo do desenvolvimento do projeto, foram adotados princípios da engenharia de software, seguindo o modelo incremental e iterativo. Essa abordagem permitiu uma adaptação contínua às necessidades e feedback dos utilizadores, garantindo a entrega de um produto mais alinhado com as expectativas.

A análise de requisitos delineou as funcionalidades essenciais da aplicação, desde a marcação de consultas até à gestão de receitas e exames. A definição de casos de uso proporcionou uma visão macro das interações entre os atores (paciente, médico e administrativo) e o sistema, contribuindo para a compreensão global das operações.

A arquitetura do sistema foi projetada com base em padrões consolidados, como o modelo-view-controller (MVC) e a arquitetura por camadas. Essa estrutura facilita a manutenção, expansão e compreensão do código, promovendo uma separação clara entre a lógica de negócios, a apresentação e o controle.

A implementação do projeto começou com a codificação de um caso de uso específico, o agendamento de consultas por parte do paciente. Além disso, foram desenvolvidos testes para validar a funcionalidade implementada, contribuindo para a qualidade do software.

No âmbito das tecnologias utilizadas, o projeto considerou o modelo de implantação, visando uma infraestrutura eficiente e escalável e cuidadosamente planeada para garantir a interoperabilidade e a segurança do sistema.

Em suma, o projeto HealthCenter representa um passo significativo na busca por soluções tecnológicas aplicadas à área médica. A implementação bem-sucedida dessas inovações não beneficiará apenas os pacientes da clínica HealthCare, mas também abre portas para futuras expansões e colaborações no setor da saúde. A aplicação proposta ambiciona centralizar o histórico médico dos pacientes em nível nacional, indicando um potencial impacto positivo no sistema de saúde como um todo.

# Anexos

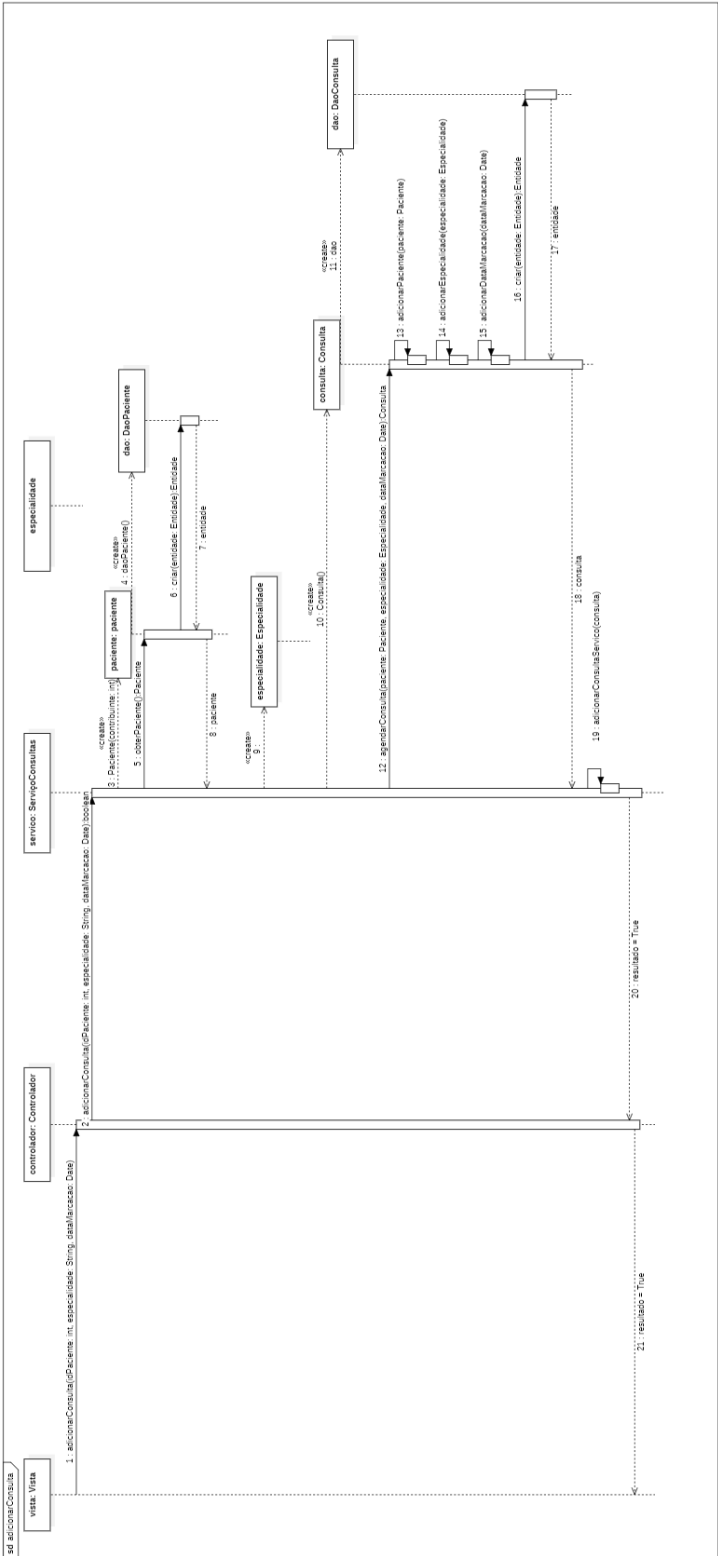


Figura 10 - Diagrama de interação adicionarConsulta



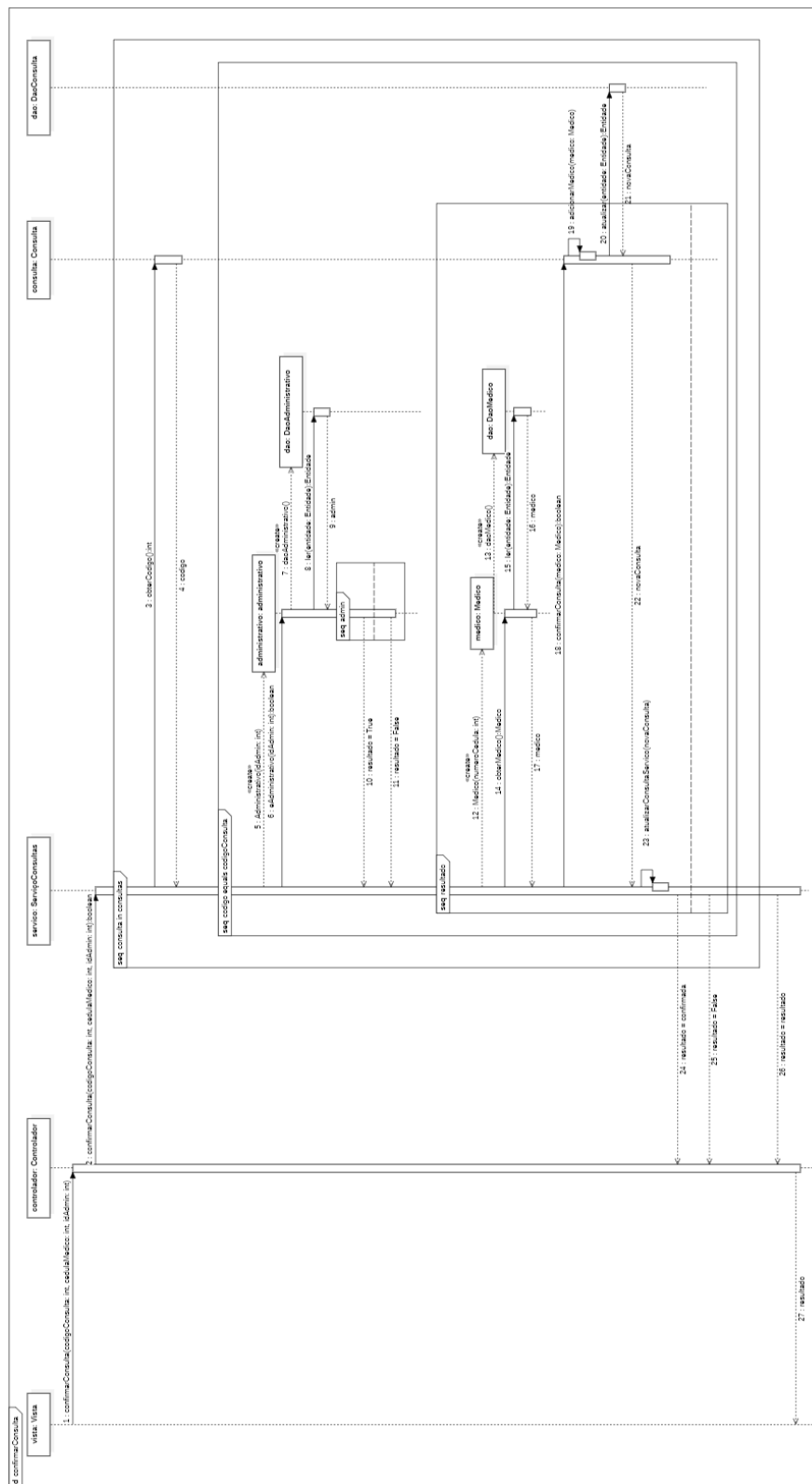


Figura 11 - Diagrama de interação confirmarConsulta

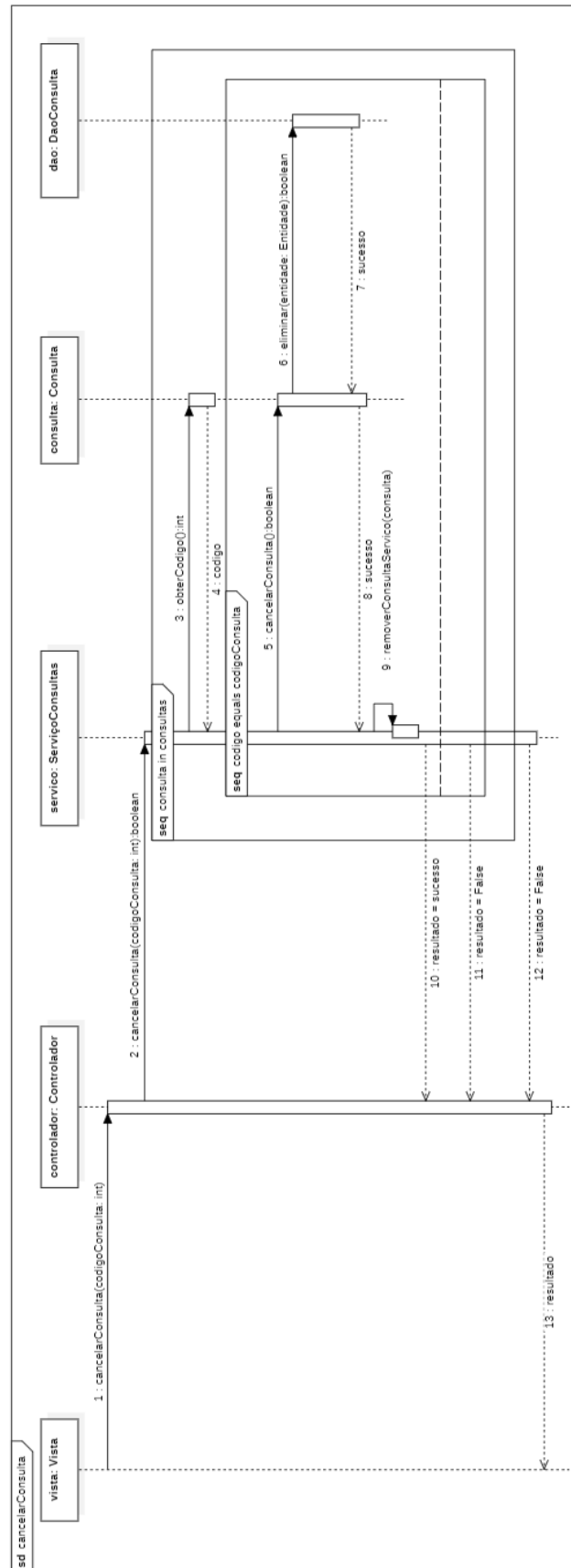


Figura 12 - Diagrama de interação cancelarConsulta

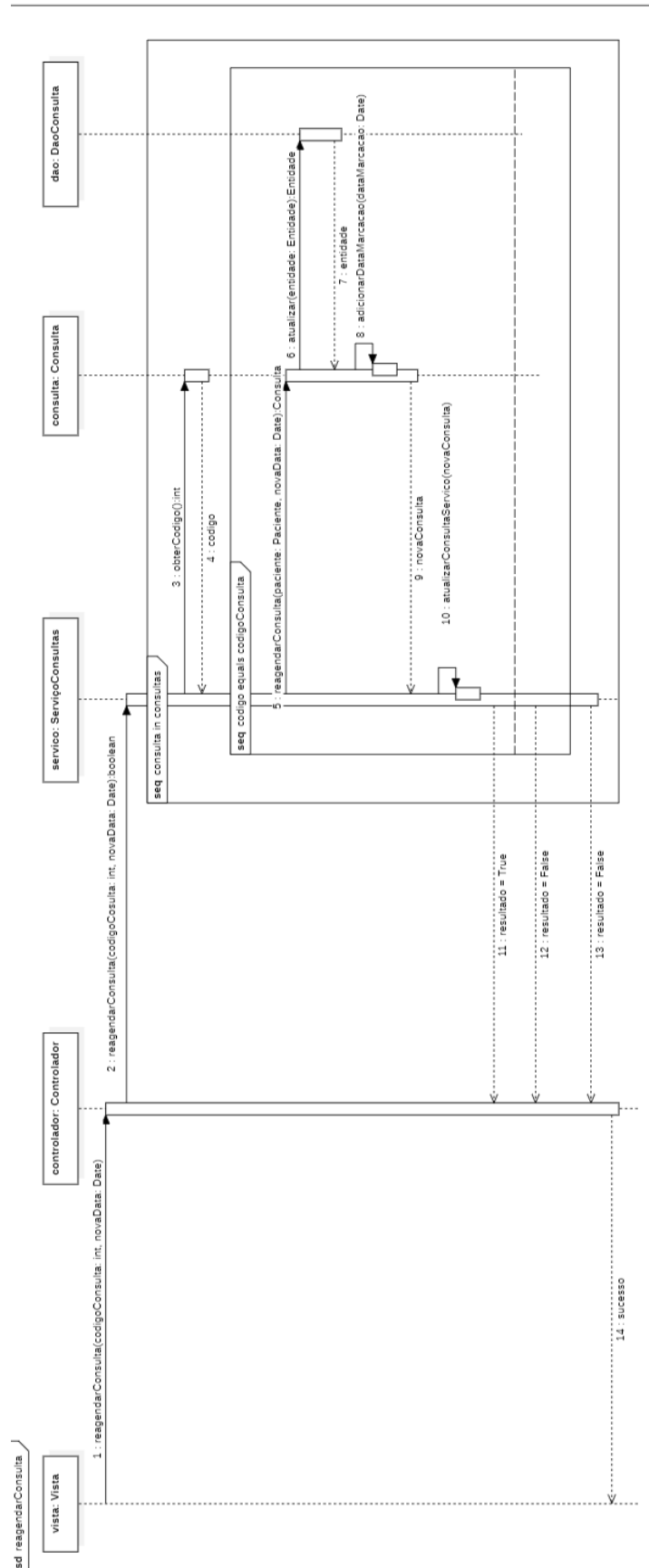


Figura 13 - Diagrama de interação reagendarConsulta

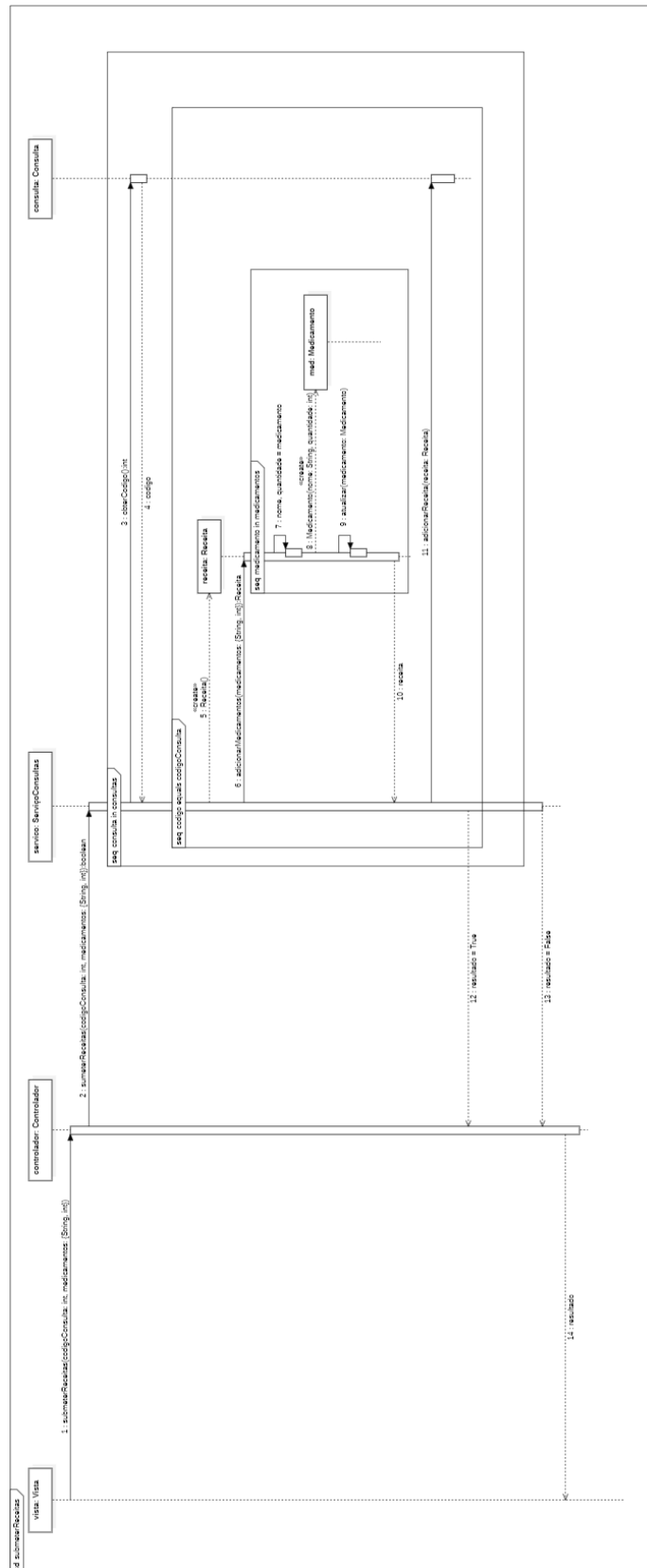


Figura 14 - Diagrama de interação submeterReceitas

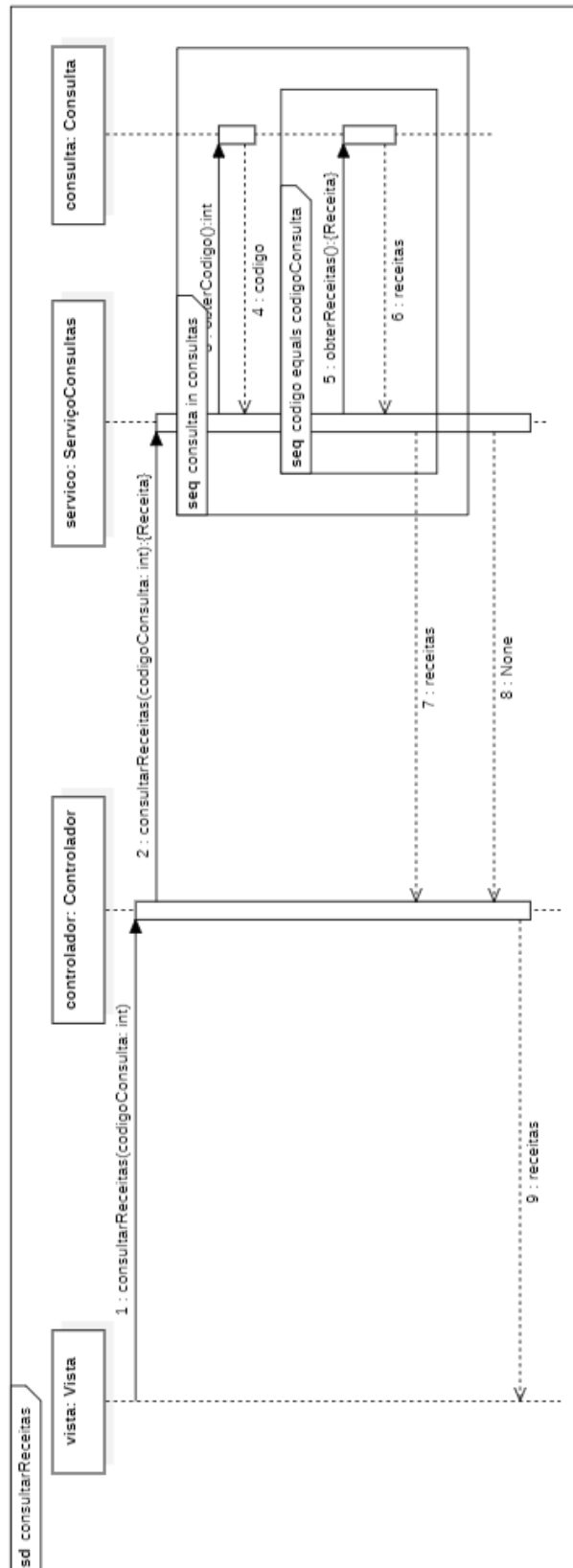


Figura 15 - Diagrama de interação consultarReceitas

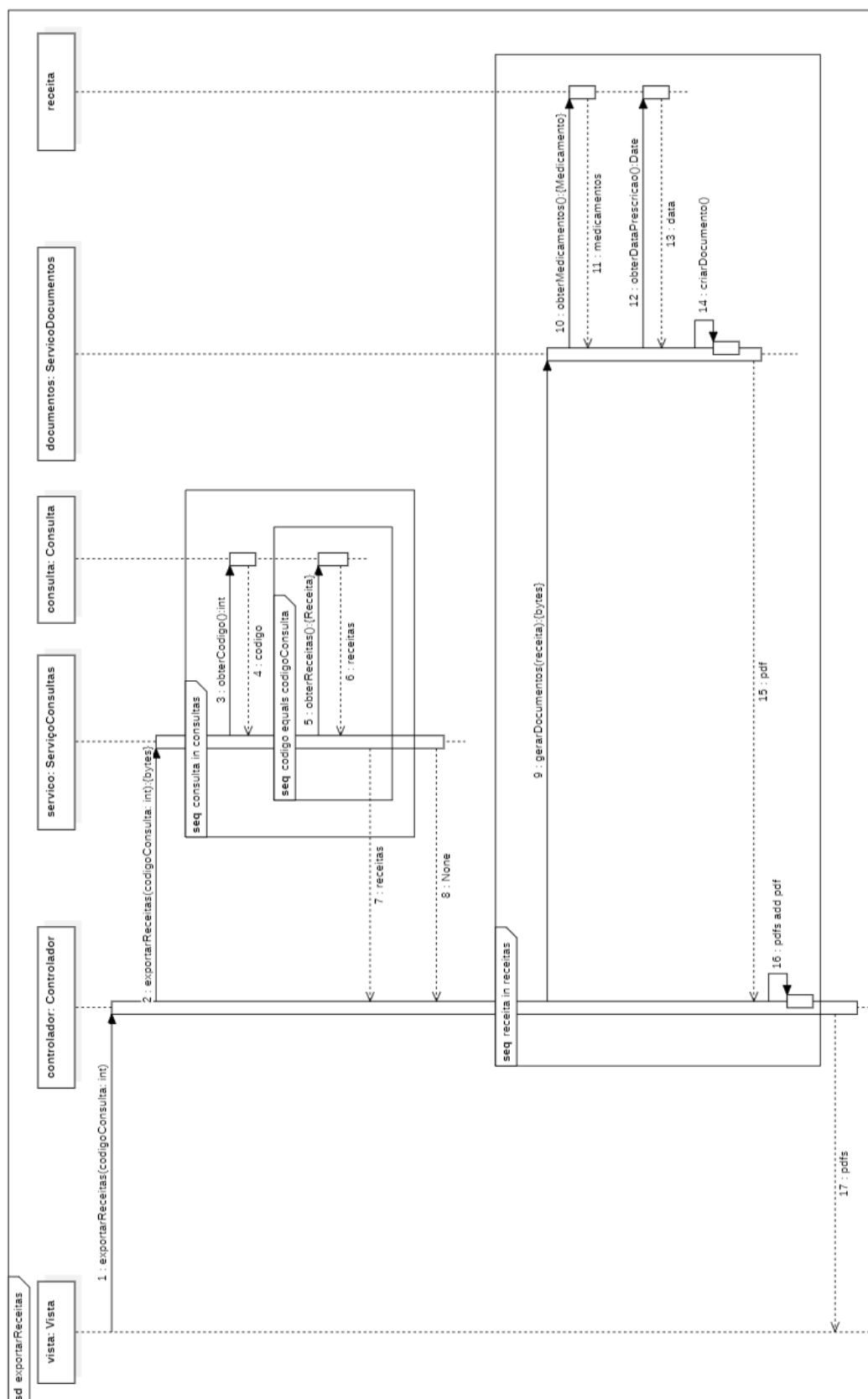


Figura 16 - Diagrama de interação exportarReceitas

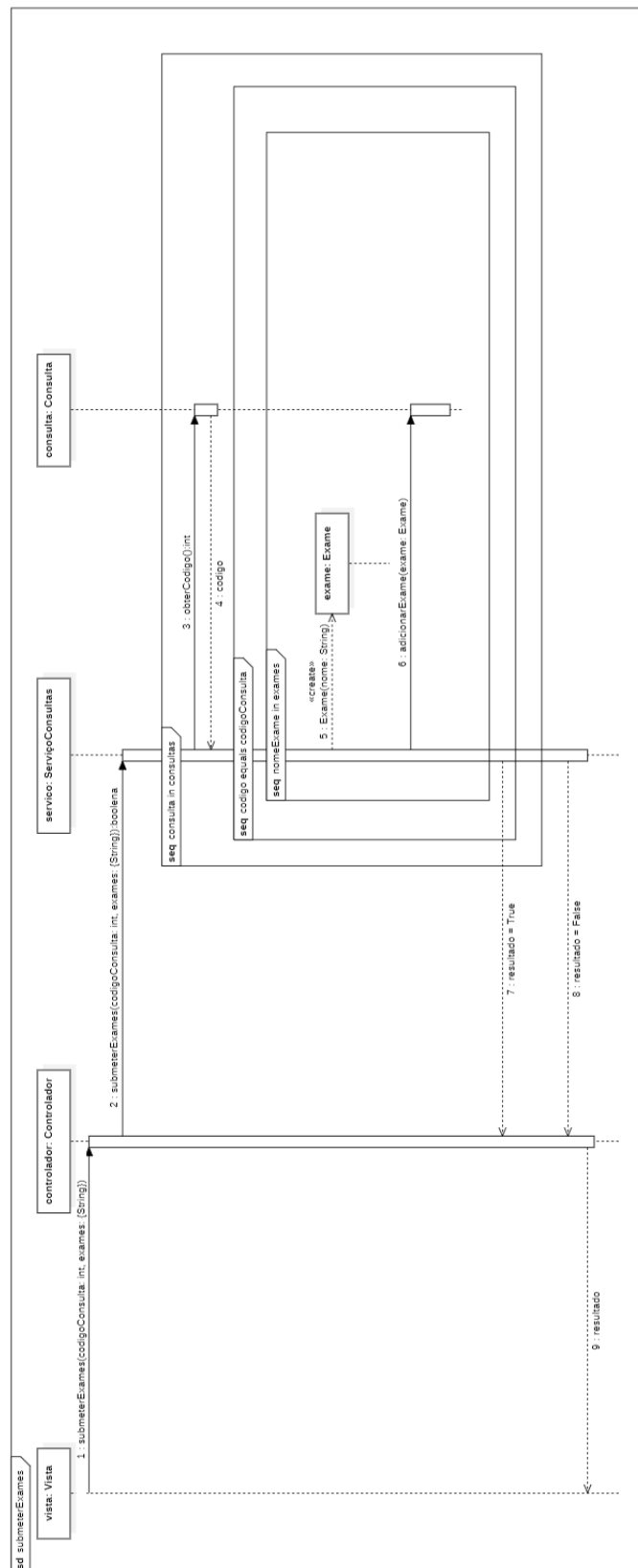


Figura 17 - Diagrama de interação submeterExames

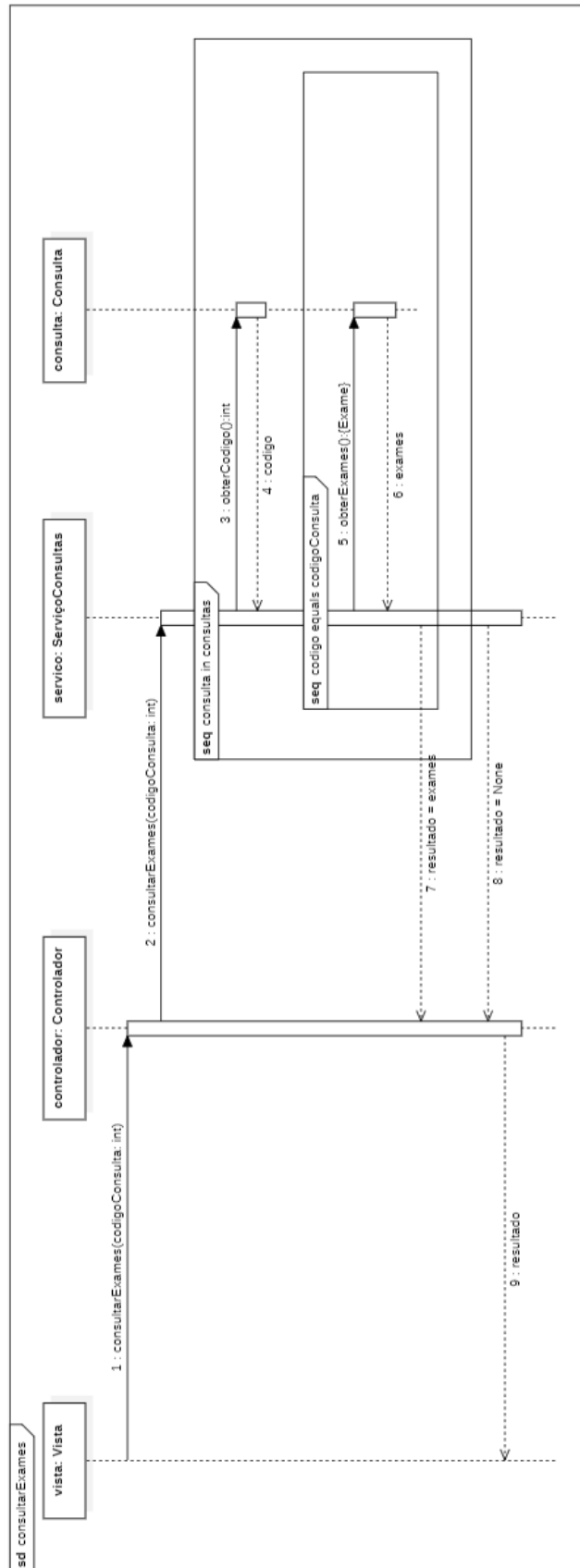


Figura 18 - Diagrama de interação consultarExames



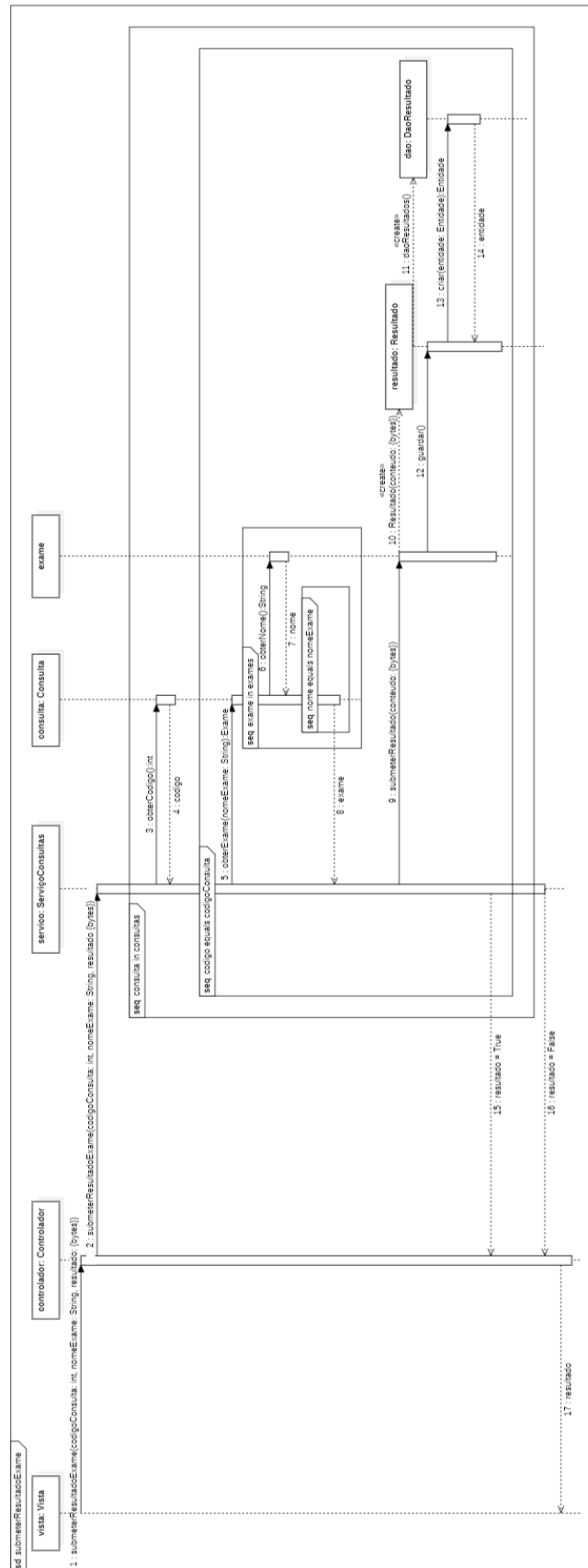


Figura 19 - Diagrama de interação submeterResultadoExame

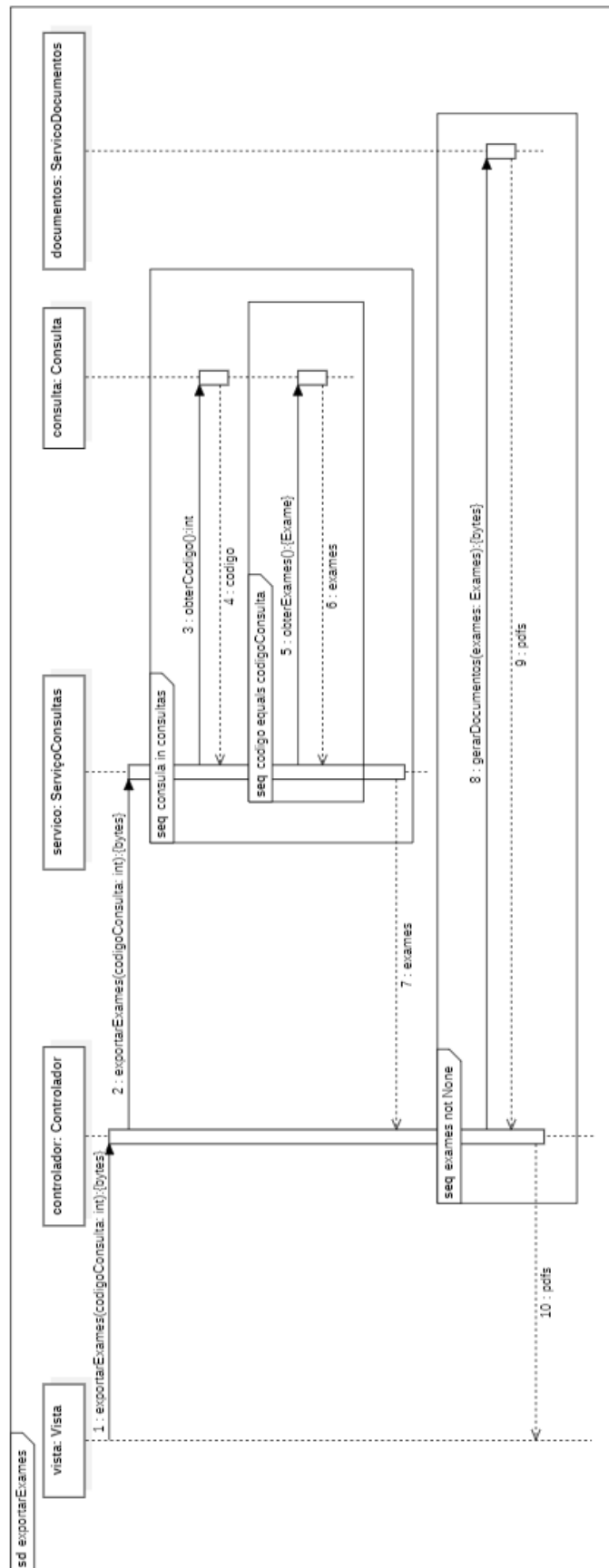


Figura 20 - Diagrama de interação exportarExames