
DAS

**Documento de Arquitetura de
Software**

V1LLSEC

Versão <Final-Revisada>

V1LLSEC	Versão: Final - Revisada
Documento de Arquitetura de Software	Data Versão: 09/08/2020

Histórico das Versões

Data	Versão	Descrição	Autor	Revisor
02/04/2020	1.0	Iniciação	EXSR	EXSR
08/05/2020	1.1	Novos Requisitos	EXSR	EXSR
27/06/2020	2.0	Re-fatoração do código mudanças em algumas funcionalidade	EXSR	EXSR
16/07/2020	2.1	Produção	EXSR	EXSR
09/08/2020	3.0	Final	EXSR	EXSR

V1LLSEC	Versão: Final - Revisada
Documento de Arquitetura de Software	Data Versão: 09/08/2020

Cliente V1LLSEC
Documento Documento de Arquitetura de Software
Data 09 de agosto de 2020
Autor (ES) Eder Xavier de Souza Rosa

Página de Assinaturas

Revisado e Eder Xavier de Souza Rosa (desenvolvedor)
Aprovado
por:

Revisado e
Aprovado
por:

Revisado e
Aprovado
por:

V1LLSEC	Versão: Final - Revisada
Documento de Arquitetura de Software	Data Versão: 09/08/2020

Índice

7.1. OBJETIVO	5
7.2. ESCOPO	5
7.3. REPRESENTAÇÃO ARQUITETURAL.....	6
7.4. RESTRIÇÕES ARQUITETURAIS.....	10
7.5. OBJETIVO E RESTRIÇÕES ARQUITETURAIS	10
7.6. VISÃO LÓGICA	11
7.7. CAMADA DE APRESENTAÇÃO – FRONTEND: ANGULAR.....	13
7.8. CAMADA DE NEGÓCIO – BACKEND JAVA	14
7.8.1. Pacote resources.....	16
7.8.2. Pacote helpers.....	17
7.8.3. Pacote dtos.....	18
7.8.4. Pacote utilities.....	19
7.8.5. Pacote validation.....	20
7.8.6. Pacote annotation.....	20
7.8.7. Pacote exceptions.....	21
7.8.8. Pacote domain.....	22
7.8.9. Pacote enums.....	23
7.8.10. Pacote configuration.....	23
7.8.11. Pacote repository.....	24
7.8.12. Pacote security.....	25
7.8.13. Pacote services.....	26
7.8.14. Pacote filters.....	28
7.9. REALIZAÇÃO DOS CASOS DE USO	29
7.9.1. Diagrama de Sequência Geral: Inserção de Registro.....	29
7.9.2. Diagrama de Sequência Geral: Alteração de Registro.....	30
7.9.3. Diagrama de Sequência Geral: Exclusão de Registro.....	31
7.9.4. Diagrama de Sequência Geral: Buscar Registro.....	32
7.9.5. Diagrama de Sequência: Login.....	33
7.10. VISÃO DE IMPLANTAÇÃO	34
7.11. VISÃO DE IMPLEMENTAÇÃO	34
7.12. VISÃO DE DADOS.....	35
7.13. TAMANHO E DESEMPENHO.....	36

V1LLSEC	Versão: Final - Revisada
Documento de Arquitetura de Software	Data Versão: 09/08/2020

7.1. Objetivo

A finalidade deste “Documento de Arquitetura de Software – DAS” é servir de agente facilitador para o desenvolvimento do software, aplicando um modelo de arquitetura e apresentando uma visão da arquitetura do software (V1LLSEC.api) possibilitando a identificação de aspectos importantes do software reunindo as informações necessárias para controlar as atividades arquiteturais de forma a oferecer uma visão macro da arquitetura do software bem como os requisitos não funcionais envolvidos.

7.2. Escopo

O escopo deste Documento de Arquitetura de Software (DAS) visa documentar os padrões de software, componentes do software e plataforma de hardware, bem como todas as ferramentas envolvidas no desenvolvimento do software (V1LLSEC.api) e toda a arquitetura necessária para apoiar e ajudar no desenvolvimento do software, facilitando assim o entendimento do mesmo e também a suas manutenções.

V1LLSEC	Versão: Final - Revisada
Documento de Arquitetura de Software	Data Versão: 09/08/2020

7.3. Representação Arquitetural.

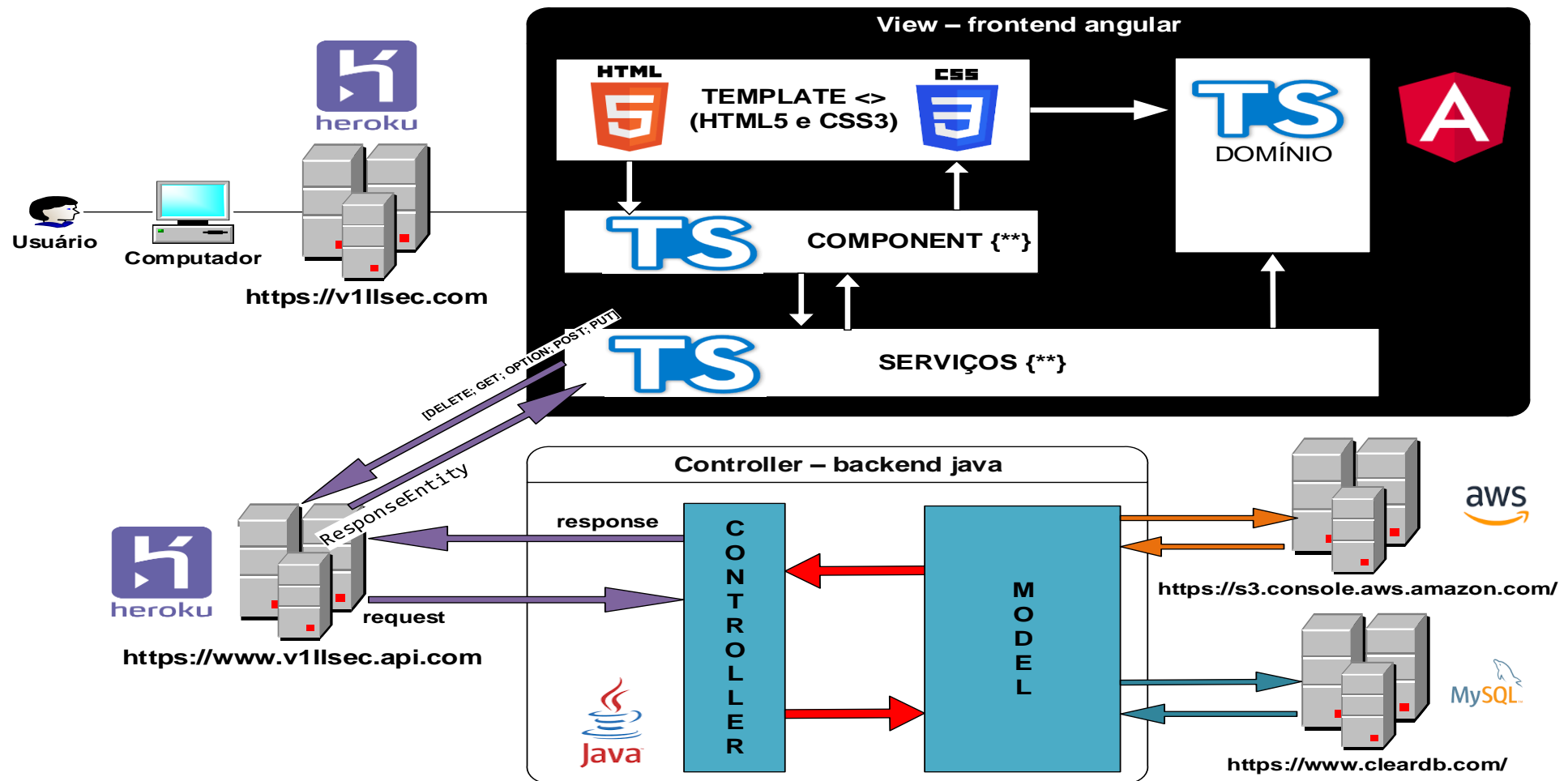


Figura 1.01 – Modelo de Arquitetura Genérico

V1LLSEC	Versão: Final - Revisada
Documento de Arquitetura de Software	Data Versão: 09/08/2020

A camada view (frontend - angular), é executada do lado do cliente, possui todas as telas, representações, serviços e tratamentos necessários para que os dados recebidos da API (backend – java) sejam exibidos para o usuário da aplicação de maneira segura e sucinta.

A camada de controller (backend – java), é executada do lado do servidor e tem como objetivo receber, preparar e enviar as requisições para a camada de Model para que a mesma realize a execução das tarefas e retorne as respostas para frontend com os dados necessários. A camada de Model (backend – java), é executada do lado do servidor e tem como objetivo tratar todas as requisições vindas dos controllers, composta por todos os serviços especializados, tratamentos de erros, DTOS, repositórios, entidades, filtros, configurações e segurança.

Por fim os servidores de banco de dados mysql do Heroku responsável por conter e gerenciar toda a base de dados do sistema, e o repositório de arquivos AWS-S3 da Amazon responsável por armazenar e disponibilizar todos os vídeos, imagens, documentos armazenados pelos usuários e usados pelo sistema.

V1LLSEC	Versão: Final - Revisada
Documento de Arquitetura de Software	Data Versão: 09/08/2020

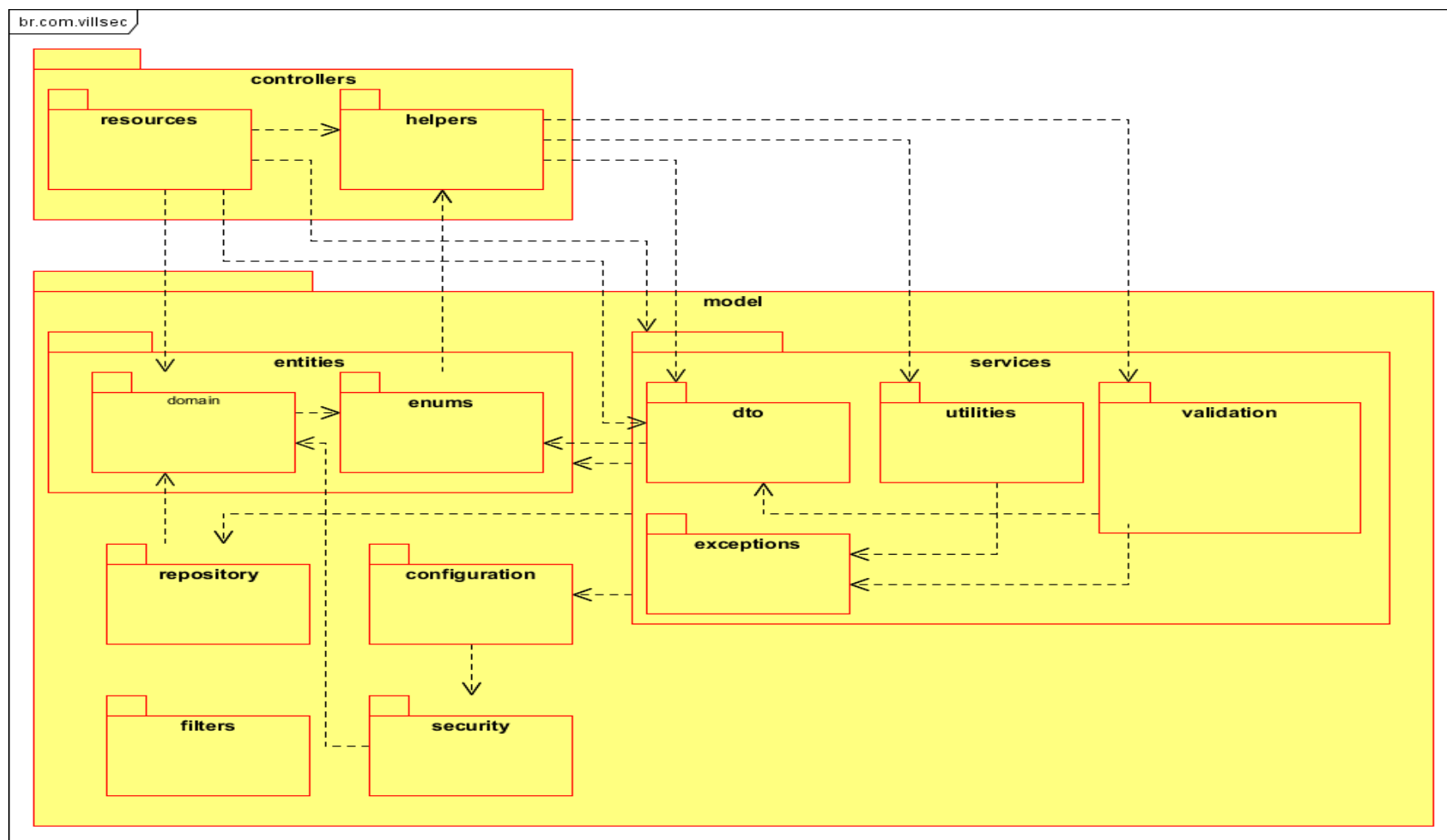


Figura 7.02 – Diagrama em Pacotes do backend.

V1LLSEC	Versão: Final - Revisada
Documento de Arquitetura de Software	Data Versão: 09/08/2020

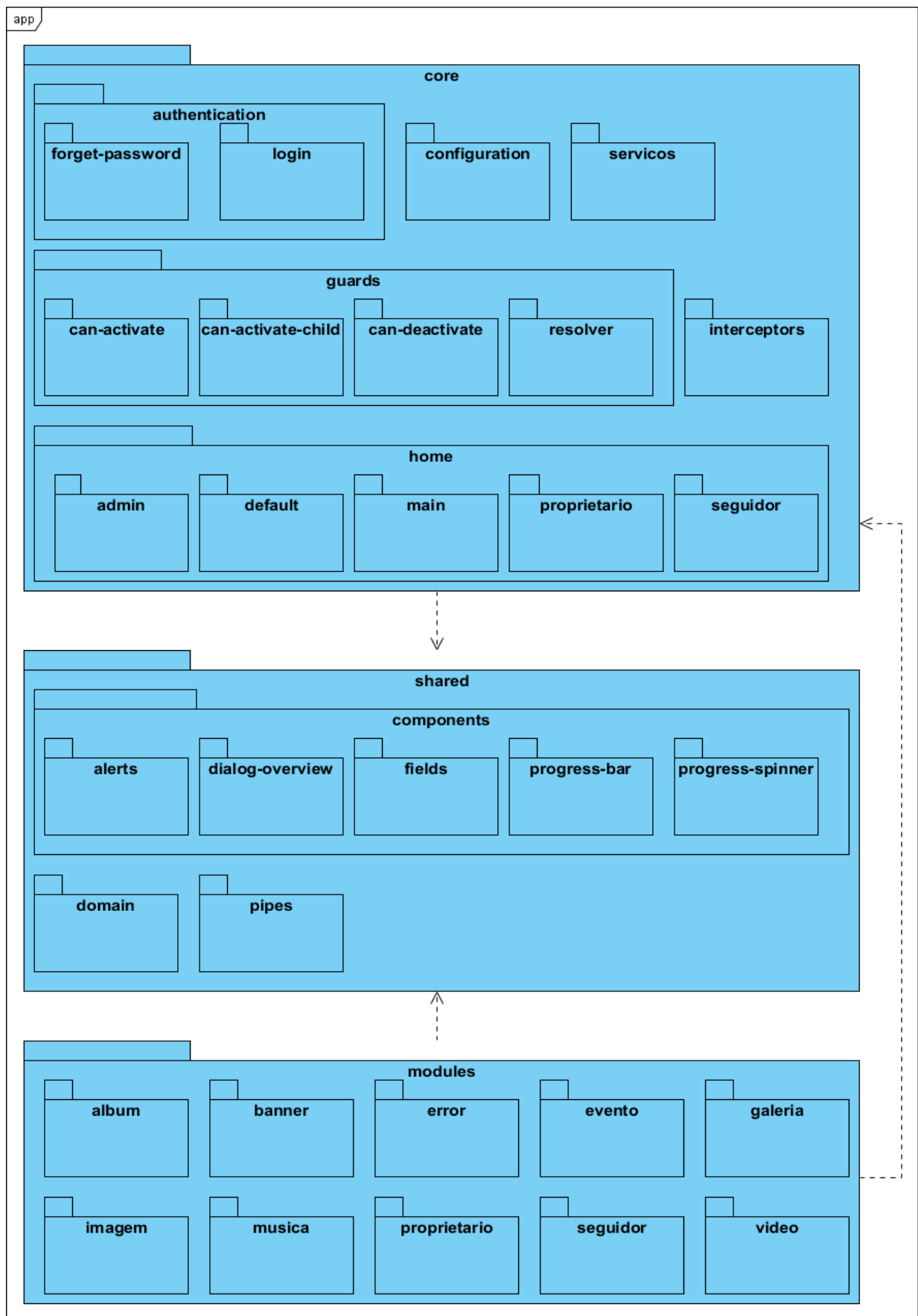


Figura 7.03 – Diagrama de pacotes – Frontend.

V1LLSEC	Versão: Final - Revisada
Documento de Arquitetura de Software	Data Versão: 09/08/2020

7.4. Restrições arquiteturais

Para o desenvolvimento deste software foram identificadas algumas diretrizes e tecnologias:

- Utilização de JDK 1.8 do JAVA.
- Utilização do Framework Spring --v 2.1.2: Boot, Data, Security e Mail.
- Utilização do AWS-Java-sdk – S3 Amazon.
- Utilização do Framework thymeleaf.
- Utilização do Framework Maven.
- Utilização do Framework Hibernate.
- Utilização do Angular update - 10.
- Utilização do Bootstrap - 4.5.0.
- Utilização do Express - 4.17.1.
- Utilização do Swiper - 5.4.5.
- Utilização do Typescript - 3.9.7.
- Utilização do Material Design Lite – 1.3.0.
- Utilização do Angular-Material – 10.1.2.
- Utilização do Node – 12.16.3.
- Utilização do NPM – 6.14.4.
- Utilização do Heroku – 7.42.3.
- Utilização do Git – 2.25.0.windows.1
- Utilização do Mysql 8.0.21.
- Utilização do GitHub.

7.5. Objetivo e Restrições arquiteturais

Considerando premissas definidas para o sistema V1LLSEC pode-se citar as seguintes restrições:

- Utilização da linguagem JAVA no backend e ANGULAR no Frontend
- O software deve funcionar na web
- O SGBD utilizado deverá ser o Mysql ou PostgreSQL
- Utilização de softwares livres quando possível para minimizar custos.

V1LLSEC	Versão: Final - Revisada
Documento de Arquitetura de Software	Data Versão: 09/08/2020

7.6. Visão Lógica

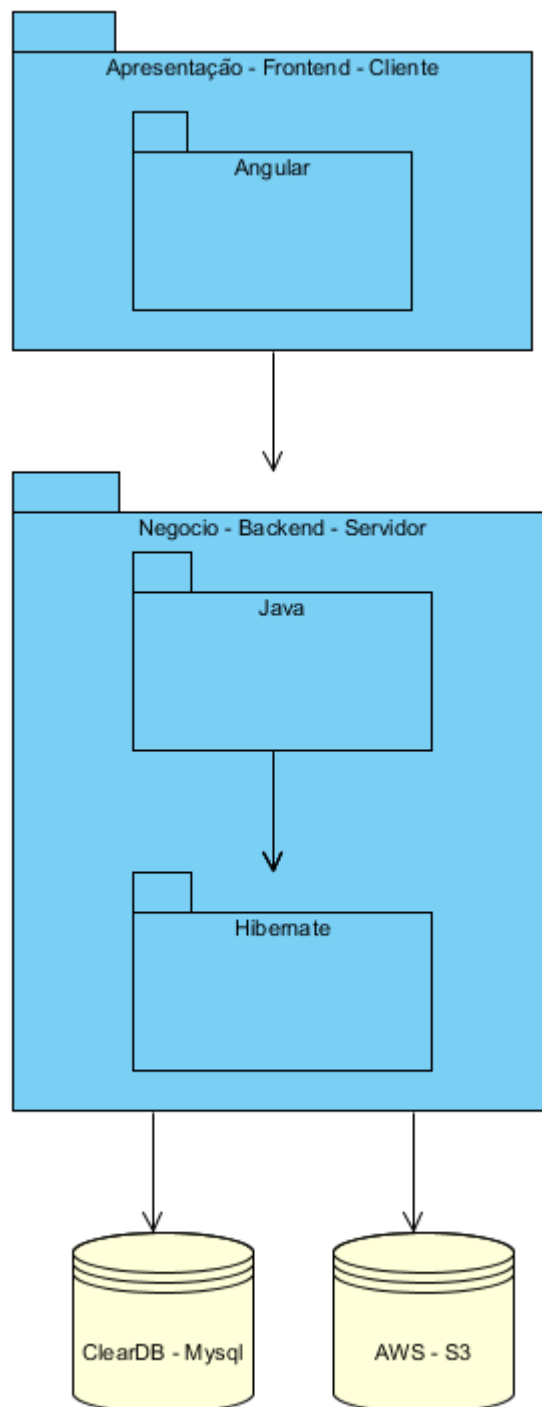


Figura 7.04 – Diagrama de Camadas.

V1LLSEC	Versão: Final - Revisada
Documento de Arquitetura de Software	Data Versão: 09/08/2020

Apresentação: Contém as interfaces gráficas que serão executadas do lado do cliente. A aplicação de frontend baseada na técnica SPA utilizando o framework angular, com objetivo de servir os usuários dados de forma agradável e dinâmica, resolvendo e formatando dados e transformando as respostas do servidor em tabelas formulários e demais formatos de tela necessários para navegação intuitiva e clara dos usuários.

Negócio: Contém a API construída em Java que é executada do lado do servidor, composta de todas as classes de controle, serviço, tratamentos de erros, mapeamento de entidades, construtores de respostas, especializados e comunicação com APIs de terceiros (AWS-S3), tratando requisições dos tipos DELETE, GET, OPTION, POST e PUT, vindas do frontend.

Persistência: Contém todo mapeamento de estrutura relacional necessário para manter os dados enviados pelos usuários de forma íntegra e segura em uma base de dados Mysql disponível 24/7 para requisições de qualquer lugar do mundo.

V1LLSEC	Versão: Final - Revisada
Documento de Arquitetura de Software	Data Versão: 09/08/2020

7.7. Camada de Apresentação – Frontend: Angular

Nesta camada, temos o pacote “views” que contém todos os arquivos relacionados a exibição de informações para o usuário, o que engloba as páginas, formulários, tabelas e botões em HTML5 e CSS3, bem como os serviços, filtros de autorização, controles de acesso, guardas de rotas, Pipes, Patterns e configurações em Typescript, além dos módulos, rotas e providers necessários para comunicação e interação entre todas estas páginas, componentes e recursos usando framework Angular (update 10) como um todo.

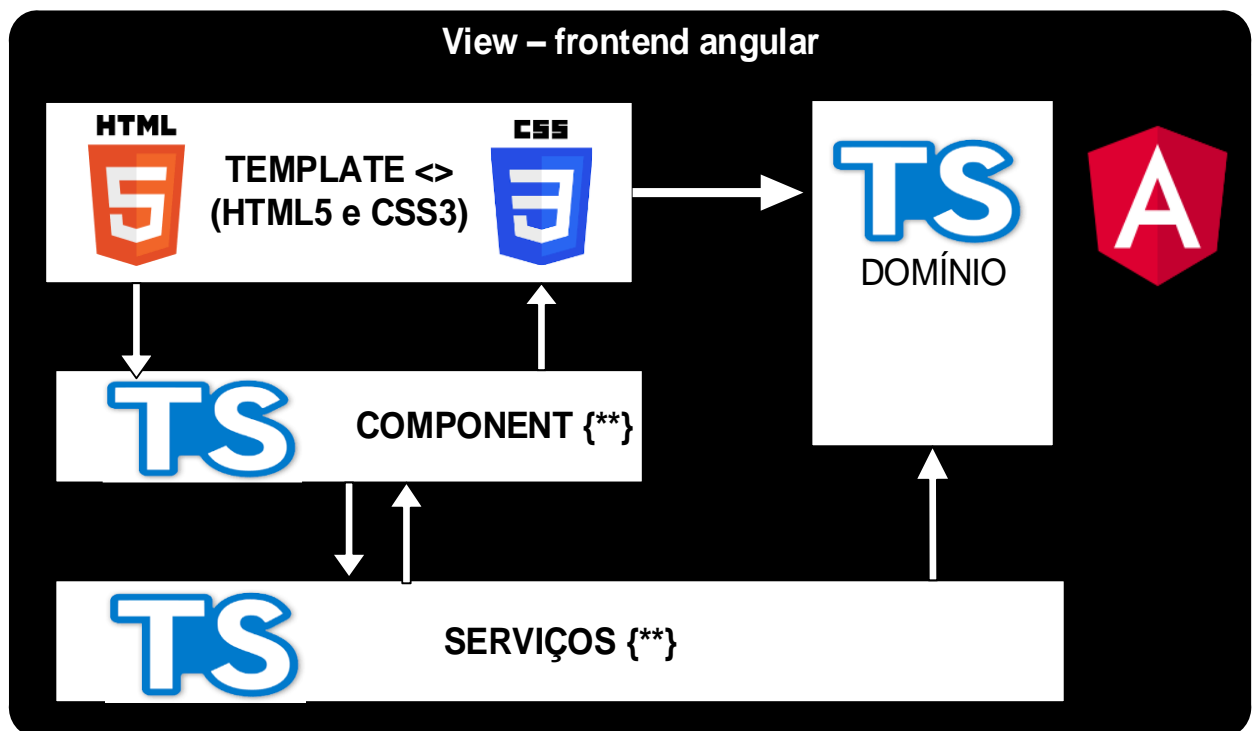


Figura 7.05 – Camada de Apresentação

V1LLSEC	Versão: Final - Revisada
Documento de Arquitetura de Software	Data Versão: 09/08/2020

7.8. Camada de Negócio – Backend JAVA

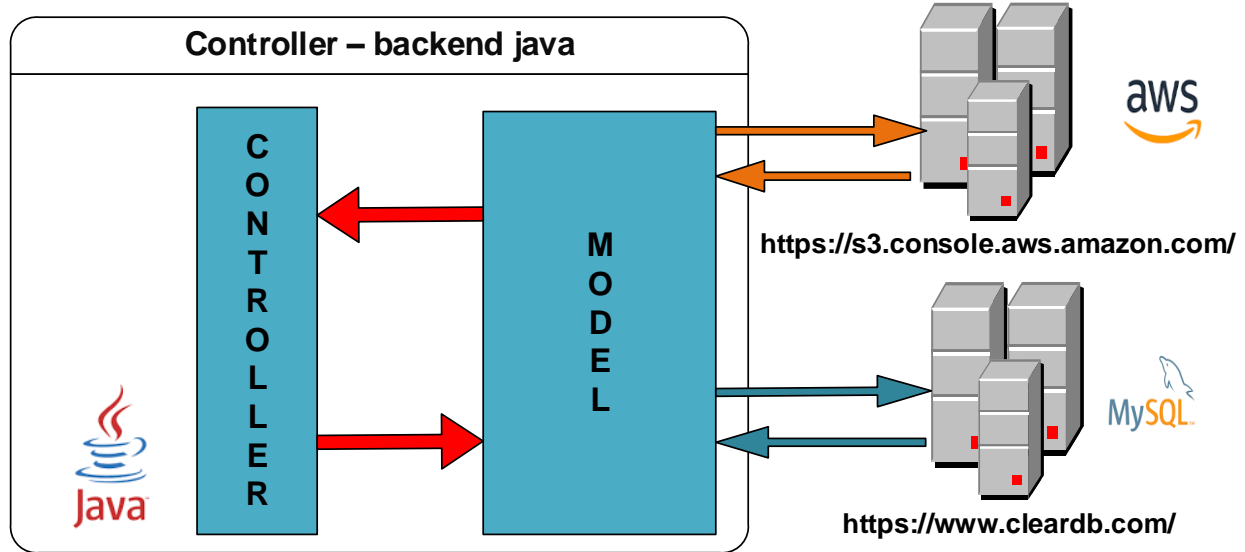


Figura 7.06 – Camada de Controle e Modelo

Nesta camada, temos pacotes de controle e de modelo que contém as classes responsáveis por controlar impor as regras de negócio da aplicação.

- **Pacote resources:** Contém os controles do backend as classes responsáveis por receber as requisições do frontend, direcionar para os serviços e Helper, bem como enviar a resposta do servidor para a camada de apresentação.
- **Pacote helpers:** Contém as classes que auxiliaram na construção dos objetos recebidos do frontend para cadastro e alteração de registros transformando-os de DTOs em entidade de domínio do backend.
- **Pacote dto:** Contém as classes que são utilizadas pelos controladores para converter os JSONs enviados pelo frontend em objetos DTOs que serão tratados pelas helper, bem como construir os objetos de resposta que serão utilizados para converter as entidades de domínio em JSONs.
- **Pacote utilities:** Contém as classes utilitários da API, responsáveis por tarefas especializadas, como formatação de data, documentos, imagens, vídeos, áudios, geração de códigos e matrículas entre outros.

V1LLSEC	Versão: Final - Revisada
Documento de Arquitetura de Software	Data Versão: 09/08/2020

- **Pacote validation:** Contém as classes de validação de dados primários dos usuários garantindo a integridade dos dados que serão armazenados no banco de dados.
- **Pacote exceptions:** Contém as classes que tratam os possíveis erros do sistema.
- **Pacote domain:** Contém as classes de domínio do sistema, bem como o mapeamento relacional que será usado pelo Hibernate para persistência no banco de dados.
- **Pacote enums:** Contém todas as enumerações utilizadas no sistema.
- **Pacote configuration:** Contém as classes de configuração do sistema, são elas: classes de configuração do AWS-S3 da Amazon, classe de configuração do sistema e beans.
- **Pacote repository:** Contém as interfaces de repositório do sistema bem como as regras e métodos que serão usados pelos serviços e demais, para comunicação com as bases de dados.
- **Pacote security:** Contém todas as classes voltadas para a segurança e controle de acesso de usuários, autenticação, autorização, filtros de acesso e construção e validação de JWT-Tokens de segurança.
- **Pacote services:** Contém todas as classes de serviços responsáveis por tratar as requisições bem como preparar as respostas que serão enviadas de volta para as classes de controle.
- **Pacote filters:** Contém as classes que filtram as requisições e identificam os JWT-Tokens e os cabeçalhos das requisições.

V1LLSEC	Versão: Final - Revisada
Documento de Arquitetura de Software	Data Versão: 09/08/2020

7.8.1. Pacote resources.

A figura 7.07 ilustra as principais classes do pacote resources.

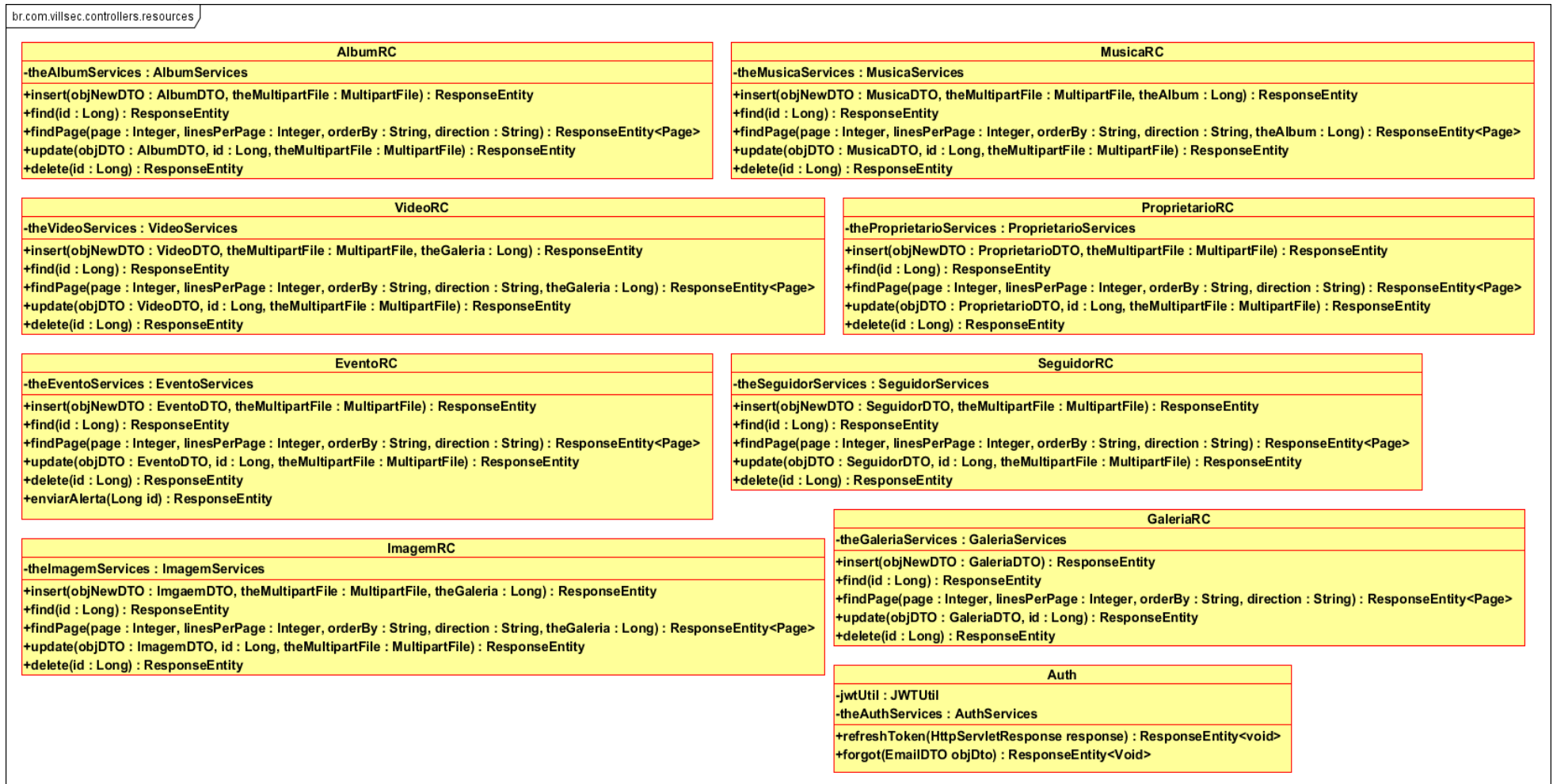


Figura 7.07 – Diagrama de classes do pacote resources.

V1LLSEC	Versão: Final - Revisada
Documento de Arquitetura de Software	Data Versão: 09/08/2020

7.8.2. Pacote helpers.

A figura 7.08 ilustra as principais classes d do pacote helpers.

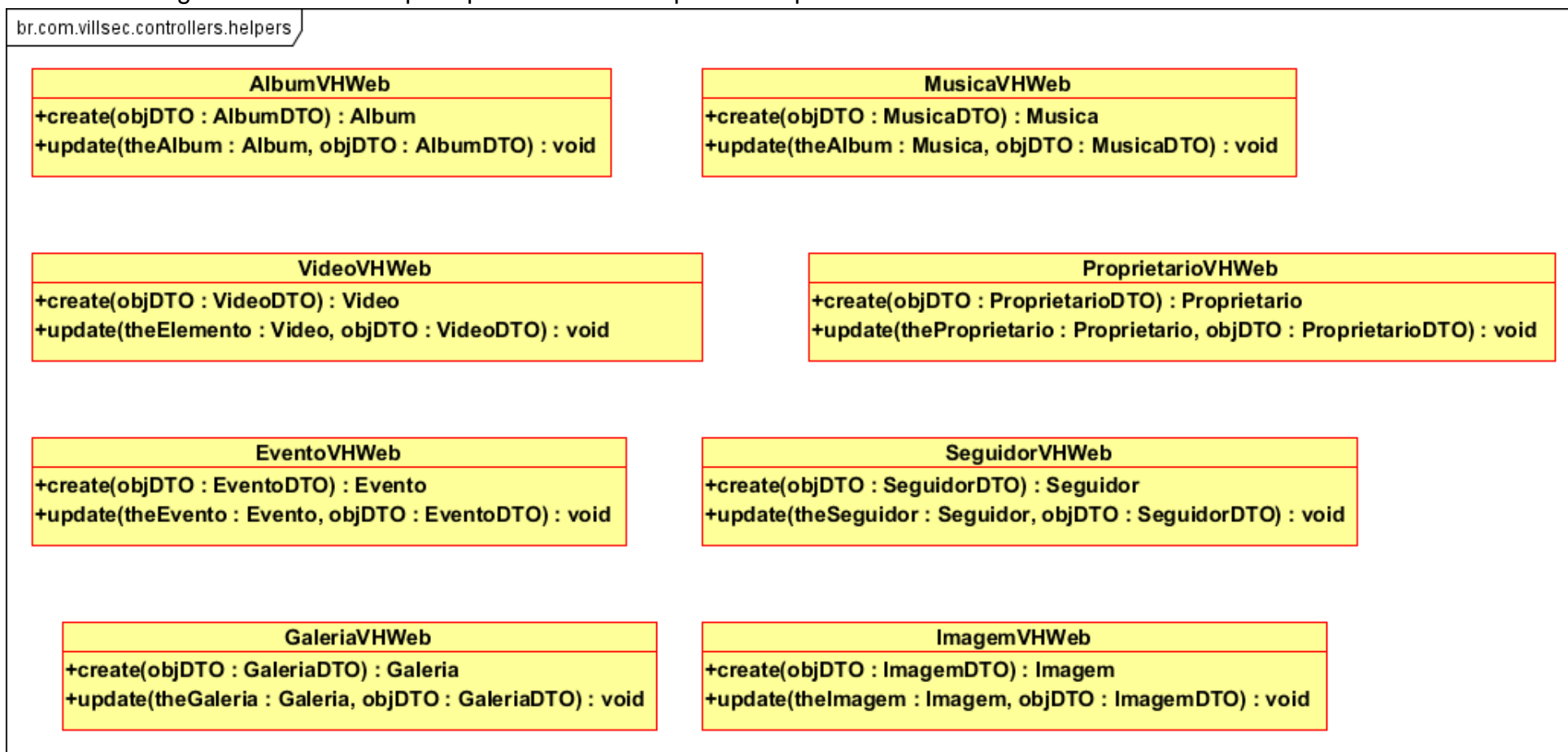


Figura 7.08 – Diagrama de classes do pacote helpers.

V1LLSEC	Versão: Final - Revisada
Documento de Arquitetura de Software	Data Versão: 09/08/2020

7.8.3. Pacote dtos.

A figura 7.09 ilustra as principais classes do pacote dtos.

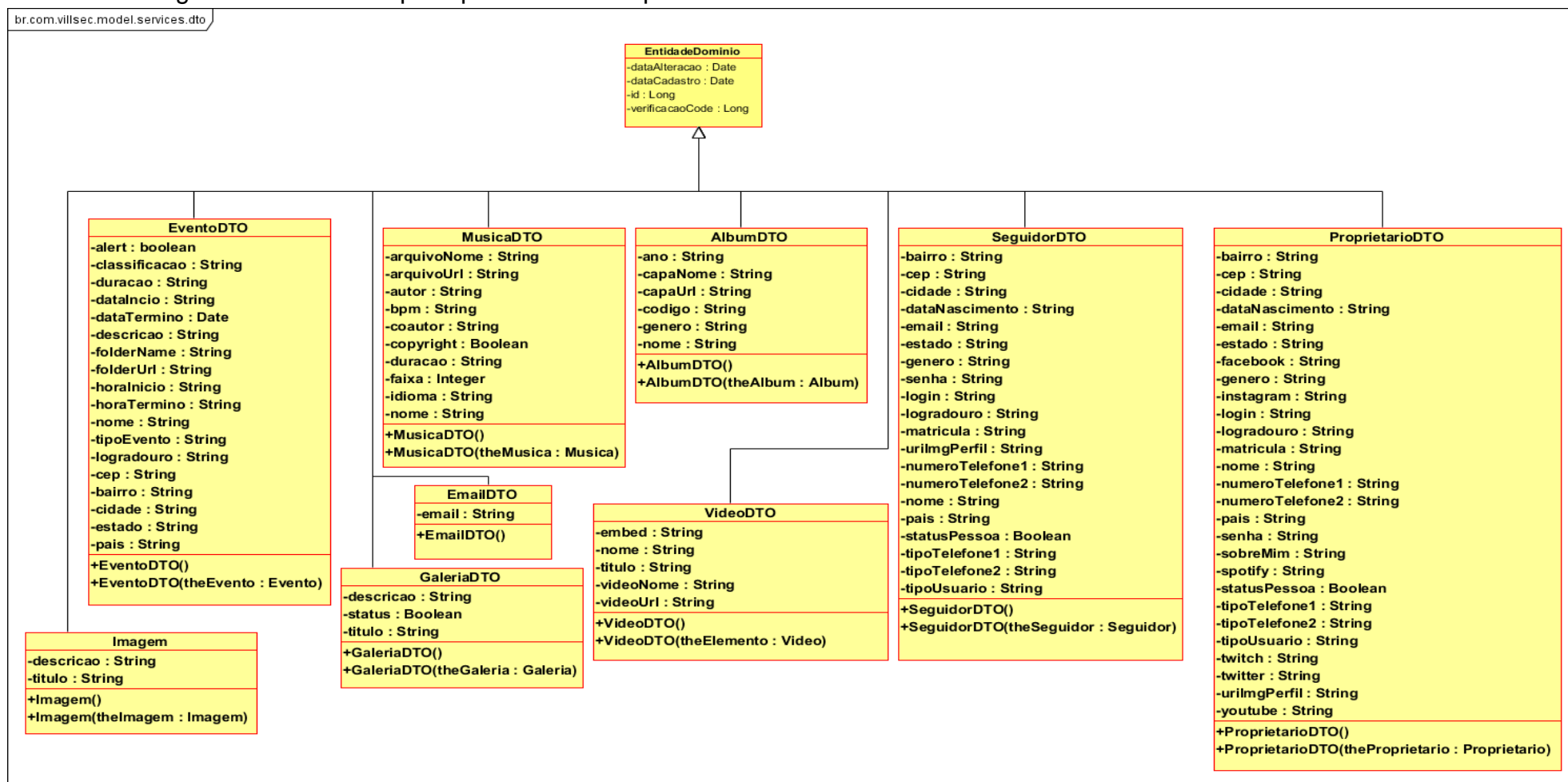


Figura 7.09 – Diagrama de classes do pacote dtos.

V1LLSEC	Versão: Final - Revisada
Documento de Arquitetura de Software	Data Versão: 09/08/2020

7.8.4. Pacote utilities.

A figura 7.10 ilustra as principais classes do pacote utilities.

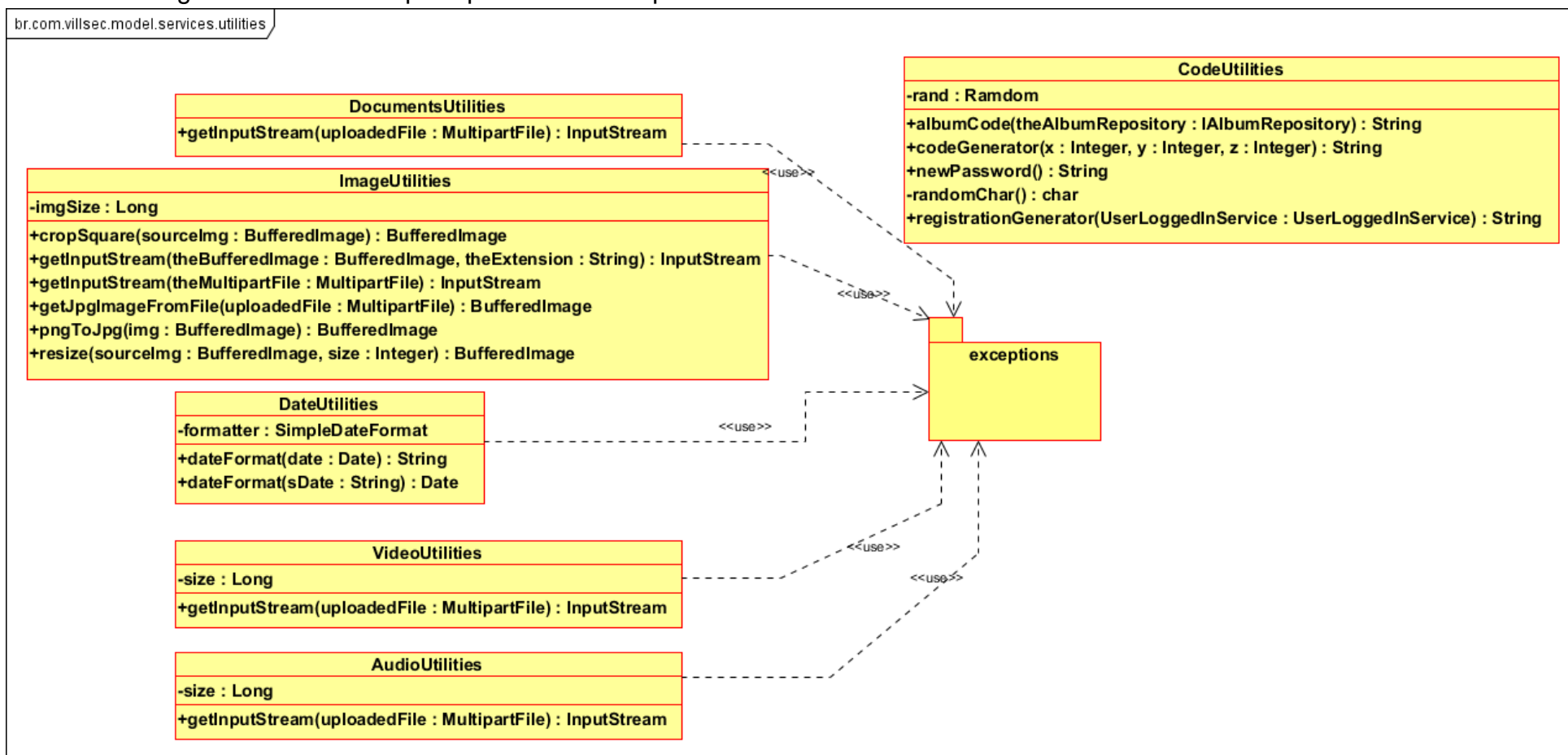


Figura 7.10 – Diagrama de Classe do Pacote utilities.

V1LLSEC	Versão: Final - Revisada
Documento de Arquitetura de Software	Data Versão: 09/08/2020

7.8.5. Pacote validation.

A figura 7.11 ilustra as principais classes do pacote validation.

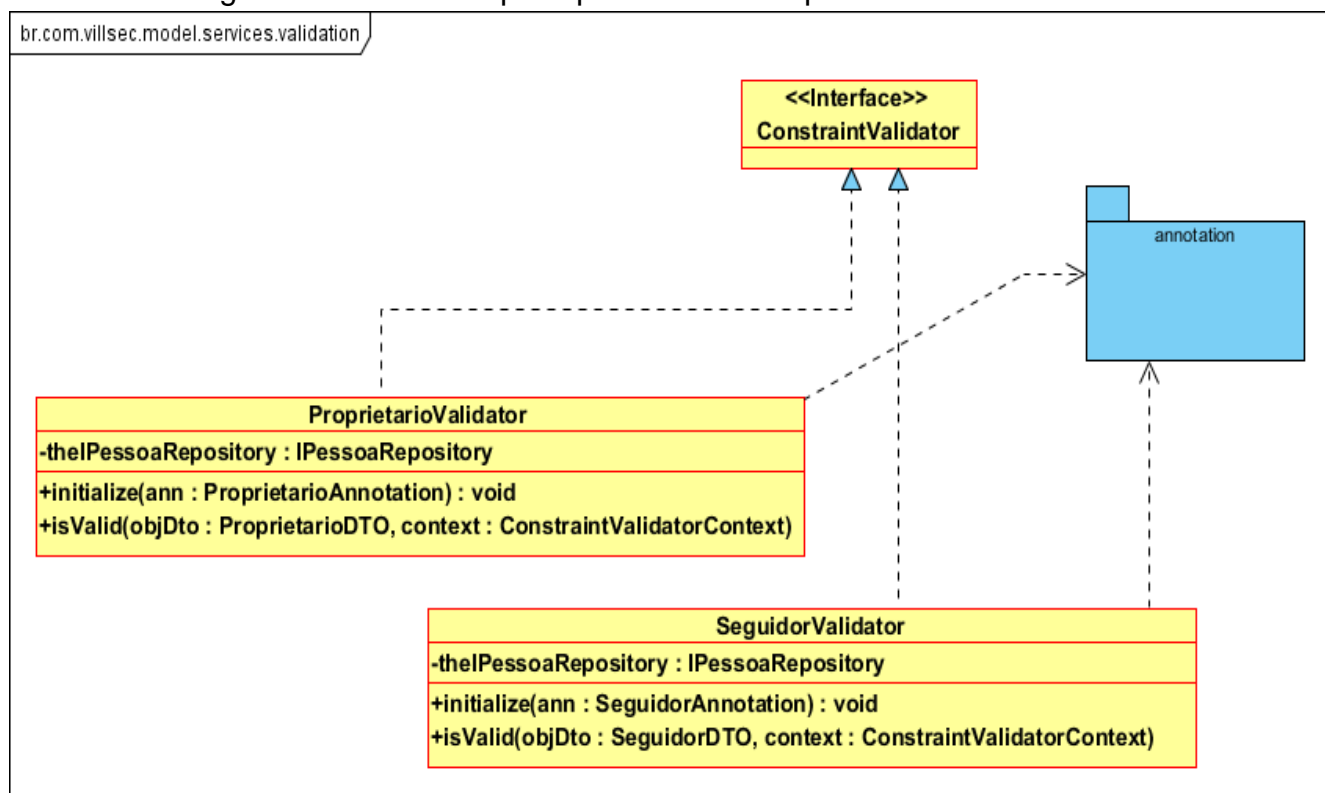


Figura 7.11 – Diagrama de Classe do validation.

7.8.6. Pacote annotation.

A figura 7.12 ilustra as principais classes de annotation.

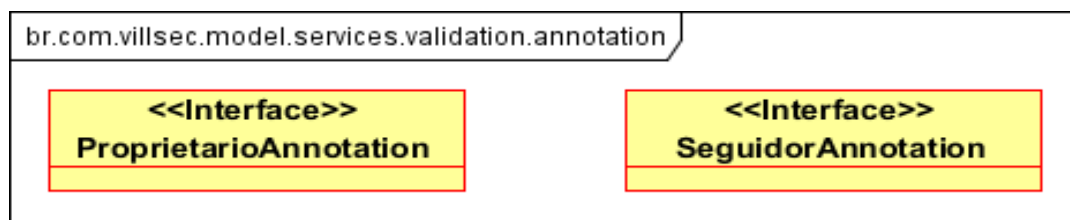


Figura 7.12 – Diagrama de Classe do Pacote annotation.

V1LLSEC	Versão: Final - Revisada
Documento de Arquitetura de Software	Data Versão: 09/08/2020

7.8.7. Pacote exceptions.

A figura 7.13 ilustra as principais classes de exceptions.

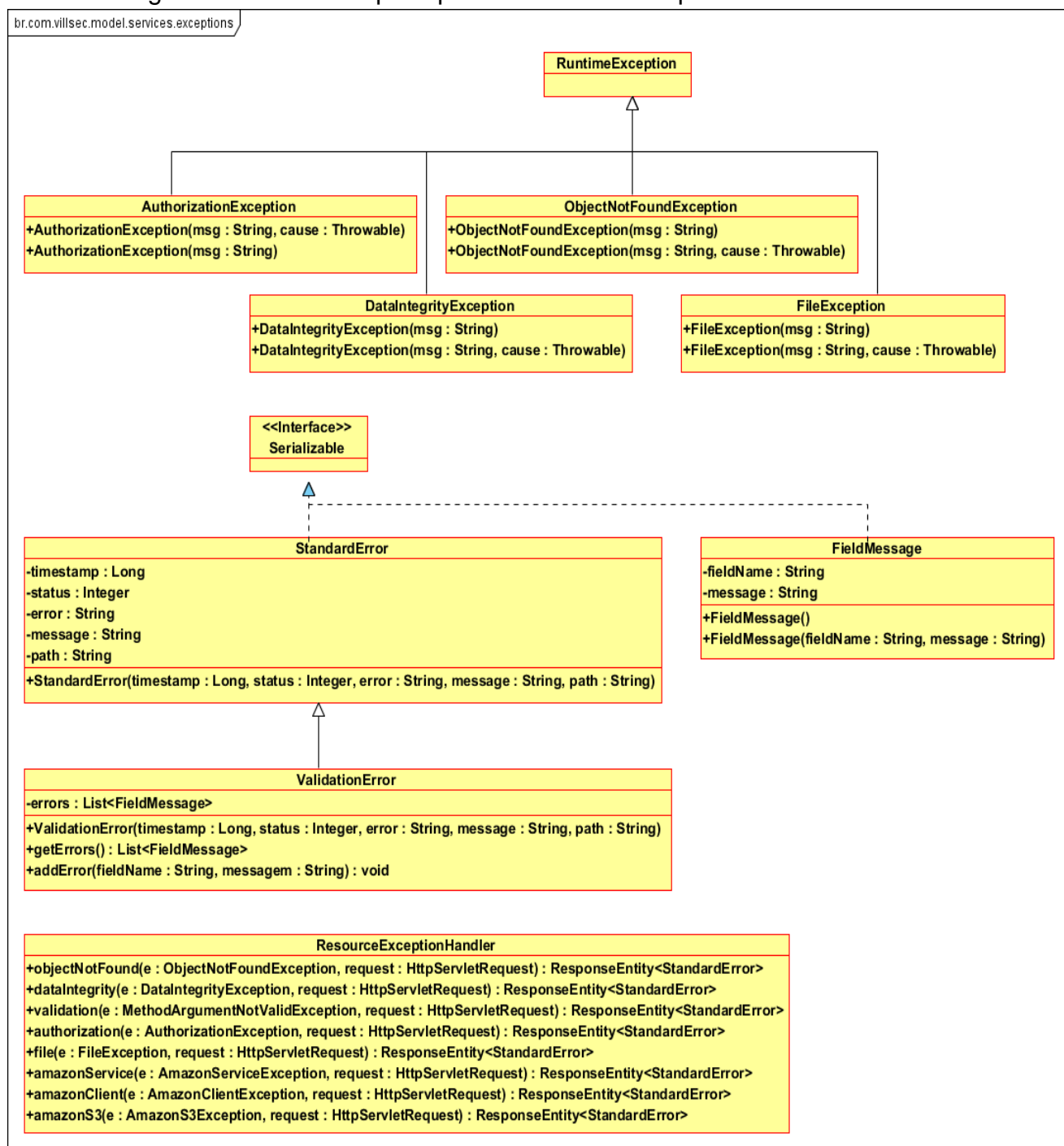


Figura 7.13 – Diagrama de Classe do Pacote exceptions.

V1LLSEC	Versão: Final - Revisada
Documento de Arquitetura de Software	Data Versão: 09/08/2020

7.8.8. Pacote domain.

A figura 7.14 ilustra as principais classes de domain

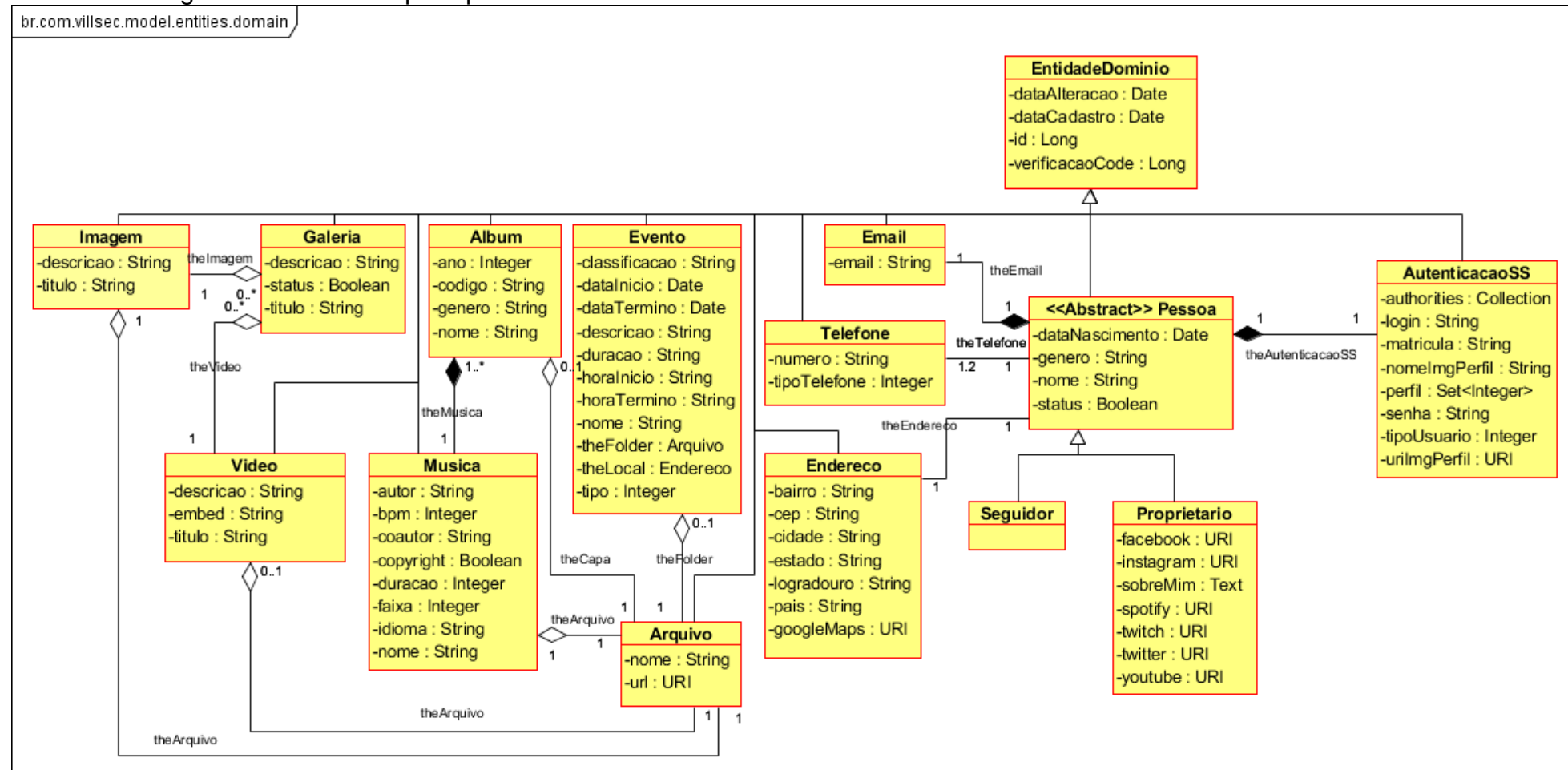


Figura 7.14 – Diagrama de Classe do Pacote domain.

V1LLSEC	Versão: Final - Revisada
Documento de Arquitetura de Software	Data Versão: 09/08/2020

7.8.9. Pacote enums.

A figura 7.15 ilustra as principais classes de enums.

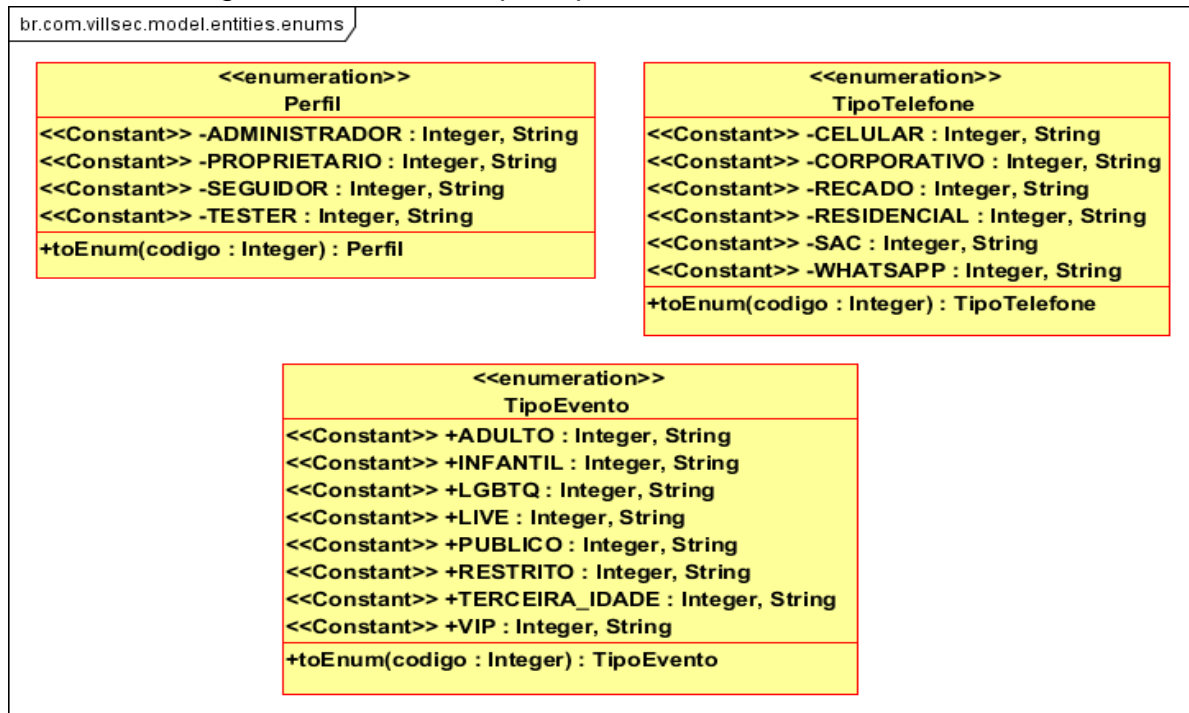


Figura 7.15 – Diagrama de Classe do Pacote annotation.

7.8.10. Pacote configuration.

A figura 7.16 ilustra as principais classes de configuration.

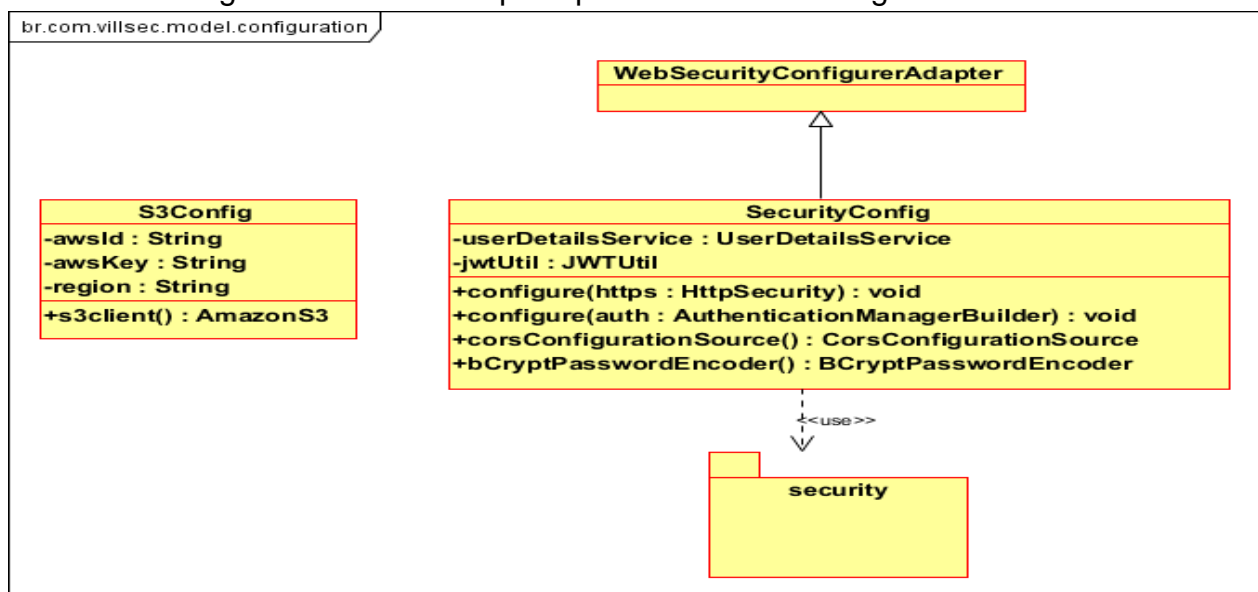


Figura 7.16 – Diagrama de Classe do Pacote configuration.

V1LLSEC	Versão: Final - Revisada
Documento de Arquitetura de Software	Data Versão: 09/08/2020

7.8.11. Pacote repository.

A figura 7.17 ilustra as principais classes de repository.

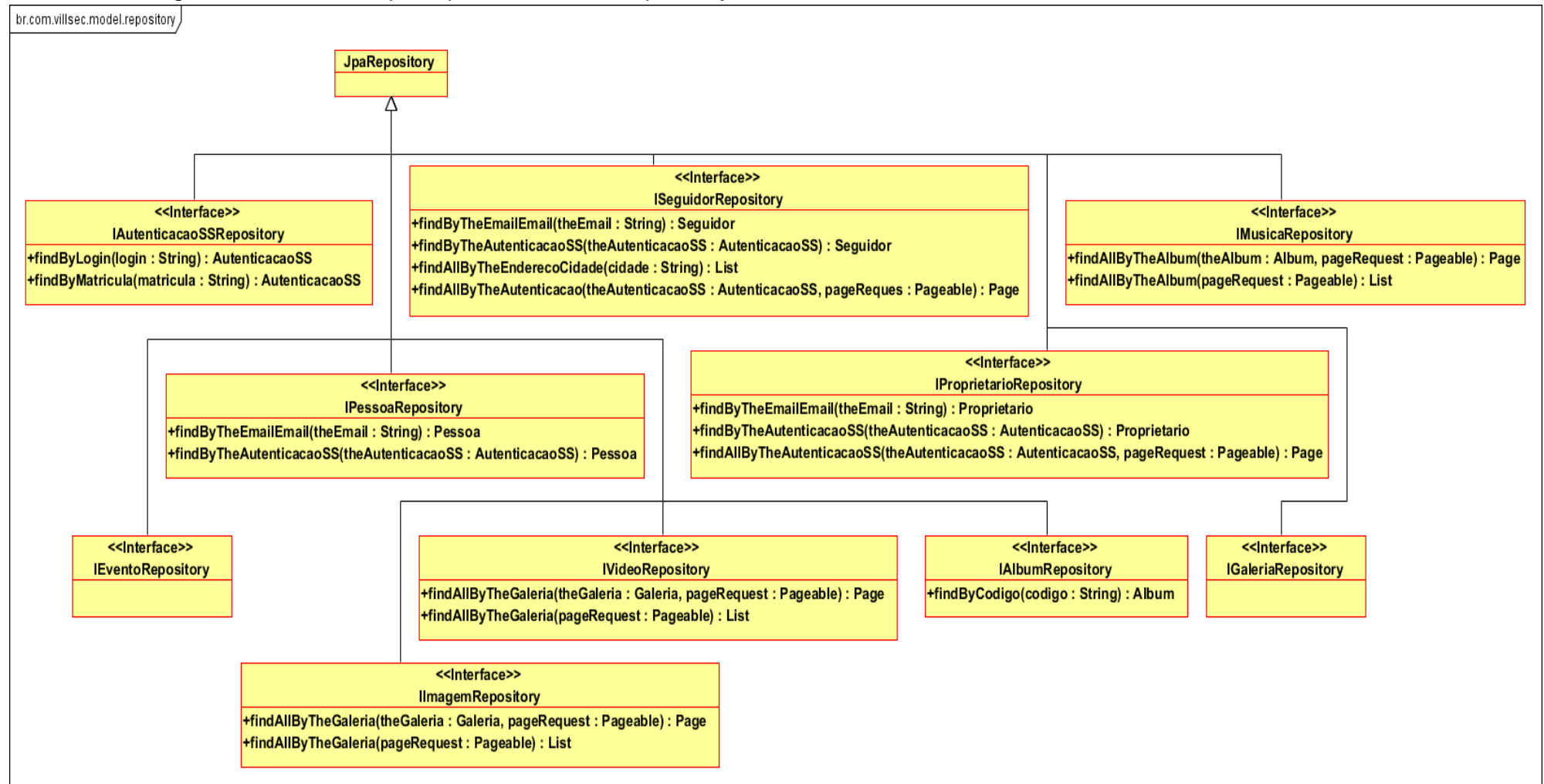


Figura 7.17 – Diagrama de Classe do Pacote repository.

V1LLSEC	Versão: Final - Revisada
Documento de Arquitetura de Software	Data Versão: 09/08/2020

7.8.12. Pacote security.

A figura 7.18 ilustra as principais classes de security.

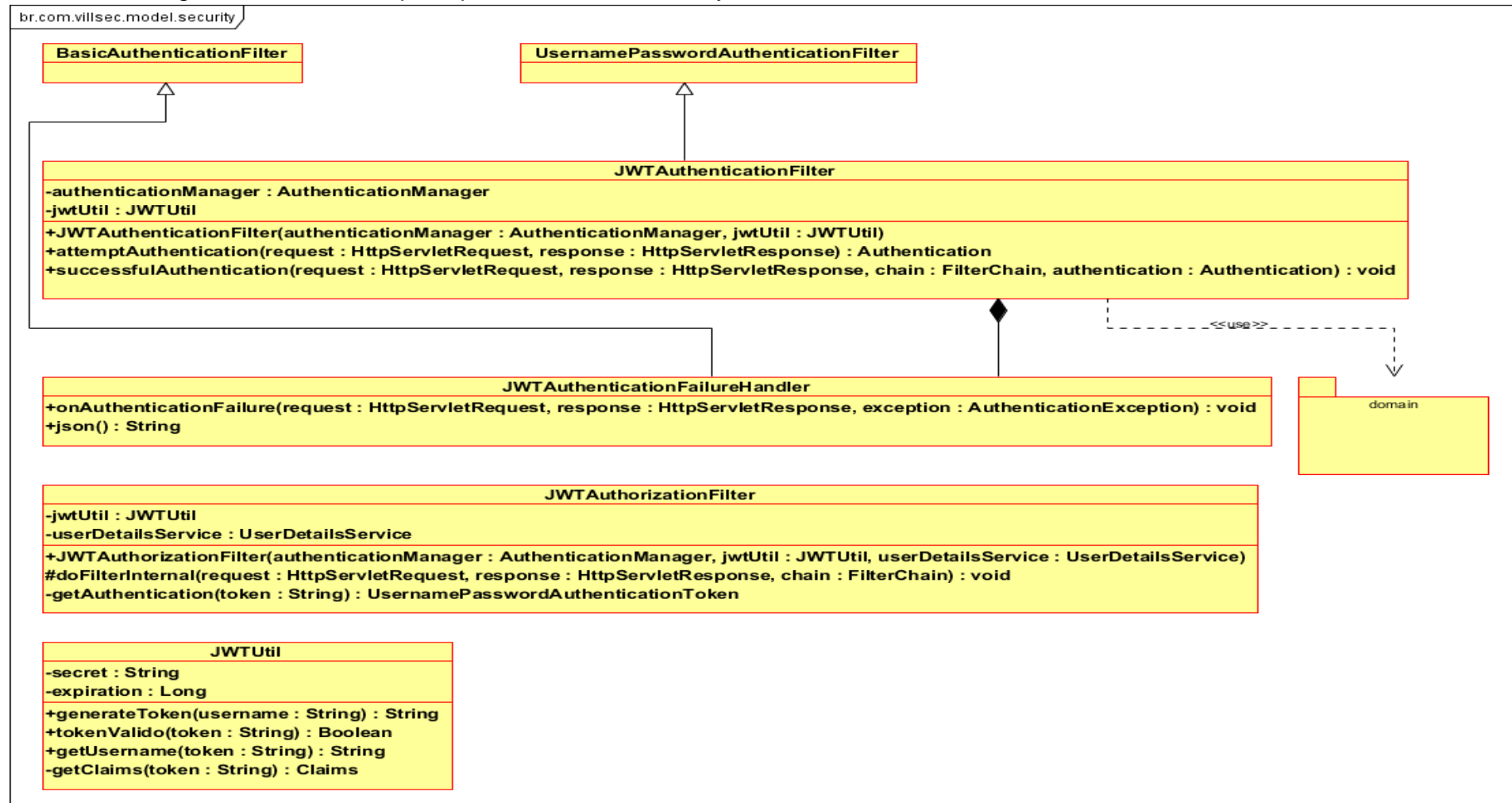


Figura 7.18 – Diagrama de Classe do Pacote security.

V1LLSEC	Versão: Final - Revisada
Documento de Arquitetura de Software	Data Versão: 09/08/2020

7.8.13. Pacote services.

A figura 7.18 ilustra as principais classes de services.

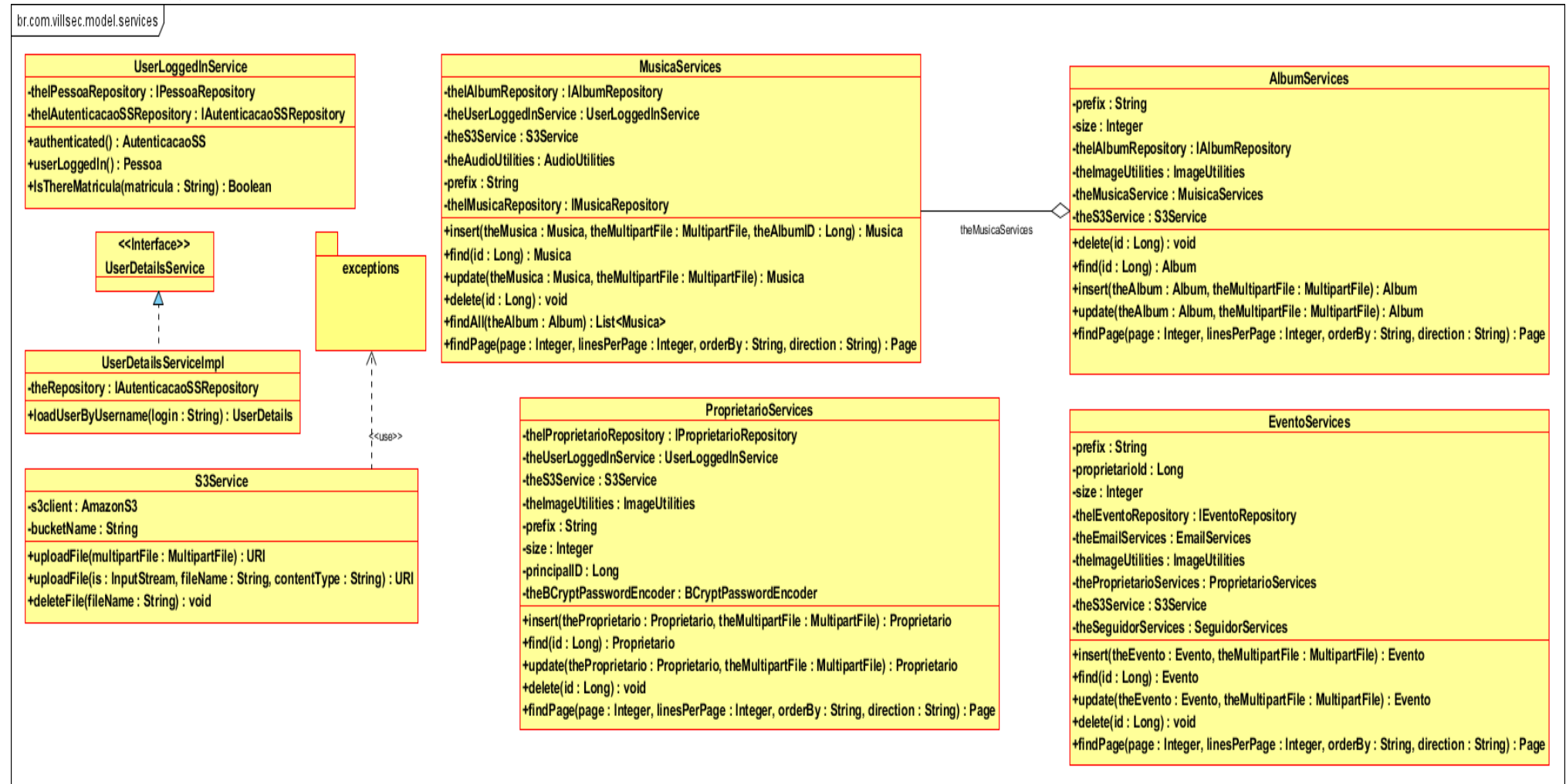


Figura 7.18 – Diagrama de Classe do Pacote services.

V1LLSEC	Versão: Final - Revisada
Documento de Arquitetura de Software	Data Versão: 09/08/2020

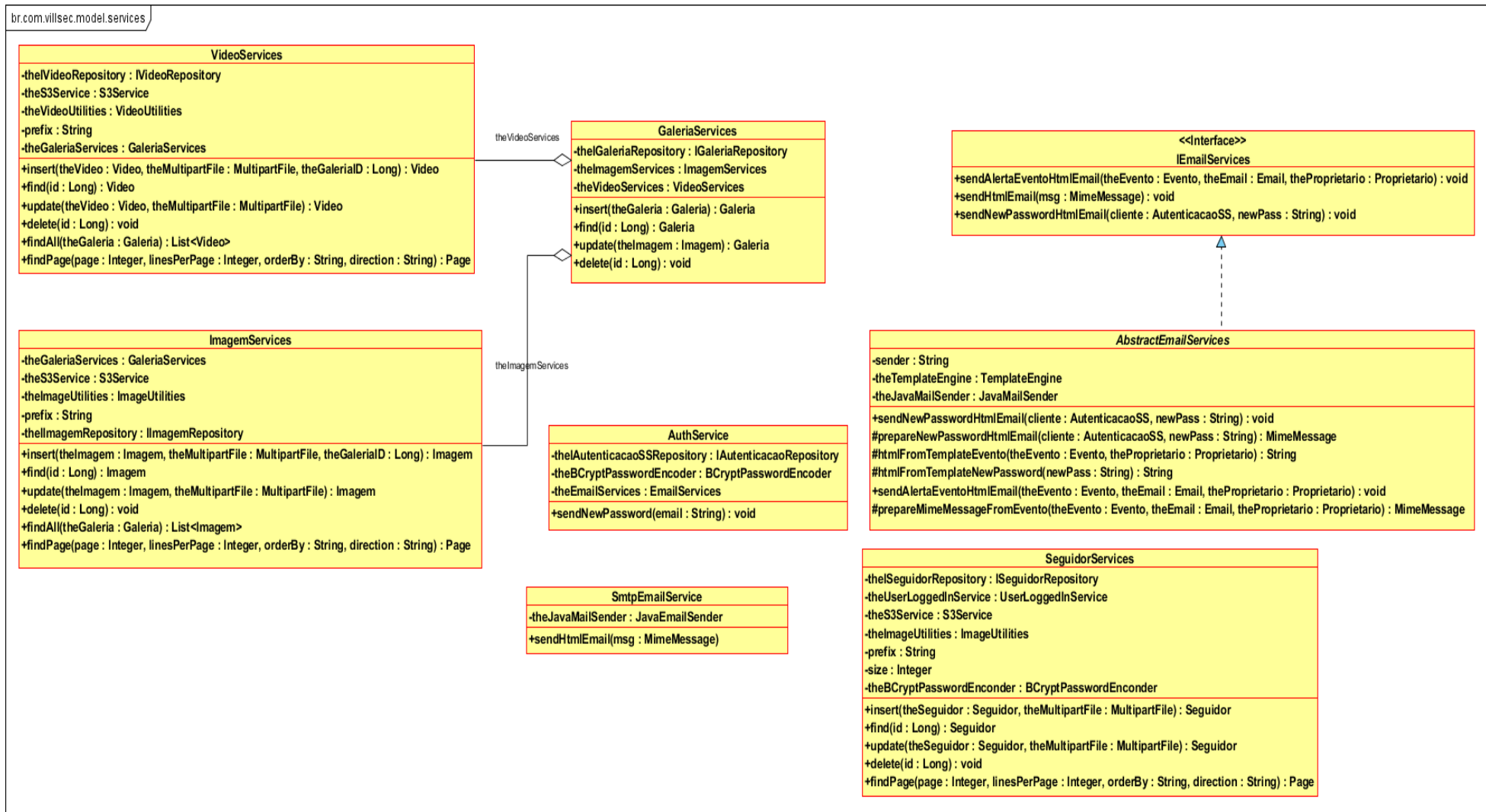


Figura 7.19 – Diagrama de Classe do Pacote services.

V1LLSEC	Versão: Final - Revisada
Documento de Arquitetura de Software	Data Versão: 09/08/2020

7.8.14. Pacote filters.

A figura 7.20 ilustra as principais classes de filters.

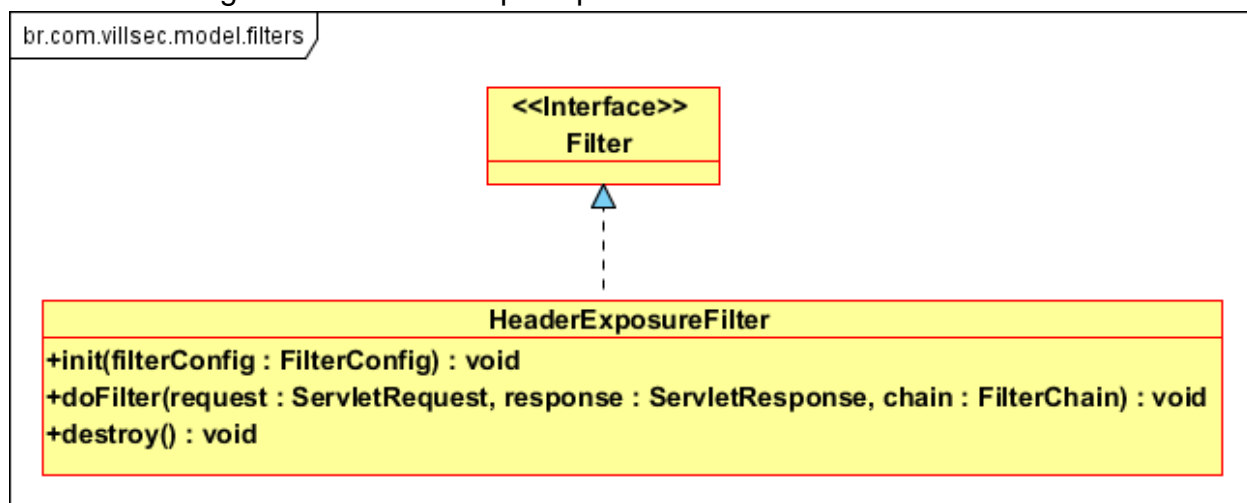


Figura 7.20 – Diagrama de Classe do Pacote repository.

V1LLSEC	Versão: Final - Revisada
Documento de Arquitetura de Software	Data Versão: 09/08/2020

7.9. Realização dos Casos de Uso

7.9.1. Diagrama de Sequência Geral: Inserção de Registro.

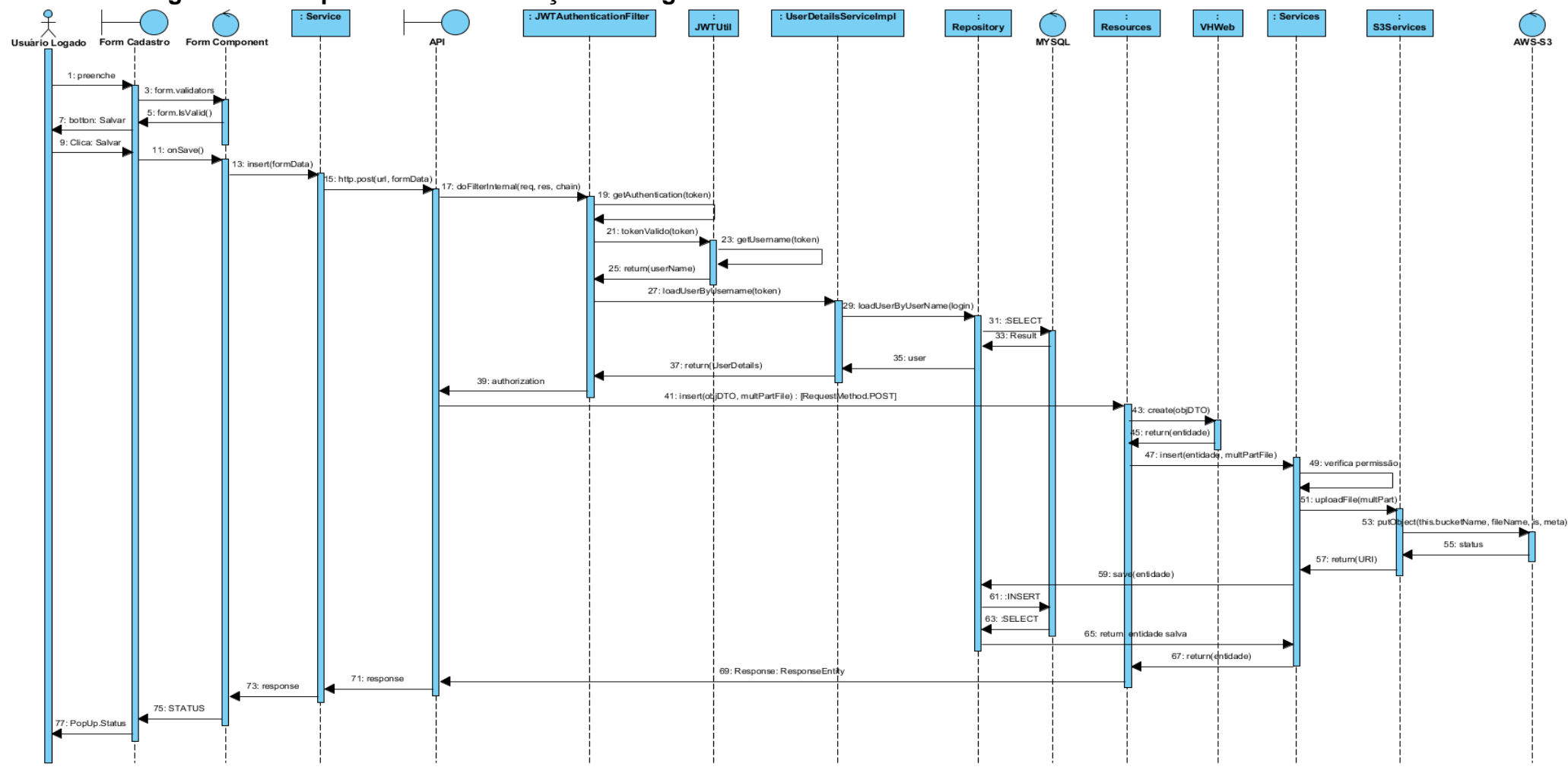


Figura 7.21 – Diagrama de Sequencia Geral: Inserção de Registro.

V1LLSEC	Versão: Final - Revisada
Documento de Arquitetura de Software	Data Versão: 09/08/2020

7.9.2. Diagrama de Sequência Geral: Alteração de Registro.

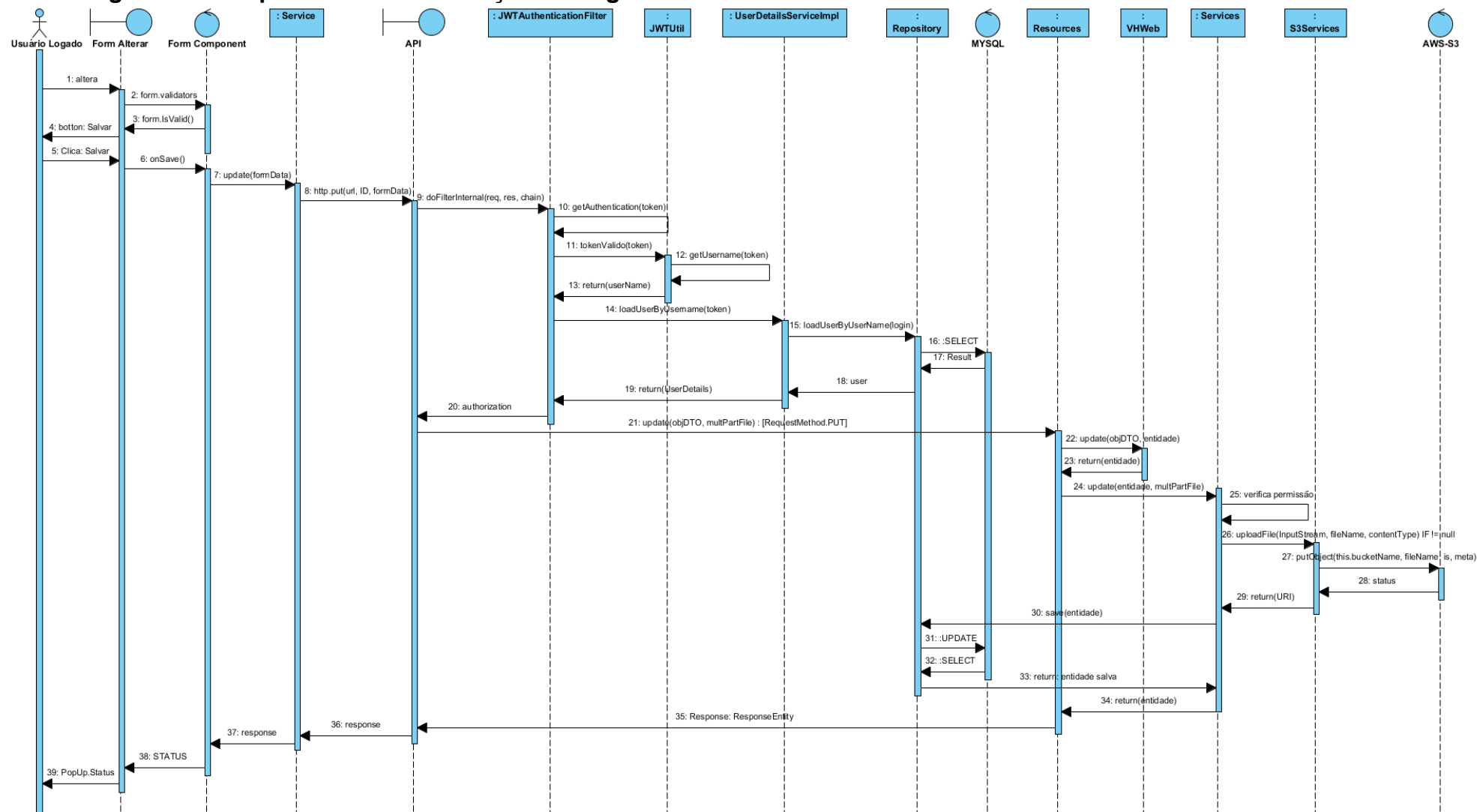


Figura 7.22 – Diagrama de Sequencia Geral: Alteração de Registro.

V1LLSEC	Versão: Final - Revisada
Documento de Arquitetura de Software	Data Versão: 09/08/2020

7.9.3. Diagrama de Sequência Geral: Exclusão de Registro.

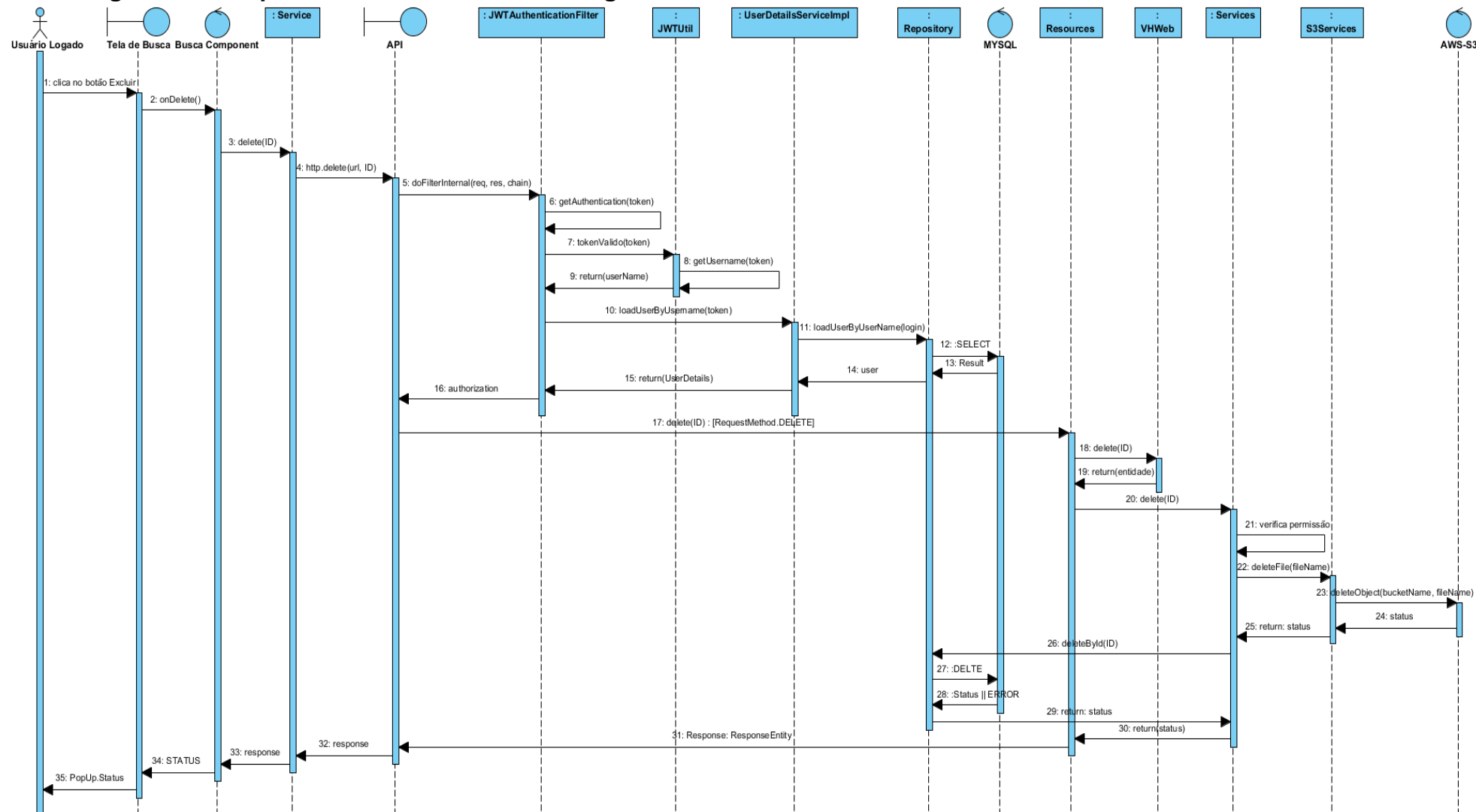


Figura 7.23 – Diagrama de Sequência Geral: Exclusão de Registro.

V1LLSEC	Versão: Final - Revisada
Documento de Arquitetura de Software	Data Versão: 09/08/2020

7.9.4. Diagrama de Sequência Geral: Buscar Registro.

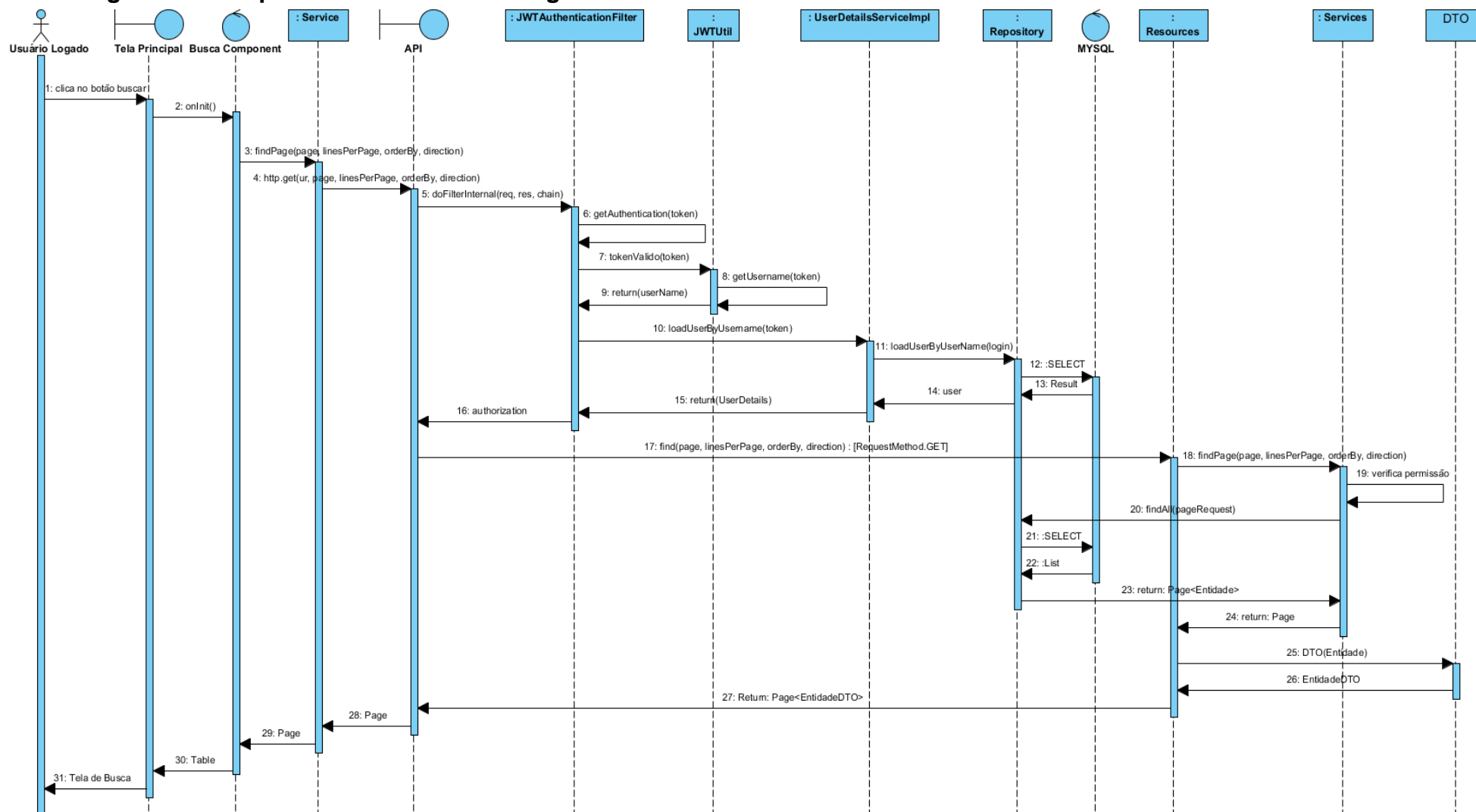


Figura 7.24 – Diagrama de Sequência Geral: Buscar Registro.

V1LLSEC	Versão: Final - Revisada
Documento de Arquitetura de Software	Data Versão: 09/08/2020

7.9.5. Diagrama de Sequência: Login.

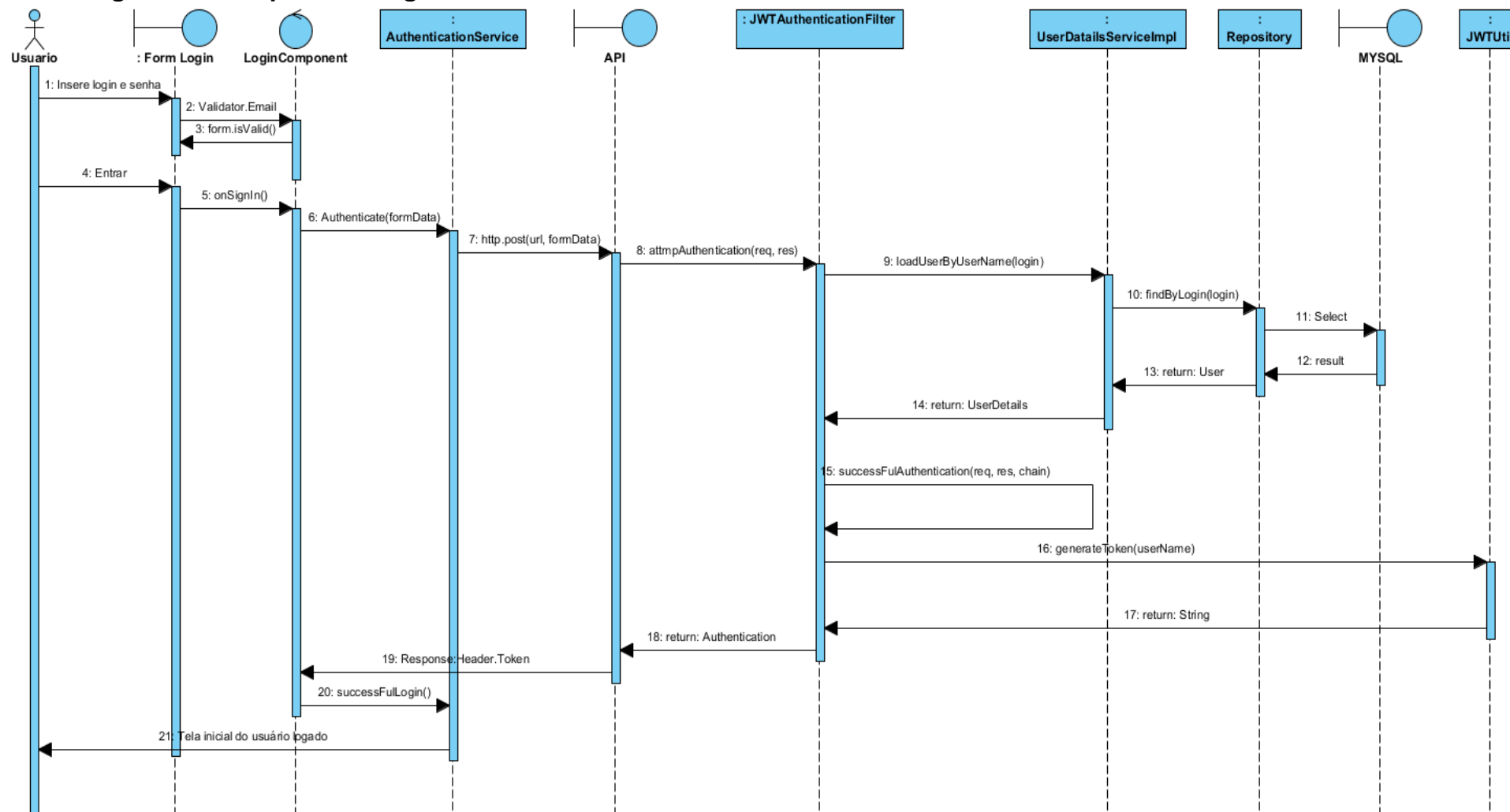


Figura 7.25 – Diagrama de Sequência: Login.

V1LLSEC	Versão: Final - Revisada
Documento de Arquitetura de Software	Data Versão: 09/08/2020

7.10. Visão de Implantação

Esta seção descreve as configurações de rede física (hardware) na qual o V1LLSEC será implantado e executado.

Trata-se de uma visão do Modelo de implantação que, para a configuração em questão, indica os nós físicos (Computadores e dispositivos moveis), que executarão o sistema V1LLSEC, e as respectivas interconexões (barramentos, LAN, WAN, etc). A figura 7.26 ilustra o modelo de implantação para o V1LLSEC.

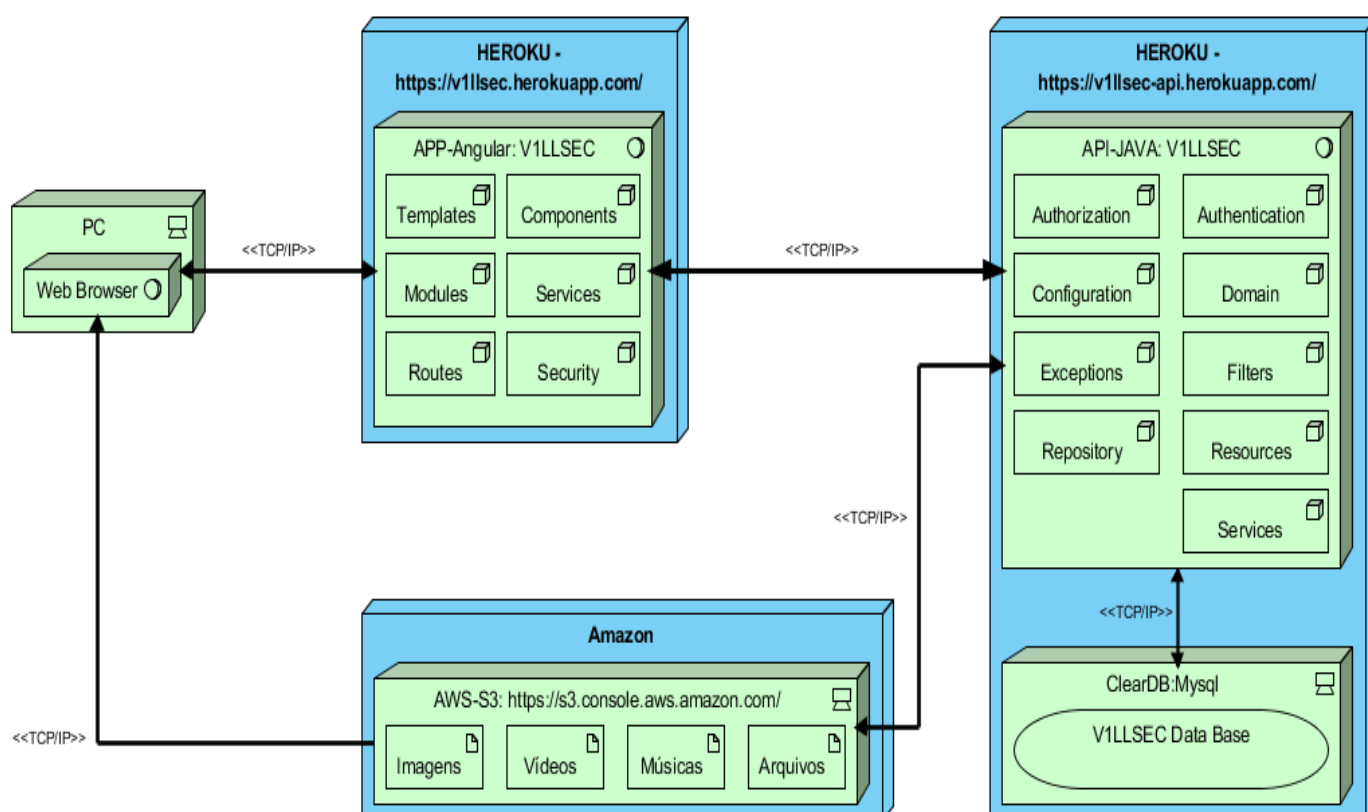


Figura 7.26 – Visão de Implantação do V1LLSEC

7.11. Visão de Implementação

Esta visão descreve a estrutura geral de implementação, a decomposição do software em camadas de implementação.

A estrutura geral de implementação para o V1LLSEC é baseada na estrutura da Visão Lógica, assim, não há necessidade de detalhar os diagramas de camadas e pacotes de implementação, uma vez que são fortemente baseados naqueles desenvolvidos para Visão Lógica.

V1LLSEC	Versão: Final - Revisada
Documento de Arquitetura de Software	Data Versão: 09/08/2020

7.12. Visão de Dados

O mecanismo de Persistência utilizado no sistema V1LLSEC utiliza-se o banco de dados relacional Oracle Mysql juntamente com o framework para mapeamento objeto-relacional, Hibernate.

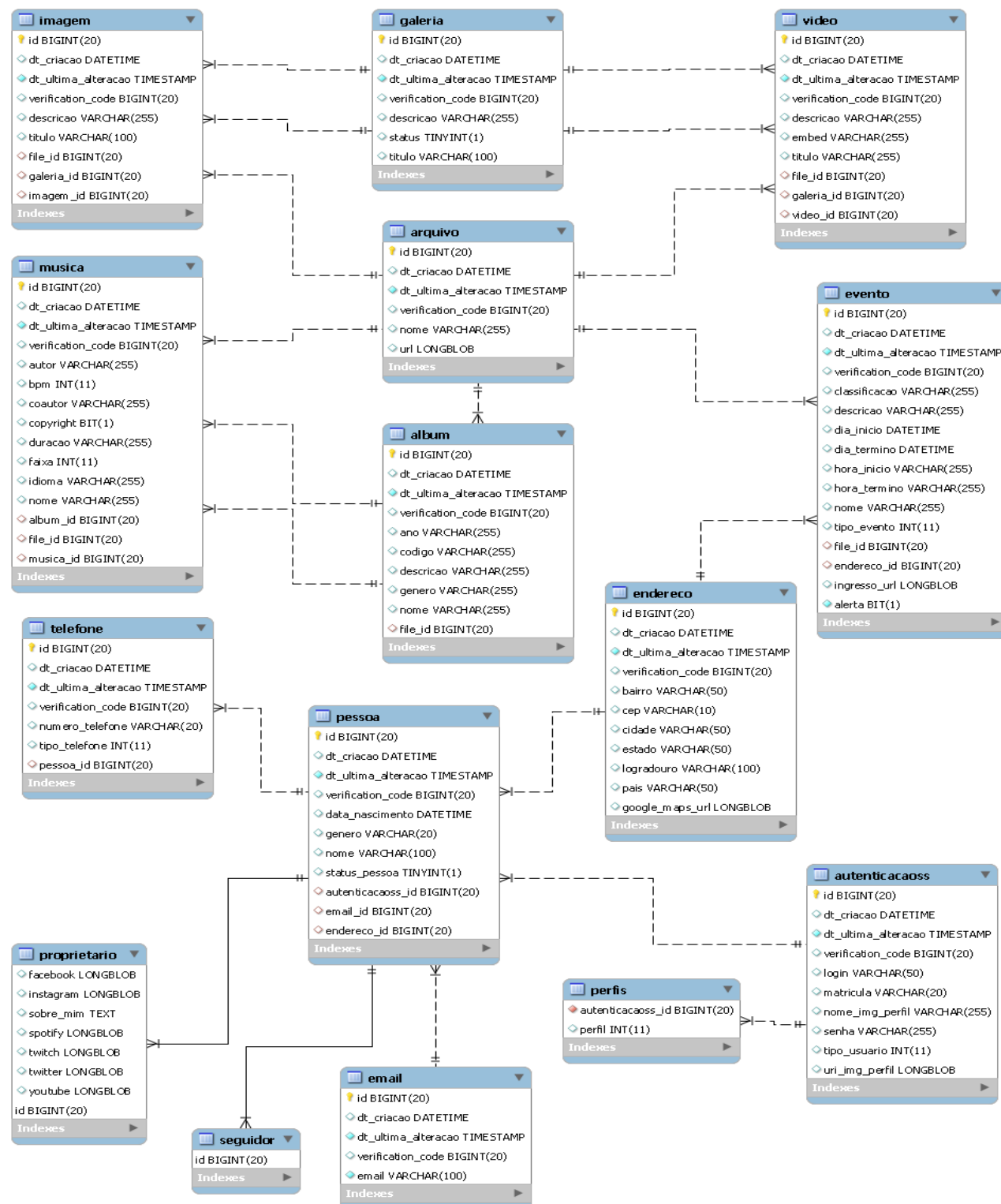


Figura 7.27 – Modelo Lógico.

V1LLSEC	Versão: Final - Revisada
Documento de Arquitetura de Software	Data Versão: 09/08/2020

7.13. Tamanho e Desempenho

O V1LLSEC será disponibilizado para uso dos clientes para armazenagem não só de dados cadastrados através do sistema, como também será possível inserir arquivos do tipo vídeo (.avi, .mp4), áudio (mp3) e Imagens (.jpg, .png), entre outros, fazendo com que o mesmo precise de uma base de dados mais robusta que é o caso do MySQL (Heroku-ClearDB) para os dados e AWS-S3 Amazon para os arquivos.

O sistema como um todo é dividido em várias camadas em que cada uma dessas camadas estão em servidores independente uma da outra na nuvem, aplicação Angular (Frontend) alocada no Heroku [<https://v1llsec.herokuapp.com/>], banco de dados mysql ClearDB – Heroku [heroku_41b5222c81cf878], API-Java(Backend) alocada no Heroku [<https://v1llsec-api.herokuapp.com/>], repositório de arquivos AWS-S3 Amazon.