ES3 – Engenharia de Software III

Prof. Antonio Sergio Bernardo

Manhã

PROJETO DE SOFTWARE

TÍTULO DO TRABALHO

Grupo AViS

Pedro Bernardo de SOUSA 0030481711006

Weuller Júnior Souza Bessa 0030481621040

Vítor Andrade Marques da Silva 0030481511040

Sorocaba

Primeiro Semestre de 2019

RESUMO

A Alloy City Linguistics desenvolveu e mantém uma plataforma de ensino e aprendizado de francês baseada em tecnologias web. O objetivo do grupo AViS até o final do curso ADS dos participantes é desenvolver um cliente para essa plataforma existente. O cliente vai se chamar AViS - Alloy Virtual Space, e vai se apoiar na RESTful API disponibilizada pela Alloy para o projeto. O cliente AViS será capaz de simular um ambiente 3D onde professor e aluno poderão interagir de maneira mais próxima de interações humanas presenciais, isto é, com a impressão de compartilhar o lugar, e não apenas o momento.

**Palavras-chave:** AViS. *Plataforma*. EAD. Ambiente.Tri-dimensional.

LISTA DE figuras

Figura 1 Símbolos xx

Figura 2 Ciclo de Vida xx

Figura 3 Processos xx

LISTA DE TABELAS

Tabela 1 Padrão xx

Tabela 2 Ciclo xx

Tabela 3 Gupos xx

SUMÁRIO

**1 INTRODUÇÃO** 5

**2 REQUISITOS NÃO FUNCIONAIS** xx

**3 REQUISITOS FUNCIONAIS** xx

**3.1** DIAGRAMA DE CASOS DE USO..................................................................... xx

**3.1** DESCRIÇÃO DOS CASOS DE USO.............................................................. xx

4 ARQUITETURA DO SISTEMA .......................................................................... xx

**4.1** DIAGRAMA DE PACOTES ............................................................................. xx

**4.2** FERRAMENTAS E TECNOLOGIA ...................................................................xx

5 DIAGRAMAS DE SEQUÊNCIA ........................................................................ xx

6 DIAGRAMAS DE ESTADOS ............................................................................ xx

7 OUTROS DIAGRAMAS ...................................................................................... xx

8 TELAS DO PROTÓTIPO .................................................................................. xx

**9 CONSIDERAÇÕES FINAIS**

**BIBLIOGRAFIA** ....................................................................................................XX

1 INTRODUÇÃO

Com o propósito de proporcionar uma experiência de aprendizado à distância mais abrangente, a plataforma do projeto AViS, busca fornecer aos usuários ferramentas em que os mais diversos aspectos da comunicação possam ser explorados durante a experiência de uso.

A comunicação humana natural se dá, não apenas através das palavras ditas e escritas. Aspectos da linguagem corporal e da entonação complementam este processo entre emissor e receptor. Assim, oferecer uma ferramenta capaz de abordar, da forma mais abrangente possível tais aspectos no processo de ensino à distância, proverá aos usuários, um significativo avanço na qualidade dos estímulos sensoriais, proporcionando que mais das informações oferecidas possam ser absorvidas.

Dois objetivos iniciais do projeto são, oferecer aos alunos um ambiente virtual em que a experiência de aprendizado aconteça como se aluno e professor partilhassem da mesma localização e, oferecer aos professores a possibilidade de receber dos alunos um volume maior de informações para avaliá-los e assim poder aprimorar suas aulas e conteúdo.

Como possibilidades a serem implementadas a longo prazo, o projeto AViS, poderá se tornar uma plataforma unificada capaz de oferecer suporte ao ensino de diversos idiomas e de forma universal, compatível com todos os principais sistemas operacionais.

**2 REQUISITOS NÃO FUNCIONAIS**

**3 REQUISITOS FUNCIONAIS**

**3.1** DIAGRAMA DE CASOS DE USO



**3.1** DESCRIÇÃO DOS CASOS DE USO

3.1.1 Comunicação

3.1.1.1 **Ouvir outros usuários presentes**

Usuário deve poder ouvir outros usuários que estiverem na mesma sala.

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | FR1: OUVIR OUTROS USUÁRIOS PRESENTES | |
| **Ator Principal** | INSTÂNCIA LOCAL | |
| **Atores Secundários** | INSTÂNCIAS REMOTAS | |
| **Pré-Condição** | 1 – O aluno está dentro da mesma sala de aula virtual que o usuário de origem do som;  2 – O host da instância local está configurado para reproduzir som;  3 – O host da instância remota está configurado para capturar som;  4 – O recurso “*mute audio*” da instância local está desabilitado;  5 – O recurso “*mute mic*” da instância remota está desabilitado. | |
| **Pós-Condição** | O som capturado pelo usuário remoto foi reproduzido claramente pelo equipamento do ALUNO. | |
| **Ações da Instância Local** | | **Ações da Instância Remota** |
|  | | Feed de áudio originado na instância remota é capturado |
|  | | Feed de áudio é compactado |
|  | | Feed de áudio é transmitido diretamente para a instância local via UDP |
| Feed de áudio é recebido | |  |
| Feed de áudio é reproduzido pelo sistema de som | |  |

3.1.1.2 **Ver outros usuários presentes**

Usuário deve poder ver outros usuários que estão na mesma sala. A sequência de ações do sistema abaixo deve ser executada para cada frame capturado pela webcam, numa frequência mínima de 24 hertz.

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | FR2: VER OUTROS USUÁRIOS PRESENTES | |
| **Ator Principal** | INSTÂNCIA LOCAL | |
| **Ator Secundário** | INSTÂNCIA REMOTA | |
| **Pré-Condição** | 1 – O aluno está dentro da mesma sala de aula virtual que o usuário a ser visto;  2 – O host da instância remota está configurado para capturar vídeo. | |
| **Pós-Condição** | 1 – O avatar do usuário remoto tem sua posição no ambiente virtual da instância local sincronizada com a posição escolhida no ambiente virtual da instância remota;  2 – O rosto do avatar do usuário remoto na instância local corresponde à expressão facial do usuário remoto. | |
| **Ações do Ator Principal** | | **Ações do Ator Secundário** |
|  | | Frame capturado pela webcam |
|  | | Rosto do usuário remoto é identificado no frame |
|  | | Frame é reduzido ao rosto (crop) |
|  | | Frame é compactado (ffmpeg) |
|  | | Mapa de bits do rosto do avatar local é atualizado com o frame |
|  | | Frame é enviado diretamente ao peer via UDP |
| Frame é recebido | |  |
| Mapa de bits do rosto do avatar do usuário remoto é atualizado com o frame recebido | |  |

3.1.1.3 **Falar**

Usuário deve poder transmitir discurso pelo sistema. Seu microfone capta o áudio, o sistema comprime o fluxo e o envia diretamente (P2P) aos usuários conectados à mesma sala.

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | FR3: FALAR | |
| **Ator Principal** | INSTÂNCIA LOCAL | |
| **Atores Secundários** | TODAS INSTÂNCIAS REMOTAS NA MESMA SALA | |
| **Pré-Condição** | 1 – O host da instância local está configurado para capturar vídeo;  2 – O recurso “*mute mic*” na instância local não está ativado. | |
| **Pós-Condição** | 1 – Todos os usuários presentes na mesma sala virtual que não tiverem o recurso “*mute audio*” ativado ouvem o áudio capturado pelo host local. | |
| **Ações do Ator Principal** | | **Ações do Atores Secundários** |
| Áudio é capturado pelo microfone do host local | |  |
| Feed de áudio é compactado | |  |
| Feed de áudio é enviado para todos os hosts remoto da mesma sala via UDP | |  |
|  | | Feed de áudio é recebido |
|  | | Feed de áudio é reproduzido pelo host simulando direção e intensidade da fonte de som com base na posição do avatar de origem |

3.1.2 Controle

3.1.2.1 **Conceder Pontos de Experiência**

O Professor deve poder selecionar um Aluno, digitar um número inteiro dentro de uma determinada faixa, a fim de dar pontos de experiência ao Aluno. A informação deve ser enviada pelo sistema tanto para os demais Usuários presentes na sala quanto ao Servidor, para que os pontos de experiência sejam mostrados nas interfaces de todos os Usuários da sala e sejam gravados na base de dados, no servidor.

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | FR4: CONCEDER PONTOS DE EXPERIÊNCIA | |
| **Ator Principal** | PROFESSOR | |
| **Ator Secundário** | ALUNO | |
| **Pré-Condição** | 1 – O aluno está dentro da mesma sala de aula virtual que o professor | |
| **Pós-Condição** | 1 – XP do aluno é atualizado na base de dados  2 – XP do aluno é atualizado no avatar do aluno em cada instância conectada à sala | |
| **Ações do Ator** | | **Ações do Sistema** |
| Professor aponta para o avatar do aluno e executa a função “*action*” | |  |
|  | | Aluno é selecionado |
|  | | Menu contextual é apresentado |
| Professor seleciona a caixa de texto “*give XPs*” | |  |
| Professor digita um valor numérico | |  |
| Professor pressiona o botão “*send*” | |  |
|  | | O valor é enviado à ponta de API /student/give-xp/ |
|  | | O valor na base de dados é atualizado |
|  | | O valor é enviado pelo servidor a todos os usuários conectados à sala via WebSocket |
|  | | O avatar do usuário beneficiado é atualizado em cada instância conectada à sala |

3.1.2.2 **Mutar Aluno**

O Professor deve poder selecionar um aluno e, em um clique, desativar a transmissão de áudio daquele aluno. A informação deve ser enviada para o aluno selecionado apenas. A instância AViS que recebe a instrução mute\_lock deve desativar o microfone e desativar o botão mute/unmute do microfone. Assim, o microfone só será reativado quando uma instrução mute\_unlock, vinda do professor, for recebida.

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | FR5: MUTAR ALUNO | |
| **Ator Principal** | PROFESSOR | |
| **Ator Secundário** | ALUNO | |
| **Pré-Condição** | 1 – O aluno está na mesma sala de aula virtual que o professor  2 – O recurso “*mute mic*” na instância do aluno não está bloqueado | |
| **Pós-Condição** | 1 – O microfone no host do aluno está desativado;  2 – O recurso “*mute mic*” na instância do aluno está ativado;  3 – O recurso “*mute mic*” na instância do aluno está bloqueado (não está sob controle do aluno). | |
| **Ações do Ator** | | **Ações do Sistema** |
| Professor aponta para o avatar do aluno e executa a função “*action*” | |  |
|  | | Aluno é selecionado |
|  | | Menu contextual é apresentado |
| Professor aciona o switch “*mute*” | |  |
|  | | A instância do professor indica o início da operação bloqueando o switch “*mute*” |
|  | | A instância do professor envia o comando “*mute\_lock*” ao aluno selecionado via TCP |
|  | | A instância do aluno recebe o comando “*mute\_lock*” |
|  | | A instância do aluno desativa o microfone do host |
|  | | A instância do aluno desativa o módulo de compactação de som |
|  | | A instância do aluno envia a mensagem “*ack\_mute\_lock*” ao professor via TPC |
|  | | A instância do professor recebe a mensagem “*ack\_mute\_lock*” |
|  | | A instância do professor indica o sucesso da operação desbloquenado o switch “*mute*” |

3.1.2.3 **Desmutar Aluno**

O Professor deve poder selecionar um aluno que foi mutado e, em um clique, reativar a transmissão de áudio daquele aluno. A informação deve ser enviada para o aluno selecionado apenas. A instância AViS que recebe a instrução mute\_unlock deve reativar o microfone e o botão mute/unmute do microfone.

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | FR6: DESMUTAR ALUNO | |
| **Ator Principal** | PROFESSOR | |
| **Ator Secundário** | ALUNO | |
| **Pré-Condição** | 1 – O aluno está na mesma sala de aula virtual que o professor  2 – O recurso “*mute mic*” na instância do aluno está bloqueado | |
| **Pós-Condição** | 1 – O microfone no host do aluno está ativado;  2 – O recurso “*mute mic*” na instância do aluno está desativado;  3 – O recurso “*mute mic*” na instância do aluno não está bloqueado (está sob o controle do aluno). | |
| **Ações do Ator** | | **Ações do Sistema** |
| Professor aponta para o avatar do aluno e executa a função “*action*” | |  |
|  | | Aluno é selecionado |
|  | | Menu contextual é apresentado |
| Professor desliga o switch “*mute*” | |  |
|  | | A instância do professor indica o início da operação bloqueando o switch “*mute*” |
|  | | A instância do professor envia o comando “*mute\_unlock*” ao aluno selecionado via TCP |
|  | | A instância do aluno recebe o comando “*mute\_unlock*” |
|  | | A instância do aluno ativa o microfone do host |
|  | | A instância do aluno ativa o módulo de compactação de som |
|  | | A instância do aluno envia a mensagem “*ack\_mute\_unlock*” ao professor via TPC |
|  | | A instância do professor recebe a mensagem “*ack\_mute\_unlock*” |
|  | | A instância do professor indica o sucesso da operação desbloquenado o switch “*mute*” |

3.1.2.4 **On/Off Microfone**

O Usuário deve poder ativar e desativar seu próprio microfone, através de um botão constantemente visível no HUD. Ao clicar nesse botão, o sistema deve desativar o microfone, o fluxo de áudio e o subsistema de compactação de áudio.

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | FR7: ON/OFF MICROFONE | |
| **Ator Principal** | ALUNO | |
| **Ator Secundário** | PROFESSOR | |
| **Pré-Condição** | 1 – O recurso “*mute mic*” não está bloqueado | |
| **Pós-Condição** | 1 – O microfone no host do aluno está ativado/desativado; | |
| **Ações do Ator** | | **Ações do Sistema** |
| Usuário aciona a função “mute\_mic” (default *M*) | |  |
|  | | Desativar/ativar microfone do host |
|  | | Alterar o ícone do microfone no HUD para microfone com/sem barra |

3.1.2.5 **Andar**

O usuário poderá controlar a posição de seu avatar no ambiente 3D através do teclado, assim como os ângulos de seu ponto de vista através do mouse.

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | FR8: ANDAR | |
| **Ator Principal** | ALUNO | |
| **Ator Secundário** | PROFESSOR | |
| **Pré-Condição** | O usuário deve estar dentro de uma sala virtual | |
| **Pós-Condição** | O avatar do usuário deve estar na posição escolhida | |
| **Ações do Ator** | | **Ações do Sistema** |
| Acionar funções de movimento (defaults *A, D, S, W*) | |  |
|  | | Movimentar o avatar do usuário conforme a função de movimento escolhida |

3.1.3 Material Didático

3.1.3.1 **Acessar Material Didático**

O usuário terá cesso ao material didático armazenado na base de dados, disponibilizado pela API RESTFul em <https://alloy.city/api>

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | FR9: ACESSAR MATERIAL DIDÁTICO | |
| **Ator Principal** | ALUNO | |
| **Ator Secundário** | PROFESSOR | |
| **Pré-Condição** | O usuário deve ter adquirido acesso ao material desejado | |
| **Pós-Condição** | Lista de capítulos e lições devidamente apresentada para o usuário | |
| **Ações do Ator** | | **Ações do Sistema** |
| No menu inicial, clicar em “Sala de Estudo” | |  |
|  | | Renderizar uma sala virtual sem conectividade P2P |
|  | | Instanciar um livro para cada capítulo ao qual o usuário tem acesso |
| Apontar para um dos livros e acionar a função “action” | |  |
|  | | Apresentar um menu contextual com a lista de lições daquele capítulo |
| Clicar em uma das lições | |  |
|  | | Instanciar os recursos daquela lição no ambiente virtual |

3.1.3.2 **Preencher Lacunas**

O usuário poderá visualizar e utilizar exercícios interativos do tipo FillGaps (preencher lacunas) do material didático.

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | FR10: PREENCHER LACUNAS | |
| **Ator Principal** | ALUNO | |
| **Ator Secundário** | PROFESSOR | |
| **Pré-Condição** | 1 – Pelo menos um objeto da classe FillGaps deverá estar instanciado no ambiente  2 – O atributo *enum* *controlMode* do objeto PlayerState deve conter o valor AVATAR | |
| **Pós-Condição** | 1 – O aluno deverá ter ganhado os XPs referentes ao exercício | |
| **Ações do Ator** | | **Ações do Sistema** |
| Se aproximar do objeto FillGaps | |  |
|  | | Marcar objeto FillGaps como *in\_range* |
| Apontar para o objeto FillGaps | |  |
|  | | Tornar objeto FillGaps como selecionável |
| Acionar função *action* | |  |
|  | | Fixar posição do objeto FillGaps em frente ao avatar do usuário |
|  | | Apresentar texto do exercício com as lacunas em forma de caixa de texto |
|  | | Atribuir o valor CURSOR ao atributo *enum controlMode* do objeto PlayerState |
| Preencher as lacunas (selecionar lacuna, preencher, repetir) | |  |
| Clicar no botão *send* | |  |
|  | | Enviar o conjunto de soluções à API Alloy |
|  | | Persistir o resultado na base de dados |
|  | | Atualizar o valor XP do aluno |
|  | | Atribuir o valor AVATAR ao atributo *enum* *controlMode* do objeto PlayerState |

3.1.3.3 **Ouvir Áudio**

O sistema deverá ser capaz de receber múltiplos fluxos de áudio provenientes da rede e reproduzi-los no sistema de saída de áudio da máquina.

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | FR11: OUVIR ÁUDIO | |
| **Ator Principal** | ALUNO | |
| **Ator Secundário** | PROFESSOR | |
| **Pré-Condição** | 1 – Pelo menos um objeto da classe Audio deverá estar instanciado no ambiente  2 – O atributo *enum* *controlMode* do objeto PlayerState deve conter o valor AVATAR | |
| **Pós-Condição** | 1 – O arquivo de áudio referente ao recurso deve ter sido reproduzido pelo sistema de som do host | |
| **Ações do Ator** | | **Ações do Sistema** |
| Se aproximar do objeto Audio | |  |
|  | | Marcar objeto Audio como *in\_range* |
| Apontar para o objeto Audio | |  |
|  | | Marcar objeto Audio como selecionável |
| Acionar função *action* | |  |
|  | | Fixar posição do objeto Audio na cabeça do avatar do usuário |
|  | | Apresentar o menu contextual com os controles do player de áudio |
|  | | Atribuir o valor CURSOR ao atributo *enum controlMode* do objeto PlayerState |
|  | | Começar a reproduzir o áudio |

3.1.3.4 **Múltipla Escolha**

O Usuário poderá visualizar e utilizar exercícios interativos do tipo MultipleChoice (múltipla escolha) do material didático.

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | FR12: MÚLTIPLA ESCOLHA | |
| **Ator Principal** | ALUNO | |
| **Ator Secundário** | PROFESSOR | |
| **Pré-Condição** | 1 – Pelo menos um objeto da classe MultipleChoice deverá estar instanciado no ambiente  2 – O atributo *enum* *controlMode* do objeto PlayerState deve conter o valor AVATAR | |
| **Pós-Condição** | 1 – O aluno deverá ter ganhado os XPs referentes ao exercício | |
| **Ações do Ator** | | **Ações do Sistema** |
| Se aproximar do objeto MultipleChoice | |  |
|  | | Marcar objeto MultipleChoice como *in\_range* |
| Apontar para o objeto MultipleChoice | |  |
|  | | Tornar objeto MultipleChoice como selecionável |
| Acionar função *action* | |  |
|  | | Fixar posição do objeto MultipleChoice em frente ao avatar do usuário |
|  | | Apresentar ao aluno o enunciado e a lista de opções possíveis |
|  | | Atribuir o valor CURSOR ao atributo *enum controlMode* do objeto PlayerState |
| Escolher uma das opções | |  |
| Acionar a opção *send* | |  |
|  | | Enviar a solução escolhida à API Alloy |
|  | | Persistir o resultado na base de dados |
|  | | Atualizar o valor XP do aluno |
|  | | Atribuir o valor AVATAR ao atributo *enum* *controlMode* do objeto PlayerState |

3.1.3.5 **Ordenação**

O Usuário poderá visualizar e utilizar exercícios interativos do tipo OrderItems (ordenar itens) do material didático.

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | FR13: ORDENAÇÃO | |
| **Ator Principal** | ALUNO | |
| **Ator Secundário** | PROFESSOR | |
| **Pré-Condição** | 1 – Pelo menos um objeto da classe OrderItems deverá estar instanciado no ambiente  2 – O atributo *enum* *controlMode* do objeto PlayerState deve conter o valor AVATAR | |
| **Pós-Condição** | 1 – O aluno deverá ter ganhado os XPs referentes ao exercício | |
| **Ações do Ator** | | **Ações do Sistema** |
| Se aproximar do objeto OrderItems | |  |
|  | | Marcar objeto OrderItems como *in\_range* |
| Apontar para o objeto OrderItems | |  |
|  | | Marcar o objeto OrderItems como selecionável |
| Acionar função *action* | |  |
|  | | Fixar posição do objeto OrderItems em frente ao avatar do usuário |
|  | | Apresentar ao aluno o enunciado e os elementos a ordenar |
|  | | Atribuir o valor CURSOR ao atributo *enum controlMode* do objeto PlayerState |
| Ordenar os elementos | |  |
| Acionar a opção *send* | |  |
|  | | Enviar a ordem escolhida à API Alloy |
|  | | Persistir o resultado na base de dados |
|  | | Atualizar o valor XP do aluno |
|  | | Atribuir o valor AVATAR ao atributo *enum* *controlMode* do objeto PlayerState |

3.1.3.6 **Ler Texto**

O Usuário poderá visualizar conteúdo didático do tipo Text (texto UTF8).

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | FR14: LER TEXTO | |
| **Ator Principal** | ALUNO | |
| **Ator Secundário** | PROFESSOR | |
| **Pré-Condição** | 1 – Pelo menos um objeto da classe TextResource deverá estar instanciado no ambiente  2 – O atributo *enum* *controlMode* do objeto PlayerState deve conter o valor AVATAR | |
| **Pós-Condição** | 1 – O texto útil do objeto TextResource é legível | |
| **Ações do Ator** | | **Ações do Sistema** |
| Se aproximar do objeto TextResource | |  |
|  | | Marcar objeto TextResource como *in\_range* |
| Apontar para o objeto TextResource | |  |
|  | | Marcar objeto TextResource como selecionável |
| Acionar função *action* | |  |
|  | | Fixar posição do objeto TextResource diante do rosto do avatar do usuário |
|  | | Apresentar o menu contextual com o conteúdo útil do objeto TextResource |

3.1.3.7 Caso de Uso: **Utilizar Recursos Interativos**

Entre os recursos do material didático, alguns tipos são interativos. O Usuário deverá poder agir sobre os recursos do tipo interativo, além de poder visualizar os recursos não interativos.

3.1.3.8 Caso de Uso: **Visualizar Imagens**

A instância cliente deverá ser capaz de mostrar claramente, no ambiente 3D, uma imagem, a partir de uma URL.

3.1.3.9 Caso de Uso: **Visualizar Vídeo**

A instância cliente deverá ser capaz de reproduzir vídeos do YouTube, a partir do ID do vídeo.

3.1.3.10 Caso de Uso: **Escrever**

O Usuário poderá responder questões abertas, digitando texto UTF8. A cada modificação no texto, o sistema deverá enviar para os professores presentes na sala o um pacote de informação sobre a modificação suficiente para que a instância do objeto correspondente na máquina do professor seja idêntica à que existe na máquina do aluno. Ao salvar a produção escrita do Usuário, o sistema deverá enviar o texto final da resposta para o servidor. O servidor se encarregará de persistir o texto na base de dados.

3.1.4 GUI

3.1.4.1 Caso de Uso: **Login**

Ao executar um cliente AViS, o Usuário vê uma tela de login, com os campos e-mail e senha. No MVP (Minimum Viable Product), o usuário só poderá fazer login com seu e-mail. A possibilidade de fazer login com redes sociais deverá ser implementada para a versão comercial do AViS, antes do lançamento 1.0.

3.1.4.2 Caso de Uso: **Logoff**

Na tela de menu, haverá um botão sair, que vai encerrar o programa. Na versão comercial 1.0, será preciso realizar o logoff do usuário antes de encerrar o programa, caso a propriedade connection.keepAuthToken seja false.

connection é um singleton de tipo Connection, responsável pela comunicação com a API remota.

3.1.4.3 Caso de Uso: **Visualizar Pontos de Experiência**

No HUD do Usuário, o número de pontos de experiência do usuário deverá estar constantemente visível. Este dado deverá ser atualizado em tempo real

3.1.4.3 Caso de Uso: **Visualizar Ondas de Áudio**

O HUD do usuário deverá mostrar uma representação gráfica, mesmo que rudimentar, do fluxo de áudio capturado pelo microfone do usuário. No MVP, essa representação poderá se limitar a uma barra de progressão que flutua com base no volume de som, originado no microfone local, sendo processado pelo sistema.

4 ARQUITETURA DO SISTEMA

<ESCREVER AQUI, DE MANEIRA SUSCINTA, QUAL FOI O MODELO

ARQUITETURAL ESCOLHIDO PELO GRUPO>

**4.1** DIAGRAMA DE PACOTES

<ESCREVER AQUI UMA DEFINIÇÃO DO QUE É O DIAGRAMA DE PACOTES, DE ACORDO COM ALGUMA BIBLIOGRAFIA DA ÁREA DE ENGENHARIA DE SOFTWARE>

<INSERIR O DIAGRAMA AQUI>

**4.2** FERRAMENTAS E TECNOLOGIA

* Unreal Engine 4

É um motor de jogo desenvolvido pela Epic Games escrito em C++;

Visual Studio Community

É uma IDE da Microsoft para desenvolvimento de softwares.

* Git

É um sistema de controle de versões utilizado para registrar os históricos de edições de qualquer tipo de arquivo

* CI (Continuous Integration)

É uma prática de desenvolvimento com o objetivo de encontrar e ivestigar bus melhorando a qualidade do software e velocidade de desenvolvimento.

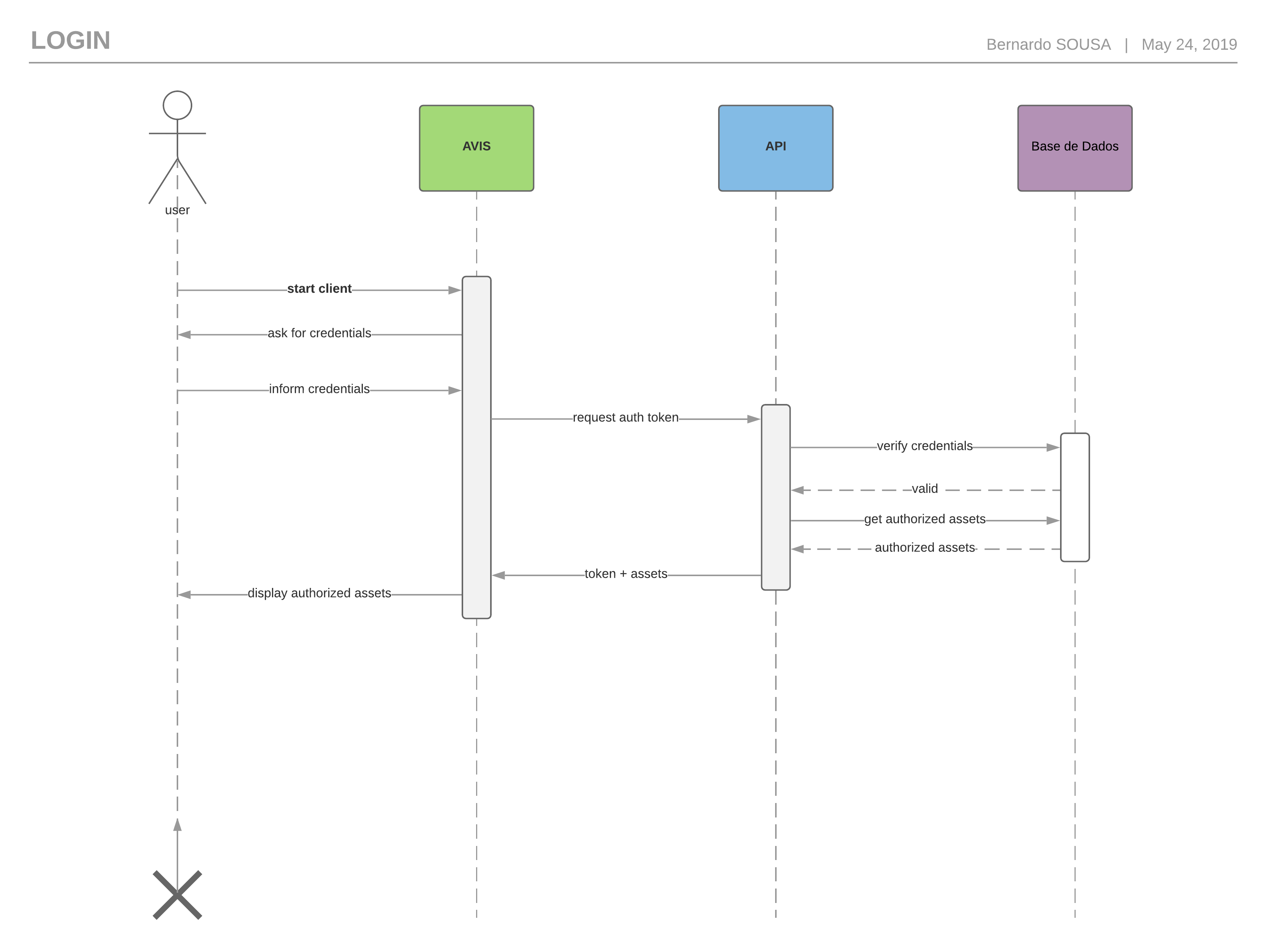
* CD (Continuous Integration)

É uma série de práticas destinadas a garantir que o código pode ser implantado com rapidez e segurança na produção, fornecendo todas as alterações em um ambiente de trabalho e garantindo que as aplicações e serviços de negócios funcionem como esperado por meio de rigorosos testes automatizados.

5 DIAGRAMAS DE SEQUENCIA

<ESCREVER AQUI UMA DEFINIÇÃO DO QUE É O DIAGRAMA DE SEQUENCIA, DE ACORDO COM ALGUMA BIBLIOGRAFIA DA ÁREA DE ENGENHARIA DE SOFTWARE, COMO POR EXEMPLO:>

O Diagrama de Sequência é um diagrama comportamental que procura determinar a sequência de eventos que ocorrem em um determinado processo, identificando quais mensagens devem ser disparadas entre os elementos envolvidos e em que ordem. (GUEDES, 2009).



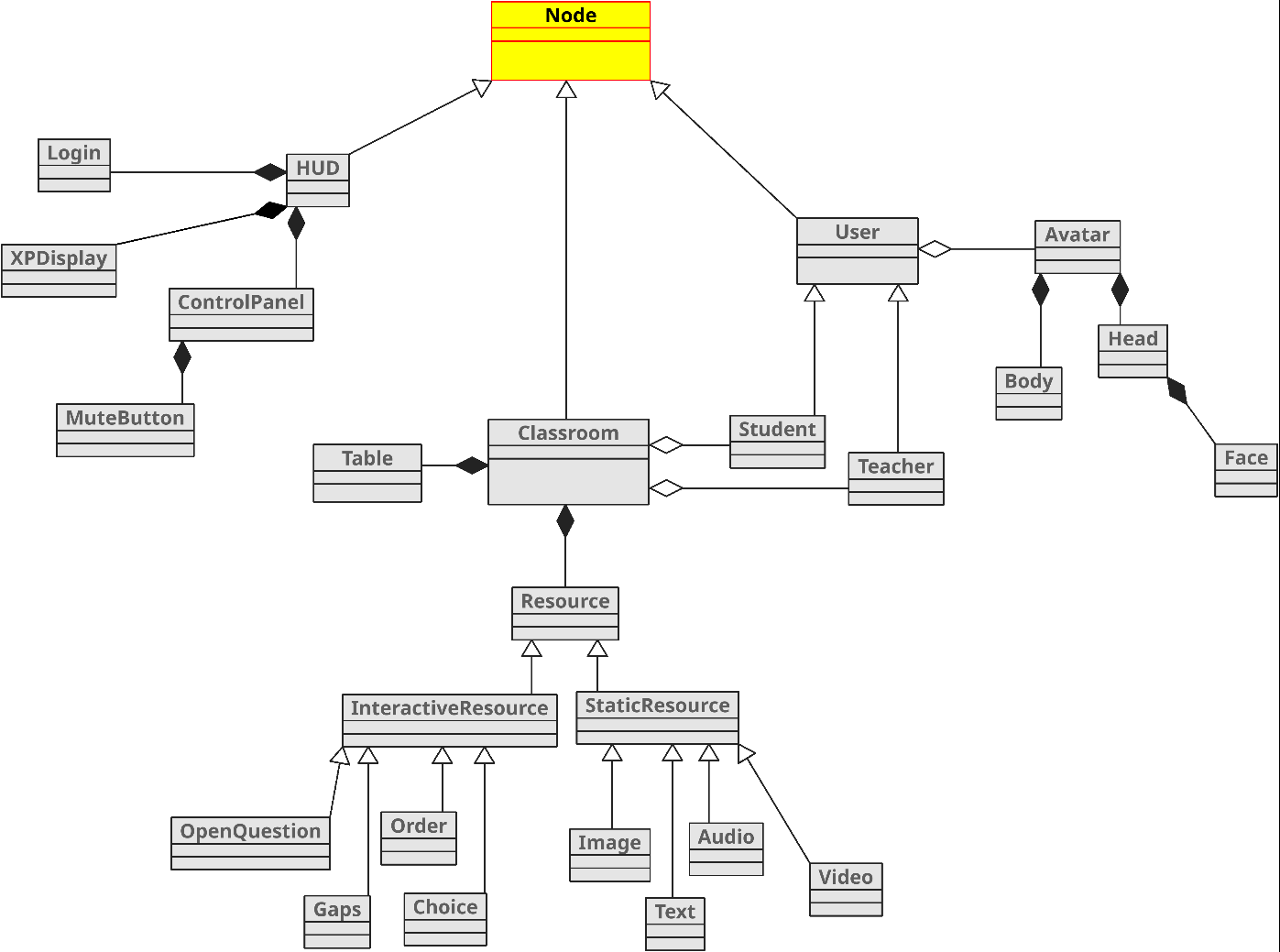
6 DIAGRAMAS DE ESTADOS

<ESCREVER AQUI UMA DEFINIÇÃO DO QUE É O DIAGRAMA DE ESTADOS, DE ACORDO COM ALGUMA BIBLIOGRAFIA DA ÁREA DE ENGENHARIA DE SOFTWARE>

<INSERIR OS DIAGRAMAS AQUI

7 OUTROS DIAGRAMAS

Esboço do principal diagrama de classes (apenas as relações)



8 TELAS DO PROTÓTIPO





9 CONSIDERAÇÕES FINAIS

Dica:

Explicar como a Questão de Pesquisa e os Objetivos foram alcançados.

BIBLIOGRAFIA

<ESCREVER AQUI, EM ORDEM ALFABÉTICA, AS REFERÊNCIAS BIBLIOGRÁFICAS USADAS PARA CONCEITUAR OS DIAGRAMAS DO TRABALHO, COMO POR EXEMPLO: >

GUEDES, Gilleanes T. A. **UML 2: uma abordagem prática.** São Paulo: Novatec Editora, 2009.