

Curso de Análise e Desenvolvimento de Sistemas

Pedro Bernardo de SOUSA 0030481711006

Weuller Júnior Souza Bessa 0030481621040

Vítor Andrade Marques da Silva 0030481511040

AViS – Alloy Virtual Space

Documentação de Desenvolvimento de Software

Sorocaba

Dezembro – 2019



Pedro Bernardo de SOUSA 0030481711006

Weuller Júnior Souza Bessa 0030481621040

Vítor Andrade Marques da Silva 0030481511040

AViS – Alloy Virtual Space

Documentação de Desenvolvimento de Software

Trabalho de Graduação apresentado à Faculdade de Tecnologia de Sorocaba – FATEC, sob orientação de **M.ª** **Maria Angélica Calixto de Andrade Cardieri**, como parte dos pré-requisitos para obtenção do título de *Tecnólogo em Análise e Desenvolvimento de Sistemas*

Sorocaba

Dezembro – 2019

Dedicatória

Agradecemos às nossas famílias pelo tenaz apoio oferecido generosamente ao longo deste curso e, mais precisamente, durantes os meses de esforço que renderam este trabalho. Sem o alto apreço por ciência e educação compartilhado pelas nossas três famílias, esta grande empreitada não teria sido possível.

Agradecimentos

Muito obrigado a todo corpo docente, por ter sido em nós uma força transformadora ao longo desses três anos. Agradecimentos muito especiais a Me. Ana Carolina Camargo, Me. Cesar Munari, Prof. Jefferson Blaitt e Me. Maria Angélica Cardieri, nossa orientadora, pelos esforços de mentoria muito além de suas funções. Obrigado por terem notado nosso módico potencial, quando jazia escondido por nossa incipiência.

* Resumo

A Alloy City Linguistics desenvolveu e mantém uma plataforma de ensino e aprendizado de francês baseada em tecnologias web. O objetivo deste trabalho é demonstrar a viabilidade de um cliente para essa plataforma existente. O cliente, chamado AViS - Alloy Virtual Space, se apoia na RESTful API disponibilizada pela Alloy para o projeto. O cliente AViS é 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 – Visão geral da gestão de pessoas 10

Figura 2 – Enfoque sistêmico nas organizações 12

Figura 3 – Esquema da estratégia empresarial e gestão de pessoas 13

Figura 4 – Fronteiras virtuais nas organizações 24

Figura 5 – Gestão das tecnologias da informação e gestão de pessoas 25

* Lista de Tabelas

Tabela 1 – Principais mudanças na Área de Recursos Humanos (PricewaterhouseCoopers) 17

Tabela 2 – Principais mudanças na Área de Recursos Humanos (Chiavenato) 22

Tabela 3 – Resumo do Estudo de Caso 1 29

Tabela 4 – Ferramentas de T.I. e seu relacionamento com os processos da ARH (estudo de caso 1) 30

Tabela 5 – Ferramentas de T.I. e seu relacionamento com os processos da ARH (estudo de caso 2) 36

Índice

[1. Introdução 6](#_Toc23859053)

[2. Embasamento teórico. 7](#_Toc23859054)

[3. Planejamento Inicial do Software 10](#_Toc23859055)

[3.1. Situação Atual 10](#_Toc23859059)

[3.1.1. Problemas Encontrados 11](#_Toc23859060)

[3.1.2. Aplicativos Disponíveis no mercado (estado da arte) 12](#_Toc23859061)

[3.2. Objetivos do Projeto 13](#_Toc23859062)

[3.3. Riscos do Projeto AViS 14](#_Toc23859063)

[3.3.1. Limitações Operacionais 14](#_Toc23859064)

[3.3.2. Considerações Legais 14](#_Toc23859065)

[3.3.3. Considerações de Hardware / Software / Rede 15](#_Toc23859066)

[3.3.4. Políticas Organizacionais 16](#_Toc23859067)

[3.3.5. Plano de Riscos 16](#_Toc23859068)

[4. Análise de Requisitos 18](#_Toc23859072)

[4.1. Descrição da técnica utilizada para levantamento dos requisitos 18](#_Toc23859074)

[4.2. Situação Proposta 18](#_Toc23859075)

[4.3. Requisitos Funcionais 19](#_Toc23859076)

[4.3.1. Casos de Uso 20](#_Toc23859077)

[4.4. Requisitos Não Funcionais 23](#_Toc23859078)

[4.4.1. Tempo de resposta 23](#_Toc23859079)

[4.4.2. Uso de memória 24](#_Toc23859080)

[4.4.3. Uso de espaço em disco 24](#_Toc23859081)

[4.4.4. Uso de recursos de processamento no servidor 24](#_Toc23859082)

[5. Projeto Detalhado do Software 25](#_Toc23859083)

[5.1. Arquitetura da aplicação proposta 25](#_Toc23859085)

[5.2. Tecnologias utilizadas e APIs 28](#_Toc23859086)

[5.2.1. Tecnologias 28](#_Toc23859087)

[5.2.2. APIs Utilizadas 29](#_Toc23859088)

[5.3. Componentes do SW 30](#_Toc23859089)

[5.4. Diagrama de Classes 31](#_Toc23859090)

[5.5. Considerações sobre o Banco de Dados Utilizado 33](#_Toc23859091)

[5.6. Diagramas de Sequência 34](#_Toc23859092)

[5.7. Diagrama Estados 37](#_Toc23859093)

[5.8. Interfaces com o usuário 38](#_Toc23859094)

[6. Implementação 39](#_Toc23859095)

[6.1. Modelagem 3D 39](#_Toc23859097)

[6.2. Captura e Processamento de Vídeo 40](#_Toc23859098)

[6.3. Captura e Processamento de Áudio 41](#_Toc23859099)

[7. Projeto de Teste 42](#_Toc23859100)

[8. Instalação do Software 42](#_Toc23859101)

[9. Análise dos Resultados 43](#_Toc23859102)

[10. Conclusão 43](#_Toc23859103)

# Introdução

Atualmente, novas metodologias de ensino estão sendo associadas a novos recursos tecnológicos, visando facilitar o processo de aprendizagem, tanto para cursos de qualificação profissional quanto para cursos na área acadêmica.

Com o propósito de proporcionar uma experiência de aprendizado à distância mais abrangente, este projeto busca estudar a viabilidade de uma ferramenta que forneceria aos usuários uma gama mais completa dos aspectos comunicacionais do processo educativo. O contexto comercial em que se insere o projeto é definido pela plataforma de ensino a distância *Pantoufle Français Online*, desenvolvida pela *Alloy City Linguistics* com base em tecnologias *web*. Esta plataforma está em produção e expõe uma API que ignora a natureza do front-end. Este contexto além de oferecer a liberdade técnica necessária para a execução do projeto, tem apresentado razões comerciais para sua existência, o que será explorado na seção 3.1.

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 a distância, proverá aos usuários, um significativo avanço na qualidade dos estímulos sensoriais, proporcionando que mais das informações oferecidas possam ser absorvidas.

Os 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.

# Embasamento teórico.

Um Ambiente Virtual de Aprendizagem, na área de educação, é uma plataforma, geralmente baseada em tecnologias web, que oferece suporte ao processo educativo. Geralmente, plataformas AVA são desenvolvidas por instituições de ensino, que veem benefícios em disponibilizar material didático em formato digital para os alunos, além de oferecer uma via suplementar de contato entre tutores e estudantes.

Ambientes virtuais de aprendizagem podem ser divididos em 4 categorias, formadas pela associação combinatória de duas grandezas: tempo e espaço. Quanto ao tempo, os participantes podem ou não interagir sincronamente, isto é, em um mesmo momento. Quanto ao espaço, os participantes podem ou não interagir em um mesmo lugar.

Quando os participantes interagem em um mesmo momento, diz-se que a aula é síncrona. Uma aula gravada, por outro lado, é chamada assíncrona, pois a interação entre o professor e o aluno acontece em momentos distintos: o professor grava o conteúdo em um momento e o aluno o consome em outro.

Quando os participantes interagem em um mesmo lugar, por exemplo em uma sala de aula tradicional, o aluno está em presença do professor. Isso classifica a aula como presencial. Oposto a isso, é possível ministrar aulas a distância, com o uso de tecnologias de comunicação, como cartas, telefone ou Internet.

Durante muitos séculos, só era possível aulas síncronas presenciais. Remotamente, não se podia gerar conteúdo educacional e esperar que alunos usufruíssem dele imediatamente. No estado tecnológico a que chegou nossa civilização, podemos observar as quatro combinações possíveis. Em salas de aulas tradicionais, temos interação direta entre tutores e alunos, o que se categoriza como educação presencial síncrona. Nessas mesmas salas de aula, o professor pode apresentar conteúdo gravado por especialistas em mídias como fita magnética ou YouTube, o que se classifica como educação presencial assíncrona. O processo educacional pode se dar na residência do aluno, com conteúdo gravado previamente, o que se classifica como educação remota assíncrona. E, finalmente, alunos e professores podem se encontrar *online*, em plataformas de videoconferência, em horários acordados previamente, para gozar de um processo educacional remoto e assíncrono. Na figura 1, observa-se em um plano cartesiano os dois eixos descritos acima, acomodando essas quatro modalidades de educação.



Figura - Modalidades de educação

Os 5 anos de experiência em educação a distância da escola Pantoufle demonstram que a associação das duas modalidades remotas de educação tem um enorme potencial de baixar custos sem sacrificar qualidade no processo educacional. É desta convicção que emana o apoio que a escola, assim como a Alloy City Linguistics, consagra ao presente projeto.

Contabilizar os custos da educação não é uma tarefa trivial. É preciso considerar gastos com transporte das pessoas que devem se deslocar até a escola, custos iniciais de investimento em infraestrutura, custos de manutenção das instalações da escola, entre muitos outros. Alguns desses custos são indiretos, como o custo do risco acrescido pelos deslocamentos frequentes. Riscos de acidentes, por exemplo. É verdade que tais custos indiretos parecem extrapolações desnecessárias nessa análise. No entanto são custos reais, pagos pela sociedade como um todo, em uma época de optimizações de grande escala e esforços de transição tecnológica, a caminho do nível 1 na escala civilizacional Kardashev (KARDASHEV, 1964).

Ao desenvolver sistemas que ofereçam a melhor experiência educacional possível a distância, se servindo para tanto de desdobramentos tecnológicos recentes, contribui-se, mesmo que modicamente, com esses esforços globais.

Atualmente, atividades educativas remotas síncronas são realizadas com sistemas de videoconferência que se concentram, sobretudo, na transmissão de vídeo e áudio via protocolo TCP/IP. Via de regra, tais projetos ignoram considerações suplementares a respeito da importância da sensação de presença oferecida por um ambiente de aula físico, talvez por ignorar a importância desse aspecto, ou talvez para evitar os prováveis embaraços técnicos da empreitada. Esses benefícios são, entretanto, flagrantes demais para se evitar indefinidamente, sobretudo na área de aquisição de língua estrangeira para adultos, especialidade da escola Pantoufle.

Lança-se, portanto, neste Trabalho de Graduação, a empreitada de estudar tais questões e oferecer um elemento de resposta, por meio de Tecnologias da Informação e Engenharia de Software.

# Planejamento Inicial do Software

Tendo em vista o contexto da escola *Pantoufle*, instituições da mesma ordem, além da problemática educacional descrita na seção 2 – Embasamento Teórico – analisa-se a seguir os processos operacionais da escola. Em seguida, tenta-se propor um direcionamento de pesquisa e desenvolvimento de software relevante e promissor.



## Situação Atual

A *Alloy City Linguistics*, fundada em 2014 por um dos autores do presente trabalho, em sociedade com a especialista em didática Me. Cecília SOUSA, desenvolveu uma plataforma AVA para a escola de francês Pantoufle Français Online. A escola utiliza a plataforma como repositório tanto do material didático como de informações sobre os alunos e professores. Ela sustenta a posição de que contato síncrono entre o aluno e o professor é essencial para o processo de aquisição de língua estrangeira. Ao mesmo tempo, a escola ministra suas aulas exclusivamente via Internet. Isso significa que a totalidade das horas de contato face a face entre aluno e professor ocorre em ferramentas de videoconferência disponíveis no mercado, como Skype. A plataforma desenvolvida, embora presente em vários processos operacionais da escola, não provê esse gênero de contato.

Os usuários do sistema são categorizados por nível de acesso.

O primeiro nível é o de aluno. Alunos podem navegar por seções previamente adquiridas do material didático, que é estruturado em temas e níveis de dificuldade. Eles podem também acessar informações sobre as turmas disponibilizadas pela coordenadoria da escola, assim como realizar tarefas de cunho administrativo, como efetuar suas próprias matrículas e pagamentos. Além disso, alunos podem visualizar informações sobre si mesmos, como presenças em aulas passadas e número de pontos de experiência adquiridos em exercícios interativos.

O segundo nível é o do professor, que tem acesso à totalidade do material didático, informações de rendimento sobre os alunos, entre outros. Professores também tem acesso a um painel de controle de presença dos alunos, onde podem alterar o status de presenças referentes às suas aulas, ou ainda visualizar presenças referentes a aulas ministradas por colegas.

O terceiro nível é reservado a criadores de conteúdo. Este nível dá acesso a ferramentas de edição e criação de conteúdo didático, que são utilizadas por um número muito restrito de professores. Trata-se de uma seleção de educadores altamente qualificados, incumbidos de garantir a qualidade do material disponibilizado na plataforma.

O quarto nível de acesso é o nível da coordenadoria, que engloba apenas pessoas a quem foi concedida a autonomia de criar turmas, alterar preços dos cursos disponibilizados e administrar as agendas dos professores.

Cada um desses níveis dá acesso a uma parte diferente do sistema, de forma que praticamente todas as facetas operacionais da escola são orquestradas por meio da plataforma. Rotineiramente, alunos interessados em um determinado curso se matriculam, fazem o pagamento pela própria plataforma e aguardam o dia do primeiro encontro online. No momento definido, a aula acontece entre o professor e seus alunos por videoconferência em grupo, no Skype. Durante a aula, alunos e professor mantêm pelo menos duas janelas abertas no sistema operacional: uma corresponde ao Skype e outra corresponde à plataforma Pantoufle. Na plataforma, o professor aciona um comando que estabelece conexões via WebSocket entre todos os clientes envolvidos na aula. Uma vez estabelecida essa conexão, o professor pode controlar a parte do material didático que aparece na tela de todos os alunos da turma. O conjunto de funcionalidades desenvolvidas para as aulas remotas e síncronas visa oferecer uma experiência de aula rica, em que os alunos podem aproveitar o melhor possível o conhecimento do professor que, por sua vez, consegue administrar o andamento da aula e guiar os alunos pelo conteúdo.

Ao longo da aula, o professor informa o sistema sobre quais alunos estão presentes, assim como sobre seus níveis de desenvolvimento em certas competências linguísticas, como compreensão oral ou expressão escrita.

### Problemas Encontrados

Um dos problemas mais citados por usuários do sistema é a falta de integração entre a videoconferência e o restante do sistema. Trata-se de dois sistemas isolados e um deles, o software de videoconferência, está absolutamente fora do controle da escola. Além disso, a necessidade de manejar duas janelas simultaneamente é inconveniente para muitos, sobretudo para usuários de computadores de pequeno porte, como telefones.

De um ponto de vista didático-pedagógico, em comparação com aulas presenciais, interações via videoconferência são limitadoras. A experiência é comparável a observar o interlocutor através de uma janela ou aparelho de televisão. Usuários do sistema que migraram recentemente, vindos de escolas tradicionais, apontam que aulas a distância são menos engajadoras e geram uma impressão de artificialidade.

### Aplicativos Disponíveis no mercado (estado da arte)

Atualmente há diversos sistemas utilizados para ensino à distância. Muitos são amplamente utilizados, outros estão ainda em desenvolvimento. Esses sistemas são categorizados como AVAs (Ambiente Virtual de Aprendizagem), ou seja, sistema de apoio ao ensino a distância proporcionando conteúdos e recursos necessários para total ou parcial aproveitamento dos cursos. Alguns dos sistemas AVA mais populares no mercado são citados abaixo:

* Amadeus LMS

*Amadeus* é um LMS - do inglês, “Learning Manager System”. Um software de ensino a distância, EaD, que propõe o conceito de *blended learning*, ou *b-learning*, derivado do *e-learning*. Propõe-se a associação de aulas a distância e aulas presenciais.

* Moodle

*Moodle* é um software que permite criação de cursos num contexto de *b-learning* ou *e-learning* amplamente utilizado em de EaD.

* Blackboard Collaborate

*Blackboard* é uma plataforma para ensino a distância e presencial permitindo que os alunos se comuniquem com seus professores utilizando chat em vídeo.

* Second Life

*Second Life* é um jogo multijogador massivo online que ambiciona simular certos aspectos da vida real. Em 2014, pesquisadores da *Texas A&M University* e da *Florida Institute of Technology* utilizaram do jogo como forma de engajar os alunos no aprendizado em um ambiente virtual.

## Objetivos do Projeto

O principal objetivo do projeto é estudar a viabilidade, técnica e comercial, da implementação de uma ferramenta de interação remota e síncrona, que ofereça melhor sensação de presença em contextos educacionais, através da renderização tridimensional de um ambiente compartilhado simulado.

Os principais desafios encontrados podem ser divididos em duas categorias: **desafios tecnológicos de implementação** e **problemáticas interdisciplinares**.

De um ponto de vista técnico, a solução que se propõe é bastante desafiadora. Especialmente em relação ao conteúdo curricular do curso de ADS. Simulações tridimensionais, VoIP e visão computacional são exemplos de assuntos que estão fora do escopo do curso e foram estudados para a implementação do projeto.

Considerando pontos de contato com outras áreas, o projeto é fortemente interdisciplinar. Os esforços de desenvolvimento são frequentemente direcionados por considerações aportados de pedagogia, psicologia e, sobretudo, linguística.

Para atingir os objetivos propostos face os desafios encontrados, assumiu-se a estratégia MVP, ou *Minimum Viable Product*. Trata-se de um protótipo executável que comporta uma seleção mínima de funcionalidades centrais. É importante notar que funcionalidades genéricas, como menus e um subsistema de autenticação, por mais importantes que sejam em um produto comercial, não se qualificam para um MVP, já que não contribuem para demonstrar a viabilidade técnica do conceito central. São funcionalidades genéricas, presentes em projetos dos mais diversos.

Com a vantagem de manter o escopo sob controle, a estratégia de desenvolvimento MVP requer uma seleção das funcionalidades centrais do conceito proposto. No caso do presente projeto, são funcionalidades que demonstrem uma simulação tridimensional onde sejam compartilhados entre os usuários uma seleção de dados com os quais seja possível gerar uma sensação de presença física mais completa do que uma simples videoconferência.

Uma análise mais detalhada dos objetivos do projeto é feita na seção 4.2 – Situação proposta.

## Riscos do Projeto AViS

Segue uma análise dos principais riscos associados ao projeto, divididos em limitações operacionais, considerações legais, considerações de hardware, software e rede, além de políticas organizacionais relevantes. A seção é encerrada com uma proposta de plano de gerenciamento dos principais riscos identificados.

### Limitações Operacionais

Entre as limitações operacionais que podem interferir no correto funcionamento da aplicação, podemos citar:

* Oscilações na largura de banda e/ou interrupção da conexão com a internet de um ou mais usuários ativos em um ambiente virtual durante suas interações;
* Falta de energia que afete o servidor ou um dos usuários ativos no ambiente virtual;
* Incompatibilidade de hardware ou software (sistema operacional) após uma atualização do sistema operacional;
* Indisponibilidade do servidor de hospedagem dos serviços necessários à aplicação.

### Considerações Legais

O MVP em si não apresenta riscos legais consideráveis, dado o alcance extremamente reduzido do software desenvolvido. O aplicativo será limitado à demonstração da tecnologia à banca avaliadora, em um ambiente controlado.

Futuramente, caso o projeto seja levado adiante e se torne um produto no mercado, o termo de compromisso utilizado pela escola Pantoufle, disponível no anexo 1 deste trabalho, continuará válido, mas seria aconselhável estudar acordos de utilização e termos de compromisso associados a sistemas de videoconferência afim de identificar preocupações recorrentes de projetos que oferecem esse tipo de serviço. Tais esforços tornariam possível expandir o termo de compromisso da escola, de forma que ele cubra a nova realidade proporcionada pelas funcionalidades acrescidas.

Quanto a software utilizado no desenvolvimento do protótipo, três esferas distintas devem ser consideradas.

Primeiramente, licenças de utilização de produtos Microsoft, como Visual Studio e Word, foram providas pelo acordo de cooperação celebrado entre o Centro Paula Souza e Microsoft Brasil, em 21 de setembro de 2007. Maiores detalhes sobre este acordo encontram-se no anexo 2.

Em segundo lugar, diversas ferramentas de código fonte abeto foram utilizados. Seguem as licenças que regem os aspectos legais de cada projeto utilizado:

* OpenCV - 3-clause BSD License – anexo 3
* Gimp – versão 3 do GNU GENERAL PUBLIC LICENSE – anexo 4
* Inkscape - versão 3 do GNU GENERAL PUBLIC LICENSE – anexo 4

Finalmente, Unreal Engine 4, apresenta os aspectos legais mais delicados do projeto, dada sua licença atípica. Trata-se de um acordo que oferece aos usuários do mecanismo acesso completo ao código fonte, inclusive permitindo sua modificação para adaptá-lo às necessidades do projeto em mãos, mas não se trata de software de código aberto. O código é acessível, mas não se pode utilizá-lo comercialmente sem prestar contas à Epic Games, criadora do mecanismo de jogo UE4. O acordo assegura o uso gratuito do mecanismo para fins educacionais e de pesquisa, o que engloba o presente MVP. Quanto à projetos futuros, derivados do MVP aqui desenvolvido, o acordo de utilização do mecanismo, disponível no anexo 5 deste trabalho, garante à Epic Games 5% do faturamento ligado às vendas do software, após os primeiros U$ 3000.

### Considerações de Hardware / Software / Rede

A aplicação foi desenvolvida com foco no sistema operacional Windows 10 de 64 bits. As demandas em relação ao que esse sistema operacional exige, especificam o mínimo necessário ao bom funcionamento da aplicação. Entretanto, para o pleno uso das funcionalidades da aplicação, exige-se também um microfone e uma webcam para capturar a voz e a imagem do usuário, além desses itens de hardware, a estação de trabalho deve dispor de uma conexão com a internet, quanto à largura de banda da conexão, um mínimo de 1Mbps para download e 1Mbps de upload.

Para o funcionamento do MVP, o projeto AViS deve contar com as portas 7777 e 7778 abertas no servidor da aplicação para os protocolos UDP e TCP. Não há a necessidade de manipulação das configurações do roteador ou das políticas do firewall nas estações de trabalho dos usuários.

### Políticas Organizacionais

As considerações que devem ser levadas em conta ao se utilizar uma ferramenta com funcionalidades de comunicação síncrona com áudio e vídeo são similares às exigidas pelo uso de Skype, por exemplo. Ao intermediar encontros entre alunos e professores, a empresa deve conscientizar todos envolvidos quanto às suas responsabilidades pessoais, diante da câmera. A postura dos envolvidos deve ser comparável a adotada tipicamente em salas de aula presenciais, com algumas ressalvas.

Quanto à segurança da informação, é crucial implementar mecanismos de validação de dados que circulam entre diferentes instancias do sistema. Tal qual implementado no momento, o MVP trabalha com dados brutos, o que, em um ambiente de produção, representaria uma vulnerabilidade de fácil exploração.

### Plano de Riscos

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PLANO DE RISCOS** | | | | |
|
| **Risco** | **Probab (P)** | **Impacto (I)** | **RiscoTotal (P x I)** | **Tratamento** |
|
| Oscilações / interrupções do provedor de internet | 5 | 4 | 20 | Buscar a opção mais estável dentre os provedores disponíveis. |
| Falta de energia | 2 | 4 | 8 | Compra de nobreaks. |
| Problemas de compatibilidade com atualizações do SO | 4 | 3 | 12 | Antes de cada atualização, criar um ponto de restauração do sistema. |
| Indisponibilidade do servidor de hospedagem | 2 | 5 | 10 | Verificar a opção mais confiável dentro do orçamento. |
| Uso de imagem indevida entre os usuários | 2 | 5 | 10 | Estabelecer um documento de termos e condições de uso. |
| Especificações incompatíveis da estação de trabalho | 1 | 5 | 5 | Realizar as adequações necessárias de hardware, software e acesso à internet. |
| Indisponibilidade da porta 7777 para os protocolos UDP e TCP | 1 | 5 | 5 | Realizar a configuração necessária para a operação. |

|  |  |
| --- | --- |
| **Probabilidade e Impacto** | |
| **Valor** | **Descrição** |
| 1 | Muito baixa |
| 2 | Baixa |
| 3 | Média |
| 4 | Alta |
| 5 | Muito alta |



# Análise de Requisitos



## Descrição da técnica utilizada para levantamento dos requisitos

Ao longo dos últimos 18 meses, desde o lançamento da terceira iteração da plataforma Pantoufle, professores e alunos da escola vêm compartilhando suas impressões sobre o sistema. É com base nesse retorno que a necessidade de mercado descrita na seção 3.1.1 foi identificada. A seleção das funcionalidades do MVP, por sua vez, foi feita a partir da distinção entre casos de uso triviais, que o sistema já faz ou poderia fazer em um ou dois ciclos de desenvolvimento, e casos de uso atípicos, que talvez tenham o potencial de revolucionar o ensino síncrono remoto.

Por um lado, a plataforma Pantoufle, atualmente em desenvolvimento contínuo, com base em ciclos curtos de desenvolvimento, coleta continuamente informações sobre as necessidades da escola. Por outro lado, numa postura exploratória de pesquisa e desenvolvimento de software, este trabalho se apoiou no contexto e *savoir faire* da escola Pantoufle para definir os requisitos que, uma vez implementados, trazem elementos de resposta à problemática delimitada.

É notável que este processo de levantamento de requisitos não é usual, mas admite-se que um projeto ousado e inovador, que vive no limite entre o conhecido e o desconhecido, não se incline a vias normalizadas e repetitivas de desenvolvimento.

## Situação Proposta

Numa videoconferência típica, transmite-se os fluxos de dados gerados por um microfone e por uma câmera. Uma abordagem simples, que provê uma experiência comparável a observar o interlocutor por um aparelho de TV. O projeto AViS pretende demonstrar a viabilidade de criar uma experiência mais rica e mais imersiva, em que um espaço tridimensional virtual simulado seja compartilhado entre os usuários. Neste espírito, o sistema desenvolvido é capaz de:

* simular um espaço virtual em três dimensões;
* gerar avatares que representem os usuários conectados;
* criar a sensação de que a voz de um determinado usuário remoto está emanando do ponto no espaço virtual onde se encontra a cabeça de seu avatar; e
* apresentar o rosto do usuário remoto no rosto de seu avatar.

Como veremos em maiores detalhes na seção 5.2.1, diversas tecnologias distintas foram associadas para implementar as funcionalidades acima. Dentre elas, as mais notáveis são Unreal Engine 4, mecanismo de jogo responsável por gerar e manter o ambiente 3D em tempo de execução, e OpenCV (*Open Computer Vision*), responsável por identificar o rosto do usuário via aprendizado de máquina.

Os demais processos envolvidos no funcionamento do MVP são descritos detalhadamente na seção 6 – Implementação.

## Requisitos Funcionais

**RF1 – Andar**

O usuário poderá movimentar seu avatar no espaço virtual da sala de aula, utilizando as teclas W, A, S e D do teclado. A posição de todas as instâncias remotas de um determinado avatar deverão ser sincronizadas conforme os movimentos registrados pela instância local, de forma que todos os usuários conectados à sala virtual vejam os demais em suas devidas posições.

**RF2 – Olhar**

O usuário poderá modificar os ângulos, em dois eixos, para ajustar o ponto de vista em que observa o ambiente virtual através de movimentos do mouse. Por exemplo, ao movimentar o mouse para frente, o ângulo de visão no eixo horizontal diminuirá, para que se olhe para baixo. Ao movimentar o mouse para um lado, o ângulo de visão no eixo vertical será ajustado. Os ajustes serão sincronizados entre todas as instâncias conectadas à sala virtual.

Este requisito atende a necessidade que o usuário terá de observar o ambiente virtual à sua volta.

**RF3 – Falar**

O usuário terá sua voz capturada pelo sistema e transmitida diretamente para as instâncias remotas. Cada instância remota reproduzirá o som recebido ajustando continuamente a posição de origem do som, no mecanismo de áudio posicional do ambiente virtual, para que ela coincida com a posição da cabeça do avatar correspondente a instância onde o som fora capturado.

**RF4 – Transmitir expressão facial**

A expressão facial do usuário será continuamente capturada pelo sistema, enquanto ele estiver com a webcam ligada. As imagens correspondentes ao rosto do usuário serão transmitidas diretamente às instâncias remotas. Cada instância remota aplicará a imagem mais recente disponível no rosto do avatar correspondente à instância onde a imagem fora capturada.

### Casos de Uso



|  |  |  |
| --- | --- | --- |
| Caso de Uso | FR1: Andar | |
| Ator Principal | Usuário local | |
| Atores Secundários | Usuários remotos | |
| Pré-Condição | O usuário está conectado a uma sala virtual | |
| Pós-Condição | Posição do avatar sincronizada entre todas as instâncias conectadas | |
| Ações do Ator | | Ações do Sistema |
| 1 – Pressiona teclas de movimento (W, A, S, D) | |  |
|  | | 2 – Envia comando ao servidor |
|  | | 3 – Servidor calcula nova posição do avatar |
|  | | 4 – Servidor envia novas coordenadas às instâncias conectadas |
|  | | 5 – Posição do avatar é atualizada |

|  |  |  |
| --- | --- | --- |
| Caso de Uso | FR2: Olhar | |
| Ator Principal | Usuário local | |
| Atores Secundários | Usuários remotos | |
| Pré-Condição | O usuário está conectado a uma sala virtual | |
| Pós-Condição | 1 - Posição da cabeça do avatar sincronizada entre todas as instâncias conectadas  2 – Ponto de vista do ator principal no ambiente virtual ajustado | |
| Ações do Ator | | Ações do Sistema |
| 1 – Movimenta o mouse | |  |
|  | | 2 – Envia comando ao servidor |
|  | | 3 – Servidor calcula novos ângulos, nos dois eixos |
|  | | 4 – Servidor envia novos ângulos às instâncias conectadas |
|  | | 5 – Instância local atualiza ângulos do ponto de vista do ator principal |
|  | | 6 – Instâncias remotas atualizam ângulo de inclinação da cabeça do avatar |
|  | | 7 – Instâncias remotas atualizam ângulo do avatar no eixo vertical |

|  |  |  |
| --- | --- | --- |
| Caso de Uso | FR3: Falar | |
| Ator Principal | Usuário local | |
| Ator Secundário | Usuários remotos | |
| Pré-Condição | O usuário está conectado a uma sala virtual | |
| Pós-Condição | Todos os usuários conectados à sala virtual ouvem a voz do ator principal | |
| Ações do Ator | | Ações do Sistema |
| 1 – Usuário fala | |  |
|  | | 2 – Captura áudio com microfone padrão |
|  | | 3 – Fragmenta áudio com base em limiar de volume |
|  | | 4 – Compacta o fragmento de áudio |
|  | | 5 – Envia o fragmento de áudio às instâncias remotas |
|  | | 6 – Instância remota descompacta o áudio |
|  | | 7 – Instância remota reproduz o áudio, utilizando a posição da cabeça do avatar correspondente à instância de origem do som, para configurar o mecanismo de áudio posicional |

|  |  |  |
| --- | --- | --- |
| Caso de Uso | FR4: Transmitir expressão facial | |
| Ator Principal | Usuário local | |
| Ator Secundário | Usuários remotos | |
| Pré-Condição | O usuário está conectado a uma sala virtual | |
| Pós-Condição | Usuários conectados à sala virtual veem os rostos uns dos outros | |
| Ações do Ator | | Ações do Sistema |
| 1 – Usuário aciona Webcam | |  |
|  | | 2 – Captura imagem |
|  | | 3 – Identifica rosto do ator principal |
|  | | 4 – Recorta imagem utilizando as coordenadas do rosto identificado |
|  | | 5 – Compacta a imagem |
|  | | 6 – Envia a imagem às instâncias remotas |
|  | | 7 – Instância remota descompacta a imagem |
|  | | 8 – Instância remota aplica a imagem no rosto do avatar correspondente à instância onde a imagem fora capturada |

## Requisitos Não Funcionais

Quatro requisitos não funcionais foram identificados e alcançados. Eles dizem respeito ao tempo de resposta entre clientes, ao uso de memória principal, ao uso de memória secundária pelos executáveis e ao uso de recursos de processamento na instância em servidor.

### Tempo de resposta

Para oferecer as funcionalidades desejadas, é importante que a comunicação entre instâncias cliente seja a menor possível. Não é razoável decidir um valor fixo para o tempo de viagem dos pacotes, já que, num cenário real, este tempo vai depender de uma rede probabilística (a Internet), e da localização geográfica dos usuários conectados pelo sistema.

Entretanto, no quadro da demonstração que se pretende fazer ao final do projeto, espera-se que a latência entre os dois computadores conectados esteja na ordem de dezenas de milissegundos, enquanto a latência entre as instâncias cliente e a instância no servidor remoto, localizado em um *datacenter*, em São Paulo, seja de até 200 milissegundos.

### Uso de memória

O programa, tanto no servidor, quanto no cliente, não deve utilizar mais do que 500 MB de memória principal.

Além disso, dada a liberdade de manipulação de memória oferecida por C++, medidas específicas devem ser tomadas para evitar vazamento de memória(*memory leak*).

### Uso de espaço em disco

O arquivo de instalação não deve ultrapassar 200 MB.

O espaço ocupado pelo sistema cliente, uma vez instalado, não deve ultrapassar 400 MB.

### Uso de recursos de processamento no servidor

A instância em execução no servidor não deve ultrapassar 20% dos recursos de processamento da máquina virtual, exceto durante a instanciação do programa.

# Projeto Detalhado do Software

A partir de uma análise preliminar, decidiu-se dividir o projeto em duas grandes fases. Durante a primeira fase, ocorrem os esforços de pesquisa e desenvolvimento de um protótipo funcional que demonstre a viabilidade da proposta: um Produto Mínimo Viável, ou MVP. Nessa fase, deve-se demonstrar que as tecnologias escolhidas para o projeto são apropriadas e serão suficientes. Além disso, o MVP deverá conter funcionalidades centrais da ideia, da forma mais minimalista possível, para que o escopo permaneça concentrado nos desafios técnicos menos usuais e mais relevantes.

As funcionalidades centrais selecionadas são a transmissão de voz e expressão facial, assim como a simulação do ambiente em 3D. Essas são, portanto, as funcionalidades que devem estar no MVP.

Futuramente, a segunda fase, que está fora do escopo acadêmico do projeto, desenvolverá um produto pronto para o mercado, onde será necessário, entre outros, um sistema de autenticação, navegação de cursos disponíveis, uma agenda, um sistema de updates automáticos, um processo bem definido de desenvolvimento contínuo, testes automáticos e compatibilidade com os principais sistemas operacionais. Tais funcionalidades se somam em um projeto de grande envergadura, estão fora do escopo da primeira fase e, portanto, do presente projeto.

O MVP, desenvolvido em um semestre letivo, demonstra a ideia central e prepara o caminho para que o projeto continue, no semestre seguinte e/ou após a graduação. Os detalhes do projeto são expostos nos subtópicos a seguir.



## Arquitetura da aplicação proposta

O modelo arquitetural proposto para a solução AViS é híbrido. São associados o modelo cliente-servidor, tipicamente utilizado por *webapps*, e um modelo P2P distribuído, como em aplicativos *torrent*.

Seguindo o modelo cliente-servidor, é possível utilizar a *API Alloy*, que já está em produção e oferece uma parte essencial ao produto cuja viabilidade pretende-se demonstrar com este MVP. Além disso, um servidor remoto, imparcial face ao cliente AViS, vai arbitrar o diálogo de dados entre clientes.

Já com o que um modelo P2P oferece, o sistema AViS será capaz de transmitir dados sensíveis ao tempo com mais agilidade. *Buffers* de áudio e vídeo serão transmitidos, via protocolo UDP, seguindo o caminho mais curto, determinado pela infraestrutura de redes, entre um cliente e outro.

Na figura 1, observa-se uma representação global da comunicação entre diferentes instâncias do cliente AViS. A figura apresenta também a infraestrutura disponibilizada pela Alloy City Linguistics. O servidor assume um papel de árbitro entre os clientes conectados, mantendo-os atualizados quanto a dados relevantes para a experiência do usuário. Os dados mais importantes a serem considerados aqui são os IPs e portas de cada instância cliente conectada. Uma lista de endereços essencial para a comunicação direta (P2P) entre instâncias clientes.

Figura – Visão global da comunicação intra instâncias

A figura 2 apresenta uma visão global da arquitetura empregada no aplicativo cliente. Unreal Engine 4, tecnologia central no projeto, é responsável tanto pela renderização do ambiente virtual quanto pela sincronização de endereços. O módulo VoIP é responsável pela captura e envio de áudio entre clientes. O módulo FMoA é responsável pela captura da expressão facial do usuário.



Figura 3 - Arquitetura do cliente AViS

## Tecnologias utilizadas e APIs

Seguem as tecnologias chave utilizadas no projeto, assim como as principais razões por trás da decisão de utilizá-las.

### Tecnologias

* C++

Uma linguagem de programação com 34 anos de amadurecimento, C++ é o padrão de algumas das indústrias mais exigentes em matéria de *software*. Ela foi escolhida para este projeto pelas razões listadas abaixo:

* Oferece acesso de baixo nível aos recursos de sistema, particularmente à memória;
* Oferece abstrações de nível mais alto, como classes e mecanismos de iteração;
* Alta performance em tempo de execução (na mesma ordem de C e Rust);
* Desenvolvimento ativo (última *release* estável em dezembro de 2017, próxima prevista para 2020);
* Escolha padrão da maior indústria de entretenimento do mundo
* Rico legado.
* Unreal Engine 4

Originalmente um mecanismo de jogo (*game engine*), Unreal Engine 4 é um motor de *renderização* 3D em tempo real. Ele costuma ser manipulado via C++ e via UE Blueprints (um formato proprietário de programação visual). Essa ferramenta será responsável pela *renderização* tridimensional do ambiente de interação entre os usuários, abstraindo do projeto as complexidades matemáticas e físicas inerentes a simulações 3D.

Além disso, Unreal Engine contém um módulo responsável pela comunicação via UDP entre usuários, tecnologia essencial ao projeto. Aprender a usar essa parte do mecanismo demanda consideravelmente menos tempo e esforço do que desenvolver a funcionalidade integralmente.

Unreal Engine 4 se apoia em 21 anos de amadurecimento e é usada hoje por desenvolvedores de jogos, artistas 3D, estúdios de arquitetura, estúdios de efeitos especiais, pela indústria automobilística, por estudantes de C++, entre outros. A tecnologia é desenvolvida por Epic Games, sob uma licença de código fonte acessível e de uso educacional livre.

* OpenCV

OpenCV (Open Source Computer Vision) é uma biblioteca de *computer vision*. Inicialmente, foi desenvolvida pela Intel, mas hoje é mantida por uma ampla comunidade de programadores independentes, empresas e universidades, sob a licença aberta BSD. O desenvolvimento está ativo, com o último lançamento estável em julho de 2019.

No projeto AViS, OpenCV será usada, sobretudo, para definir as coordenadas do rosto do usuário em cada quadro do fluxo de vídeo.

* Blender

Programa de modelagem 3D de código fonte aberto, Blender pode ser usado para a criação de modelos estáticos 3D (*meshes*). Especificamente neste projeto, Blender é utilizado para a criação do avatar utilizado pelo programa para representar os usuários no ambiente virtual.

### APIs Utilizadas

A API do mecanismo de renderização é a interface entre o *software* do projeto e UE4[[1]](#footnote-1). Esta API é exaustivamente documentada no domínio <https://docs.unrealengine.com> e é acessível via C++, UE4 Blueprints ou Python. Neste projeto, o acesso será feito, majoritariamente, via C++.

Acesso à API Alloy (<https://alloy.city>), usada no aplicativo web <https://pantoufle.online>, será demonstrado, mas o uso da API está fora do escopo deste projeto, visto que só será necessário futuramente, caso seja demonstrada a viabilidade tecnológica da ideia aqui explorada.

Ambas as APIs são de acesso local. A API UE4 é acessível por meio da inclusão de arquivos de interface nos programas desenvolvidos. A API Alloy é acessível por meio de chamadas HTTP locais, realizadas exclusivamente pela instância servidor.

## Componentes do SW



Figura - Diagrama de Componentes

## Diagrama de Classes

## Considerações sobre o Banco de Dados Utilizado

O sistema utilizado pelo aplicativo <https://pantoufle.online> atualmente se apoia em um banco de dados NoSQL MongoDB, conectado à API Alloy via Mongoose JS. Futuramente, seria necessário acessar esse banco de dados pela referida API. Por isso, o MVP demonstra a habilidade de realizar essa conexão, mas julgamos desnecessário, tendo em vista o escopo do projeto, mapear ou modelar o banco de dados, tal qual ele existe em produção hoje. Primeiramente, porque o contato com o banco é intermediado pela API. Basta conhecer a interface da API e o acesso é feito sem maiores complicações, conforme demonstra o MVP. Além disso, o MVP vai demonstrar capacidades técnicas efêmeras, que utilizam a memória principal, apenas. Não será necessário, no escopo do MVP, persistir dados gerados ou coletados durante a execução do programa, tanto na instância em servidor quanto nas instâncias clientes.

## Diagramas de Sequência

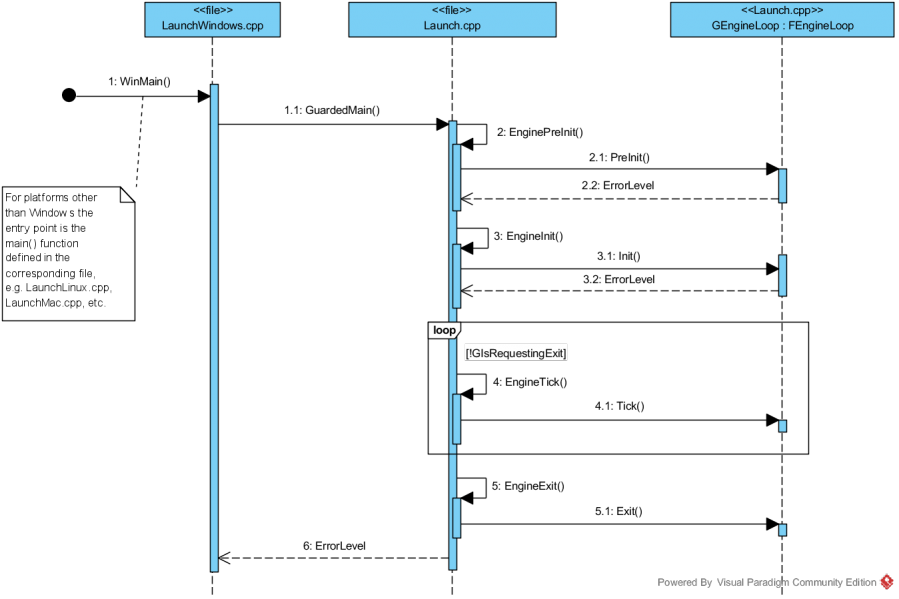


Figura - Sequência de Inicialização do mecanismo Unreal Engine 4

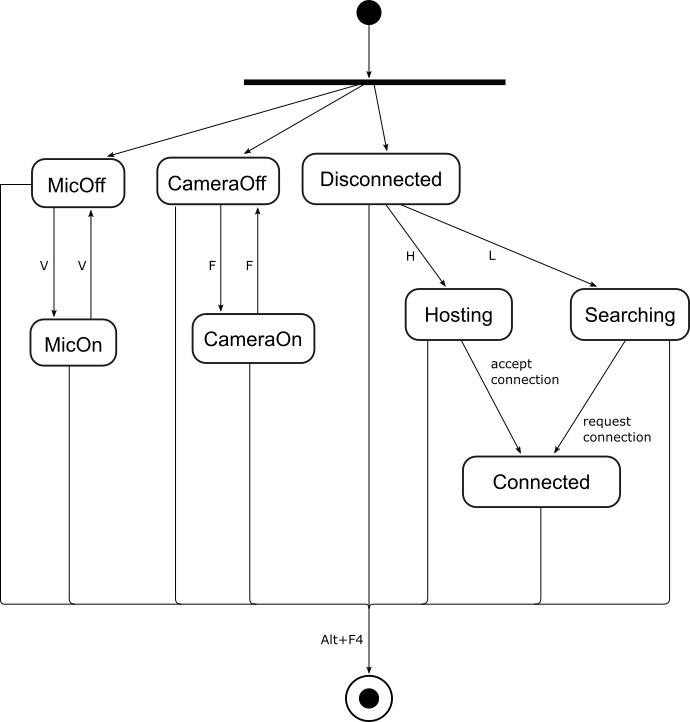


Figura - Diagrama de Sequência – TurnCameraOn



Figura - Diagrama de Sequência - TurnCameraOff

## Diagrama Estados



## Interfaces com o usuário

A interface gráfica do MVP consiste no ambiente 3D propriamente dito, já que as funcionalidades que se pretende demonstrar podem ser acessíveis via linha de comando, com atalhos de teclado provisórios, ou mesmo automatizadas e otimizadas para o ambiente de demonstração. Juntamente com esse ambiente simulado, o MVP conta com uma janela para inspecionar o trabalho do algoritmo de identificação de rostos, conforme figura 8. Esta janela de inspeção pode ser instanciada com a tecla B ou com o comando *StartVideoCaptureDebugger*.

Figura - Debugger do algoritmo de identificação de rostos

Após a implementação do MVP, na segunda fase do desenvolvimento, será necessário implementar interfaces que cubram o mesmo conjunto de funcionalidades oferecido pelo aplicativo Web atualmente em produção.

# Implementação

O código fonte do projeto está integralmente disponível no GitHub, assim como *releases* pré-compiladas, no seguinte endereço: <https://github.com/alloy-city/AViS>



## Modelagem 3D

Para implementar o protótipo de demonstração, ou MVP, foi necessário estudar as partes relevantes da vasta documentação de Unreal Engine 4. Essa documentação cobre quase toda a API exposta pelo mecanismo de jogo. Eventualmente, foi necessária a utilização de métodos e atributos, as vezes classes inteiras, que não são mencionadas na documentação. Nesses casos raros, é sempre possível consultar o código, diretamente, que, embora não seja totalmente aberto, é acessível para consulta. Mesmo quando a documentação cobre o recurso utilizado, a maneira mais efetiva se assegurar a compreensão sobre como UE4 funciona é através do código fonte. O programa AViS é, portanto, desenvolvido ao lado do código do mecanismo UE4, em uma única solução no Visual Studio.

A modelagem do ambiente virtual, que representa uma sala de aula, foi feita no próprio Editor UE4, utilizando modelos geométricos simples, disponíveis em qualquer instalação recente do mecanismo. Para essa tarefa da implementação, foi necessário estudar como UE4 simula geometria 3D, iluminação, texturas e materiais.

Em seguida, implementou-se um avatar, controlável pelo usuário, com base na classe *Pawn* exposta pelo mecanismo de jogo. Essa classe conta com um ponto de vista na sala virtual, que pode ser manipulado pelos mecanismos de entrada de comandos, como mouse e teclado.

Para que a esfera, que representa a cabeça do avatar, seja devidamente capaz de comportar o mapa de bits correspondente a um quadro de rosto, recebido pela rede, foi necessário realizar um procedimento chamado *UV unwrapping*. Este procedimento associa coordenadas de renderização de texturas a um modelo 3D. Neste caso em particular, o modelo 3D é uma esfera, mas as texturas são mapas de bits quadrados, de 128 por 128 pontos. Foi preciso portanto definir as coordenadas das texturas na esfera de forma que o rosto sofra a menor deformação geométrica possível. Esta tarefa foi realizada no software de modelagem 3D Blender, apresentado em maiores detalhes na seção 5.2.1 - Tecnologias.

## Captura e Processamento de Vídeo

Para ter acesso à webcam, integrou-se ao projeto o software de visão de máquina OpenCV. O primeiro desafio dessa tarefa foi aprender a utilizar as funcionalidades mais elementares do OpenCV, isoladamente. A saber, como capturar o vídeo da webcam e apresentá-lo, sem modificação alguma, de volta ao usuário, em uma nova janela. Em seguida, ao tentar integrar o OpenCV a um projeto baseado em UE4, descobriu-se algumas colisões entre as duas ferramentas. Uma função OpenCV tem o mesmo nome de uma classe da biblioteca de funções matemáticas Kismet Math, utilizada pelo mecanismo UE4. Em iterações futuras do projeto, conflitos como este poderão ser evitados através do mecanismo de subsistemas de UE4. Este mecanismo é usado internamente para estruturar os diferentes componentes UE4. É possível utilizar essa mesma estrutura para integrar OpenCV ao projeto, criando para ele um espaço de memória privado, diferente do espaço global. Essa solução, entretanto, requer a reestruturação de uma parte substancial da implementação. Optou-se por renomear a função no código fonte do OpenCV e utilizar essa versão alterada do programa. É preciso manter em vista essa modificação e implementar uma solução definitiva antes de tentar atualizar OpenCV para uma versão mais recente.

Uma vez que OpenCV e UE4 estavam trabalhando juntos, em um mesmo projeto, implementaram-se, um a um, os métodos responsáveis por tratar o fluxo de vídeo no cliente de origem. A saber:

1. captura-se um quadro;
2. identifica-se as coordenadas e o raio de um rosto no quadro;
3. recorta-se o quadro utilizando as coordenadas encontradas;
4. redimensiona-se o rosto para 128 por 128 pontos;
5. compacta-se o rosto em JPG com índice de qualidade 80; e
6. envia-se o mapa de bits compactado ao objeto responsável por comunicação em rede.

Este processo foi implementado na classe Webcam

A classe StreamService é responsável por enviar os quadros processados pela rede. Essa classe foi projetada para processar 30 quadros por segundo, o que a classifica como ponto potencial de estrangulamento de performance. Ela deve iniciar um loop de observação, para aguardar conexões de rede. Para que o programa não pare até que uma conexão seja estabelecida, o loop de observação é executado em um thread paralelo ao programa principal.

Os bits correspondentes ao rosto são enviados à máquina remota. Uma vez recebidos pela máquina de destino, uma função exposta pela API UE4 chamada ImportBufferAsTexture2D é utilizada para descompactar a imagem JPG. Essa função recebe um buffer correspondente a uma imagem, expande a imagem caso ela esteja compactada, e cria um objeto da classe nativa Texture2D. A cada quadro recebido pela rede, essa função é chamada. Se ela retornar um objeto da classe Texture2D válido, aplica-se a textura ao avatar correspondente ao usuário que gerou o quadro em questão.

Todo esse processo cria a impressão de que um determinado avatar, presente na sala de aula virtual, tem o rosto do usuário que o controla.

## Captura e Processamento de Áudio

Finalmente, para implementar a funcionalidade de transmissão de voz, foi necessário aprender a utilizar os recursos de processamento de áudio do mecanismo de jogo. O primeiro passo é ganhar acesso ao microfone do usuário através da ativação do plugin Voice, desenvolvido também pela Epic Games. Ao assegurar que os usuários estão em uma mesma sessão (USession), UE4 se encarrega de enviar os pacotes de áudio compactados, para todos os usuários conectados à mesma sessão. Entretanto, para aumentar a sensação de presença, alterou-se o comportamento do mecanismo para que a voz, no usuário remoto, seja reproduzida a partir de um ponto específico na sala de aula virtual, e não como um som ambiente independente da simulação. Para isso, utilizou-se a classe nativa USoundCue, que pode ser posicionada numa cena virtual e emitir sons. Os sons emitidos por um objeto dessa classe contam com diversos recursos de processamento de som em tempo de execução, entre eles atenuação diferenciada entre os dois canais. O mecanismo UE4 reproduz os sons do programa em execução em dois canais: esquerdo e direito. Ao atenuar por exemplo o lado direito mais do que o esquerdo ao reproduzir um determinado som, o usuário tem a sensação de que o som está vindo do seu lado esquerdo.

Para usufruir dos recursos de atenuação diferenciada oferecidos pela classe USoundCue, instancia-se um objeto dessa classe juntamente com o avatar, mantendo os atributos de coordenadas espaciais do objeto sincronizados com as coordenadas da cabeça do avatar. Durante a instanciação do avatar e do seu componente de som, define-se uma relação de parentalidade espacial entre os dois objetos. Dessa forma, os cálculos dessa sincronização são delegados ao mecanismo de jogo.

Alcançou-se todos esses passos de implementação na versão 0.5 do cliente AViS. Tanto o código fonte quanto um arquivo compactado contendo todos os arquivos necessários para a demonstração da tecnologia estão disponíveis no repositório do projeto.

# Projeto de Teste

Em um contexto futuro, a equipe recomenda o uso do subsistema de testes automáticos providos pela Epic, nativamente, no mecanismo Unreal Engine 4, chamado *UE4 Functional Testing Framework*. No contexto do presente MVP, verifica-se manualmente que todos os requisitos funcionais respondem apropriadamente seguindo a lista de ações abaixo:

1. Conectar os PCs 1 e 2 em uma mesma rede local;
2. Lançar o executável AViSClient.exe no PC 1;
3. Lançar o executável AViSClient.exe no PC 2;
4. Pressionar H no PC 1 para iniciar o serviço de hospedagem;
5. Pressionar L no PC 2 para iniciar a busca de serviços de hospedagem na rede local;
6. Aguardar a conexão;
7. Pressionar F em ambos os PCs para acionar o subsistema FoIP; e
8. Pressionar V em ambos os PCs para acionar o subsistema VoIP;

O rosto do usuário do PC 1 deve ser visível no PC 2. O rosto do usuário do PC 2, por sua vez, deve ser visível no PC 1.

Para verificar o funcionamento correto do subsistema VoIP, recomenda-se os seguintes passos, após a conclusão dos passos acima:

1. Produzir diante do PC 1 um som inteligível e relativamente contínuo, como música ou a leitura de um parágrafo;
2. Movimentar o avatar do PC 2 ao redor do avatar remoto.

O som reproduzido pelo PC 2 não pode ser contínuo, como a fonte. Ao se afastar da posição do avatar remoto, o volume de som deve diminuir. Ao se aproximar, o volume deve aumentar. Ao posicionar lateralmente a cabeça do avatar do PC 2, o som deve ser modulado diferentemente entre os dois canais de áudio (esquerdo e direito). Com isso, verifica-se que o subsistema VoIP está funcionando apropriadamente.

# Instalação do Software

O MVP exige que os binários do OpenCV estejam disponíveis no Path do Windows. Além disso, o arquivo cascada de aprendizado de máquina para identificação da posição do rosto no quadro precisa estar em um caminho específico. Portanto, os passos seguintes devem ser seguidos precisamente, em um ambiente Windows 10, para que a demonstração ocorra normalmente.

1. Baixar o arquivo zip da versão 0.5;
2. descompactar o arquivo zip;
3. criar a estrutura de pastas C:/AViS/Plugins/OpenCV/Resources/Data/haarcascades/;
4. copiar o arquivo haarcascade\_frontalface\_default.xml para a pasta criada no passo 3;
5. incluir as bibliotecas OpenCV opencv\_videoio\_ffmpeg411.dll e opencv\_videoio\_ffmpeg411\_64.dll ao Path do Windows;
6. lançar a primeira instância do executável AViS.exe em uma das máquinas disponíveis para a demonstração;
7. pressionar a tecla H para começar o serviço local de recepção;
8. lançar a segunda instância do executável, em uma segunda máquina;
9. em ambas as máquinas, pressionar a tecla F para iniciar o processo de captura de rosto; e
10. em ambas as máquinas, pressionar a tecla V para iniciar o processo de captura de voz.

# Análise dos Resultados

# Conclusão

* Referências

WELLER, Martin (2007). *Virtual learning environments: using, choosing and developing your VLE*. London: Routledge. pp. 4–5. ISBN 9780415414302.

KARDASHEV, Nikolai (1964). *Transmission of Information by Extraterrestrial Civilizations*. Soviet Astronomy.

* Glossário

**Student**: usuário com nível de acesso 0;

**Teacher**: usuário com nível de acesso 1;

**Creator**: usuário com nível de acesso 2;

**Coordinator**: usuário com nível de acesso 3;

**Admin**: usuário com nível de acesso 4;

**Resource**: menor unidade do material didático;

**Lesson**: conjunto de Resources;

**Chapter**: conjunto de Lessons;

**Meeting**: par de momentos no tempo, definidos em UTC, que representam uma aula ao vivo;

**Course**: conjunto de Meetings

**Pack**: conjunto de Products (exceto outros Packs)

**Product**: unidade comercializável (Meetings, Courses, Lessons, Chapters and Packs);

**CR**: Cliente AViS remoto

**CL**: Cliente AViS local

**Webapp**: aplicativo projetado para funcionar em um ambiente provido por um navegador web.

**P2P**: *Pear to Pear*; estratégia de comunicação em rede que envolve duas instâncias remotas equivalentes, sem a intermediação de um servidor

**Torrent**: protocolo de transmissão de dados *P2P*.

**API Alloy**: Interface de acesso à base dados da empresa Alloy City Linguistics. Esta API foi desenvolvida por um dos integrantes do presente trabalho e está na base do principal aplicativo mantido pela empresa, o webapp Pantoufle (https://pantoufle.online).

* Apêndice

Anexo 1 – Termo de Compromisso da escola Pantoufle

seção 1

A Escola de Francês Pantoufle se compromete a:

1. prover os encontros online listados acima, se houver, de 50 minutos cada, de aqui em diante chamados AULAS, ao longo de até 6 (seis) semanas, sempre na presença de um colaborador experiente em ensino do francês, de aqui em diante chamado facilitador;
2. repor AULAS inviabilizadas por ausência do facilitador ou falha na conexão à Internet do facilitador ou dos servidores da Alloy City Linguistics;
3. acompanhar e assistir o usuário em seu aprendizado ao longo do módulo;
4. avaliar o nível alcançado pelo aluno ao final do nível; e
5. prover acesso ao capítulos e/ou lições listados acima, se houver, após confirmação do pagamento.

seção 2

Eu, <nome\_do\_aluno>, me comprometo a:

1. realizar o pagamento deste módulo até a data acordada da primeira AULA;
2. comparecer a pelo menos 75% das AULAS;
3. realizar pelo menos 75% das atividades (missões);
4. adicionar o facilitador no Google Hangouts ou no Skype, conforme o caso, a fim de poder participar das aulas; e
5. instalar pelo menos um dos navegadores compatíveis (Chrome, Firefox, Edge e Safari).

seção 3

Atenção!

* A Escola de Francês Pantoufle não se responsabilizará por falhas e interrupções de serviços diversos (eletricidade, conexão à Internet, água, transporte, etc.), prestados ao aluno, que venham impossibilitá-lo de participar de uma ou mais aulas.
* No caso da modalidade Français Exclusivité, cancelamentos só resultarão em redefinição de horários sem prejuízo ao cliente se requisitados com pelo menos 24 horas de antecedência.
* No caso das modalidades Français à Deux ou Regular, cancelamentos só resultarão em redefinição de horários sem prejuízo ao cliente, se requisitados com pelo menos 24 horas de antecedência, sob condição do restante do grupo concordar com a mudança.
* Cancelamentos com menos de 24 horas de antecedência não darão direito a reposição de aula nem a reembolso, parcial ou total.
* Cancelamentos, em geral, não revogam o direito do aluno a acesso ao material didático correspondente ao módulo contratado.

Anexo 2 – Acordo de Cooperação entre o Centro Paula Souza e a Microsoft Brasil

Parceria incluir escolas do Centro Paula Souza nos programas acadêmicos da Microsoft Brasil que oferece tecnologia e treinamento de professores

A Microsoft Brasil e o Centro Paula Souza, autarquia especial do Governo do Estado de São Paulo ligada à Secretaria de Desenvolvimento, assinam nesta sexta-feira, dia 21 de setembro de 2007, um acordo de cooperação que irá beneficiar as Fatecs e Etecs com as iniciativas do programa de parcerias acadêmicas que a empresa mantém no país. A parceria formalizada às 10h30 no Palácio dos Bandeirantes contou com a presença do governador em exercício do Estado de São Paulo, Alberto Goldman, do vice-presidente mundial para o setor público, Michel Van der Belt, do presidente da Microsoft Brasil, Michel Levy e da diretora-superintendente do Centro Paula Souza, Laura Laganá.

Pelo acordo, a Microsoft Brasil irá investir R$ 2 milhões em doação de softwares e fornecimento de material didático que será usado para a capacitação dos professores de informática e web design das escolas administradas pelo Centro Paula Souza. A expectativa é beneficiar mais de 20 mil alunos dessas áreas com os programas educacionais da empresa por meio da oferta de conteúdo didático e suporte técnico básico para projetos de inclusão digital. A parceria também beneficia as escolas técnicas da instituição com o programa acadêmico MSDNAA, que dá acesso as ferramentas de desenvolvimento de software da Microsoft para uso educacional e de pesquisa.

“As Fatecs e Etecs do Centro Paula Souza são instrumentos fundamentais para o desenvolvimento de São Paulo, pois qualificam os jovens e são fatores de atração dos investimentos que darão oportunidades a esses mesmos jovens”, disse o governador em exercício Alberto Goldman. “É muito gratificante poder contar com uma empresa do porte da Microsoft como parceira, acreditando em um projeto do Governo de São Paulo que combina desenvolvimento com inclusão social”, completou.

“A Microsoft tem um compromisso de longo prazo com o Brasil e a meta de tornar a tecnologia cada vez mais acessível, disponível e relevante. Acreditamos que aliar tecnologia e educação é a melhor forma de ajudar as pessoas a atingirem o seu potencial e, consequentemente, construir um país mais justo do ponto de vista social e mais produtivo e competitivo no âmbito econômico”, afirmou o presidente da Microsoft Brasil, Michel Levy. “Com esse acordo, vamos facilitar o acesso dos professores e dos estudantes à tecnologia de ponta desenvolvida pela Microsoft e ensinar tecnologia básica nas escolas de ensino médio do Centro Paula Souza”, concluiu Levy.

“Entre as diversas parcerias que já fizemos com a Mircrosoft, esta é uma que se destaca pela abrangência e pela qualidade. É uma oportunidade preciosa para nossos professores e alunos terem acesso às mais atuais tecnologias, postas a serviço da melhoria do ensino”, diz Laura Laganá.

O programa de parcerias acadêmicas é uma ação da iniciativa Potencial Ilimitado da Microsoft, criada em 2003 para reforçar o compromisso de longo prazo da empresa com o Brasil. O Potencial Ilimitado reflete a crença da Microsoft de que por meio do acesso à tecnologias inovadoras e de parcerias locais é possível transformar a educação, incentivar a inovação e gerar oportunidades de emprego, estabelecendo um ciclo continuo de sustentabilidade social e econômica para o país.

Anexo 3 – License Agreement For Open Source Computer Vision Library

License Agreement For Open Source Computer Vision Library

(3-clause BSD License)

Copyright (C) 2000-2019, Intel Corporation, all rights reserved.

Copyright (C) 2009-2011, Willow Garage Inc., all rights reserved.

Copyright (C) 2009-2016, NVIDIA Corporation, all rights reserved.

Copyright (C) 2010-2013, Advanced Micro Devices, Inc., all rights reserved.

Copyright (C) 2015-2016, OpenCV Foundation, all rights reserved.

Copyright (C) 2015-2016, Itseez Inc., all rights reserved.

Third party copyrights are property of their respective owners.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the names of the copyright holders nor the names of the contributors may be used to endorse or promote products derived from this software without specific prior written permission.

This software is provided by the copyright holders and contributors “as is” and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall copyright holders or contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

Anexo 4 - GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies

of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for

software and other kinds of works.

The licenses for most software and other practical works are designed

to take away your freedom to share and change the works. By contrast,

the GNU General Public License is intended to guarantee your freedom to

share and change all versions of a program--to make sure it remains free

software for all its users. We, the Free Software Foundation, use the

GNU General Public License for most of our software; it applies also to

any other work released this way by its authors. You can apply it to

your programs, too.

When we speak of free software, we are referring to freedom, not

price. Our General Public Licenses are designed to make sure that you

have the freedom to distribute copies of free software (and charge for

them if you wish), that you receive source code or can get it if you

want it, that you can change the software or use pieces of it in new

free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you

these rights or asking you to surrender the rights. Therefore, you have

certain responsibilities if you distribute copies of the software, or if

you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether

gratis or for a fee, you must pass on to the recipients the same

freedoms that you received. You must make sure that they, too, receive

or can get the source code. And you must show them these terms so they

know their rights.

Developers that use the GNU GPL protect your rights with two steps:

(1) assert copyright on the software, and (2) offer you this License

giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains

that there is no warranty for this free software. For both users' and

authors' sake, the GPL requires that modified versions be marked as

changed, so that their problems will not be attributed erroneously to

authors of previous versions.

Some devices are designed to deny users access to install or run

modified versions of the software inside them, although the manufacturer

can do so. This is fundamentally incompatible with the aim of

protecting users' freedom to change the software. The systematic

pattern of such abuse occurs in the area of products for individuals to

use, which is precisely where it is most unacceptable. Therefore, we

have designed this version of the GPL to prohibit the practice for those

products. If such problems arise substantially in other domains, we

stand ready to extend this provision to those domains in future versions

of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents.

States should not allow patents to restrict development and use of

software on general-purpose computers, but in those that do, we wish to

avoid the special danger that patents applied to a free program could

make it effectively proprietary. To prevent this, the GPL assures that

patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and

modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of

works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this

License. Each licensee is addressed as "you". "Licensees" and

"recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work

in a fashion requiring copyright permission, other than the making of an

exact copy. The resulting work is called a "modified version" of the

earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based

on the Program.

To "propagate" a work means to do anything with it that, without

permission, would make you directly or secondarily liable for

infringement under applicable copyright law, except executing it on a

computer or modifying a private copy. Propagation includes copying,

distribution (with or without modification), making available to the

public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other

parties to make or receive copies. Mere interaction with a user through

a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices"

to the extent that it includes a convenient and prominently visible

feature that (1) displays an appropriate copyright notice, and (2)

tells the user that there is no warranty for the work (except to the

extent that warranties are provided), that licensees may convey the

work under this License, and how to view a copy of this License. If

the interface presents a list of user commands or options, such as a

menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work

for making modifications to it. "Object code" means any non-source

form of a work.

A "Standard Interface" means an interface that either is an official

standard defined by a recognized standards body, or, in the case of

interfaces specified for a particular programming language, one that

is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other

than the work as a whole, that (a) is included in the normal form of

packaging a Major Component, but which is not part of that Major

Component, and (b) serves only to enable use of the work with that

Major Component, or to implement a Standard Interface for which an

implementation is available to the public in source code form. A

"Major Component", in this context, means a major essential component

(kernel, window system, and so on) of the specific operating system

(if any) on which the executable work runs, or a compiler used to

produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all

the source code needed to generate, install, and (for an executable

work) run the object code and to modify the work, including scripts to

control those activities. However, it does not include the work's

System Libraries, or general-purpose tools or generally available free

programs which are used unmodified in performing those activities but

which are not part of the work. For example, Corresponding Source

includes interface definition files associated with source files for

the work, and the source code for shared libraries and dynamically

linked subprograms that the work is specifically designed to require,

such as by intimate data communication or control flow between those

subprograms and other parts of the work.

The Corresponding Source need not include anything that users

can regenerate automatically from other parts of the Corresponding

Source.

The Corresponding Source for a work in source code form is that

same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of

copyright on the Program, and are irrevocable provided the stated

conditions are met. This License explicitly affirms your unlimited

permission to run the unmodified Program. The output from running a

covered work is covered by this License only if the output, given its

content, constitutes a covered work. This License acknowledges your

rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not

convey, without conditions so long as your license otherwise remains

in force. You may convey covered works to others for the sole purpose

of having them make modifications exclusively for you, or provide you

with facilities for running those works, provided that you comply with

the terms of this License in conveying all material for which you do

not control copyright. Those thus making or running the covered works

for you must do so exclusively on your behalf, under your direction

and control, on terms that prohibit them from making any copies of

your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under

the conditions stated below. Sublicensing is not allowed; section 10

makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological

measure under any applicable law fulfilling obligations under article

11 of the WIPO copyright treaty adopted on 20 December 1996, or

similar laws prohibiting or restricting circumvention of such

measures.

When you convey a covered work, you waive any legal power to forbid

circumvention of technological measures to the extent such circumvention

is effected by exercising rights under this License with respect to

the covered work, and you disclaim any intention to limit operation or

modification of the work as a means of enforcing, against the work's

users, your or third parties' legal rights to forbid circumvention of

technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you

receive it, in any medium, provided that you conspicuously and

appropriately publish on each copy an appropriate copyright notice;

keep intact all notices stating that this License and any

non-permissive terms added in accord with section 7 apply to the code;

keep intact all notices of the absence of any warranty; and give all

recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey,

and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to

produce it from the Program, in the form of source code under the

terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified

it, and giving a relevant date.

b) The work must carry prominent notices stating that it is

released under this License and any conditions added under section

7. This requirement modifies the requirement in section 4 to

"keep intact all notices".

c) You must license the entire work, as a whole, under this

License to anyone who comes into possession of a copy. This

License will therefore apply, along with any applicable section 7

additional terms, to the whole of the work, and all its parts,

regardless of how they are packaged. This License gives no

permission to license the work in any other way, but it does not

invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display

Appropriate Legal Notices; however, if the Program has interactive

interfaces that do not display Appropriate Legal Notices, your

work need not make them do so.

A compilation of a covered work with other separate and independent

works, which are not by their nature extensions of the covered work,

and which are not combined with it such as to form a larger program,

in or on a volume of a storage or distribution medium, is called an

"aggregate" if the compilation and its resulting copyright are not

used to limit the access or legal rights of the compilation's users

beyond what the individual works permit. Inclusion of a covered work

in an aggregate does not cause this License to apply to the other

parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms

of sections 4 and 5, provided that you also convey the

machine-readable Corresponding Source under the terms of this License,

in one of these ways:

a) Convey the object code in, or embodied in, a physical product

(including a physical distribution medium), accompanied by the

Corresponding Source fixed on a durable physical medium

customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product

(including a physical distribution medium), accompanied by a

written offer, valid for at least three years and valid for as

long as you offer spare parts or customer support for that product

model, to give anyone who possesses the object code either (1) a

copy of the Corresponding Source for all the software in the

product that is covered by this License, on a durable physical

medium customarily used for software interchange, for a price no

more than your reasonable cost of physically performing this

conveying of source, or (2) access to copy the

Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the

written offer to provide the Corresponding Source. This

alternative is allowed only occasionally and noncommercially, and

only if you received the object code with such an offer, in accord

with subsection 6b.

d) Convey the object code by offering access from a designated

place (gratis or for a charge), and offer equivalent access to the

Corresponding Source in the same way through the same place at no

further charge. You need not require recipients to copy the

Corresponding Source along with the object code. If the place to

copy the object code is a network server, the Corresponding Source

may be on a different server (operated by you or a third party)

that supports equivalent copying facilities, provided you maintain

clear directions next to the object code saying where to find the

Corresponding Source. Regardless of what server hosts the

Corresponding Source, you remain obligated to ensure that it is

available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided

you inform other peers where the object code and Corresponding

Source of the work are being offered to the general public at no

charge under subsection 6d.

A separable portion of the object code, whose source code is excluded

from the Corresponding Source as a System Library, need not be

included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any

tangible personal property which is normally used for personal, family,

or household purposes, or (2) anything designed or sold for incorporation

into a dwelling. In determining whether a product is a consumer product,

doubtful cases shall be resolved in favor of coverage. For a particular

product received by a particular user, "normally used" refers to a

typical or common use of that class of product, regardless of the status

of the particular user or of the way in which the particular user

actually uses, or expects or is expected to use, the product. A product

is a consumer product regardless of whether the product has substantial

commercial, industrial or non-consumer uses, unless such uses represent

the only significant mode of use of the product.

"Installation Information" for a User Product means any methods,

procedures, authorization keys, or other information required to install

and execute modified versions of a covered work in that User Product from

a modified version of its Corresponding Source. The information must

suffice to ensure that the continued functioning of the modified object

code is in no case prevented or interfered with solely because

modification has been made.

If you convey an object code work under this section in, or with, or

specifically for use in, a User Product, and the conveying occurs as

part of a transaction in which the right of possession and use of the

User Product is transferred to the recipient in perpetuity or for a

fixed term (regardless of how the transaction is characterized), the

Corresponding Source conveyed under this section must be accompanied

by the Installation Information. But this requirement does not apply

if neither you nor any third party retains the ability to install

modified object code on the User Product (for example, the work has

been installed in ROM).

The requirement to provide Installation Information does not include a

requirement to continue to provide support service, warranty, or updates

for a work that has been modified or installed by the recipient, or for

the User Product in which it has been modified or installed. Access to a

network may be denied when the modification itself materially and

adversely affects the operation of the network or violates the rules and

protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided,

in accord with this section must be in a format that is publicly

documented (and with an implementation available to the public in

source code form), and must require no special password or key for

unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this

License by making exceptions from one or more of its conditions.

Additional permissions that are applicable to the entire Program shall

be treated as though they were included in this License, to the extent

that they are valid under applicable law. If additional permissions

apply only to part of the Program, that part may be used separately

under those permissions, but the entire Program remains governed by

this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option

remove any additional permissions from that copy, or from any part of

it. (Additional permissions may be written to require their own

removal in certain cases when you modify the work.) You may place

additional permissions on material, added by you to a covered work,

for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you

add to a covered work, you may (if authorized by the copyright holders of

that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the

terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or

author attributions in that material or in the Appropriate Legal

Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or

requiring that modified versions of such material be marked in

reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or

authors of the material; or

e) Declining to grant rights under trademark law for use of some

trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that

material by anyone who conveys the material (or modified versions of

it) with contractual assumptions of liability to the recipient, for

any liability that these contractual assumptions directly impose on

those licensors and authors.

All other non-permissive additional terms are considered "further

restrictions" within the meaning of section 10. If the Program as you

received it, or any part of it, contains a notice stating that it is

governed by this License along with a term that is a further

restriction, you may remove that term. If a license document contains

a further restriction but permits relicensing or conveying under this

License, you may add to a covered work material governed by the terms

of that license document, provided that the further restriction does

not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you

must place, in the relevant source files, a statement of the

additional terms that apply to those files, or a notice indicating

where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the

form of a separately written license, or stated as exceptions;

the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly

provided under this License. Any attempt otherwise to propagate or

modify it is void, and will automatically terminate your rights under

this License (including any patent licenses granted under the third

paragraph of section 11).

However, if you cease all violation of this License, then your

license from a particular copyright holder is reinstated (a)

provisionally, unless and until the copyright holder explicitly and

finally terminates your license, and (b) permanently, if the copyright

holder fails to notify you of the violation by some reasonable means

prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is

reinstated permanently if the copyright holder notifies you of the

violation by some reasonable means, this is the first time you have

received notice of violation of this License (for any work) from that

copyright holder, and you cure the violation prior to 30 days after

your receipt of the notice.

Termination of your rights under this section does not terminate the

licenses of parties who have received copies or rights from you under

this License. If your rights have been terminated and not permanently

reinstated, you do not qualify to receive new licenses for the same

material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or

run a copy of the Program. Ancillary propagation of a covered work

occurring solely as a consequence of using peer-to-peer transmission

to receive a copy likewise does not require acceptance. However,

nothing other than this License grants you permission to propagate or

modify any covered work. These actions infringe copyright if you do

not accept this License. Therefore, by modifying or propagating a

covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically

receives a license from the original licensors, to run, modify and

propagate that work, subject to this License. You are not responsible

for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an

organization, or substantially all assets of one, or subdividing an

organization, or merging organizations. If propagation of a covered

work results from an entity transaction, each party to that

transaction who receives a copy of the work also receives whatever

licenses to the work the party's predecessor in interest had or could

give under the previous paragraph, plus a right to possession of the

Corresponding Source of the work from the predecessor in interest, if

the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the

rights granted or affirmed under this License. For example, you may

not impose a license fee, royalty, or other charge for exercise of

rights granted under this License, and you may not initiate litigation

(including a cross-claim or counterclaim in a lawsuit) alleging that

any patent claim is infringed by making, using, selling, offering for

sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this

License of the Program or a work on which the Program is based. The

work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims

owned or controlled by the contributor, whether already acquired or

hereafter acquired, that would be infringed by some manner, permitted

by this License, of making, using, or selling its contributor version,

but do not include claims that would be infringed only as a

consequence of further modification of the contributor version. For

purposes of this definition, "control" includes the right to grant

patent sublicenses in a manner consistent with the requirements of

this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free

patent license under the contributor's essential patent claims, to

make, use, sell, offer for sale, import and otherwise run, modify and

propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express

agreement or commitment, however denominated, not to enforce a patent

(such as an express permission to practice a patent or covenant not to

sue for patent infringement). To "grant" such a patent license to a

party means to make such an agreement or commitment not to enforce a

patent against the party.

If you convey a covered work, knowingly relying on a patent license,

and the Corresponding Source of the work is not available for anyone

to copy, free of charge and under the terms of this License, through a

publicly available network server or other readily accessible means,

then you must either (1) cause the Corresponding Source to be so

available, or (2) arrange to deprive yourself of the benefit of the

patent license for this particular work, or (3) arrange, in a manner

consistent with the requirements of this License, to extend the patent

license to downstream recipients. "Knowingly relying" means you have

actual knowledge that, but for the patent license, your conveying the

covered work in a country, or your recipient's use of the covered work

in a country, would infringe one or more identifiable patents in that

country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or

arrangement, you convey, or propagate by procuring conveyance of, a

covered work, and grant a patent license to some of the parties

receiving the covered work authorizing them to use, propagate, modify

or convey a specific copy of the covered work, then the patent license

you grant is automatically extended to all recipients of the covered

work and works based on it.

A patent license is "discriminatory" if it does not include within

the scope of its coverage, prohibits the exercise of, or is

conditioned on the non-exercise of one or more of the rights that are

specifically granted under this License. You may not convey a covered

work if you are a party to an arrangement with a third party that is

in the business of distributing software, under which you make payment

to the third party based on the extent of your activity of conveying

the work, and under which the third party grants, to any of the

parties who would receive the covered work from you, a discriminatory

patent license (a) in connection with copies of the covered work

conveyed by you (or copies made from those copies), or (b) primarily

for and in connection with specific products or compilations that

contain the covered work, unless you entered into that arrangement,

or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting

any implied license or other defenses to infringement that may

otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or

otherwise) that contradict the conditions of this License, they do not

excuse you from the conditions of this License. If you cannot convey a

covered work so as to satisfy simultaneously your obligations under this

License and any other pertinent obligations, then as a consequence you may

not convey it at all. For example, if you agree to terms that obligate you

to collect a royalty for further conveying from those to whom you convey

the Program, the only way you could satisfy both those terms and this

License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have

permission to link or combine any covered work with a work licensed

under version 3 of the GNU Affero General Public License into a single

combined work, and to convey the resulting work. The terms of this

License will continue to apply to the part which is the covered work,

but the special requirements of the GNU Affero General Public License,

section 13, concerning interaction through a network will apply to the

combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of

the GNU General Public License from time to time. Such new versions will

be similar in spirit to the present version, but may differ in detail to

address new problems or concerns.

Each version is given a distinguishing version number. If the

Program specifies that a certain numbered version of the GNU General

Public License "or any later version" applies to it, you have the

option of following the terms and conditions either of that numbered

version or of any later version published by the Free Software

Foundation. If the Program does not specify a version number of the

GNU General Public License, you may choose any version ever published

by the Free Software Foundation.

If the Program specifies that a proxy can decide which future

versions of the GNU General Public License can be used, that proxy's

public statement of acceptance of a version permanently authorizes you

to choose that version for the Program.

Later license versions may give you additional or different

permissions. However, no additional obligations are imposed on any

author or copyright holder as a result of your choosing to follow a

later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY

APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT

HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY

OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO,

THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR

PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM

IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF

ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING

WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS

THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY

GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE

USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF

DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD

PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS),

EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF

SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided

above cannot be given local legal effect according to their terms,

reviewing courts shall apply local law that most closely approximates

an absolute waiver of all civil liability in connection with the

Program, unless a warranty or assumption of liability accompanies a

copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

Anexo 5 - Unreal® Engine End User License Agreement

Please read this Agreement carefully. It is a legal document that explains your rights and obligations related to your use of the Unreal® Engine and related content. By downloading or using this software or any related content, you are agreeing to be bound by the terms of this Agreement. If you do not or cannot agree to the terms of this Agreement, please do not download or use this software or any related content. Certain words or phrases are defined to have certain meanings when used in this Agreement. Those words and phrases are defined below in Section 25.If your primary residence (or primary place of business, if you are a legal entity like a corporation or an academic institution) is in the United States of America, your agreement is with Epic Games, Inc. If it is not in the United States of America, your agreement is with Epic Games International S.à r.l., acting through its Swiss branch. If you are separately licensed by Epic under a Custom License, certain terms of this Agreement do not apply to your Custom Products. Those terms are described below in Section 26. 1.License Grant Epic grants you a non-exclusive, non-transferable, non-sublicensable (except as described in this Agreement) license to use, reproduce, display, perform, and modifythe Licensed Technology for any lawful purpose (the “License”). However, the rights that Epic grants you under the License are subject to the terms of this Agreement, and you may only make use of the License if you comply with all applicable terms.The License becomes effective on the date you accept this Agreement or download the Engine Code or any Content. The License does not grant you any title or ownership in the Licensed Technology. (A)Allowed forms of Distribution and SublicensingYou may only Distribute the Licensed Technology as follows:a.Distribution to end users-You may Distribute the Licensed Technology incorporated in object code format only as an inseparable part of a Product to end users who are subject to an end user license agreementwhich explicitly disclaims any representations, warranties, conditions, and liabilities related to the Licensed Technology. The Product may not contain any Paid Content Distributed in uncooked source format or any Engine Tools. b.Distribution to otherlicensees-You may Distribute Engine Code (including as modified by you under the License) in Source Code or object code format, or any Content, to an Engine Licensee who has rights under its license to the same Version of the Engine Code or Content thatyou are Distributing.

2Any public Distribution (i.e., intended for Engine Licensees generally) which includes Engine Tools (including as modified by you under the License) must take place either through the Marketplace (e.g., for distributing a Product’s modding tool or editor to end users) or through a fork of Epic’s GitHub UnrealEngine Network (e.g., for distributing source code). c.Distributions to employees and contractors-You also may Distribute Content (other than Paid Plug-ins) to an Engine Licensee who is your employee or your contractor who does not have rights under their license to the same Content, but only to permit that Engine Licensee to utilize that Content in good faith to develop a Product on your behalf for Distribution by you underthe License, and not for the purpose of Content pooling or any other Distribution or sublicensing of Content that is not permitted under this Agreement. Recipients of such a Distribution have a limited license to use, reproduce, display, perform, and modify that Content to develop your Product as outlined above, and for no other purpose. d.Distribution of Paid Plug-ins-You may Distribute Paid Plug-ins to each of your Paid Plug-in Users so that they may use those Paid Plug-ins on your behalf under theLicense. e.Distribution and sublicensing of Examples-You may Distribute or sublicense Examples (including as modified by you under the License) in Source Code or object code format to any third party. However, the rights in this Section 1(e) do not expand or modify your limited Distribution and sublicensing rights for Engine Code and Content (including as modified by you under the License) that are not Examples. f.Distribution of Non-C++ Programming Language Integration-You may Distribute an integration of a programming language other than C++ for the Licensed Technology, but if you do, the integration must be Distributed free of charge to all Engine Licensees, must be available in source code form (including, but not limited to, any compiler, linker, toolchain, and runtime), and must permit Distribution free of charge, on all platforms, in any Product.You are permitted to post snippets of Engine Code, up to 30 lines of code in length, online in public forums for the sole purpose of discussing thecontent of the snippet, or distribute such snippets in connection with supporting patches and plug-ins for the Licensed Technology, so long as it is not for the purpose of enabling non-Engine Licensees to use or modify any Engine Code, or to aggregate, recombine, or reconstruct any larger portion of the Engine Code.You may not sublicense the Licensed Technology in Source Code format. You may not sublicense the Licensed Technology in object code format, or any Content, except to grant end users the ability to use, or to permit your publishers and distributors to market and Distribute, a Product that you Distribute as permitted in Section 1(a) above. This paragraph does not limit your rights to Distribute and sublicense Examples.

1. [↑](#footnote-ref-1)