

**UNIVERSIDADE DE CAXIAS DO SUL
ÁREA DO CONHECIMENTO DE CIÊNCIAS EXATAS E ENGENHARIAS
CURSO DE SISTEMAS DE INFORMAÇÃO**

RONALD GAMBA DA SILVA

**ENTREGAS DE PENNY: DESENVOLVIMENTO DE UM SERIOUS GAME PARA
APRENDIZAGEM SOBRE REDES DE COMPUTADORES**

CAXIAS DO SUL - RS

2019

UNIVERSIDADE DE CAXIAS DO SUL
ÁREA DO CONHECIMENTO DE CIÊNCIAS EXATAS E ENGENHARIAS
CURSO DE SISTEMAS DE INFORMAÇÃO

RONALD GAMBA DA SILVA

**ENTREGAS DE PENNY: DESENVOLVIMENTO DE UM SERIOUS GAME PARA
APRENDIZAGEM SOBRE REDES DE COMPUTADORES**

Trabalho de Conclusão de Curso para obtenção do título de Bacharel em Sistemas de Informação pela Universidade de Caxias do Sul, na Área do Conhecimento de Ciências Exatas e Engenharias.

Orientador: Prof. Marcelo Luís Fardo.

CAXIAS DO SUL - RS

2019

RONALD GAMBA DA SILVA

**ENTREGAS DE PENNY: DESENVOLVIMENTO DE UM SERIOUS GAME PARA
APRENDIZAGEM SOBRE REDES DE COMPUTADORES**

Trabalho de Conclusão de Curso para obtenção do título de Bacharel em Sistemas de Informação pela Universidade de Caxias do Sul, na Área do Conhecimento de Ciências Exatas e Engenharias.

Aprovada em: 28/11/2019

Banca Examinadora

Prof. M.e Marcelo Luís Fardo
Universidade de Caxias do Sul – UCS

Prof. Dr. Maria de Fatima Webber do Prado Lima
Universidade de Caxias do Sul - UCS

Prof. Dr. Elisa Boff
Universidade de Caxias do Sul – UCS

AGRADECIMENTOS

Gostaria de agradecer em especial a minha família, que sempre foi a base e espelho para que eu pudesse seguir e chegar até aqui.

À Suelen Scariott, que por vezes me ajudou, aconselhou e dedicou do seu tempo para me ajudar em diversos aspectos relacionados a este trabalho.

À todos os colegas que por vezes me acompanharam nos estudos extraclasse, acabando por estar sábados inteiros na biblioteca estudando para provas e trabalhos.

À meu orientador prof. Marcelo Fardo, por ter me dado todas as direções, por me incentivar e por sempre demonstrar disposição com o esclarecimento de dúvidas e questionamentos.

A todos que direta ou indiretamente contribuíram para que este trabalho fosse possível.

Meu mais sincero obrigado!

RESUMO

O objetivo principal do trabalho é criar um *serious game*, chamado Entregas de Penny, a respeito de redes de computadores, com o foco na arquitetura TCP/IP. Esta é uma área de ensino onde os alunos apresentam dificuldades de aprendizado. Entregas de Penny vem ao encontro desta necessidade, buscando proporcionar o aprendizado deste conteúdo através de uma abordagem descontraída. *Serious game* trata-se de um tipo de jogo que procura, além de proporcionar diversão, também ensinar algum conteúdo. A metodologia utilizada é a DPE (*design, play and experience*), com o propósito de validar o jogo realizado.

ABSTRACT

The main objective of the work is to create a serious game, called Entregas de Penny, regarding computer networks, with focus on the TCP/IP architecture. This is a teaching area where students experience learning difficulties. Entregas de Penny come to meet this need, seeking to provide learning from this content through a laid-back approach. Serious game is a game that seeks, besides providing fun, also teach some content. The methodology used is the DPE (design, play and experience), with the purpose of validating the game performed.

LISTA DE FIGURAS

Figura 1 - Camadas da arquitetura TCP/IP	25
Figura 2 - Comunicação TCP/IP	25
Figura 3 - Redes de Computadores: Processos	26
Figura 4 - Redes de Computadores: Transferência de dados	27
Figura 5 - Redes de Computadores: Entrega de e-mail.....	28
Figura 6 - Redes de Computadores: DNS	29
Figura 7 - Redes de Computadores: <i>Peer-to-Peer</i>	31
Figura 8 - Redes de Computadores: Transporte	31
Figura 9 - Redes de Computadores: Processo de execução.....	34
Figura 10 - Redes de Computadores: Roteamento	35
Figura 11 - Redes de Computadores: Tunelamento de um datagrama.....	37
Figura 12 - Redes de Computadores: Interação entre nós	38
Figura 13 - Redes de Computadores: Detecção de colisões.....	40
Figura 14 - Redes de Computadores: Rede com roteador	40
Figura 15 - Redes de Computadores: Dados analógico ou digital.....	41
Figura 16 - Redes de Computadores: Onda senoidal.....	42
Figura 17 - Redes de Computadores: Modulação digital	43
Figura 18 - Redes de Computadores: Multiplexador.....	43
Figura 19 – <i>Framework</i> DPE expandida	45
Figura 20 - Aprendizagem Baseada em Problemas	46
Figura 21 - Tela de menu Inicial	51
Figura 22 - Explicação a direita do que deve ser realizado	52
Figura 23 - Elementos.....	53
Figura 24 - Especificações botões.....	53
Figura 25 - Tela do jogo.....	54
Figura 26 - Tela de comandos	54
Figura 27 - Execução com êxito	55
Figura 28 - Entregas de Penny - Protótipo tela "Menu Inicial"	60
Figura 29 - Entregas de Penny – Tela "Menu Inicial".....	61
Figura 30 - Entregas de Penny - Protótipo tela Ranking.....	62
Figura 31 - Entregas de Penny – Tela "Tutorial".....	62
Figura 32 - Entregas de Penny - Protótipo: Tela "Seleção de Nível"	63

Figura 33 - Entregas de Penny - Tela “Seleção de Nível”.....	64
Figura 34 - Entregas de Penny – Protótipo: “Tela de explicação”	65
Figura 35 - Entregas de Penny – Tela de Explicação: Camada de Aplicação	66
Figura 36 - Entregas de Penny - Tela de Explicação: Camada de Rede.....	66
Figura 37 - Entregas de Penny - Tela de Explicação: Camada de Rede.....	67
Figura 38 - Entregas de Penny - Tela de Explicação: Camada de Enlace	67
Figura 39 - Entregas de Penny - Tela de Explicação: Camada Física.....	68
Figura 40 - Entregas de Penny - Tela do jogo	68
Figura 41 - Entregas de Penny - <i>Layout</i> Base	69
Figura 42 - Entregas de Penny - Protótipo: Menu <i>pause</i>	70
Figura 43 - Entregas de Penny - Menu <i>pause</i>	70
Figura 44 - Entregas de Penny – Protótipo: Nível camada de aplicação	71
Figura 45 - Entregas de Penny - Explicação inicial.....	72
Figura 46 – Entregas de Penny - Camada de Aplicação	73
Figura 47 - Entregas de Penny - Bloqueio final nível um	74
Figura 48 - Entregas de Penny - Instruções de entrar na van	74
Figura 49 - Entregas de Penny - Final camada de aplicação	75
Figura 50 - Entregas de Penny - Explicação inicial camada de transporte	76
Figura 51 - Entregas de Penny - Ordem dos pacotes.....	76
Figura 52 - Entregas de Penny - Placa informativa	77
Figura 53 - Entregas de Penny - Final objetivo nível transporte	78
Figura 54 - Entregas de Penny - Final camada de Transporte	78
Figura 55 - Entregas de Penny - Casas nível de rede	79
Figura 56 - Entregas de Penny - Caixa de diálogo	80
Figura 57 - Entregas de Penny - Mensagem roteador	80
Figura 58 - Entregas de Penny - Final camada de rede	81
Figura 59 - Entregas de Penny - Informação camada de enlace	82
Figura 60 - Entregas de Penny - Pacote a ser entregue.....	83
Figura 61 - Entregas de Penny - Final nível enlace	84
Figura 62 - Entregas de Penny - Final camada de enlace.....	84
Figura 63 - Entregas de Penny - Informações camada física	85
Figura 64 - Entregas de Penny - Plataforma digital	86
Figura 65 - Entregas de Penny - Destino onda digital	86
Figura 66 - Entregas de Penny - Formato plataforma analógica	87

Figura 67 - Entregas de Penny - Final objetivo camada física.....	88
Figura 68 - Entregas de Penny - Final camada Física.....	88

LISTA DE QUADROS

Quadro 1 - Camadas do modelo OSI.....	23
---------------------------------------	----

LISTA DE TABELAS

Tabela 1 - Elementos do jogo	19
Tabela 2 - Exemplos de portas mais utilizadas.....	33

LISTA DE SIGLAS

Sigla	Significado
AGDC	<i>Australian Game Developers' Conference</i>
API	<i>Application Programing Interface</i>
ARPA	<i>Advanced Research Projects Agency</i>
C#	<i>C Sharp</i>
DNS	<i>Domain Name System</i>
DPE	<i>Design Play Experience</i>
FTP	<i>File Transfer Protocol</i>
GBL	<i>Game Based Learning</i>
HTTP	<i>HyperText Transfer Protocol</i>
IMAP	<i>Internet Message Access Protocol</i>
IP	<i>Internet Protocol</i>
IPv4	<i>Internet Protocol version 4</i>
IPv6	<i>Internet Protocol version 6</i>
MIT	<i>Tecnologia de Massachusetts</i>
NCP	<i>Network Control Protocol</i>
OSI	<i>Open Systems Interconnection</i>
POP3	<i>Post Office Protocol</i>
PPP	<i>Point-to-Point Protocol</i>
SCPT	<i>Stream Control Transmission Protocol</i>
SMTP	<i>Simple Mail Transfer Protocol</i>
TCP	<i>Transmission Control Protocol</i>
UDP	<i>User Datagram Protocol</i>
2D	<i>2 Dimensions</i>

SUMÁRIO

1	INTRODUÇÃO	16
1.1	OBJETIVOS.....	17
2	REFERENCIAL TEÓRICO	18
2.1	JOGOS E APRENDIZAGEM	18
2.2	<i>SERIOUS GAME</i>	21
2.3	ARQUITETURA TCP/IP.....	22
2.4	CAMADAS DA ARQUITETURA TCP/IP	24
2.4.1	Camada de aplicação	25
2.4.2	Camada de transporte	31
2.4.3	Camada de rede	34
2.4.4	Camada de enlace de dados.....	37
2.4.5	Camada física.....	41
3	METODOLOGIA.....	45
3.1	APRENDIZAGEM	46
3.2	NARRATIVA	47
3.3	<i>GAMEPLAY</i>	47
3.4	EXPERIÊNCIA DO USUÁRIO	48
3.5	TECNOLOGIA	48
4	ESTUDO DE CASO.....	50
4.1	NETSIM	50
4.1.1	Narrativa Netsim	52
4.1.2	Gameplay Netsim.....	53
4.1.3	Experiência do jogador	56
5	DESENVOLVIMENTO	57
5.1	ENTREGAS DE PENNY - GÊNERO	57

5.2	ENTREGAS DE PENNY – NARRATIVA	57
5.3	ENTREGAS DE PENNY – MECÂNICA	58
5.4	ENTREGAS DE PENNY – APRENDIZAGEM	59
5.5	ENTREGAS DE PENNY – JOGABILIDADE	60
5.5.1	Nível - Camada de Aplicação	71
5.5.2	Nível - Camada de Transporte	75
5.5.3	Nível - Camada de Rede	79
5.5.4	Nível - Camada de Enlace	82
5.5.5	Nível - Camada Física	85
5.6	ENTREGAS DE PENNY – EXPERIÊNCIA	89
5.7	ENTREGAS DE PENNY – TECNOLOGIA	89
6	CONSIDERAÇÕES FINAIS	91
6.1	TRABALHOS FUTUROS	93
	REFERÊNCIAS	94

1 INTRODUÇÃO

Em um mundo cada vez mais imerso na tecnologia, onde a evolução e adaptação dos produtos de *software* estão deixando de ser apenas ferramentas para a diversão, e tornando um instrumento de forte ajuda para aprendizagem em diversas áreas, os jogos digitais vem se destacando neste papel.

Com conceito bem definido, um jogo estabelece pontos básicos e chaves para que o jogador evolua no mesmo. Tendo um universo criado por regras, são baseadas nestas regras, e na evolução do jogador sobre elas, que haverá os resultados bem estabelecidos e podem usar dos resultados como parâmetro se o objetivo do jogo foi ou não alcançado.

Em 2003, o professor Stapleton apresentou a sua obra *Why Videogames are Cool and School Sucks!* na *Australian Game Developers' Conference* (AGDC), trazendo como tema base a utilização de jogos sérios como principal conceito de aprendizado em suas aulas, assim alegando que a tecnologia estava cada vez mais tomando conta das aulas e envolvendo as pessoas a terem uma aprendizagem mais rápida e com uma melhor performance do que somente através dos métodos tradicionais.

Segundo Stapleton (2003), a experiência de obter o conhecimento através de um jogo fazia com que o jogador ficasse mais envolvido com o aprender, assim rapidamente o conceito e os objetivos de tal tarefa eram conquistados do que estudando da maneira tradicional, desta forma o jogador se desenvolveria melhor e em menos tempo.

"Pense no que é preciso para aprender um jogo comparado a o que tem que ser feito na escola. Para jogar bem um jogo requer os mesmos tipos de aprendizado, estudo, compreensão e prática, conforme exigido de qualquer atividade educativa"¹ (Norman 1993, p.32, tradução nossa).

Este trabalho realiza a criação de um jogo sério, com o intuito de promover aprendizagem sobre o assunto de redes de computadores.

¹ **Do original:** "Think of what it takes to learn a game compared to what has to be done in school. To play a game well requires the same kinds of learning, study, understanding, and practice as are required of any educational activity. (Norman 1993, p.32)."

O motivo principal levado em conta para a criação do *serious game*, foi de que jogos educacionais vem cada vez mais se destacando como ferramenta importante no aprendizado de determinados conteúdos.

Contudo, outro motivo chave que levou ao objetivo da criação do *serious game* para abordar o assunto TCP/IP, foi que segundo a professora Maria de Fatima Webber do Prado Lima, da Universidade de Caxias do Sul, a arquitetura TCP/IP é um assunto ainda tratado como complexo ao ser abordado nas aulas tradicionais, sendo assim a necessidade de criar uma interatividade maior ao introduzir o conteúdo com os alunos nas aulas ministradas.

Estamos fortemente influenciados pela abundância de tecnologia que exige a adaptação de novos conhecimentos de maneiras diferentes, com isso foi verificado que o assunto de redes de computadores, mais em específico a arquitetura TCP/IP é um bom tema para ser abordado dentro de um *serious game*.

Como o objetivo é ensinar conceitos básicos da arquitetura TCP/IP de uma maneira mais eficiente, a criação de um *serious game* pode conceder aos jogadores o conhecimento de maneira mais rápida e eficaz, através de conceitos aplicados pela metodologia DPE (*Design Play and Experience*) que será abordada no capítulo 4 deste trabalho.

Foi realizado também um estudo de caso, do jogo “Netsim”, que é uma proposta de *serious game* criado com o intuito de gerar o conhecimento sobre alguns assuntos de redes de computadores.

1.1 OBJETIVOS

O objetivo deste trabalho é desenvolver um *serious game* que possa auxiliar os jogadores no estudo sobre redes de computadores, mais especificamente sobre o assunto da arquitetura de rede TCP/IP.

Além do objetivo geral deste trabalho, serão realizados os seguintes objetivos específicos

- Abordagem sobre o que é um *Serious Game*
- Pesquisa da arquitetura TCP/IP de redes de computadores
- Explicação da metodologia DPE (*design, play and experience*)
- Estudo de caso sobre o jogo Netsim

2 REFERENCIAL TEÓRICO

Primeiramente, neste capítulo, será abordada a contextualização do que é um jogo e como ele pode aprimorar o conhecimento de seus clientes "jogadores". Então, será introduzido o assunto de *serious game*, explicando o que é este termo e como ele se diferencia dos jogos casuais. Para concluir, será explicada a conceitualização da arquitetura TCP/IP e de seu funcionamento.

2.1 JOGOS E APRENDIZAGEM

Baseado nos conceitos de game design, Salen e Zimmerman contemplam em sua obra "*Rules of Play*" o conceito de que jogo é, "um sistema em que os jogadores engajam em um conflito artificial, definido por regras, cujo resultado é quantificável"² (SALEN;ZIMMERMAN, 2004, p. 80).

Portanto, pode-se definir um jogo como um sistema em que jogadores (tomadores de decisões) entram na atividade de solucionar um desafio artificial, definido por regras, com o objetivo de superação, assim existindo a meta do jogador, no qual o resultado é quantificável.

Para Michael e Chen (2005) as ações e decisões dos jogadores possuem um limite, o qual é abrangido por um conjunto de regras dentro de um universo. Essas regras são controladas pela equipe desenvolvedora do jogo. O universo do jogo contextualiza as decisões das ações tomadas pelo jogador, que formam a narrativa do jogo. Sendo assim as regras definem limitações do que pode ou não ser realizado.

Existe uma combinação de regras e objetivos a serem alcançados em um jogo. Cabe ao jogador tomar a melhor decisão para alcançar os objetivos previamente estabelecidos dentro do jogo.

Para isto, há elementos básicos na sua construção conforme tabela 1:

² **Do original:** "A game is a system in which players engage in an artificial conflict, defined by rules, that results in a quantifiable outcome". (Salen;Zimmerman, 2004, p. 80).

Tabela 1 - Elementos do jogo

Elemento	Detalhamento
Interatividade	Atividade do jogador no jogo.
<i>Meaningful play</i>	Resultado de cada ação, tornando a significativa.
Imersão	Sensação de alheamento da realidade.
Desafio apropriado	Ponto central em um jogo não é o final, mas sim o desafio em si.
Experiência emocional	Simulando realidades emocionais, a superação de desafios gera emoções reais de alegria e prazer.
Consistência interna	Depois de aprendidas as mecânicas do jogo, o jogador começa a experimentar novas soluções usando estas mecânicas como ferramentas.
Segurança com ilusão de risco	O jogador se sentirá livre para experimentar e aprender com os erros.

Fonte: Do autor.

Jogos, por si só, possuem diferentes classificações, gêneros, mecânicas, plataformas e objetivos. Isto é o que constitui o seu universo, que engloba o enredo do jogo, a forma que o mesmo será jogado e sua interface interativa:

- Enredo: Ambientação do jogo, neste momento são definidos o tema e a sua narrativa, seus objetivos e acontecimentos.
- Forma do jogo: Chamado também de "motor", neste ponto é realizado os limites e as regras do jogo, definindo reações para cada decisão tomada do jogador.

- Interface interativa: Permitindo a interação do jogador com o motor do jogo, deixando o jogador realizar uma ação e retornando ao mesmo a sua resposta audiovisual.

Jogos digitais ocupam uma grande camada de adeptos ao redor do mundo. Variados tipos de pessoas tornam-se jogadores, independente de idade, gênero, raça, ideologia, etc. Com isso o público alvo acaba sendo hipnotizado pelos jogos, permanecendo longos períodos engajados nos desafios propostos.

Para que isso aconteça, os jogos acabam criando viciantes sistemas de recompensas, tornando os jogadores progressivamente mais envolvidos no universo do jogo.

Muitos jogos trabalham com o conceito de rivalidade entre jogadores, onde o ego é utilizado, pois o jogador quer mostrar que é "melhor" do que os demais jogadores. Esta tática é baseada no contexto de recompensa aplicado em determinados momentos. Desta forma os jogadores querem se destacar mostrando que possuem algo que é cobiçado entre os jogadores daquele universo ou por simplesmente exibir que suas habilidades são melhores do que as dos demais.

Michael e Chen (2005) afirmam que, em outras palavras, se o jogador não achar que o jogo é divertido, é improvável que ele o escolha para jogar novamente.

Desta forma, é necessário o investimento de inúmeras horas de jogo para alcançar os objetivos por eles propostos, que poderia ser aplicado em demais tarefas, como os estudos, práticas de esporte, entre outras diversas atividades.

Em momentos nos quais estão jogando, os jogadores são forçados a realizar tarefas complicadas e isso está relacionado com o aprendizado. Ao executar essas tarefas, estão aprimorando seus conhecimentos em determinada ação.

Game-based learning da sigla em inglês GBL, e em português conhecido como "Aprendizagem Baseada em Jogos", é a abordagem que utiliza jogos para melhorar a experiência de aprendizado do jogador.

Os métodos usados para o ensino tradicional são através de conteúdos teóricos. Os estudantes observavam apresentações do raciocínio pronto de um professor e absorvem o máximo do mesmo. Com a evolução, pode se observar a transformação do ensino, tornando o estudante um ator ativo, trazendo o desenvolvimento do conhecimento através de habilidades.

Prensky (2001), ressalta a diferença entre as gerações, sendo que a nova geração é chamada por ele de "nativos digitais". Essa geração é composta pelas

pessoas que nasceram e se desenvolveram em meio a tecnologia.

Com isso, há uma aprendizagem mais ativa entre os "nativos digitais", pois são estimulados a participar cada vez mais de jogos digitais, como ferramenta para obter o aprendizado proposto nos mesmos.

2.2 *SERIOUS GAME*

Jogos sérios, ou do inglês *Serious Games*, são *softwares* criados com o conceito de *design* de jogos. Eles possibilitam que o usuário (jogador) tenha a experiência de jogar algum jogo com o resultado de obter o conhecimento do assunto proposto por este jogo.

Thompson (2008) diz que esses jogos são capazes de levar o usuário a uma imersão, estimulando conhecimento, atenção, autocontrole, tomada de decisão, eficiência e motivação dos sentimentos de competência dos mesmos.

Portanto eles são jogos utilizados com o propósito de ensino e aprendizagem ou treinamento, não apenas de entretenimento.

Clark menciona que "o termo 'sério' também é usado no sentido de estudo, relativo a assunto de grande interesse e importância"³ (1987, pg. 10, tradução nossa), desta forma afirmando que os *serious games* contemplam questões de ensino ao jogador.

Com isto, os *serious games* permitem a criação de um ambiente controlado, com metas e regras previamente estabelecidas, que contemplem a experiência do jogador com o jogo e agregando competências e habilidades aos usuários.

Segundo Michael e Chen, (2006) jogos convencionais, que não possuem o objetivo de passar algum tipo conhecimento, focam somente no entretenimento dos jogadores, sendo que para a construção de um jogo sério é necessário mesclar o entretenimento com o aprendizado e experiência para o usuário, através de métodos incluídos nos processos dos jogos.

Trybus (2010) afirma que um jogo sério deve trazer aprendizagem ativa, e a práticas de competências. Pode-se dizer que uma forma de auxiliar na motivação do jogador é ter um objetivo claro e pré-requisitos bem definidos.

³ **Do original:** "The term "serious" is also used in the sense of study, relating to matter of great interest and importance". (Clack, 1987, pg. 10).

Os *serious games* vem cada vez mais se popularizando no cenário mundial dos jogos, como afirma Dias (2015). Pesquisadores e professores estão buscando esta temática para instigar cada vez mais os seus seguidores e alunos a adquirir conhecimento. Segundo Fleury (2014), o mecanismo de *serious games* é uma potencial ferramenta de motivação para as pessoas buscarem o aprendizado em determinadas áreas.

O uso de Jogos Sérios é um exemplo de recurso didático que tem se destacado nos últimos tempos e tem unido ensino e diversão (Savi e Ulbricht 2008).

Cada vez mais os jogos sérios vêm se tornando uma ferramenta crucial nas escolas, academias, faculdades, universidades e etc, pois ele está vinculado com o aprendizado e a forma simples e envolvente de passar o conhecimento de um assunto em específico de maneira coerente e eficaz.

Diferente dos jogos tradicionais, onde os mesmos abordam primariamente os princípios do entretenimento, os jogos sérios geralmente são encomendados para o treinamento de funcionários e engajamento de equipes. Desta forma, podem ser utilizados para promover a prática em diferentes grupos, sendo contextualizados dentro das necessidades de cada empresa para a melhora de resultados dos usuários (colaboradores).

Segundo Michael e Chen (2005), dizem que a definição mais simples a se dar para um *serious game* é de um jogo que não tem entretenimento, prazer ou diversão como propósito primário. Isso não significa que os jogos sérios não podem ser interessantes, prazerosos e/ou divertidos, porém o seu conceito principal é instigar o jogador a aprender sobre o assunto proposto. Desta forma, percebe-se o grande potencial que os *serious games* possuem no aspecto de ensino.

Os jogos sérios possuem um tema central para o qual se deseja desenvolver o aprendizado. Neste trabalho, o assunto que será abordado é a arquitetura de rede TCP/IP, que se trata da comunicação entre dois *hosts* através de um meio físico.

2.3 ARQUITETURA TCP/IP

Baseado no conceito principal de comunicação entre computadores, TCP/IP é uma arquitetura usada para realizar o transporte da informação de um *hardware* a outro.

O modelo de referência para a implementação das arquiteturas é a OSI (*Open Systems Interconnection*). Sugerindo a utilização de sete camadas de abstração dos dados. A quadro 1 com mostra o nível (número da camada), e a função de cada camada no modelo OSI.

Quadro 1 - Camadas do modelo OSI

Camada	Função	
7	Aplicação	Camada utilizada para transporte efetivo dos dados de interesse do usuário.
6	Apresentação	Define a sintaxe e a semântica dos dados transmitidos a fim de tornar possível a comunicação entre computadores que utilizam diferentes representações internas.
5	Sessão	Permite que usuários em diferentes computadores estabeleçam sessões de comunicação, oferecendo serviços de controle do diálogo entre eles.
4	Transporte	Fragmenta os pacotes de dados para transmissão e oferece um canal fim-a-fim confiável (livre de erros).
3	Rede	Determina a forma como os pacotes de dados devem ser roteados entre a origem e o destino e gerencia a demanda de transmissão de dados a fim de evitar gargalos.
2	Enlace	Transfere unidades de dados do serviço de enlace entre diversos circuitos da camada física, além de detectar e corrigir erros dessa camada.
1	Física	Meio mecânico ou elétrico para a transmissão de <i>bits</i> entre entidades de enlace.

Fonte: Tanenbaum (2011).

Para que seja realizada essa troca de informação entre *hardwares*, houve a necessidade da criação de formatos pré-estabelecidos, com uma estrutura bem definida de transmissão dos dados. Assim surgiram os protocolos de comunicação.

Os protocolos de comunicação podem ser definidos como uma conversa entre diversos computadores. Eles necessitam de um *software* para realizar esta tarefa, que por sua vez é implementado em cada *hardware* de rede. Segundo Tanenbaum (2011), este *software* está estruturado em camadas, sendo cada uma responsável por executar determinadas funções, sendo assim uma camada presta serviço as camadas superiores.

Neste caso, um exemplo que pode se usar para explicar melhor a funcionalidade do protocolo é o de transmitir uma informação (conhecido também como "pacote" ou "dado") de um computador para outro computador, ambos devidamente configurados de maneira correta.

Ao realizar a transferência de um pacote entre os computadores, existe a necessidade de conhecer o receptor dessa informação, para isso são utilizados protocolos de comunicação.

Desta forma o protocolo é uma norma, um padrão de comunicação que

acrescenta informações de controle que a camada par do destino analisará.

Desenvolvido pela ARPA (Rede da Agência para Projetos de Pesquisa Avançada⁴), o TCP/IP foi projetado para o Departamento de Defesa dos Estados Unidos da América (EUA), segundo Forouzan e Mosharraf (2006, p.52):

Interessada em encontrar uma maneira de conectar computadores para que os pesquisadores que patrocinavam pudessem compartilhar seus conhecimentos, reduzindo, assim, os custos e eliminando a duplicação de esforços.

Em 1967, em uma reunião da ACM (*Association for Computing Machinery*), a ARPA apresentou a idéia da ARPANET, uma pequena rede de computadores conectados. A idéia era que cada computador *host* (não necessariamente do mesmo fabricante) fosse ligado a um computador especializado.

Em 1969, a ARPANET estava consolidando o seu projeto interligando as Universidades da Califórnia em Los Angeles, a Universidade da Califórnia em Santa Bárbara, a Universidade de Utah e a Universidade de Stanford, formando a primeira rede de computadores, intitulado de NCP (*Network Control Protocol*).

Após a implementação do NCP, Vincent Cerf e Bob Khan, que faziam parte do grupo que criou a ARPANET, decidiram que precisavam distribuir mais a solução de ponto a ponto de entrega de informação entre demais computadores. Assim foi criado um protocolo com controle de transmissão, chamado de TCP (*Transmission Control Protocol*), incluindo assim conceitos de encapsulamento, função de *gateway* e datagrama na comunicação entre os *hardwares*.

Segundo Forouzan (2006), em meados de 1977, houve a decisão de dividir o protocolo TCP em dois, assim criando o próprio TCP (*Transmission Control Protocol*) que ficaria responsável pelas funções de nível mais alto, e o IP (*Internet Protocol*) que trataria do roteamento de datagrama.

Com isso foi formado o TPC/IP, que é a mescla de dois protocolos, onde a junção destes permitem o tráfego em uma conexão de maneira sequencial e sem a duplicação da informação entre *hardwares*.

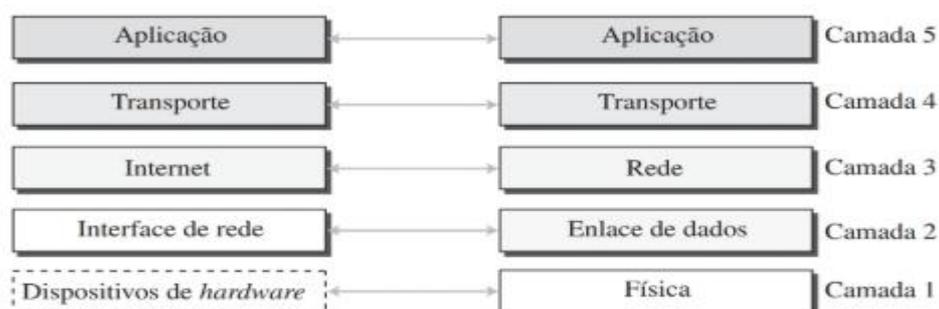
2.4 CAMADAS DA ARQUITETURA TCP/IP

A arquitetura TCP/IP é formado por 4 níveis conceituais, hierarquicamente, cada nível promove sua funcionalidade específica.

⁴ *Advanced Research Projects Agency*

Originalmente construído com quatro camadas de *software*, Behmouz e Fegan (2000) mostram, no entanto, que o TPC/IP é visto com cinco camadas em seu modelo atual, conforme a figura 2. Onde a camada física é considerada uma camada nova.

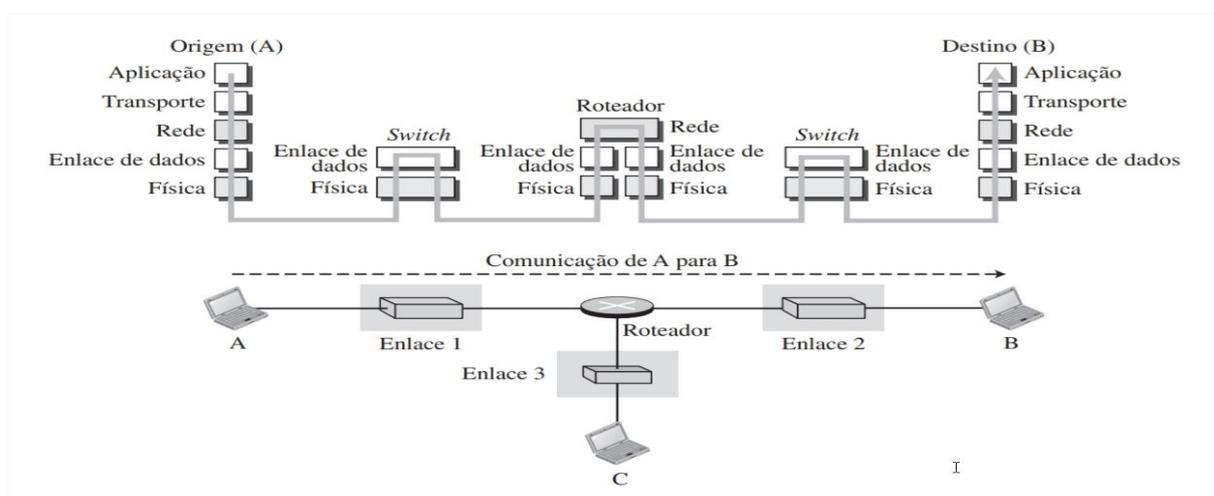
Figura 1 - Camadas da arquitetura TCP/IP



Fonte: Behmouz e Fegan (2000. p.12).

A figura 2 ilustra como é realizada uma comunicação com a arquitetura TCP/IP.

Figura 2 - Comunicação TCP/IP



Fonte: Forouzan e Mosharraf (2013, p. 43)

2.4.1 Camada de aplicação

A camada de aplicação tem por finalidade interligar dois *hosts*, através de uma conexão lógica, aonde significa que duas camadas de aplicação consideram a existência de uma conexão imaginária entre elas, podendo trocar dados entre ambas através de serviços.

Para a execução dessa comunicação entre os serviços, existem dois paradigmas que executam essa função, o paradigma cliente-servidor e o paradigma *peer-to-peer*.

No paradigma cliente-servidor, há um provedor de serviços chamado de processo-servidor, ele é executado continuamente esperando o retorno de outro aplicativo, sendo este chamado de processo-cliente. O processo-servidor fica todo tempo executando sua função, enquanto o processo-cliente é iniciado quando o cliente necessita do serviço a ser executado.

Sendo assim o processo-servidor está executando um ou mais serviços que compartilham recursos com o processo-cliente.

Forouzan e Mosharraf (2013, p. 68) destacam que:

processos: um cliente e um servidor. Um cliente é um programa em execução que inicia a comunicação enviando um pedido (ou solicitação); um servidor é outro programa que aguarda pedidos de clientes. O servidor trata o pedido recebido, prepara um resultado e envia o resultado de volta ao cliente.

Nessa comunicação entre cliente e servidor, é usado o conjunto de instruções chamado de Interface de Programação de Aplicação (API), a API realiza a comunicação entre os processos através de um sistema operacional, possibilitando o envio e o recebimento de mensagens no canal lógico criado.

Um exemplo é a API *Socket* que surgiu no início de 1980 na Universidade de Berkeley, que realiza um conjunto de instruções comunicando a camada de aplicação com o sistema operacional do *hardware*.

Desta forma existem dois *sockets* que se comunicam, codificando os dados da camada de aplicação, para fazer com que os dados das mensagens trocadas entre processo cliente e servidor possam ser decodificadas e traduzidas para o sistema operacional poder executar as ações dos processos conforme figura 3.

Figura 3 - Redes de Computadores: Processos



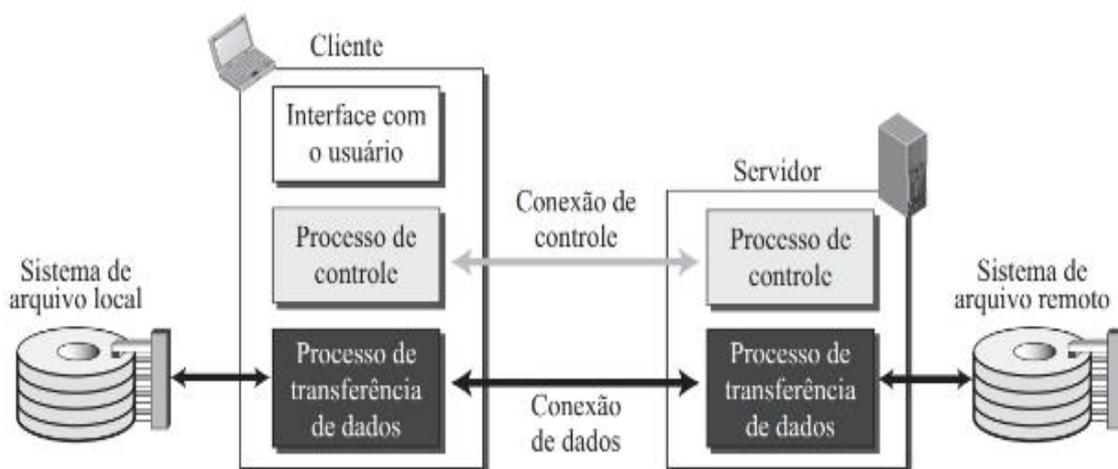
Alguns protocolos são bem conhecidos por executarem suas funções com excelência e estarem bem consolidados em seu uso.

Para a *internet*, o protocolo que define como os navegadores requisitam as páginas de servidores *web* é o HTTP (*HyperText Transfer Protocol*). Isso acontece com o cliente requisitando o acesso a porta 80 do servidor via uma conexão TCP, com o acesso permitido o cliente faz a requisição do protocolo HTTP, que por sua vez retorna com o requerimento solicitado.

Quanto a transferência de arquivos entre dois computadores, o protocolo é o FTP, conhecido também como "*File Transfer Protocol*". Ele utiliza duas conexões paralelas TCP para transferir os arquivos. Sendo uma para controle da conexão e a outra para a transferência dos dados.

Na conexão de controle são passadas as identificações do usuário pedindo a permissão de fazer a busca ou envio dos arquivos. Caso haja algum problema ou erro é retornado o erro e a conexão é fechada; caso contrário é estabelecida a verificação e os arquivos ficam disponíveis para serem baixados ou enviados, conforme figura 4.

Figura 4 - Redes de Computadores: Transferência de dados

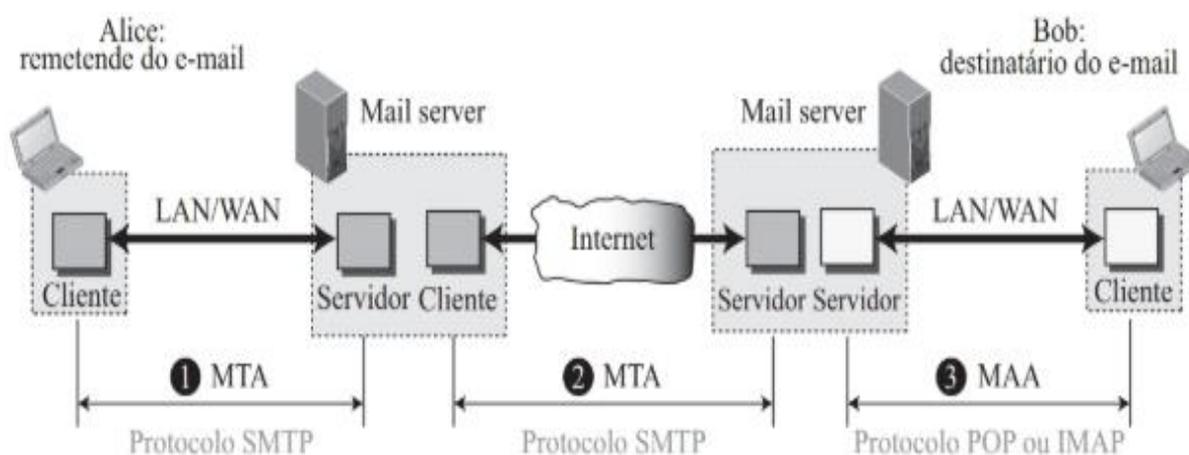


Fonte: Forouzan e Mosharraf (2013 p.89)

O protocolo que define a transferência de e-mails na rede é o SMTP (*Simple Mail Transfer Protocol*). Ele realiza o envio das mensagens origem e destino do remetente para seu servidor de e-mail, e do servidor de e-mail do remetente para o servidor de e-mail do destinatário.

Desta forma o SMTP recebe uma lista com os destinatários da mensagem, e encaminha a mesma a todos os especificados nela. Como o SMTP é um protocolo somente de envio, ele necessita de um cliente de e-mail conhecido como “agentes de acessos de mensagens”. Este realiza a entrega do e-mail ao computador final. Segue figura 5 com um exemplo da remetente Aline encaminhando um e-mail para o Bob, que é o destinatário.

Figura 5 - Redes de Computadores: Entrega de e-mail



Fonte: Forouzan e Mosharraf (2013 p.67)

O destinatário realiza o acesso a sua caixa postal através de dois protocolos o POP3 (*Post Office Protocol*) ou IMAP (*Internet Message Access Protocol*).

No protocolo POP3, o *software* cliente é instalado no computador do destinatário, enquanto o *software* servidor é instalado no servidor de e-mail. O cliente realiza a consulta no servidor de e-mail e se existe algum destinado ao mesmo, ele pode fazer o *download*, listar, recuperar ou deletar as mensagens uma a uma. Com isso o cliente pode decidir excluir ou manter a mensagem na caixa de correios do seu servidor.

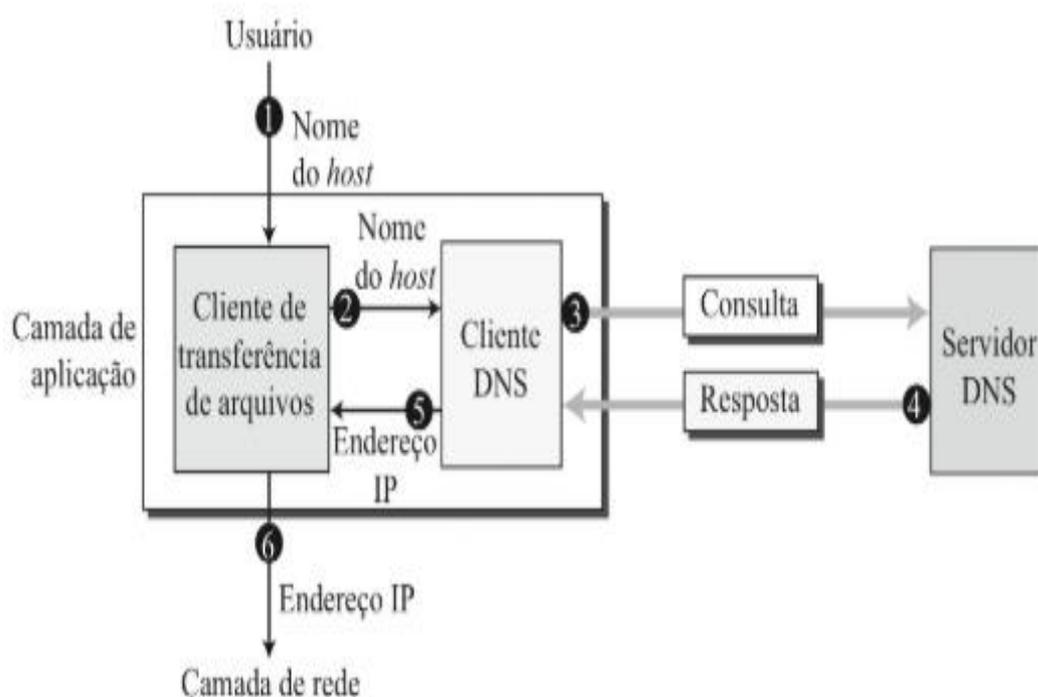
Com o protocolo IMAP, o cliente tem mais funcionalidades na leitura do e-mail no servidor. Diferentemente do POP3 que tem que baixar o e-mail completo, no IMAP ele lê somente o cabeçalho do e-mail antes de fazer o *download* do mesmo ou ele pode procurar por palavras chaves e encontrar o e-mail no servidor. O cliente pode renomear as caixas de correios no seu servidor, diferentemente do POP3 que não permite essa funcionalidade e com o IMAP, o cliente pode criar uma hierarquia de pastas para organização desses e-mails no servidor.

O DNS (*Domain Name System*), é o protocolo que permite traduzir nome de domínio em um endereço IP. Assim facilitando o uso por não necessariamente termos em memória o IP correspondente a busca, mas sim em nomes mais específicos do assunto.

Como exemplo será utilizado o *site* da empresa Autotravi, que é “www.autotravi.com” e, se não houvesse o DNS dando o nome da empresa ao *site*, para conectar no mesmo seria necessário acessar ele pelo seu IP, que é “200.198.179.55”. Seria mais complicado precisar decorar o IP que se pretende acessar na *internet*, tornando mais viável o conhecimento das palavras chaves de acesso, do que propriamente o IP do *site*.

Para resolver estes nomes, o DNS usa de um banco de dados distribuído, armazenando em sistemas cooperativos, chamados de "resolvedores de nomes", que realizam a verificação de nomes de *sites* para seu respectivo IP. Conforme figura 6, pode-se verificar como o DNS trabalha para obter os endereços solicitados.

Figura 6 - Redes de Computadores: DNS



Fonte: Forouzan e Mosharraf (2013 p.82)

O paradigma *peer-to-peer*, conhecido também como "par a par", é um paradigma onde não existe a necessidade de um processo-servidor, pois não há a

execução contínua do servidor de processo esperando por um cliente, e sim o processo servidor sendo cliente também. Então, neste caso existem dois ou mais processos clientes que podem ou não serem servidores ao mesmo tempo, trocando mensagens.

A primeira rede *peer-to-peer* segundo Forouzan e Mosharraf (2013) foi a WWIVnet, sendo um componente de rede do "Guerra Mundial Quatro", um sistema de quadro de avisos criado em 1987 por Wayne Sino.

Alguns anos depois, em 1999, Ian Clarke projetou a Freenet, um sistema descentralizado, distribuído e resistente que armazenava dados com objetivo de fornecer liberdade de expressão para os usuários através do anonimato do conteúdo.

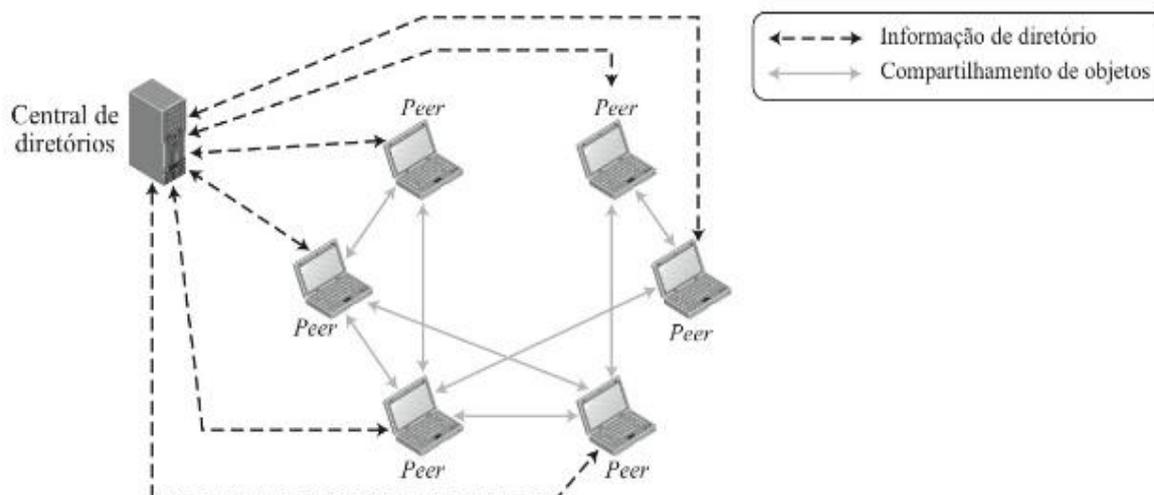
Nasce a rede Napster em 1999. Também criada por Shawn Fanning, era um serviço que permitia o compartilhamento de arquivos de música *online* entre os usuários. Porém o mesmo foi fechado logo em seguida devido a uma ação judicial de violação dos direitos autorais contidos no *site*.

Logo em seguida surge a rede BitTorrent criada por Bram Cohen, uma das mais usadas redes par a par hoje em dia segundo Forouzan e Mosharraf (2013), sendo a mesma criada para o compartilhamento de arquivos considerados grandes.

A rede *torrent* foi criada para que mais de um computador possa encaminhar o arquivo solicitado por um par ao mesmo tempo, assim os arquivos podem ser enviados em partes, gerando um aglomerado de computadores que enviam os arquivos solicitados para par que necessita do arquivo, até o mesmo ter o arquivo por completo.

A rede *peer-to-peer* pode ser centralizada ou descentralizada, dependendo da aplicação imposta na construção da mesma. Uma rede centralizada, é aquela onde existe um servidor que controla os arquivos e acessos dos pares, desta forma realizando o papel de controlador da rede, sendo chamado de centralizador. Já as redes descentralizadas, são as que não possuem um centralizador que executa as tarefas de verificação de dados e acessos, e sim deixando com que os pares façam essa função por si próprios.

Segue a figura 7 com um exemplo de rede *peer-to-peer*.

Figura 7 - Redes de Computadores: *Peer-to-Peer*

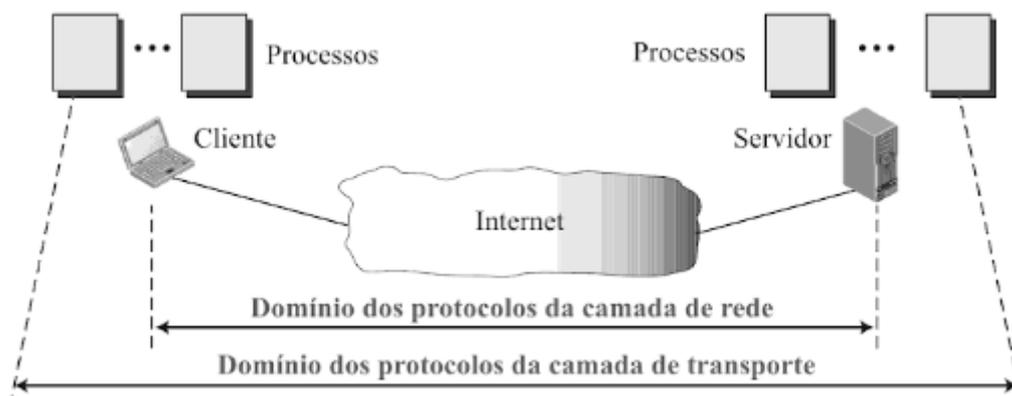
Fonte: Forouzan e Mosharraf (2013, p. 124)

2.4.2 Camada de transporte

É a camada que fornece a comunicação entre processos, ou seja, entre duas camadas de aplicação de diferentes *hosts*.

A camada de transporte tem como função principal criar um canal lógico entre os processos das aplicações, fazendo com que haja uma comunicação fim-a-fim entre elas. Conforme ilustra figura 8.

Figura 8 - Redes de Computadores: Transporte



Fonte: Forouzan e Mosharraf (2013 p.141)

Para que haja essa comunicação entre cliente servidor com o uso de uma API,

é necessário usar um serviço de transporte de dados. Onde os mais comuns são o UDP, TCP e SCTP.

O protocolo UDP, fornece um datagrama que não exige conexão lógica entre os dois sistemas que trocam as mensagens. Esse datagrama é uma entidade encapsulada enviado por uma mensagem, contendo um pacote de dados. Entretanto este pacote enviado não consegue criar uma ligação entre os pacotes já encaminhados, tornando o mesmo um protocolo não muito confiável.

Diferentemente o protocolo TCP, que fornece um serviço de conexão confiável para a transferência dos pacotes. Ele exige que dois sistemas criem uma conexão lógica entre eles através da troca de alguns pacotes de estabelecimento de comunicação (conexão).

Conhecido como “aperto de mãos” estabelece alguns parâmetros entre os sistemas finais. Ele informa o tamanho dos pacotes de dados a serem trocados, enviando um cabeçalho com os buffers que armazena a quantidade de dados da mensagem completa, e outros modelos de especificação do pacote que está sendo enviado. Essa verificação é realizada baseado no tamanho geral do pacote de dados enviado.

Caso haja dados perdidos no meio do caminho, o receptor pode solicitar o reenvio do pacote, pois ele sabe exatamente a quantidade de dados que está sendo enviada ao realizar a comunicação inicial através do cabeçalho de verificação. Isso faz com que ele possa controlar o fluxo e o congestionamento dos dados também, tornando o protocolo TCP confiável na troca de pacotes de dados.

Já o protocolo SCPT, fornece um serviço combinando os protocolos TCP e UDP, usando de múltiplos fluxos de múltiplas conexões de transporte de pacotes.

Quem emite a mensagem, fragmenta e encapsula ela em um PDUs (*Protocol Data Unit*), ou unidade de dados de protocolo, segmentando a mesma e especificando um cabeçalho de validação. Essa segmentação é então colocada no datagrama do pacote da rede, para que esse dado possa realizar as validações de entrega da mensagem ao destinatário correto.

Forouzan e Mosharraf (2013, p. 171) explicam essa função com o seguinte detalhe:

Um protocolo de camada de rede pode entregar a mensagem apenas para o computador de destino, mas a entrega ainda não está completa. A mensagem ainda precisa ser entregue ao processo correto. Esse é o ponto no qual um protocolo da camada de transporte assume a entrega. Um protocolo de

camada de transporte é responsável pela entrega da mensagem para o processo apropriado.

Para que possa existir a comunicação entre os processos, é necessário criar um canal único entre eles, desta forma são usados identificadores de grupo, também chamado de "porta de acesso". No protocolo TCP/IP esses números variam entre 0 e 65.535.

Existem números de portas conhecida entre os protocolos, pois essas não variam, segue a tabela 2 com alguns exemplos das mais utilizadas:

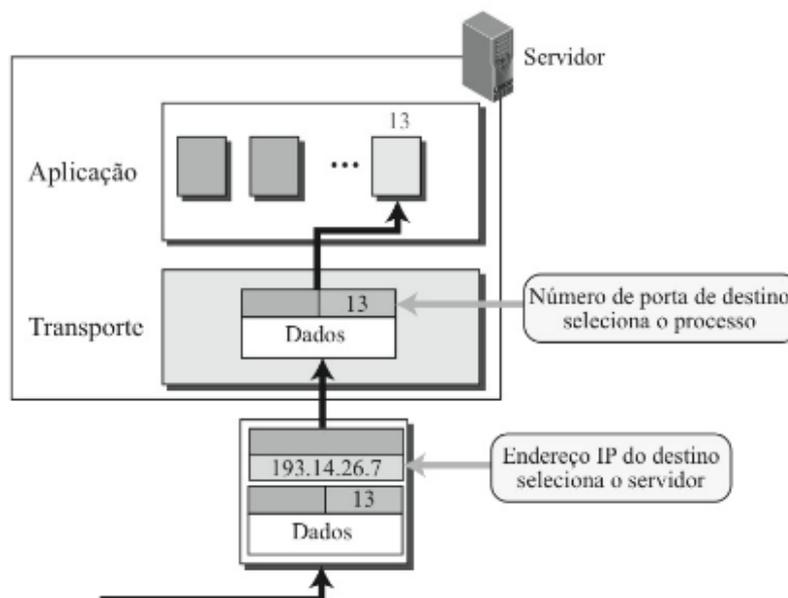
Tabela 2 - Exemplos de portas mais utilizadas

Porta	Protocolo
20	TCP FTP <i>Data File Transfer</i>
21	TCP FTP <i>File Transfer</i>
22	TCP SSH <i>Remote Login Protocol</i>
23	TCP Telnet
25	TCP SMTP <i>Simple Mail Transfer</i>
53	UDP <i>Domain Name Server "DNS"</i>
80	TCP WWW-HTTP [<i>World Wide Web</i>]
110	TCP <i>Pop3 Post Office Protocol</i>
443	TCP HTTPS <i>Protocol Over "TLS/SSL"</i>

Fonte: Do autor.

O endereço IP e os números de porta desempenham ações diferentes, o IP define para qual computador de destino a conexão deve ser estabelecida, depois de conhecer e realizar a configuração dessa rede. Já o número da porta define para qual processo se deseja executar essa conexão. Segue figura 9 com o detalhamento de execução da aplicação na porta 13.

Figura 9 - Redes de Computadores: Processo de execução



Fonte: Forouzan e Mosharraf (2013 p.142)

Neste momento são aplicados controles gerais na rede estabelecida entre os dois *hosts*. O controle de fluxo que realiza o balanceamento correto entre a quantidade de dados que devem ir em um pacote entre o emissor e o receptor. O controle de erros, que garante a confiabilidade desta rede lógica, descartando pacotes corrompidos, rastreando tanto os pacotes perdidos quanto os pacotes duplicados e eliminando os mesmos, e mantendo a ordenação correta dos pacotes de dados.

O mecanismo *Stop-and-Wait* é usado na camada de transporte quando é necessário um controle do fluxo dos pacotes enviados. Neste caso o emissor envia um pacote por vez, e espera uma confirmação do destino antes de encaminhar o próximo pacote, assim gerando uma confiabilidade de controle dos pacotes recebidos na rede. Em caso de perda de algum pacote nessa rede, ele é reenviado após um tempo, pois um dos campos do cabeçalho do protocolo TCP envia o número do byte enviado e o número do próximo byte que ele espera receba, desta forma os pacotes são gerados conforme retorno do destinatário que mantém o controle dos pacotes já recebidos.

2.4.3 Camada de rede

Responsável pela entrega *host a host*, através de protocolos de roteamento

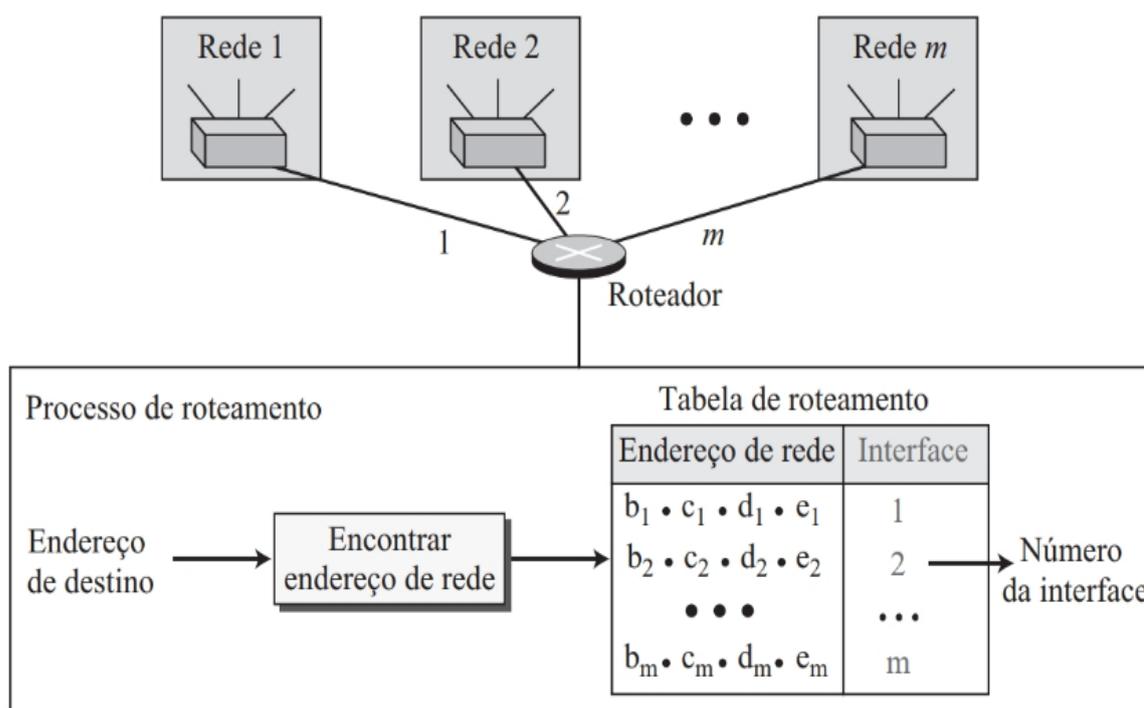
que fazem com que os pacotes encontrem o seu caminho de destino. Nesta camada é realizada a identificação de quem está encaminhando a mensagem, e quem vai receber a mesma.

O empacotamento é um deles, sendo sua principal tarefa encapsular os dados recebidos da camada superior em um pacote, que sai da sua origem indo até o seu destino. Quando chega no destino o pacote é desencapsulado e utilizado.

Para isso a camada de rede exige que o pacote de dados possua um cabeçalho com alguns dados previamente cadastrados, sendo o endereço de origem e o seu destino como informação principal, entregando o pacote para a camada de enlace de dados que realiza o encaminhamento do pacote para o destino, dentro da rede local.

Outra tarefa é o roteamento, em que a camada de rede é responsável pelo encaminhamento do pacote de sua origem ao seu destino. Porém pode existir mais de uma rota a ser traçada da origem até o destino do pacote, sendo necessário encontrar a melhor rota. Através dos protocolos de roteamento implementados na camada de rede, é gerenciado entre todos os dispositivos de rede qual o melhor caminho que o pacote pode percorrer para realizar a tarefa de chegada ao destino. Essa função é desempenhada pelo roteamento, conforme figura 10.

Figura 10 - Redes de Computadores: Roteamento



Segundo Forouzan e Mosharraf (2013 p.270) “o roteamento consiste em aplicar estratégias e em executar alguns protocolos de roteamento para criar as tabelas de decisão em cada roteador”, assim formando a melhor rota para o pacote percorrer até chegar no seu destino.

Para que o roteamento aconteça, é necessário que cada equipamento encaminhe o pacote para o equipamento seguinte. A rede pode conter diversos nós, fazendo com que existam muitos caminhos para conectar uma rede entre dois computadores. Quem executa a decisão de qual é o melhor caminho, é o serviço de encaminhamento, baseado na rota delimitada pelo serviço de roteamento que define a origem e o destino, que é a decisão de qual o próximo roteador a ser enviado consultando a tabela de roteamento.

Os mecanismos de controle de erros são pouco usados na camada de rede, sendo eles mais eficientes na camada de transporte e camada de enlace, pois na camada de rede existe a questão que o pacote neste momento, pode ser fragmentado em cada ponto de rede que passa, assim tornando a verificação de erro ineficiente.

Dentro da camada de rede o protocolo IPv4 conhecido também como *internet Protocol Version 4*, é atualmente o protocolo mais usado nas redes de computadores do mundo afirma (CANTÚ, p.51) utilizando um endereço de *bits*. Ele disponibiliza cerca de 4,29 bilhões de endereços na *internet*, isso ocasiona que com um número finito, logo esse protocolo não suportará mais endereços, pois há uma expansão cada vez maior na tecnologia mundial.

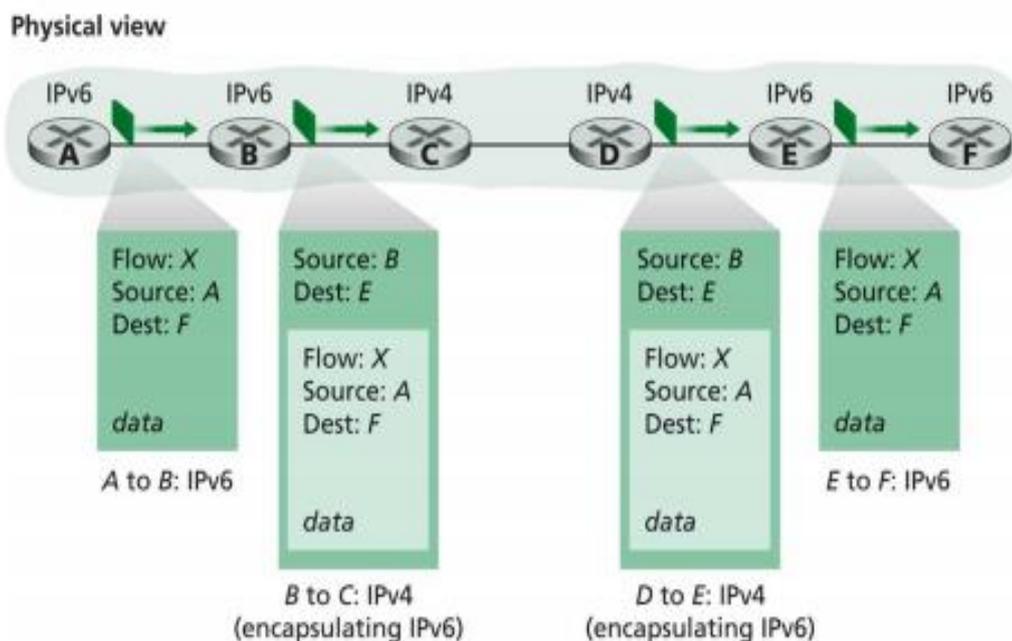
Tendo em conta que esse protocolo irá logo deixar de existir, pois há cerca de 7,67 bilhões de pessoas no mundo ,segundo o *site* (www.populationpyramid.net), e cada ser humano possui computadores, tablets, smartphones, smartwatch e demais dispositivos que se conectam a rede assim possuindo um IP único de rede, existiu a necessidade da criação de um novo protocolo para que possa atender a demanda crescente dos dispositivos de rede.

Com isso nasce o IPv6, chamado de *Internet Protocol Version 6*, que utiliza um endereço de 128 *bits*, deixando assim uma imensa amplitude de dispositivos que podem se conectar com esse protocolo de rede pelo mundo.

Esta transição de protocolos está acontecendo gradativamente, onde existe a técnica de tunelamento que realiza as transformações de datagramas de IPv6 para IPv4. Ela carrega os datagramas entre os dispositivos de rede que necessitam deste formato, como ilustra a figura 11. Atualmente os novos equipamentos conectados nas

redes de computadores estão aderindo o uso do protocolo IPv6 como padrão.

Figura 11 - Redes de Computadores: Tunelamento de um datagrama



Fonte: Kurose e Ross (2006).

2.4.4 Camada de enlace de dados

A camada de enlace é responsável por realizar a integração entre dois meios, para isso ela usa diferentes protocolos de comunicação com tipos de redes, WAN (redes de longas distâncias), LAN (rede local) e WLAN (rede local sem fio). Ela é responsável por traduzir as informações passadas da camada de rede em *bits* para que seja enviada pela camada física.

Redes locais são redes de computadores conectadas entre uma área geográfica restrita, permitindo nesta localidade que computadores e usuários compartilhem os recursos desta rede. As conexões destas redes locais entre si, formam o que chamamos de *Internet*, que por sua vez está cada vez mais crescendo no mundo. Forouzan e Mosharraf (2013 p.398) citam a seguinte frase: "A *Internet*, porém, é uma combinação de redes ligadas entre si por meio de dispositivos de conexão (roteadores ou *switches*)".

Desta forma, foi relatado que a camada de rede é quem faz a comunicação

entre dois *hosts*, através do envio de um datagrama. Com isso, essa comunicação que parte do *host* de origem acaba passando por uma série de roteadores e *switches* até chegar no seu destino. Cada equipamento pertencente a rede, como os *host* e roteadores são chamados de "nós" e o canal de comunicação entre dois nós na rede, é chamado de enlace.

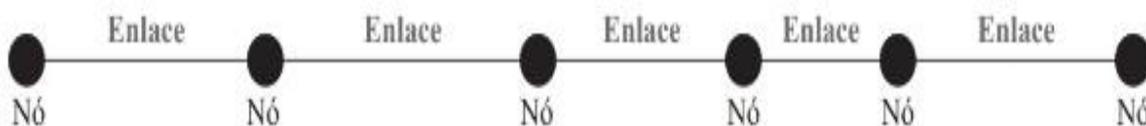
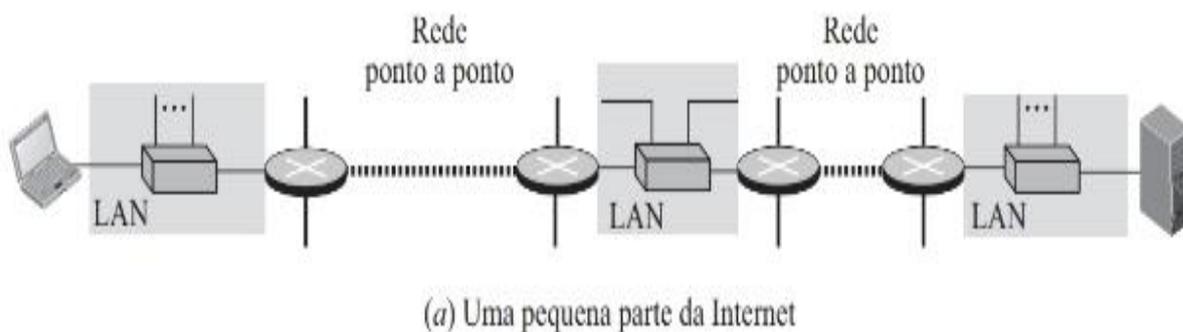
Esses enlaces podem ser suportados por diferentes tipos de tecnologias, onde cada uma delas executa a verificação e codificação do protocolo de enlace que é transmitido entre os nós. A unidade de dados trocado entre os nós é chamado de *frames*, ou em português, "quadros", que são tipicamente encapsulados no datagrama da rede.

CANTÚ (2010 p.79) diz:

Assim como os protocolos Inter rede são protocolos fim a fim que movem datagramas de um *host* a outro, os protocolos de enlace são protocolos nó a nó, movendo quadros sobre um simples enlace entre nós vizinhos.

Segue a figura 12 mostrando a interação entre os nós.

Figura 12 - Redes de Computadores: Interação entre nós



Fonte: Forouzan e Mosharraf (2013 p.399)

Encapsulamento é um dos serviços oferecido pela camada de enlace na rede, outro serviço é o de acesso ao meio físico. No caso da *Internet*, na geração do datagrama de dados, são encapsulados os dados e via um protocolo específico é

realizado o acesso do meio físico da rede.

Desta forma pode-se ter dois grandes grupos no protocolo de enlace, os protocolos de enlace ponto a ponto e os protocolos de enlace multiponto.

Segundo Forouzan e Mosharraf (2013 p.422) o protocolo ponto a ponto (PPP - *Point-to-Point Protocol*), é um dos mais utilizados para acesso no mundo. Sendo que milhões de usuários da *internet* utilizam do mesmo diariamente. Ele oferece a autenticação de conexão, criptografia de transmissão e a compressão.

Já o protocolo de enlace multiponto realiza a conexão entre duas extremidades também, porém compartilhando esse canal para que mais usuários possam usar esse recurso.

Como vários usuários podem transmitir quadros ao mesmo tempo, eles podem colidir entre si e serão perdidos, pois estão compartilhando do mesmo canal. Este é um problema que pode ocorrer dentro do formato no protocolo de enlace multiponto. Para correção desta falha, existe a determinação de quem e quando deve transmitir o quadro por este canal.

Para segmentar uma ordem de troca de quadros por esse meio compartilhado, foram desenvolvidos diversos protocolos que realizam o controle dos nós.

Desenvolvido no início de 1970 pela Universidade do Havaí, o ALOHA é um protocolo que corrige as colisões que acontecem no meio multiponto.

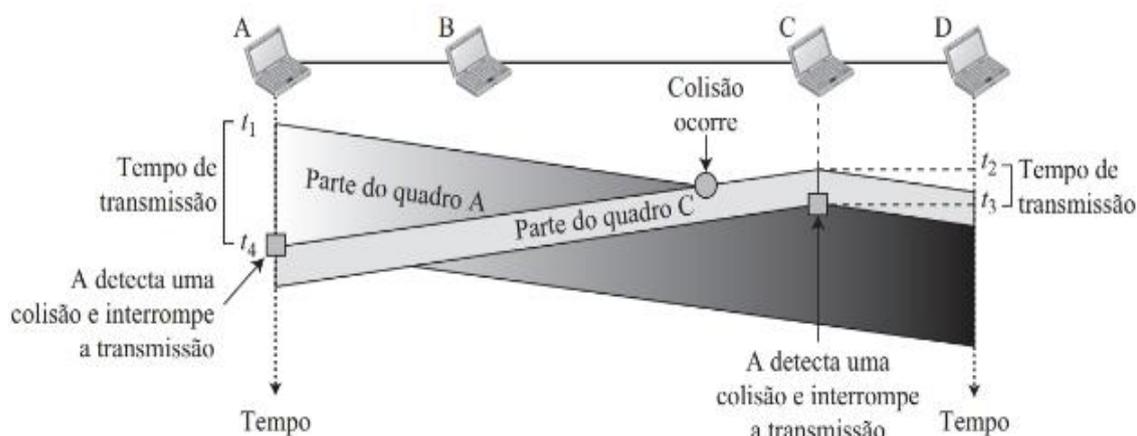
Ele permite que o usuário transmita o quadro sempre que tiver dados a serem enviados. Naturalmente haverá colisões, e com isso os quadros serão destruídos. Para corrigir isso foi implantada a propriedade de *feedback* da difusão, que é uma verificação do transmissor sobre o quadro que foi enviado. Ele pode ver quando o seu quadro é recebido pelo destinatário, caso depois de um certo tempo estipulado o destinatário não envie a confirmação de recibo do quadro, é constatado que existiu uma colisão no caminho e o quadro é reenviado.

Uma melhoria do protocolo ALOHA é o CSMA/CD (*Carrier Sense Multiple Access/Collision Detection*) ou Detecção de Acesso Múltiplo de Portadora/Detecção de Colisão. Este por sua vez faz com que o nó escute o canal, caso o canal estiver livre ele transmite o quadro, se o canal estiver ocupado ele volta ao ponto inicial e depois de decorrido um certo tempo randômico tenta uma nova transmissão do pacote voltando a escutar o canal.

Caso um nó comece a transmitir ao mesmo tempo que outro, é realizado o procedimento de detecção de colisões, onde o nó continua ouvindo o canal enquanto

transmite, em caso de colisão dos pacotes no nó, ele para imediatamente a transmissão e é refeito o envio do pacote pelo *host* emissor, conforme figura 13.

Figura 13 - Redes de Computadores: Detecção de colisões



Fonte: Forouzan e Mosharraf (2013 p.435)

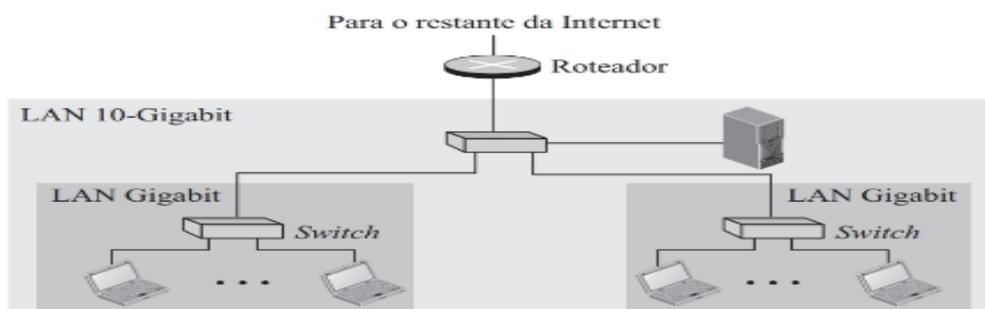
Na conexão da camada de enlace, existem muitos dispositivos de conexão que operam para a criação da rede, entre eles estão os *hubs*, *switches*, cabos e roteadores.

O cabeamento que conecta as redes de computadores possui uma metragem específica de transporte de dados. Baseado pela estrutura e o material que o mesmo é construído, por se tratar de um meio físico eles possuem especificações de transporte de dados prelimitadas já.

Para que possam ser realizados os envios de pacotes para áreas mais longas, são usados outros dispositivos.

Segue figura 14 com o detalhamento de uma rede com roteador.

Figura 14 - Redes de Computadores: Rede com roteador



Fonte: Forouzan e Mosharraf (2013 p.486)

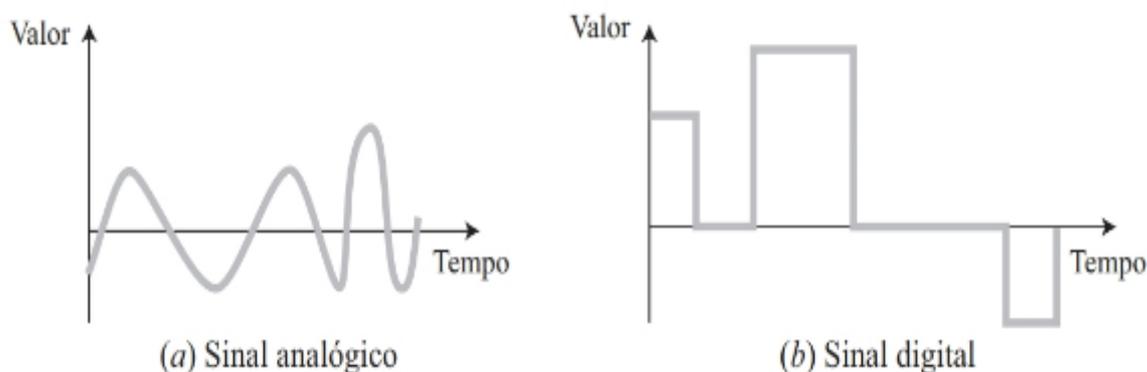
2.4.5 Camada física

É a camada que realiza o envio e o recebimento das mensagens entre os *hardwares*. O principal motivo desta camada é fazer com que as informações que estão sendo enviadas ao receptor, cheguem de forma íntegra, sem perda ou distorções.

Essa comunicação é nó a nó, porém trocando sinais eletromagnéticos. Afirma Forouzan e Mosharraf (2013 p.565) "camada física, cujo papel, é importante ressaltar, é receber os *bits* da camada de enlace de dados e convertê-los em sinais eletromagnéticos para a transmissão".

Esses dados eletromagnéticos podem ser gerados de duas formas: analógico ou digital, conforme figura 15.

Figura 15 - Redes de Computadores: Dados analógico ou digital



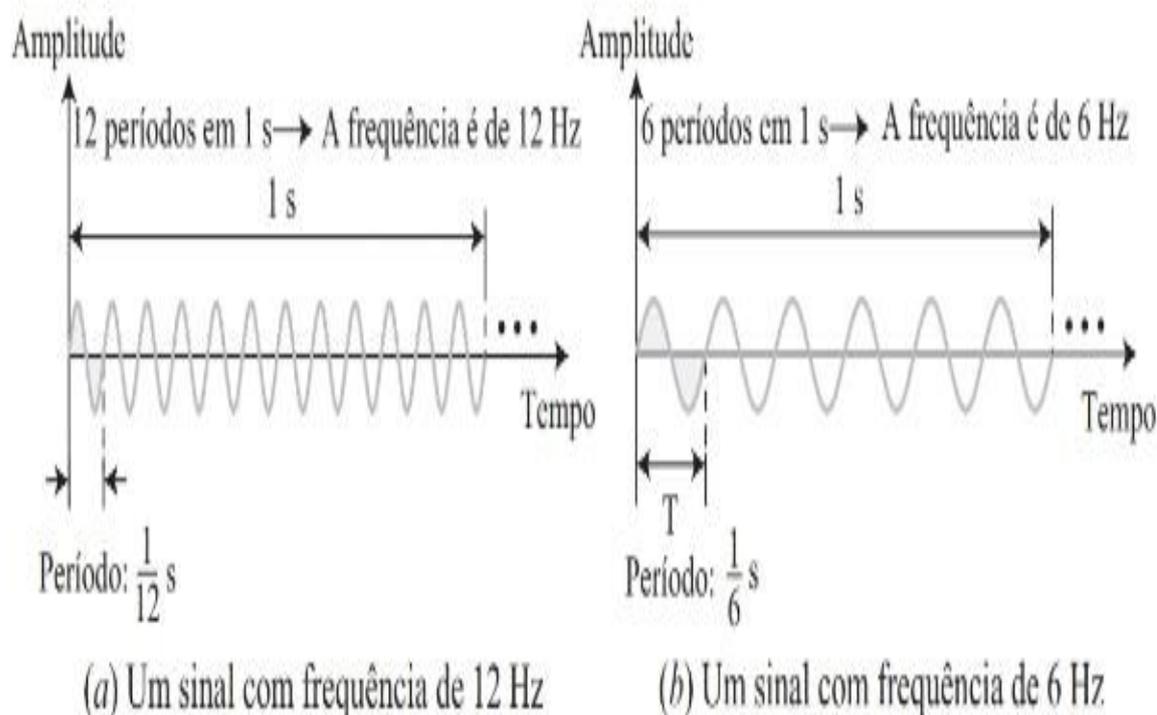
Fonte: Forouzan e Mosharraf (2013 p.567)

Os dados analógicos possuem um valor contínuo, apresentando um número muito grande de níveis de intensidade em um certo intervalo de tempo. Fazendo com que o mesmo percorra um maior espaço geográfico. Isso pois o sinal analógico é formado por uma onda senoidal, que essa é representada por três parâmetros, a amplitude de pico, a frequência e a fase.

A amplitude de pico é proporcional a energia que o sinal transporta, sendo o mesmo um valor absoluto, normalmente é medido em *volts* (V). A frequência, geralmente é medida em *Hertz* (Hz), refere se ao número do período em um segundo da onda que ela se repete. E a fase descreve a posição da onda em relação ao instante zero de tempo do sinal.

Segue figura 16 com o detalhamento de uma onda senoidal.

Figura 16 - Redes de Computadores: Onda senoidal



Fonte: Forouzan e Mosharraf (2013 p.568)

Desta forma há o comprimento da onda, que é a distância onde o sinal viaja em um período. O comprimento vincula o período de uma onda senoidal à velocidade de propagação no meio físico, quando esse período termina, a onda acaba.

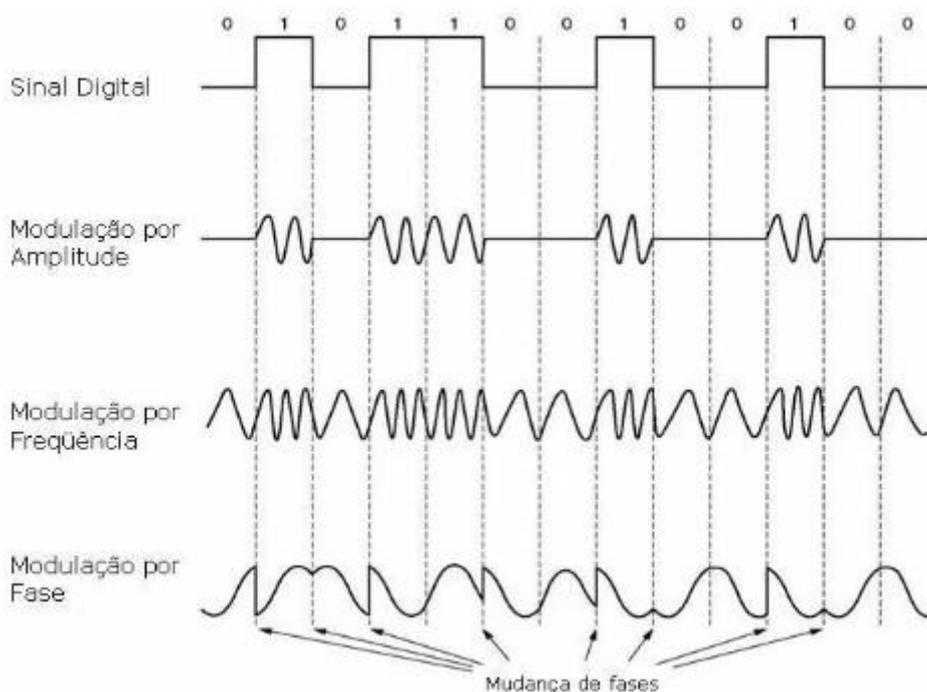
Já o sinal digital é discreto (descontínuo) no tempo e em amplitude, sendo assim ele só é definido em um certo espaço de tempo, com isso assumindo um espaço de tempo finito.

O sinal digital é uma sequência de pulsos de voltagem transmitido via um meio de comunicação físico. Se comparado com o analógico ele é mais barato, menos suscetível a interferência, sofre maior atenuação e é somente enviado via um meio físico cabeado.

Para atingir um espaço geográfico maior, é realizado a técnica de modulação do sinal, que consiste em aumentar a energia do sinal, essa modulação pode ser por amplitude, frequência ou fase, dependendo muito da necessidade da rede.

Conforme figura 17 com desenho da modulação.

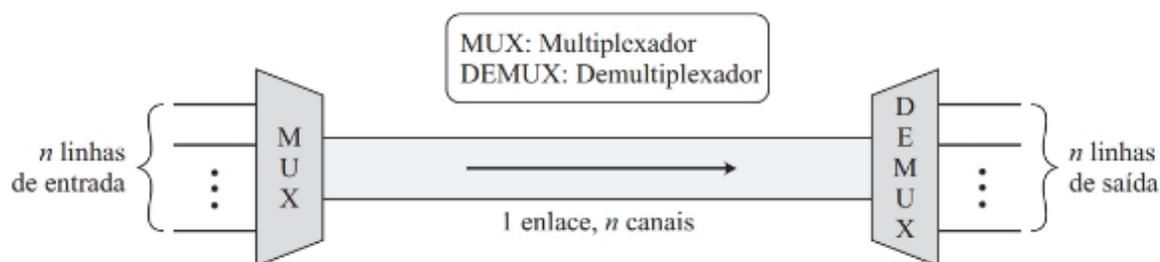
Figura 17 - Redes de Computadores: Modulação digital



Fonte: Manik (2017)

A técnica que permite diversos sinais de transmissão em um único enlace de dados é a multiplexação. No sistema multiplexado, existem muitas linhas compartilhando a largura da banda do enlace selecionado, sendo direcionados até o seu destino, quando o destinatário recebe os dados do enlace ele passa por um demultiplexador, que separa o fluxo em componentes de transmissão, e direciona o mesmo para a linha de saída correspondente ao cabeçalho do enlace, conforme figura 18.

Figura 18 - Redes de Computadores: Multiplexador



Fonte: Forouzan e Mosharraf (2013 p.600)

Para realizar essa troca de dados entre um *host* e outro é necessário um meio

de transmissão de dados. Normalmente é um cabo metálico, fibra ótica ou pelo ar.

Conhecidos como meios guiados, os cabos podem ser de diversos formatos e materiais, cada um com suas especificações, sendo os mais utilizados os cabos de par trançado, cabos coaxiais e os cabos de fibra ótica. Onde a função principal do mesmo é interligar através de um canal dois meios de comunicação, conduzindo um sinal por esse caminho.

Já os meios não guiados, transportam ondas eletromagnéticas sem usar um condutor físico, sendo assim utilizam da propagação de ondas pelo ar, denominada então uma conexão wireless.

3 METODOLOGIA

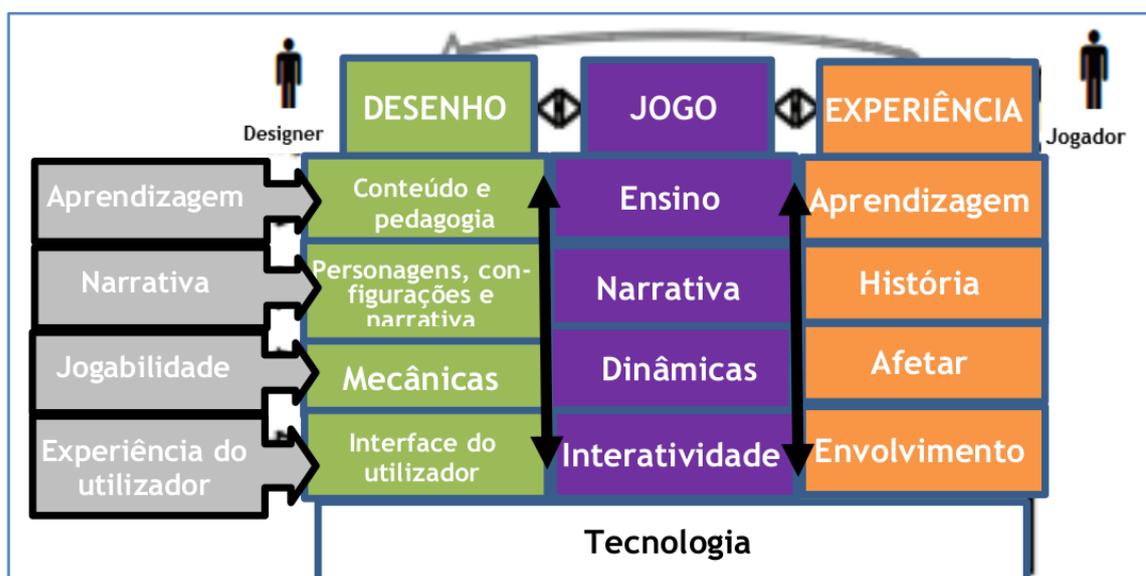
A metodologia escolhida para desenvolvimento do *serious game* deste trabalho de conclusão de curso foi a do framework DPE (*design, play and experience*) ou projetar, jogar e experimentar. Criado por Brian Winn (2008), ela é considerada expansão do modelo MDA (*mechanics, dynamics and Aesthetics*) mecânicas, dinâmicas e estéticas.

A metodologia DPE oferece funções para gerenciar e avaliar um jogo sério, visa descrever a relação de design e jogador, com as especificações dos conceitos que correspondam a camada projetada. Formado por quatro camadas bases (aprendizagem, narrativa, jogabilidade e experiência) cada uma delas com as três fases respectivas (*design, play e experience*).

Desta forma, usando as camadas do modelo DPE, o jogo é formado, cria-se o(s) personagem(s), a narrativa e o objetivo principal do jogo. Em seguida o jogador experimenta este jogo, analisando o mesmo e retornando sua experiência, verificando se o jogo corresponde com a sua proposta de *serious game*.

O *framework* DPE, fornece a linguagem e o processo para a criação de um *serious game*, diz Winn (2008). A utilização deste modelo pode diminuir muitos dos problemas encontrados em outras metodologias adaptadas para o desenvolvimento de jogos sérios.

Figura 19 – Framework DPE expandida



Fonte: Brian Winn (2008).

Nesta representação percebe-se que o modelo propõe iterações, onde começa no *design*, segue para a jogabilidade e retorna para a experiência obtida do jogador. Caso necessário o ciclo é retomado até atingir os objetivos da experiência planejada para o jogo, conforme figura 19.

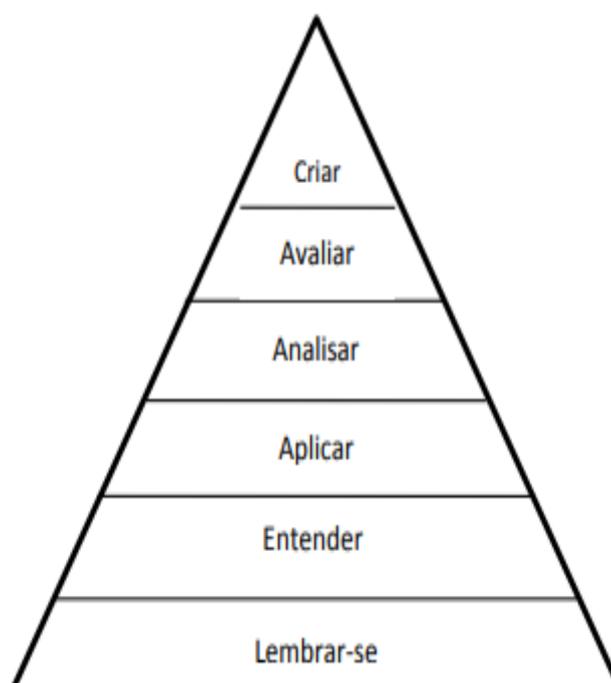
3.1 APRENDIZAGEM

Nesta camada é realizada a elaboração do conteúdo, juntamente com a parte pedagógica do jogo. O objetivo da aprendizagem deve ser estabelecido claramente neste momento, para que em seguida possa realizar o desenvolvimento do jogo baseado no que o jogo irá ensinar.

Visando a experiência do jogador, o aprendizado deve ser o resultado do objetivo proposto pelo design do jogo, assim, posteriormente, ao elaborar a proposta do conteúdo, poderá ser medido se o jogador obteve ou não o aprendizado.

A Taxonomia de Bloom (1956) para ensinar e aprender, criada por Benjamin Bloom, é a sugerida por Winn (2008), como recurso para estruturar e gerar resultados da aprendizagem em um *serious game*. Ela é uma organização hierárquica de objetivos educacionais, que modela e classifica os diferentes níveis de cognição humana, conforme figura 20.

Figura 20 - Aprendizagem Baseada em Problemas



3.2 NARRATIVA

Narrativa é a construção de uma história dentro de um cenário em um jogo, e pode ser proposta de duas formas: com a perspectiva do jogador, ou a do designer. A do designer é a proposta que ele cria para introduzir o assunto abordado no *serious game*, já a do jogador é a história que o mesmo criou ao jogar o jogo.

Conforme afirma Winn (2008), ela ocorre durante o jogo, combinando com o que foi proposto pelo designer do jogo, com as escolhas realizadas pelo jogador. Essa combinação resulta na narrativa geral vista pelo jogador ao jogar e executar os objetivos propostos.

A narrativa do designer não necessariamente deve existir, caso o desenvolvedor do jogo ache viável que em certo nível do jogo não necessite de narrativa, ele pode desenvolver dessa forma, porém o jogador sempre terá a sua narrativa, pois ela retrata a história dos desafios do jogo perante ao proposto naquele dado momento.

Winn (2008) afirma que a narrativa deve condizer com o objetivo proposto na camada de aprendizagem, para que ambas trabalhem sobre o mesmo contexto. Mas que na criação de um *serious game*, isso geralmente dificulta o processo de execução do objetivo proposto de aprendizagem para o jogador.

3.3 GAMEPLAY

A camada que delimita o que o jogador pode ou não executar no jogo. Baseada na narrativa do jogo, ela se divide em três ramificações, que são: mecânica, dinâmica e afetos.

Segundo Winn (2008), mecânicas são as regras que definem a operação do jogo, sendo assim é o que o jogador pode fazer, os desafios que ele vai ter e os objetivos do jogador.

Essas mecânicas são atingidas através do teste do jogo propriamente feito pelo jogador.

Dinâmicas são os comportamentos resultantes das interações dos jogadores com as regras, “comportamento resultante quando as regras são instanciadas ao longo do tempo a partir da influência das interações do jogador” (WINN, 2008, pg. 16,

tradução nossa 3)”.
As emoções e experiências do jogador, é o afeto do mesmo pelo jogo.

Desta forma o designer do jogo pode alterar as informações do mesmo adequando ele a um certo grupo de “usuários”. Para Winn (2008), é realizado um balanceamento do jogo nesse momento, pois são feitos protótipos e revisões do jogo para se adaptar ao público alvo.

Uma das formas mais comuns de balancear um jogo e através do nível de dificuldade, outra forma é baseada na frequência com que as recompensas são disponibilizadas. Geralmente são disponibilizadas melhores recompensas para níveis mais difíceis do jogo, isso faz com que haja um mérito por parte do jogador ao executar tal ação complexa ter um ganho maior.

Balancear o jogo baseado nas habilidades do jogador, Winn (2008) afirma que o jogador deve ter uma progressão no nível do jogo. Que os desafios devem ser mais fáceis no início por estar aprendendo a jogar, e que conforme o jogo vai se passando, os desafios aumentam conforme os objetivos são alcançados.

3.4 EXPERIÊNCIA DO USUÁRIO

É o momento que o jogador relaciona visivelmente como o jogo foi projetado para ele jogar. A interface, a interatividade e o engajamento dos jogadores são analisados através das interações do mesmo.

Neste momento são verificados os resultados desejados pelo jogador, isto quanto ao ponto de vista de um *serious game*.

Para o designer é o momento de submergir o jogador no mundo do jogo, envolvendo o mesmo para realizar as experiências dele com a narrativa.

Baseado na experiência do jogador que se dá o sucesso ou o fracasso do jogo proposto, pois é nesse momento que as métricas são avaliadas e o *feedback* perante o jogo todo é disponibilizado.

3.5 TECNOLOGIA

A tecnologia pode facilitar, como também limitar o projeto de um *serious game*. Capacidades e limitações da tecnologia a ser implantada resultam diretamente com a experiência do jogador com o jogo. No design do jogo as mesmas devem estar bem alinhadas com o objetivo geral do *serious game*.

Para Winn (2008), a escolha de *design* do jogo é mais dependente que a escolha da tecnologia, pois a experiência do usuário é mais vinculada a sua experiência jogando, do que tecnologia empregada ao realizar o jogo.

Baseada na tecnologia que é focado o público alvo do jogo, pois podemos criar os jogos para diferentes plataformas, como exemplo jogos somente para celulares ou somente para computadores, e com isso estamos restringindo um publicou ou outro.

4 ESTUDO DE CASO

Neste capítulo, foi realizada uma avaliação baseada com os conceitos do modelo de *framework* DPE do jogo Netsim, que é uma proposta de um jogo sério para ensinar alguns assuntos de redes de computadores.

Esse estudo serviu como inspiração para a elaboração do jogo proposto neste trabalho.

4.1 NETSIM

O jogo CS4G Netsim foi desenvolvido por Erinn Atwater e Cecylia Brocovich com a finalidade de ensinar alguns conceitos de redes de computadores.

Este jogo é composto por 13 níveis diferentes, onde cada um deles aborda um assunto sobre redes de computadores, partindo de um nível básico de conhecimento até um mais avançado, proporcionando ao jogador o aprimoramento do conhecimento baseado na dificuldade dos níveis propostos.

Projetado em uma interface *web*, disponível em (<https://netsim.erinn.io/>), o game disponibiliza desafios que vão aumentando conforme a dificuldade de resolver os *puzzles* aumentam também. É necessário conhecimento de níveis mais simples para poder resolver os mais avançados. Desta forma impondo um nivelamento de dificuldade no jogo.

Segue figura 21 com a interface *web* do mesmo no menu do jogo.

Figura 21 - Tela de menu Inicial

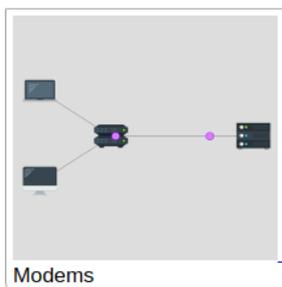
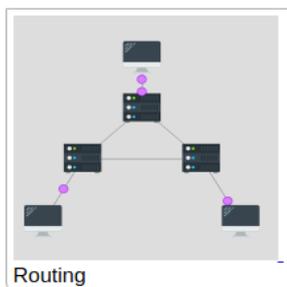
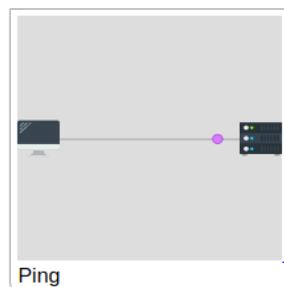
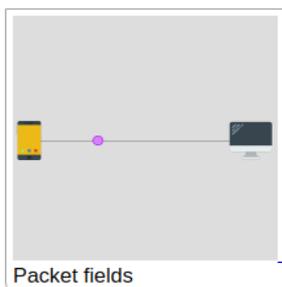
Netsim

Welcome to Netsim! If this is your first time playing, we recommend you start from the first level below, and work your way forward.

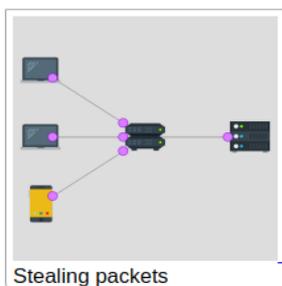
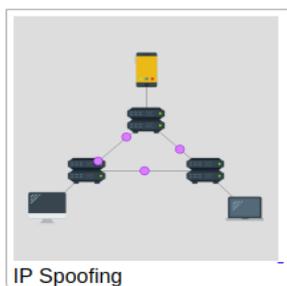
Log out

Please note that this project is still in **beta**. If you find any bugs, you can report them to [@errorinn](#) or open an issue on [Github](#).

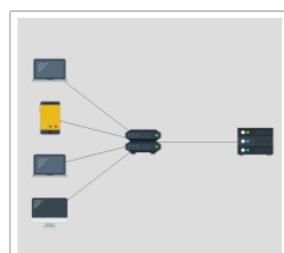
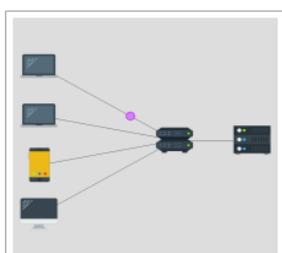
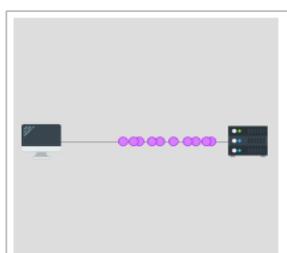
Basics



SpooFs



Denial of Service



Fonte: Do autor.

Não existe restrição por idade ou gênero para este jogo, porém é necessário

que o jogador compreenda a língua inglesa para poder resolver os *puzzles*, pois o game ainda não possui tradução para o português.

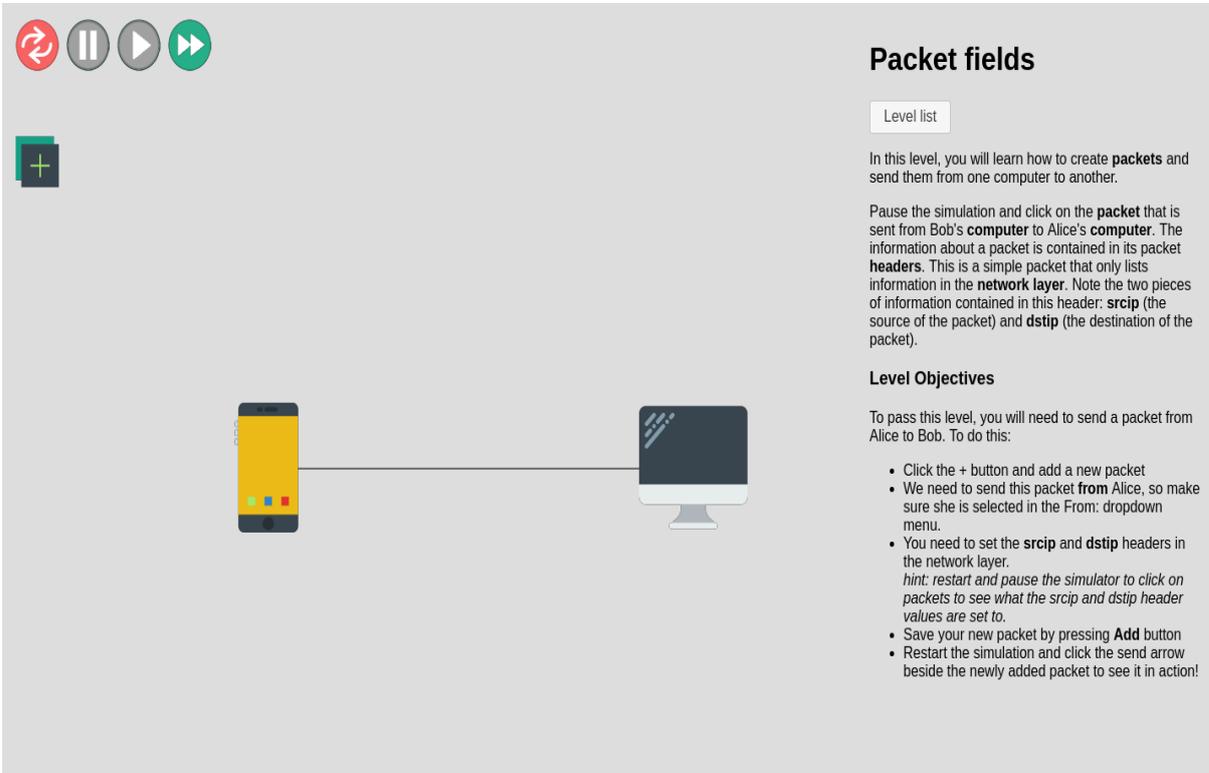
Também não é necessário conhecimento prévio em redes de computadores, pois ele aborda os assuntos simples deste tema, assim deixando um público alvo mais abrangente, e seus *puzzles* possuem a explicação necessária para resolução.

O jogo aborda os conteúdos de encaminhamento de pacote entre duas redes, a execução do comando *ping*, o roteamento, como funcionam os modems, perda de pacotes, mascarar um IP, ataques de negação de serviço e como mapear os locais para onde houve o encaminhamento de um pacote.

4.1.1 Narrativa Netsim

O jogo não possui uma narrativa ou história para envolver o jogador, porém cada nível do jogo traz uma explicação do que deve ser feito para realizar o desafio com sucesso e assim subir de nível conforme visto na figura 22.

Figura 22 - Explicação a direita do que deve ser realizado



The screenshot shows the Netsim game interface. On the left, there are control buttons: a red circular arrow for 'Restart', a grey square with two vertical bars for 'Pause', a grey square with a right-pointing triangle for 'Play', and a green square with two right-pointing triangles for 'Fast Forward'. Below these is a green square with a white plus sign for 'Add Packet'. The main area shows a yellow smartphone on the left and a grey computer monitor on the right, connected by a horizontal line representing a network link. On the right side of the interface, there is a 'Packet fields' section. It contains a 'Level list' button, a paragraph of text explaining the level's goal, a 'Level Objectives' section with a list of instructions, and a 'hint' section.

Packet fields

Level list

In this level, you will learn how to create **packets** and send them from one computer to another.

Pause the simulation and click on the **packet** that is sent from Bob's **computer** to Alice's **computer**. The information about a packet is contained in its **packet headers**. This is a simple packet that only lists information in the **network layer**. Note the two pieces of information contained in this header: **srcip** (the source of the packet) and **dstip** (the destination of the packet).

Level Objectives

To pass this level, you will need to send a packet from Alice to Bob. To do this:

- Click the + button and add a new packet
- We need to send this packet **from** Alice, so make sure she is selected in the From: dropdown menu.
- You need to set the **srcip** and **dstip** headers in the network layer.
hint: restart and pause the simulator to click on packets to see what the srcip and dstip header values are set to.
- Save your new packet by pressing **Add** button
- Restart the simulation and click the send arrow beside the newly added packet to see it in action!

Fonte: Do autor.

Os elementos deste jogo são compostos por cinco tipos (*notebook*, *switch*, computador, servidor e celular), conforme figura 23.

Figura 23 - Elementos

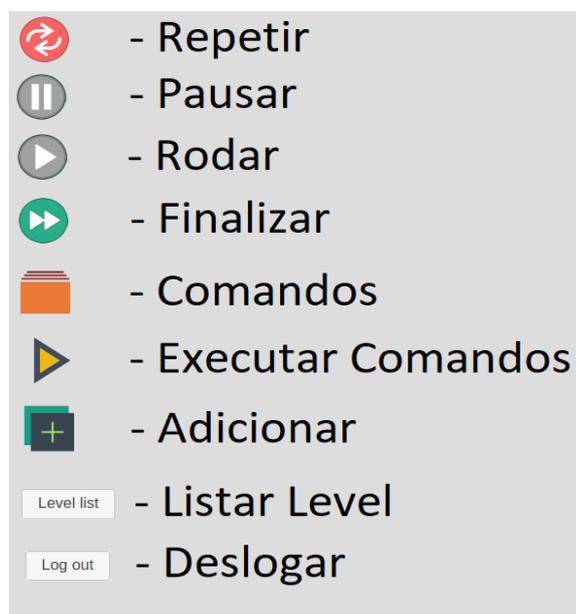


Fonte: Do autor.

4.1.2 Gameplay Netsim

Para explicar melhor como funciona o *gameplay* deste jogo, a figura 24 mostra o detalhamento de cada botão. A funcionalidade de cada um deles está representada pelo texto informado ao lado, possibilitando assim um melhor entendimento.

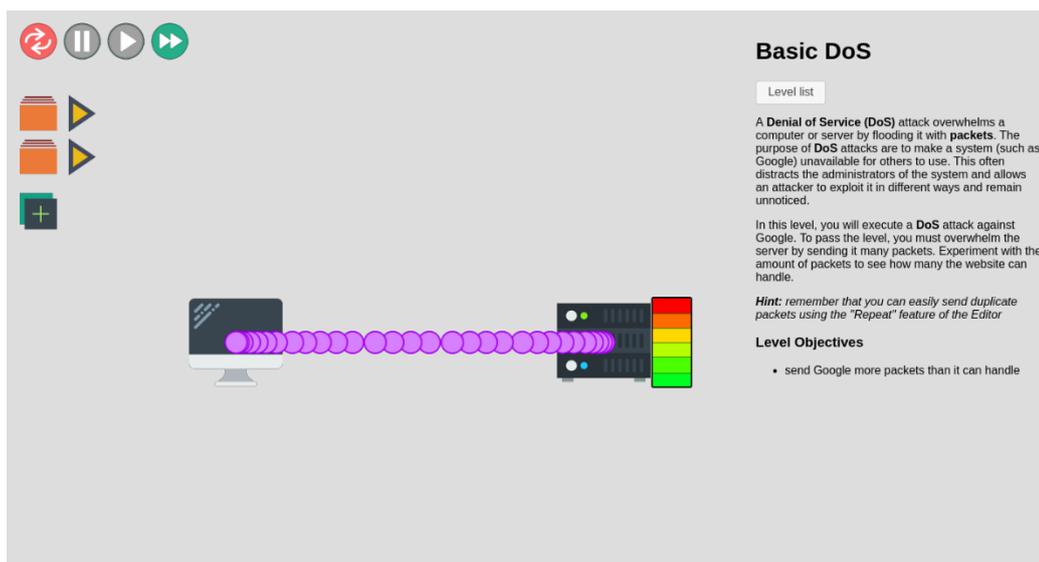
Figura 24 - Especificações botões



Fonte: Do autor.

O *gameplay* executado neste jogo, é baseado em duas interfaces de comunicação, sendo uma delas a superior esquerda que contém os botões (repetir, pausar, rodar, finalizar e adicionar) figura 25, e a outra interface a de edição dos comandos, figura 26, que é acessada ao clicar no botão “Comandos”.

Figura 25 - Tela do jogo



Fonte: Do autor.

Figura 26 - Tela de comandos

Add packet
✕

Sent from:

Alice
▼

network

srcip:

dstip:

transport

proto:

ttl:

application

type:

key:

Repeat:

Remove
Add

Fonte: Do autor.

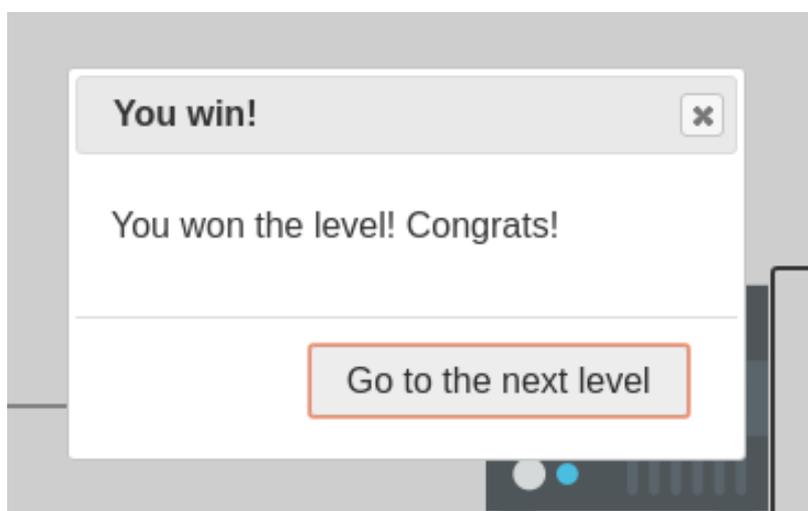
Os comandos utilizados para execução das tarefas, são disparados através de cliques do mouse nos botões. Assim como a entrada dos dados pelo teclado quando necessário.

Ao clicar no botão “Adicionar”, abre a janela de inserção dos comandos, conforme figura 25, o mesmo é editável. Ao preencher os dados e executar o botão “add”, é adicionado esse comando na tela do jogo e o botão fica disponível visivelmente, conforme figura 26.

Realizando a edição dos comandos, conforme proposto para solução do *puzzle*, o jogador clica no botão “rodar” para executar as suas instruções e aguarda o processo definido por ele terminar.

Cada nível concluído com sucesso, é gerado um alerta ao usuário informando e parabenizando o mesmo pela execução correta do desafio. Juntamente foi implementado o botão de passar para o próximo nível do jogo, conforme figura 27.

Figura 27 - Execução com êxito



Fonte: Do autor.

Caso o usuário não tenha executado o comando corretamente, não acontece nada, o jogo não emite nenhum alerta informando que a execução está errada e que tem que ser refeita, somente fica aguardando que o usuário refaça o *puzzle*.

O cenário do jogo muda conforme o nível, pois ele exibe a execução das ações realizadas pelo jogador via os comandos gerados. Como cada nível possui um assunto diferente, o cenário se adapta para realizar as ações de acordo com sua proposta do nível.

4.1.3 Experiência do jogador

A experiência do usuário como jogador é confusa, pois algumas funcionalidades básicas em jogos, estão faltando, por exemplo uma mensagem de erro ao executar uma tarefa errada, e que a mesma necessita ser refeita.

Porém o conhecimento superficial sobre o assunto é muito bem abordado. Fazendo com que o propósito de sua criação seja alcançado e assim obtendo sucesso como projeto, pois envolve o jogador a resolver o enigma para que o mesmo adquira conhecimento de como aquele específico assunto de redes de computadores acontece.

5 DESENVOLVIMENTO

Neste trabalho foi desenvolvido um *serious game* que aborda alguns assuntos da arquitetura TCP/IP em redes de computadores, utilizando o *framework* DPE (*design, play and experience*). O nome do jogo é “Entregas de Penny”.

Ao longo do capítulo cinco, ocorre o detalhamento do jogo baseado no estudo feito no decorrer deste trabalho, especificando o gênero, as mecânicas e as regras gerais do jogo.

5.1 ENTREGAS DE PENNY - GÊNERO

O gênero escolhido para realizar este jogo foi o de plataforma 2D. Ele condiciona a existência um personagem principal, que é controlado pelo jogador através de suas mecânicas.

O personagem principal pode executar ações como correr, pular, coletar objetos, dentre outras. Ele caminha sob plataformas e precisa vencer obstáculos e enfrentar inimigos para poder resolver o objetivo do jogo.

Esse gênero está presente em diversos tipos de jogos, podendo ser em 2D ou 3D, dependendo do que foi proposto pelo designer. O mesmo abrange diversas possibilidades de temáticas a serem abordadas, o que foi um ponto importante na decisão de usar este gênero como base para execução deste trabalho de conclusão de curso.

No "Entregas de Penny" foi verificado que o personagem precisa executar suas ações para poder resolver os objetivos propostos em cada nível. Desta forma o gênero plataforma 2D se encaixa no contexto de opções para fazer níveis interessantes ao jogador, e que possa gerar o conhecimento necessário que o assunto que se pretende passar neste *serious game*.

5.2 ENTREGAS DE PENNY – NARRATIVA

A história criada para este jogo é fictícia, sendo apresentada aos jogadores com o intuito de validar o objetivo do jogo, que é a criação de um *serious game* sobre o assunto de redes de computadores, mais específico na arquitetura TCP/IP. Para isso foi definido o personagem principal, a ambientação e a história em que o jogo acontece.

A personagem principal é uma menina que se chama Penny Parker, com idade de 19 anos, loira, que trabalha na empresa Rapidex. A empresa Rapidex realiza entrega de correspondências e encomendas, e seu horário de trabalho é das 8h da manhã até as 17h da tarde, tendo um intervalo das 12h às 13h para almoço.

Penny é responsável pela entrega de correspondência em uma determinada área da cidade chamada de Gwin, onde já é bem conhecida pelos clientes e tem um vasto conhecimento da região, pois trabalha há um ano naquele mesmo local.

Porém recebeu um novo desafio da sua empresa, que seria entregar correspondências em uma área que jamais tinha ido naquela cidade. O local que Penny foi enviada é chamado de Anor, e possui um vasto território a ser explorado.

Então Penny usou dos conhecimentos que adquiriu na Rapidex para resolver o problema de não conhecer a região, pois precisa executar seu trabalho de entregar a correspondência no tempo em que foi estimado.

Portanto, Penny sai de sua casa cedo, vai até o seu trabalho e realiza as funções que foram designadas para ela. No final do dia ela volta a sua casa e aguarda o seu próximo dia de trabalho.

5.3 ENTREGAS DE PENNY – MECÂNICA

As mecânicas do jogo definem o que o jogador pode realizar enquanto joga. Elas são as regras impostas pelo designer. Baseado no gênero, as mecânicas limitam o tipo de jogo que é proposto, neste caso, como será criado um jogo de plataforma, sendo que existem algumas mecânicas básicas que todo jogo de plataforma utiliza.

Elas devem ser escolhidas como componentes para explorar motivações, interesses e desejos dos jogadores. Primeiro se entende as motivações do seu público alvo, verifica o que eles acham divertido e suas restrições, baseado neste resultado se modela a mecânica do jogo.

As mecânicas podem ser interpretadas como sendo os verbos do jogo. Dentre os mais usados nos jogos de plataforma podemos citar os seguintes: mover, parar, pular, correr, segurar, voar, quebrar, pegar, atirar.

Elas também trazem elementos, como rolamento bidirecional, plataformas e estruturas que sobem, descem e deslizam, itens posicionados em locais estratégicos e adversários com modo de eliminação padrão, seja por ataque simples com pulo sobre cabeças ou por golpes simples.

Com isso a mecânica do jogo está muito ligada ao que o jogo quer realizar como tema. No caso do “Entregas de Penny”, o objetivo principal é fornecer elementos que possam auxiliar os jogadores a adquirir conhecimento do assunto TCP/IP o jogo necessita importar mecânicas que subsidiem essa condição.

A movimentação pode ser tratada como a principal mecânica. Como o propósito do jogo é a criação de um jogo 2D, os movimentos básicos são o de se mover para frente, se mover para trás, parar e pular. Possibilitando as oito direções (incluindo diagonais 2D) de acompanhamento ao personagem central, de forma fluida, dinâmica e rápida.

Outras mecânicas que estão presentes nos jogos, são as de segurar um determinado objeto, tendo que mover ele de um lado para o outro, empurrar determinados elementos do cenário, coletar itens dispersos no nível, executar botões e entrar em portas.

Entregas de Penny foi realizado por níveis, cada um deles possui suas próprias mecânicas impostas para resolver o objetivo principal do nível. Dentre todos os níveis do jogo foram implantadas as seguintes mecânicas: movimentar para direita e esquerda, pular, mover objetos, abrir portas e coletar itens.

5.4 ENTREGAS DE PENNY – APRENDIZAGEM

De acordo com a proposta DPE, é essencial definir as metas de aprendizagem logo na construção inicial do jogo. Para isso foi avaliado que o jogo “Entregas de Penny” tem como base ensinar princípios da arquitetura de rede TCP/IP, através da experiência do jogador é medida a sua aprendizagem.

Levando em conta que o público alvo para este *serious game* são pessoas que já possuem um conhecimento sobre redes de computadores, é proposto uma aprendizagem simples e de fácil compreensão aos jogadores.

O jogo é estruturado a abordar um conteúdo por vez da arquitetura TCP/IP, que foi dividido em cinco níveis, cada um deles recebendo seu assunto e a proposta de *gameplay* conforme conteúdo abordado.

Através dos comandos que o personagem possui, o jogador utiliza dos mesmos para mover o personagem para atingir o objetivo do nível. Todos os comandos usados estão explicados no “Menu Inicial” do jogo, dentro da tela “Tutorial”, para que o jogador

possa compreender o que pode realizar durante o jogo.

Este *serious game* é estruturado para desafiar o jogador em suas habilidades motoras em fazer o personagem principal passar pelas plataformas e executar algumas ações para resolver o objetivo do nível.

O ato de aprendizagem envolve a repetição de atividades pelo jogador no determinado nível. Fazendo com que o mesmo fixe o ato que está realizando e associando com o conteúdo abordado pelo nível.

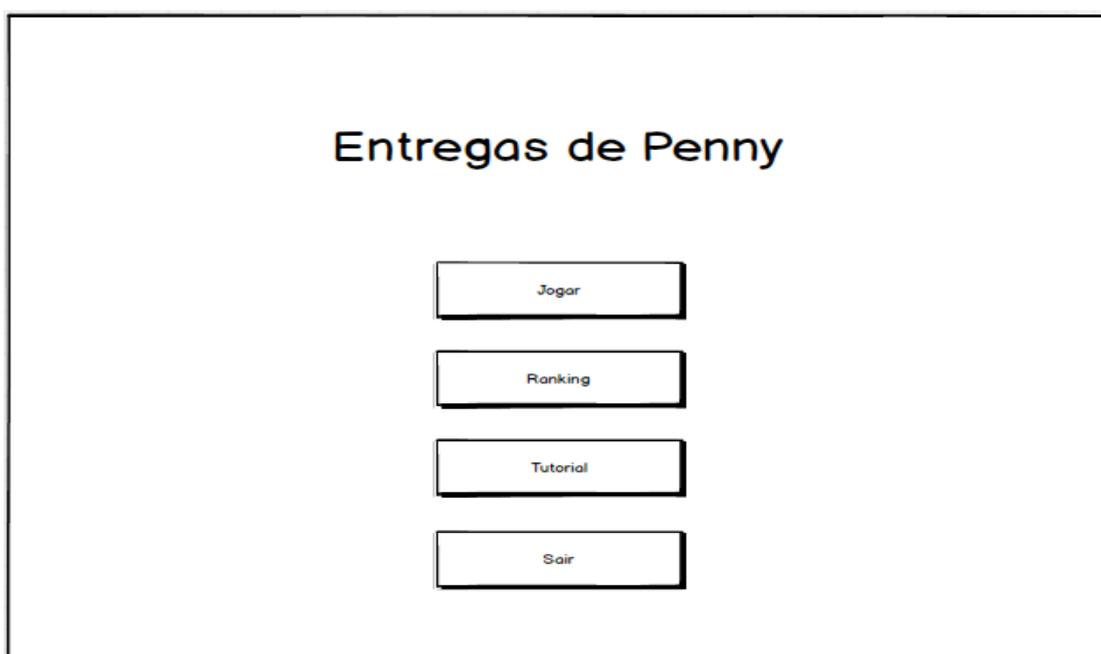
Intenciona-se que caso o jogador consiga resolver o objetivo principal de cada nível, ele obtenha o conhecimento gerado deste nível, pois a associação do conteúdo com a *gameplay* é explicada no final de cada nível.

5.5 ENTREGAS DE PENNY – JOGABILIDADE

Entregas de Penny, é um jogo proposto com cinco diferentes níveis, cada nível aborda uma das camadas da arquitetura TCP/IP.

Inicialmente existe uma tela de menu conforme protótipo de tela a figura 28, que tem o objetivo de guiar o jogador, informando qual funcionalidade ele pretende executar.

Figura 28 - Entregas de Penny - Protótipo tela "Menu Inicial"



Fonte: Do autor.

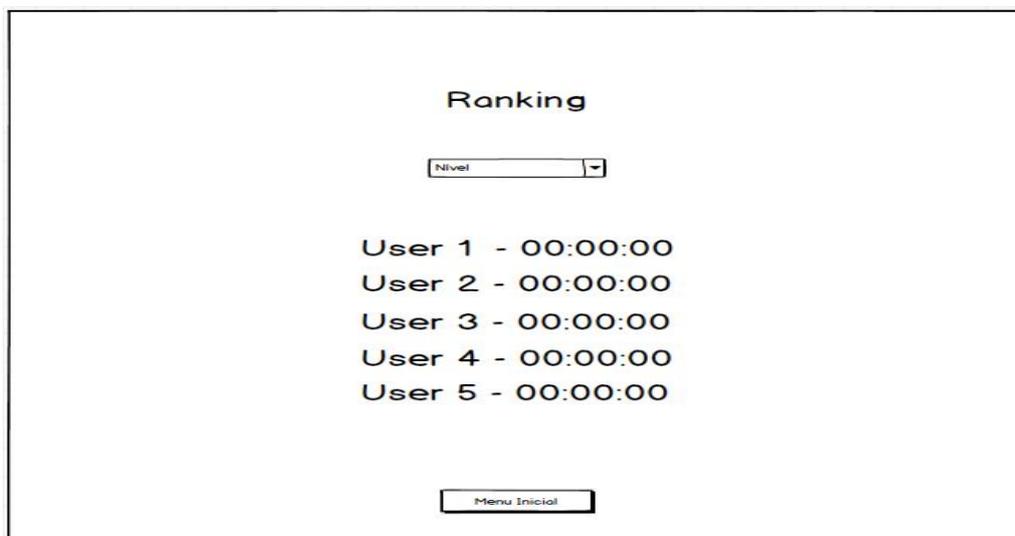
Porem a tela acabou sendo modificada e ficou conforme a figura 29, sendo removido o botão “Ranking”, essa funcionalidade foi retirada do jogo, conforme abordado a seguir.

Figura 29 - Entregas de Penny – Tela “Menu Inicial”



Conforme planejado na estruturação inicial, o botão “Ranking” iria trazer o melhor tempo de execução para cada nível conforme o protótipo na figura 30, juntamente com o nome do jogador. Contudo, essa funcionalidade foi removida, pois não há necessidade de validar o tempo em um *serious game*, visando que a métrica principal deste jogo, não é concluir o nível rápido, mas sim obter o conhecimento que ele tem a oferecer.

Figura 30 - Entregas de Penny - Protótipo tela Ranking



Fonte: Do autor.

O botão "Tutorial" traz um review do jogo, com suas mecânicas e funcionalidades, conforme figura 31. Dentro da tela de tutorial, o botão "Voltar" faz o jogador retornar ao menu inicial.

Figura 31 - Entregas de Penny – Tela "Tutorial"



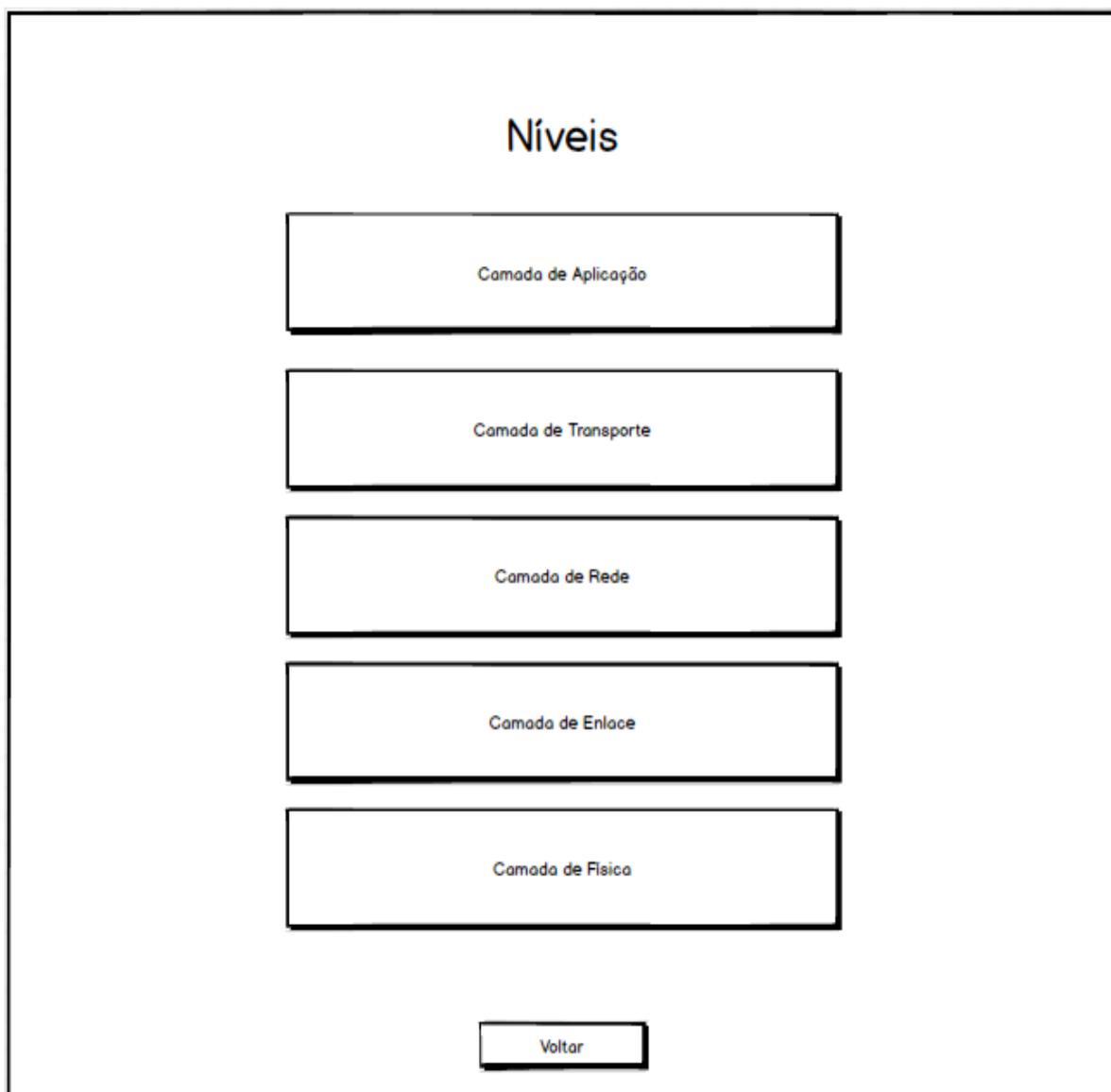
Fonte: Do autor.

O botão “Sair” do menu inicial, conforme figura 29, tem a função de fechar o jogo.

Ao clicar no botão “Jogar” do menu, é aberta uma nova tela que mostra os cinco níveis disponíveis ao jogador, cada nível com o seu nome, baseado nos nomes das camadas da arquitetura TCP/IP.

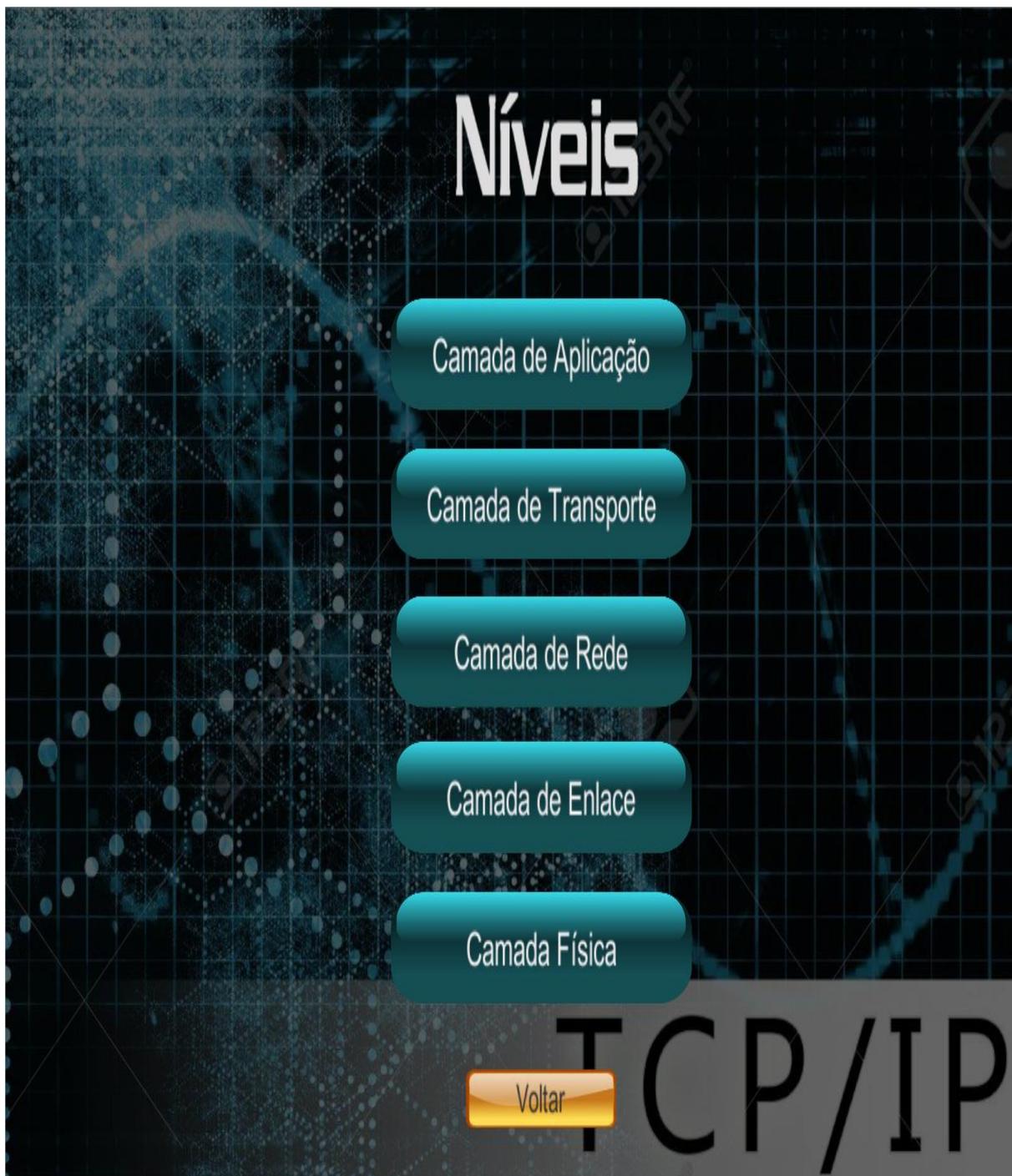
O jogador seleciona o nível que quer participar clicando no mesmo e é enviado para a próxima tela do jogo. A figura 32 ilustra como foi projetada a tela de níveis, e a figura 33 mostra como ficou a tela desenvolvida. O botão “Voltar” nesta tela retorna o jogo para o menu inicial.

Figura 32 - Entregas de Penny - Protótipo: Tela “Seleção de Nível”



Fonte: Do autor.

Figura 33 - Entregas de Penny - Tela “Seleção de Nível”



Fonte: Do autor.

Antes de entrar propriamente no jogo, em cada nível, haverá uma tela com a explicação resumida do assunto que se trata aquele nível na arquitetura TCP/IP. Essa tela tem dois botões, um para que o jogador possa avançar e jogar o nível (botão “Jogar”) e o outro com a funcionalidade de voltar ao menu inicial (botão “Menu Inicial”), conforme pode ser visto no protótipo de tela na figura 34.

Figura 35 - Entregas de Penny – Tela de Explicação: Camada de Aplicação



Fonte: Do autor.

Figura 36 - Entregas de Penny - Tela de Explicação: Camada de Rede



Fonte: Do autor.

Figura 37 - Entregas de Penny - Tela de Explicação: Camada de Rede



Fonte: Do autor.

Figura 38 - Entregas de Penny - Tela de Explicação: Camada de Enlace



Fonte: Do autor.

Figura 39 - Entregas de Penny - Tela de Explicação: Camada Física

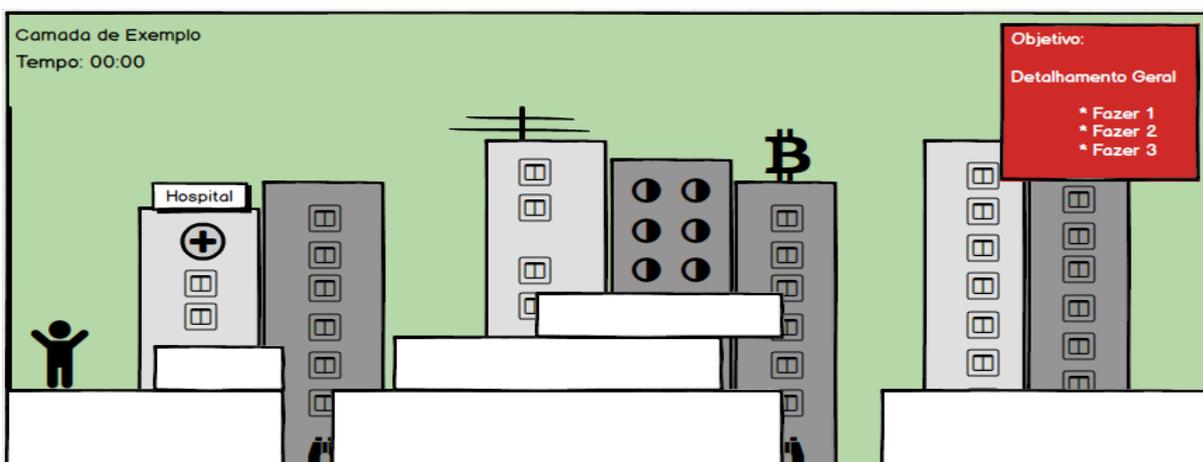


Fonte: Do autor.

Selecionando o botão “Jogar”, o jogador será direcionado ao nível escolhido, e terá como objetivo completar o mesmo com sucesso, através da mecânica implementada. O detalhamento do objetivo do nível ficará disponível no canto superior direito da tela do jogo. Cada nível tem seu próprio objetivo a ser executado, conforme o assunto que é tratado no mesmo.

O protótipo, mostrado na figura 40 informa como foi planejado o layout na tela do jogo.

Figura 40 - Entregas de Penny - Tela do jogo



Fonte: Do autor.

O *layout* foi modificado retirando o tempo de execução do nível do canto superior esquerdo, e adicionando um botão com a imagem “X”, que fecha o jogo em caso de necessidade do jogador. A figura 41 mostra como ficou o *layout* geral do jogo com uma breve explicação que aparece logo no início do nível 1 (Camada de Aplicação), antes do jogador iniciar a partida. Desta forma, é explicado o *layout* base do jogo ao jogador.

Figura 41 - Entregas de Penny - *Layout* Base



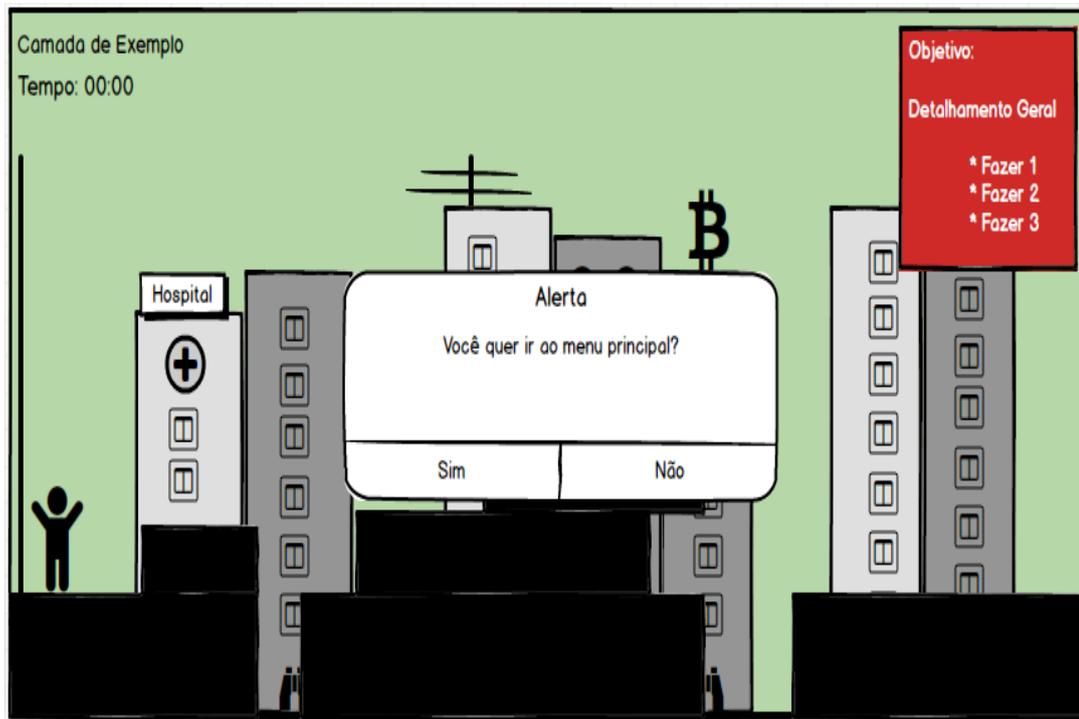
Fonte: Do autor.

Foi implementado um menu de *pause* (parar) nos níveis, caso o jogador queira sair do jogo ou reiniciar o nível que está jogando. O protótipo inicial desta função está na figura 42, porem existiu a necessidade de ser alterado ela colocando a opção de reiniciar o nível caso o jogador execute alguma ação errônea.

Para abrir essa função o jogador precisa pressionar a tecla “ESC” que será aberto um menu com as opções conforme a figura 43. Neste momento o jogo é pausado, a tela acaba ficando mais escura indicando que o jogo mudou de ação assim o jogador pode escolher uma das três ações do menu, clicando no botão “Voltar” ele

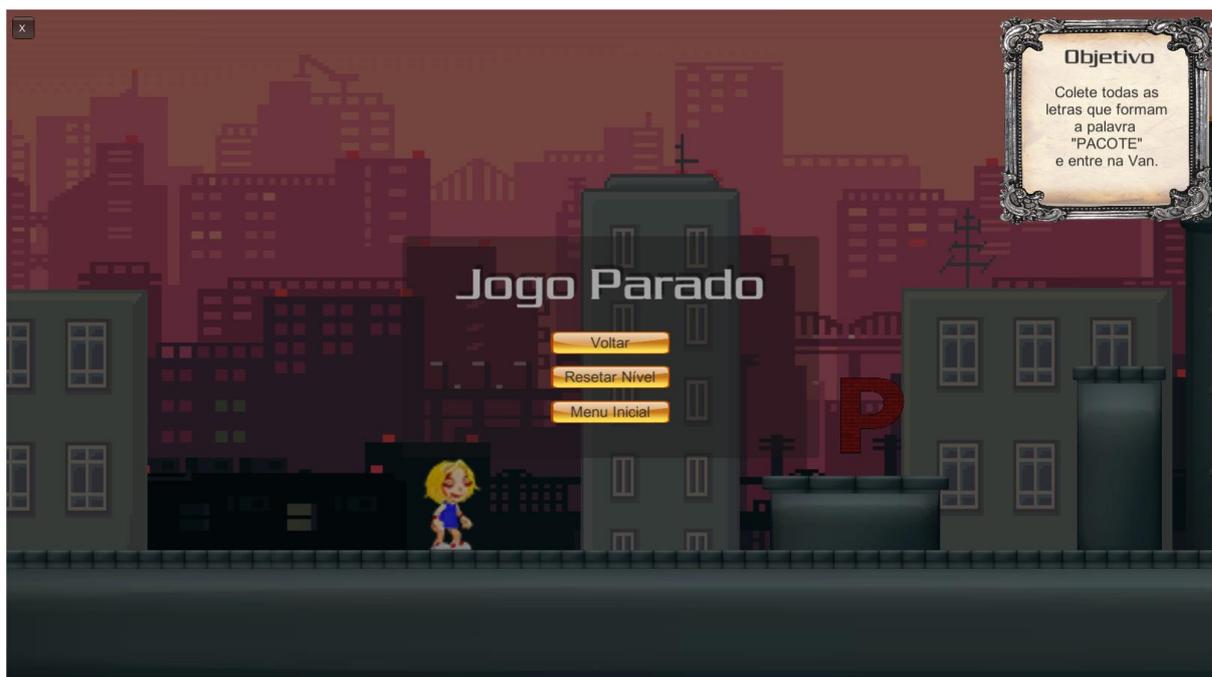
retorna ao jogo do ponto que parou, no botão “Resetar Nível” o nível atual é iniciado novamente e no botão “Menu Inicial” o jogo é retornado ao menu principal.

Figura 42 - Entregas de Penny - Protótipo: Menu *pause*



Fonte: Do autor.

Figura 43 - Entregas de Penny - Menu *pause*

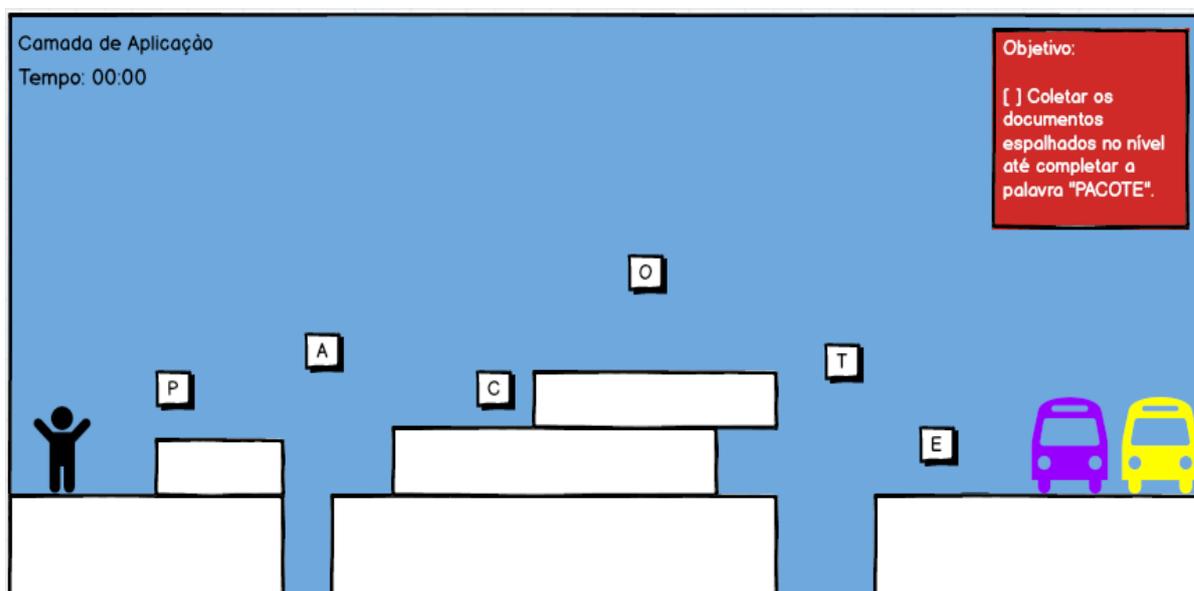


Fonte: Do autor.

5.5.1 Nível - Camada de Aplicação

O nível um é o que traz o assunto da camada de aplicação. Ele foi desenvolvido com a personagem Penny coletando os “documentos” espalhados no nível, sendo estes 6 arquivos, cada um com uma letra diferente, formando a palavra “PACOTE”, conforme o protótipo na figura 44.

Figura 44 - Entregas de Penny – Protótipo: Nível camada de aplicação



Fonte: Do autor.

Por se tratar do primeiro contato do jogador com o jogo, foi criado logo no início do jogo uma explicação da narrativa principal do jogo, conforme figura 45. Assim explicando a história base, contextualizando o jogo.

Figura 45 - Entregas de Penny - Explicação inicial



Fonte: Do autor.

Em seguida, depois de passar pela explicação principal clicando a tecla "Enter", vem a explicação do *layout* base, já explicado acima na figura 41.

Após passar pela explicação do *layout* o jogo é iniciado com Penny parada, esperando o jogador executar as mecânicas imposta pelo desenvolvedor para resolver o objetivo do nível até concluir o mesmo, conforme figura 46. Uma música ambiente fica tocando enquanto o nível está sendo executado.

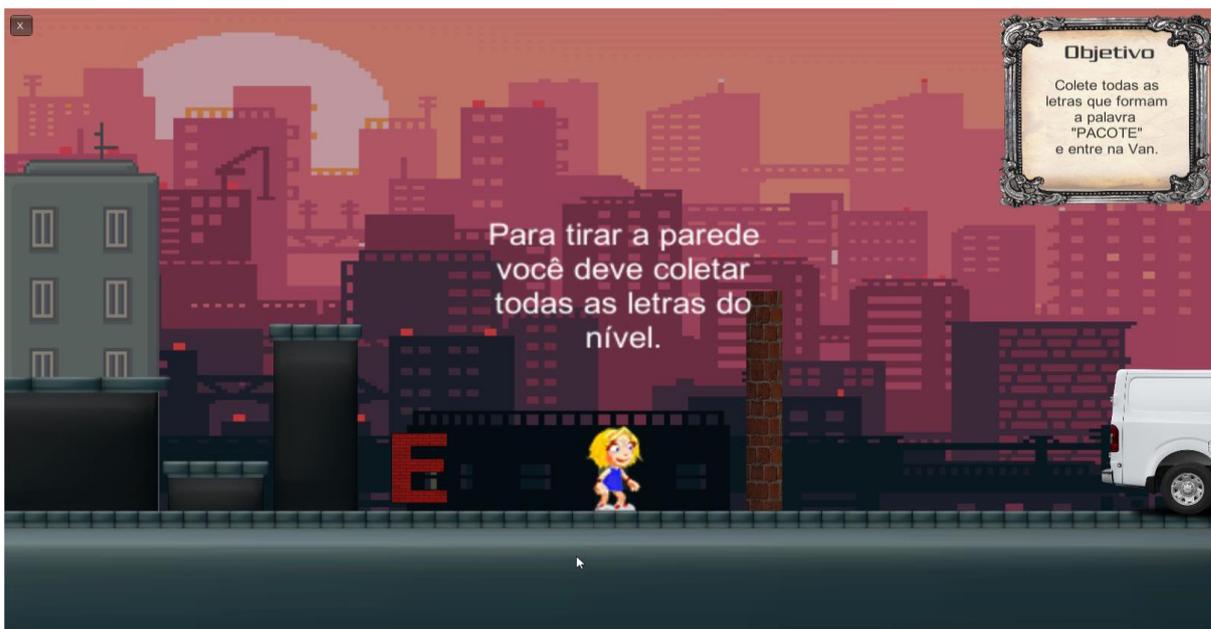
Figura 46 – Entregas de Penny - Camada de Aplicação



Fonte: Do autor.

O desafio está em passar pelas plataformas pegando os objetos “P”, “A”, “C”, “O”, “T” e “E”, ao coletar os documentos é emitido um *bip* sonoro informando ao jogador que foi realizado essa ação com sucesso. Caso o jogador não colete um desses documentos é informado no final do nível que ele não coletou todos os itens necessário para derrubar a parede de tijolo que impede de chegar até a van, ilustrado na figura 47. Assim o jogador não consegue obter o objetivo principal deste nível e é necessário reiniciar o mesmo.

Figura 47 - Entregas de Penny - Bloqueio final nível um



Fonte: Do autor.

Após coletar todos os documentos, Penny deve entrar na van que utilizará para fazer as entregas, assim completando o nível, ao se aproximar da van, é exibida uma mensagem com as instruções de como o jogador deve proceder para poder realizar essa tarefa, conforme ilustra a figura 48.

Figura 48 - Entregas de Penny - Instruções de entrar na van



Fonte: Do autor.

Após realizar o que é solicitado para entrar na van, ilustrado na figura 48, o nível termina. O jogador é redirecionado para uma nova tela, aonde é explicada a relação do nível com o conteúdo ministrado, a figura 49 mostra como é essa tela.

Nessa tela é executado uma música de vitória, fazendo com que o jogador possa ter o sentimento de objetivo cumprido, juntamente com a explicação da analogia criada do nível com o conteúdo da arquitetura TCP/IP.

Figura 49 - Entregas de Penny - Final camada de aplicação



Fonte: Do autor.

Na tela final de nível somente existe um botão, "Nível", que redireciona o jogador para a tela de seleção de nível conforme visto na figura 33.

5.5.2 Nível - Camada de Transporte

O nível da camada de transporte foi desenvolvido visando o assunto de entrega dos dados ordenadamente para o receptor. Portanto ao iniciar o nível existe uma breve explicação sobre o objetivo do nível conforme ilustrado na figura 50.

Figura 50 - Entregas de Penny - Explicação inicial camada de transporte



Fonte: Do autor.

Ao iniciar o nível, após saber que o objetivo do nível é entregar os quatro pacotes ordenadamente pela cor dos mesmos, existe um *outdoor* com a ordem correta a ser entregue, assim mostrado no jogo conforme figura 51.

Figura 51 - Entregas de Penny - Ordem dos pacotes



Fonte: Do autor.

Após saber a ordem correta dos pacotes a serem entregues, Penny chega até o primeiro pacote. Neste momento existe uma placa verde grande com a indicação ao jogador, informando de que forma ele deve executar as mecânicas do jogo para poder segurar o pacote e carregar o mesmo até o final do nível, onde o mesmo os colocará dentro da van, conforme pode-se ver na figura 52.

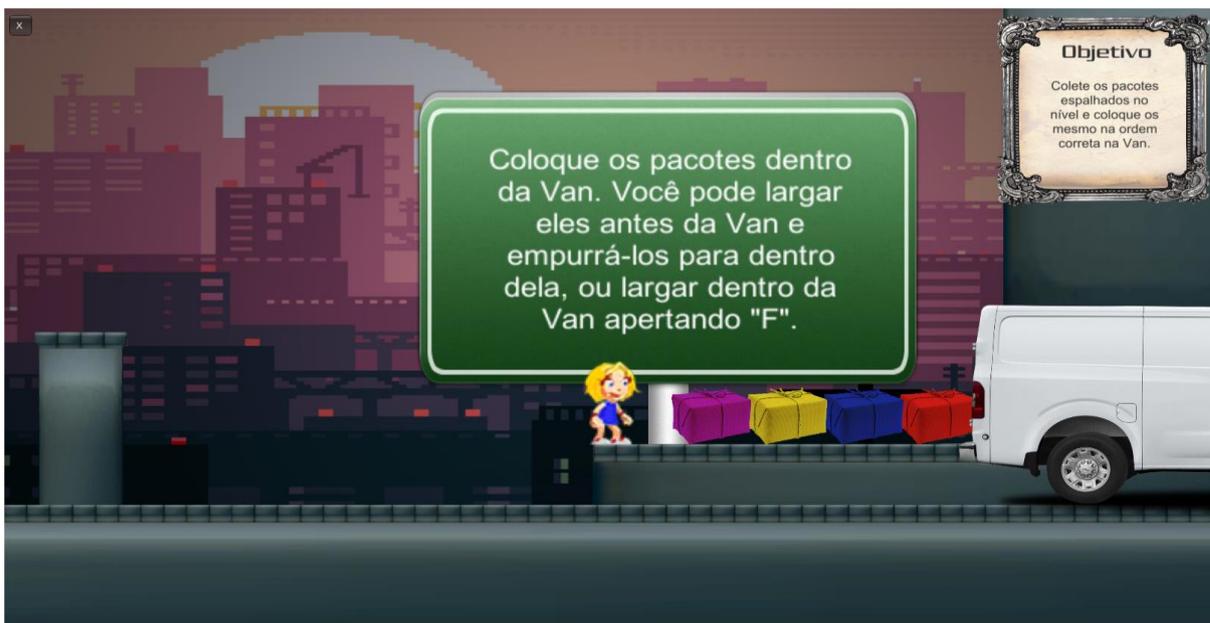
Figura 52 - Entregas de Penny - Placa informativa



Fonte: Do autor.

Após coletar os pacotes e chegar no final do nível, existe uma nova placa verde informando que existe mais de uma maneira de colocar os pacotes na van, ou colocando um a um dentro da van, ou deixando os mesmos enfileirados e empurrando eles, conforme a figura 53.

Figura 53 - Entregas de Penny - Final objetivo nível transporte



Fonte: Do autor.

Com isso o objetivo da camada de transporte é realizado corretamente e o jogador termina o necessário para concluir o nível. Assim é exibida a tela de conclusão de nível, junto com a explicação da analogia realizada pelo desenvolvedor, conforme validado na figura 54.

Figura 54 - Entregas de Penny - Final camada de Transporte



Fonte: Do autor.

5.5.3 Nível - Camada de Rede

O nível da camada de rede traz como principal objetivo o jogador descobrir em qual casa ele deve entregar a encomenda. No final do nível existem três casas de cores diferentes, sendo elas verde, azul e vermelha conforme ilustra a figura 55.

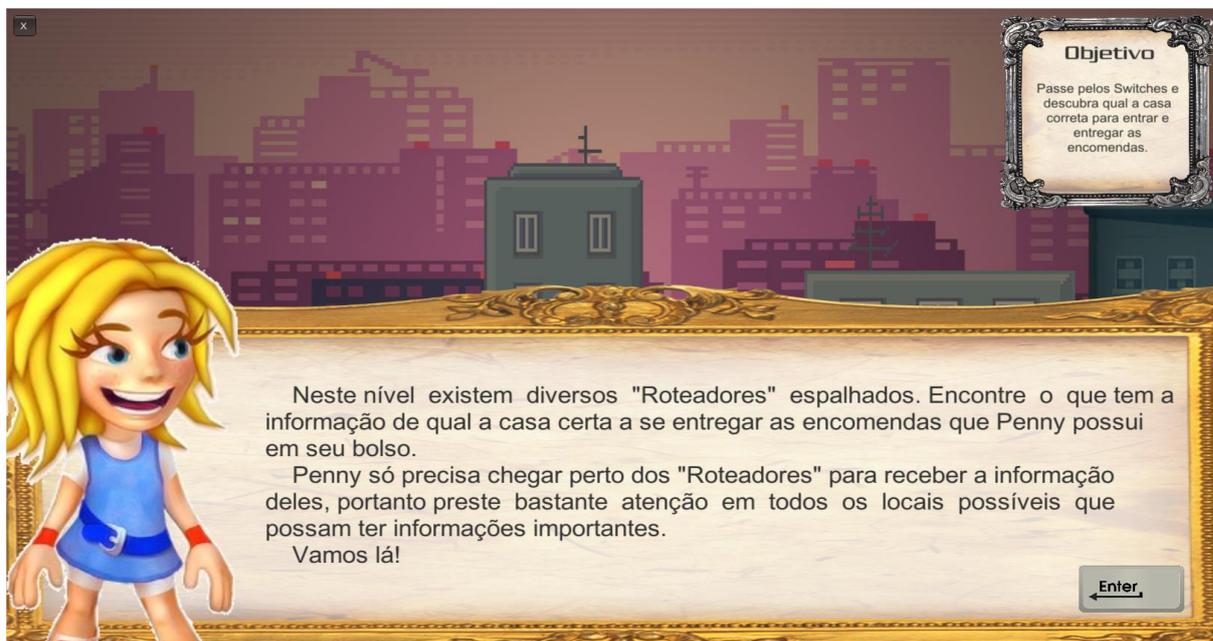
Figura 55 - Entregas de Penny - Casas nível de rede



Fonte: Do autor.

O objetivo deste nível é o jogador passar pelos roteadores espalhados pelo nível, até encontrar o roteador que tem a informação correta de qual a casa certa a se entrar, conforme explicado no início do nível na caixa de diálogo, assim como mostra a figura 56.

Figura 56 - Entregas de Penny - Caixa de diálogo



Fonte: Do autor.

Conforme figura 57, ao chegar perto de um roteador, é exibida uma mensagem na tela ao jogador, assim o jogador deve achar o roteador que tem a mensagem correta para concluir o objetivo deste nível.

Figura 57 - Entregas de Penny - Mensagem roteador



Fonte: Do autor.

O primeiro roteador que Penny encontra é o branco, que emite a seguinte informação: “Passe pelo roteador Verde, pois ele tem a informação mais correta”. Após prosseguir no nível, ela encontra o roteador verde que informa o seguinte: “Pois é, somente o roteador Vermelho sabe a informação de qual é a casa correta”. O próximo é o roteador vermelho, que informa a mensagem conforme a figura 57.

Depois o próximo roteador por qual Penny passa é o roxo, que tem a seguinte mensagem: “Não tenho certeza se é na casa Azul ou na Vermelha. Quem sabe a casa certa é o roteador Azul”. Desta forma Penny chega até o roteador azul que diz o seguinte: “Não sei qual a casa é, mas tente achar o roteador Amarelo, ele conhece essa informação”. Chegando no roteador amarelo, ele informa a casa certa através da seguinte mensagem: “Entre na casa Vermelha para entregar os pacotes corretamente, lembre-se que para entrar na casa é necessário apertar a tecla "E"”.

Com a informação certa de qual a casa a entrar, o jogador só necessita chegar na mesma e pressionar a tecla “E” para concluir o nível com sucesso.

Desta forma é concluído o nível, e aparece a próxima tela, com as informações finais sobre a analogia do nível desenvolvido com o conteúdo da camada de rede na arquitetura TCP/IP, conforme figura 58.

Figura 58 - Entregas de Penny - Final camada de rede



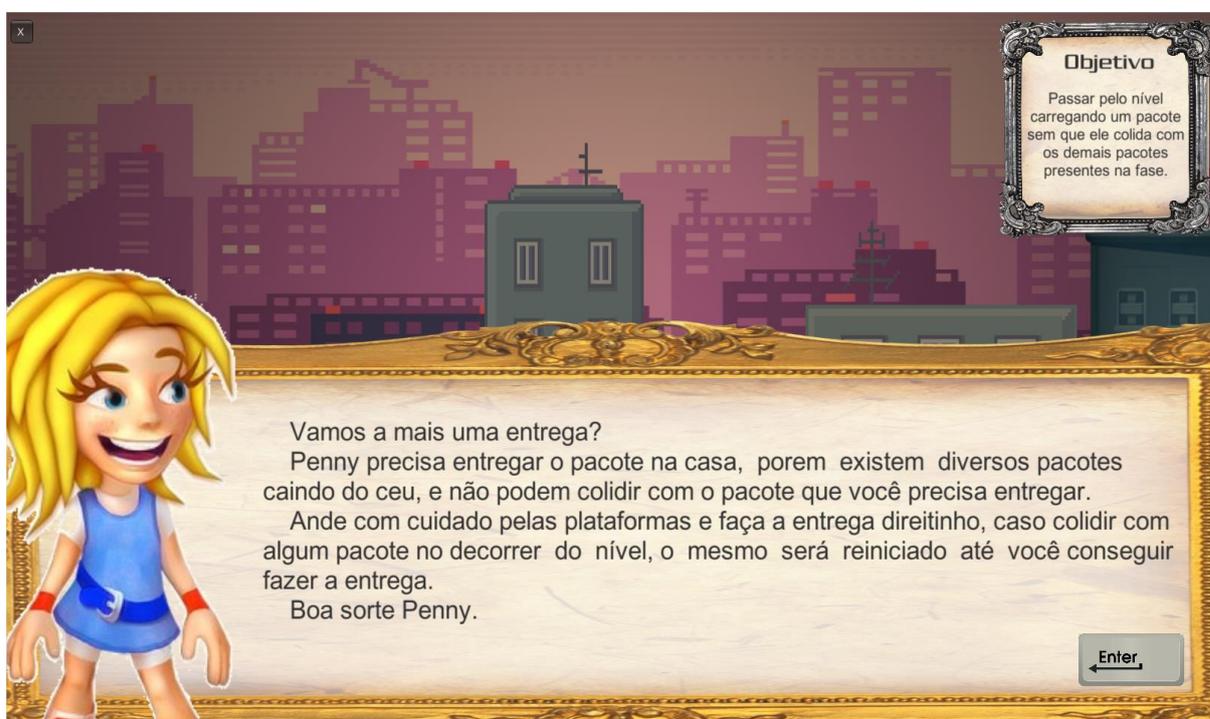
Fonte: Do autor.

5.5.4 Nível - Camada de Enlace

O objetivo para concluir a camada de enlace é percorrer o nível com o pacote a ser entregue e entrar na casa, sem que os outros pacotes (vermelhos), que ficam a todo momento caindo do céu, colidam com a Penny.

Na figura 59, existe a explicação do que o jogador deve realizar para concluir o nível. Essa é a primeira informação que ele recebe antes mesmo de iniciar o jogo.

Figura 59 - Entregas de Penny - Informação camada de enlace

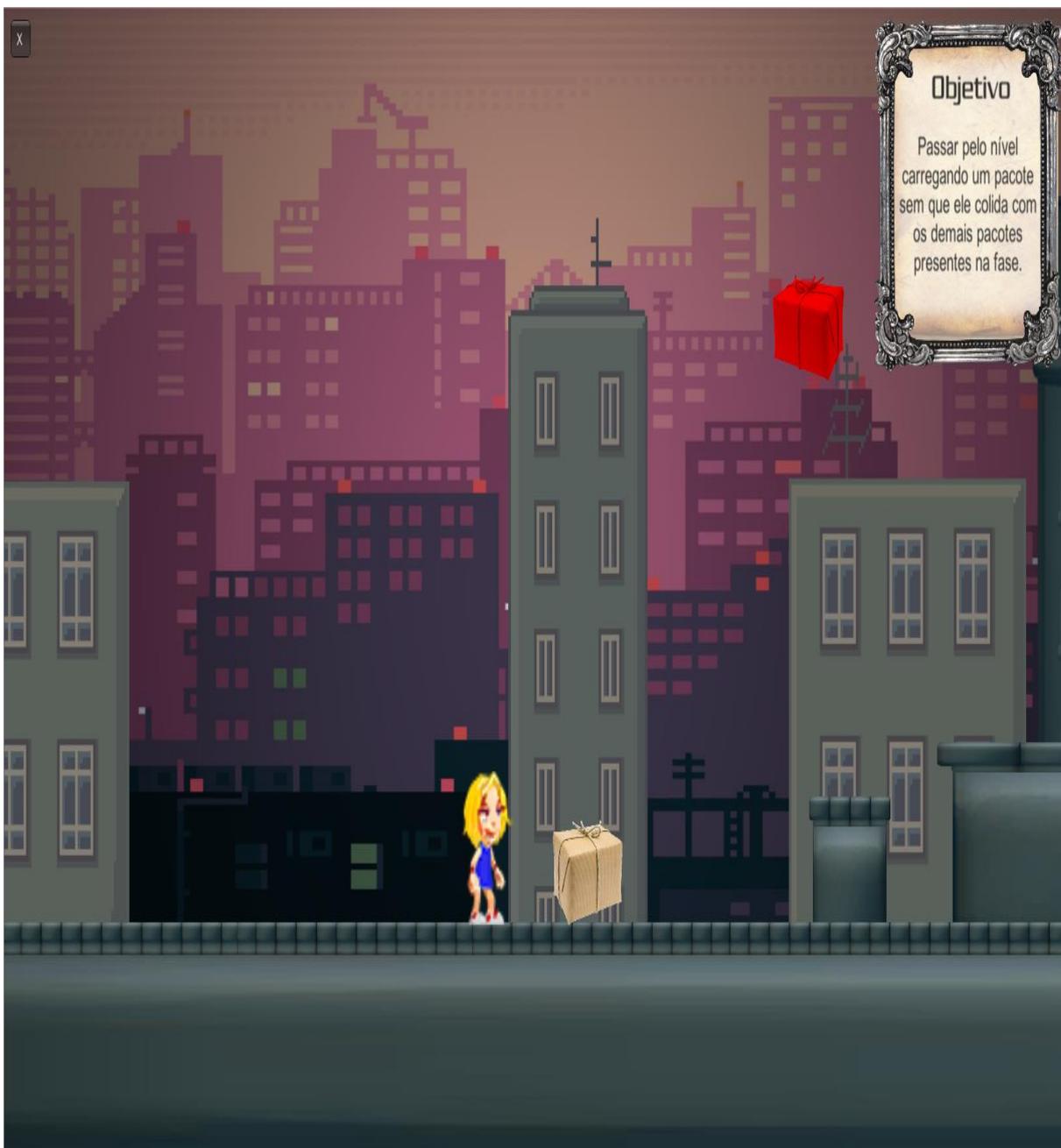


Fonte: Do autor.

Ao clicar "Enter", o jogador é levado ao nível e então os pacotes começam aleatoriamente a cair do céu, eles possuem velocidades e tempos diferentes, fazendo com que não haja um padrão de tempo e espaço desses pacotes, isso faz com que o jogo comece a ficar mais difícil do que no início, exigindo uma melhor jogabilidade do jogador.

Logo no início do nível, em frente a Penny, existe o pacote da cor amarela, que ela deve segurar e levar até achar a casa para entregar ele, conforme ilustrado na figura 60.

Figura 60 - Entregas de Penny - Pacote a ser entregue



Fonte: Do autor.

O jogador segura o pacote apertando a tecla “F” e carrega o mesmo ao percorrer o nível até chegar na casa que ele deve entregar. Enquanto isso, Penny deve passar pelas plataformas implementadas neste nível e não colidir com nenhum pacote vermelho que ficam caindo do céu, caso Penny encoste em algum pacote vermelho o nível é reiniciado.

Ao chegar na casa, o jogador apenas precisa pressionar a tecla “E” para entrar nela e fazer a entrega corretamente mostrado pela figura 61.

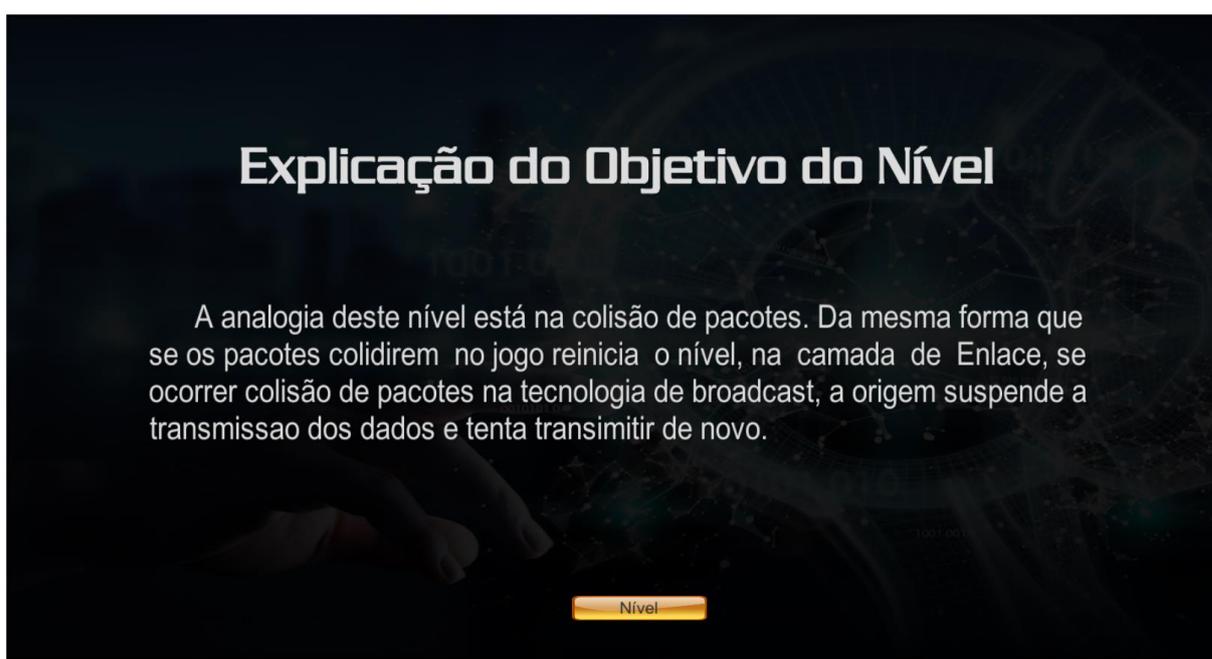
Figura 61 - Entregas de Penny - Final nível enlace



Fonte: Do autor.

Realizando o objetivo de entregar o pacote, o jogador é redirecionado para a tela de explicação do nível, onde essa explica a analogia do nível com o assunto da camada de enlace desenvolvido, conforme figura 62.

Figura 62 - Entregas de Penny - Final camada de enlace

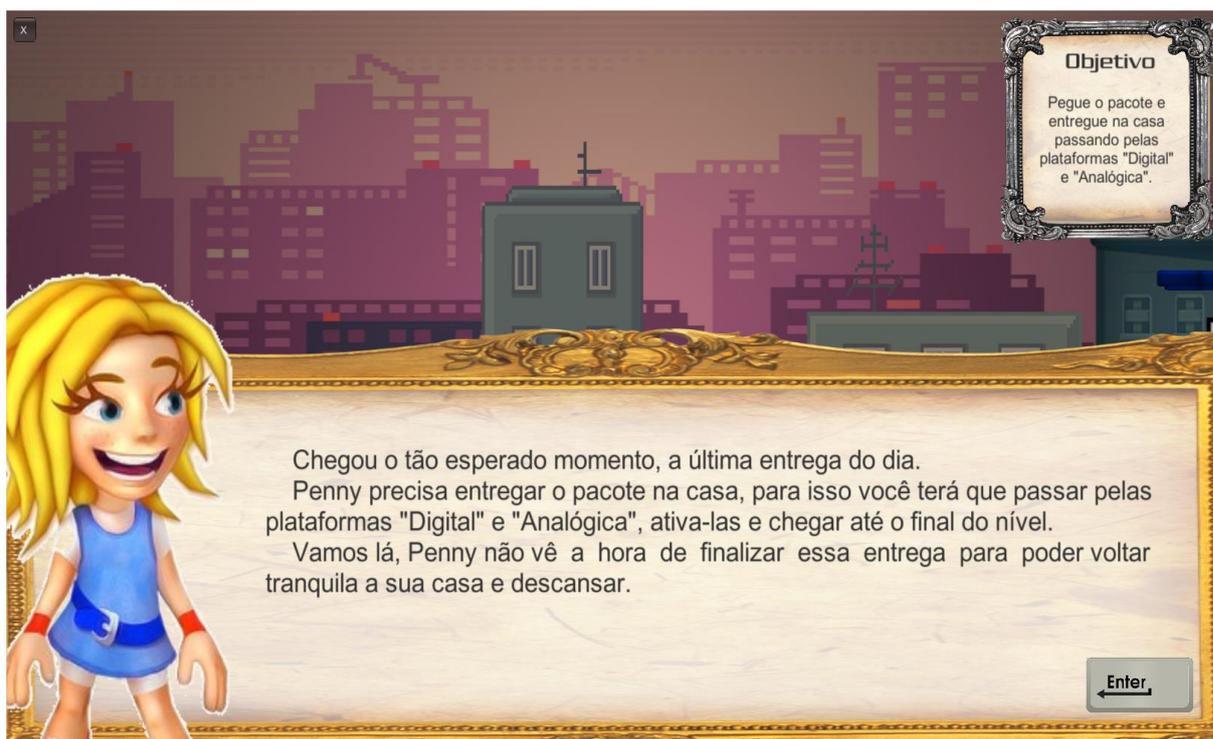


Fonte: Do autor.

5.5.5 Nível - Camada Física

Como em todos os níveis, inicialmente existe uma mensagem informando o jogador do objetivo principal para concluir o nível, junto com uma frase informando que este é o último nível do jogo, conforme figura 63.

Figura 63 - Entregas de Penny - Informações camada física



Fonte: Do autor.

No nível da camada física, o objetivo principal é Penny entregar o pacote na casa passando por duas plataformas que se movem, chamada de plataforma "Digital" e plataforma "Analógica".

Ambas possuem o seu formato conforme analogia criado com o sinal digital e analógico de uma onda senoidal na camada física da arquitetura TCP/IP.

A onda digital é formada por traços retos até chegar ao seu destino. Logo no início do nível, Penny sobe na plataforma da cor azul, que representa a plataforma digital, e existe uma placa informando o que deve ser realizado para que a plataforma comece a se movimentar, conforme ilustrado na figura 64.

Figura 64 - Entregas de Penny - Plataforma digital



Fonte: Do autor.

A figura 65 mostra como é o destino da plataforma digital, onde existe um canal que conecta ambas as plataformas.

Figura 65 - Entregas de Penny - Destino onda digital



Fonte: Do autor.

Uma das características da plataforma analógica, é de que ela possui a cor vermelha, assim diferenciando ambas as plataformas, outro detalhe é que o formato dela é em modo de ondas, fazendo assim a analogia com a onda senoidal analógica na camada de rede da arquitetura TCP/IP, como ilustrado na figura 66.

Ao subir na plataforma analógica, existe outro *outdoor* informando o que é necessário o jogador realizar, para assim fazer com que a plataforma se mova e ele possa chegar até o final do nível, já ilustrado na figura 66.

Figura 66 - Entregas de Penny - Formato plataforma analógica



Fonte: Do autor.

A figura 67 ilustra como é o final do nível, onde o jogador apenas precisa entrar na porta com o pacote em mão para concluir o objetivo.

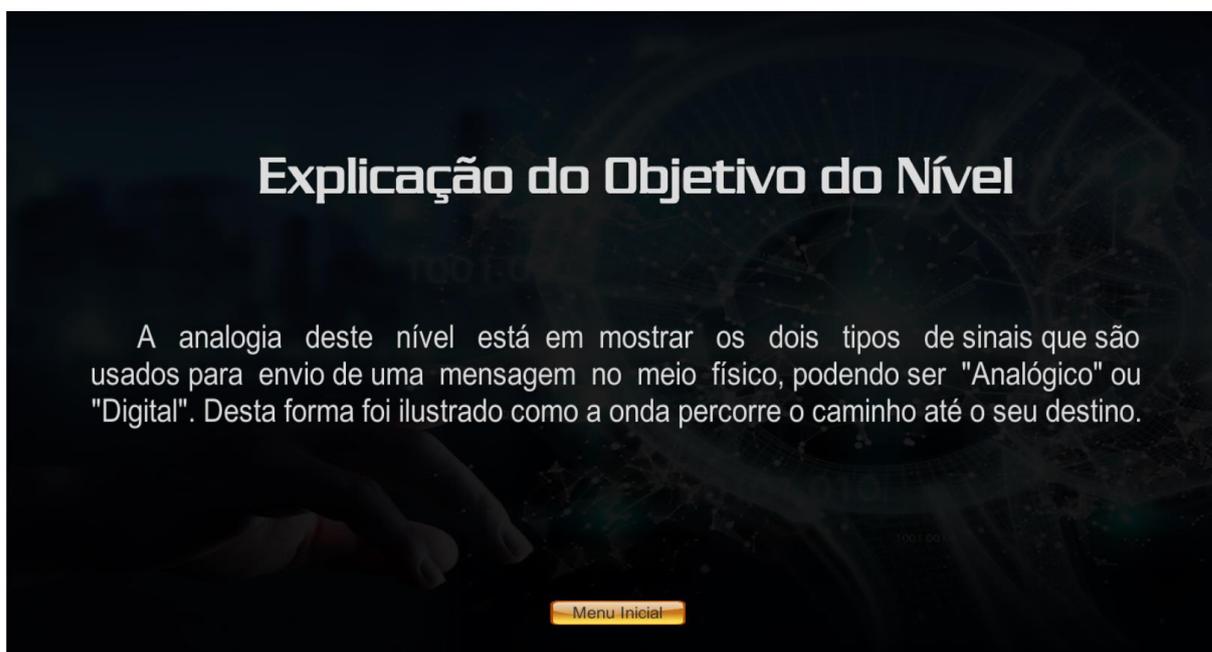
Figura 67 - Entregas de Penny - Final objetivo camada física



Fonte: Do autor.

Concluindo o objetivo, o jogador é redirecionado para outra tela, conforme ilustrada na figura 68, essa informa a analogia do nível com o conteúdo abordado sobre a camada física do protocolo TCP/IP.

Figura 68 - Entregas de Penny - Final camada Física



Fonte: Do autor.

5.6 ENTREGAS DE PENNY – EXPERIÊNCIA

Os principais fatores avaliados na experiência do usuário são os audiovisuais, pois neste ponto é validado o jogo por completo, juntando todos os recursos que construíram ele e sendo avaliados pelo usuário.

Com relação a experiência do jogador de Entregas de Penny, ele vai presenciar um ambiente 2D, pois o jogo foi construído com essa premissa. Todos os conjuntos de objetos usados no cenário e personagem foram selecionados na *internet* com o intuito de fornecer vínculos com a narrativa do jogo.

Os efeitos sonoros e as trilhas do jogo foram realizadas conforme a narrativa do jogo, com detalhamento das ações que ocorrem durante cada nível do jogo. Eles possuem a função de motivar, ou avisar o jogador de ações a serem realizadas no jogo.

Se tratando de um *serious game* a recompensa é a abstração do conteúdo proposto na resolução do nível. Entende-se que se o jogador conseguiu concluir o objetivo do nível estipulado, ele obteve o conhecimento da analogia criada entre o jogo e o assunto abordado em cada nível.

5.7 ENTREGAS DE PENNY – TECNOLOGIA

A tecnologia escolhida para o desenvolvimento de “Entregas de Penny” foi baseada na necessidade da criação de um jogo 2D com elementos que proporcionam o desafio de implementar o objetivo da criação do *serious game*.

Chamadas de *Engine* (plataforma de desenvolvimento para simplificar a produção de jogos eletrônicos), elas proporcionam um ambiente de desenvolvimento desses jogos através de motores gráficos, facilitando o uso das ferramentas e programação.

Portanto a *engine* de desenvolvimento é fator muito importante na criação do jogo, pois ela delimita funcionalidades que podem existir nas ferramentas. Entre as mais conhecidas no mercado estão Unreal Engine, Unity, Godot, Construct, Game Maker, entre outras.

A escolhida para realizar este trabalho foi a Unity, por ser uma ferramenta

gratuita e com os recursos bem aprimorados para execução de jogos 2D. A mesma possui também uma ampla documentação, isso facilita na hora de sanar eventuais dúvidas que possa vir a acontecer.

A Unity tem suporte a linguagem de programação escolhida também, sendo a C# (C Sharp), pela familiaridade do desenvolvedor e por ela ser uma linguagem orientada a objetos.

O jogo foi desenvolvido para rodar nos computadores pessoais (*desktops* ou *notebooks*), como aplicação no sistema operacional. Desta forma o jogador interage diretamente com o jogo através de teclado, mouse e dispositivos de som, gerando o *gameplay*, que é analisado pela *engine* e a mesma retorna o *feedback* para o jogador das ações realizadas.

A versão final do jogo neste trabalho, conta com 285mb de dados, dentre sprites, imagens, objetos, sons, scripts e demais artefatos que constroem o jogo.

O jogo também tem até o momento 21 scripts desenvolvidos em C#, inúmeras funções implementadas, 20 cenas criadas dentre muitos outros objetos que existem nesta versão final criada para este trabalho.

6 CONSIDERAÇÕES FINAIS

Com base no estudo e pesquisa dos conceitos elencados neste trabalho, podemos compreender melhor e de forma mais clara o que é um *serious game* e como ele é construído a partir de um planejamento.

Este trabalho foi projetado em duas etapas, onde na primeira foi modelado e conceituado o que seria necessário para criar um *serious game* através de um planejamento, e na segunda o desenvolvimento dele.

Na primeira etapa foi realizado um estudo mais aprofundado sobre o conteúdo abordado ao decorrer deste trabalho. Foi feita uma investigação sobre como conceber um *serious game* e de que forma ele poderia ser realizado baseado no tema principal que foi abordado por este jogo.

O assunto base para a criação deste jogo foi o da arquitetura TCP/IP em redes de computadores. Pode-se observar que esse assunto é complexo e muito importante para a sociedade nos dias de hoje, pois com o avanço da tecnologia as pessoas estão cada vez mais conectadas e desta forma utilizando mais os conceitos e serviços mediados pela arquitetura TCP/IP.

Sendo assim, inicialmente foi realizado todo um estudo sobre a arquitetura TCP/IP e como seria possível criar o jogo com esse assunto. Foi então planejado e modelado como seria o jogo e qual o público alvo que ele atingiria com base na metodologia implementada para a criação deste jogo. Neste caso foi escolhido o *framework* DPE (*design, play, experience*).

O *framework* é adequado para a modelagem e planejamento de um *serious game*, abrangendo a elaboração do conteúdo. O estudo sobre este *framework* permitiu encontrar a forma apropriada de estruturar o *serious game*.

Neste ponto foi possível relacionar os conceitos que o jogo abordaria com o que seria desenvolvido nele, criando o vínculo do planejamento com o desenvolvimento do jogo.

Para melhorar ainda mais o planejamento deste jogo, neste trabalho foi realizado um estudo de caso do jogo *Netsim*, que ajudou a compreensão de como criar um *serious game*.

A segunda etapa deste trabalho foi o desenvolvimento do jogo. Conforme planejado, o desenvolvimento do jogo foi a etapa principal deste trabalho, pois depois de estudar as mecânicas e assuntos relacionados a *serious game*, foi necessário

colocar os conceitos em prática para validação desta proposta.

O jogo foi implementado na *engine* Unity, com a criação de materiais audiovisuais, cenas, scripts e demais artefatos que compunham este jogo. Muitos artefatos visuais foram retirados da *internet* com a identificação de não necessitar de permissão para uso dos direitos autorais deles.

Todas as atividades deste projeto correspondem responder à questão que direcionou este trabalho: “Criar um jogo educativo para ensino de redes de computadores”.

O desenvolvimento do jogo foi finalizado, porém não pôde ser validado como um *serious game*, pois, para isso, seria necessário ter resultados de teste do jogo. Conforme planejado inicialmente o jogo seria disponibilizado para uma turma de estudantes de redes de computadores da Universidade de Caxias do Sul. Porém, até a finalização deste trabalho não foi possível realizar a validação com um número relevante de usuários.

Não foi possível essa validação, pois o jogo ficou pronto tarde demais, e criar um ambiente de teste do jogo demandaria muito tempo. Com isso foi possível fazer a validação se esse jogo é ou não um *serious game*. Por outro lado, o projeto do *serious game* foi verificado e moldado pela professora responsável por todas as disciplinas de redes computacionais da Universidade de Caxias do Sul.

A professora Maria de Fatima Webber do Prado Lima, validou o conteúdo do jogo, avaliando a jogabilidade do mesmo e moldando alguns detalhes. O parecer da dela é de que o jogo, sim, contempla alguns assuntos que compõem o tema da disciplina, que é a arquitetura TCP/IP em redes de computadores. Sendo assim, para a professora esse jogo pode ser considerado de utilidade na fixação do conteúdo TCP/IP.

Portanto, os objetivos de planejamento e desenvolvimento do jogo foram alcançados neste trabalho, ficando somente a validação se ele é ou não um *serious game* por ser realizada.

É importante ressaltar que as expectativas do desenvolvimento eram altas, e foram atingidas. Realizar a criação dos cinco níveis, implementar os assuntos, criar o jogo, a narrativa e a jogabilidade, foram atividades muito desafiadoras. Desenvolver um jogo não é simples, existem diversos detalhes que são necessários.

O jogo possui muitos aspectos a serem aprimorados, contudo o objetivo principal e o objetivo pessoal de criar um jogo, como trabalho de conclusão foi

cumprido.

6.1 TRABALHOS FUTUROS

Existem diversos pontos que poderiam ser melhorados no jogo, além de detalhes que seriam necessários para a validação deste jogo.

Desta forma segue uma lista com os trabalhos futuros deste desenvolvimento:

- Criar um *playtesting* (ambiente de teste do jogo para diversos grupos de pessoas distintas, sendo este um local único de acesso para todas as pessoas interessadas em jogar o jogo);
- Criar um relatório de validação do *playtesting* (para poder verificar os resultados dos jogadores e experiência deles ao jogar o jogo e elencar se o jogo é ou não um *serious game*);
- Apresentar uma narrativa do jogo mais detalhada (desta forma criando meios de explicar melhor a história do jogo com o jogador);
- Melhorar os objetos audiovisuais do jogo (muitos *assets* poderiam ser melhorados, deixando mais explicativo o jogo);
- Implementar mecânicas novas.

REFERÊNCIAS

ABT, Clark C.. **Serious Games**. 1987. Disponível em: <https://books.google.com.br/books?id=axUs9HA-hF8C&printsec=frontcover&dq=inauthor:%22Clark+C.+Abt%22&hl=pt-BR&sa=X&ved=0ahUKEwietpLqid_iAhUBGLkGHcceAa4Q6AEIKTAA#v=onepage&q&f=false>. Acesso em: 25 abr. 2019.

ADAMS, Ernest. **Fundamentals of Game Design, Third Edition**: 2014. Disponível em: <<http://ptgmedia.pearsoncmg.com/images/9780321929679/samplepages/0321929675.pdf>>. Acesso em: 3 maio 2019.

ALDRICH, Clark. **LEARNING BY DOING: A Comprehensive Guide to Simulations, Computer Games, and Pedagogy in e-Learning and Other Educational Experiences**. 2005. Disponível em: <[http://elearning.fit.hcmup.edu.vn/~longld/References%20for%20TeachingMethod&EduTechnology%20-%20Tai%20lieu%20PPDH%20&%20Cong%20Nghe%20Day%20Hoc/\(Book\)%20-%20Sach%20tham%20khao%20-%20Teaching%20Method/2005%20C.%20Aldrich%20-%20Learning%20by%20Doing-A%20Comprehensive%20Guide.pdf](http://elearning.fit.hcmup.edu.vn/~longld/References%20for%20TeachingMethod&EduTechnology%20-%20Tai%20lieu%20PPDH%20&%20Cong%20Nghe%20Day%20Hoc/(Book)%20-%20Sach%20tham%20khao%20-%20Teaching%20Method/2005%20C.%20Aldrich%20-%20Learning%20by%20Doing-A%20Comprehensive%20Guide.pdf)>. Acesso em: 06 maio 2019.

CANTÚ, Evandro. **Redes de Computadores e a Internet**. 2010. Disponível em: <<https://wiki.sj.ifsc.edu.br/wiki/images/2/2d/ApostilaRedes2010.pdf>>. Acesso em: 5 jun. 2019.

DIAS, Jéssica David; TIBES, Chris Mayara dos Santos; ZEM-MASCARENHAS, Silvia Helena. **USO DE SERIOUS GAMES PARA ENFRENTAMENTO DA OBESIDADE INFANTIL: REVISÃO INTEGRATIVA DA LITERATURA**. 2015. Disponível em: <http://www.scielo.br/pdf/tce/v26n1/pt_1980-265X-tce-26-01-e3010015.pdf>. Acesso em: 18 maio 2019.

FOROUZAN, Behrouz A.; FEGAN, Sophia Chung. **TCP/IP protocol suite**. 2000. Disponível em: <https://cdu-edu-primo.hosted.exlibrisgroup.com/primo-explore/fulldisplay?docid=61CDU_Alma2130410630003446&context=L&vid=61CDU&lang=en_US&search_scope=default_scope&adaptor=Local%20Search%20Engine&isFrbr=true&tab=default_tab&query=isbn,contains,025624166X,AND&sortby=date&facet=frbrgroupid,include,765207105&mode=advanced&offset=0>. Acesso em: 06 maio 2019.

FLEURY, Afonso; NAKANO, Davi; CORDEIRO, José Henrique Dell Osso. **Mapeamento da indústria brasileira e global Jogos Digitais**. 2014 Disponível em: <http://www.abragames.org/uploads/5/6/8/0/56805537/mapeamento_da_industria_brasil_e_global_de_jogos_digitais.pdf>. Acesso em: 3 maio 2019.

FOROUZAN, Behrouz A.; MOSHARRAF, Firouz. **Redes de Computadores uma Abordagem Top-Down**. Porto Alegre Rs: Amgh Editora Ltda, 2013. 834 p. Disponível em:

<<https://books.google.com.br/books?id=57BIAgAAQBAJ&pg=PA20&dq=tcp/ip&hl=pt-BR&ei=lgXrXKupN7rgwQXJ4L6oDQ&cd=5#v=onepage&q=tcp%2Fip&f=true>>.

Acesso em: 26 maio 2019.

FOROUZAN, Behrouz A.; MOSHARRAF, Firouz. **REDES DE COMPUTADORES: UMA ABORDAGEM TOP-DOWN**. 2013. Disponível em:

<<https://integrada.minhabiblioteca.com.br/#/books/9788580551693/cfi/2!/4/4@0.00:0.00>>. Acesso em: 25 abr. 2019.

FREITAS, Sara de; LIAROKAPIS, Fotis. **Serious Games: A New Paradigm for Education?** 2011. Disponível em:

<https://www.researchgate.net/publication/262545606_Serious_Games_A_New_Paradigm_for_Education>. Acesso em: 06 maio 2019.

KUROSE, J. F.; ROSS, K. W. *Redes de computadores e a Internet: uma abordagem top-down*. 3.ed. São Paulo: Pearson Addison Wesley, 2006. ISBN 8588639181

KUROSE, Jim; ROSS, Keith. **Redes de computadores e a internet**. 2013. Disponível em:

<http://www.sj.ifsc.edu.br/~tisemp/Redes%20de%20Computadores%20e%20a%20Internet_Kurose.pdf>. Acesso em: 01 jun. 2019.

MICHAEL, David; CHEN, Sande. **Serious games: Games that Educate, Train and Inform**. 2005. Disponível em:

<https://anagroudeva.files.wordpress.com/2013/06/serious_games__games_that_educate__train__and_inform.pdf>. Acesso em: 29 abr. 2019.

MORAIS, Alana Marques de. **Planejamento e desenvolvimento de um serious game voltado ao ensino de saúde bucal em bebês**. 2011. Disponível em:

<http://de.ufpb.br/~mds/DissertacoesAprovadas/Dissertacao_Alana_Marques_de_Morais-2011.pdf>. Acesso em: 20 maio 2019.

NORMAN, Don. **Things that Make Us Smart**. 1993. Disponível em:

<<https://books.google.com.br/books?id=yPKkBQAAQBAJ&printsec=frontcover&dq=Things+that+Make+Us+Smart&hl=pt-BR&sa=X&ved=0ahUKEwjcnYrmloDjAhVuK7kGH8Oa8OAEIMDAB#v=onepage&q=Things%20that%20Make%20Us%20Smart&f=false>>.

Acesso em: 21 maio 2019.

PRENSKY, Marc. **Digital Natives, Digital Immigrants**. 2001. Disponível em:

<<https://www.marcprensky.com/writing/Prensky%20-%20Digital%20Natives,%20Digital%20Immigrants%20-%20Part1.pdf>>. Acesso em:

20 maio 2019.

SAVI, Rafael; ULBRICHT, Vania Ribas. **JOGOS DIGITAIS EDUCACIONAIS: BENEFÍCIOS E DESAFIOS.** 2008. Disponível em: <<https://seer.ufrgs.br/renote/article/view/14405/8310>>. Acesso em: 06 maio 2019.

SEVERGNINI, Luis Filipe. **Serious game como ferramenta de ensino de lógica de programação para crian_cas.** 2016. Disponível em: <<https://repositorio.ucs.br/xmlui/bitstream/handle/11338/1601/TCC%20Luis%20Felipi%20Severgnini.pdf?sequence=1&isAllowed=y>>. Acesso em: 3 maio 2019.

SILVEIRA, Sidnei Renato et al. **Aprendizagem Baseada em Problemas: possibilidades para o ensino de programação de computadores.** 2018. Disponível em: <<http://www.br-ie.org/pub/index.php/wcbie/article/view/8333/6008>>. Acesso em: 14 jun. 2019.

STAPLETON, Andrew J.; TAYLOR, Peter C.. **Why Videogames are Cool & School Sucks!** 2003. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.323.4909&rep=rep1&type=pdf>>. Acesso em: 05 maio 2019.

TANENBAUM, Andrew S.. **Redes de Computadores.** 2011. Disponível em: <http://www.vazzi.com.br/arquivos_moodle/Redes%20de%20Computadores%20-%20Tanenbaum.pdf>. Acesso em: 01 maio 2019.

THOMPSON, Debbe. **Serious Video Games for Health: How Behavioral Science Guided the Development of a Serious Video Game.** 2008. Disponível em: <<https://journals.sagepub.com/doi/pdf/10.1177/1046878108328087>>. Acesso em: 24 maio 2019.

TRYBUS, Jessica. **Game-Based Learning: What it is, Why it Works, and Where it's Going.** 2010. Disponível em: <https://docs.google.com/document/d/1zLs8vElkTlftxy8t-TRyG_-hEyDVOj2FTu4GU6g-i2g/edit>. Acesso em: 05 maio 2019.

UNITY3D. Unity - Game Engine. 2015. Disponível em: <<https://unity3d.com/pt>>. Acesso em: 2 nov. 2019.

MANIK, Rajali Ginting. **Modulação.** 2017. Disponível em: <https://jfmm19rcom.files.wordpress.com/2012/09/modulacao_72dpi.gif>. Acesso em: 14 jun. 2019.

WINN, Brian M.. **The Design, Play, and Experience Framework.** 2015. Disponível em: <<http://ksuweb.kennesaw.edu/~jprest20/cgdd4303/winn-dpe-chapter.pdf>>. Acesso em: 11 jun. 2019.

WINN, Brian; HEETER, Carrie. **Resolving Conflicts in Educational Game Design Through Playtesting.** 2006. Disponível em: <<https://core.ac.uk/download/pdf/51073533.pdf>>. Acesso em: 11 jun. 2019.

ZIMMERMAN, Eric; SALEN, Katie. **Rules of Play: Game Design Fundamentals**. Massachusetts. 2004. Disponível em: <<https://gamifique.files.wordpress.com/2011/11/1-rules-of-play-game-design-fundamentals.pdf>>. Acesso em: 06 jun. 2019.