

Revista Hackerspace

Vol. 3

Projeto Hackerspace
2021

Apresentação

O Projeto Hackerspace é um projeto de extensão promovido pela UFSCar Campus Sorocaba e realizado na forma de encontros que discutem a contracultura hacker, apresentando personagens, acontecimentos, aspectos socioculturais, artefatos e atividades com a finalidade de que os participantes sejam expostos à contracultura em questão e desenvolvam um projeto ou material acerca da mesma.

A principal intenção do projeto é desmistificar o conceito e figura do hacker, enquanto proporcionando um espaço de exposição, produção e compartilhamento de conteúdo que se configura como dentro da contracultura de forma que participantes de diferentes níveis de conhecimento técnico possam participar.

Esta revista tem como objetivo divulgar os trabalhos desenvolvidos pelos alunos participantes do Projeto Hackerspace no ano de 2021. Esperamos que esta revista possibilite a difusão dos conhecimentos adquiridos na atividade para a comunidade em geral. Gostaríamos de agradecer todos os alunos que contribuíram com artigos e projetos para esta edição da revista.

Organização e edição:

Bruno Frítoli Carraza

Caio César Brandini

Gustavo de Jesus Rodrigues

Marcus Vinicius Natrielli Garcia

Supervisão:

Gustavo M. D. Vieira

Conteúdo

Conteúdo	iii
I Artigos	1
1 Concepção de Jammer de pulso eletromagnético em dardos ou flechas III - Início dos testes	2
LUCAS MARTINS SILVA lucas.silva@dcomp.sor.ufscar.br	
2 Concepção de Jammer de pulso eletromagnético em dardos ou flechas IV - Versão Alfa	8
LUCAS MARTINS SILVA lucas.silva@dcomp.sor.ufscar.br	
3 Cibersegurança: Histórico, aspectos e tipos de Ciberataques	14
MATHEUS FERNANDO VIEIRA PINTO	
4 Web Scraping com Python - Estudo de caso	20
ENZO L. FERNANDES FRANCIENE BERNARDI	
5 NAT Punchthrough: Multiplayer, P2P e adversidades	23
ANDERSON PINHEIRO GARROTE DANILO ISAMU INAFUKU	
6 O hacking social por trás de jogos "Gacha"	27

AGATA RANGEL DRIGO
agata.drigo@estudante.ufscar.br

FELIPE BERTONI SALVATI
felipesalvati@estudante.ufscar.br

FERNANDO FAVARETO ABROMOVICK
fernandofa@estudante.ufscar.br

GIOVANNI MARÇON ROSSI
giovanni.rossi@estudante.ufscar.br

MARIANE LAMAS MALHEIROS
mariane.malheiros@estudante.ufscar.br

7	Introdução a Hash e seus usos	35
	GABRIEL GOULART HOMEM	
8	Um guia introdutório extensivo, porém não exaustivo sobre Hardware e Software relativamente modernos - parte I: hardware	41
	THIAGO DOMINGUES DA SILVA	
II	Projetos	72
9	ADBlock a nível de DNS	73
	VINÍCIUS CARVALHO VENTURINI	
10	Timeckoon	75
	RYAN SAKURAI	
11	Projeto TFMD - automação de cálculo de orçamentos	78
	JOATAN MARQUES	
12	Dúvidas IP	80
	MAURÍCIO CÂNDIDO DE SOUZA MICHEL RIBEIRO KOBÁ	
13	Editor de texto para Dragon Quest II (Nintendo Switch)	82
	VÍTOR RIBEIRO	
14	Wiki de Bravelly Default	84
	VÍTOR RIBEIRO	

Parte I
Artigos

Capítulo 1

Concepção de Jammer de pulso eletromagnético em dardos ou flechas III - Início dos testes

LUCAS MARTINS SILVA
lucas.silva@dcomp.sor.ufscar.br

Resumo

Este trabalho apresenta a continuidade do desenvolvimento da pesquisa com relação à possível montagem de um *jammer* de pulso eletromagnético sobre uma flecha ou dardo, dando maior foco aos testes realizados e informações obtidas, bem como mudanças realizadas com relação aos projetos anteriores. Será evidenciado o prosseguimento das atividades referentes ao projeto desde o último marco. Neste trabalho também poderá ser encontrada a revisão de direcionamentos de projeto, possivelmente úteis a quaisquer futuros interessados em desenvolvimentos correlatos.

Palavras-chave: flecha; arquearia; *jamming*; pulso eletromagnético; PEM.

1.1 Introdução

Dando continuidade aos trabalhos predecessores dentro do mesmo projeto, este texto tem por finalidade apresentar e descrever os acontecimentos correlatos ao desenvolvimento de um *jammer* de pulso eletromagnético acoplado a uma flecha, referenciando o período composto pelos primeiros seis meses do ano de 2021. Para tal,

serão ressaltadas as técnicas utilizadas e áreas de interesse prioritizadas, bem como adições teóricas nos ramos de arquearia e física, além da descrição dos primeiros testes realizados e seus resultados.

Sendo este o terceiro documento referente ao projeto, espera-se do leitor a leitura das primeiras publicações, visto que parte dos conceitos será retomada, enquanto outra parte será presumida conhecida para fins de fluidez textual e redução da carga de leitura.

É necessário retomar no entanto, que, em essência, um *jammer* de pulso eletromagnético é um dispositivo simples, composto geralmente por uma espira condutora por onde trafegará uma corrente elétrica criada por uma diferença de potencial. A passagem da corrente é acompanhada da formação de um campo eletromagnético nos arredores da área do condutor [4], sendo este campo responsável por interferir no funcionamento de dispositivos eletrônicos nesta área. O efeito é notavelmente relevante para dispositivos computacionais comerciais, visto que tais dispositivos são geralmente compostos por semicondutores de óxido metálico, altamente sensíveis a perturbações de alta tensão [5].

Ademais, é necessário ressaltar que há vários tipos de *jammers*, sendo o escopo deste projeto voltado a um *jammer* baseado no pulso eletro-

magnético gerado por uma corrente em um condutor. É importante mencionar também, que a utilização de disruptores ou *jammers* em território brasileiro é regulamentada por lei e sua utilização desregulada é definida como crime [1]. Desta forma, o desenvolvimento desta pesquisa tem suas bases em propostas de análise científica, bem como do aproveitamento das já mencionadas características do armamento a ser modificado para fins úteis à sociedade, como desarme de dispositivos perigosos, munidos de componente eletrônico, sem a necessidade de aproximação de um operador, ou mesmo desativação de drones em regiões onde não são permitidos (como penitenciárias e conglomerados semelhantes).

Por conseguinte, são evidenciados, neste documento, recursos adicionais para a continuação do projeto de formulação de uma flecha portadora de um *jammer* eletromagnético, além da apresentação de resultados obtidos através de testes realizados. Espera-se também consolidar conceitos aplicáveis a outros tipos de flechas portadoras.

1.2 Adições teóricas e alterações

Seguindo o princípio adotado ao início deste estudo, novas pesquisas foram realizadas em campos da arquearia e da física, a fim de consolidar a viabilidade do projeto e alicerçar de maneira coesa, através do que se conhece da ciência referente às nuances do projeto, as concepções de design a serem adotadas para esta flecha portadora, bem como para trabalhos futuros de quaisquer interessados. Além disso, dada a disponibilidade de parte dos equipamentos cogitados, testes puderam ser realizados e o impacto de seus resultados nas considerações teóricas inicialmente feitas serão também avaliados.

Será retomado, inicialmente, o fenômeno conhecido como o “paradoxo do arqueiro”, referente ao surpreendente movimento coeso de uma flecha disparada por um arco, somente possível dada a flexibilidade presente no corpo da flecha [6]. Este fenômeno é fortemente considerado para fins de precisão, sendo adequado ao uso de cada indivíduo através de testes e diferentes combinações de flechas e arcos. Vale lembrar que

este fenômeno está intrinsecamente ligado aos conceitos de espinha de flecha (medida de sua rigidez e flexibilidade [2]), além da angulação entre o centro esperado de tiro (presente na mesma direção da corda do arco) e o posicionamento real da flecha no momento da puxada.

Tendo isso em vista, é importante mencionar que, inesperadamente, o arco obtido para treino e teste possui característica bem próxima à de “tiro central”, o que significa que a flecha em descanso se posiciona, em quase totalidade, no plano configurado pela corda em relação às lâminas do arco. Tal característica resulta em um caminho de flecha retilíneo conforme ocorre a soltura, o que neutraliza parte do paradoxo do arqueiro e possibilita maior flexibilidade na escolha de espinhas de flecha.

Dentro desta característica, é necessário ressaltar que o bônus da flexibilização de escolhas de espinha de flecha é acompanhado de um interessante ônus: segundo discussões dentro do ramo da arquearia, o posicionamento de tiro central e a ausência da presença forte do paradoxo do arqueiro tendem a ressaltar pequenas perturbações na corda oriundas da utilização de um arco sem a ferramenta de soltura por gatilho. Na puxada, o punho impõe sobre a corda torção lateral que varia conforme fatores como a técnica utilizada, costume com o peso do arco e cansaço do atirador. Esta tensão é capaz de alterar o fluxo do disparo, o que requisita ainda mais perícia na soltura por parte do atirador, sendo defendida por alguns arqueiros a posição de que é impossível eliminar completamente as perturbações causadas na puxada com os dedos [7].

O fenômeno acima descrito foi notado exaustivamente durante os meses referenciados neste documento, sendo inicialmente cogitada a utilização incorreta de espinhas de flecha para o arco em mãos. As perturbações de voo foram atenuadas conforme a progressão de técnica do atirador e posterior entendimento do mecanismo envolvido.

1.3 O problema do *jammer* em flecha

Conforme descrito nos trabalhos predecessores, o problema a que se propõe este projeto é o do desenvolvimento de uma flecha portadora de um *jammer* de pulso eletromagnético. A maior parte das considerações acerca da formulação do problema permanecem inalteradas, e os testes realizados objetivaram simular o ambiente especificado na formulação do problema. Tal ambiente pode ser sumarizado como a utilização de uma flecha portadora (com o sistema anteriormente mencionado) com alcance efetivo dentro da distância de 10 metros.

1.4 Treinamento, testes e progressões

Para esta etapa do projeto foram obtidos equipamentos básicos a fim de possibilitar o início de testes capazes de auxiliar o desenvolvimento da flecha portadora especificada. Dessa forma, nesta seção estarão descritos os materiais e procedimentos de teste utilizados nesta etapa das atividades de desenvolvimento. Ademais, será também descrita a preparação realizada para utilização devida do equipamento de arquearia e obtenção de resultados concisos na execução dos testes.

Inicialmente, é necessário considerar a preparação para realização de testes concisos. Como o processo relacionado à arquearia requisita a utilização eficiente do aparato, ocorreu intenso foco no treinamento de corpo e técnica do atirador a ser utilizado como ator nos disparos e testes, sendo este também o autor da pesquisa e do desenvolvimento do projeto e dos protótipos. Com esse intuito, o treinamento foi distribuído em etapas.

A primeira etapa foi referente à adaptação ao uso do arco e seu peso. Como mencionado no trabalho anterior, foi providenciado um arco recurvo de 45 libras de peso de puxada, o que se converte para algo em torno de 20 quilogramas, sem a presença de *let-off*. Dessa forma, as primeiras semanas foram compostas por treinos curtos e intensos iniciando em 6 disparos ao dia de treino e progredindo semanalmente.

Em conjunto, foi mantido treino calistênico já previamente realizado, para auxiliar no processo de desenvolvimento da musculatura necessária ao uso consistente do aparato.

A segunda etapa se deu através de um aumento expressivo do número de disparos em curto intervalo de tempo, distribuindo os disparos em 2 a 3 sessões semanais e sendo almejada a marca de 30 disparos em sequência dentro de uma mesma sessão. Esta marca é considerada por alguns caçadores como uma medida eficiente de avaliação de costume e eficiência de um arqueiro com seu equipamento [3]. Ao atingir a marca de 25 disparos, preferiu-se adiantar a terceira etapa de treinamento.

Durante a terceira e final etapa do treinamento (que ainda perdura), o foco foi dado à realização eficiente de disparos e ao aprimoramento da técnica, com grande ênfase no aperfeiçoamento da soltura de flecha, ainda grande fator de interferência na precisão dos disparos. Durante esta fase, busca-se a realização de um número médio de disparos (a saber, por volta de 7 a 10), capazes de aproveitar a maior adequação ao equipamento para imprimir profundo foco no desenvolvimento da técnica. Além disso, esta fase do treino é caracterizada por grande frequência de sessões de treino, que são realizadas quase diariamente.

Ainda sobre os equipamentos utilizados nos treinos e testes, foram utilizados, além do já mencionado arco recurvo de 45 libras, conjuntos de flechas de alumínio e fibra de carbono, um alvo de material adaptado ao recebimento de disparos de flecha e luvas de puxada para proteção dos dedos em contato com a corda. Foi preferido o uso de flechas de fibra de carbono, dada sua precisão e durabilidade, e este tipo de munição é o utilizado atualmente no prosseguimento do projeto. Com o intuito de garantir precisão nos testes realizados, foi priorizado o aperfeiçoamento da técnica tendo em vista garantir a repetibilidade dos disparos realizados. Em outras palavras, foi objetivada a capacitação com relação a execução de tiros de precisão semelhante, para que as inserções e modificações nas flechas não fossem, durante os testes, ofuscadas pela imperícia do

usuário. Relativo progresso foi feito nesta direção.

A fim de simular as modificações previstas para as flechas, foram acoplados, por meio de adesivos removíveis, corpos de características semelhantes ao conversor de tensão a ser utilizado no sistema a ser portado pela flecha. Além disso, foram observados os desdobramentos de se inserir partes adicionais no projétil em posições que descansem em contato com o arco no momento da puxada, sendo estas regiões mais propícias à destruição advinda do atrito decorrente da rápida propulsão da seta pela corda.

Como alguns resultados, pôde-se reafirmar a necessidade de um bom balanceamento de peso ao longo do corpo das flechas, bem como a disposição adequada dos aparatos de forma a evitar danos adicionais advindos do já mencionado atrito.

Os testes e progressões realizados culminaram com as alterações e considerações de design apontados na sessão seguinte.

1.5 Resoluções e design

Durante a realização dos testes anteriormente mencionados, algumas relevantes informações foram levantadas, gerando alterações e reafirmações no design previamente cogitado à flecha portadora de *jammer* de pulso eletromagnético. Esta seção se dedica a explicitar tais informações, mudanças e conformidades com relação ao design prévio.

Os testes de disparo revelaram a necessidade de dispor elementos mais sensíveis no corpo da seta de forma a deixar livre de acoplamentos um caminho que percore por todo o comprimento do projétil. Esta disposição visa reduzir os danos aos incrementos além de garantir maior precisão nos disparos. Considerando este tópico, elementos incapazes de serem isolados em certas regiões da flecha receberam deliberações e se encontram sob foco de desenvolvimento de possíveis estratégias de proteção (como invólucros de materiais recicláveis).

Outra informação advinda de experimentação e passível de menção se encontra na análise da capacidade de penetração das flechas utilizando pontas de treino. Através da utilização do alvo destinado à arquearia, foram calculadas as médias de mínima e máxima penetração conforme os treinamentos e testes se deram. Observou-se, desta forma, que a penetração mínima das flechas gira em torno de 4cm para o alvo utilizado, enquanto a máxima gira em torno de 16cm. Dessa forma, buscou-se aprofundar a ideia de utilizar o impacto da seta no alvo como gatilho para o fechamento de circuito gerador de pulso eletromagnético.

A fim de utilizar as peculiaridades da penetração do projétil no alvo, foi elaborada a ideia de utilização de um *slider* no corpo da seta. Este *slider* seria responsável por, através do impacto e penetração de parte do corpo da flecha no alvo, deslocar uma das extremidades do fio condutor para contato com a extremidade presente em uma parte anterior do corpo, fechando o circuito elétrico e possibilitando a passagem da corrente capaz de gerar o campo eletromagnético.

Tendo em vista a elaboração deste sistema de deslizamento, foram providenciados rolamentos lineares de 10 milímetros de espaçamento interno, visto que o corpo das flechas utilizadas possui espessura de 8 milímetros. Desta forma, o design até o momento concebido e para o qual se rumo prototipar pode ser apresentado como se segue:

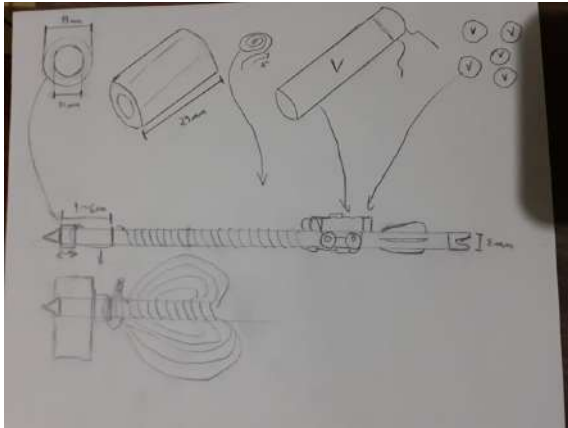


Figura 1.1: Representação do design atual da flecha portadora

Na imagem é possível notar o rolamento linear sendo utilizado como mecanismo de gatilho, levando uma das pontas do fio condutor até outra. Além disso, o conversor de tensão e as baterias podem ser dispostas na parte anterior da seta (como denotada na imagem) ou distribuídas de forma equivalente sobre o corpo do projétil. Tais deliberações, bem como a definição do sentido da corrente utilizado com o circuito, serão foco da continuidade do projeto, após a realização dos primeiros testes de viabilidade do design formulado.

1.6 Estado atual e continuidade

Ainda dentro do período de pandemia global, é válido ressaltar que a disponibilidade de artefatos necessários à constituição do projeto se mostra defasada, especialmente no que tange a composição elétrica do dispositivo. No entanto, é possível ressaltar que a continuidade geral do projeto não foi efetivamente depreciada.

Através do reformulado design, a prototipagem e novos testes são foco das próximas etapas, bem como a melhoria na técnica de arquearia do atirador envolvido. A pesquisa e o desenvolvimento da flecha portadora persistem e seus resultados serão ainda apresentados em futuros projetos.

1.7 Indicações a futuros projetos

Reafirmando o intuito do projeto de estimular formulações e aplicações reais referentes aos tópicos de flechas portadoras, nesta seção são descritas algumas recomendações àqueles interessados em formular projéteis como o deste trabalho ou de demais variedades.

Como observado nas seções anteriores, a capacidade de realizar disparos repetíveis e concisos é fundamental para o bom desenvolvimento de flechas portadoras. Tal restrição deve ser aplicada de modo a isolar os distúrbios de voo dos projéteis advindos da falta de técnica dos advindos das alterações de aerodinâmica realizadas durante as incorporações de dispositivos.

Para esta finalidade, pode ser eficiente utilizar um sistema de treinamento em etapas semelhante ao realizado neste projeto, de forma a desenvolver tanto força quanto resistência muscular, em caso de utilização de um arco de peso de puxada considerável. O autor deste trabalho recomenda a utilização de arcos de bom peso de puxada, a fim de garantir o desempenho dentro da distância esperada, ainda que esta não seja uma característica restritiva ao projeto.

Por fim, a concepção de gatilhos através de um aparato deslizante no corpo da flecha parece capaz de solucionar boa parte dos problemas de ativação dos dispositivos de flechas portadoras e acredita-se ser um grande avanço não apenas para o dispositivo portado descrito neste projeto, mas de muitos outros que se valham de ativações próximas ao ponto de impacto ou mesmo necessariamente distantes do atirador.

1.8 Conclusões

A criação de uma flecha portadora aliada a um gerador de pulso eletromagnético permanece factível e cada vez mais palpável através dos testes e concepções realizadas até o momento neste trabalho. As novas adições teóricas constituem ferramental precioso na formulação do design do projeto, enquanto os resultados de testes obtidos apontam necessidades e riscos a serem

considerados no prosseguimento do projeto.

O processo de desenvolvimento persiste, tendo em vista a miríade de aspectos revelados a cada etapa de pesquisa, teste e adequação. Consideram-se satisfatórios os desdobramentos e observações do período de seis meses evidenciado por este documento. Desta forma, conclui-se esta etapa do trabalho, tendo em vista sua continuidade e aperfeiçoamento.

1.9 Bibliografia

- [1] ANATEL. Resolução nº 308, de 11 de setembro de 2002. URL: <https://www.anatel.gov.br/legislacao/resolucoes/2002/257-resolucao-308>.
- [2] Easton Archery. Making sense of arrow spine. URL: <https://eastonarchery.com/2014/07/making-sense-of-arrow-spine/>.
- [3] BowHunting. How much draw weight for hunting? URL: <https://www.bowhunting.com/bowhunt101/much-draw-weight-hunting>.
- [4] Young e Freedman. *FISICA III - 12ª edição*.
- [5] C.N. Ghosh. *EMP weapons, Strategic Analysis*. 2000.
- [6] Real World Physics Problems. Physics of archery – archer’s paradox. URL: <https://www.real-world-physics-problems.com/physics-of-archery.html>.
- [7] Archery Talk. What does center shot mean? URL: <https://www.archerytalk.com/threads/what-does-center-shot-mean.1066481/>.

Capítulo 2

Concepção de Jammer de pulso eletromagnético em dardos ou flechas IV - Versão Alfa

LUCAS MARTINS SILVA
lucas.silva@dcomp.sor.ufscar.br

Resumo

Este trabalho apresenta a continuidade do desenvolvimento da pesquisa com relação à possível montagem de um *jammer* de pulso eletromagnético sobre uma flecha ou dardo, relatando principalmente o surgimento de uma primeira versão do sistema, bem como mudanças realizadas com relação aos projetos anteriores. Será evidenciado o prosseguimento das atividades referentes ao projeto desde o último marco. Neste trabalho também poderá ser encontrada a revisão de direcionamentos de projeto, possivelmente úteis a quaisquer futuros interessados em desenvolvimentos correlatos.

Palavras-chave: flecha; arquearia; *jamming*; pulso eletromagnético; PEM.

2.1 Introdução

Dando continuidade aos trabalhos predecessores dentro do mesmo projeto, este texto tem por finalidade apresentar e descrever os acontecimentos correlatos ao desenvolvimento de um *jammer* de pulso eletromagnético acoplado a uma flecha, referenciando o período composto pelos

últimos seis meses do ano de 2021. Para tal, serão ressaltadas as técnicas utilizadas e áreas de interesse prioritizadas, bem como adições teóricas nos ramos de arquearia e física, além da descrição dos testes realizados através da montagem de versão inicial do aparato referente a esta série de trabalhos.

Sendo este o quarto documento referente ao projeto, espera-se do leitor a leitura das primeiras publicações, visto que parte dos conceitos será retomada, enquanto outra parte será presumida conhecida para fins de fluidez textual e redução da carga de leitura.

É necessário retomar, no entanto, que, em essência, um *jammer* de pulso eletromagnético é um dispositivo simples, composto geralmente por uma espira condutora por onde trafega uma corrente elétrica criada por uma diferença de potencial. A passagem da corrente é acompanhada da formação de um campo eletromagnético nos arredores da área do condutor [4], sendo este campo responsável por interferir no funcionamento de dispositivos eletrônicos nesta área.

O efeito é notavelmente relevante para dispositivos computacionais comerciais, visto que tais dispositivos são geralmente compostos por semicondutores de óxido metálico, altamente sensíveis a perturbações de alta tensão [6].

Ademais, é necessário ressaltar que há vários tipos de *jammers*, sendo o escopo deste projeto voltado a um *jammer* baseado no pulso eletromagnético gerado por uma corrente em um condutor. É importante mencionar também, que a utilização de disruptores ou *jammers* em território brasileiro é regulamentada por lei e sua utilização desregulada é definida como crime [1].

Desta forma, o desenvolvimento desta pesquisa tem suas bases em propostas de análise científica, bem como do aproveitamento das já mencionadas características do armamento a ser modificado para fins úteis à sociedade, como desarme de dispositivos perigosos, munidos de componente eletrônico, sem a necessidade de aproximação de um operador, ou mesmo desativação de drones em regiões onde não são permitidos (como penitenciárias e conglomerados semelhantes).

Por conseguinte, são evidenciados, neste documento, recursos adicionais para a continuação do projeto de formulação de uma flecha portadora de um *jammer* eletromagnético, além da apresentação dos resultados obtidos após a montagem de uma versão inicial do sistema, a ser chamada versão alfa. É esperada a conclusão do projeto dentro de alguns meses, a ser evidenciada também em um quinto artigo de registro. Espera-se também consolidar conceitos aplicáveis a outros tipos de flechas portadoras.

2.2 O problema do *jammer* em flecha

Conforme descrito nos trabalhos predecessores, o problema a que se propõe este projeto é o do desenvolvimento de uma flecha portadora de um *jammer* de pulso eletromagnético. A maior parte das considerações acerca da formulação do problema permanecem inalteradas, e os testes realizados objetivaram simular o ambiente especificado na formulação do problema. Tal ambiente pode ser sumarizado como a utilização de uma flecha portadora (com o sistema anteriormente mencionado) com alcance efetivo dentro da distância de 10 metros.

2.3 Adições teóricas e alterações

Para o desenvolvimento e teste de uma primeira versão do projeto referenciado por esta série de trabalhos, foi realizada uma revisão da pesquisa acerca de materiais disponíveis e sua possível utilização. Dentro desta revisão, foram realizadas mudanças no design previamente cogitado, a serem abordadas em seções subsequentes.

Dentre as revisões iniciais, é possível observar a mudança das lâminas do arco utilizado, sendo realizado um leve acréscimo no peso de puxada e, conseqüentemente, na energia transmitida à flecha pelo arco. O equipamento, que previamente contava com 45 libras de peso de puxada, teve seus membros substituídos por membros de 50 libras, realizando um acréscimo de aproximadamente 2 quilogramas em peso de puxada.

Ainda dentro do âmbito de materiais, foram adquiridos novos componentes para a montagem e teste de uma versão básica, como rolamentos lineares responsáveis por simular a função de gatilho do sistema, fios elétricos de 0,1mm de diâmetro, pilhas de 1,5V e um conversor de tensão de 6V para até 40kV. As demais peças utilizadas foram adesivos básicos e materiais recicláveis reutilizados.

No âmbito teórico, é importante ressaltar as características do campo de eletromagnetismo relevantes ao prosseguimento do trabalho, bem como pequenas novas nuances de arquearia observadas ao longo do desenvolvimento. Serão explicitadas algumas destas características que se apresentaram como de grande relevância às deliberações efetuadas com relação ao design e teste do dispositivo gerado.

Inicialmente, é necessário ressaltar que, a partir do aprofundamento dos estudos acerca do campo de eletromagnetismo, pôde-se notar problemas conceituais a serem resolvidos para que se consiga um campo eletromagnético de intensidade relevante. Buscando um sistema capaz de gerar o campo eletromagnético mais intenso nos arredores do impacto da flecha, foram cogitados os designs de circuito com espira, bobina plana e solenoide.

No caso do campo magnético gerado pela passagem da corrente em uma espira, é relevante para o projeto mencionar que há a geração de campo magnético no interior e no exterior da espira [5], o que a tornaria uma candidata plausível para a utilização no projeto.

No caso do campo magnético gerado pela passagem da corrente em uma bobina plana, é relevante ao projeto mencionar que suas características são semelhantes às do campo gerado pela passagem da corrente em uma espira singular, mas com a possibilidade de acréscimo da intensidade do campo magnético conforme mais voltas são adicionadas na bobina (desde que não alcancem comprimento relativo ao raio da espira suficiente para que seja considerada um solenoide) [5]. Esta opção também se mostra como viável, e considerada até o momento como possivelmente mais eficiente para os objetivos do projeto.

Já no caso do campo magnético gerado pela passagem da corrente em um solenoide, é válido referenciar, tendo em vista os objetivos do projeto, que o campo magnético gerado no exterior do solenoide tende a ser nulo quanto mais definido for o solenoide, sendo gerado neste caso um campo magnético de intensidade relevante apenas no interior do solenoide [5]. Como o objetivo inicial do projeto é verificar a possibilidade de causar perturbações magnéticas nos arredores do impacto da flecha, esta opção será deixada como secundária e pouco viável.

Ademais, é importante ressaltar que a intensidade do campo magnético gerado por passagem de corrente elétrica em condutor é sempre dependente da intensidade da corrente elétrica, que, por sua vez, quando a tensão elétrica se mantém constante, é inversamente proporcional à resistência elétrica do condutor.

Além disso, também é necessário ressaltar que a área de seção transversal (ou o diâmetro) do fio condutor é também inversamente proporcional à resistência elétrica, o que indica que fios de maior diâmetro são mais eficientes para a passagem de corrente de maior intensidade [5].

Dentro do campo de arquearia, é válido mencionar que a deliberação acerca do peso total da flecha mostrou-se relevante ao projeto dado o peso do equipamento a ser acoplado. Desta forma, foram buscadas evidências e comparações referentes ao impacto do peso excessivo de flecha em seu desempenho. Foram encontradas nesta pesquisa, evidências referentes ao aumento da penetração causada por flechas de grande peso, o que gerou novas preocupações com relação ao design da flecha.

Por fim, é importante retomar os conceitos de espinha de flecha e *Front of Center* (FOC), sendo o primeiro um parâmetro referente à rigidez do corpo da seta [2] e o segundo um parâmetro referente à distribuição do peso ao longo do corpo do projétil [3].

2.4 Resoluções e design

Após os estudos realizados e verificação dos equipamentos obtidos, pôde-se deliberar acerca da eficiência do que foi obtido e realizar a montagem de uma versão inicial a fim de obter maiores informações acerca de possíveis empecilhos ao prosseguimento do projeto de maneira eficiente.

Desta forma, inicialmente foi mantido o design previamente cogitado, com a alimentação posicionada na parte anterior da seta, o conversor próximo ao centro, seguido da espira e do rolamento linear, ambos próximos da ponta da flecha. A fim de identificar possíveis problemas com relação ao peso e FOC do aparato, foram realizados alguns cálculos e registros.

O primeiro deles, foi realizar medições acerca do peso da flecha base a ser utilizada e de todos os adereços aos quais se busca empregar no projeto. Utilizando uma balança de precisão, foram obtidos os seguintes resultados:

	Flecha	Rolamento Linear	Conversor de Tensão	Pilhas	Total
Peso em gramas	36g	24g	42g	4x11g (44g)	146g
Peso em grãos	555gr	370gr	648gr	4x169gr (679gr)	2253gr

Após a observação destes resultados, foi realizada uma comparação com pesos conhecidos previamente de flechas utilizadas para a caça, onde o potencial de penetração é de grande valia. Realizando as conversões diretas, foi possível observar que o peso total para o aparato está estimado em pouco mais de 146 gramas ou 2253 grãos (fios não inclusos no peso para estas medidas).

Este peso pode ser considerado extremo, dado que mesmo flechas utilizadas para caça com objetivo de perfurar ossos são consideradas de peso elevado quando se encontram no intervalo de 650gr(42g) a 950gr(61g) [7]. Por conseguinte, foi necessário destinar maior atenção a um mecanismo de parada para o rolamento linear, que corre o risco de sofrer maior propulsão no impacto dada a maior penetração no alvo, o que poderia danificar a fiação elétrica próxima à ponta da seta.

Ainda considerando o peso da flecha e tendo em foco o parâmetro FOC, foram realizadas medições e cálculos à medida que foram acopladas ao projétil os aparatos cuja posição são consideradas fixas para o design do projeto (até o momento): as pilhas/baterias para alimentação na parte anterior, o rolamento linear na parte posterior e o conversor de tensão próximo ao centro. Este processo foi realizado de maneira iterativa, sendo os resultados para FOC referenciados como se segue:

	Flecha limpa	Flecha com Rolamento Linear	Flecha com Rolamento Linear e Pilhas	Flecha com Rolamento, Pilhas e Conversor
FOC	9,35%	24,83%	4,19%	6,77%

A montagem dos componentes na flecha se deu cronologicamente como se observa da esquerda para a direita na tabela, sendo os valores de FOC obtidos a cada iteração utilizados para se determinar a posição do Conversor de Tensão no corpo da seta, visto que este componente representaria o maior acréscimo unitário de peso, alterando em maior escala a proporção e o FOC. Objetivou-se com sucesso manter um FOC baixo e próximo ao da flecha sem adereços, sendo mantido para a versão alfa em 6,77% e em deliberação para

possível acréscimo.

Para a versão alfa, não foi adicionada a espira e não foram realizadas as ligações elétricas, visto que a falta de conhecimento acerca do perfil de voo da flecha com a montagem de componentes requisitou maior cuidado para que os componentes não fossem danificados nos primeiros testes.

No entanto, já foi realizada ligação de teste para as pilhas de alimentação, que foram interligadas em série para que a tensão nominal resultante dos polos do sistema de alimentação pudesse atingir o marco de 6 volts, requisitado para realizar a alimentação do conversor de tensão.

Desta forma, pode-se visualizar como se segue uma imagem da versão alfa da flecha portadora referente a esta série de trabalhos:



Figura 2.1: Versão Alfa sobre a mesa com cálculos de FOC

Foi realizado um disparo de testes com a versão Alfa, que resultou em desacoplamento do conversor de tensão, bem como perturbações e danos na fiação disposta na seta. O voo do projétil foi disforme como esperado, e o teste ressaltou novas questões de aprimoramento. Uma imagem do pós disparo pode ser vista adiante, onde é possível notar que a flecha acertou o alvo em posição deslocada para a lateral esquerda, e o conversor foi arremessado ao chão nas proximidades da área de testes:



Figura 2.2: Resultado do disparo de testes com a versão Alfa

2.5 Estado atual e continuidade

Após os estudos, deliberações, cálculos, montagem e teste da versão alfa, alguns tópicos no design puderam ser apontados como falhos e ideias de melhorias foram cogitadas. Esta seção se propõe a mapear tais informações.

Inicialmente a partir da percepção teórica de que a área de seção transversal do fio condutor é inversamente proporcional à resistência do condutor, bem como a observação de que a resistência se mostra inversamente proporcional a intensidade da corrente elétrica, foi possível identificar uma falha na escolha do fio condutor, que possui um diâmetro de 0,1mm. Além disso, os fios de cobre de diâmetro reduzido se mostraram suscetíveis à quebra, gerando mais um ponto de fragilidade do sistema.

Desta forma, será buscada, para montagens

futuras da flecha portadora, a utilização de fios condutores de maior diâmetro, capazes de transportar mais facilmente maiores intensidades de corrente e dissipando, por terem menor resistência, menor quantidade de energia em forma de calor.

Além disso, foi observada a necessidade ainda maior de criação de travas para o recuo do rolamento linear posicionado na ponta do projétil, bem como maneiras de se conter um possível giro do rolamento linear decorrente da aplicação da resistência do ar durante o voo da seta, o que faria com que o fio encontrasse o circuito em uma posição desfavorável para a ligação e passagem de corrente. A fim de sanar possíveis vulnerabilidades, foi cogitado o acoplamento de duas pequenas hastes de madeira na parte posterior da flecha, de forma a reter o recuo do rolamento linear durante a penetração da flecha.

Foi cogitada também uma terceira trava de madeira acoplada ao corpo da flecha, posicionada de forma a gerar, em conjunto com uma das hastes de trava, uma restrição para rotações em formato de "V". Em conjunto a estas hastes acopladas no corpo da seta, também cogita-se acoplar uma haste horizontal no próprio rolamento, a ser encaixada no espaço em "V", impedindo a rotação do rolamento.

Por fim, as demais partes do design previamente proposto parecem passíveis de permanência no projeto, sendo novos testes previstos para um período próximo.

2.6 Indicações a futuros projetos

Reafirmando o intuito do projeto de estimular formulações e aplicações reais referentes aos tópicos de flechas portadoras, nesta seção são descritas algumas recomendações àqueles interessados em formular projéteis como o deste trabalho ou de demais variedades.

É importante considerar o peso total obtido em uma flecha portadora, visto que um acréscimo neste fator acarretará em um acréscimo de momento no disparo, resultando em maior

penetração e causará impacto na utilização do gatilho de ativação cogitado referente ao deslocamento reverso de um rolamento linear posicionado próximo à ponta da seta. Ainda assim, a utilização desta técnica para criação de gatilhos de ativação dos sistemas portados ainda se mostra como factível e de grande relevância.

O posicionamento das diferentes peças ao longo do corpo do projétil pode ser feito de maneira iterativa como propõe este trabalho, sendo esta tática muito eficiente para deliberar acerca do posicionamento eficiente (com relação ao FOC) de cada um dos componentes cogitados para a flecha portadora almejada.

Por fim, a reutilização de materiais recicláveis pode se apresentar como uma eficiente solução para ajustes e montagens de mecanismos necessários para o bom funcionamento de uma flecha portadora.

2.7 Conclusões

A criação de uma flecha portadora aliada a um gerador de pulso eletromagnético permanece factível e cada vez mais palpável através dos testes e concepções realizadas até o momento neste trabalho. As novas adições teóricas constituem ferramental precioso na formulação do design do projeto, enquanto os resultados de testes obtidos apontam necessidades e riscos a serem considerados no prosseguimento do projeto.

O processo de desenvolvimento persiste, tendo em vista a miríade de aspectos revelados a cada etapa de pesquisa, teste e adequação. Consideram-se satisfatórios os desdobramentos e observações do período de seis meses evidenciado por este documento. Desta forma, conclui-se esta etapa do trabalho, tendo em vista sua continuidade e aperfeiçoamento.

2.8 Bibliografia

- [1] ANATEL. Resolução nº 308, de 11 de setembro de 2002. URL: <https://www.anatel.gov.br/legislacao/resolucoes/2002/257-resolucao-308>.
- [2] Easton Archery. Making sense of arrow spine. URL: <https://eastonarchery.com/2014/07/making-sense-of-arrow-spine/>.
- [3] Easton Archery. What is f.o.c. and how does it affect arrow flight? URL: <https://eastonarchery.com/2014/06/foc/>.
- [4] Young e Freedman. *FISICA III - 12ª edição*.
- [5] Halliday e Resnick. *Fundamentos de física - eletromagnetismo 10ª edição*.
- [6] C.N. Ghosh. *EMP weapons, Strategic Analysis*. 2000.
- [7] Ashby Reports. 12 arrow penetration enhancing factors. URL: <https://static1.squarespace.com/static/5d0443b188b6c90011e0ccc/t/5e9378b48f4a085e431232d4/1586722996915/2019+Terminal+Arrow+Performance+Update.pdf>.

Capítulo 3

Cibersegurança: Histórico, aspectos e tipos de Ciberataques

MATHEUS FERNANDO VIEIRA PINTO

3.1 Introdução

O mundo em que vivemos hoje é cercado por tecnologias digitais que proporcionam às pessoas facilidades em suas atividades diárias ao mesmo tempo em que geram informações e dados sobre o estilo de vida de cada um.

Ao nos comunicarmos por um aplicativo de mensagem ou rede social estamos utilizando a internet, uma grande rede de computadores que interliga milhares de dispositivos em diferentes localidades do mundo, as grandes organizações obtêm informações através dessas ferramentas para encontrar padrões e vendê-los para serem utilizados no *e-commerce*.

A cibersegurança estuda técnicas para manter as aplicações, dados e informações protegidas de um criminoso, para tanto é necessário o estudo dos ataques cibernéticos e as fragilidades encontradas nos dispositivos de informática da atualidade.

3.2 História da Cibersegurança

A cibersegurança tem origem junto às primeiras teorias de programas de computador, sendo o primeiro estudo publicado em 1949 por John von Neumann, que levantou discussões e possibilida-

des sobre a segurança em computadores [1].

Em 1967 a IBM ao testar seu novo computador com o auxílio de estudantes que possuem um conhecimento mais profundo em tais tecnologias percebeu a necessidade de medidas de segurança para evitar que terceiros tivessem acesso a configurações do sistema que deveriam ser restritas.

Em 1970 muitos foram os fatores históricos que contribuíram para o desenvolvimento da cibersegurança, entre eles a Guerra Fria que levou as duas superpotências da época EUA e União Soviética a desenvolverem tecnologias que mantivessem seus dados protegidos contra espões.

Em 1987, o Departamento de defesa dos EUA publicou um livro intitulado “Livro Laranja” destinado a fabricantes e desenvolvedores de dispositivos de informática, o livro apresentava um guia de como descobrir possíveis ameaças que os sistemas poderiam sofrer, as medidas de prevenção e as soluções para resolvê-los. No mesmo ano, após ser vítima de um ataque digital pelo vírus *Cascade* a IBM desenvolveu o primeiro antivírus, que posteriormente foi distribuído ao público.

A partir dos anos 2000 o cenário da cibersegurança mudou completamente, com a baixa nos preços dos computadores, o surgimento de dispositivos cada vez mais inteligentes como *smartpho-*

nes, a expansão da internet e o desenvolvimento de softwares para facilitar as atividades da população permitiu o acesso de diversas pessoas a tais tecnologias. Diante disso, houve a necessidade do estudo de ferramentas e técnicas para manter a proteção das informações e componentes de hardware e software contra cibercriminosos.

3.3 Aspectos da Cibersegurança

Confidencialidade, integridade e disponibilidade são os três aspectos principais da cibersegurança que serão discutidos nas seções a seguir.

Confidencialidade

Garante que apenas pessoas autorizadas tenham acesso a informação e configurações de um determinado sistema. Com a criação de técnicas criptográficas esse aspecto já é uma realidade [11].

Integridade

É a garantia de que nenhuma parte da informação será alterada por terceiros. Esse aspecto é dependente da confidencialidade, pois é necessário ter acesso ao conteúdo para posteriormente modificá-lo [11].

Disponibilidade

É a capacidade de disponibilizar acesso a um serviço ou recurso para um usuário autorizado assim que solicitado [11].

3.4 Principais tipos de Ciberataques

Um ciberataque ou ataque cibernético é qualquer tentativa de invasão a computadores pessoais, sistema de computadores ou redes de computadores. O principal objetivo do invasor (pessoa ou programa) ao realizar um ataque cibernético é acessar áreas privilegiadas e restritas de um determinado software/hardware no qual ele não possui acesso a fim de corrompê-lo, desativá-lo, roubar ou expor informações.

O estudo dos ciberataques é um tópico especial dentro da cibersegurança, pois para criar ferramentas capazes de proteger os sistemas de com-

putadores devemos conhecer quais são os tipos de ataques realizados pelos cibercriminosos.

Phishing

Phishing é uma técnica de Engenharia Social que tem por objetivo coletar informações pessoais (nomes, senhas, detalhes do cartão de crédito, etc) de usuários desprevenidos [9]. Para aplicar essa técnica os atacantes simulam mensagens aparentemente reais de bancos, redes sociais ou processadores de pagamentos online. Em muitos dos casos são enviados e-mails para as vítimas solicitando o “recadastramento” de dados referentes a uma determinada conta que ela possui em uma empresa ou organização e ao clicar em um determinado link seus dados são coletados pelos criminosos.

Existem três tipos principais de *phishing* sendo elas:

Spear phishing: essa técnica é direcionada a instituições ou indivíduos específicos.

Clone phishing: é a tentativa de direcionar a vítima para um site que é um clone do site original ao qual ela pretende acessar e desse modo obter informações pessoais da vítima durante o login.

Whaling phishing: o objetivo dessa técnica é obter informações de vítimas que possuem altos cargos ou personalidades de relevância.

Algumas etapas devem ser seguidas pelos criminosos para realizar um ataque tradicional de *phishing* sendo elas:

- **Fase de planejamento**: é nessa fase em que o atacante escolhe quem será a vítima, quais são as informações que serão coletadas e define o objetivo do ataque.
- **Fase de preparação**: é nessa etapa em que o atacante prepara todos os recursos (e-mails, sites falsos, etc) necessários para realizar o ataque.
- **Fase de ataque**: é nessa fase em que o ataque realmente ocorre, o atacante encaminha a mensagem para a vítima de acordo com o meio (e-mail, websites, aplicativos de mensagens instantâneas, etc) que ele escolheu na fase de planejamento.

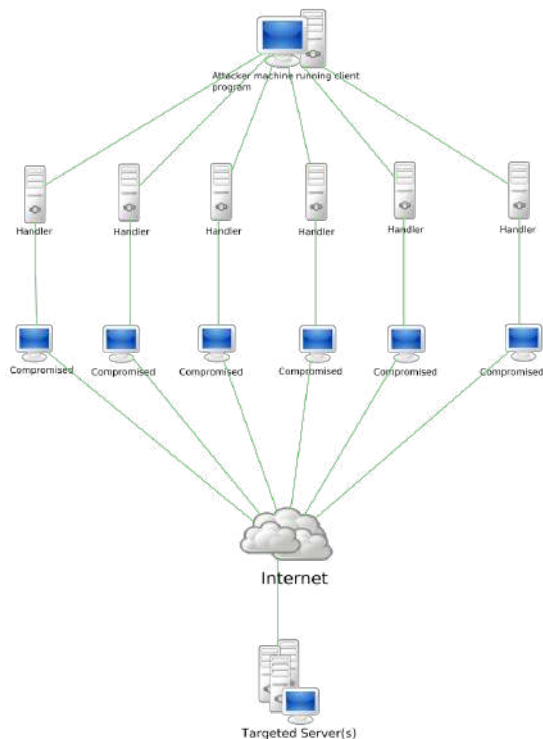


Figura 3.2: Exemplo do funcionamento de um ataque do tipo DDoS. [7]

dos campos de um formulário de uma página web permitem ao atacante “injetar” partes de instruções SQL na instrução original, ou seja, modificá-la afim de realizar consultas ou alterações no banco de dados podendo obter informações restritas ou até mesmo corrompê-lo. Para compreendermos melhor como esse ataque funciona vamos analisar um pequeno exemplo.

Considere uma parte do código do *backend* em PHP de uma aplicação web.

```
$user = $_POST['user'];
$password = $_POST['password'];
$sql = "SELECT * FROM users
      WHERE user = '". $user. "' AND
      password = '". $password. "'
      ";
$process = mysql_query($sql);
```

O trecho de código acima atribui o valor dos campos do formulário da aplicação *user* e *password* as variáveis *\$user* e *\$password* respectivamente, e em seguida cria uma consulta ao banco de dados conectado a aplicação que é representada pela variável *\$sql*, o método *mysql_query* realiza a consulta e atribui o resultado a variável *\$process*.

Perceba que o desenvolvedor do backend dessa aplicação não se preocupou em realizar a checagem dos valores dos campos recebidos no formulário e é nesse ponto que um ataque do tipo injeção SQL pode ocorrer.

A instrução SQL armazenada na variável *\$sql* solicitará ao banco de dados que retorne todos os usuários que possuem o atributo *user* igual ao valor da variável *\$user* e o atributo *password* igual ao valor da variável *\$password*. Considere que o campo *user* armazene o valor *user = p'*; *DROP TABLE users; -*, nesse caso a instrução SQL seria alterada devido a sua estrutura e desse modo o conteúdo da variável *\$sql* seria *SELECT * FROM users WHERE user = 'p'; DROP TABLE users; -' AND password= '1234'*.

A nova instrução obtida não irá realizar o que o sistema estava programado para fazer, ao contrário disso ela irá remover a tabela *users* do banco de dados fazendo com que a aplicação perca todos os dados relativos aos seus usuários. Os ataques podem variar de atacante para atacante dependendo dos seus objetivos, porém a abordagem é a mesma.

Para evitar esse tipo de fragilidade é importante que os desenvolvedores do *backend* da aplicação tenham atenção, de modo a evitar que os valores vindos do campo do formulário da aplicação sejam inseridos diretamente a uma consulta SQL que será realizada em um banco de dados, mas sim utilizar funções específicas para preparar as consultas.

Ransomware

Ransomware é um *malware* que é utilizado para infectar uma determinada máquina durante um ataque, bloqueando todo o seu sistema opera-

cional de modo a impedir o acesso da vítima aos dados ou sistema [10]. Após realizar o ataque, o atacante solicita à vítima o pagamento em criptomoedas para liberar o acesso à máquina.

Quando uma máquina é infectada, o *malware* irá codificar os dados do usuário em segundo plano até que todo o sistema esteja infectado, sem levantar nenhuma suspeita. Uma vez que o vírus se espalha por toda a máquina, uma mensagem é emitida ao usuário através de um *pop-up*. Se a vítima não realizar o pagamento, assim como solicitado pelo atacante seus dados podem ser perdidos ou expostos [10].

Um ataque por *ransomware* geralmente se inicia através de um e-mail contendo um anexo para o *malware* ou um link para um site malicioso (canais utilizados durante a técnica de *phishing*) no qual o usuário recebe uma mensagem alertando que o seu computador está infectado e em seguida solicita que ele acesse um link para desbloqueá-lo.

Existem dois tipos de *ransomware* sendo eles:

Crypto-ransomware: É o tipo mais utilizado, ele utiliza técnicas de criptografia para bloquear dados importantes da vítima. Um tipo famoso de *crypto-ransomware* é o CryptoLocker que utiliza criptografia do RSA de até 2048 bits [3].

Locker-ransomware: Esse tipo de *ransomware* não utiliza técnicas de criptografia, seu principal objetivo é bloquear o acesso da vítima ao sistema [3].

Para nos prevenirmos de ataques que utilizam *ransomware* é importante que ao receber um e-mail com algum link verificar a veracidade do remetente, evitar fazer downloads de aplicativos em sites duvidosos, manter o sistema operacional e os navegadores atualizados, usar software de segurança (antivírus) de empresas confiáveis, etc.

3.5 Exemplos de ciberataques reais

WannaCry

Em 2017 uma ameaça chamada WannaCry invadiu de maneira mundial a rede de computado-

res, seus principais alvos eram as máquinas operadas pelo sistema Microsoft Windows [4]. O objetivo principalmente desse *crypto-ransomware* era encontrar dados importantes e de valor para os usuários e encriptá-los utilizando técnicas de criptografia conhecidas, dessa maneira os afetados não poderiam ter acesso aos seus arquivos pessoais. Os dados só seriam liberados se as vítimas realizassem pagamentos em Bitcoins assim como solicitados pelos hackers.

Stuxnet

Stuxnet é um *worm* (vírus de computador auto replicante) que foi descoberto em 2010, seu principal alvo era o sistema operacional SCADA desenvolvido pela Siemens. O ataque ocorreu em 2010 às centrífugas de urânio do Irã que ao serem infectadas tiveram o seu funcionamento bloqueado e como consequência o atraso no desenvolvimento nuclear do país [4].

Mirai

Mirai foi um ataque do tipo DDoS que ocorreu em 2016, seu principal alvo era a empresa Dyn uma provedora de serviços de DNS. O ataque ocorreu com a utilização de milhares de dispositivos conectados à internet (gravadores de vídeo, câmeras IP, etc) infectados por um *malware* que ao realizarem requisições ao servidor da empresa inviabilizaram o seu funcionamento [4]. Serviços como Paypal, Twitter, Netflix e Spotify foram interrompidos como consequência do ataque Mirai.

Ataque às Lojas Renner

Entre os dias 19 e 21 de agosto de 2021, as lojas Renner foram vítimas de uma ataque cibernético que utilizou um *ransomware* para bloquear e inviabilizar o acesso dos usuários ao site da empresa. Ao todo, cerca de 1,3 mil servidores foram afetados pelo ataque [6].

3.6 Conclusão

O avanço tecnológico nos meios de comunicação nos trouxe facilidades no dia a dia, com o surgimento da internet e a criação de dispositivos capazes de se comunicarem através dela permitiu

o surgimento de aplicações capazes de realizar tarefas do dia a dia das pessoas com compras online, pagamentos bancários, etc.

A cibersegurança possui um papel primordial nessa nova maneira de se comunicar, pois é através dela que podemos manter nossas informações, dados e serviços seguros de terceiros. É importante conhecer quais são os principais ataques cibernéticos da atualidade para poder se prevenir deles.

3.7 Bibliografia

- [1] CanalTech. História da segurança virtual: a origem da cibersegurança. URL: <https://canaltech.com.br/seguranca/historia-da-seguranca-virtual-a-origem-da-ciberseguranca-200930/>.
- [2] CanalTech. O que é dos e ddos? URL: <https://canaltech.com.br/produtos/o-que-e-dos-e-ddos/>.
- [3] InfoWester. Ransomware. URL: <https://www.infowester.com/ransomware.php>.
- [4] Kaspersky. Os ciberataques mais famosos dos últimos tempos. URL: <https://www.kaspersky.com.br/blog/five-most-notorious-cyberattacks/11042/>.
- [5] MalwareBytes. Phishing. URL: <https://br.malwarebytes.com/phishing/>.
- [6] Consumidor Moderno. Lojas renner afirmam que não houve vazamento de dados no ataque cibernético. URL: <https://www.consumidormoderno.com.br/2021/08/30/lojas-renner-vazamento-dados/>.
- [7] Wikipedia. Ataque de negação de serviço. URL: https://pt.wikipedia.org/wiki/Ataque_de_nega%C3%A7%C3%A3o_de_servi%C3%A7o.
- [8] Wikipedia. Injeção de sql. URL: https://pt.wikipedia.org/wiki/Inje%C3%A7%C3%A3o_de_SQL.
- [9] Wikipedia. Phishing. URL: <https://pt.wikipedia.org/wiki/Phishing>.
- [10] Wikipedia. Ransomware. URL: <https://pt.wikipedia.org/wiki/Ransomware>.
- [11] Wikipedia. Segurança de computadores. URL: https://pt.wikipedia.org/wiki/Seguran%C3%A7a_de_computadores.

Capítulo 4

Web Scraping com Python - Estudo de caso

ENZO L. FERNANDES

FRANCIENE BERNARDI

4.1 Introdução

Web scraping é o processo para coletar dados de um site ou plataforma web, para extrair dados. Esse projeto tem como objetivo, utilizar da técnica de *web scraping*, para extrair informações de qualquer plataforma web, que possua uma boa estruturação HTML, e demonstrar como é possível implementar uma API, para filtrar e consumir esses dados.

4.2 Desenvolvimento

Nas seções a seguir são apresentados a implementação e demonstração das técnicas de *web scraping*, além de exemplos da utilização destas técnicas em alguns sites de notícias.

Implementação

Para a implementação do projeto foi utilizada a Python (versão 3.8.6) pela sua praticidade e simplicidade, além de dispor de bibliotecas adequadas para realizar os objetivos propostos. Durante o desenvolvimento do projeto foram utilizadas as seguintes bibliotecas:

- *beautifulsoup4*: para analisar e filtrar conteúdos do código fonte. Por meio de um *parser* é possível identificar os conteúdos das páginas e realizar a filtragem destes conteúdos utilizando *tags* e *divs*.

- *requests*: para realizar requisições a servidores web e obter o código fonte das páginas.
- *re (Regular Expressions)*: para filtrar os resultados encontrados pelos métodos utilizados pela biblioteca *beautifulsoup4*.

Demonstração

As técnicas de *web scraping* foram demonstradas por meio das implementações de métodos capazes de obter informações de alguns sites. Inicialmente as técnicas foram utilizadas em sites de notícias para obter as principais notícias, para demonstrar a técnica e posteriormente foi feita uma API para obter informações da web utilizando *web scraping*.

As informações obtidas dos sites consistem de títulos de notícias que estavam presentes nas páginas principais dos portais de notícias. Os sites utilizados durante os testes das técnicas foram:

- www.uol.com.br: Para o site da UOL, utilizamos da *tag* HTML e de uma classe.

```
dados uol = soup.find_all("h1", class_ = re.compile("^(titulo|color3)"))
```

- www.g1.globo.com: Neste site, utilizamos da *tag* HTML e uma classe. Percebemos que neste site, ele vai carregando mais notícias de acordo com o *scroll* que o usuário dá na

tela, por isso quando extraímos as informações das notícias, retorna sempre a quantidade *default* a página principal.

```
dados_globo = soup.find_all(
    "a",
    class_ = re.compile("
feed-post-link
gui-color-primary
gui-color-hover|_b"))
```

- www.folha.uol.com.br: Para este site, utilizamos da uma *tag* HTML e uma classe.

```
dados_folha = soup.find_all(
    "h2", class_ = "c-
headline_title")
```

- www.ufscar.br: No site da UFSCar, nosso objetivo era filtrar as últimas notícias, para isso utilizamos uma *tag* HTML.

```
dados_ufscar = soup.find_all(
    "tbody")
```

4.3 Estudo de caso

Para o estudo de caso de aplicação das técnicas de *web scraping* vistas foi escolhido o site Fundamentus responsável por disponibilizar informações sobre Investimentos em ativos de Renda Variável. O site apresenta as informações por meio da busca realizada na página inicial e mostrando informações relativas à empresa como indicadores, lucro, rendimento nos últimos meses, etc.

Disclaimer: Métodos de busca automatizada em páginas da web, tais como *web scraping*, não são sempre autorizados pelos administradores dos sites já que a sua utilização sistemática pode gerar sobrecarga nos servidores quando utilizada em larga escala. Os desenvolvedores deste projeto entraram em contato com o administrador do site Fundamentus que negou o uso da API criada utilizando *web scraping*, desta forma o conteúdo demonstrado neste seção tem apenas caráter demonstrativo.



Figura 4.1: Página do site Fundamentus contendo informações sobre a ação B3 S/A cujo código de negociação é B3SA3.

Para este site foi desenvolvida uma API capaz de realizar requisições ao site, filtrar os conteúdos utilizando *web scraping* e encapsulá-los por meio de orientação a objetos. A API foi desenvolvida para ser utilizada na linguagem de programação Python de modo que seja possível obter as informações de forma sistematizada do site.

Neste sentido a API é capaz de obter as informações que estão sendo exibidas como na Figura 1 de uma maneira que pode ser acessada pelos usuários ao utilizar a API. Um possível exemplo de sua utilização é dado abaixo no formato do *shell* iPython:

```
>>> from fundamentus import Acao
...
>>> weg = Acao(ticker = "WEGE3")
>>> info_weg = info_weg.request_fundamentos()
...
>>> info_weg["Cotacao"]
34,25
...
>>> info_weg["Nro. Acoes"]
4.197.320.000
...
>>> info_weg["P/L"]
53,94
...
>>> info_weg["ROE"]
```

As buscas realizadas pela API se limitam à página inicial de cada uma das ações, entretanto elas poderiam ser expandidas para outras páginas como *proventos*, utilização das mesmas técnicas para Fundos Imobiliários em páginas como *Administrador*, *Proventos*, etc.

4.4 Resultados e conclusão

Nesta sessão vamos discutir alguns casos especiais encontrados no momento da aplicação para demonstração.

Percebemos que em alguns sites como os desenvolvidos em Flutter ¹, não é possível utilizar dessa técnica, já que não possui um HTML estruturado. Pelos testes realizados acredita-se que o conteúdo das páginas que utilizam Flutter é renderizado tal qual uma imagem e como a técnica utilizada neste projeto depende diretamente do conteúdo do código fonte da página não é possível extrair as informações utilizando *web scraping*.

Outro ponto importante observado nos resultados foi que em alguns sites, como no G1, algumas ações que dependem da interação do usuário impedem que as informações sejam obtidas. Neste site em particular, o conteúdo do site apresentada na *home* é carregado dinamicamente dependendo da quantidade de *scroll* feita pelo usuário.

Desta forma as técnicas de *web scraping* são capazes de obter apenas a menor parte do conteúdo carregado por padrão (sem nenhum *scroll*). Para o site do G1 são apresentadas por padrão três notícias principais e mais sete notícias secundárias que são carregadas dinamicamente conforme o usuário navega pela página.

Por fim, concluímos que o *web scraping* é uma técnica eficaz para obtenção de informações com base no código fonte de páginas web. Entretanto, seu funcionamento depende que os dados estejam codificados dentro do HTML da página e do conhecimento da estruturação do HTML por

parte do usuário para que seja possível obter as informações de forma efetiva.

Estas técnicas podem ser aproveitadas desde o uso em projetos de pequena escala, ou projetos pessoais, até projetos para serem distribuídos em larga escala para usuários desde que os sites analisados deem suporte para as buscas automatizadas e haja o consentimento dos administradores.

¹Disponível em: <https://flutter.dev/>

Capítulo 5

NAT Punchthrough: Multiplayer, P2P e adversidades

ANDERSON PINHEIRO GARROTE
DANILO ISAMU INAFUKU

Resumo

Este trabalho tem por objetivo apresentar o conceito de NAT (*Network Address Translation*), sua adversidade com aplicações P2P, e o *NAT Punchthrough*, um sistema criado na tentativa de contornar tal adversidade. Uma versão deste trabalho em vídeo também existe, acessível em <https://youtu.be/5NtiRYI1C8g>.

Palavras-chave: *Networking*; NAT; P2P; Jogos multiplayer.

5.1 Introdução

Multiplayer sempre foi uma parte importante da história dos jogos em geral, e com videogames, isso não é diferente. Afinal, a interação social advinda de poder jogar com outras pessoas através da rede é algo essencial para vários dos mais populares jogos atualmente.

Porém, é de conhecimento geral que a maioria dos jogos são hospedados exclusivamente pela empresa desenvolvedora - jogadores têm permissão para se conectar a servidores, mas não para criá-los. O que acontece, então, quando a empresa decide desligar os servidores, por qualquer razão que seja? Como os jogadores não têm permissão

nem a informação necessária para hospedar as próprias partidas, o jogo efetivamente deixa de existir.

Conexões P2P (*peer-to-peer*) poderiam ser vistas como uma solução para este problema - com esta estrutura, jogadores não iriam depender de servidores externos, apenas precisando se conectar diretamente. Devido ao sistema NAT (*Network Address Translation*), porém, consequência dos roteadores e da necessidade de multiplicar a quantidade de IPs válidos dentro da configuração IPv4, isso se tornou complicado.

Para contornar isso, o sistema de *NAT Punchthrough* foi levantado como uma forma de “atravessar” a barreira criada pelo NAT. Apesar de ainda depender de um servidor externo, esse servidor serviria apenas para criar a conexão inicial, podendo assim ser mantido com muito mais facilidade e podendo até ser utilizado para vários diferentes jogos ao mesmo tempo (Um exemplo desta estrutura é o Xbox Live).

Este artigo procura providenciar uma explicação maior sobre este sistema.

5.2 Conceitos Básicos

Para entender como o *NAT Punchthrough* funciona, é necessário entender os básicos de *Networking* pela Internet. Para localizar e estabelecer conexões com outros dispositivos, um endereço (co-

hecido como IP) é utilizado para providenciar um identificador para uma máquina. Cada máquina também possui várias portas, utilizadas primariamente para identificar o processo dentro da máquina responsável pela conexão.



Figura 5.1: IP e porta - cada porta, no geral, identifica um processo.

Isso, porém, fica mais complicado. A maioria das pessoas possuem roteadores em suas casas, que fornecem internet para vários aparelhos ao mesmo tempo. Como o roteador age como o dispositivo “externo”, todas as máquinas que se conectam por ele possuem o mesmo endereço IP, o que cria um problema. Como distinguir entre elas, se elas compartilham o mesmo endereço?

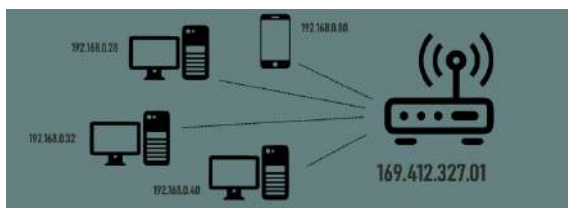


Figura 5.2: O roteador e os endereços internos.

A solução é a criação de endereços internos. Esses endereços funcionam apenas dentro da sub rede alimentada pelo roteador, e servem para distinguir entre diferentes dispositivos dentro dessa rede privada. Isso, porém, traz outro empecilho, já que como esse endereço funciona apenas internamente, ele não pode ser utilizado por uma máquina externa para fins de conexão. O NAT veio para solucionar isso.

Network Address Translation

O NAT mapeia uma das portas do roteador com uma combinação de endereço e porta interna. Assim, basta uma máquina externa conhecer a porta

do roteador certa para poder se conectar com uma porta específica de um dispositivo específico dentro da sub rede.

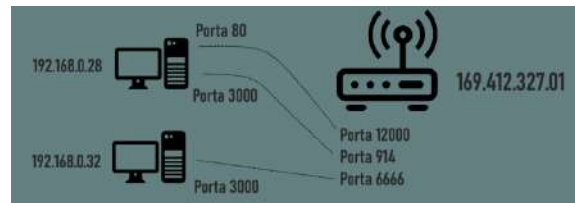


Figura 5.3: O NAT mapeia uma das portas do roteador com um endereço+porta internos.

E como forma de segurança, o roteador bloqueia endereços desconhecidos de enviar informações para suas portas. É isso, inclusive, que causa o problema para o P2P - endereços tentam se conectar, mas não conseguem devido ao roteador os bloqueando. Para fazer com que ele autorize-os, é preciso que o roteador os conheça. E uma forma de “introduzir” um endereço ao roteador é enviar um pacote para esse endereço.

NAT Punchthrough

O NAT *Punchthrough* funciona usando essa ideia de registrar um endereço como válido enviando um pacote para ele. Primeiro, um usuário envia um pacote com um nome identificador (neste exemplo, será “Fulano”) e seu endereço e porta internos, em direção a um servidor externo.

O servidor externo, por sua vez, examina o endereço e a porta de saída do remetente (ou seja, o endereço e porta externos), insere ambos no pacote recebido, e armazena ele.



Figura 5.4: Fulano envia seu pacote.

Do outro lado, outro cliente faz a mesma coisa, mas ao invés de providenciar um nome identi-

gador para se identificar, ele providencia-o com o intuito de indicar com quem ele quer se conectar (neste caso, “Fulano”). O servidor também recebe essas informações e se prepara para fazer a troca.



Figura 5.5: Cicrano também envia seu pacote.

Com as informações de ambos os clientes em mãos, o servidor passa o endereço do cliente 1 para o cliente 2 e o endereço do cliente 2 para o cliente 1. Ao receber esses endereços, os clientes então começam a enviar pacotes para o IP e porta recebidos, na tentativa de estabelecer uma conexão.



Figura 5.6: Troca de pacotes.

Ao enviar um pacote, o roteador registra o endereço destinatário em sua tabela de roteamento, assim registrando-o. Quando o pacote chegar do outro lado, será rejeitado, pois ele ainda não foi registrado pelo roteador do outro lado. Porém, assim que o outro lado enviar um pacote, esse pacote conseguirá passar devido ao registro na tabela de roteamento. Se ambos os lados fizerem seu trabalho, eles poderão se comunicar [3][2]

5.3 Limitações

Um detalhe ainda não comentado são os endereços internos que foram passados para o servidor externo. Afinal, se eles não podem ser usados para se conectar externamente, por que eles são passados?

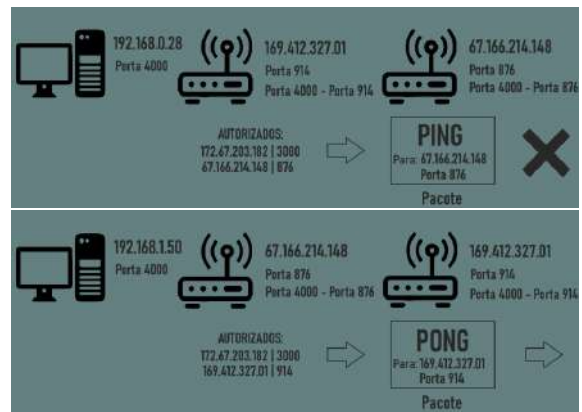


Figura 5.7: Fulano e Cicrano enviam pacotes um para o outro.

O motivo é porque existe a possibilidade de ambos os clientes estarem (possivelmente sem saber) dentro da mesma sub rede. Este problema, conhecido como “problema de Hairpin” [2], caracteriza-se pelo fato de que muitos roteadores são incapazes de enviar um pacote para si próprio. Assim, para lidar com esta situação de forma mais consistente, é preferível enviar o endereço interno também para fazer a conexão em ambiente local.

A situação mais limitante, porém, ocorre quando os dois roteadores estão em um NAT maior, um terceiro NAT, normalmente devido às operadoras criarem seu próprio espaço de endereço. Neste caso, o endereço externo do roteador fica indisponível ao servidor externo, já que ele, por sua vez, só terá acesso ao endereço do NAT maior. Além disso, caso ambos os clientes estivessem no mesmo NAT maior, o mesmo problema de Hairpin ocorre, apenas a um nível acima [2].

5.4 Considerações finais

Devido às várias limitações delineadas acima, o NAT *Punchthrough* provou-se ser, infelizmente, uma técnica inconsistente. Aplicações P2P normalmente costumam demandar dos clientes uma configuração própria de *port forwarding* em seus roteadores, ou utilizam servidores externos para agir como *relays* de mensagens [1][4].

Mesmo assim, é uma técnica interessante e utilizável como tática secundária em certas situações, e por isso, merece ainda ser citada tanto de forma didática quanto de forma prática.

5.5 Bibliografia

- [1] Chris Butcher. Recreating the lan party online. Acesso em: 28 mai. 2021. URL: http://halo.bungie.org/misc/gdc2005_recreating_lanparty/.
- [2] Bryan Ford. Peer-to-peer communication across network address translators. Acesso em: 20 apr. 2021. URL: <https://bford.info/pub/net/p2pnat/>.
- [3] Keith Johnson. Nat punch-through for multi-player games. Acesso em: 20 apr. 2021. URL: <https://keithjohnston.wordpress.com/2014/02/17/nat-punch-through-for-multiplayer-games/>.
- [4] One One Seven. A guide to networking, matchmaking, and host in halo, version 2.1.1. Acesso em: 28 mai. 2021. URL: <http://halo.bungie.org/misc/networkguide.html/>.

Capítulo 6

O hacking social por trás de jogos “Gacha”

AGATA RANGEL DRIGO
agata.drigo@estudante.ufscar.br

FELIPE BERTONI SALVATI
felipesalvati@estudante.ufscar.br

FERNANDO FAVARETO ABROMOVICK
fernandofa@estudante.ufscar.br

GIOVANNI MARÇON ROSSI
giovanni.rossi@estudante.ufscar.br

MARIANE LAMAS MALHEIROS
mariane.malheiros@estudante.ufscar.br

Resumo

Jogos do gênero *Free to Play* do tipo *gacha* estão aumentando tanto em popularidade quanto em lucro nas duas últimas décadas, com o jogo *Fire Emblem: Heroes* lucrando mais de 850 milhões de reais ([3]).

Através deste artigo, os autores buscam entender melhor como esse aumento de popularidade aconteceu e quais técnicas ao utilizadas para gerar um lucro tão grande, mesmo com o produto final sendo gratuito.

Palavras-chave: *hacking* social, *gacha*, jogo de azar, manipulação social, jogos *Free to Play*, jogos

Freemium

6.1 Introdução

Dentro do mercado de jogos, o gênero de jogos *gacha* se destacou nas últimas décadas, devido a sua geração de lucros em comparação com o modelo de negócio atual de alguns outros gêneros de jogos. Em uma notícia, foi revelado que a os jogos para celulares feitos pela Nintendo alcançaram no total mais de 1.5 bilhões de reais em lucro no ano de 2020 [3].

Para entender como que esse gênero consegue gerar tanto lucro, primeiro é necessário entender os conceitos básicos que todos os jogos *gachas* operam. Estes conceitos são baseados em máquinas de *Gachapon*.

O termo *gacha* foi originado por máquinas japonesas que, ao inserir uma quantidade de dinheiro, a maquina fornece um prêmio colecionável (um boneco, chaveiro, entre outros).

Dentro destas máquinas existem vários tipos diferentes de prêmios que um usuário pode receber, com alguns desses prêmios sendo mais raros de adquirir do que outros. Essa raridade e determinada em quão frequente esse prêmio existe em uma determinada máquina.

Para jogos virtuais, o conceito é similar, mas ao invés da raridade ser determinada pela frequência em uma determinada máquina, ela é determinada pelos criadores do jogo como

um sistema de porcentagem; quanto menor a porcentagem de chances de adquirir o prêmio (ou seja, mais difícil de conseguir), melhores os atributos, habilidades e características do prêmio.

Com isso em mente, o objetivo deste artigo é entender como esse gênero ganhou tanta popularidade nas últimas décadas e quais as técnicas que são utilizadas pelo gênero para manipular seus usuários a gastarem mais dinheiro.

6.2 Monetização em jogos *gacha*

Embora existam vários jogos do tipo *gacha* e eles sejam diferentes em seu conteúdo e jogabilidade, a maioria deles segue um padrão similar às máquinas *Gachapon*: O jogador coloca dinheiro para ter uma chance de receber um prêmio, mas neste caso, o prêmio é um personagem jogável, sendo que cada personagem possui diferentes atributos, características físicas e poderes.

Assim o jogador não tem o controle garantido de poder jogar com um personagem que ele deseje usar, as vezes alterando as chances do jogador ter sucesso no jogo.

Normalmente os jogos *gacha* possuem vários personagens de raridades diferentes, e quanto mais forte e raro um personagem, menor a probabilidade de obtê-lo. Assim, quando o jogador junta uma certa quantidade de moedas especiais (exclusivas de um jogo), ele pode trocá-las por uma ou mais chances de obter novos personagens.

A Figura 1 mostra uma tela onde um jogador consegue trocar a moeda especial em um ou mais personagens, essa tela é conhecida como um *banner*.

O jogador pode adquirir as moedas especiais de duas formas: ao completar missões e objetivos no jogo, que disponibilizam uma quantidade pequena das moedas especiais, ou comprando pacotes que contém uma quantidade pré-determinada da moeda especial, usando dinheiro de verdade.

Na grande maioria dos jogos *gacha*, a quantidade de moedas especiais adquirida gratuita-



Figura 6.1: Tela de obtenção de novos personagens, conhecida como *banner*, no jogo *Genshin Impact*. Podemos ver a diferença de raridades entre os personagens (4 estrelas e 5 estrelas) - fonte

mente por missões é relativamente baixa e as missões geralmente exigem um tempo considerável para serem completadas, então é mais difícil e lento o processo de tentar obter um personagem nos *banners* de maneira gratuita. Essa frustração acaba estimulando os jogadores a gastarem dinheiro para ter uma chance de conseguir o personagem almejado.

6.3 Tipos de jogadores

Para entender como os usuários são manipulados a gastar, é preciso primeiro entender como esses usuários são vistos pela comunidade e por outros jogadores. Dentro de comunidades de jogos *gacha* existem perfis específicos de jogadores que gastam uma certa quantidade de dinheiro. São eles: *F2P*, *dolphin* e *whale*. Os valores limítrofes de cada categoria não são bem definidos, logo os perfis citados acima serão explicados com mais detalhes nas subseções seguintes.

Algumas comunidades acrescentam mais subcategorias, porém, por questões de simplificação, elas serão desconsideradas neste artigo.

Para definir a porcentagem de usuários, duas fontes principais foram usadas: um Google Forms com os usuários de uma comunidade do Reddit do jogo *gacha Fire Emblem: Heroes* ([4]), e uma pesquisa desenvolvida por um estudante da *National Sun Yat-sen University* ([5]).

F2P (Free to Play)

Essa categoria inclui a maior parte da base de usuários, incorporando aqueles que não gastam dinheiro com o jogo ou, se gastam, gastam uma quantidade menor do que 50 dólares por mês.

No caso de ([5]), a categoria F2P inclui as categorias *Free to play* - usuários que não gastam dinheiro algum - e *Minimum spender* - usuários que gastam só até uma certa quantia baixa de dinheiro.

Em geral, tem-se que por volta de 50% ([4]) - 72% ([5]) dos usuários se considera F2P, englobando a grande maioria da *playerbase* (jogadores de um jogo).

Dolphin (Golfinho)

Essa subcategoria começa a contar com as pessoas que gastam um montante mais significativo, de 250 a 500 dólares por mês. De um ponto de vista estrutural, são esses jogadores que começam a beneficiar os desenvolvedores do jogo, fornecendo uma fonte estável, porém pequena, de dinheiro por mês.

Eles são a segunda categoria com a maior porcentagem de usuários, formando aproximadamente 16% ([4]) - 24% ([5]) da *playerbase*.

Whale (Baleia)

Essa subcategoria é de onde o lucro das empresas de *gacha* realmente vem. Os gastos de *whales* tendem a ser acima dos 1000 dólares por mês, assim indiretamente financiando o jogo para que ele se mantenha atualizado conforme o tempo passa.

Sem *whales*, é difícil que um jogo *gacha* se mantenha além do primeiro ano, considerando a necessidade financeira de manter os servidores atualizados, de pagar artistas, escritores e programadores para produzir novos conteúdos, além do lucro da empresa.

De um ponto de vista de porcentual, *whales* consistem da menor parte da *playerbase*, apenas 4% ([5]) - 20% ([4]).

6.4 Técnicas predatórias

Jogos do gênero *gacha* são conhecidos por possuir técnicas monetárias predatórias, manipulando jogadores a gastarem dinheiro de maneira compulsiva em seus jogos, como o caso de um jogador que gastou 70.000 dólares em apenas um ano em um jogo deste gênero ([2]). Uma comparação que é feita com frequência com jogos deste gênero e com cassinos, pois os dois exploram usuários que tem dificuldade com gasto compulsivo de dinheiro.

Isso se dá por conta do chamado *Sunk Cost Fallacy* (ou "Custo irrecuperável" em português, apesar de não ser um termo muito utilizado): quanto mais tempo ou dinheiro se é investido em alguma coisa, mais difícil é de parar de fazer essa coisa, por conta do pensamento de que é necessário recuperar ou compensar esses recursos perdidos de alguma maneira.

Alguns dos jogos que serão discutidos se aproveitam disso através do sistema de *pity* (como *Genshin Impact* ou *Fire Emblem: Heroes*), que faz com que sua chance de conseguir um personagem raro aumenta conforme mais da moeda especial é gasta. Então, para os jogadores, vale mais a pena continuar gastando para aproveitar o aumento percentual de conseguir o personagem raro, senão tudo o que foi gasto até o momento acaba sendo considerado perdido.

Powercreep

Uma das maneiras clássicas de incentivar os jogadores a começar ou continuar gastando dinheiro e através do desenvolvimento de novos personagens que são cada vez mais fortes do que os anteriores.

Logo, se o jogador quer se manter com um time forte o suficiente para continuar conseguindo desbloquear áreas novas do jogo, ou manter seu time viável para jogar competitivamente com

outros usuários, precisa conseguir esses novos personagens, fazendo com que ele seja pressionado a gastar dinheiro no jogo para continuar forte.

Esse conceito, *powercreep*, é usado para descrever situações onde, conforme novas unidades de personagens são lançadas em um jogo, mais os personagens já obtidos ficam obsoletos. Isso pode trazer dificuldades para os jogadores conforme mais tempo de jogo passa, pois os personagens novos são cada vez mais fortes, e os personagens antigos mantêm seus atributos, ficando cada vez mais fracos em relação aos novos. Assim, se cria uma necessidade quase urgente de obter os novos personagens caso se queira continuar tendo sucesso no jogo.

Esse conceito só passa a ser problemático quando ocorre de maneira súbita (com um personagem muito melhor do que os personagens anteriores sendo lançado) ou quando ocorre de maneira recorrente (mostrando que o objetivo do lançamento do personagem é apenas conseguir dinheiro).

Um exemplo do uso dessa técnica é visto no jogo *Fire Emblem: Heroes*. A Figura 2 mostra uma foto em que usuários fizeram comparação de pontos de atributos entre personagens durante os anos.

Alguns outros conceitos de *powercreep* existem, mas não se aplicam como uma técnica predatória. Porém, acabam sendo interessantes para demonstrar outros motivos de usuários gastarem dinheiro: O conceito de **powercreep de arte** indica que o personagem recentemente lançado possui um design mais interessante do que os anteriores, e um *powercreep de voz* pode indicar um trabalho mais bem feito dos dubladores, entre outros.

Todos esses tipos de *powercreep* podem fazer com que os jogadores gastem mais dinheiro no jogo, como uma tentativa de incentivar os usuários a gastarem mais dinheiro, mas também podem ser vistos como indicadores de que os desenvolvedores do jogo estão dando atenção aos seus jogadores e tendo cuidado em seu desenvol-

vimento.



Figura 6.2: *Powercreep* de personagens da classe arqueiro em *Fire Emblem: Heroes* feito por u/CyanYoh no Reddit, sendo que BST é a soma de todos os pontos base do personagem - fonte

Por tempo limitado

Outro método de predação é o conceito de conteúdos disponibilizados por tempo limitado. A ideia por trás disso é o sentimento de exclusividade e urgência: "se você não tentar conseguir esse personagem/item agora, nunca mais terá a oportunidade". A Figura 3 exemplifica esse método.

Essa técnica se aproveita do *Fear of Missing Out* (FOMO) (ou "o medo de perder algo", em português) que muitas pessoas têm. O conceito de perder a oportunidade de aproveitar de uma situação que outros podem estar aproveitando é encontrado em várias outras facetas do dia-a-dia (sendo que o termo FOMO é bastante usado em investimentos na bolsa de valores ou em criptomoedas), e que, no caso de jogos gacha, se combina com o *powercreep* descrito anteriormente para incentivar o gasto.

Bônus por moedas pagas

Uma estratégia que não é vista com muita frequência em tantos jogos, mas que esta cada vez mais popular, é a diferenciação entre moedas compradas através de microtransações e moedas ganhas de graça ao jogar o jogo. Essa diferença é utilizada para proteger o valor gasto pelos jogadores, sendo que as moedas de graça são gastas antes das pagas.

Porem, alguns jogos vão um pouco além com essa distinção, oferecendo descontos que só são ganhados quando moedas pagas são utilizadas ou com "caixas" especiais, com porcentagens diferentes e outros benefícios. A Figura 4 mostra essa diferença entre moedas pagas e gratuitas.



Figura 6.3: Banner de personagens no jogo *Love Live! School Idol Festival*. Perceba o uso de *limited*, indicando que o conteúdo não estará sempre disponível no jogo - fonte



Figura 6.4: Banner de *Love Live! School Idol Festival*, note a diferença de preço das moedas (denotadas como corações) na versão paga (*paid*) laranja e a versão gratuita (*free*) rosa - fonte: screenshot dos participantes.

Alters e Skins

Um dos fatores que também determina se alguns jogadores querem conseguir um personagem e proveniente de um apelo por meio da sua estética e personalidade. A introdução de *alters* (ou só *alt*) e *skins* permitem que um jogador que gosta de um certo personagem, consiga obter ele novamente,

com poderes e/ou estética diferentes. Essas estratégias focam no vínculo que o jogador forma com o personagem do jogo para fazer com que ele tenha todas as versões do personagem.

Alters

Esse conceito indica a reintrodução de um personagem já existente no jogo, sendo uma versão alternativa (logo *alternativa*), com visuais, comportamento e atributos diferentes.

Nas Figuras 5 e 6, o personagem mostrado em ambas é o mesmo, Arjuna. Porém, é possível ver que eles possuem características diferentes. A quantidade de ataque (em 5 com o valor 14500 e em 6 com o valor 10982), de pontos de vida em (5 com o valor 15484 e em 6 com o valor 12814), e a classe/categoria deles (para Arjuna Archer (Arqueiro) e para a versão alternativa Berserker) são todos diferentes.



Figura 6.5: Imagem do personagem Arjuna do jogo *Fate/Grand Order* - fonte

Skins

Já o conceito de *skin* é diferente do de *alter*. *Skins* são uma arte diferente para um personagem que já está presente no jogo, muitas vezes sendo temática para a época do ano em que a *skin* foi lançada, variando desde conceitos mais gerais (Natal e verão) e de eventos tradicionais (como o festival de *Tanabata* japonês).

É possível ver essa estratégia nas Figuras 7 e 8 no jogo *Genshin Impact*. A Figura 7 mostra a



Figura 6.6: Imagem do personagem Arjuna, versão alternativa, do jogo *Fate/Grand Order* - fonte



Figura 6.8: Imagem da personagem Jean, com sua roupa de verão, do jogo *Genshin Impact* - fonte

arte original da personagem, enquanto a Figura 8 mostra a personagem em uma roupa de verão.



Figura 6.7: Imagem da personagem Jean do jogo *Genshin Impact* - fonte

Enquanto em alguns jogos as *skins* são disponibilizadas por meio de recompensas em eventos ou compradas com as moedas *premium* que podem ser obtidas gratuitamente (mesmo que lentamente), no caso dessa *skin* no jogo *Genshin Impact* a *skin* só podia ser obtida por meio de um tipo diferente de moeda *premium* paga, mostrando o uso do "Bônus por moedas pagas" demonstrado anteriormente na seção 4.3, e forçando o usuário a gastar dinheiro de verdade no jogo caso quisesse obter essa *skin*.

Eventos

Essa técnica se difere das outras já apresentadas no artigo, à medida que seu objetivo primário é manter os jogadores interessados em continuar

jogando com uma alta frequência. Eventos ocorrem durante um período e o conteúdo do evento é disponível somente durante a duração do evento.

Essa técnica também faz uso do *Fear of Missing Out* (FOMO), descrito na seção 4.2, o jogo mantém seus usuários interessados em participar dos eventos deixando claro que será muito difícil, senão impossível, de conseguir o que se anunciou após o fim do evento. A Figura 9 mostra essa técnica sendo utilizada.



Figura 6.9: Imagem do anúncio de um evento de *Love Live! School Idol Festival*. Note o tempo limitado do dia 7 de outubro até dia 17 de outubro - fonte

6.5 Leis contra jogos *gacha*

Neste tópico, o objetivo é demonstrar como os governos de diferentes países abordam e utilizam leis para combater as táticas utilizadas pelas empresas criadoras de jogos *gacha*. Há de se notar que toda a indústria de jogos é afetada por estas leis, e caso sejam implementadas de modo injusto, um jogo do tipo *gacha* pode ser

considerado um jogo de azar.

Um exemplo seria o de jogos que utilizam de *loot boxes* que não necessariamente se encaixam como *gacha games*, mas que foram impactados devido a criação de leis que obrigam desenvolvedores de jogos a mostrar os percentuais de *drop* (chance de obtenção) de cada item que estiver contido dentro da *loot box*.

Primeiro escândalo: contra Gacha completo (*Kompu Gacha*)

A origem dos jogos *gacha* tiveram início no Japão, no ano de 2004. Após a criação e desenvolvimento desses jogos eletrônicos por 8 anos, houve o primeiro incidente notável no mundo dos jogos *gacha*. O caso em questão foi o desenvolvimento e utilização de um método praticado pelas empresas chamado *gacha* completo (ou *Kompu gacha*, em japonês).

Gacha completo consiste em um tipo de monetização que, para obter um item ou personagem de boa qualidade e atributos, o jogador precisa antes obter um determinado número de cartas ou itens para completar um conjunto, e que por fim resulta nesse prêmio ao jogador.

A Figura 10 mostra como um *gacha* do tipo Complete *gacha* funciona.



Figura 6.10: Explicação do funcionamento de *Kompu Gacha* (ou "Gacha completo", em português) - fonte

O uso dessa tática pelas empresas resultou em um grande número de denúncias de jogadores que se sentiram extorquidos, devido a não conseguirem completar um conjunto que desejavam após terem gastado uma considerável quantia de dinheiro. Isso resultou em uma ação tomada pela agência japonesa *Consumer Affairs Agency* que veio a banir por completo essa tática das empresas.

Segundo escândalo: a favor da mostra de probabilidade

Outro escândalo aconteceu no ano de 2016, durante um evento de um jogo chamado *Gran Blue Fantasy*. O evento havia sido disponibilizado normalmente, como qualquer outro evento dentro do jogo.

No entanto, o problema veio a ocorrer quando o jogo avisou seus jogadores de que a chance para obter certo personagem haviam sido "dobradas". Assim, um grande número de jogadores gastaram uma grande quantidade de dinheiro para obter o personagem, mas não conseguiram devido a probabilidade de obter esse determinado personagem ser extremamente baixa, mesmo dobrada: uma pessoa chegou a gastar 6000 dólares ([1]) para conseguir esse personagem.

Isso chamou a atenção dos noticiários da época, e resultou na criação de diretrizes que obrigavam os criadores de jogos que utilizavam algum método de sorte a mostrar as probabilidades de conseguir qualquer item ou personagem, independente da raridade dele.

A Figura 11 mostra o resultado desse escândalo, onde a chance de se conseguir um determinado item deve sempre ser mostrada em algum lugar do jogo.

6.6 Conclusão

A partir das informações exibidas no decorrer do artigo, podemos concluir que existem diversas técnicas minuciosamente planejadas pelas empresas desenvolvedoras de jogos do tipo *gacha* para gerar lucro através de microtransações.

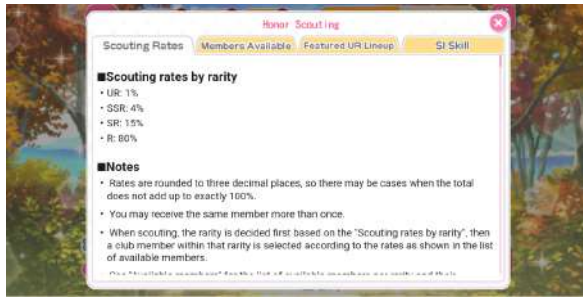


Figura 6.11: Demonstração de probabilidade para conseguir um personagem baseado na sua raridade, no jogo *Love Live! School Idol Festival* - fonte: screenshot dos participantes.

As estratégias usadas podem ser variadas, e como esses tipos de jogos atraem uma grande quantidade de jogadores, cada pequena compra feita por jogador impacta a receita da empresa, deixando evidente o quão importante são essas técnicas para esses jogos.

Um exemplo da predação funcionando até bem demais, é encontrada na referência ([2]), contando a história de uma pessoa que chegou a gastar 70 000 dólares no jogo *Fate/Grand Order*, sendo esse dinheiro boa parte de suas economias. Segundo ele, valeu a pena investir esse dinheiro porque ele foi comovido pelo jogo.

Logo, é importante deixar claro que *gachas* só conseguem ganhar a quantidade de dinheiro que ganham porque conseguem criar um ciclo de jogabilidade interessante o suficiente para manter os usuários interessados a jogar, e então investir no jogo de volta, utilizando o dinheiro ganho a partir das técnicas explicadas.

Se não por esse ciclo, as empresas conseguem a lealdade dos fãs ao criar histórias e personagens interessantes o suficiente para manter o interesse dos jogadores, que então gastam dinheiro para ver mais histórias e mais personagens.

E possível então chegar a conclusão de que, apesar da predação de empresas, o objetivo desse artigo não é inibir ou desencorajar a participação desses jogos, mas sim aconselhar o cuidado e a

atenção em como as empresas fazem pequenas coisas pra incentivar seus jogadores a gastar.

Os autores também não gostariam de desincentivar por completo o gasto de dinheiro nesses jogos, sendo uma possível maneira de demonstrar para os criadores que há interesse no jogo e que existem pessoas que gostariam de continuar jogando, apenas avisar contra gasto excessivo. Essas estratégias, embora predatórias, são totalmente válidas, e na maioria dos casos é o que sustenta a empresa desenvolvedora do jogo e a motiva à produzir mais conteúdo.

6.7 Bibliografia

- [1] O. Barder. *Japanese mobile gaming still can't shake off the spectre of exploitation*. 2016. URL: <https://www.forbes.com/sites/olliebarder/2016/04/04/japanese-mobile-gaming-still-cant-shake-off-the-spectre-of-exploitation/>.
- [2] D. Epstein and M. Inada. *Meet the man who's spent \$70000 playing a mobile game*. 2018. URL: <https://www.wsj.com/articles/meet-the-man-whos-spent-70-000-playing-a-mobile-game-1521107255>.
- [3] M. Tunholi. *Mario kart tour é o 2º maior jogo mobile da Nintendo em downloads*. 2021. URL: <https://tecnoblog.net/noticias/2021/04/22/mario-kart-tour-e-o-2o-maior-jogo-mobile-da-nintendo-em-downloads/>.
- [4] u/70sBlueAce Reddit. *How much have you spent on FEH? +Survey : r/FireEmblemHeroes*. 2021. URL: https://www.reddit.com/r/FireEmblemHeroes/comments/kpswf0/how_much_have_you_spent_on_feh_survey/.
- [5] C. Yuen-Ki. *Saving for merlin: Consumer motivation in gachabased mobile gaming*. 2019.

Capítulo 7

Introdução a Hash e seus usos

GABRIEL GOULART HOMEM

Resumo

Este documento busca realizar uma introdução geral a várias aplicações baseadas em *Hash* e suas funções.

Palavras-chave: Computação; Tabela *Hash*; Algoritmos *Hash*; Criptografia.

7.1 Introdução

Funções *Hash* estão presentes em grande parte das aplicações, e são usadas de diversas maneiras, sendo elas para organizar informações de forma rápida e eficiente, ou garantir uma maior segurança no armazenamento dos dados.

Apesar de seu unificado nome, as diversas funções *Hash* podem variar completamente umas das outras, tendo em comum apenas algumas características e seu objetivo final, de mapear uma informação, possivelmente de tamanho variável, em outra de tamanho fixo.

É o objetivo deste projeto apresentar as características principais de tais funções, e seus principais usos como tabelas *Hash* e criptografia

7.2 Conceitos básicos

Como dito anteriormente, *Hash* é uma função que converte qualquer entrada de dados com

diferentes tamanhos em uma saída de tamanho fixo. O motivo pelo qual tais funções são utilizadas, é que devido a essa característica é possível usar as saídas tanto como índices para armazenamento das informações quanto para garantir a autenticidade de informações.

A eficiência e qualidade de um algoritmo que utiliza *Hash* é diretamente relacionado a sua função *Hash*. Desse modo uma boa função *Hash* precisa atender algumas características, que são: [3]

- Valor *Hash* de saída é totalmente determinado pelos dados inseridos na função.
- A função *Hash* utiliza todos os dados recebidos.
- A função *Hash* distribui de forma uniforme os dados entre todas as possibilidades de saída *Hash*.
- A saída da função gera valores completamente diferentes de *Hash* para entradas similares.

Algumas das funções *Hash* mais conhecidas e usadas são MD5, SHA1, SHA2, NTLM, LANMAN [7] e *bcrypt*.

7.3 Tabelas *Hash*

Uma tabela de dispersão ou tabela de *Hash* (*Hash table*) é um vetor cujas posições armazenam zero, uma, ou mais chaves (e valores associados)[4], dependendo da forma como a tabela trabalha com colisões. Nessa tabela os endereços do vetor

são mapeados ao módulo das saídas da função *Hash*, fazendo com que seja possível recuperar qualquer informação de maneira direta sem precisar recorrer a buscas

A responsabilidade da função *Hash* nesse tipo de algoritmo é formar uma relação entre os dados de entrada com uma das posições da tabela, fazendo possível o algoritmo ter uma notação sempre próxima $O(1)$ para tanto pesquisa, inserção e remoção.

Entretanto, como o espaço em que as informações podem ser armazenadas é limitado e o número possível de entradas é infinito, em muitos casos se torna impossível criar uma tabela *Hash* onde dois valores nunca possuam a mesma posição, fazendo com que seja necessário que a tabela decida uma forma como tratar com essas colisões.

7.4 Tabela *Hash* perfeita

Uma tabela *Hash* perfeita consiste em uma tabela no qual todos os dados de entrada são conhecidos fazendo com que seja possível planejar uma função *Hash* no qual nunca ocorram colisões. Além disso, também se torna possível criar uma tabela *Hash* mínima perfeita no qual todas as posições são usadas.

7.5 Funções *Hash* para tabela

Como foi dito anteriormente, uma função *Hash* idealmente precisa atender às características descritas nos conceitos básicos, sendo de característica mais importante o 3 item listado, pois o único custo real da tabela é o tempo que leva para a função *Hash* ser calculada, que muitas vezes é irrelevante, e o tratamento de colisão realizado.

Uma função *Hash* que não distribui as chaves de maneira aproximadamente uniforme na tabela acaba causando com que toda a ideia por traz da tabela *Hash* perca seu sentido, pois após a colisão, mesmo que existam diversas formas de tratar desse caso, o que acaba ocorrendo é o percorrer dos índices da tabela até encontrar a posição realmente desejada.

7.6 Tamanho da tabela *Hash* e o uso de módulo

Outro aspecto das tabelas *Hash* é o uso de módulo, pois é a forma mais simples de transformar a saída da função *Hash* em uma das limitadas posições da tabela, sendo que o valor usado no módulo é o tamanho da tabela em si. Desse modo, torna-se importante a decisão sobre o número de posições.

O método mais comum para a escolha do tamanho da tabela é que a mesma possua por volta de 1,3 vezes o tamanho total de dados que se espera inserir e que o valor final seja um número primo [1], pois enquanto as posições já alocadas da tabela não influenciam a chance das outras posições serem preenchidas, quanto mais “cheia” a tabela mais comumente ocorrerá colisões e em casos com dados não randômicos, a tabela *Hash* com um número primo de posições irá produzir um melhor espalhamento das posições dos índices [2].

7.7 *Rehashing* (redimensionamento)

Em muitas aplicações o número total de dados inseridos na tabela *Hash* não irá ser fixo fazendo com que a decisão de um tamanho definitivo da tabela não seja possível. De tal modo, nessas situações ocorre a criação de um algoritmo que possa mudar o tamanho da tabela de forma dinâmica.

Alguns dos algoritmos mais comuns de *rehashing* envolvem:

- Criação de uma segunda tabela com o novo tamanho, seguido do refatoramento imediato de todas as chaves da tabela antiga para nova.
- Criação de uma segunda tabela com o novo tamanho, mas mantém a tabela antiga e ao longo do tempo, conforme novos dados são adicionados, transfere uma porção dos dados da tabela antiga para a nova.
- Criação de uma função *Hash* que após a criação de uma nova tabela todos os valores antigos continuam ocupando a mesma posição

na tabela nova, fazendo com que não seja necessário realizar o cálculo e basta copiar as chaves.

7.8 Resolução de colisões

Como dito anteriormente, uma função *Hash* ideal nunca irá gerar colisões, mas como isso não é sempre possível, acaba sendo necessário definir uma forma de lidar com esses casos. Para isso nós temos os algoritmos de resolução de colisões, e eles variam principalmente entre dois tipos que são o endereçamento aberto e o encadeamento.

O endereçamento aberto envolve o uso de alguma função extra para realocar a chave para outra posição livre da tabela, ao adotar esse tipo de encadeamento todas as operações que trabalham na tabela precisam levar em consideração esse fato, por exemplo, ao se buscar um elemento na tabela, e encontrar outro valor na posição devolvida pela função *Hash* é necessário continuar percorrendo a tabela usando a função extra, até encontrar o valor desejado ou uma posição vazia, que acontece quando o valor não está na tabela.

Dentre os algoritmos de resolução que usam endereçamento aberto alguns dos mais comuns são: [9]

- Sondagem linear (Figura 1): Ao ocorrer uma colisão aumenta o valor da posição em uma quantidade fixa, normalmente 1, esse algoritmo é eficiente devido ao alto desempenho quando usado em conjunto com o cache de CPU, sendo uma das mais usadas em tabelas *Hash*.
- Sondagem quadrática: No qual a próxima posição é escolhida a partir da soma de sucessivas saídas de polinômios quadráticos ao valor original da função *Hash* (Ex: $h(x) + 1^2$, $h(x) + 1^2 + 2^2$, etc).
- Sondagem dupla: No qual a posição é escolhida a partir da execução de uma segunda função *Hash*.

O uso de encadeamento (Figura 2) envolve a utilização de outras formas de armazenagem em

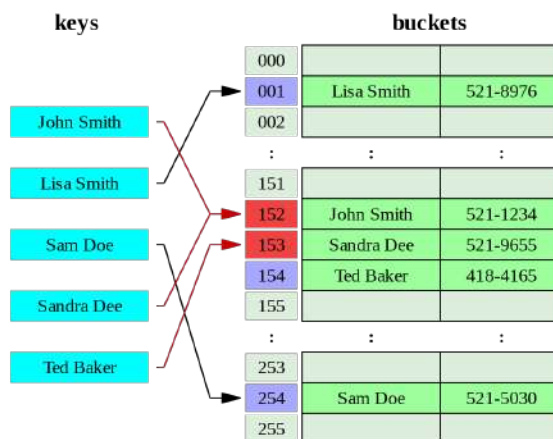


Figura 7.1: Exemplo de sondagem linear [9]

conjunto com a tabela *Hash*, por exemplo, ao ocorrer uma colisão o novo valor é alocado a uma lista encadeada, árvore ou outros tipos de algoritmos usando o valor na posição da tabela *Hash* como nó inicial para o algoritmo. A vantagem da utilização desse método é que o tamanho da tabela não mais limita o número de valores que podem ser inseridos, antes que seja necessário o *rehashing* da tabela.

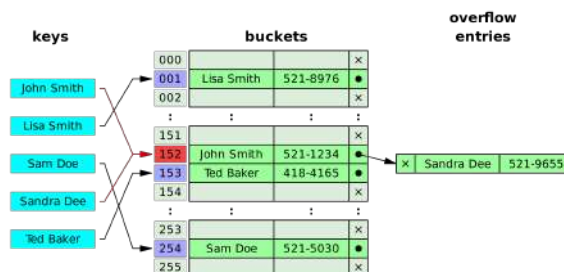


Figura 7.2: Exemplo de encadeamento [9]

7.9 Hash como criptografia

Funções *Hash* são operações matemáticas que são fáceis de executar mas extremamente difíceis de reverter [6], tornando as mesmas em excelentes escolhas como métodos de criptografia. Entre-

tanto, para que a mesma possa realizar essa tarefa é essencial que ela possua as seguintes propriedades: [5]

- *Pre-Image Resistance*: Essa propriedade indica que deve ser computacionalmente difícil de reverter essa função *Hash*.
- *Second Pre-Image Resistance*: Essa propriedade indica que ao receber uma entrada e seu *Hash*, deve ser difícil de encontrar uma segunda entrada com o mesmo *Hash*.
- *Collision Resistance (Collision free)*: Essa propriedade indica que deve ser difícil ter duas entradas diferentes, de qualquer tamanho, resultando no mesmo *Hash*.

7.10 Árvores de Merkle

Árvores de Merkle (figura 3) adotam a estrutura de uma árvore binária, cada folha contém o *Hash* de uma informação. Os nodos superiores, ou pais, armazenam o *Hash* da combinação dos *hashes* dos filhos. Se $H(A)$ e $H(B)$ são os *hashes* das informações A e B, o pai destes será $H(H(A)+H(B))$. O processo se repete até a raiz ser obtida [8].

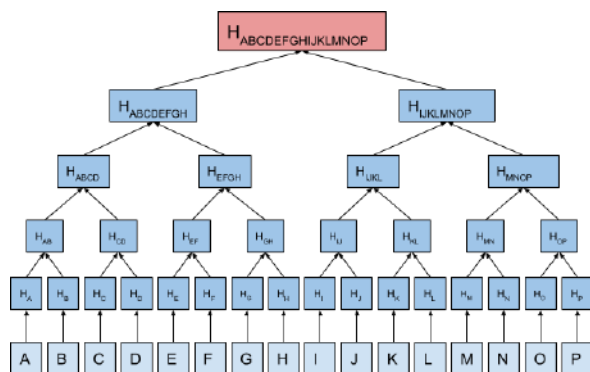


Figura 7.3: Exemplo de árvore de Merkle [8]

Tais árvores são usadas para autenticar grandes arquivos que são divididos em várias partes (como downloads usando conexão P2P), ou simplesmente quando se tem uma coleção de vários arquivos ou dados diferentes (como é o caso das transações em criptomoedas), usando apenas uma saída *Hash*, pois uma mudança em qualquer

parte dos dados utilizado irá resultar em um completamente diferente *Hash*.

7.11 Algumas aplicações de Hash

Os principais usos das tabelas *Hash* são para o armazenamento e recuperação de um grande número de informações de forma rápida, e pela busca por elementos repetidos, dado que dois elementos repetidos sempre apontaram para a mesma posição da tabela.

Além dos usos já explicados também é possível utilizar tabelas *Hash* para particionar dados (dividir dados em diferentes nodos seguindo um padrão), para esse tipo de operação ao invés de utilizar todos os dados da entrada, apenas são usadas as partes relevantes para o particionamento. Sendo que, ao utilizar as funções *Hash* para esse propósito, os dados deixam de ser divididos de forma uniforme na tabela.

Um uso extremamente comum das funções *hash*, que é usado por praticamente qualquer aplicação, é no armazenamento de senhas (Figura 4), no qual ao invés de guardar de forma direta a senha do usuário no banco de dados, a aplicação guarda a saída da função *Hash*, adicionando assim uma segunda camada de proteção sobre essa informação crítica.

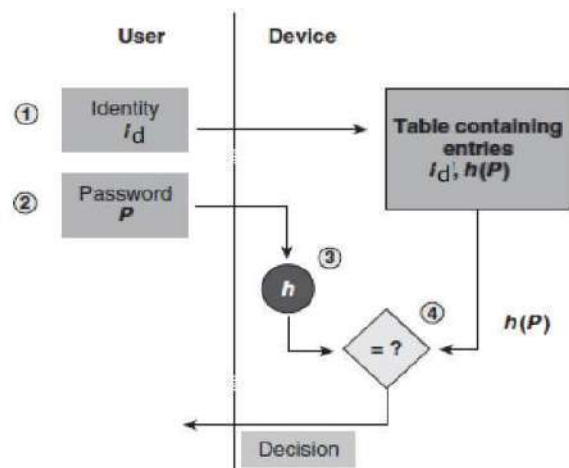


Figura 7.4: Exemplo do uso de *Hash* no armazenamento de senhas [5]

Outro uso das funções *Hash* é na realização da validação de integridade de uma informação (Figura 5), ela é realizada através da execução e armazenamento da saída da função *Hash* sobre a informação que se deseja preservar, e posteriormente na hora de verificar se a informação ainda está íntegra, basta comparar a saída *Hash* realizada anteriormente com o valor que a informação gera atualmente após passar pela mesma função *Hash*.

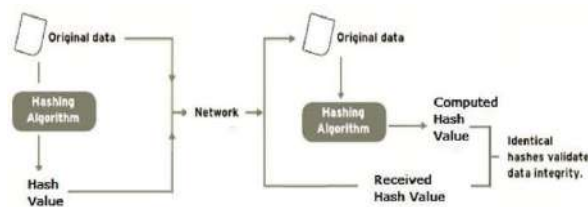


Figura 7.5: Exemplo do uso de *Hash* no armazenamento de senhas [5]

Indo mais a fundo nos dois usos mencionados na seção das árvores de Merkle, que são os downloads P2P e seu uso na *blockchain*. Temos que, nas conexões P2P os downloads efetuados são realizados através de dados disponibilizados por diversos usuários distintos dessa conexão.

Se não existir nenhum controle sobre os dados transmitidos, facilmente alguém mal intencionado conseguiria utilizar dessa conexão, as árvores de Merkle garantem essa segurança através da obtenção de uma chave raiz por um meio confiável antes de iniciar qualquer transferência, e utilizando essa chave verificar todos os dados que serão transferidos.

Já para o uso das árvores de Merkle em *blockchain* (utilizando a rede Bitcoin nesse exemplo), a sua existência é fundamental para toda a ideia e funcionamento das mesmas, sendo que as árvores de Merkle são utilizadas para organizar as transações de um bloco, com cada folha da árvore sendo uma transação.

A raiz é registrada em um campo especial no cabeçalho do bloco e, portanto, verificada durante a mineração. Durante o download da *blockchain* o integrante da rede Bitcoin reconstrói a árvore de

Merkle e comparando o *Hash* da raiz com o do cabeçalho valida a integridade das transações [8].

Outra aplicação das árvores de Merkle é em sistemas git, no qual, pela característica de que a todo momento diversos arquivos são salvos em múltiplos computadores ao mesmo tempo, existe a necessidade de se verificar por qualquer mudança para garantir que os dados fiquem consistentes em todos os computadores.

7.12 Considerações finais

Hash é uma tecnologia que está constantemente evoluindo e que pode ser usada em todos os tipos de aplicações, atendendo os mais diversos propósitos. Portanto é essencial que todo trabalhador da área de computação possua pelo menos um básico entendimento de seus usos e funcionamento.

7.13 Bibliografia

- [1] CSE 100. Hash table size. Acesso em 5 de jun. 2021. (tradução própria). URL: <https://cseweb.ucsd.edu/~kubec/cls/100/Lectures/lec16/lec16-8.html>.
- [2] Madeline Corman. Why should the length of your hash table be a prime number? Acesso em 5 de jun. 2021. (tradução própria). URL: <https://medium.com/swlh/why-should-the-length-of-your-hash-table-be-a-prime-number-760ec65a75d1>.
- [3] SPARKNOTES EDITORS. Hash function. Acesso em: 4 jun. 2021. (tradução própria). URL: <https://www.sparknotes.com/cs/searching/hashtables/section2/>.
- [4] Paulo Feofiloff. Estruturas de dados: Hashing. Acesso em 4 jun. 2021. URL: <https://www.ime.usp.br/~pf/estruturas-de-dados/aulas/st-hash.html>.
- [5] Tutorials Point. Cryptography hash functions. Acesso em 8 de jun. 2021. (tradução própria). URL: https://www.tutorialspoint.com/cryptography/cryptography_hash_functions.htm.

- [6] Shaan Ray. Cryptographic hashing. Acesso em 7 de jun. 2021. (tradução própria). URL: <https://hackernoon.com/cryptographic-hashing-c25da23609c3>.
- [7] Derrick Rountree. Hashing algorithm. Acesso em: 4 jun. 2021. URL: <https://www.sciencedirect.com/topics/computer-science/hashing-algorithm>.
- [8] Henrique Seffrin. Descomplicando o bitcoin: Árvores de merkle. Acesso em 10 de jun. 2021. URL: <https://medium.com/fs-dev/descomplicando-o-bitcoin-árvores-de-merkle-3a1b53fa4fd9>.
- [9] Wikipedia. Hash table. Acesso em 5 de jun. 2021. (tradução própria). URL: https://en.wikipedia.org/wiki/Hash_table.

Capítulo 8

Um guia introdutório extensivo, porém não exaustivo sobre Hardware e Software relativamente modernos - parte I: hardware

THIAGO DOMINGUES DA SILVA

8.1 Introdução

Veja uma versão em melhor qualidade do artigo e das imagens [neste Google Drive](#).

Antes de começar a falar sobre *hardware*, gostaria de chamar atenção à parte “relativamente modernos” do título. É bom deixar claro que não pretendo descrever ou traçar uma história inteira da evolução e desenvolvimento dos computadores, ou descrever em altíssimo detalhe pedaço por pedaço o patente de cada componente. Isso também se encaixa na parte “não exaustivo”, o objetivo deste artigo é, e somente é, guiar alguém que nunca teve contato com o *hardware* de seu PC “*Personal Computer*” de uso diário para que não fiquem perdidos na ocasional vez que precisarem ou quiserem mexer nele diretamente.

Consoles, aparatos, bugigangas também apresentam *hardware* diversificado que pode ou não ser de interesse do leitor de conhecer, temo porém que não cabe a este artigo de cobrir as partes mais especializadas. Creio que será de maior utilidade (e conveniência) cobrir as classes mais generalistas de *laptops* e *desktops*.

Não só eu não seria capaz de fazer isso com

meus recursos e conhecimentos limitados, como também seria um grande desserviço não apontar fontes já existentes que sanam esta exata curiosidade e interesse. Logo, já de antemão para os interessados recomendo o canal **Fábio Akita** do YouTube, ou mais especificamente este vídeo dele “**O Guia +Hardcore de Introdução à COMPUTAÇÃO**” se é do desejo do leitor um conteúdo mais extenso em formato de vídeo que cobre mais em detalhe as fundações de um computador, fica de aviso que o conteúdo dele não é para os fracos de coração, é *hardcore* mesmo. Para os mais aventureiros e fluentes na língua inglesa recomendo ver a série inteira do canal mencionado naquele vídeo, “**Build a 65c02-based computer from scratch**”, garanto ser um dos melhores materiais livres e disponíveis gratuitamente na *Internet* que cobre esse mesmo assunto.

Novamente, a partir do “não exaustivo” se destaca também um limite autoimposto neste artigo que é o de cobrir apenas o *hardware* que eu (o autor) conheço ou tenho/tive acesso à. Nesta nota, mencionarei inclusive que todas as fotos presentes foram batidas por mim a não ser que seja mencionado o contrário em casos a parte.

Eventualmente este fato possa preocupar o leitor de que não será de nenhuma utilidade o conteúdo do artigo, já que os modelos descritos

nele são insuficientes para abranger a vasta quantidade de componentes ou placas de circuitos existentes na atualidade. Isso seria uma verdade antes do século XXI, entretanto com a fascinante e veloz globalização da *Internet* muitos esforços foram feitos para promover a compatibilidade internacional.

Organizações como ANSI, IEEE, VESA, ISO, ITU e empresas como Microsoft, IBM, Intel, AMD, Apple entre outras são responsáveis por promover padrões ou especificações como ATX, USB, PCIE, VGA, UEFI entre outros que definem boa parte do que vemos e utilizamos nos dias de hoje. Com o tempo tudo foi padronizado de forma que *hardware* produzido em diferentes partes do mundo pudessem se comunicar sem problemas, como consequência disso apesar de existirem diversas marcas diferentes com produtos diferentes e potências ou velocidades de operação diferentes, muito tende a ser repetido em plataformas diversas com alguns pequenos ajustes.

Assim, basta somente ter uma ideia do que procurar para em pouco tempo entender a função de uma peça em sistemas distintos. Logo, processador, memória, disco rígido e outras tecnologias ubíquas serão o nosso maior foco.

Também não pretendo explicar componentes básicos de circuito (capacitor, transistor, resistor, etc.), pois seriam assunto de eletrônica, que também não é o propósito. Assim, deixarei como referência a série/playlist *ManualMaker* do canal brasileiro **Manual do Mundo**, mais especificamente, o vídeo **“Para que servem os componentes eletrônicos? #ManualMaker Aula 3, Vídeo 1”**.

Disclaimer 1: Todas as informações postadas aqui servirão apenas como conteúdo educativo e não servem de incentivo para abrir ou modificar o *hardware* de seu próprio computador ou notebook pessoal. Caso deseje fazer isso assumo o próprio risco e o autor deste artigo não se responsabiliza por qualquer dano causado à partes alheias pelo seu mau manuseio.

Disclaimer 2: Este artigo não é completo. Tentei

compilar a maior quantidade de informação que senti necessário, porém computadores são muito, mas muito complexos. Um artigo desses quando analisado por cima de uma lente pragmática se demonstrará deficiente de muita terminologia e em alguns casos sofrerá inclusive de algumas abstrações que, até na melhor perspectiva, são imprecisas (principalmente nos casos de ponte norte e ponte sul), apesar disso eu ainda acho que é um preço que se vale pagar em função da maior acessibilidade. Por isso leitor, leve tudo dito aqui com um grão de sal e se sentir vagueza desnecessária, sintase livre para entrar em contato e enviar reclamações e ou sugestões no e-mail **thiagods@ufscar.br** com destinatário Thiago Domingues da Silva.

8.2 Como as peças se parecem

Para pontificar cada peça e sua função, será útil antes mostrar fisicamente como cada uma se parece. Assim, neste capítulo existirão muitas fotos com explicações subsequentes detalhando a importância delas. Como a essência é similar, mas a forma é diferente em plataformas móveis (*laptop*) quando comparadas a plataformas de área de trabalho (*desktop*), veremos uma de cada vez.

Olhando para notebooks

Acer Aspire 5002WLCI

O *Aspire 5002WLCI* é um modelo de *notebook* lançado em 2005 pela marca Acer. As especificações do produto detalham:

- *Designed for Windows XP*
- *AMD Turion 64 Mobile ML-30 processor*
- *15.4" WXGA Acer CrystalBrite TFT LCD*
- *60GB HDD*
- *DVD/CD-RW combo*
- *512MB DDR*
- *802.11b/g wireless LAN*

Notaremos em breve que essa lista não é exaustiva do *hardware* presente nele, sendo assim é apenas “marquetado” as partes que eram consideradas mais relevantes na época:

- Um sistema operacional popular e de fácil uso
- Um processador relativamente bom
- Uma tela de 15.4 polegadas
- Um *hard drive disk* ou disco rígido de 60 gigabytes
- Capacidade de ler 'R'/'Read' e escrever 'W'/'Write' CDs e DVDs que eram extremamente comuns
- Uma boa quantidade de memória
- Capacidade de *networking* assim como definido no padrão IEEE 802.11 com protocolos 802.11b ou 802.11g

das partes responsáveis pelas funções designadas.

Assim podemos já de cara riscar a tela da lista, presente em na foto anterior, possui uma resolução de 1280 por 800 *pixels*. O seu funcionamento é de providenciar uma interface humana para o computador.

Removendo todos os parafusos e travas soltamos todas as peças e o que resta é o que se chama de chassi ou *chassis* em inglês. Mais comumente também chamado de carcaça. Ela é o esqueleto do notebook, o que dá a sua forma externa e protege as partes internas de choques físicos e outros elementos. A primeira imagem é parte de cima e a segunda a de baixo.



Vamos entender em breve o que algumas dessas coisas significam no capítulo 2 deste guia. No momento desejamos somente procurar cada uma



Na próxima imagem, temos a placa-mãe ou *motherboard* por baixo de todos os componentes e circuitos integrados. Ela é o que liga tanto fisicamente através de soquetes, quanto eletri-

camente através de rastros condutivos todos os componentes da placa. A partir dessa e da imagem seguinte exploraremos cada um dos componentes relevantes que estão ligados à placa.



Vamos ver cada componente separado, mas só a ponto de listar, temos aqui da esquerda para a direita:

- A bateria
- A ventoinha ou *cooler* que arrefece o calor transmitido pelo tubo de cobre oriundo do processador
- A unidade de disco CD/DVD
- A unidade de processamento central
- A ponte norte ou *northbridge* coberta pelo radiador
- Adaptador de rede de *Wi-Fi*
- Um chip CMOS de memória *flash* multiúso que provavelmente contém a BIOS
- Chip integrado controlador de IO
- Bateria alimentadora do CMOS
- Interface *PC Card*
- Dois pentes de memória RAM SO-DIMM de 256MB
- A ponte sul ou *southbridge*
- O disco rígido ou *hard drive disk*



Aqui temos a parte de trás da placa-mãe. As novas partes são:

- Botões genéricos juntos com o botão de ligar (*power on switch*)
- Teclado
- *Touchpad* ou almofada de toque
- Adaptador de *PC Card*
- Adaptador de rede de *Ethernet*
- Botões de mouse, incluindo botão esquerdo, botões quadricionais de *scroll* e botão direito
- *Speaker* ou alto-falante

Vamos começar pelo maior deles. O CD/DVD *drive* da *Hitachi-LG Data Storage*. É uma peça relativamente simples, só possui um botão externo que ejeta a gaveta quando pressionado. Tem a capacidade de ler e escrever CD/DVDs (se permitido pela mídia, e.g. DVD-R não pode ser escrito mais de uma vez). Normalmente usam a interface SATA ou *Serial ATA* para transmitir dados, porém este presente nas fotos é antigo e ainda usa o IDE/PATA ou *Integrated Drive Electronics/Parallel ATA*.

Um fato que será notável conforme veremos nos *notebooks* de próximas gerações é que eles tendem a ficar menores, ou melhor dizendo, com peças mais compactas e mais leves. Assim muito do que vemos nesse modelo de 2005 já estará obsoleto por volta de 2008. O *drive* por

si só já pesa 181 gramas, que é 27 gramas a mais que o disco rígido que veremos a seguir, na época que foi lançado os notebooks tinham o papel de replicar um *desktop* com a menor quantia de compromissos possível e o peso não era o foco, assim o Acer Aspire 5002WLMI pesa 2.8Kg, relativamente pesado, cheio de interfaces e componentes que eram considerados úteis para os consumidores da geração.



A segunda maior vem a seguir, a bateria de lítio. Como esperado ela serve para alimentar a placa-mãe e todos os seus circuitos integrados, ainda assim a consideraria apenas a segunda peça mais importante para o usuário depois do disco. Afinal, apesar de não ser possível usar o *notebook* desconectado sem a bateria, com uma fonte ligada na tomada não há problemas.

Tende também a ser a peça mais perigosa, pois

é a única (excluindo capacitores) que pode falhar espetacularmente com uma explosão [6], isso faz com que a compra, ou mais especificamente, a entrega internacional de notebooks seja muito mais cara do que é primeiramente esperado pois é altamente regulada [7] devido a essa delicadeza requerida ao empacotá-la.



A seguir temos o HDD "*Hard Drive Disk*". Disco "rígido"? Comparado ao que? Bem, veremos em breve a memória de acesso aleatório e ela é o que chamaríamos de memória volátil, ou seja, que não é fixa ou permanente.

O disco rígido então seria o contrário disso, uma memória fixa, que continua e persiste "*power ons*" e "*resets*", algo absolutamente necessário a não ser que seu uso ideal seja autocontido e por sessão independente. Assim sua função será salvar dados que o usuário deseja usar em futuras

sessões, normalmente onde será instalado tudo e o que o usuário comum consideraria o seu "PC" pois é ele que conterà todos os dados pessoais mais relevantes.

Tende ser mais vezes do que não a peça mais importante do computador, pois pode falecer o processador, a memória, a placa gráfica, a tela, nada importa desde que o disco rígido esteja intacto, porquanto basta o jogar em outro computador com partes funcionais e tudo fica bem [1] quando termina bem, nada é perdido e a vida continua.

Sua interface de transmissão de dados é a mesma do DVD, IDE/PATA, pois novamente se trata de uma peça antiga, um WD600 Scorpio lançado em 2004 pesando 154 gramas.



Vemos novamente a placa-mãe só que sem as peças que já demonstramos. Bem menos assusta-

dor, não? Bem, a partir de agora os componentes são mais próximos ao processador, não só fisicamente, como também eletronicamente falando. Vamos continuar tirando em ordem decrescente de tamanho, assim começaremos pela placa adaptadora de rede Wi-Fi.



Temos uma placa Wi-Fi da Broadcom que opera na frequência de 2.4Ghz com aproximadamente incríveis, fantásticos 2.5MB/s de transferência de dados (quase 400 vezes mais rápido que a *Internet* discada da década anterior).

A função desta peça você provavelmente já conhece, conectar-se com a *Internet*. Não é a única forma como veremos em breve, mas veio se tornando cada vez mais a forma padrão de se realizar a interface.

Leitores preparados podem estar exclamando um detalhe intencionalmente omitido até o momento, porém, afague os ânimos, não o deixarei de lado, apenas o apontarei posteriormente quando o tempo for mais oportuno.



A seguir vem talvez um dos itens mais antecipados. A memória de acesso aleatório! Ou mais especificamente, dois pentes RAM contendo 256MB cada um.

Pensando na fundação base de um computador, a memória só é precedida pela unidade de processamento central no quesito de importância, pois quase todas as instruções de um CPU a utilizam de alguma forma.

Sem ela é impossível fazer qualquer coisa produtiva com um computador, pois não há forma alguma de carregar o sistema operacional sem RAM e todo e qualquer software, trabalho ou processo inicializado precisa residir em algum lugar.

O leitor possivelmente esteja habituado com a BSOD "Blue Screen Of Death" no Windows quando não resta memória suficiente para serviços es-

senciais do sistema ou usa um sistema operacional baseado em Linux e percebe processos de baixa prioridade caindo a esquerda e a direita pelo mesmo motivo. Na segunda parte veremos como a memória é tratada, mas desde aqui já é possível notar sua tremenda relevância.



Chegou a vez dos chips maiores, mas antes disso é necessário remover as partes de difusão térmica. O maior deles, do qual foi desparafusado o radiador é a ponte norte ou *northbridge*. Até leitores experientes podem nunca ter ouvido falar deste componente, pois é uma relíquia do passado, sendo que recentemente as funções desempenhadas por essa peça vem sendo integradas no próprio CPU.





são gráfico recomendo o vídeo “*The world’s worst video card?*” do canal recorrente **Ben Eater**.

Para que ela serve então? Bem, até eu não tinha certeza devido sua obscuridade, mas segundo o manual [3], o M760GX da SiS ou “*Silicon Integrated Systems*” atua junto com o 302ELV logo acima para controlar os dados com relação ao vídeo, ou seja, é basicamente uma placa gráfica. Providencia aceleração gráfica 3D com suporte do Microsoft DirectX 8.0 (sim, o de 2003).



Segundo ainda outras fontes [10], [12] ele teoricamente pode produzir até 1536x1080 de saída através do VGA, que são resoluções relativamente grandes para um notebook da época.

O grande momento se aproxima, removemos agora a ventoinha com o tubo de cobre. Ela tem o propósito de expulsar o ar quente interno favorecendo a perda de calor transmitido do CPU pelo cano metálico.



O que é VGA? A sigla significa “*Video Graphics Array*”, mas de nosso interesse só será de saber como a sua interface em *hardware* se parece, para isso vide a imagem logo acima. Para quem de-seja saber as implicações do padrão de transmis-



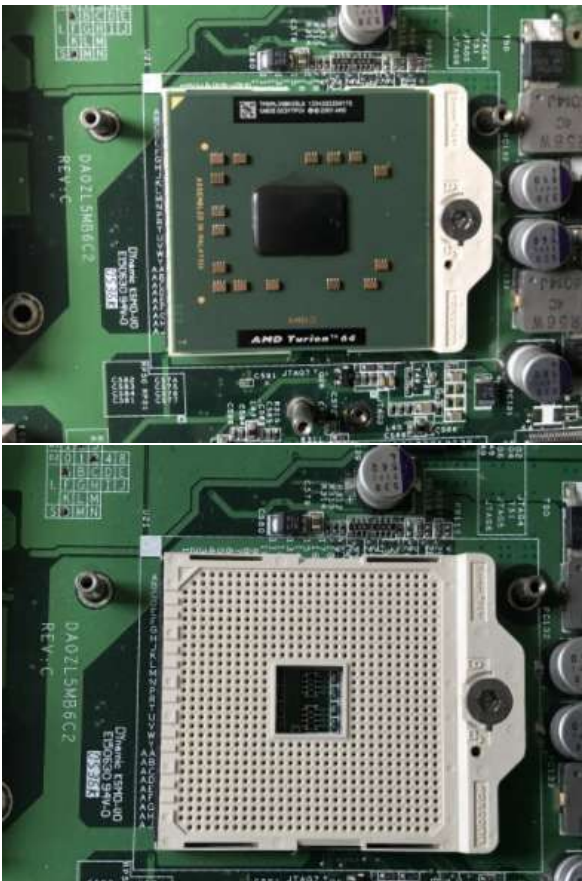


memória. Só recebe maior precedência por ser capaz de realizar instruções (mesmo que simples ou inúteis) independente da existência de RAM.

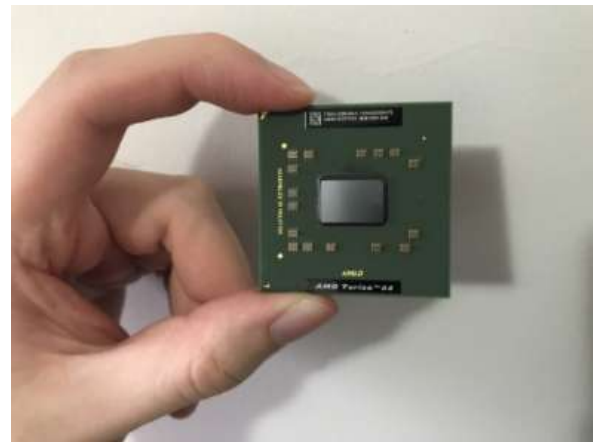
Sua função é o que a maioria das pessoas associaria a “computar”, pois é ele que calcula o resultado das operações lógicas e matemáticas. O 64 em seu nome faz referência à arquitetura x86-64 que permite tanto instruções x86 com endereçamento de 32 bits quanto a sua extensão x86-64 de 64 bits.

O que ela escondia? O CPU “*Central Processing Unit*”:

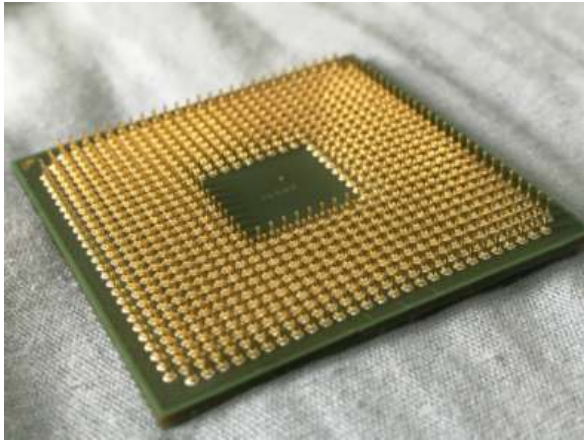
Apesar de já serem suportados $2^{64}-1$ endereços no hardware, era incomum deles serem utilizados no seu potencial máximo, não só porque memória não era barato (512MB custavam em torno de R\$400 ou um pouco acima de R\$1000 quando corrigido à inflação), mas também porque o próprio Windows XP quando o notebook foi lançado não possuía suporte para 64 bits. Assim impondo um limite máximo de 4294967295 bits ou 4GiB de memória RAM.



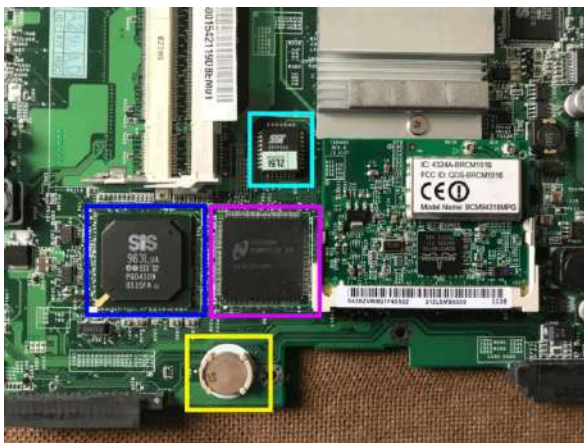
O processador AMD Turion 64 em específico. Pensando na fundação de um computador, essa seria a peça mais essencial juntamente da



Os pinos dourados são os que transmitem os dados de dentro para fora e de fora para dentro no CPU.



Antes de virarmos a placa-mãe e explorarmos os componentes do outro lado dela, acho bom chamar atenção a algumas outras peças:



Em ciano, temos o SST39VF040, um chip de memória flash multiúso. O que é memória *flash*? Não sei se eu seria capaz de descrever, mas é útil pensar em cartões SD, espetos USB “Pen Drive” ou SSDs mesmo na maior parte.

Lendo o seu manual [2] eu não vejo outro chip nesta placa-mãe que possa conter a BIOS que não seja este. Pois ele possui todas as funções que seriam esperadas de um CMOS contendo esse *firmware*. Logo, por isso que declarei antes na lista que nele provavelmente fica a BIOS deste *notebook*.

O que é a BIOS? Deixarei para o leitor investigar ou caso deseje esperar pretendo explicar

na segunda parte deste artigo (a parte de *software*).

Em amarelo, temos uma parte muitas vezes negligenciada por muitos, a bateria do *clock* interno ou RTC “*Real-Time Clock*”. Você já parou para pensar como o seu computador lembra o horário quando você liga ele novamente depois? Pois é, esta bateria toma conta de alimentar o *clock* interno para exatamente este propósito. Assim basta associar a sua função a de uma contida num relógio de pulso comum mesmo.

O *clock* normalmente fica junto de um CMOS pequeno no mesmo chip que guarda tanto a informação de tempo quanto as configurações da BIOS, desse modo ao remover a bateria, o tempo e as configurações são perdidos efetivamente restaurando-as para as de fábrica.

Por isso ao ligar um computador antigo (pode até mesmo ser um console, e.g. *PlayStation 3*), se sua bateria estiver esgotada de energia, ele impelirá o usuário a digitar o dia, o mês e o ano atual toda vez que inicializado (ou não e você involuntariamente viajará no tempo para os anos 70, 80 ou 90).

Claro, assim que conectados a *Internet*, é possível simplesmente sincronizar o relógio interno com algum presente na *Internet* com qualquer fuso horário desejado.

Em azul e rosa, temos o SiS963L e o PC97551 respectivamente. Decidi mencionar eles juntos, pois lendo os seus manuais [11], [9] vemos que conversam entre si. O 963L em específico se chama ponte sul “*southbridge*” e tem como função ministrar todos os barreamentos de comunicação com o IO “*Input/Output*” externo antes de passar os dados para a ponte norte.

Na verdade, tudo é sempre mais complicado do que parece, e nesse caso o controlador integrado PC97551 parece terceirizar quase todas as funções da ponte sul. O importante só é saber que no caso da ponte norte ela conversa e auxilia mais diretamente o CPU, enquanto a ponte sul faz interface com os barreamentos de IO como, por exemplo, o USB “*Universal Serial Bus*”, o SATA

“Serial ATA” ou PATA “Parallel ATA”.

Chegou o momento de ver os componentes do outro lado. Os três de fundo, já conhecemos, são a bateria, o CD/DVD drive e o HDD. Removeremos as maiores peças novamente para obter algo semelhante à segunda figura.



O maior de todos que engloba quase todo o quadro é o teclado, seguido do *touchpad* e outros botões. Como também as outras peças são interfaces humanas com a mesma função em essência falarei de todas em uma só fornada.

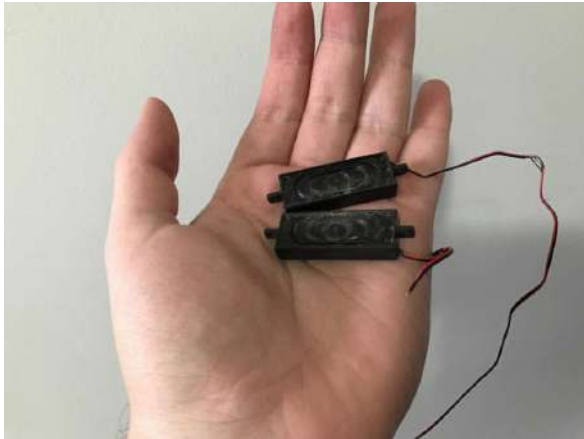
Na primeira coluna temos os botões de mouse seguidos do *touchpad* ou *trackpad*. Eles servem de substituto ao comum mouse de *desktop* e servem para mover um cursor na tela, permitindo uma interação mais direta entre o humano e a



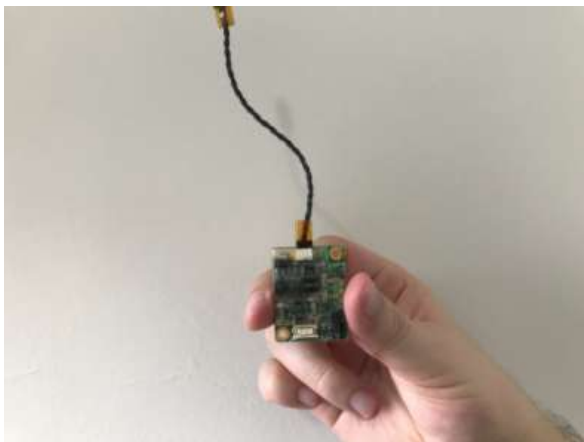
máquina. O *pad* move e os botões executam a ação, sendo o botão esquerdo e direito em cada ponta com *scroll* quadirecional no centro.

Na segunda coluna temos quatro botões de macro programáveis [4] com o quinto sendo o de ligar o *notebook*. Logo abaixo se tem o onipresente teclado, talvez a interface humana mais importante e relevante depois da tela, pois sua função permite o usuário mandar entrada direta ao computador.

Uma peça que talvez tenha passada despercebida se encontra na primeira imagem da placa-mãe de costas no canto inferior direito. É o alto-falante “*loudspeaker*”, existem dois deles neste Acer e sua função é de tocar som quando inquirido pelo sistema operacional ou software.



O próximo componente é o adaptador de rede *Ethernet*. Uma observação que merece ser feita é do fato de "*Ethernet*" ser um termo usado tanto para a interface em *software* quanto em *hardware*, porém na segunda imagem podemos ver da esquerda para a direita a interface RJ11 e RJ45 que são os verdadeiros termos quando se referido a *hardware*. Assim apesar de *Ethernet* ser usado quando se referindo a *hardware* é bom lembrar que ele na verdade se refere a interface de *software* que faz uso da placa de rede.

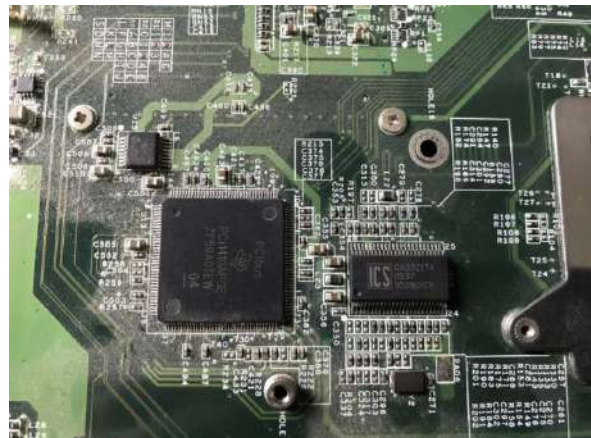


Neste caso em específico ainda se nota ainda o RJ11 que era usado para se comunicar com um modem ou fax. Com a obsolescência do segundo e a integração do primeiro pelo RJ45, essa interface simplesmente desapareceu na próxima década.



A função desta peça é semelhante à da placa Wi-Fi, a diferença é que a comunicação é através de um cabo diretamente e assim tende a ser mais fidedigna com menor interferência e velocidades de transferências maiores.

Finalmente, chegamos nos últimos componentes dos quais desejo falar.



O chip da esquerda é o PCI1410APGE do *Texas Instruments*, ele é o controlador da interface *PC Card*. Pularei a explicação desta interface por enquanto.

O da direita é o gerador de *clock* ICS952801CF. Ele é quem sinaliza quando os componentes síncronos (como o CPU, RAM, entre outros) devem agir. É o alimentado pela bateria mencionada

anteriormente quando o sistema está desligado.

Acabamos, conhecemos agora cada um dessas peças. Vamos ver a partir de agora como elas evoluíram nas próximas iterações de notebooks. Não repetirei suas funções, em vez disso apontarei somente diferenças de novas gerações.

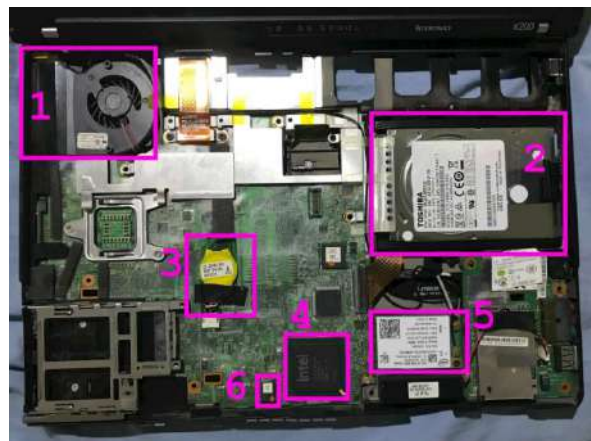


Lenovo Thinkpad X200

O X200 foi lançado em 2008, três anos depois da Lenovo adquirir a marca IBM e com isso a sua linha Thinkpad de *notebooks*.



Como veremos a seguir, muitas partes serão homogêneas, então partirei direto da placa-mãe para apontar cada componente repetido.



- A ventoinha que auxilia no resfriamento do CPU, da marca Toshiba (será chamado de *cooler* daqui em diante)
- O disco rígido, é da marca Toshiba também e possui 340GB (será chamado de HDD daqui em diante)
- A bateria que alimenta o CMOS do gerador de *clock* (será chamada de bateria CMOS daqui em diante)
- A ponte sul, um Intel 82801IEM, cuida das interfaces de IO (e.g. SATA)
- O adaptador de rede WiFi, um Intel Wifi Link 5100
- O chip CMOS que contém a BIOS, um 25X64VSFIG da *Winbond Electronics* (será chamado de BIOS apenas daqui em diante)

Chegou a hora de eu descrever o detalhe da placa de Wi-Fi que havia sido pulado anteriormente. O leitor pode ter notado dois cabos conectados nela, o que são eles afinal? Para isso, observaremos a tela.



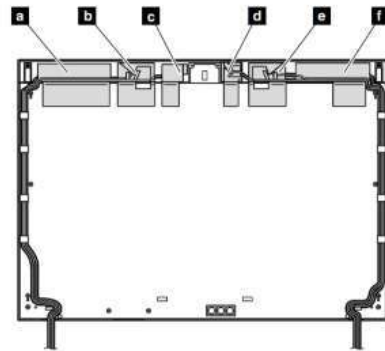
Esses cabos são as antenas de Wi-Fi que captam o sinal analógico oriundo do roteador de rede.

Como podemos ver no manual [13] oficial da Lenovo ao lado direito a antena cinza é a principal enquanto a preta é a auxiliar, mas não se engane, ambas são absolutamente necessárias e também polarizadas, tem um lugar certo para conectar cada uma.

Note como o fio segue desde o topo da tela até o circuito integrado da placa, pois de lá consegue obter uma qualidade melhor de sinal.

Obtido o sinal, ele é passado para o adaptador de rede que o digitaliza e passa para o *driver* do sistema operacional (assim como na placa *Ethernet*).

- a Wireless WAN antenna, AUX (blue)
- b Wireless LAN antenna, AUX (black)
- c Wireless USB antenna (yellow)
- d MIMO antenna (white)
- e Wireless LAN antenna, MAIN (gray)
- f Wireless WAN antenna, MAIN (red)



As diferenças:

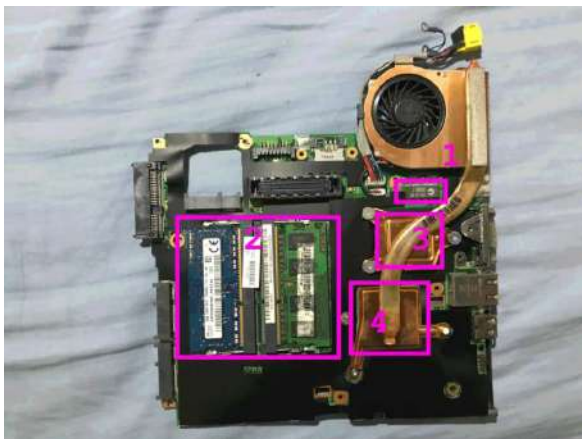




O HDD usa interface SATA em vez de PATA. Todos os futuros discos rígidos com mais de 2.5 polegadas usarão a mesma interface.

O teclado possui um ponto vermelho, que é chamado de *TrackPoint*. Serve de substituto para o *trackpad*, para mover o cursor do mouse. Também ali na parte inferior existe um sensor de digital conectado.

Virando a placa-mãe:



- O circuito integrado gerador de *clock* (chamarei só de *clock* ou RTC daqui em diante)
- Memória RAM, dois pentes, cada um contendo 2GB. Uma é da SK Hynix, a outra é da Samsung.
- CPU coberto pelo tubo de cobre. É um Intel Core 2 Duo P8700.

- Ponte norte coberta pelo cano. É um Intel GMA 4500MHD. Observação: a partir de agora a ponte norte junto com a ponte sul serão chamados de *chipset*, no caso da ponte norte estar integrada no CPU (placa gráfica integrada) somente a ponte sul será chamada de *chipset*.

Com a placa-mãe removido, retornaremos para o chassi, pois resta uma parte ali.



Essa placa genérica que estava conectada por fita "*ribbon cable*" à placa-mãe é denominada apropriadamente de placa filha ou "*daughterboard*". Nela temos, da esquerda para a direita e subindo, um alto-falante, um leitor de cartão SD, uma interface USB 2.0, uma saída de fone de ouvido, uma entrada de microfone e uma interface RJ11 (aquela do fax, lembrando que a

Ethernet é RJ45). Não foi possível ver, pois estava escondido, mas tinha um outro alto-falante em baixo do leitor de digital no canto esquerdo, totalizando 2 alto-falantes no X200.

Para amarrar a outra ponta solta que eu tinha criado, olharemos para isso:



Mas afinal, o que são essas interfaces bizarras? Pois é, o autor também nunca tinha ouvido falar.

O que fazer quando estamos perdidos? Lemos documentações, neste caso, o manual do X200. O manual porém só mostra o que o entalhe é, no caso um encaixe *ExpressCard/54*. Então o que é um *ExpressCard*? Bom, como não havia nada revelador no documento oficial resolvi procurar a enciclopédia *Thinkpad* da comunidade [4]. Também não havia esclarecimento.

Então resolvi procurar em lojas mesmo, o que quer que fosse o produto "*ExpressCard*" e obtive minhas respostas.

Ele é como se fosse uma placa de expansão. Pense no conceito de placa filha "*daughterboard*" que adiciona funcionalidades e é conectada à placa-mãe, é como se fosse isso mesmo. Tudo que se pode imaginar existe um *ExpressCard* para, seja uma placa *Ethernet* com interface RJ45, um adaptador *Wireless*, memória *flash* (como um espeto USB ou cartão SD), interface SATA, interface USB, entre várias outras. Quase tudo que poderia existir em uma *daughterboard* poderia



ser acoplado em um *ExpressCard*.



A interface do Acer Aspire 5002WLMI é de *PC Card* que é a antecessora do *ExpressCard* (assim como *PATA* é antecessor do *SATA*), então pense



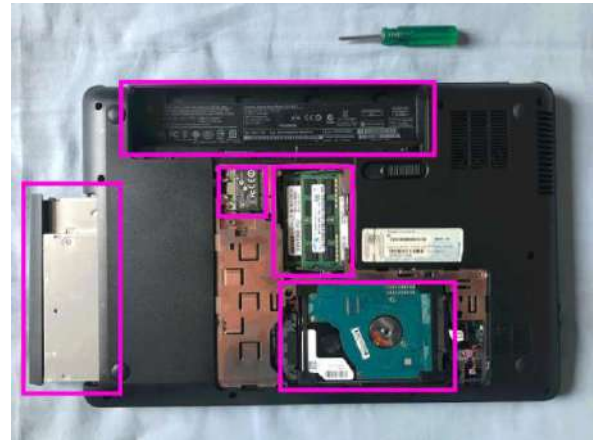
O Pavilion G4 é um modelo da HP lançado em 2013.

Sem perder tempo, já que estamos cada vez mais acostumados a esse ponto. Temos aqui, de cima para baixo, esquerda para a direita:

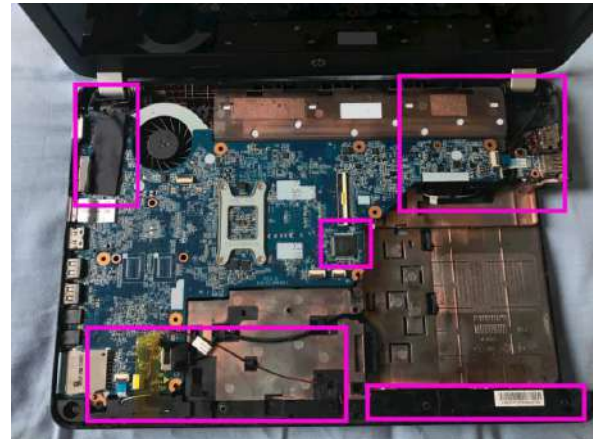
- Slot da bateria
- Drive CD/DVD
- Placa Wi-Fi
- Memória RAM
- Disco rígido

que é basicamente a mesma coisa só com uma banda larga menor.

HP Pavilion G4



Aqui dividiremos em três partes, o topo, o meio e o fundo.



No topo, temos as partes relacionadas à tela, eu não havia mencionado até então, mas o leitor

atencioso pode ter percebido que a tela se conecta na placa por um cabo fita (chamarei de *ribbon cable* daqui em diante). No lado direito temos os cabos de antena Wi-Fi que já vimos anteriormente.

No meio podemos ver a BIOS.

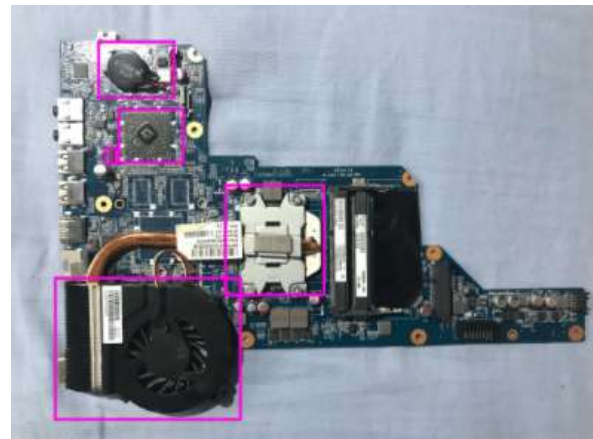
No fundo podemos ver os alto-falantes. Na verdade eles estão virados para frente, então não conseguimos enxergá-los perfeitamente, porém basta somente notar o cabo preto e vermelho que nos acompanha desde o Acer quando se trata de ligar *speakers*. Note que quase sempre teremos um par de alto-falantes, pois é o mínimo necessário para canal de áudio stereo [14].

O que restou da “tampa”?



Partes conhecidas, como o teclado, botão de ligar, *touchpad* e botões do mouse.

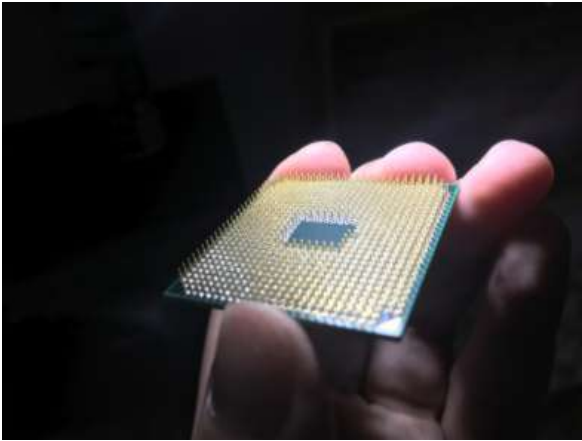
Virando a placa.



Analogamente, encontramos mais partes familiares:

- A bateria CMOS
- O RTC (*Real Time Clock*)
- A ponte sul
- O CPU que é um AMD A6-3400M
- A ventoinha

Desta vez, a ponte norte se encontra funcionalmente integrada no próprio CPU na forma da placa gráfica integrada Radeon HD 6520G. Será assim, daqui para a frente, não é mais comum dela possuir um chip dedicado.



O elemento estranho sobre a parte metálica do CPU é o que chamamos de pasta térmica, ela contribui para a melhor e mais eficiente dissipação do calor gerado no processador para os tubos de cobres ligados à ventoinha.

Neste caso, como é de um notebook velho, ela se encontra seca. Basta comprar separadamente e reaplicar.

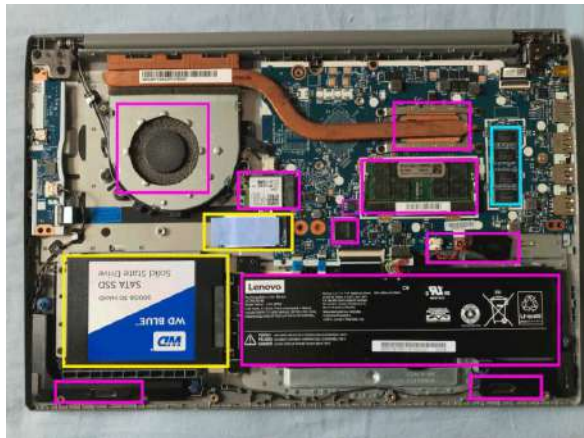


Lenovo Ideapad S145

O Lenovo Ideapad S145 foi lançado em 2019, assim fechando a nossa linha do tempo. Vemos notebooks de 2005, 2008, 2013 e agora 2019.



Será que no decorrer de quase 15 anos as coisas melhoraram muito? Veremos!

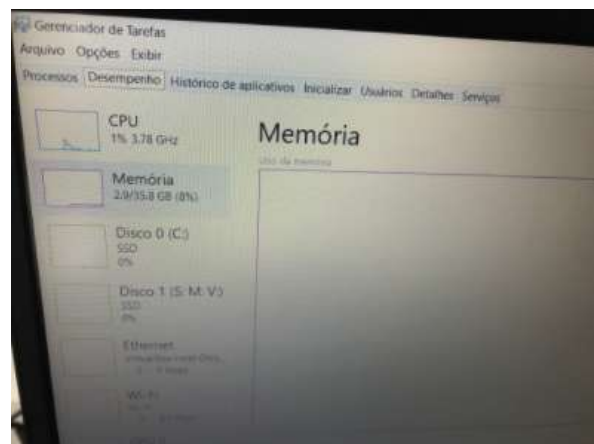


Em rosa temos o nosso café com leite:

- Ventoinha, TÜV Rheinland
- CPU, Intel Core i7-1065G7
- Adaptador de rede Wi-Fi, Intel 9462NGW
- BIOS, Winbond 25Q128JVSQ
- RAM, Micron 32GB
- Southbridge, Lenovo IT8227E-128
- Bateria CMOS
- Bateria, Lenovo 2ICP6/54/90
- Alto-falantes esquerdo e direito

Em ciano temos algo que não vimos até então, RAM integrada ou melhor dizendo, soldada à placa-mãe. Podemos notar também que diferente do Pavilion G4 e do Acer Aspire 5002WLM1 o CPU também é soldado. O que isso implica? Que não é possível remover ou trocar essas peças em específico, ou pelo menos, não sem o conhecimento de solda e equipamento necessários.

Neste caso, há 4GB de RAM da marca SMART soldadas na placa. Podemos verificar isso dando *boot* no computador sem o pente principal.



Isso será algo extremamente comum em *note-*

books mais recentes, eles costumam ser mais finos e soldar diretamente um componente pode ser uma forma de reduzir o espaço ocupado. Para a maioria dos usuários pode não ser grande coisa, mas isso acaba reduzindo a modularidade de cada peça drasticamente.

Assim, caso se queira reparar ou “upgradear” o laptop, será necessário ajuda profissional que pode ser possivelmente mais caro do que comprar um produto novo. Conhecemos isso como obsolescência programada [8].

Existem ainda no mercado hoje opções completamente modulares [15], porém tendem a ser fora do orçamento de um consumidor mediano.

Finalmente, em amarelo temos as grandes diferenças nesse notebook, SSDs “Solid State Drives”.



Qual a diferença de um HDD para um SSD?

A mais imediata seria a massa, o da primeira imagem massa um pouco menos que 40 gramas enquanto o da segunda massa 3, lembrando que os HDDs que vimos anteriormente ficam na faixa das 150 gramas. Uma simplificação seria dizer que um possui cabeças mecânicas magnéticas que leem de um disco ótico enquanto o outro age mais como um circuito de memória *flash*, mas em prol de uma explicação superior, removerei todos os freios e recomendarei esses vídeos do grandíssimo Fábio Akita: “Quebrei 3 HDs: Entendendo Armazenamento” e “Tudo que Você Queria Saber Sobre Dispositivos de Armazenamento”.

Na primeira imagem temos um SSD de interface SATA e na segunda um SSD de interface PCIe. Vale a pena mencionar aqui que a “interface” referida é a interface de software, e não de hardware. Isso pode gerar confusão, pois o encaixe de *hardware* SATA tem o mesmo nome da interface de software (diferente do RJ45 e o *Ethernet*, apesar de ambos serem comumente chamados de *Ethernet*).

O encaixe (usarei slot daqui para a frente) do SSD da primeira imagem é o SATA enquanto o slot do segundo é o M.2, isso deve permitir o leitor de notar que é possível existir um SSD de slot M.2 com a interface SATA.

E qual é a diferença? Banda larga! Ou como estamos mais acostumados a dizer, velocidade de transferência de dados. O barramento SATA III permite um limite teórico de 750MB/s enquanto o PCIe 4.0 permite até 64GB/s. Na prática, o primeiro fica mais em torno de 500MB/s enquanto o segundo fica em torno de 3GB/s, ou seja, uma razão de até 6 vezes (generalizando, pois depende altamente da marca, modelo, etc.).

Se comparado a um HDD que fica em torno de 80MB/s na prática, a diferença pode ser impressionante, quase 6x mais para um SSD SATA ou até 36x para um SSD PCIe. Em outras palavras, um Windows 10 que demora 90 segundos para iniciar em um HDD demoraria aproximadamente 15 segundos em um SSD SATA e menos de 3 segundos em um SSD PCIe (imaginando que o *boot* só dependesse do armazenamento)!

Com todos os componentes relevantes de *notebooks* mostrados, podemos encerrar esta seção e seguir para a próxima, onde veremos como os computadores de área de trabalho (os chamarei de *desktops* daqui em diante) se diferenciam dos *laptops*.

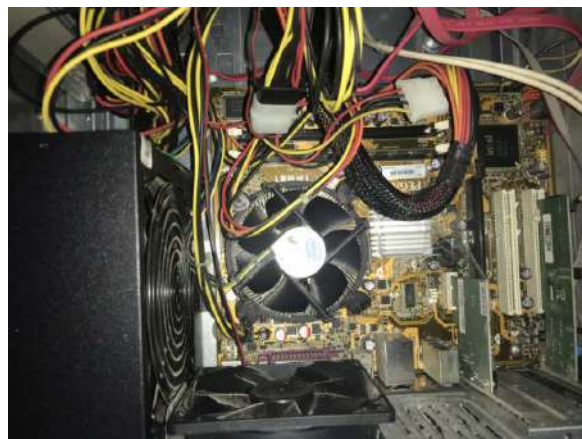
Olhando para desktops

UPD MEGAHOME DC SERIES

Este modelo é antigo e data de aproximadamente 2008, não se engane pela data no adesivo, ela representa apenas o dia da última manutenção realizada.



Abrindo o painel frontal, foi batida esta foto, o que consegue identificar nela?



Se sua resposta foi tudo, bom ou eres capaz de prever o futuro, ou tu já sabias o que os outros componentes obscuros são, o que acho ambos altamente improváveis. Mas permita-me lançar uma luz e marcar com a cor rosa as peças que você leitor razoavelmente poderia ter identificado.

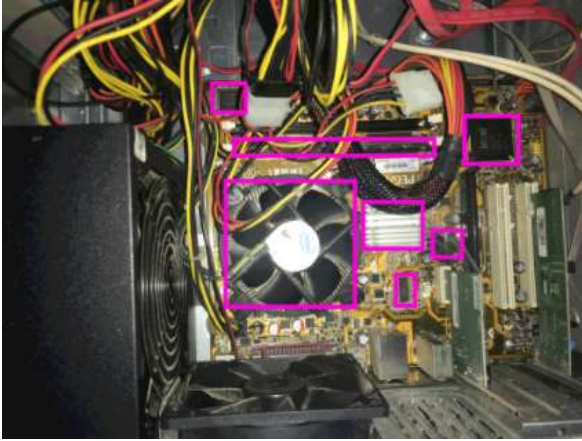
Na ordem de escrita:

- BIOS
- RAM
- *Southbridge*
- *Cooler*
- CPU (presumidamente sob o *cooler*)
- *Northbridge* (presumidamente sob o radiador)
- Bateria CMOS
- RTC

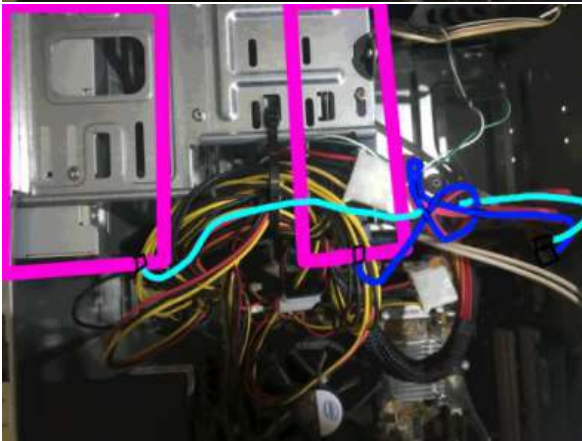
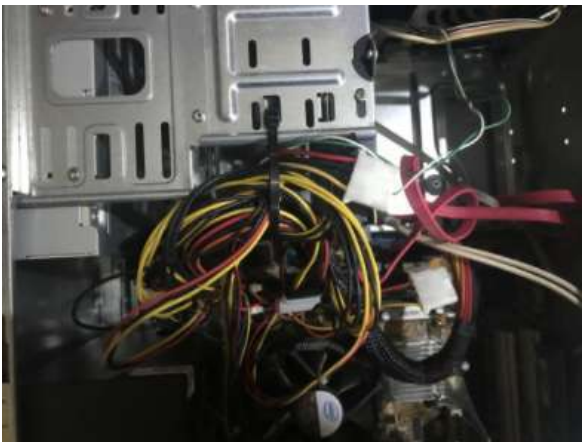
Já conseguimos reconhecer e associar algumas marcas à componentes específicos a esse ponto. Note como a BIOS é da *Winbond*, a ponte sul da Intel e o *clock* da ICS.

Ainda porém restam objetos enormes como a caixa na esquerda, o *cooler* no fundo, as placas na direita e o HDD nem se encontra no horizonte.

Resolveremos esses mistérios parte a parte.



Primeiro, revelarei uma foto mais distante.



Enfim achamos as outras peças que faltavam. Na primeira imagem o CD/DVD drive e na segunda o HDD. Acentuados em ciano e azul estão os cabos SATA que conectam eles à placa-mãe,

era possível vê-los melhor nas figuras anteriores com maior ampliação.

Como podemos ver, são gigantes! Nem consigo segurá-los direito.



Ambos são maiores que 3 polegadas (7,5 cm).

Porém se esteve prestando atenção, deve ter notado que ambos usam a mesma interface SATA que as suas versões menores de notebook usam, isso significa que são compatíveis? Sim, absolutamente sim! Claro que não seria prático tentar conectar eles em um laptop, mas o contrário seria mais viável, conectando os menores no *desktop* o fazem funcionar exatamente como o esperado.

Entre as duas placas observadas anteriormente, a primeira é a familiar interface RJ45 que corres-

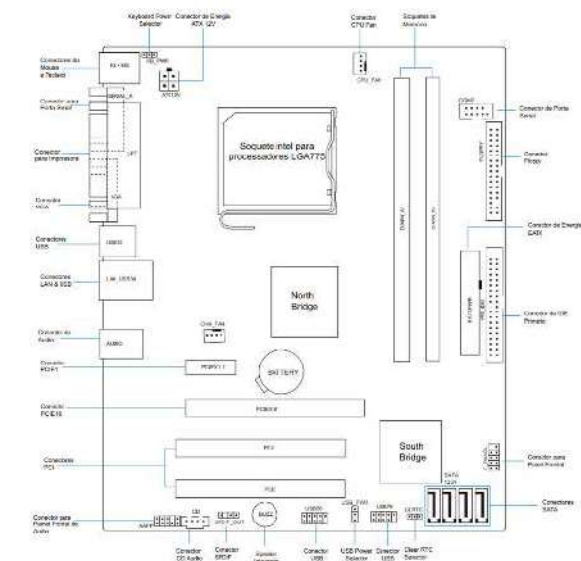
ponde à placa *Ethernet*.



Retornando à placa-mãe que é uma IPM31 da *Pegatron*, removido o emaranhado de fios, conseguimos acompanhar o manual [5] na direita. Ele confirma boa parte do que já sabíamos, além de adicionar as peças que faltavam.

E a segunda é a RJ11 que conecta ao modem telefônico, usado em *fax*.





A grande coluna da esquerda abrange o IO traseiro. Vimos ele na primeira imagem desta seção, os conectores de mouse e teclado são os antigos PS/2. Em baixo do conector de impressora estão o conector de porta serial que é praticamente obsoleto, apesar de alguns maquinários de fábricas ainda usarem, e o VGA. Por fim os conectores USB 2.0 com o RJ45 em cima e conectores de entrada e saída de áudio ou microfone na direita.

Na lacuna superior temos o “Keyboard Power Selector” que permite ajustar se o teclado terá energia mesmo quando o computador estiver desligado ou não (pode ser útil caso exista um botão de ligar o computador no próprio teclado). Logo abaixo o conector de força ATX de 12V que serve como alimentação extra para o CPU. E na direita um conector de quatro pinos para alimentar a ventoinha do processador.

Na grande coluna da direita temos o que sobrou das partes de IO. No meio o grande conector de 24 pinos é o cabo alimentador da placa-mãe, do seu lado direito está a interface IDE/PATA. O resto que sobrou se remete tudo ao IO frontal: o conector de porta serial, conector de Floppy Disk e no fundo o conector de painel frontal que inclui os diversos botões.

Na lacuna inferior temos os mesmos três

conectores de áudio que estavam no IO traseiro só que agora no IO frontal. Um conector de CD Audio, que aparentemente permitiria a tocagem direta de CDs de áudio. Um conector SPDIF que também parece ser relacionado a áudio, mas eu nunca usei. Um *speaker* integrado.

Dois conectores de USB 2.0 frontal e um “USB Power Select” que também configura se o USB será alimentado ou não enquanto o PC estiver desligado. O clear RTC faz o mesmo que remover a bateria CMOS e colocar novamente, reseta a BIOS e o *clock*. E por fim os conectores SATA que conhecemos.

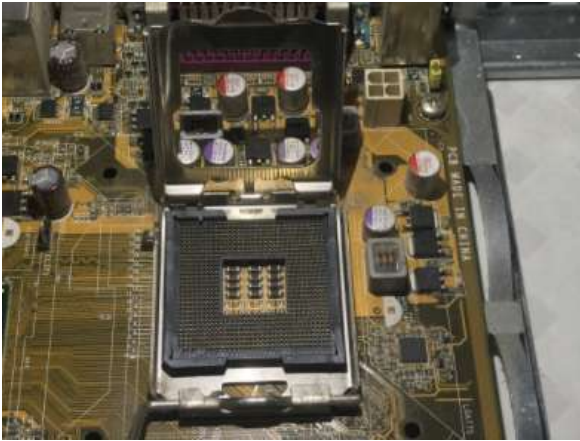
Agora só resta o centro. Já conhecemos o CPU, *northbridge*, *southbridge* e bateria CMOS. Temos outro conector de quatro pinos que dessa vez alimenta a ventoinha do chassi que é parafusada diretamente no gabinete e providencia melhor fluxo de ar interno.

Os próximos e últimos conectores são os PCIe, temos de cima para baixo: PCIe x1, PCIe x16, PCIe x8 e PCIe x8. O número após o x representa a quantidade de barramentos de dados ou “lanes” que é como os chamarei daqui para a frente. Quanto mais lanes maior a banda larga disponível, assim é ideal conectarmos placas gráficas nas lanes de maior banda.

Lembra daquela “grande caixa à esquerda”? Pois é, ela é o que chamamos de fonte de alimentação ou “Power Supply” ou mais simplesmente PSU que é como pretendo chamá-la. É ela que é conectada na tomada ou estabilizador e que fornece a energia para o computador todo.

Assim, todos os cabos de x pinos de alimentação que eu havia mencionado anteriormente se remetem a cabos que saem diretamente da fonte, que juntos formavam aquele grande emaranhado de fios. No caso do notebook era a bateria quem fazia isso.

Antes de seguir para o último computador que analisaremos neste artigo, gostaria de mostrar o CPU deste.



Soltando a trava e retirando o Intel Pentium E5200, pode-se notar que existem pinos no próprio soquete. Olhando-se para o CPU, é que se percebe:



Este CPU não possui pinos. A Intel usa um LGA "Land Grid Array" para fazer um design de CPU com pequenos círculos condutivos que faz uso dos pinos na própria placa-mãe.

Isso torna a nomenclatura de seus soquetes extremamente intuitiva: LGA (que é este design) X (que é a quantidade de pinos contidos no soquete).

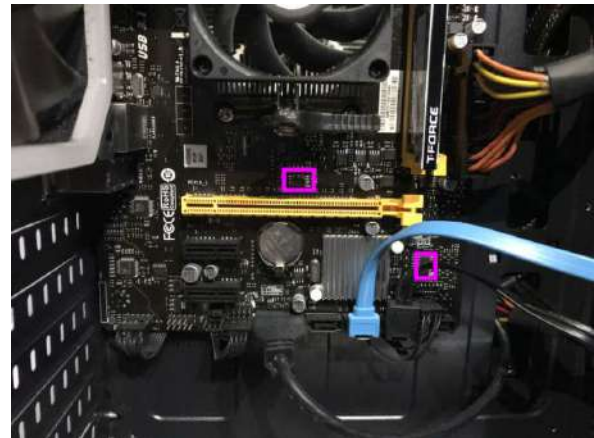
Então este soquete se chama LGA775 pelo fato dos pinos residirem na placa-mãe e de haverem 775 deles.

Prebuilt da Pichau

Prebuilt se refere a um computador *desktop* comprado já montado, ao oposto de comprar peça a peça (gabinete, placa-mãe, CPU, PSU, GPU, etc.) e montar você mesmo que se chamaria *custom built* ou simplesmente DIY "Do It Yourself".

Bom, vamos bater o olho como sempre. Quais são as coisas grandes que conseguimos ver de longe?

- Fonte, Nidus 500W
- Cooler do chassi(gabinete) e do CPU
- CPU (sob o cooler central)
- 2 slots de RAM, mas apenas 1 pente conectado da T-FORCE
- Conector de alimentação da placa de 24 pinos



- PCIe x16
- 2 PCIe x1
- Bateria CMOS
- Southbridge sob radiador, chamaremos de chipset a partir de agora
- Os vários conectores de IO com quatro SATA no final
- Um SSD SATA
- Um HDD

Aproximando mais, eu sinceramente não sei para o que estou olhando, a documentação providenciada da BIOSTAR é horrível e não ajuda em nada. Não achei os *chips* individualmente também, então dessa vez não tenho certeza.

Considerando que os *chips* do canto inferior esquerdo estão próximos ao IO de áudio, além do maior ser da Realtek eu chuto que o marcado em rosa de cima seja a BIOS.

O *chip* comprido da direita faria sentido de ser o RTC considerando a sua proximidade a tanto a bateria CMOS quanto ao oscilador de cristal ao seu lado.

Finalmente vamos instalar algo no PCIe x16, e é uma placa de vídeo RX580 da AMD.



Em baixo dela podemos ver a conexão PCIe e na segunda imagem podemos ver as suas interfaces. Temos no topo 1 DVI-D (digital

somente) dual link que permite até 2560x1600 com taxa de atualização de 60 Hz. E em baixo temos 3 *DisplayPorts* com 1 HDMI no canto direito. Usaremos elas para ligar em saídas de vídeo, sejam elas televisões, monitores ou qualquer coisa que aceite entrada de vídeo nesses formatos.

Note que como ela possui duas ventoinhas acopladas é de se esperar que receba dois cabos de quatro pinos para alimentá-las.



E é isso, creio que vimos todo tipo de interface e conector relevantes para a maioria. Deve ser o suficiente para bater o olho em algum computador e saber, pelo menos, o que boa parte dos componentes faz e como se encaixam.

8.3 Esculpindo um pouco mais certos componentes

Será explicado melhor algumas partes que ficaram vagas no primeiro capítulo.

A RAM

Nota-se uma certa nomenclatura específica quando se trata da memória, o que ela significa no fim das contas?

Suponha um pente de DDR SDRAM SO-DIMM PC3-12800, quebrando parte a parte:

DDR = “*Double Data Rate*”. Note que como RAM é um circuito síncrono, ele sempre muda

de estado quando há algum pulso de *clock*. Para RAM SDR isso seria feito somente na subida do *clock*, mas para DDR isso é feito na subida e descida do pulso.

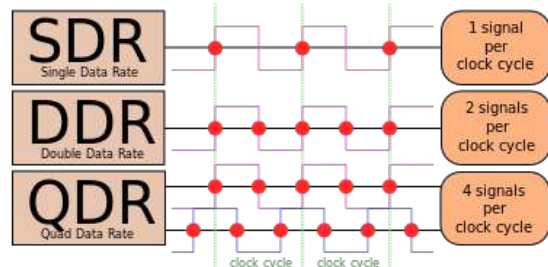


Figura 8.1: Fonte da figura: https://commons.wikimedia.org/wiki/File:SDR_DDR_QDR.svg

SDRAM = “*Synchronous Dynamic Random-Access Memory*”. Ou seja, uma memória dinâmica síncrona de acesso aleatório.

SO-DIMM = “*Small Outline Dual In-line Memory Module*”. “Pequeno Contorno”, comparado à que? Para isso vamos colocar a DIMM do primeiro *desktop* e a SO-DIMM do laptop Acer lado a lado:





Veja a diferença de tamanho, o contorno da RAM de *desktop* é muito mais comprido. Na parte de DIMM, basta notar que a memória possui *chips* dos dois lados da placa.

A parte do PC indica o número da geração. No caso PC3 indicaria memória de terceira geração. A *Infineon* da segunda imagem não possui dígito sufixado, então é da primeira geração.

Normalmente é abreviado para DDR3. O número após o hífen representa a taxa de transferência de dados em MB/s. Que seria 12800 MB/s ou 12.8GB/s.

Quando na forma DDR representa “dados” sem unidade de medida. Como em PC é medido em bytes, basta somente dividir por 8 para obter a versão abreviada. PC3-12800 então ficaria abreviado como DDR3-1600.

Adaptadores de rede Wi-Fi

Você pode ter notado que do Acer para o HP para o *Thinkpad* as placas de rede estavam em tamanhos diferentes. Veremos porque.

Existe um conceito chamado de forma fator ou “Form Factor”.



A mais antiga da esquerda se conecta na interface Mini PCI, a do meio na Mini PCIe e a da direita se conecta na mesma da do meio, porém possui uma forma fator *half* de “*half width*” ou metade da altura.

Dispositivos de armazenamento

Da mesma forma, os HDDs IDE/PATA ou SATA vinham em duas forma fatores diferentes, um de duas polegadas e meia ou 2.5” que normalmente são usados em *laptops* e outro de três polegadas e meia ou 3.5” que são usados em *desktops*.





No caso de SSDs, os de interface SATA só costumam vir no tamanho de 2.5", mas os de encaixe M.2 possuem três formas fatores: 2242, 2260 e 2280



Na primeira imagem temos o 2242 e na segunda o 2280. Os primeiros dois dígitos formam a lar-

gura do SSD em milímetros enquanto os últimos dois formam o comprimento na mesma unidade.

8.4 Conclusão

O objetivo inicialmente deste artigo era catalogar o meu aprendizado de *hardware*, destacando cada peça específica com uma foto, para que depois possa lembrar como se parece e para o que serve. Começou pequeno, com apenas um *notebook* velho e com o tempo se expandiu para vários outros e até mesmo computadores *desktop*. Ainda pretendo lançar uma revisão no futuro com mais *hardware* e com melhores explicações se possível, além de adicionar conteúdo pois alguns assuntos não foram tocados que possam ser de interesse dos leitores.

No fim, o que eu espero ter conseguido, é demonstrar a terminologia e aparência de dispositivos que se encontram dentro dos computadores que usamos no dia a dia para que possamos ter um olhar mais objetivo caso seja necessário abrir e mexer neles ou comprar novos.

8.5 Bibliografia

- [1] Windows e ou MacOS podem não gostar disso e rejeitar a máquina mesmo assim como bem quiserem, tanto quanto um órgão transplantado pode rejeitar seu hospedeiro, seja por não permitir sua utilização ou rescindindo seu direito de uso do software. Ver mais: "**Entendendo Apple, GPL e compiladores**".
- [2] 39vf040 datasheet. URL: <https://datasheetspdf.com/pdf-file/205497/SiliconStorageTechnologyInc/39VF040/1>.
- [3] Acer aspire 5002 user manual. URL: <https://www.manualslib.com/products/Acer-Aspire-5002-1735287.html>.
- [4] Expresscard slot. URL: https://www.thinkwiki.org/wiki/ExpressCard_slot.
- [5] Ipm31. URL: <https://pdfslide.tips/vembed/placa-manual-ipm-31>.

- [6] Laptop explodes and burns down office building - 986613. URL: <https://www.youtube.com/watch?v=ehcGWLOH-Js>.
- [7] Múltiplas fontes. <https://www.fedex.com/content/dam/fedex/us-united-states/services/Shipping-Lithium-Batteries-via-FedEx-Ground.pdf>
- https://www.post.japanpost.jp/int/use/restriction/restriction02_en.pdf
- <https://www.dhl.com/content/dam/dhl/local/us/dhl-ecommerce/documents/pdf/us-ecommerce-battery-shippingpolicy.pdf>
- https://www.ups.com/assets/resources/media/en_US/pack_ship_batteries.pdf.
- [8] Obsolescência programada. URL: https://pt.wikipedia.org/wiki/Obsolesc%C3%Aancia_programada.
- [9] Pc97551 datasheet. <https://datasheetpdf.com/pdf-file/604967/WinbondElectronics/PC97551/1>
- <https://datasheetpdf.com/pdf/604967/WinbondElectronics/PC97551/1>.
- [10] Product info - sis302elv. URL: <https://www.datasheetarchive.com/pdf/download.php?id=ea562429ad5915b89472d607fc5e0100bffd5b&type=P&term=-SiS302ELV>.
- [11] Product info - sis963l. URL: <https://www.datasheetarchive.com/pdf/download.php?id=2879b562e1929b91a8d2f49608792a8b8243c0&type=P&term=-SiS963L>.
- [12] Sis/xgi graphics chipsets. URL: <https://web.archive.org/web/20061103192354/http://www.winischhofer.eu/linuxsispart1.shtml>.
- [13] Thinkpad® x200 and x200s hardware maintenance manual. URL: https://download.lenovo.com/ibmdl/pub/pc/pccbbs/mobiles_pdf/43y6632_03.pdf.
- [14] Timur doumler "audio in standard c++". URL: <https://youtu.be/jNSiZqSQis4?t=935>.
- [15] The upgradeable mainboard. <https://frame.work/blog/the-upgradeable-mainboard>
<https://na.panasonic.com/us/computers-tablets-handhelds/computers/laptops/toughbook-55>.

Parte II
Projetos

Capítulo 9

ADBlock a nível de DNS

VINÍCIUS CARVALHO VENTURINI

9.1 Conceito do projeto

Este projeto tem a intenção de ser um servidor DNS (*Domain Name System*) para a rede local, no qual os dispositivos conectados terão requisições DNS para servidores considerados maliciosos, ou servidores de anúncios, bloqueadas. Assim, os usuários poderão ter uma navegação mais segura e sem anúncios aparecendo em sua tela.

9.2 Pré-requisitos e recursos utilizados

Os recursos utilizados no projeto foram a linguagem de programação **Python**, em sua versão 3, e o sistema de banco de dados **MySQL**, para armazenar as URLs que são bloqueadas no servidor.

Não é necessário ter ambos pré-instalados caso seja utilizada uma distribuição Linux, já que o *script* de instalação, feito em Shell Script, instala todas as dependências necessárias para a execução do servidor na máquina.

9.3 Instalação

Para instalar o DNS em sua máquina, clone o repositório utilizando o comando:

```
git clone https://github.com/Vinicius-Venturini/ADBlock-DNS.git
```

Após isso, dentro da pasta clonada, execute o *script* de instalação com o comando `./install.sh` (caso não tenha permissão para utilizar o *script*, utilize o comando `chmod +x install.sh`). Será pedido sua senha de usuário, insira e pressione enter.

Em um certo momento da instalação (como mostrado na imagem abaixo) será pedido uma outra senha, se você não tiver o MySQL previamente instalado apenas pressione enter (o que está sendo pedido é a senha do usuário *root* do servidor MySQL, a senha padrão é vazia, por isso não insira nada e apenas pressione enter), caso você já tivesse o MySQL instalado antes de rodar o *script* de instalação, insira sua senha de *root* e pressione enter.



```
synchronizing state of mysql.service with SysV service script with /lib/systemd/systemd-sysv-install:
Executing: /lib/systemd/systemd-sysv-install enable mysql
Enter password: |
```

9.4 Execução

Para a execução do servidor, dentro da pasta dos arquivos clonados execute o comando:

```
sudo python3 server.py
```

Logo, seu servidor já estará funcionando. Basta alterar o servidor DNS nos dispositivos que você quer que se conectem no servidor (seu celular, computador, smartTV, etc...), lembrando que esse servidor é apenas IPV4, então desative o DNS em IPV6 no dispositivo que irá se conectar caso queira que toda sua conexão passe pelo servidor.

Todas as últimas 10000 requisições, autorizadas ou bloqueadas, serão documentadas no arquivo *server.log*, assim você poderá ter um controle sobre sua rede e o que está acontecendo.

[3] Este projeto foi inspirado no projeto **PiHole**.

Caso queira adicionar novas URLs a serem bloqueadas no seu banco de dados, coloque-as no arquivo *blocklist.list* com um enter no final de cada URL, e execute o comando:

```
sudo python3 addurl.py
```

```
1 google.com
2 exemplo.com
3 www.amazon.com.br
4 |
```

Todas as URLs contidas no arquivo serão adicionadas no banco de dados, e o arquivo ficará limpo para uma próxima adição.

9.5 Bugs/problemas conhecidos

Este DNS não é 100% efetivo, visto que ele utiliza uma lista de URLs, é impossível de impedir que todos os anúncios ou sites maliciosos cheguem até o usuário final.

9.6 Repositório

O repositório com os arquivos e demais informações sobre o projeto se encontra em:

<https://github.com/Vinicius-Venturini/ADBlock-DNS>

9.7 Bibliografia

[1] A lista inicial de URLs do DNS foi feita com uma junção de listas disponibilizadas pelo projeto **The Block List Project**.

[2] Como ajuda para fazer o script do servidor DNS, foi utilizada **esta playlist** do canal **how-Code**.

Capítulo 10

Timeckoon

RYAN SAKURAI

10.1 Conceito do projeto

Projeto que visa facilitar reuniões entre os membros de uma equipe. Após informar os períodos de tempo em que os membros da equipe estão ocupados, você poderá saber quais são os períodos de tempo em que todos estão disponíveis.

10.2 Recursos utilizados

O programa foi feito usando as bibliotecas *stdio.h*, *stdlib.h* e *string.h* da linguagem C.

10.3 Passo a passo

O projeto foi criado durante o processo aprendizado de programação em C.

- Criação de um programa simples usando poucos recursos da linguagem, que tem menu simples, trabalha com um dia qualquer, usa como *output* o terminal e faz uso de comandos do terminal Windows.
- Implementação do uso de arranjos multidimensionais e strings e criação do segundo menu, que possibilita trabalhar com uma semana inteira ou parte dela.
- Implementação da manipulação de arquivos como forma de armazenamento permanente.

- Implementação de recursos mais avançados da linguagem, como *structs*, *typedef*, *enum*, funções e procedimentos.

- Aprimoramento da interface e limpeza do código.

10.4 Instalação

- Instale um compilador da linguagem C, como o MinGW.

- Baixe a pasta compactada com o arquivo *.c* do repositório no GitHub.

- Abra o terminal do Windows.

- Abra o diretório em que está localizado o arquivo *.c*, digitando *cd*, clicando e arrastando a pasta até o terminal e apertando enter.

- Compile o arquivo usando *gcc timeckoon.c -o timeckoon*.

- Execute o arquivo gerado digitando o nome do arquivo no terminal.

10.5 Execução

Ao executar o programa, menu inicial aparecerá.


```

0000000000 ddb          000
000  YBP          000
000          000
000 000 00000b,d00b,  d00b,  d0000b 000 000  d00b,  d00b, 00000b,
000 000 000 "00b d0P Y0b d00P" 000 00P d00"00b d00"000 000 00b
000 000 000 000 00000000 000 000000K 000 000 000 000 000 000
000 000 000 000 Y0b,  Y00b, 000 "00b Y00..00P Y00..00P 000 000
000 000 000 000 "Y0000 "Y0000P 000 000 "Y00P" "Y00P" 000 000

Uma calculadora de tempos disponíveis entre os membros de um grupo!

[0] Sair do programa
[1] Calcular horários disponíveis em um dia qualquer
[2] Calcular horários disponíveis em uma semana
Opcao:

```

```

000 000 000 000 000 00000000 000 000000K 000 000 000 000 000 000
000 000 000 000 000 Y0b,  Y00b, 000 "00b Y00..00P Y00..00P 000 000
000 000 000 000 000 "Y0000 "Y0000P 000 000 "Y00P" "Y00P" 000 000

Uma calculadora de tempos disponíveis entre os membros de um grupo!

[0] Sair do menu
[1] Definir início e fim do dia
[2] Inserir um período de tempo ocupado
[3] Mostrar horários disponíveis
[4] Criar arquivo com horários disponíveis
< Horarios vao de 00:00 a 24:00 >
Opcao: 3
Períodos de tempo disponíveis:
> 00:00 - 15:00
Duracao: 07:00
> 17:00 - 22:00
Duracao: 05:00
Pressione qualquer tecla para continuar. . .

```

Você poderá ir tanto para o menu que trabalha com um único dia quanto o que trabalha com vários dias de uma semana.

Para criar um arquivo com os horários, selecione "criar arquivo com horários disponíveis".

Calcular horários disponíveis em um dia qualquer

Calcular horários disponíveis em uma semana

Selecionando o menu de dia único o seguinte menu aparecerá:

Selecionando o menu da semana o seguinte menu aparecerá:

```

0000000000 ddb          000
000  YBP          000
000          000
000 000 00000b,d00b,  d00b,  d0000b 000 000  d00b,  d00b, 00000b,
000 000 000 "00b d0P Y0b d00P" 000 00P d00"00b d00"000 000 00b
000 000 000 000 00000000 000 000000K 000 000 000 000 000 000
000 000 000 000 Y0b,  Y00b, 000 "00b Y00..00P Y00..00P 000 000
000 000 000 000 "Y0000 "Y0000P 000 000 "Y00P" "Y00P" 000 000

Uma calculadora de tempos disponíveis entre os membros de um grupo!

[0] Sair do menu
[1] Definir início e fim do dia
[2] Inserir um período de tempo ocupado
[3] Mostrar horários disponíveis
[4] Criar arquivo com horários disponíveis
< Horarios vao de 00:00 a 24:00 >
Opcao:

```

```

0000000000 ddb          000
000  YBP          000
000          000
000 000 00000b,d00b,  d00b,  d0000b 000 000  d00b,  d00b, 00000b,
000 000 000 "00b d0P Y0b d00P" 000 00P d00"00b d00"000 000 00b
000 000 000 000 00000000 000 000000K 000 000 000 000 000 000
000 000 000 000 Y0b,  Y00b, 000 "00b Y00..00P Y00..00P 000 000
000 000 000 000 "Y0000 "Y0000P 000 000 "Y00P" "Y00P" 000 000

Uma calculadora de tempos disponíveis entre os membros de um grupo!

[0] Sair do menu
[1] Definir início e fim do dia
[2] Define dia da semana para inserir período de tempo
[3] Inserir um período de tempo ocupado
[4] Mostrar horários disponíveis
[5] Criar arquivo com horario disponíveis
< Horarios vao de 00:00 a 24:00 >
< Nao usar palavra com acento ou cedilha >
Opcao:

```

Neste menu você poderá definir o início (hora de acordar) e o fim (hora de dormir) do dia e horários ocupados.

Neste menu você poderá definir o início e o fim de dias específicos ou da semana inteira, definir o dia da semana cujos horários você irá inserir e inserir um período (é preciso ter um dia definido).

Para imprimir os horários livres na tela, selecione "mostrar horários disponíveis".

Para imprimir os horários livres na tela, selecione "mostrar horários disponíveis" e escolha entre imprimir os horários de dias específicos ou da semana inteira.

```
Opcao: 2
Domingo
Periodos de tempo disponiveis:
> 12:00 - 24:00
Duracao: 12:00

SEGUNDA-FEIRA
Periodos de tempo disponiveis:
> 08:00 - 22:00
Duracao: 14:00

TERCA-FEIRA
Periodos de tempo disponiveis:
> 08:00 - 22:00
Duracao: 14:00

QUARTA-FEIRA
Periodos de tempo disponiveis:
> 08:00 - 13:00
Duracao: 05:00
> 16:00 - 22:00
```

Para criar um arquivo com os horários, selecione "criar arquivo com horários disponíveis" e escolha entre horários de dias específicos ou da semana inteira.

10.6 Bugs/problemas conhecidos

Devido ao uso de comandos do terminal Windows na execução do programa, ele não funciona bem em outros sistemas operacionais.

10.7 Repositório

O repositório com os arquivos e demais informações sobre o projeto se encontra em:

<https://github.com/kkurasai/timeckoon>

Capítulo 11

Projeto TFMD - automação de cálculo de orçamentos

JOATAN MARQUES

11.1 Conceito do projeto

O TFMD é uma aplicação feita para a realização de orçamentos de diversos tipos de serviços voltado a construção. A aplicação realiza as operações necessárias e informa o usuário valor de cada comodo, e também, o valor total do orçamento.

A aplicação possui interface simples justamente para poder ser usada por pessoas que não estão muito familiarizadas com a tecnologia, que são o público alvo da aplicação.

11.2 O que é

O TFMD foi criado com o objetivo de fornecer uma aplicação de uso simples para pessoas que não possuem familiaridade com a tecnologia para que ela possa ser usada com facilidade por qualquer pessoa.

A meta principal do projeto é conseguir impactar positivamente a vida das pessoas e ser útil para todos, mas principalmente, para pessoas de comunidades carentes ou que não possuem tempo para criar um orçamento em uma planilha, por exemplo.

11.3 Como surgiu a ideia

É de conhecimento de todos a quantidade de trabalhadores autônomos no Brasil, sejam pedreiros, gesseiros, costureiras, etc. Assim como esses milhões de brasileiros meu pai também é um trabalhador autônomo e observando sua rotina de trabalho, percebi que, além de passar o dia todo trabalhando fora, ele precisava passar horas calculando orçamentos após chegar em casa.

Foi ai que surgiu a ideia do TFMD, uma aplicação que calcula automaticamente orçamentos, ajudando, não só a automatizar tarefas, mas também a poupar o tempo de pessoas que já passam o dia trabalhando e etc.

11.4 Como usar

A aplicação foi desenvolvida toda em Java e possui uma interface muito simples e o mais intuitiva possível. Para usar a aplicação basta seguir os seguintes passos

- Baixar o projeto
- Instalar o JRE (*Java Runtime Environment*), no mínimo na versão 1.8, que pode ser encontrado **aqui**
- Ir até a pasta *target* no projeto
- Executar o arquivo .jar

The image shows a web application interface with a blue theme. On the left, there is a vertical sidebar with a dark blue header and two buttons: "Novo Orçamento" and "Sair". The main content area is divided into a top section with five input fields labeled "Comodo", "Serviço", "Preço (R\$)", "Largura (m)", and "Comprimento (m)". Below these fields is a large empty blue rectangular area. At the bottom of the main area, there is a text label "Total: R\$00,00" and two buttons: "Novo Comodo" and "Resultado".

11.5 Repositório

O repositório com os arquivos e demais informações sobre o projeto se encontra em:

<https://github.com/JoatanMarques/-ProjetoTFMD>

Capítulo 12

Dúvidas IP

MAURÍCIO CÂNDIDO DE SOUZA
MICHEL RIBEIRO KOBA

12.1 Conceito do projeto

Projeto de website para dúvidas da matéria de Introdução a Programação - UFSCAR.

Este projeto foi criado com o intuito de ser um facilitador para alunos que acabaram de entrar na faculdade no modelo EaD, proporcionando uma espécie de fórum, onde alunos podem cadastrar suas dúvidas sem ter que se preocupar em quais monitores estão disponíveis e os poupando do trabalho de buscar pelo contato dos mesmos.

A sua implementação é bem simples, para os alunos é apresentado um site, no qual os alunos podem selecionar o exercício em questão e escrever sua dúvida, logo após o cadastro será emitido um código, que pode ser usado para acompanhar a situação da dúvida e comentários dos monitores.

Para os monitores, as dúvidas ficam armazenadas em um banco de dados e contém todas informações tanto do aluno quanto da dúvida, permitindo que se já analisem o problema e se tornem responsáveis pela dúvida quando estiverem disponíveis para respondê-la.

12.2 Recursos utilizados

Foram utilizadas várias tecnologias para o desenvolvimento do projeto. NodeJS para o backend, com várias bibliotecas usadas para criptografia e para o *host* (*crypto*, *express*, ...). Foi utilizado HTML e CSS para o frontend, utilizando-se da biblioteca Bootstrap para auxiliar o desenvolvimento.

12.3 Passo a Passo

No projeto, foi primeiro feita a modelagem do banco de dados, acompanhada com o cadastro backend das dúvidas. Após essa parte ser completa, foi realizado o desenvolvimento do frontend do site e testes com o auxílio de várias pessoas.

Para medidas de segurança, foi adicionado o re-Captcha, prevenindo assim ataques DDoS a plataforma. Por fim, foi adicionado um sistema de reconhecimento de estudante, com dados fornecidos pelo professor, para tornar mais simples o cadastro de dúvidas.

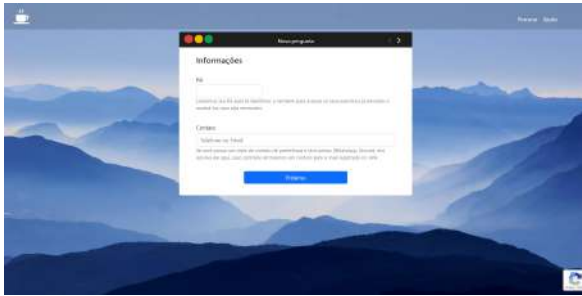
12.4 Instalação e execução

- Baixe o projeto.
- É necessário o NodeJS instalado na máquina, ele pode ser encontrado aqui.
- Também é necessária a hospedagem e a criação do banco de dados, para o funcionamento completo do projeto.

- É preciso configurar o reCaptcha para o funcionamento total do sistema, entretanto o mesmo pode ser removido para testes.
- Com tudo instalado e preparado, basta inicializar o projeto no terminal com `npm start`.
- Após tudo isso, o projeto será inicializado em `localhost:3000`

<https://github.com/ymeww/Duvidas-IP>

12.5 Imagens



12.6 Repositório

O repositório com os arquivos e demais informações sobre o projeto se encontra em:

Capítulo 13

Editor de texto para Dragon Quest II (Nintendo Switch)

VÍTOR RIBEIRO

13.1 Conceito do projeto

Este repositório contém o código para editar textos dos arquivos *stringDataEn.bin* e *stringMapEn.bin* do jogo Dragon Quest II - versão de Nintendo Switch (testado apenas com *stringDataEn.bin* do DQ II por enquanto).

Provavelmente este código funcionará para Dragon Quest I e Dragon Quest III, com alguns pequenos ajustes. Irei trabalhar com estes dois jogos também.

13.2 Pré-requisitos e recursos utilizados

A linguagem de programação C# foi utilizada para criar o *script*, juntamente com a interface do usuário (UI).

13.3 Passo a passo

- Eu estudei o arquivo *stringDataEn.bin* em sua forma hexadecimal, junto com testes conceituais no Nintendo Switch, para verificar se as mudanças feitas manualmente apareceriam no console
- Depois, criei uma ferramenta que mostra os textos do jogo formatados como uma Árvore

de Busca. Dessa forma, é possível ver a estrutura textual inteira de uma forma mais visual e editar os textos usando a fonte oficial do jogo

13.4 Execução

Para testar o programa, você precisa compilar o projeto, preferencialmente com a IDE Visual Studio.

Depois disso você precisa do arquivo *stringDataEn.bin* do jogo Dragon Quest II para Nintendo Switch (nenhum detalhe sobre como extrair este arquivo de texto do jogo ou como baixá-lo serão fornecidos).

Lista de coisas a fazer - Dragon Quest II

Feito:

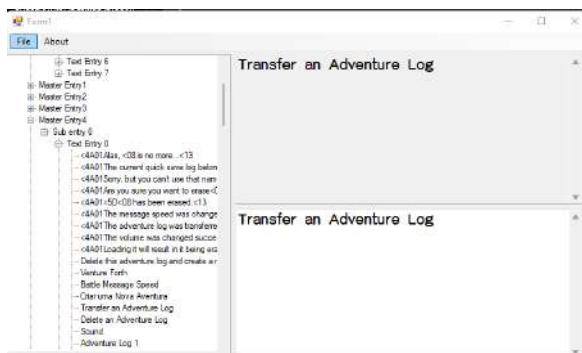
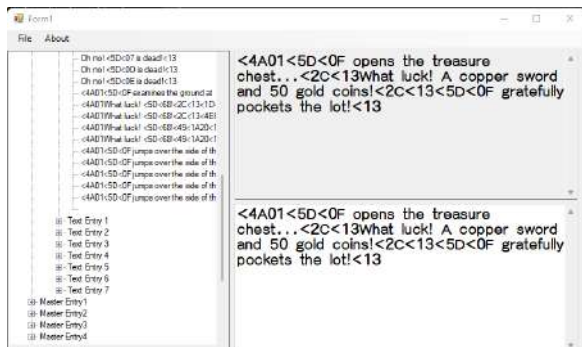
- Criar o extrator do *stringDataEn.bin*
- Incluir os dados extraídos do arquivo *stringDataEn.bin* na Árvore de Busca para dar suporte a edições do texto

Pendente:

- Criar uma função para salvar o repack do *stringDataEn.bin*
- Criar a função para extrair dados do *stringMapEn.bin*

- Incluir os dados extraídos do arquivo *stringMapEn.bin* na Árvore de Busca para dar suporte a edições do texto
- Criar uma função para salvar o repack do *stringMapEn.bin*
- Criar a função de buscar texto

13.5 Imagens



13.6 Repositório

O repositório com os arquivos e demais informações sobre o projeto se encontra em:

https://github.com/MrVtR/Dragon_Quest_II_Nintendo_Switch_Text_Editor

Capítulo 14

Wiki de Bravelly Default

VÍTOR RIBEIRO

14.1 Conceito do projeto

Este projeto consiste em construir uma Wiki para o jogo Bravelly Default do Nintendo 3DS, tendo o objetivo de fazer um projeto FullStack, utilizando ReactJS para o frontend e NodeJS para o backend.

A parte frontend do projeto é o design e funções web das páginas, além do consumo da API feita por mim, enquanto o backend fica com a construção e configuração da API, que podem ser verificadas no repositório deste [link](#).

O site criado conta com Cinco abas principais planejadas:

- **Home** - Página inicial do projeto, onde é apresentado a origem do jogo, juntamente com sua história completa. Além disso, mostramos também as capas de todas as versões regionais do jogo: Americana, Europeia e Japonesa
- **Personagens** - Página com a função de exibir os personagens principais do jogo. No momento, esta aba está em progresso, contando atualmente com os 4 personagens jogáveis, suas origens e uma função de clique para ver mais detalhes do personagem, incluindo sua história inteira dentro do jogo. Futuramente, esta aba será contemplada com todos os personagens que ajudam os principais ao decorrer da história

- **Vilões** - Página com a função de exibir todos os vilões do jogo de forma individual em uma listagem, tendo um parágrafo de origem para apresentar o vilão ao usuário e sendo possível clicar na imagem do vilão para obter mais detalhes dele, como sua história durante o jogo. Futuramente, esta aba será complementada para que todos os vilões estejam listados em ordem alfabética.
- **Locais** - Página que terá a função de exibir todos os locais pelos quais o jogador poderá passar durante a história, revelando a origem destes locais, bem como segredos que podem ser encontrados dentro deles, no momento, esta página ainda não foi iniciada e está planejada para a segunda fase do projeto
- **Organizações** - Página com a função de exibir todas as grupos de vilões que existem dentro do jogo, apresentando ao usuário um parágrafo do papel destas organizações dentro do jogo e expandindo os detalhes em uma nova página ao clicar em suas imagens, mostrando a história completa, desde a origem até a queda da organização

14.2 Lista de tarefas

Feito:

- Criação e configuração para consumo da API na página Home
- Criação e configuração para consumo da API na página Personagens

- Criação e configuração para consumo da API na página Vilões
- Criação e configuração para consumo da API na página Organizações
- Implementação de função de Mais Detalhes para ser utilizado nas páginas
- Implementação de Página de Loading, contando com tempo de *loading* baseado no consumo da API e estilização interativa com o auxílio do React Tilt

Pendente:

- Acrescentar mais dados na API para as abas de Personagens e Vilões
- Traduzir textos que estão alocados na API para que a aplicação inteira fique em Português
- Criar uma galeria de fotos individual para cada item que será mostrado na função de Mais Detalhes, utilizada por todas as páginas, com exceção da Home
- Criar página de Locais e acrescentar os dados necessários na API
- Planejar páginas de descrições de Itens e Classes do jogo

14.3 Pré-requisitos e recursos utilizados

Para o projeto ser realizado, foi utilizado as seguintes dependências para ReactJS:

- **ReactJS: 0.17.1**
- **React-router-dom: 5.2.0**
- **Node Sass: 4.14.1**
- **Axios: 0.21.1**
- **React Tilt**

14.4 Passo a passo

- Estudei como fazer um esboço do projeto baseado na **Wiki "Oficial"** do jogo
- Criei o projeto e mapeei as rotas através do React-router-dom, além de fazer a configuração dos *requests* da API com o Axios
- Criei os componentes comuns que poderiam ser utilizados no projeto todo
- Montei as páginas principais e estilei usando Sass e o React Tilt
- Efetuei um *Code Review* em cada um dos componentes e páginas para buscar *bugs* e otimizar os códigos produzidos

14.5 Instalação e execução

Para instalar as dependências do projeto localmente, utilize a instrução *yarn install* no terminal e espere pela instalação das dependências. Após isso, siga para a execução.

Para executar o programa é necessário instalar as dependências utilizadas, que estão descritas no arquivo *package.json*.

Depois disso, para executar o projeto, digite no terminal *yarn start* para rodar o código em modo desenvolvimento.

14.6 Imagens





14.7 Repositório

O repositório principal com os arquivos e demais informações sobre o projeto se encontra em:

https://github.com/MrVtR/Bravely_Default_Wiki

Outros repositórios, com partes adicionais da implementação deste, se encontram em:

- https://github.com/MrVtR/Bravely_Default_API
- https://github.com/MrVtR/Bravely_Default_Image_DB