

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE DESENHO INDUSTRIAL
TECNOLOGIA EM DESIGN GRÁFICO**

HUMBERTO WILLIAM SALMAZO

**MEMORIAL CÉU BRASILEIRO BOT: DESIGN GRÁFICO, ARTE GENERATIVA,
VISUALIZAÇÃO DE DADOS E AUTOMAÇÃO NO TWITTER**

CURITIBA

2021

HUMBERTO WILLIAM SALMAZO

**MEMORIAL CÉU BRASILEIRO BOT: DESIGN GRÁFICO, ARTE GENERATIVA,
VISUALIZAÇÃO DE DADOS E AUTOMAÇÃO NO TWITTER**

**Céu Brasileiro Memorial Bot: graphic design, generative art, data visualization
and automation in twitter**

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Tecnólogo do Curso de Tecnologia em Design Gráfico da Universidade Tecnológica Federal do Paraná, como requisito parcial à obtenção do título de Tecnólogo em Design Gráfico.

Orientador: Prof. Dr. Frederick Marinus
Constant Van Amstel

CURITIBA

2021



[4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Esta licença permite remixe, adaptação e criação a partir do trabalho, para fins não comerciais, desde que sejam atribuídos créditos ao(s) autor(es) e que licenciem as novas criações sob termos idênticos. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

HUMBERTO WILLIAM SALMAZO

**MEMORIAL CÉU BRASILEIRO BOT: DESIGN GRÁFICO, ARTE
GENERATIVA, VISUALIZAÇÃO DE DADOS E AUTOMAÇÃO NO TWITTER**

Trabalho de Conclusão de Curso de Graduação
apresentado como requisito para obtenção do título
de Tecnólogo em Design Gráfico da Universidade
Tecnológica Federal do Paraná (UTFPR).

Data de aprovação: 16 de dezembro de 2021

Frederick Marinus Constant Van Amstel
Doutor
Universidade Tecnológica Federal do Paraná

Claudia Bordin Rodrigues da Silva
Doutora
Universidade Tecnológica Federal do Paraná

Cayley Guimaraes
Doutor
Universidade Tecnológica Federal do Paraná

“A Folha de Aprovação assinada encontra-se na Coordenação do Curso”.

CURITIBA

2021

Dedico este trabalho à minha mãe, meus irmãos e
minhas cachorras, sem os quais não continuaria.

AGRADECIMENTOS

A minha mãe, pelos infinitos esforços e sacrifícios que fez, faz e fará em meu benefício, dos quais, em grande maioria, jamais saberei.

Ao meu irmão mais velho, pela eterna inspiração.

Ao meu irmão mais novo, pela constante contestação.

Ao meu amigo Roger, pela companhia, amizade e suporte durante o decorrer do curso.

Enfim, a todos que em 2018 não apertaram BRANCO nem 17.

RESUMO

Este projeto explora a combinação de conhecimentos técnicos de Design Gráfico e Desenvolvimento Web para a criação de um robô que interage com a rede social Twitter. O objetivo primário é automatizar a geração e publicação de imagens GIF que representam o número de óbitos diários por COVID-19 no Brasil como estrelas brilhando no céu em uma perspectiva humanizada de visualização desses dados. Em segunda instância, é viabilizada a possibilidade de interação com outros usuários para a construção de um memorial virtual coletivo. As etapas e tecnologias utilizadas durante o processo são descritas, acompanhadas pelas motivações pessoais e escolhas criativas. Para gerar as visualizações, foram utilizadas as tecnologias Processing, P5JS, NodeJS, PM2, MongoDB, Mongoose, Heroku, Twitter API, COVID-19 API e Linux. A conclusão deste discorre sobre a recepção do projeto, e apresenta reflexões sobre novas formas de transmitir conhecimentos técnicos e o uso da tecnologia como ferramenta de auto-afirmação e resistência.

Palavras-chave: COVID-19. Memorial. Arte Generativa. Visualização de Dados. Redes Sociais.

ABSTRACT

This project explores the combination of Graphic Design and Web Development expertise to create a robot that interacts with the social network Twitter. The primary goal is to automate the generation and publication of GIF images that represent the number of daily deaths by COVID-19 in Brazil as stars shining in the sky in a humanized perspective of visualization of this data. In a second instance, the possibility of interaction with other users for the construction of a collective virtual memorial is made possible. The steps and technologies used during the process are described, accompanied by personal motivations and creative choices. To generate the visualizations, Processing, P5JS, NodeJS, PM2, MongoDB, Mongoose, Heroku, Twitter API, COVID-19 API and Linux technologies were used. The conclusion of this paper discusses the reception of the project and presents reflections on new ways of transmitting technical knowledge and the use of technology as a tool for self-affirmation and resistance.

Keywords: COVID-19. Memorial. Generative Art. Data Visualisation. Social Media.

LISTA DE ILUSTRAÇÕES

Figura 1 - Memorial judaico de Dortmund	16
Figura 2 - Tela de título Super Mario World: Yoshi's Island	19
Figura 3 - Fundo do jogo Super Mario World 2: Yoshi's Island	20
Figura 4 - Bandeira brasileira	20
Figura 5 - Caixa alta Press Start 2P	22
Figura 6 - Fonte Mister Pixel	23
Figura 7 - Captura de tela Blender	23
Figura 8 - Imagem de perfil	24
Figura 9 - Marca com a fonte Press Start 2P	25
Figura 10 - Marca com a fonte Mister Pixel	25
Figura 11 - Exemplo de <i>Array Objects</i>	27
Figura 12 - Estrelas como <i>Array Objects</i>	29
Figura 13 - Diferença entre RGB e HEX no Photoshop	30
Figura 14 - Sequência de <i>frames</i> de estrelas	31
Figura 15 - Fundo gradiente	34
Figura 16 - Nova paleta de cores	35
Figura 17 - Exemplo de post atualizado	35
Figura 18 - Margem e moldura	37
Figura 19 - Fluxograma do robô primário	39
Figura 20 - Documentação de API do Twitter	41
Figura 21 - Documentação de API COVID-19	42
Figura 22 - Exemplo de arte resposta	46
Figura 23 - Demonstração de movimento de estrela	47
Figura 24 - Exemplo de resposta positiva	48
Figura 25 - Imagem explicativa para interagir com o robô	51

LISTA DE ABREVIATURAS, SIGLAS E ACRÔNIMOS

LISTA DE ACRÔNIMOS

API	Application Programming Interface
UI	User Interface
UX	User Experience
SSH	Secure Shell / Secure Socket Shell

SUMÁRIO

1 MOTIVAÇÃO: MINHAS CIRCUNSTÂNCIAS E MEUS SAPATOS	14
2 MONTANDO O QUEBRA-CABEÇA	16
3 PROJETO GRÁFICO	19
3.1 NOSTALGIA DOS PIXELS	19
3.2 CÉU ESTRELADO	20
3.3 TIPOGRAFIA	22
3.4 MARCA	24
4 APLICAÇÃO EM P5JS	27
4.1 TAMANHO IMPORTA	27
4.2 OBJETOS DE MATRIZ	28
4.3 A HORA DA ESTRELA	29
4.4 TODAS AS CORES DO ARCO-ÍRIS	30
4.4.1 As cores das estrelas	31
4.4.2 As cores de destaque	33
4.4.3 A cor do céu	34
4.4.4 Atualização das cores	35
4.5 ELEMENTOS TEXTUAIS	37
4.6 DIAGRAMAÇÃO	37
5 PROJETO DO ROBÔ	40
5.1 ARQUITETURA DO SISTEMA	40
5.2 NODEJS & P5JS	41
5.3 TWITTER API	42
5.4 COVID-19 API	43
5.5 PM2	43
5.6 MONGODB & MONGOOSE	44
5.7 HEROKU	44
5.8 SERVIDOR LOCAL	44
6 DESIGN DE INTERAÇÃO	46
6.1 QUAIS INFORMAÇÕES	46
6.2 NOVOS ELEMENTOS	47
6.3 FALA QUE EU TE ESCUTO	48
6.4 PESCADOR	49
6.3 FALADOR	50
7 APRESENTAÇÃO DO PROJETO	51
8 CONCLUSÕES	53
8.1 AMIGOS SÃO PARA ISSO	53
8.2 TERRENO DE GRINGO	53

8.3 UMA NOVA ESPERANÇA	54
8.4 CONSIDERAÇÕES ACADÊMICAS	55
9 EPÍLOGO	57

1 MOTIVAÇÃO: MINHAS CIRCUNSTÂNCIAS E MEUS SAPATOS

No final de setembro de 2021, quando começo a escrever esse documento, o Brasil está a menos de 5 mil para chegar a 600 mil mortes por COVID-19, isso claro, das que foram devidamente notificadas e documentadas. Esse número na realidade já deve ter sido ultrapassado.

Tive o privilégio de poder manter o distanciamento social durante esse 1 ano e meio em que já dura a pandemia, e da segurança frágil da minha bolha, digerir pedaços dessa complexa mistura de solidão, medo e instabilidade que o vírus nos apresentou - mas que já existiam desde as eleições de 2018.

Durante esses delírios de consciência, alguns pontos saltam mais a minha cabeça que outros. A polarização política, a falta de entendimento do pacto cívico, a fragilidade e a fragilização das instituições públicas e o mercenarismo do mercado privado são os mais frequentes.

Acho que todos esses pontos se constroem e se fortalecem de uma mesma dinâmica, a completa falta de empatia pela vida em pró do lucro e da manutenção do sistema, e é do intuito de tentar estimular a empatia que se desenvolve esse projeto.

Um outro aspecto importante da perspectiva desse projeto é minha imensa decepção com os rumos do design. Desde a formação tecnicista e datada que foi apresentada nesta instituição até a aplicação medíocre que o grande mercado exige.

Para além da minha frustração, há outras forças que vejo em jogo. O desenvolvimento das tecnologias tem permitido um maior acesso às ferramentas e conhecimentos da área e imagino a médio prazo a automação dos processos criativos de modo que haverá uma desvalorização de um conhecimento técnico criativo da produção de comunicação, que já é pouco reconhecido no país.

Em contrapartida temos o fortalecimento do que o mercado chama de Design de Serviços ou de UX/UI, Experiência do Usuário e Interface do Usuário, respectivamente, em tradução livre. Outro desdobramento do mercado que vejo como uma aplicação sistematizada para desumanizar e manipular pessoas através de interfaces e aplicações.

Meu ponto com estes dois parágrafos é apenas justificar porque desejo me distanciar dessas abordagens de Design, apesar de estar ciente de que dependo do mercado para sobreviver.

Do meu ponto de vista, se houver algum futuro ele terá máquinas e sistemas, e mais avançados que os atuais. Onde há máquinas há obrigatoriamente a manutenção, de modo que vejo a área de Tecnologia da Informação mais estável além de estar com alta demanda e com salários mais altos que os de Design.

Estas são as perspectivas das quais abordo o espaço deste trabalho de conclusão de curso, de modo que a pergunta que se desdobrou frente a mim foi:

Como aplicar os conhecimentos que tenho da área de Design juntos a conhecimentos que desejo desenvolver da área de TI para estimular a reflexão sobre as dinâmicas pandêmicas atuais?

2 MONTANDO O QUEBRA-CABEÇA

Durante umas das reuniões de orientação com o Prof. Fred, chegamos ao consenso de que a linguagem de programação Processing serviria de boa ponte entre as duas áreas, uma vez que ela permite a criação de imagens através de programação, possui o código aberto e tem o desenvolvimento pensado em iniciantes.

Contudo, apenas criar imagens com essa tecnologia não parecia ser relevante o suficiente de modo que não aproveitar da flexibilidade de criação possibilitada pela automação do processo me parecia uma mau uso da oportunidade além de pouco desafiante.

Realizei uma busca por meios de utilizar bases de informação já disponíveis por APIs (Interfaces de Programação de Aplicações), que em suma são padrões de interação de serviços, para utilização através de código. Encontrei um serviço gratuito que oferece números atualizados sobre COVID-19 separados por países e dias.

Esses dados me pareceram interessantes e relevantes socialmente, mas a princípio não sabia como processá-los de maneira outra da qual a mídia já tem feito durante toda a cobertura da pandemia.

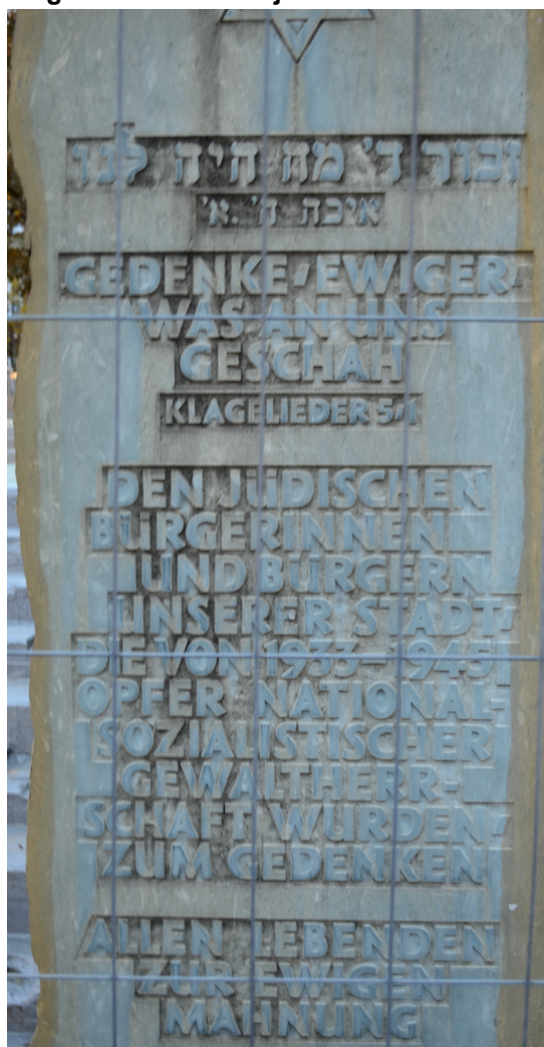
Foi enquanto com meu irmão, quando ele me mostrou fotos que havia tirado durante seu intercâmbio na Alemanha, que a ideia tomou forma. Em específico, uma foto do memorial judaico de Dortmund com os seguintes dizeres:

Lembrem-se para sempre do que nos aconteceu...

Aos cidadãos e cidadãs judaicos de nossa cidade que de 1933 a 1945 foram vítimas do regime violento nacional socialista.

A todos os vivos o eterno pesar da memória.

Figura 1 - Memorial judaico de Dortmund



Fonte: Ronnye Salmazo (2013)

Diante dessa fala que tive o “clique”. No decorrer da pandemia presenciei inúmeras vezes pessoas e empresas desprezando completamente as perdas humanas e comemorando a “vitória” sobre os desafios da exploração digital por causa do distanciamento social.

Para além da desumanidade do mercado, também aponto a minha. Ao ver diariamente as mortes documentadas pela mídia, em determinado momento a abstração dos números toma conta, torna-se fácil esquecer o imensurável valor da perda que eles escondem pela perspectiva estatística.

Dessa epifania mórbida surgiu a ideia de processar as informações da API, que eu já sabia existir, separando pelo número de mortes diárias por COVID e

apresentar esses dados de um modo menos abstrato e mais humano, através de arte e design.

Sabendo que o resultado seria um artefato digital, não apenas pela sua origem, mas pela possibilidade de alcance, retenção de custos e alinhamentos com as normativas da OMS para manter o distanciamento social, precisava definir uma plataforma digital para hospedar esse conteúdo.

Dentre as grandes redes sociais, o Twitter foi escolhido para hospedar o projeto pela facilidade que ele oferece para a interação através de programação e por ser um espaço notavelmente tóxico e fonte de propagação de notícias falsas. Ocupar esse espaço com conteúdo cívico de caráter conscientizador se alinha com os propósitos do projeto.

Assim se formou esse projeto: um memorial virtual para as vítimas de COVID-19 no Brasil, composto de um processo automatizado de geração de artes através do Processing, sua publicação no Twitter através de API, para informar e humanizar os números e propôr um estímulo a reflexão e conscientização cívica.

3 PROJETO GRÁFICO

Desde o momento em que a ideia surgiu em minha cabeça, os elementos soltos já foram se conectando, e alguns direcionamentos conceituais, artísticos e de abordagem foram tomando forma, em um processo criativo orgânico. Este capítulo apresenta essas ideias e observações que definiram os rumos do projeto antes mesmo de a caneta chegar ao papel.

3.1 NOSTALGIA DOS PIXELS

A linguagem visual definida para o projeto foi a de Pixel Arte. Composições gráficas de baixa resolução onde os pixels são evidentes.

Essa linguagem é característica dos primórdios da Computação Gráfica, amplamente explorada em jogos digitais dos anos 70 aos 90 pela grande indústria. Atualmente tida como vintage, ela é normalmente utilizada por saudosistas e desenvolvedores independentes. Esse posicionamento temporal bem marcado carrega consigo um valor nostálgico aos que tiveram contato com essas mídias durante esse período.

Outro ponto importante é a leveza dos arquivos e do processamento computacional necessário. Especialmente quando gerando artes através de código em servidores remotos onde os recursos são limitados.

Em suma, dentro do escopo do projeto, a estética de Pixel Arte une o útil ao agradável, ao facilitar o desenvolvimento do projeto e agregar valor nostálgico, e por conseguinte afetivo, que é utilizado como ferramenta de aproximação e humanização dos dados.

Como apontado anteriormente, a estética Pixelart foi amplamente utilizada por jogos digitais. De modo que as referências buscadas para o projeto focaram nesse nicho, em especial os jogos desenvolvidos pela Nintendo para o console Super Nintendo.

Dentre as pesquisas realizadas, o jogo selecionado como fonte de referência foi *Super Mario World: Yoshi's Island*, lançado em 1995 e consagrado por trazer uma estilização inédita aos gráficos.

Figura 2 - Tela de título Super Mario World 2: Yoshi's Island



Fonte: Almars Guide (2021)

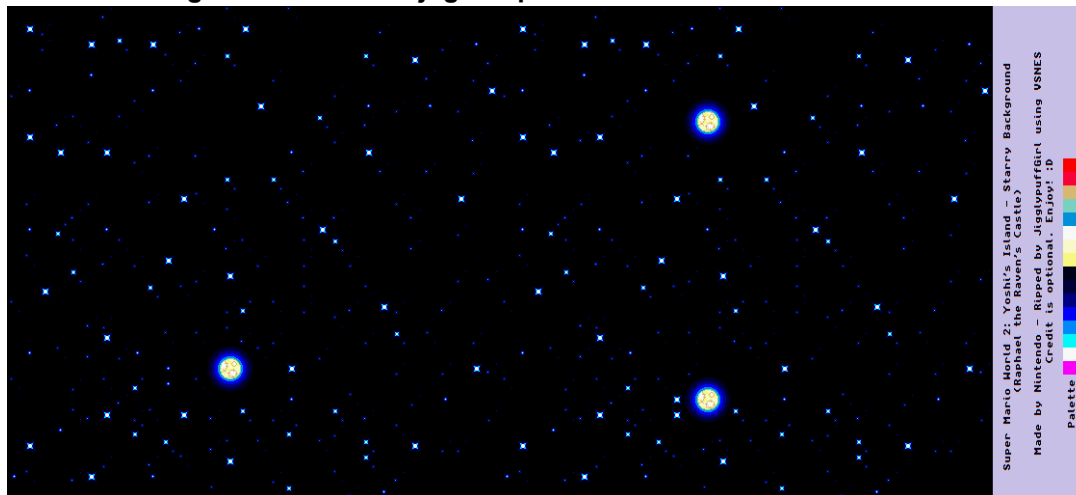
3.2 CÉU ESTRELADO

Como definido no princípio do projeto, o intuito é abordar os números de um modo mais representativo, tornando visível a quantidade massiva de mortes que aconteceram, de modo menos abstrato.

Ao juntar as referências visuais para o projeto, as imagens de fundo (backgrounds) das fases chamaram atenção. As cenas de céu noturno com estrelas brilhando ao fundo proporcionam uma certa melancolia e tranquilidade, de modo que utilizar essa composição para traduzir os números me pareceu um encaixe adequado: representar cada morte uma estrela piscando no céu. Uma vez que,

dentro do contexto coloquial, normalmente é dito às crianças em algumas regiões do Brasil, quando alguém querido nos deixa torna-se uma estrela no céu.

Figura 3 - Fundo do jogo Super Mario World 2: Yoshi's Island



Fonte: The Spriters Resource (2021)

Acredito importante apontar também o paralelo com a constelação Cruzeiro do Sul e a iconografia nacional das estrelas presentes na bandeira nacional.

Figura 4 - Bandeira brasileira



Fonte: Image PNG (2021)

Apesar da pandemia não ter feito distinções geopolíticas, seu impacto foi bem diferente ao redor do globo, reflexo das gestões públicas de cada comunidade.

No nosso caso, o vírus nos encontrou em um momento delicado, de instabilidade e polarização política, e isso impactou diretamente na reação que os brasileiros tiveram e têm com a pandemia.

O misto de estupidez, negligência e políticas públicas assassinas das lideranças federais do governo em conjunção com a burocracia da máquina pública brasileira. Ironicamente, esse grupo político que defende ferozmente a manutenção do *status quo* se apropriou da identidade nacional. Desde 2016, as cores nacionais da bandeira se tornaram sinônimo de pessoas que se alinham a essas políticas, por ganância ou ignorância (se não pelos dois), enquanto qualquer crítico é visto como comunista logo vermelho.

Acredito que o espaço cívico da identidade nacional deva ser livre e de uso coletivo, portanto utilizo alguns destes elementos com o objetivo de estimular a conscientização cívica na expectativa de retomar, ainda que minimamente, para aplicações cívicas, e não de manipulação partidária.

Por esse motivo o nome do projeto foi definido como Céu Brasileiro, e elementos da iconografia nacional foram aplicados na identidade visual, em específico as cores da bandeira.

As escolhas gráficas do projeto foram alteradas conforme novos cenários e necessidades surgiram. De modo que o processo não foi exatamente linear. Esse processo fragmentado de resolução de problemas e adaptação é reflexo de múltiplos componentes do projeto. Contudo, para manter uma organização lógica das etapas, mantive os tópicos agrupados dentro das convenções de projeto para facilitar a compreensão.

3.3 TIPOGRAFIA

Dada a escolha da estética de pixel arte para guiar o projeto, nada mais adequado que utilizar famílias tipográficas que reflitam essa estética.

A princípio, imaginei que seria mais eficiente utilizar fontes do Google Fonts, não apenas por serem gratuitas mas também por permitirem seu uso através da incorporação de links ao código, de modo a diminuir o tamanho do projeto e facilitar o desenvolvimento em geral.

Seguindo essa lógica, as únicas fontes de estilo pixelizado disponíveis são a Press Start 2P, inspirada em jogos Arcade dos anos 80, e a VT323, inspirada nas fontes de terminais. Pela correlação direta com as guias do projeto, a princípio optei pela Press Start 2P.

Figura 5 - Caixa alta Press Start 2P

**ABCDEFGHIJKLM
NOPQRSTUVWXYZ**

Fonte: F Fonts (2021)

Essa tipografia de peso único foi aplicada tanto no desenvolvimento da marca quanto no corpo do projeto.

Entretanto, durante o desenvolvimento, descobri que a biblioteca do P5Js, utilizada para gerar as imagens, requer o arquivo de fonte disponível na pasta do código. Essa necessidade me liberou do vínculo com o Google Fonts.

Outro ponto relevante foi a distribuição horizontal da fonte Press Start 2P, que dificultava o uso de parágrafos médios na arte e restringia as informações que eu poderia inserir dentro da arte.

Por esses motivos busquei outras tipografias para o projeto. Dessa busca encontrei a Sysfont – baseada no sistema Mac, a W95FA – baseada no sistema Windows 95, e a Mister Pixel – sem referências informadas. Todas fontes livres.

Entretanto, a Sysfont é corrompida quando carregada dentro do programa, a W95FA é muito fina e dificultava a leitura, de modo que foi definida a Mister Pixel para o projeto, por atender a todas as necessidades já mencionadas além de apresentar um desenho levemente mais sofisticado de pixels que acredito melhorar a legibilidade.

Figura 6 - Fonte Mister Pixel

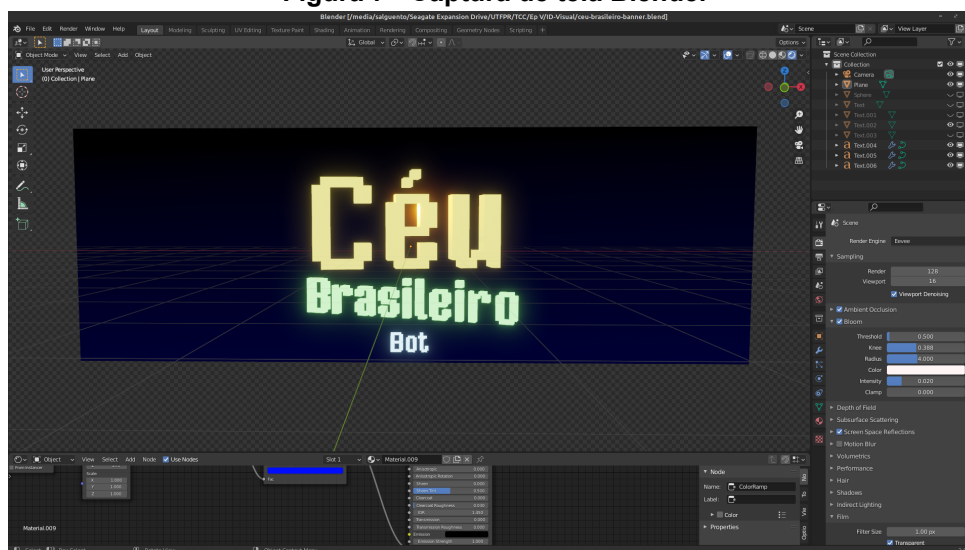
Mister Pixel 16 pt - Regular
 the quick brown fox jumps over the lazy
 THE QUICK BROWN FOX JUMPS OVER TH
 0123456789

Fonte: Fonts 2K (2021)

3.4 A MARCA

Com o propósito de distinguir o projeto dentro do Twitter foi desenvolvida uma marca, utilizando a tipografia definida e as cores da bandeira brasileira dentro do contexto do projeto. Para, exercitar outra área de interesse que tenho explorado e, dinamizar o conteúdo, a identidade foi feita em 3D, como o programa livre Blender.

Figura 7 - Captura de tela Blender



Fonte: Autoria própria

A marca é composta do nome do projeto “Céu Brasileiro Bot” e de um símbolo asterisco que representa as estrelas, utilizados como banner e foto de perfil, respectivamente.

O logotipo traz o nome do projeto escrito na fonte principal – no primeiro caso a Press Start 2P, quebrado em 3 linhas hierárquicas, cada uma com umas das 3 cores principais da bandeira em tonalidades pastel/neon para apresentar maior legibilidade contra o fundo escuro e também por analogia as estrelas reluzentes. O ícone da imagem de perfil é uma asterisco com um círculo ao centro, em cor azul clara em analogia ao azul da bandeira e ao brilho das estrelas no céu. Ao fundo há um degradê de azul para preto.

Figura 8 - Imagem de perfil



Fonte: Autoria própria

Os dizeres “Céu” e “Brasileiro” apresentam profundidade e uma ligeira rotação no eixo z – de 8 e 18 graus, respectivamente, para explorar o dinamismo da terceira dimensão. Na mesma lógica, foi utilizado uma iluminação ambiente, conhecida como HDR, que consiste de uma fotografia em 360° que simula um cenário. Neste caso, o de um pôr do sol. É do reflexo do material desse foco de iluminação que temos o brilho no começo da letra “U”. A presença desse artifício é para estimular uma percepção sensível e poética do trabalho.

O brilho presente nas duas artes também é originado por uma técnica chamada Bloom, disponível dentro do sistema de renderização Eevee, nativo do Blender. Essa técnica possibilita reflexos desfocados, similares ao neon.

Visto que o ambiente da criação é mais complexo que o ambiente 2D convencional, as cores estão suscetíveis as dinâmicas espaciais que as circundam. Logo, apesar de poder ser apontados os valores absolutos das cores utilizadas nos materiais de cada objeto, o valor real das cores não é o mesmo.

Dado isso, seguem as descrições técnicas de cada material na cena:

Como pontuado anteriormente, no trecho sobre a tipografia, durante o desenvolvimento houveram alterações que impactaram diversos aspectos. A identidade visual foi um dos aspectos que sofreu impacto da mudança de fonte, atualizando da Press Start 2P para a Mister Pixel, como demonstrado abaixo:

Figura 9 - Marca com a fonte Press Start 2P



Fonte: Autoria própria

Figura 10 - Marca com a fonte Mister Pixel



Fonte: Autoria própria

Acredito que tal atualização trouxe os mesmos benefícios de melhor relação espacial e legibilidade dos que apresentados no corpo do projeto, e que serão exemplificados mais a frente.

4 APLICAÇÃO EM P5JS

Antes de apresentar o processo de desenvolvimento, gostaria de apresentar a ferramenta utilizada. O *P5JS* é uma tradução de uma linguagem de programação chamada *Processing* para uma biblioteca em *JavaScript*, uma linguagem que pode ser executada diretamente em navegadores da web quanto em servidores web. Em ambos os casos o foco de aplicação é o desenvolvimento de codificação criativa acessível e gratuita, com uma baixa curva de aprendizado, para ser utilizada por artistas, designers e educadores.

A escolha dessa tecnologia se deu, para além dos benefícios que lhe são próprios, por ser uma tradução em *JavaScript*, uma linguagem de programação fundamental para o desenvolvimento *Web*, e portanto com um vasto catálogo de tutoriais e cursos disponíveis bem como um amplo campo de aplicações profissionais, além de permitir a interação com as APIs e a manipulação dos dados retornados, como será explicado mais detalhadamente no capítulo seguinte.

4.1 TAMANHO IMPORTA

A primeira configuração a ser determinada quando se inicia um projeto em *P5JS* é o tamanho do *Canvas* (Tela). Essa escolha influencia drasticamente o consumo de recursos computacionais necessários para computar os outros parâmetros que serão definidos, e impacta diretamente o tempo de renderização da arte.

Outra variável a ser considerada são as guias de tamanho de mídias de cada rede social. Cada plataforma tem seus próprios padrões e limites para cada tipo de mídia. No caso, o Twitter, tinha até este ano, um algoritmo que cortava automaticamente o enquadramento de uma imagem quando apresentado no *feed*. O que requer que a pessoa utilizando a plataforma clicasse na imagem para abri-la em suas dimensões reais. Felizmente essa função foi desativada, de modo que essa restrição não impactou a definição do tamanho da tela neste caso.

Tendo em vista a padronização de imagens em formato quadrado (1:1) popularizado pelo Instagram e amplamente adotado na internet, essas dimensões

também foram adotadas para o projeto. Contudo, normalmente a resolução indicada atualmente é de 1080 por 1080 *pixels*, para garantir uma boa resolução em múltiplos dispositivos. Adotar esse tamanho exigiria um processamento substancial, quando processado pelo *P5JS*, de modo que optei por uma resolução ligeiramente inferior, o que reforçaria a estética *pixelizada* adotada anteriormente.

4.2 OBJETOS DE MATRIZ

Uma vez definido o tamanho do *canvas* (tela), busquei métodos para criar elementos individuais e controláveis dentro desse espaço. Felizmente, dentro da documentação do *P5JS* existem exemplos de múltiplas aplicações possíveis. Uma delas o *Array Objects* (objetos de matriz). Ele permite criar elementos individuais espaçados igualmente pelo tamanho da tela e que atualizam sua posição a cada *frame* baseado em uma função de aleatoriedade. O resultado são pontos que se movem independentes uns aos outros dentro do espaço especificado da tela.

Figura 11 - Exemplo de *Array Objects*



Fonte: *P5JS* (2021)

Dentro desse método precisamos de um valor para *unit* que será quanto espaço vertical e horizontal cada elemento terá da tela. Esse valor é importante porque define quantos elementos apareceram na tela. Por exemplo: em uma tela de tamanho 800 x 800, com *unit* = 100, teremos 64 elementos individuais, resultado de 8 colunas e 8 linhas, formando uma grade.

Essa etapa é calculada dividindo a altura e o comprimento da tela, individualmente, pelo valor de *unit*. O produto dessas duas equações gera o número de objetos mostrado na tela.

Como o número de objetos é variável (o número de mortes diárias por COVID-19), foi preciso encontrar uma equação que permitisse uma definição automática para o valor de *unit*. Utilizando a calculadora de equações *Symbolab*, resolvi a equação. Sendo *C* o número total de objetos, *H* a altura, *W* a largura e *U* a variável que busco, temos: $C=(H/U)*(W/U)$, logo $U=\sqrt{HW/C}$.

Aplicando essa fórmula é possível utilizar o código fonte do exemplo dado na página do *P5JS* para gerar qualquer número de objetos igualmente espaçados em qualquer tamanho determinado de tela.

4.3 A HORA DA ESTRELA

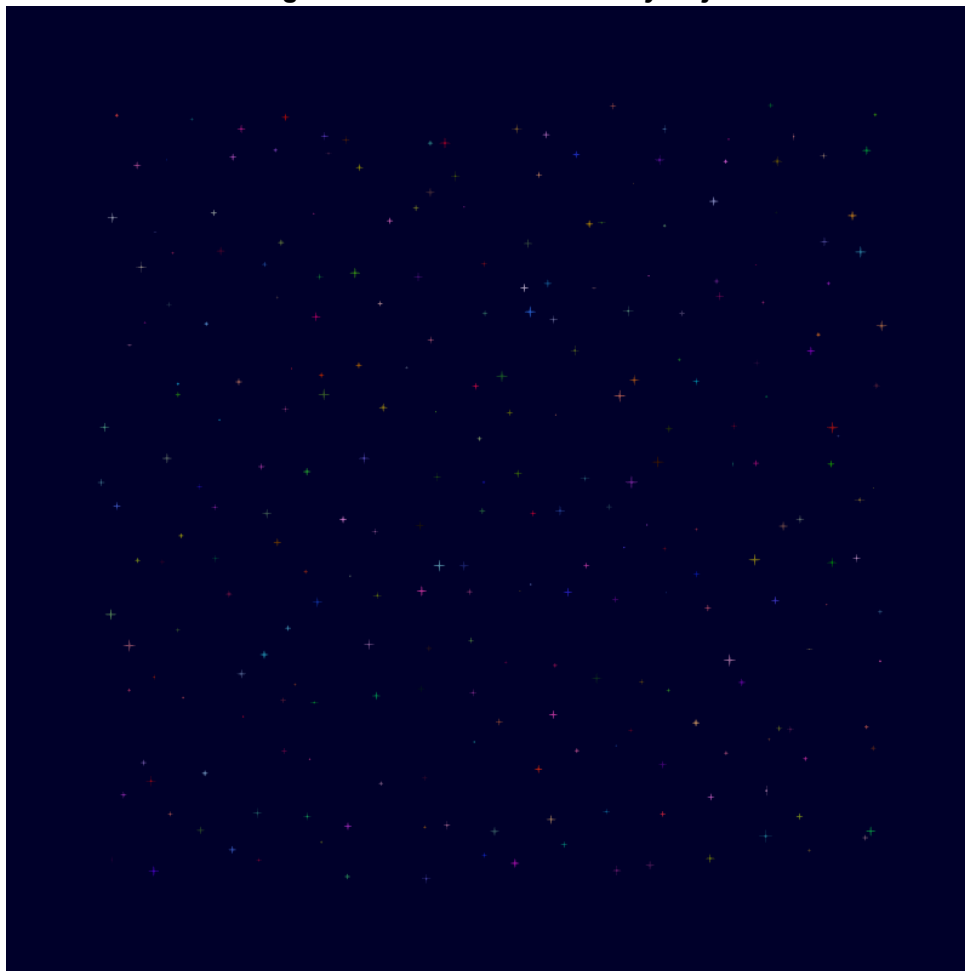
Como definido anteriormente, o intuito do projeto é representar cada órbita diário como uma estrela. Seguindo o quadro de referências de pixel arte combinado com as funções já existentes do *P5JS*, consegui criar as estrelas a partir de duas elipses cruzadas e um círculo no centro.

A função de elipse aceita a definição do seu posicionamento bem como das suas dimensões no eixo X e Y. Assim, ambas têm o mesmo valor de posição dentro da tela de *sketch* (desenho), como é chamado o arquivo onde o código inicial é escrito, variando os valores do seu tamanho eixo X e Y. Seus tamanhos oscilam usando a função *Noise* de um valor *t* (uma variável crescente a cada segundo) multiplicada por 13 na dimensão maior, e a função *Sin* de *t* para a dimensão menor.

Na prática isso significa que a cada *frame* que o código é executado, o valor de *t* aumenta em 1, logo o resultado das expressões $noise(t)*13$ e $sin(t)$ se modificam. A função *noise* retorna um valor pseudo-aleatório orgânico entre 0.0 e 1, o que implica que a elipse pode ter valores mínimos e máximos que oscilam entre 0 e 13 pixels. Já a função *sin* calcula o seno de um ângulo, em radianos, podendo retornar qualquer valor entre -1 e 1 dependendo do valor de *t*, o que implica que a menor dimensão das elipses oscila entre esses valores.

Quando justapostas com valores opostos para as dimensões X e Y o resultado é uma cruzeta piscando, pois tem suas dimensões alteradas entre 0 e 13 pixels de tamanho.

Figura 12 - Estrelas como *Array Objects*



Fonte: Autoria Própria

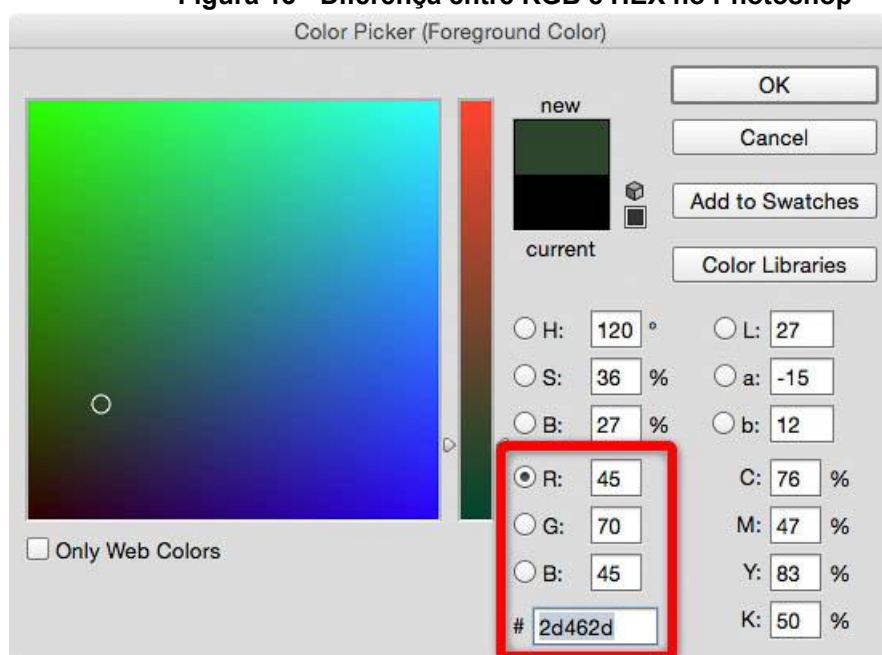
4.4 TODAS AS CORES DO ARCO-ÍRIS

A ausência de definições de cores no capítulo sobre projeto gráfico (com exceção das cores da marca) é proposital. Deixei para abordar esse assunto dentro deste capítulo porque o modo com que o P5JS processa as cores afetou diretamente o seu uso dentro do projeto.

Uma vez que o projeto busca representar e humanizar os dados estatísticos dos óbitos por COVID-19 no Brasil, acredito ser importante representar a pluralidade de possibilidades de existência humana através das cores. De modo que já estava

definido utilizar todos os valores do arco-íris. Contudo, enquanto designer, estava acostumado a trabalhar com cores dentro dos valores de CMYK para impressão e hexadecimal para RGB, e para a minha surpresa, o P5JS processa as cores dentro dos valores RGB ou RGBA.

Figura 13 - Diferença entre RGB e HEX no Photoshop



Fonte: Creative Pro (2021)

Isso implicou, a princípio, uma abertura em como eu abordaria as definições cromáticas. Ou seja, ao invés de definir um valor cromático a partir de um código específico, pelo hexadecimal, no P5JS os valores de cada canal cromático são definidos individualmente.

Em razão dessa nova liberdade matemática para definir as cores, tive duas abordagens distintas. Uma para definir a cor das estrelas e outra para definir a cor de destaque de outros elementos gráficos da peça, como informações textuais e borda.

5.4.1 As cores das estrelas

Para as cores das estrelas, utilizei a função *random* que gera um valor pseudo aleatório entre 0 e (mas não incluso) 1, caso não seja informado nenhum

valor, ou entre 0 e (mas não incluso) um valor máximo, caso seja informado apenas um valor, ou um valor entre um intervalo, caso seja informado dois valores.

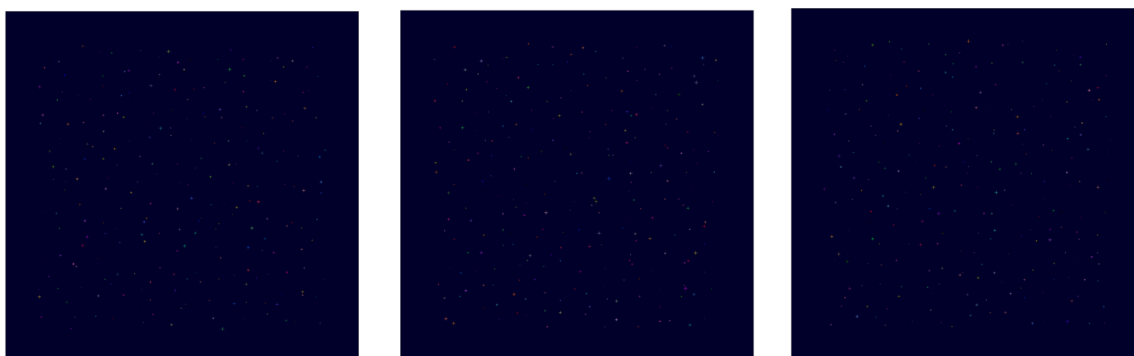
Cada objeto da estrela, quando está sendo processado, roda essa função *random* para cada um dos três canais cromáticos RGB. Contudo os valores não são idênticos. Para o canal R (vermelho) o valor por ser entre 0 e 200, para G (verde) entre 0 e 150, para B (azul) entre 0 e 200.

O motivo para a diferenciação entre os valores é que quando o canal G tem valor máximo igual ao dos outros, parte dos resultados das cores são néon e extremamente fortes, como o Verde Terminal ou Ciano puro. O que resulta, ao meu ver, em valores muito artificiais, que corroboram para um distanciamento do objetivo final de humanização. Além de ter baixo apreço estético.

O motivo para o canal verde, e não o azul nem o vermelho, ter seu valor máximo reduzido, foi de que qualquer combinação com um valor máximo alto de verde tem uma presença frequente de Verde Terminal como resultado. Diminuindo o seu valor, essa cor e outras derivadas desse valor ainda aparecem, mas em menor frequência, resultando em um conjunto mais harmônico.

Vale pontuar que a função *random* é rodada constantemente, a cada *frame*. Isso implica que as cores das estrelas são alteradas cada vez que o loop é rodado. O que resulta em um efeito visual similar ao de luzes de natal ou *pisca-pisca*. O que acredito que irá fortalecer o impacto empático que o projeto almeja.

Figura 14 - Sequência de *frames* de estrelas



Fonte: Autoria própria

5.4.2 As cores de destaque

A principal diferença entre a composição cromática das cores destaque e as cores das estrelas é que elas são estáticas. Uma vez que o código é iniciado, uma cor é definida e não se altera até que o código seja iniciado outra vez. Essa definição foi dada por motivos de legibilidade e equilíbrio, é difícil ler um texto piscando e os elementos de destaque competiriam com as próprias estrelas pela atenção do olhar.

A origem dessa diferença está na função utilizada para definir o valor de cada canal. Aqui, assim como no tamanho das estrelas, foi utilizada a função *noise*. Sua diferença entre a função *random* está no retorno de intervalos orgânicos, ou seja, com a *noise* cada valor gerado tem uma relação com o valor anterior, resultado em uma aleatoriedade mais organizada (chamado de ruído Perlin, inventado por Kim Perlin em 1980), enquanto a *random* gera valores não correlatos entre si. Entretanto, para poder gerar esses valores harmônicos, a *noise* carrega essa sequência quando o código é iniciado, e não altera até ser reiniciado. Ou seja, uma vez iniciado, o resultado de *noise(200)* será o mesmo, toda vez que o *loop* passar, ao passo que com *random(200)* o resultado é diferente. Por esse motivo, apesar das cores de destaque também serem geradas aleatoriamente dentro um intervalo dado, elas não se alteram na mesma arte.

Outra diferença é que o ruído Perlin sempre retorna um valor entre 0 e 1, o valor que ele aceita quando a função é chamada apenas define a localização desse valor dentro do um quadro tridimensional já gerado. Por esse motivo os valores definidos para uso com *noise* são mais altos que a função *random*. Caso uma estrela em um determinado *frame* tenha um valor cromático muito escuro, no próximo ela provavelmente terá um novo valor com maior contraste. Já com *noise* esse valor é fixo e utilizado em elementos que requerem legibilidade. Por isso os valores das cores destaque ficam restritos a faixas mais iluminadas do eixo cromático, ou seja, valores mais altos.

O valor de R é definido pela função *noise(3) * 350*, o valor de G por *noise(2) * 300* e o valor de B por *noise(5) * 400*. Outra diferença aqui é que os valores de R e B não são idênticos, pois acontece a presença exagerada de cores rosadas ou arroxeadas. Os valores passados em *noise* em cada uma das funções representa a sua proximidade dentro da escala Perlin gerada, ou seja, são valores próximos mas não idênticos.

5.4.3 A cor do céu

O fundo da arte, ou seja, a cor do céu, a princípio deveria ser um azul escuro RGB(0, 0, 42), representando um céu noturno, com ruídos cinzas se movendo entre as estrelas simulando nuvens. Contudo a simulação das nuvens através de código se mostrou um desafio maior do minhas habilidades, além de diminuir a percepção das estrelas. Por esses motivos essa ideia foi abandonada. Outros elementos celestes, como a lua ou cometas, também seriam ruídos para as estrelas e poderiam sugerir uma interpretação mais infantilizada de um cenário que não busca essa leitura.

Para não deixar o céu apenas azul, o que pesaria demais a percepção visual, além de ser monótono, foi aplicado um um gradiente de azul para preto, para simular as luzes da cidade ou um nascer do sol se aproximando. Esse efeito foi atingido a função *loadpixels* que permite definir o valor individual para cada pixel no *canvas*.

O efeito é construído através de dois *loops*, um para o eixo X e outro para o eixo Y, o último inserido dentro do primeiro *loop*. Ou seja, para cada pixel do eixo X o código busca todos os pixels no eixo Y individualmente. Quando o segundo roda, uma variável *index* é definida para a soma de do valor x e y multiplicado pelo tamanho da largura multiplicado por 4: $index = (x + y * width) * 4$, em seguida o valor de $pixels[index + 2] = y / 8 - 30$.

Esse processo foi aplicado a partir do tutorial do canal *The Coding Train*.

Figura 15 - Fundo gradiente

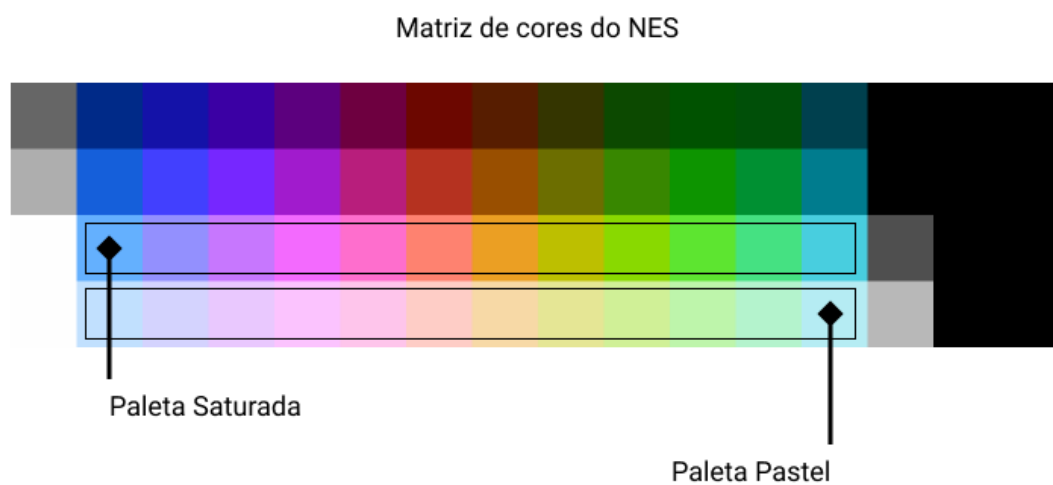


Fonte: Autoria própria

4.4.4 Atualização das cores

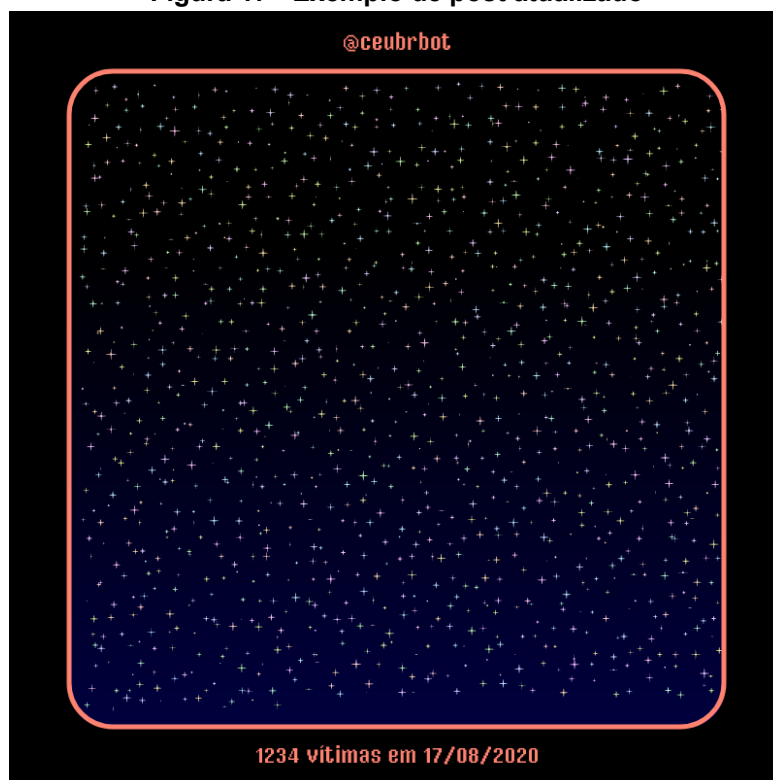
Durante o desenvolvimento mantive uma certa frustração em não ter um maior controle sobre a paleta cromática do projeto. Desde o princípio, tinha a ideia de aplicar a mesma paleta cromática base de 56 cores disponível para NES (Nintendo Entertainment System).

Depois de fazer alguns testes em outros projetos, descobri que poderia passar a informação de cor como *String* no p5.js, desde que estivesse em RGBA. Então montei dois *Arrays* contendo duas paletas de cores. A primeira, de tons saturados, foi utilizada como cor de destaque, enquanto que a segunda, de tons pastéis, foi utilizada para as estrelas. A aleatoriedade das escolhas das cores ainda foi mantida, apenas restringida em cada caso para os valores pré-definidos.

Figura 16 - Nova paleta de cores

Fonte: Wikipedia/Autoria própria (2021)

O resultado dessa atualização permitiu uma maior variação e distinção entre tonalidades que o método anterior, além de evidenciar mais as estrelas uma vez que todos os tons são claros e de alto contraste com o fundo.

Figura 17 - Exemplo de post atualizado

Fonte: Autoria própria

5.5 ELEMENTOS TEXTUAIS

Os elementos textuais de cada arte foram pensados para serem breves e de fácil compreensão. O objetivo é impactar pelos elementos gráficos, e permitir a visualização imediata da quantidade (massiva) de óbitos, vítimas de COVID-19. Contudo algumas informações textuais são necessárias para o entendimento de cada arte.

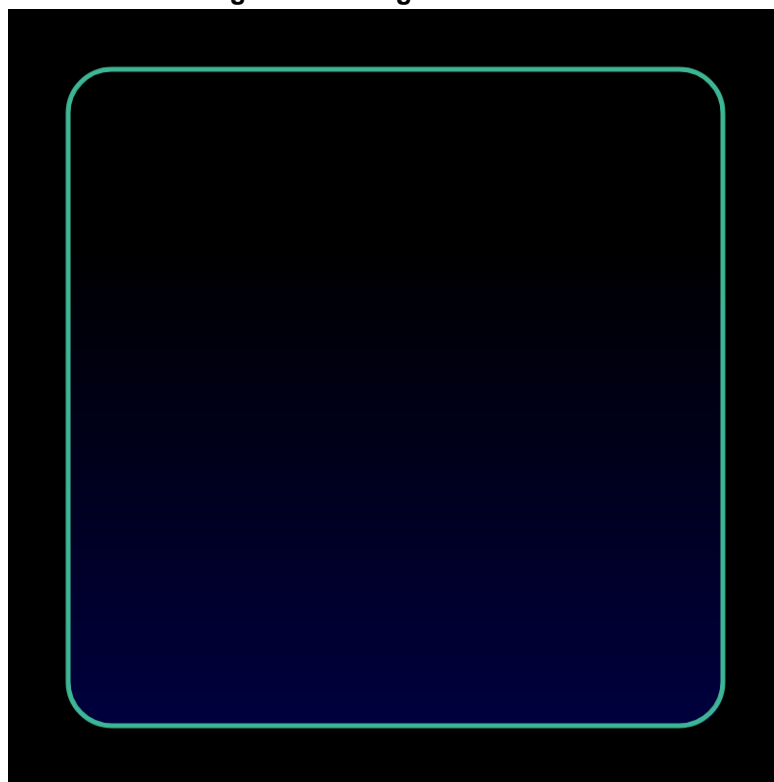
O primeiro elemento textual é o endereço da página no Twitter, *@ceubrbot*, versão encurtada do nome do projeto. Se tratando de redes sociais foi considerado importante acrescentar um identificador na arte para que permita que página seja encontrada caso alguma arte seja compartilhada fora do Twitter ou ainda que dentro da plataforma mas de modo que não afilie a página na publicação.

Os outros dois elementos são o número de mortes e o dia em questão. Esses dois elementos localizam as informações temporalmente e também apresentam o valor concreto, uma vez que a representação quantitativa das estrelas não garante a compreensão dos números.

Fora da arte, cada tuíte também acompanha as mesmas informações. O corpo de texto da publicação traz o número de mortos e o dia, seguido de votos póstumos: “Descansem em paz”.

5.6 DIAGRAMAÇÃO

Para permitir a inserção dos elementos textuais sem encobrir as estrelas ou ter a legibilidade comprometida em função delas, foi criada uma margem de 150 pixels, onde as informações foram diagramadas. Para distinguir essa margem do céu foi aplicada uma moldura de espessura de 5 pixels, formando uma janela que representa o distanciamento social e a quarentena. Como se fosse a janela de uma casa por onde alguém vê o céu.

Figura 18 - Margem e moldura

Fonte: Autoria própria

Tanto a moldura quanto os elementos textuais fora dela possuem a mesma cor de destaque, seguindo as definições informadas previamente. O tamanho da fonte para esses elementos também é o mesmo, 24 pixels. O arroba da página é posicionado na margem superior, centralizado no eixo X e centralizado entre o espaço da margem de 150 pixels no eixo Y.

O número de vítimas e a data, a princípio eram duas informações separadas, posicionadas na margem inferior da arte. O número de vítimas ao lado esquerdo, com alinhamento à esquerda, alinhado levemente para dentro da moldura. E a data posicionada ao lado direito, também com texto alinhado à esquerda, mas com o alinhamento mais adentro, de modo que terminasse antes da moldura. Ambos centralizados no eixo Y dentro do espaço da margem.

Contudo, foi pontuado pelo orientador, que esse posicionamento, somado a estética de pixel arte com referência a jogos virtuais, estava se assemelhando a uma tela de pontuação no final de um jogo (score). A possibilidade dessa leitura poderia ser interpretada como incentivo a um alto número de vítimas e desrespeito com as

perdas. Por esse motivo essas duas informações foram unidas em um único elemento textual e centralizado.

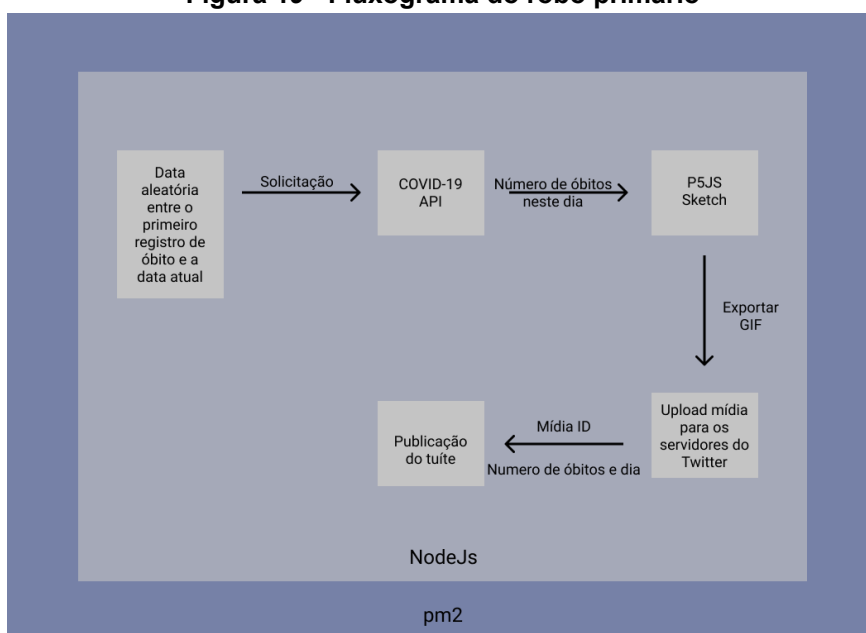
5 PROJETO DO ROBÔ

Como o intuito do projeto é construir um bot (robô) que gere e publique a arte automaticamente a partir de uma base de dados ou de interações com usuários, é necessário um sistema que gerencie esses processos. Como a biblioteca do P5JS é escrita em *JavaScript*, a ferramenta mais popular para gerenciamento de processos escritos nessa língua é o *NodeJs*. A qual também já tinha tido algum contato nos cursos de Desenvolvimento Web que fiz.

5.1 ARQUITETURA DO SISTEMA

O robô primário funciona em cinco etapas. Iniciando pela seleção de uma data aleatória desde o registro da primeira morte até o dia atual. A partir dessa seleção, o número de óbitos desse dia é solicitado a COVID-19 API. Uma vez que esse valor é retornado ele é passado ao *Sketch* que irá gerar a imagem no servidor e exportar a mídia GIF. Concluída a exportação, essa mídia é subida ao servidor do twitter por meio da sua API, o que retorna um número de identificação que é utilizado para chamar essa mídia quando o tuíte é de feito com as informações em texto.

Figura 19 - Fluxograma do robô primário



Fonte: Autoria própria

5.2 NODEJS & P5JS

Até aqui, todo código estava sendo escrito e rodado em um servidor local pelo VS *Codium* e visualizado pelo navegador. Na minha ingenuidade sobre as tecnologias web, eu não tinha percebido que a biblioteca do *P5JS* foi feita para ser rodada em um navegador. Ela precisa de DOM (*Document Object Model*, Modelo de Objeto de Documento), basicamente um espaço onde os elementos, textos e imagens de uma página são carregados.

Como o projeto prevê a criação das artes apenas no servidor, sem dispor de alguém acessando um endereço, eu precisava de um modo de simular esses processos. Felizmente dentro da biblioteca de pacotes do *NodeJS* havia uma implementação do *P5JS* justamente feita para esse propósito, a *Node-P5*.

A primeira etapa foi reajustar o código que eu já tinha escrito para funcionar com essa implementação, pois ela requer a adição de alguns termos ao chamar funções específicas do *P5JS*. Mas junto com o trabalho da alteração vieram algumas funções para problemas que eu ainda não tinha encontrado solução, como exportar essas imagens com o qual trabalharia.

Como a arte foi definida para apresentar movimento, já sabia que não era interessante publicar apenas imagens estáticas, que não trariam o mesmo impacto. Mas exportar e publicar arquivos de vídeo requer recursos computacionais e conhecimentos que eu não dispunha. A solução já estava integrada ao *Node-P5* que permite exportar *frames* e compilá-los em *GIF* de modo prático.

Além de prática, a solução também se adequa aos recursos disponíveis e às linguagens de comunicação do Twitter. A única limitação imposta foi o limite de *frames* permitido, que acredito vir da própria função *saveFrames* do *P5JS*. Caso um número superior a 15 *frames* seja definido a imagem nunca é finalizada, o arquivo fica corrompido e seu *upload* não é reconhecido pela *API* do Twitter. Muitas horas foram gastas até encontrar essa limitação como origem do problema.

5.3 TWITTER API

Para poder interagir com a plataforma do Twitter através de código é necessário acesso a *API (Application Programming Interface)* do Twitter, Interface de Programação da Aplicação. Esse acesso é garantido através da solicitação de uma conta de desenvolvedor associada a uma conta do Twitter.

O processo de solicitação consistiu no preenchimento de formulários que explicavam os usos previstos para a aplicação. Depois de preenchidos os formulários, recebi dois e-mail pedindo maiores explicações sobre o projeto, em especial por ter assinalado que compartilharia dados com uma instituição governamental, no caso este documento com a UTFPR.

Depois de uma semana, recebi o acesso como desenvolvedor, o que me garantiu chaves de acesso para poder solicitar fazer publicações, solicitar informações e mais entre uma série de outras funcionalidades que não foram utilizadas neste projeto.

Como a *API* do Twitter é relativamente complexa para mim, uma série de especificações sobre como configurar uma solicitação, como múltiplos campos que não estavam bem explicados como serem preenchidos, optei por utilizar outro pacote disponível para *NodeJS* que simplifica essas interações, o *Twitter API v2*.

Com esse pacote foi incrivelmente simples fazer o *upload* de mídias, publicar os tuítes e solicitar e responder menções.

Figura 20 - Documentação de API do Twitter

Parameters

Name	Required	Description	Default Value	Example
media	required	The raw binary file content being uploaded. Cannot be used with <code>media_data</code> .		
media_category	optional	The category that represents how the media will be used. This field is required when using the media with the Ads API. Possible values: <code>amplify_video</code> , <code>tweet_gif</code> , <code>tweet_image</code> , and <code>tweet_video</code> .		
media_data	required	The base64-encoded file content being uploaded. Cannot be used with <code>media</code> .		
additional_owners	optional	A comma-separated list of user IDs to set as additional owners allowed to use the returned <code>media_id</code> in Tweets or Cards. Up to 100 additional owners may be specified.		

Example Request

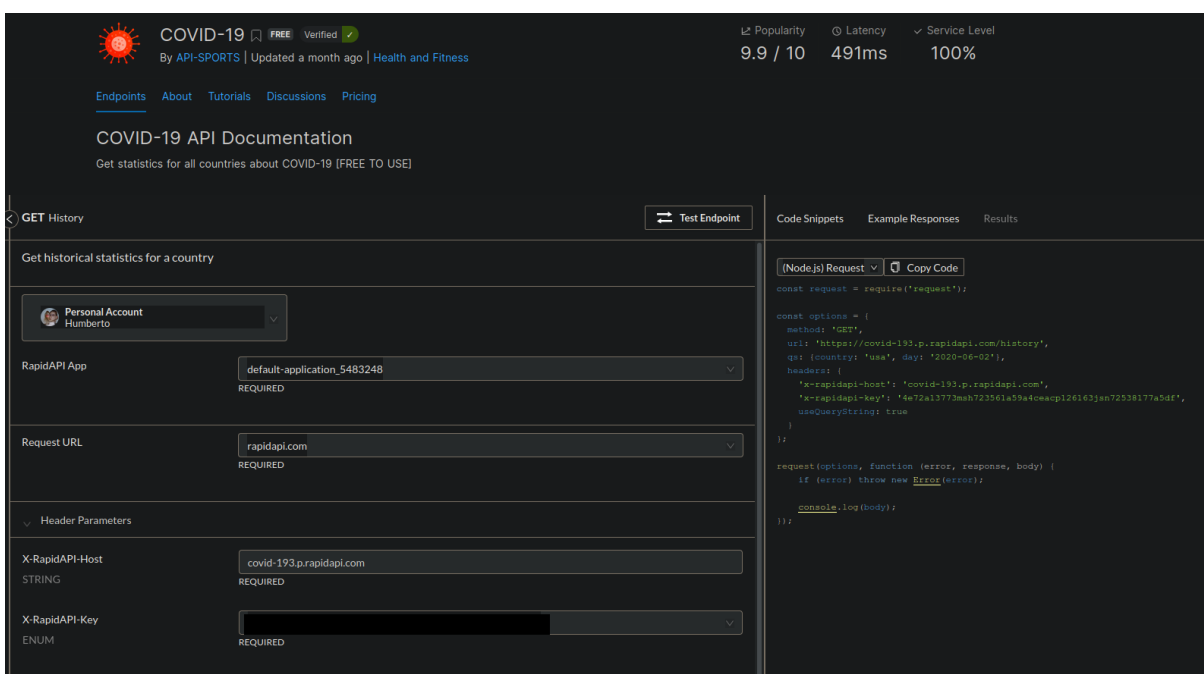
```
POST https://upload.twitter.com/1.1/media/upload.json?media_category=tweet_image
```

Fonte: Twitter Developer (2021)

5.4 COVID-19 API

Como mencionado na introdução, todo o processo desse projeto na verdade começou com a descoberta dessa *API* que permite a solicitação de dados sobre o COVID-19 separados por dia e países. Ela oferece o número atual de contagiados, o número de recuperados e o número de óbitos, tanto em valores absolutos como em valores diários. Todas essas informações estão disponíveis gratuitamente.

Figura 21 - Documentação de API do COVID-19



```
const request = require("request");

const options = {
  method: "GET",
  url: "https://covid-193.p.rapidapi.com/history",
  qs: {country: "usa", day: "2020-06-02"},
  headers: {
    "x-rapidapi-host": "covid-193.p.rapidapi.com",
    "x-rapidapi-key": "4e72a13772mah723f61a29a4eeap126163jan72538177a5df",
    useQueryString: true
  }
};

request(options, function (error, response, body) {
  if (error) throw new Error(error);
  console.log(body);
});
```

Fonte: Rapid API (2021)

5.5 PM2

Outra peça fundamental para o funcionamento do *bot* foi o pacote *PM2*, um gerenciador de processos para *NodeJS*. Como será explicado mais à frente, o *bot* é fragmentado em várias etapas, todas passíveis de erro e algumas impossíveis de serem finalizadas, como é o caso do *P5JS*. Uma vez que ele é iniciado, não há como pará-lo sem interromper toda a execução do código.

Nesse caso o *PM2* permite que a execução seja encerrada e retomada automaticamente, seja por erro ou definição explícita. Essa função foi imprescindível na configuração dos temporizadores das publicações e de outras funções apresentadas mais à frente.

5.6 MONGODB & MONGOOSE

Para gerenciar as menções do segundo *bot* foi necessário criar uma base de dados para controlar quais menções foram devidamente formatadas, quais não estão e quais precisam ser ou já foram renderizadas. Dentre os serviços que busquei para criar uma base de dados, o *Mongodb* pareceu ser o único a oferecer uma versão gratuita. Além da conveniência de eu já ter praticado exercícios de um curso utilizando essa plataforma.

Junto com ele utilizei também o pacote para *NodeJS* chamado *Mongoose*. Assim como o pacote para a *API* do *Twitter*, ele facilita o uso do serviço e diminui a quantidade de código necessário para realizar determinadas funções.

A aplicação dessas ferramentas será explicada mais detalhadamente no capítulo seguinte.

5.7 HEROKU

Heroku é uma empresa que oferece serviços de hospedagem na nuvem de aplicações. O seu pacote básico oferece serviços gratuitos e limitados. Esse serviço foi utilizado no princípio para hospedar o *bot* e desenvolver os primeiros testes. Contudo, como o *bot* precisa rodar 24 horas por dia, todos os dias, ele excede o limite gratuito ofertado. Por essa razão essa solução foi trocada por um servidor local.

5.8 SERVIDOR LOCAL

Juntando peças de dois *notebooks* antigos e em desuso que tinha em casa, consegui montar uma máquina funcional rodando *Linux Mint XFCE*. Apesar dos

recursos serem ligeiramente inferiores aos oferecidos pelo Heroku, o que demandou atualizações na arquitetura do projeto todo, ele funciona ininterruptamente e permite um controle mais preciso através do acesso direto ou por *SSH*.

6 DESIGN DE INTERAÇÃO

Para além da publicação automatizada das artes, foi levantado pelo orientador a possibilidade do *bot* interagir com pessoas através da plataforma. Essa ideia encontrou solo fértil na necessidade de humanizar ainda mais os dados apresentados.

Apesar de se propor a humanizar os números, a arte das estrelas não se relacionava diretamente com nenhuma vítima. É comum memoriais trazerem o registro das pessoas envolvidas nas tragédias. Geralmente uma lista de nomes, descrições, fotos, etc. Informações ou artefatos que caracterizem essas pessoas.

Da união dessas duas perspectivas foi desenvolvido um segundo *bot* com o propósito de gerar artes personalizadas com informações fornecidas através de menção pelas outras pessoas da plataforma, para lembrarem de seus entes queridos perdidos.

6.1 QUAIS INFORMAÇÕES

Na era da informação, acúmulo e vazamento de dados são assuntos frequentes e as definições de privacidade estão cada vez mais difíceis de serem definidas e alcançadas. Por esse motivo considerei importante limitar a indicação de informações a serem compartilhadas pelas outras pessoas ao solicitarem participar do projeto.

Um segundo motivo para a limitação das informações é o espaço disponível na arte para tal. Largos corpos de texto, além de consumirem o espaço contemplativo proposto pelas estrelas, também teriam uma baixa legibilidade.

Dentro dessas perspectivas, foi recomendado informar apenas: o primeiro nome, a idade, um local (que pode ser cidade ou estado), a data do óbito e uma breve mensagem livre, podendo ser uma descrição afetiva ou uma despedida.

6.2 NOVOS ELEMENTOS

Os novos elementos textuais, mencionados anteriormente, foram inseridos e diagramados dentro do espaço do céu, para sinalizar uma individualização. Todos os elementos foram centralizados, variando o tamanho da fonte e a disposição dos mesmos de acordo com uma hierarquia visual.

O nome aparece um pouco abaixo da metade vertical da arte, exatamente a $3/2$ da altura, com fonte em tamanho 60. A localização e a idade aparecem na mesma linha, separadas por hífen, logo abaixo, em tamanho 28. Por último a mensagem ou dedicatória, em tamanho 24. Seu posicionamento vertical é responsivo, já que leva em consideração a quantidade de texto e linhas necessárias, prevendo mensagens curtas ou longas.

Figura 22 - Exemplo de arte reposta



Fonte: Autoria própria

Além dos elementos textuais, outro foi adicionado para evidenciar o caráter individual dessa arte, uma estrela branca maior, centralizada na metade superior da arte. Assim como as estrelas menores, ela é composta de duas funções de elipse e uma de círculo centralizado. As únicas modificações, além da cor, são suas dimensões, atualizadas para oscilar entre 0 e 150 pixels, no eixo maior, e 0 e 2 pixels no eixo menor.

A princípio essa estrela tinha posicionamento fixo. Contudo foi pontuado pelo orientador que adicionar movimentação iria proporcionar maior humanização desse ícone. A fórmula que rege sua movimentação é baseada no valor do seno e cosseno de um conjunto de valores variáveis que permite delimitar sua movimentação dentro de raio de 150 pixels em movimento de curva algébrica de quarto grau (similar ao símbolo do infinito).

Figura 23 - Demonstração de movimento de estrela



Fonte: Autoria própria

6.3 FALA QUE EU TE ESCUTO

Para solicitar a participação no projeto, e interação com o *bot*, é necessário fazer um tuíte mencionando o arroba da conta, @ceubrbot. Logo após a menção é necessário informar uma chave, que diferencia essa menção de outras possíveis, como caso um usuário queira mencionar a conta com outro objetivo que não participar. Essa chave é a palavra “céu”.

Em seguida da chave, devem ser informados o primeiro nome (ou apelido), a idade apenas em números, as data do óbito em formato DD/MM/AAAA (como é convencional no país), a localização e por último a mensagem.

Todas as informações precisam ser separadas por um sinal de til (“~”) que serve para quebrar a *string* informada pela *API* do Twitter em diferentes campos, que permitem a busca pela informação do número de óbitos do dia especificado e diagramar as informações na arte.

Este é um exemplo informado de como chamar o *bot* a partir de um tweet publicado por qualquer usuário do twitter:

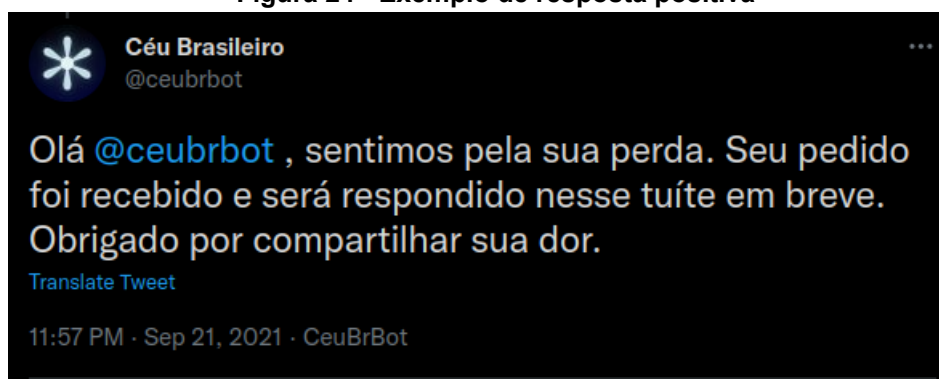
@ceubrbot_céu~Mariano~42~08/04/2021~Rio Branco-AC~Esposo, tio e amigo... a saudade é grande!

6.4 PESCADOR

A primeira parte do *bot* consiste em buscar frequentemente por novas menções à conta em tuítes publicados no Twitter, diferenciar as que têm a palavra chave, e caso tenham, conferir se a data informada corresponde ao período do início da pandemia até o dia atual. Todo esse processo é automatizado com Javascript e a API do twitter, e gerenciado pelo PM2 com NodeJs.

Caso todas as etapas aconteçam sem erro, o número identificador desse tuíte junto com um *boolean* que informa que ele não foi renderizado são salvos para uma base de dados, para ser localizado na próxima etapa. Após esses dados serem salvos, um tuíte de resposta informa a pessoa que a solicitação foi recebida e que a arte já está na fila para produção.

Figura 24 - Exemplo de resposta positiva



Fonte: Autoria própria

Caso a data não se enquadre dentro deste intervalo ou haja algum outro erro que não permita a identificação da data, o tuíte é salvo dentro de outra base de dados, uma resposta informa que há um erro com a data e pede que o usuário confira os dados e faça a solicitação novamente.

Esse processo de solicitação da lista de menções e análises acontece a cada 5 minutos, e caso uma mensagem já tenha sido inserida na base de dados, seja na de produção ou na de erros, elas são ignoradas.

6.5 FALADOR

A segunda parte do *bot* consiste em buscar na base de dados de produção, por objetos que tenham o valor de renderizado como falso. Uma vez que ele foi localizado, o número identificador é utilizado para buscar esse tuíte na base de dados do twitter e começar o processo. O sistema foi projetado dessa maneira para não armazenar nenhuma informação pessoal, apenas o número de identificação do tuíte, que é público. Essa escolha busca evitar vazamento de informações, assim como diminuir o potencial tamanho dos arquivos na base de dados.

Quando o tuíte é localizado, todo o seu corpo de texto é analisado e quebrado nas categorias de cada informação. A data, a principal informação, é formatada para o formato ISO (AAAA/MM/DD) necessário para solicitar as informações da *API COVID-19*. Quando esse dado é retornado, ele é passado para o *Sketch* do *P5JS* junto com as outras informações, para gerar o *GIF*.

Após gerado, através da *API* do Twitter é feito o *upload* da arte para seus servidores, em seguida publicada como um tuíte, respondendo a conta que fez a menção. Depois de postado o *boolean* desse objeto na base de dados é marcado como *true*, para impedir que seja repetido.

7 APRESENTAÇÃO DO PROJETO

Para apresentar o projeto a qualquer um que encontre a conta no Twitter, foi escrito um texto introdutório para explicar seu propósito e suas funcionalidades. O texto foi organizado em formato de *thread* e fixado no topo do perfil para facilitar a visualização e a leitura. Segue o texto:

Olá! Esse é um bot para te lembrar.

Lembrar das tantas pessoas que se foram, vítimas de COVID e da má gestão pública.

Lembrar que a pandemia não acabou. Se vacinar, se manter seguro e evitar a propagação do vírus ainda é o melhor que podemos fazer em uma escala individual.

Lembrar que o futuro será coletivo ou não será.

Como funciona:

O bot publica 4 vezes ao dia uma GIF com estrelas representando o número de vítimas de um dia da pandemia, selecionado aleatoriamente.

Como contribuir:

Muito mais importante que os números são as pessoas que eles representam. Por isso, você pode contribuir com o Céu Brasileiro informando dados básicos e gerar uma GIF única para uma pessoa querida que nos deixou.

Para o bot conseguir entender você é preciso seguir esse modelo:

Esse é um exemplo de como escrever o seu tuíte e de como o bot vai responder se ele te entender.

@ceubrbot céu~Mariano~42~08/04/2021~Rio Branco-AC~Esposo, tio e amigo... a saudade é grande!

Aviso:

O projeto está em fase de testes. Todo e qualquer comentário, sugestão ou crítica são bem vindos.

Se desejar que sua contribuição seja apagada, só entrar em contato.

A DM está aberta.

É preciso estar atento e forte

Abrço

Figura 25 - Imagem explicativa para interagir com o robô

Passo a passo para contribuir com o Céu Brasileiro

1. Mencione a conta
2. Use a chave "céu" para ativar o bot
3. Separe todos os campos com ~ sem espaços
4. Por motivos de privacidade, recomendamos que não informe o nome completo, apenas como voce conhecia a pessoa, apelido ou primeiro nome
5. A idade em números
6. A data do óbito em formato DD/MM/AAAA
7. O local, cidade e/ou estado
8. Mensagem livre

1 2 4 5 6 7 8
@ceubrbot céu~Nome~Idade~Data do Óbito~Local~Mensagem
3

Fonte: Autoria própria

8 CONCLUSÕES

Como o projeto consiste de múltiplas camadas que influenciaram seus direcionamentos e resultados, abordarei esses recortes separadamente, para destinar a devida atenção a cada segmento.

8.1 AMIGOS SÃO PARA ISSO

Durante o processo de desenvolvimento, apresentei o projeto a amigos e pedi suas opiniões. Uma vez que o consegui resolver os eventuais *bugs* do código, coloquei-o no ar dia 22 de Setembro deste ano. Compartilhei o mesmo nas minhas redes sociais solicitando *feedback*. Em auxílio na divulgação, meu orientador compartilhou o projeto em suas redes em distintos momentos.

De modo geral, o engajamento e retorno foram ínfimos. Dentre os meus conhecidos e amigos, houve umas cinco reações positivas, entre “curtir” e adjetivos. Alguns deram sugestões que já haviam sido consideradas e inviabilizadas por vários motivos, como a adição de elementos celestes para uma maior ambientação ou efeitos visuais mais elaborados e dramáticos que fugiam aos recursos disponíveis.

Um em especial que foi aplicado foi a formatação da data informada na arte e no corpo do tuíte. Como a *COVID-19 API* requer a formatação da data em ISO (AAAA/MM/DD), a praticidade me levou a apresentar essa informação nesse formato. Um amigo querido sugeriu que essa formatação dificultava a leitura porque estamos acostumados a ler em formato inverso. E essa alteração foi aplicada ainda em estágio de desenvolvimento.

8.2 TERRENO DE GRINGO

Acho fundamental pontuar que todas as tecnologias que utilizei são estrangeiras, majoritariamente estadunidenses e européias. Os conteúdos que tive que consumir foram todos em inglês, desde cursos inteiros encontrados por Torrent a vídeos tutoriais disponíveis no YouTube. Toda a dinâmica dessas tecnologias acontecem em um fluxo hierárquico muito acima de mim, um homem gay queer

latino americano (que necessito até de termos estrangeiros para pontuar minha própria identidade).

E apesar de entender algumas das dinâmicas de poder que gerenciam essas tecnologias, devo pontuar que suas comunidades são muito ativas e acessíveis. Jamais teria conseguido desenvolver esse projeto sem os materiais de apoio do canal The Coding Train ou das incontáveis respostas do StackOverflow que me ofereceram o código e a lógica que eu precisava para montar as peças.

Trago essa observação para uma reflexão sobre como o conhecimento é transmitido tradicionalmente na educação formal, uma vez que nos últimos cinco anos, tenho aprendido teorias e práticas de ferramentas diversas de modo muito mais orgânico e acessível através da internet e das comunidades que a mantêm do que tive durante toda minha formação acadêmica.

8.3 UMA NOVA ESPERANÇA

Desempregado, respeitando o distanciamento social, portanto isolado, e sendo instigado diariamente pelas inúmeras tragédias políticas que todo o mundo tem passado, mas em especial o Brasil, confesso que aplicar os conhecimentos sobre desenvolvimento web, que adquiri de forma independente, e solucionar passo a passo os problemas do projeto, foram umas das poucas satisfações pessoais que tive neste ano.

A pandemia obrigou a todos a se aproximarem mais das máquinas como uma forma segura de existir. Para nós que já trabalhávamos com elas diariamente, imagino que essa aproximação foi ainda mais intensa. Ou pelo menos posso afirmar que foi o meu caso, e, ao aceitá-la como extensão do meu corpo, quis entendê-la melhor, não apenas utilizá-la segundo os padrões de mercado e da industrial. Quis, e quero ainda mais, explorar o meu potencial criativo combinado com o potencial computacional, e de quebra ser melhor remunerado do que já fui, sendo “Canvas” humano para o marketing.

Diante de um mundo à beira de um colapso climático, deterioração das relações, exploração ambiental, animal e humana a favor dos interesses do capital e dos senhores do mundo, todo o caos documentado e vendido diariamente como

entretenimento, fico feliz em descobrir que posso encontrar um pouco de paz junto com os binários das máquinas, pois aqui as constantes são a regra e as palavras tem poder de criação.

Continuarei explorando essas tecnologias não apenas como meio de existência e expressão individual mas também, e principalmente, como recursos fundamentais para as lutas sociais e de afirmação da humanidade dentro do ciberespaço.

8.4 CONSIDERAÇÕES ACADÊMICAS

É do meu entendimento que este trabalho foge a inúmeros padrões de TCC da instituição e do departamento do curso. Não apenas pelas perspectivas do que pode ser um projeto de Design Gráfico mas também pelo modo em que todo o processo é documentado e apresentado.

Não apresento nem me baseio em pontos teóricos marcados. Não faço análises profundas de minhas abordagens, apesar de saber que elas estão disponíveis e me foram apresentadas durante o desenvolvimento do projeto.

Essa “escolha” acontece por alguns motivos, como o fato d’eu abordar assuntos que não foram trabalhados teoricamente durante minha formação, e que me foram cobrados de última hora durante o desenvolvimento. Fazer leituras, interpretá-las e associá-las com interpretações do projeto foi uma tarefa maior do que minhas condições mentais me permitiram.

Também gostaria de enfatizar a discordância entre uma formação tecnológica, que supostamente me preparou para atuar no mercado gráfico dos anos 2000, e uma argumentação teórica de um trabalho prático, que aplica conhecimentos que não foram abordados durante o curso.

Desenvolvi o trabalho de modo orgânico, entendendo como um exercício prático dos conhecimentos que desenvolvi sobre a área durante minha vida, dentro, e principalmente, fora da academia.

Entendo meu dever com a sociedade brasileira, que me ofereceu esse acesso, e busquei honrar esse compromisso abordando um tema de interesse

coletivo e apresentá-lo sobre uma perspectiva nova dentro do ecossistema do departamento.

9 EPÍLOGO

O perfil do twitter do bot está disponível pelo link <https://twitter.com/ceubrbot> e permanecerá funcional até o final do ano de 2021, quando eu pretendo desativar o servidor e arquivar o projeto. A conta permanecerá disponível no twitter por quanto tempo for permitido.

Entendo que a única funcionalidade que esse documento pode ter, além de concluir minha formação, é servir de algum apoio/incentivo a estudantes que também tenham interesse em programação e design. Por isso, mantendo a coerência com a minha formação pública e com as comunidades de *open software* e *creative coding*, disponibilizo o código do projeto para estudo/uso <https://github.com/salguento/ceubrbot> e me coloco a disposição para compartilhar mais detalhes do projeto, dos processos e das tecnologias utilizadas ou criar novos projetos juntas. As informações de contato estão disponíveis no perfil do Github.