UNIVERSIDADE D
COIMBRA

Fábio André Dos Santos Pereira

DO ESQUELETO À FORMA

EXPLORAÇÕES ALGORÍTMICAS NA CRIAÇÃO DE GLIFOS

Dissertação no âmbito do Mestrado em Design e Multimédia,
orientada pelo Professor Tiago Filipe dos Santos Martins
e pelo Professor Doutor João Manuel Frade Belo Bicker,
e apresentada ao Departamento de Engenharia Informática
da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

Setembro de 2019

Agradecimentos

Agradeço aos meus pais, *Dorinda* e *José*, por me terem proporcionado, a melhor jornada da minha vida e pelo amor incondicional.

Agradeço aos meus irmãos, *Rita* e *Rafael* por crescerem comigo e me tornarem na pessoa que sou hoje.

Agradeço à família que escolhi e que permanecerá para o resto da minha vida, os meus amigos.

Agradeço aos meus orientadores a oportunidade dada, a partilha de conhecimento e acima de tudo o voto de confiança.

Agradeço a todas as pessoas que contribuíram de alguma forma para a realização desta dissertação.

Obrigado Coimbra.

Resumo

Design de tipos
Esqueleto tipográfico
Design generativo
Sistema web
Tipografia

É através da tipografia que é dada ordem e forma, visível e durável, à comunicação escrita. Os autores transpõem as suas ideias em palavras e os designers gráficos procuram tornar estas palavras acessíveis e atraentes ao leitor. A história da tipografia revela uma contínua reinvenção a par do aparecimento de novas tecnologias, às quais, designers e tipógrafos se têm vindo a adaptar.

A utilização destas tecnologias no processo de design abriu caminho a novas explorações nunca antes imaginadas ou possíveis. Designers começaram a explorar processos generativos em diferentes projetos de design, como identidades visuais dinâmicas ou tipografia generativa. Este tipo de processos permitiu quebrar limitações de abordagens de design existentes, dando lugar a soluções experimentais e únicas.

Neste contexto, este trabalho procura aliar o design de tipos a processos generativos num sistema algorítmico web capaz de gerar glifos a partir de esqueletos tipográficos. Este sistema, que se encontra online em cdv.dei.uc.pt/2019/letterspecies, gera uma fonte OpenType preenchendo um esqueleto tipográfico através de uma técnica de desenho, ambas personalizáveis. O utilizador pode alterar um conjunto de parâmetros para alterar a forma dos glifos. As fontes resultantes podem ser aplicadas nos mais diversos projetos e *media*.

Abstract

Generative design
Type design
Typography
Typographical skeleton
Web system

It is through typography that it is given visible and durable order to written communication. The authors transpose their ideas into words and graphic designers try to make these words accessible and attractive to the reader. The history of typography reveals a continuous reinvention along with the emergence of new technologies, which designers and typographers have been adapting.

The use of these technologies in the design process paved the way for new explorations never before imagined or possible. Designers began to explore generative processes in different design projects, such as dynamic visual identities or generative typography. This type of process has allowed breaking the existing limitations of design approaches and giving rise to experimental and unique solutions.

In this context, this work seeks to combine type design with generative processes in a web system capable of automatically generating glyphs from typographic skeletons. This system, which is online at cdv.dei.uc.pt/2019/letterspecies, generates an OpenType font by filling a typographic skeleton through a drawing technique, both customizable. The user can change a set of parameters to modify the shapes of the glyphs. The resulting fonts can be applied in the diverse design projects and media.

Glossário

008 — 011

***Asynchronous Javascript and XML* (AJAX)** Consiste no uso de métodos JavaScript e XML para a troca de informação através de pedidos assíncronos ao servidor. Muito utilizado para o desenvolvimento de aplicações *web* no lado do cliente de forma dinâmica e interativa.

Bitmap Tipo de representação, na qual cada item corresponde a um ou dois *bits* de informação. Uma letra pode ser representada como uma série de bits.

Caractere Qualquer elemento tipográfico incluído numa fonte como a letra, o número e sinal de pontuação.

Contraste No contexto tipográfico, consiste no grau de diferença entre os traços grossos e finos de uma determinada letra; pouca diferença significa pouco contraste, muita diferença significa forte contraste.

Curva de *Bézier* Curva polinomial expressa como a interpolação linear entre pontos representativos, chamados de pontos de controlo ou pontos de âncora.

Eme Medida tipográfica equivalente à distância do tamanho do tipo. Num tipo de 12pt, um eme tem 12pt. Também chamado de *quadratim* em português.

Família Tipográfica Conjunto de fontes relacionadas pelas mesmas características e projetadas para funcionarem em conjunto.

Fonte Conjunto completo de caracteres num único estilo ou peso. Por exemplo, todos os caracteres na versão bold de um determinado tipo de letra. Ambos os termos, fonte e tipo de letra, significam a mesma coisa.

GIF Animação em formato de imagem *bitmap*.

Glifo Versão particular de um caractere. Por exemplo, A e *A* são dois glifos diferentes (da mesma fonte) para o mesmo caractere. Designers de tipos, criam glifos. Equivalente a *letterfom.*

Input No contexto tecnológico, consiste na entrada de um conjunto de dados num sistema.

***JavaScript Object Notation* (JSON)** Tipo de formato para a fácil troca de dados entre sistemas.

Kern Parte de uma letra que invade o espaço de outra.

Kerning Consiste em ajustar (aumentar ou diminuir) o espaço entre pares de letras específicos, para assegurar uma aparência uniforme e consistente, de forma a que os olhos do leitor não sejam distraídos.

Legibility Diz respeito ao design de uma fonte e às formas dos glifos; capacidade de o leitor percecionar a relação das letras entre elas.

Letterform Equivalente a glifo (ver Glifo).

Lettering Arte do desenho de letras para uma única utilização e finalidade.

Link Consiste numa hiperligação entre documentos na *web*. Texto ou imagem com hiperligação, ao ser clicado provoca a exibição de um novo documento.

Metafont Linguagem de programação com a finalidade de preparar fontes para impressão. Foi desenvolvida por Donald Knuth em 1979.

OpenCV Biblioteca de programação com suporte para visão computacional em tempo real.

Open source *Software* de código aberto, desenvolvido de forma colaborativa.

OpenType Font (OTF) Consiste na combinação do formato de fontes *PostScript* e TrueType. Desta forma, contém mais caracteres e alternativas que não eram possíveis antes. Desenvolvido pela Microsoft em colaboração com a Adobe Systems.

PostScript Linguagem de programação desenvolvida para a visualização de informações e descrição de páginas pela Adobe Systems.

Processing É uma linguagem de programação *open source,* desenvolvido para a exploração de arte digital.

Readability Diz respeito ao uso da linguagem e apresentação do texto; capacidade de o leitor conseguir percecionar a relação das letras num texto.

Ruído branco Sinal aleatório com igual intensidade em diferentes frequências, o que lhe dá uma densidade espectral de potência constante.

Scalable Vector Graphics (SVG) Consiste num formato de imagem vetorial para gráficos bidimensionais com suporte para interatividade e animação baseado numa *Extensible Markup Language* (XML).

Tipo Termo utilizado para as letras de metal utilizadas na composição tipográfica. Refere-se às letras de forma geral. Também pode ser usado para letras feitas digitalmente.

Tipo de letra Equivalente a Fonte (ver Fonte).

Tracking Consiste no controlo de todos os espaços entre letras num dado bloco de texto, deste modo, quando é alterado, afeta todos estes espaços de forma igual.

TrueType Font (TTF) Formato de fonte desenvolvido pela Apple e Microsoft.

Type Foundry Empresa que fabrica e/ou comercializa fontes individuais ou famílias tipográficas.

Web *World Wide Web* é um espaço de informação em que documentos e outros são identificados por *Uniform Resource Locators* (URL), interligados por *links* e acessíveis pela Internet.

Web Open Font Format (WOFF) Formato de fonte para uso em páginas *web*. São fontes OpenType ou TrueType, compactadas com metadados XML adicionados.

Website Coleção de várias páginas da *Web* relacionadas, incluindo conteúdo multimédia, normalmente identificado com um domínio e publicado num servidor.

Índice

1 — Introdução

014 — 017

A tipografia é uma das componentes do design gráfico com mais importância. É através da tipografia que, na qualidade de designers, damos ordem e forma, visível e durável, à comunicação escrita. Esta faz fronteira, por um lado, com a escrita e a edição e, por outro, com o design gráfico. A escrita é do domínio dos autores, no sentido em que são estes que formulam as suas ideias e as transpõem para palavras. Por outro lado, de forma a tornar estas palavras acessíveis e atraentes ao leitor, falamos do domínio dos designers gráficos, dos tipógrafos, e ainda — e não menos relevante — dos designers de tipos. Estes últimos, criam tipos de letra com base num conjunto de características que reflete o propósito dos mesmos (BRINGHURST, 2008; UNGER, 2018).

A história do design gráfico revela uma contínua reinvenção da disciplina simultaneamente com o surgimento de novas tecnologias. Ao longo dos anos de evolução tecnológica, duas importantes tecnologias definiram a prática dos designers: o computador pessoal fornecendo computação para as massas e a Internet conectando pessoas e informação em grande escala (ARMSTRONG, 2016).

> *The personal computer morphed into a large networked mind through which creatives could think, make, collaborate, and distribute. Users commonly experienced content through active engagement online: pressing a button, scrolling down a page, uploading content, customizing interfaces. Interactivity took over.*
>
> ARMSTRONG, 2016:15

Como Helen Armstrong afirma, estas duas ferramentas combinadas forneceram os meios necessários para uma nova forma de pensar, comunicar e interagir abrindo caminho para novas e inesperadas explorações na prática dos designers.

No contexto do design de tipos, a introdução de novas tecnologias veio impulsionar o processo de criação, pois veio facilitar o desenho, a modificação e a experimentação de tipos de letra até ficarem de acordo com o gosto do designer. No entanto, Gerard Unger afirma que «*while the technology for type design and the design process have changed fundamentally, letterforms have changed very little with the transition from tangible and analog type to digital and immaterial types*». Sharon Poggenpohl relembra o vasto leque de possibilidades proporcionadas por um sistema computacional interativo que pode executar operações como «*fragment, distort, incline, change weight, etc.*» em tipos de letra, com simples instruções (POGGENPOHL, 1983; UNGER, 2018).

Desta forma, há espaço para uma investigação neste campo, adotando uma abordagem experimental no processo de criação de *letterforms* (glifos), com o intuito de abrir espaço para novas formas. Com a intenção de concretizar esta nova abordagem, surgiu a ideia de aliar o desenho de tipos a processos generativos, num sistema interativo que explora a criação algorítmica de tipos de letra, personalizável por

cada utilizador. Este trabalho vem ainda dar a possibilidade de desenvolver e aprofundar conhecimentos nas áreas de desenho de tipos e de programação no design já introduzidas na licenciatura em Design e Multimédia (Universidade de Coimbra) e que me suscitaram interesse.

1.1. Enquadramento

A constante inovação tecnológica ao longo dos últimos séculos tem vindo a proporcionar ferramentas com enorme potencial nas mais diversas áreas. O design gráfico não é exceção e os designers começaram a adaptar a sua forma de trabalhar às novas tecnologias, muito devido ao aparecimento de novas tecnologias digitais como por exemplo, o *Apple Macintosh* (1984), a linguagem *PostScript* (1984), o programa *Adobe PageMaker* (1985) e o formato *OpenType* (1996). Estas tecnologias aceleraram o processo do design e facilitaram a produção e reprodução de tipos de letra digitais, assim como abriram novos horizontes na exploração criativa. (UNGER, 2018)

> *From the beginnings of typography, in the middle of the fifteenth century, technology has had much influence on letterforms, both in the making of typefaces as well as on their application in typography.*
>
> UNGER, 2018:89

Por consequência, começaram a ser explorados processos algorítmicos e generativos, anteriormente insustentáveis e/ou impensáveis no contexto do Design Gráfico, como por exemplo, identidades visuais dinâmicas ou tipografia generativa. Através destes processos, o designer consegue automatizar tarefas e introduzir novas variáveis nos artefactos produzidos, como a aleatoriedade. Desta forma, este trabalho pretende fazer uso destes processos de forma criteriosa, tendo em conta a história da tipografia e o design contemporâneo, para explorar novas possibilidades criativas na criação de tipos de letra.

1.2. Objetivos

Nesta dissertação, pretende-se a conceptualização e desenvolvimento de um sistema algorítmico *web* capaz de gerar glifos a partir de esqueletos extraídos de tipos de letra existentes. Os glifos gerados são o resultado do preenchimento do esqueleto utilizando diferentes formas visuais e técnicas de desenho. Assim, o sistema tem como *input* um esqueleto e uma técnica de desenho pretendida pelo utilizador, e como *output*, glifos que podem ser visualizados e configuráveis em tempo real por parte do utilizador, tendo a seu dispor um conjunto de variáveis associadas à anatomia da letra e à técnica de desenho escolhida. Estes glifos podem ser utilizados nas mais diversas aplicações, e para isso, é possível exportar o conjunto dos glifos como um tipo de letra. É também um objetivo fazer do sistema uma ferramenta prática para designers e outros utilizadores interessados pelo desenho de tipos.

1.3. Estrutura da Dissertação

A presente dissertação encontra-se estruturada em seis capítulos: Introdução, Plano de Trabalhos, Estado da Arte, Trabalho Relacionado, Projeto Prático, e Conclusão.

No primeiro capítulo (INTRODUÇÃO) é apresentado e contextualizado o tema da dissertação, assim como os objetivos pretendidos. No segundo capítulo (PLANO DE TRABALHOS) são definidas as tarefas necessárias para atingir os objetivos propostos. É ainda apresentado o método adotado para a realização do projeto prático. O terceiro capítulo (ESTADO DA ARTE) apresenta a síntese da investigação bibliográfica realizada sobre os principais temas inerentes ao tema da dissertação. Em primeiro lugar, são apresentados temas relacionados com tipografia: história, anatomia da letra, *legibility* e classificação tipográfica. No quarto capítulo (TRABALHO RELACIONADO) são expostos trabalhos e experimentações realizados no âmbito da tipografia e de processos generativos na criação de tipos de letra. No quinto capítulo (PROJETO PRÁTICO) é formalizada a proposta para o sistema a desenvolver e é apresentando o trabalho desenvolvido nesse sentido. No último capítulo (CONCLUSÃO) são apresentadas as conclusões retiradas durante o desenvolvimento do projeto assim como é apresentado o trabalho futuro com o intuito de aperfeiçoar o sistema desenvolvido.

2 — Plano de Trabalhos

018 — 021

2.1. Tarefas

Nesta secção foram identificadas as tarefas necessárias para ir ao encontro dos objetivos definidos. Esta secção foi essencial para fazer uma estimativa do tempo de duração de cada tarefa, e verificar dependências entre tarefas (Figura 1). Terminado o trabalho, foi feita a comparação com o tempo real necessário para cada tarefa (Figura 2). Assim, são apresentadas de seguida as tarefas agrupadas por tema: escrita da dissertação, experimentação, desenvolvimento do projeto e, por fim, aplicações e testes.

Escrita da dissertação

A escrita da dissertação documenta todo o processo desenvolvido durante o trabalho. Esta resume a investigação bibliográfica realizada — estado da arte — e documenta todo o trabalho desenvolvido no projeto prático. Esta tarefa divide-se em duas fases principais de acordo com a entrega intermédia e entrega final da dissertação.

Experimentação

A experimentação desde cedo se revelou uma necessidade neste trabalho, devido ao seu carácter experimental. Deste modo, em simultâneo com a pesquisa bibliográfica, foi desenvolvido um protótipo funcional do sistema, e em simultâneo com o desenvolvimento do projeto foi feita uma exploração de possíveis abordagens para a implementação de técnicas de desenho.

Desenvolvimento do projeto

Nesta tarefa é definido o conceito, as funcionalidades do sistema e é apresentado desenvolvimento do projeto.

Testes e aplicações

Esta tarefa apresenta os testes com utilizadores realizados ao sistema e visou procurar situações em que a o sistema desenvolvido pode ser aplicado, assim como realizar testes experimentais nas situações identificadas.

2.2. Metodologia

De forma a conseguir controlar e assegurar o progresso no desenvolvimento da componente prática desta dissertação, utilizamos o modelo em cascata introduzido por Winston Royce, em 1970. Este modelo consiste em realizar uma tarefa de forma correta antes de passar para à próxima, sendo possível voltar à tarefa anterior corrigindo possíveis erros (DUBBERLY, 2004).

Foi necessário adicionar a possibilidade de voltar a definir requisitos devido à iteratividade do processo criativo de exploração das técnicas de preenchimento dos esqueletos, como é possível verificar na Figura 3.

Figura 3
Representação do modelo cascata adotado para o desenvolvimento do projeto.

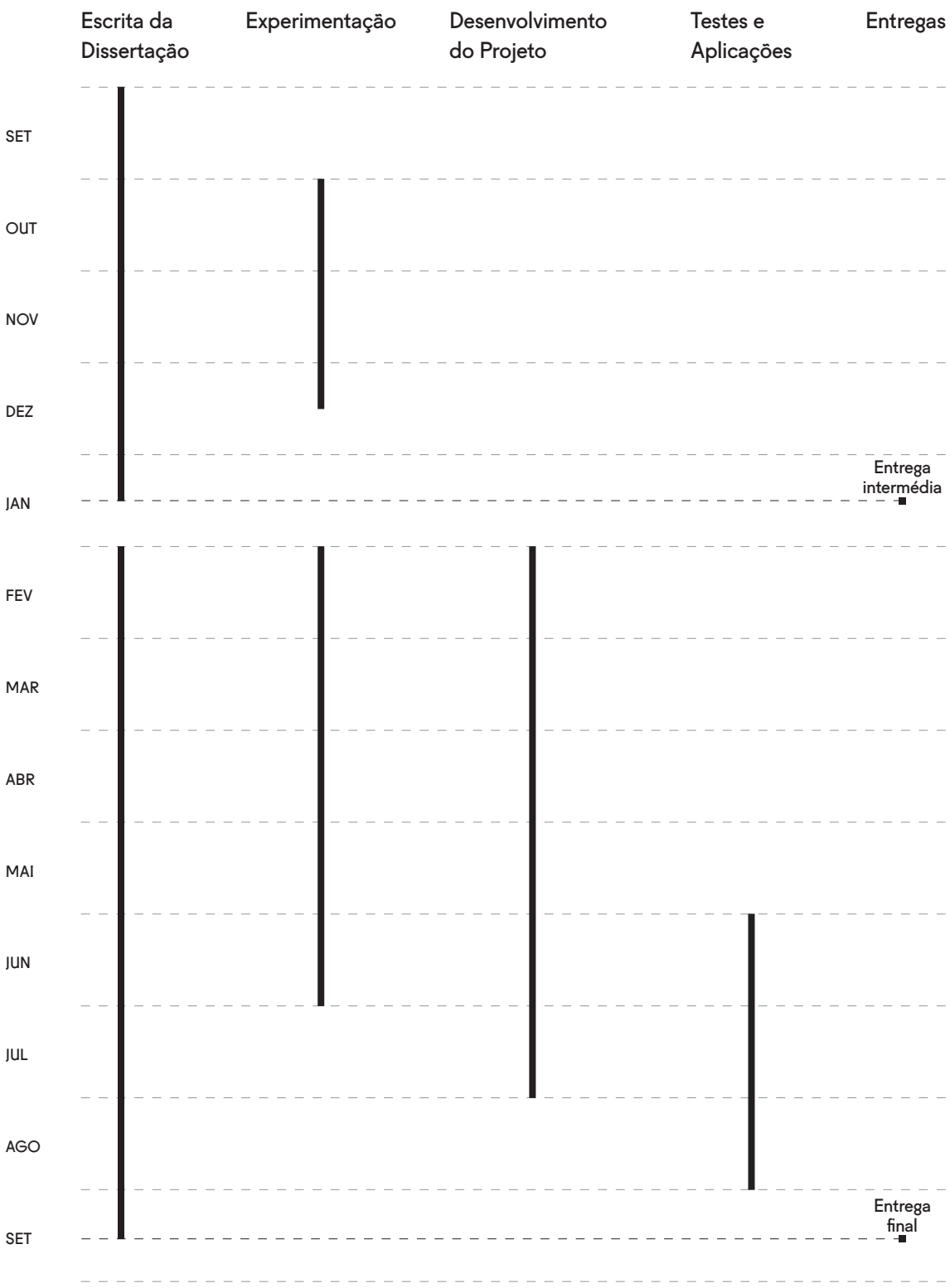


Figura 1
Diagrama de Gantt com respetivo tempo previsto das tarefas (versão do relatório intermédio).

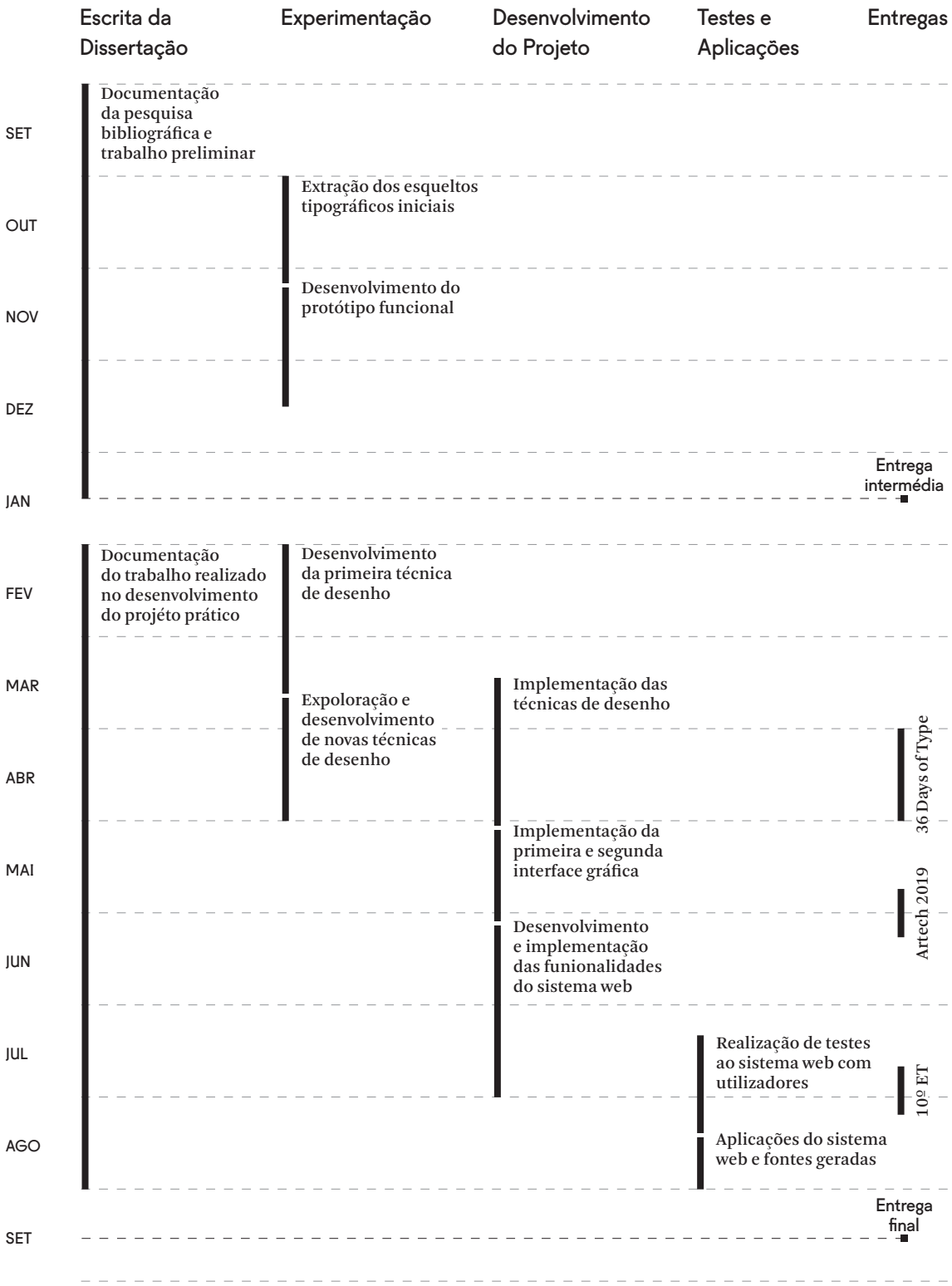


Figura 2
Diagrama de Gantt com respetivo tempo real das tarefas.

3 — Estado da Arte

022 — 037

No presente capítulo, é compilada toda a informação relevante ao tema da dissertação, recolhida na investigação bibliográfica realizada com o objetivo de reunir conhecimentos necessários para fundamentar o projeto prático. Sendo este um trabalho que visa a criação de tipos de letra é essencial analisar a história da tipografia com o intuito de perceber as origens da tipografia assim como as personalidades que mais influenciaram a sua evolução até aos dias de hoje. Assim como, os termos técnicos da anatomia tipográfica, a *legibility,* de forma entender como o ser humano perceciona as letras, e a classificação tipográfica — um dos temas mais controversos do debate tipográfico afirma Catherine Dixon. Para concluir, são apresentados trabalhos e experiências de artistas que já exploraram possibilidades de tipos de letra generativos nos seus trabalhos, de modo a ter conhecimento sobre o que já foi feito e de que forma este projeto se vem diferenciar.

3.1. Breve história da tipografia

> *A história da tipografia reflete uma tensão contínua entre a mão e a máquina, o orgânico e o geométrico, o corpo humano e o sistema abstrato.*
>
> LUPTON, 2004:13

Como Ellen Lupton afirma, esta tensão existente marca o nascimento das letras há mais de quinhentos anos, e perdura até aos dias de hoje, sendo necessária uma procura pelo estudo das relações entre as letras e todas as outras atividades humanas — política, filosofia, arte e a história das ideias (BRINGHURST, 2008; LUPTON, 2004).

Na década de 1450, surge na Alemanha a impressão com tipos móveis por obra de Johannes Gutenberg. Ainda que esta tenha sido inventada na China por volta de 1040, foi na Europa que encontrou a fértil história da letra romana e a sinergia entre as duas revolucionou a escrita ocidental. Este sucesso deveu-se ao número reduzido de caracteres das escritas europeias — mais apropriadas à mecanização — em oposição aos milhares de caracteres presentes na escrita chinesa, que possibilitou a produção em massa de livros e documentos (BRINGHURST, 2008; LUPTON, 2004).

As inscrições de maiúsculas gregas gravadas em pedra, são os vestígios de letras mais antigas na Europa (Figura 4). Estas apresentavam traços finos e esqueléticos, com linhas retas e aberturas muito grandes nas curvaturas — letras como S, C e M, tinham a maior abertura possível. Desenhadas à mão livre, não apresentavam serifas, nem qualquer outro detalhe. Mais tarde, os traços começavam a engrossar, a abertura diminuía e as serifas surgiram. Utilizadas para inscrições do império grego, estas novas formas (com abertura modesta, traço modulado e serifas formais), serviram para escrituras formais na Roma Imperial,

assim como serviram de modelos para calígrafos e designers de tipos ao longo dos últimos dois mil anos (BRINGHURST, 2008).

Figura 4
Trajan, desenhada por Carol Twombly em 1988, baseada na inscrição da base da Coluna de Trajano em Roma, gravada em 113 aC.

Entretanto, a escrita espalhou-se pela europa e por consequência surgiram vários alfabetos, assim como as maiúsculas/caixa-alta — letras grandes e formais — e minúsculas/caixa-baixa — letras mais pequenas e informais (BRINGHURST, 2008).

Por volta de 1469, Nicolas Jenson, veio influenciar a criação das letras renascentistas com a sua influente gráfica em Veneza. Os tipos de letra por ele criados, exibiam de uma forma exímia a simbiose entre as tradições góticas e o gosto italiano por formas leves e redondas (Figura 5). Por este motivo, são considerados uns dos primeiros — e mais elegantes — tipos de letra romanos (LUPTON, 2004).

Figura 5
Centaur, desenhada por Bruce Rogers por volta de 1914, a partir da obra de Nicolas Jenson.

Continuando em Itália, o estudioso e editor Aldus Manutius encomendou o primeiro tipo itálico a Francesco Griffo em 1499. As letras itálicas eram um estilo mais casual da caligrafia e nas suas primeiras versões possuíam apenas caixa-baixa. Manutius utilizou tipos itálicos na sua série de pequenos e económicos livros, pois a forma cursiva poupa espaço e por sua vez dinheiro na sua produção. Nesses livros, era comum o uso de letras itálicas com versais romanas, pois os estilos eram considerados bastante distintos. Mais tarde, no século XVI, começou-se a integrar as letras romanas e itálicas na mesma família tipográfica de acordo com os pesos e a altura-x (BRINGHURST, 2008; LUPTON, 2004).

Os artistas renascentistas acreditavam que as proporções das letras deviam refletir as proporções do corpo humano ideal. Em 1529, Geofroy Tory, publicou uma série de diagramas que relacionavam a anatomia das letras à anatomia do corpo humano (LUPTON, 2004).

Em 1693, um comité composto por dois padres, um escriturário e um engenheiro, desenvolveu a primeira fonte neoclássica a mando do rei Louis XIV de França (Figura 6). Conhecida como *Roman du Roi,* foi rigorosamente desenvolvida numa grelha que lhe proporcionou um caráter científico (BRINGHURST, 2008; LUPTON, 2004).

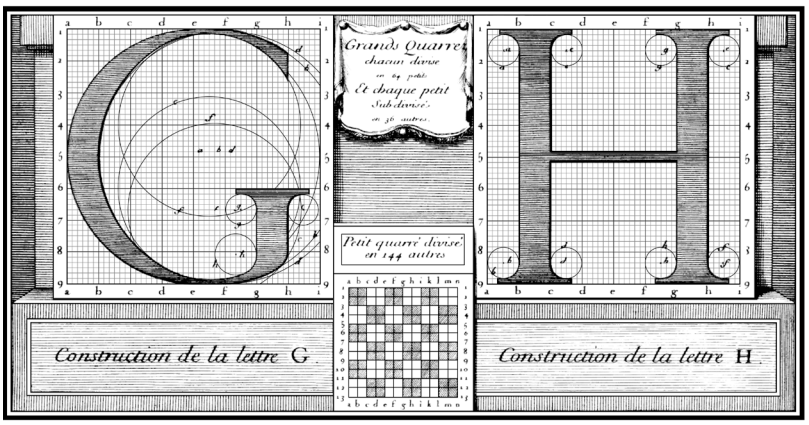


Figura 6
Romain du Roi.
Construção da letra G e H.

No século XVIII, a tipografia veio ser influenciada por novos estilos de caligrafia devido à substituição da pena larga pela pena pontiaguda e flexível. A pena larga deixa um traço suave cuja espessura varia com a direção, e o oposto acontece com a pena flexível, pois produz um traço com grande contraste devido à mudança da espessura de acordo com as variações de pressão.

William Caslon e John Baskerville, em 1720 e 1750 respetivamente, aderiram a esta nova influência e produziram tipos de letra de acordo com este novo estilo. Baskerville, mestre da caligrafia, chegou a ser acusado de "cegar" os leitores pela nitidez e contraste tão dramáticos que eram evidentes nos seus tipos de letra.

Este estilo veio a ser explorado ao extremo por Giambattista Bodoni em Itália e por Firmin Didot em França por volta do século XIX. Os tipos de letra por eles criados apresentavam como características: eixo completamente vertical; contraste extremo entre traço fino e grosso; e serifas horizontais. Na procura por uma beleza tão racional quanto sublime, a Bodoni e a Didot (Figura 7 e 8) vieram abrir novos horizontes na exploração tipográfica com a sua abordagem distante da humanização no desenho das letras (BRINGHURST, 2008; LUPTON, 2004).

Figura 7
Berthold Bodoni, desenhada a partir das fontes gravadas por Giambattista Bodoni.

abcdeoqri

Figura 8
Linotype Didot, desenhada por Adrian Frutiger, baseada nas fontes de Firmin Didot.

abcdeoqri

Até aos dias de hoje, muitas foram as tecnologias que proporcionaram progresso na forma de compor os tipos de letra para jornais, revistas, posters e outros meios de comunicação.

Uma delas, foi a máquina *Linotype* inventada por Ottmar Mergenthaler em 1880. Esta consistia principalmente, na fusão de uma máquina de fundir e uma máquina de escrever, onde era possível controlar uma série de mecanismos por um grande teclado mecânico. O sistema compõe uma linha de matrizes de acordo com os caracteres pressionados no teclado, e de seguida funde essa linha numa única forma metálica num processo denominado de *hot metal typesetting*.

O design de tipos para a *Linotype* era limitado por três fatores — *kerning* não era possível, eme dividido em apenas dezoito unidades e as matrizes itálicas e romanas precisavam da mesma largura —, ainda assim foram projetadas fontes como a Aldus e a Optima de Herman Zapf, entre outras, que obtiveram grande sucesso artístico (BRINGHURST, 2008).

Mais tarde, surge uma máquina que estampava as letras em metal frio e de seguida compunha as linhas, por obra de Tolbert Lanston em 1887. Esta rapidamente foi abandonada para dar lugar a uma máquina que moldava as letras individuais utilizando metal derretido, em 1900 por John Bancroft, conhecida como *Monotype*. Esta máquina era composta por um terminal e um dispositivo de saída. No terminal existia um teclado mecânico — contendo sete alfabetos —, que ao ser pressionado furava uma fita de papel por obra de um sistema de ar comprimido. Esta fita de papel era posteriormente colocada no dispositivo de saída, para ser lida e de seguida fundir e compor as letras. O design de tipos para a *Monotype*, era também limitado pelo eme dividido em dezoito unidades, no entanto as matrizes itálicas e romanas tinham larguras independentes e era possível fazer *kerning* (BRINGHURST, 2008).

> *The contribution made to typographic history by the casting-machine manufactures was not solely in the impact of their technological advances, but also in their commissioning of many of the most widely used and enduring typefaces of the twentieth century.*
>
> MCNEIL,2017:155

A *Monotype Corporation*, na década de 1920, criou um programa de pesquisa e reconstrução de alguns dos tipos de letra mais célebres de todos os períodos históricos, enquanto publicava novos tipos de letra de tipógrafos contemporâneos. Este processo de revivalismo tipográfico, teve grande impacto na melhoria dos processos de impressão e publicação tidos como norma durante a época (MCNEIL, 2017).

Outras tecnologias como a *Photocomposition* e *dry-tranfer lettering*, tiveram um papel de grande importância na evolução do processo de composição tipográfica. Estas novas tecnologias combinadas com a rápida, barata e eficiente impressão de litografia *offset*, proporciona-

ram novas abordagens na forma como a comunicação podia ser feita, mudando as principais atividades das tradicionais casas de impressão para os designers, tipógrafos e agências de publicidade. Estas alterações estimularam o crescimento da profissão de designer gráfico e deram controlo aos utilizadores sobre a origem da comunicação impressa levando diretamente à revolução digital dos anos 1980 e 1990 (McNeil, 2017).

3.2. Anatomia da letra

É indiscutível a relevância do estudo da forma das letras e as suas aplicações pelos mais diversos profissionais, que fazem delas uma das principais ferramentas de trabalho — designers gráficos, designers de tipos, editores, calígrafos e tipógrafos/impressores —, neste trabalho tendo em conta que a letra vai ser objeto de experiências, é necessário adquirir conhecimentos suficientes para a criação de tipos de letra de forma adequada.

> *Of all designed objects, letters are probably the most pervasive, very familiar yet amazingly diverse in their appearance.*
>
> Unger, 2018:11

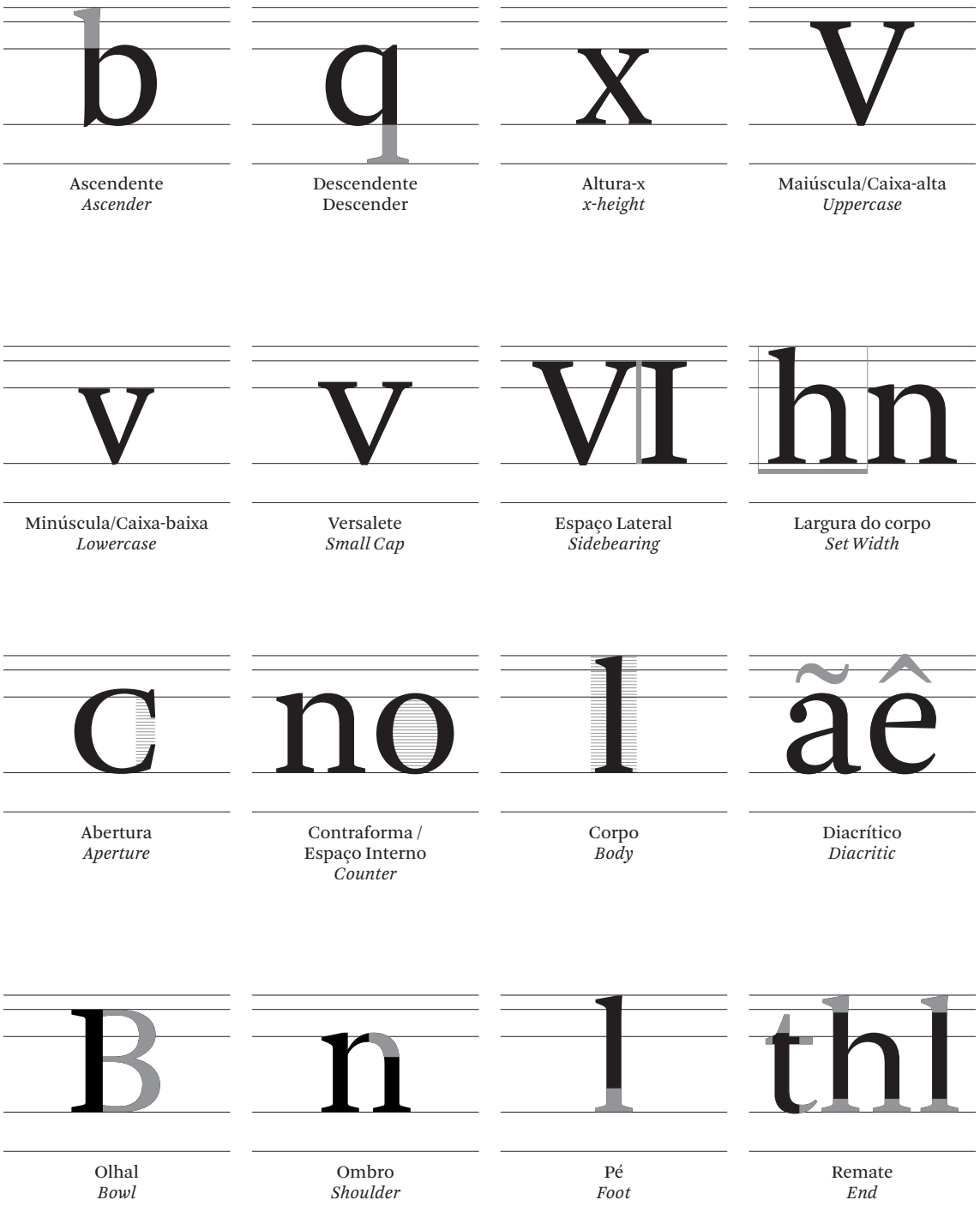
Como Unger refere, as letras são muito parecidas e ainda assim bastante díspares na sua aparência. Este facto deve-se às diferentes variações existentes nas partes anatómicas que constituem uma letra. Assim é essencial abordar os termos técnicos necessários para uma melhor compreensão da letra.

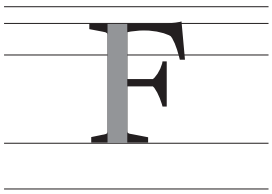
Na Figura 9 são apresentados os termos relativos às medidas que suportam a criação de tipos de letra e a composição tipográfica. A linha de base é o principal alicerce das letras pois é onde estas assentam e se guiam ao longo de um texto. A altura-x, é definida pela distância da linha de base até à altura do corpo principal da minúscula, descartando descendentes e ascendentes. A linha dos descendentes é definida pela distância da linha de base até ao limite do descendente de uma letra, o oposto acontece com a linha dos ascendentes, que é definida pela distância da altura-x até ao limite do ascendente de uma letra. A linha das maiúsculas é definida pela distância da linha de base até à altura do corpo da maiúscula.

Figura 9
Medidas da anatomia tipográfica. (adaptado de Amado & Silva, 2011).

Na Figura 10 estão representados os principais termos da anatomia da letra relevantes para o desenvolvimento deste trabalho. A melhor compreensão das características anatómicas, proporciona uma mais valia para a exploração de possíveis abordagens na componente prática.

Figura 10
Anatomia da letra. (adaptado de Amado & Silva, 2011).





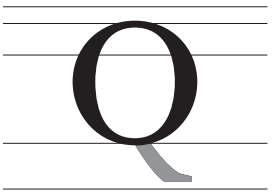
Traço
Stroke

Perna
Leg

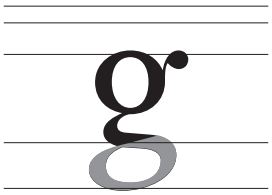
Barra Transversal/Travessão
Crossbar/Bar

Barriga
Bowl

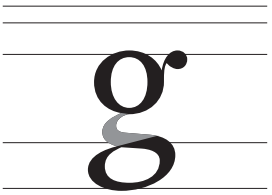
Braço
Arm



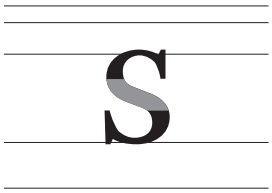
Cauda
Tail



Contracurva/Laço/Olhal
Loop



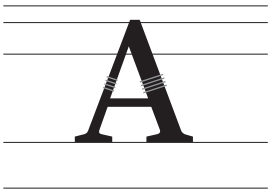
Pescoço
Link/Neck



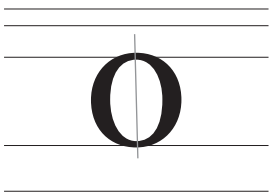
Espinha/Arco Duplo
Spine

Filete/Perfil
Hairline

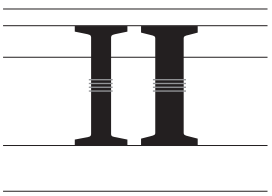
Ângulo
Stress / Bias



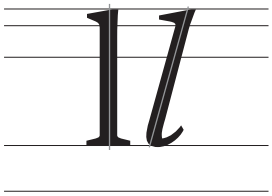
Contraste
Contrast



Eixo
Axis



Espessura/Peso
Stroke Width/Weight



Inclinação
Slope

Serifa
Serif

3.3. *Legibility* e *Kerning*

Dado o carácter experimental deste trabalho, é necessário entender como percecionamos e diferenciamos as letras, de forma a poder medir possíveis limitações das abordagens que irão ser exploradas no projeto prático, tendo em conta o propósito de gerar tipos de letras funcionais.

Devido à incoerência existente na tradução dos termos ingleses *legibility* e *readability* para português, decidimos adotar os termos ingleses. Muitas vezes os termos são confundidos. *Legibility* diz respeito ao design de uma fonte — formas dos glifos — e à capacidade de o leitor percecionar a relação das letras entre elas. Unger acrescenta ainda que, no campo sensorial e neuronal, *legibility* pode ser considerada como a facilidade com que os símbolos visuais podem ser descodificados. *Readability* diz respeito ao uso da linguagem, apresentação do texto e à capacidade de o leitor conseguir percecionar a relação das letras num texto (UNGER, 2018).

Existe o termo "leiturabilidade" como equivalente do termo inglês *readability*, no entanto, decidimos não o adotar pois não vai de encontro à prática.

> *Reading functions due to convection: the knowledge shared by readers of a particular script, the generally accepted forms of letters and other typographic signs.*
>
> UNGER, 2018:63

Gerard Unger afirma que este conhecimento, embebido no cérebro, aproxima a forma de ler a um processo automatizado, onde proporciona ao leitor facilidade no processo de reconhecimento das letras e das linhas de texto enquanto se concentra no seu conteúdo (UNGER, 2018).

Dos vários modelos propostos para o reconhecimento de *letterforms*, *feature detection theory,* parece ser o mais eficiente. Consiste no reconhecimento de detalhes (partes anatómicas) nas letras — linhas horizontais, verticais, oblíquas e curvas — e na posterior comparação com os detalhes presentes na nossa memória. Por exemplo, nos pares de letras "C-G" e "O-Q", existe uma grande similaridade e sobreposição de detalhes. O elemento que distingue o "C" do "G" é a linha horizontal (queixo) presente apenas no "G" e a linha oblíqua (cauda) presente apenas no "Q" distingue-o do "O" (RAYNER, POLLATSEK, ASHBY, & CLIFTON, 2012; UNGER, 2018).

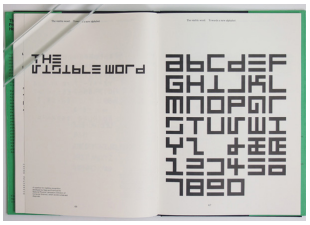


Figura 11
Páginas de *The Visible World: Problems of Legibility* de Herbert Spencer. É possível ver uma fonte à qual foram retiradas todas as formas curvas e diagonais para facilitar o reconhecimento óptico mecânico.

Deste modo, o modelo sugere que o cérebro consegue percecionar uma letra visualizando apenas alguns dos detalhes da mesma (Figura 11), o que é algo vantajoso para o desenvolvimento deste trabalho, pois permite experimentar abordagens mais abstratas na criação dos glifos. Este aspeto cognitivo foi explorado na criação da fonte Sans Forgetica (Figura 12). Desenvolvida por um conjunto de investigadores Australianos, do laboratório comportamental e da escola de design do *Royal Melbourne Institute of Technology,* faz uso do design e de teorias psicológicas para auxiliar na retenção de memória (MARTIN, 2018).

«*The mind will naturally seek to complete those shapes and so by doing that it slows the reading and triggers memory*» afirma Stephen Banham, tipógrafo e designer de tipos, numa entrevista ao jornal *The Guardian* em 2018.

Figura 12
Sans Forgetica, desenhada com o intuito de melhorar a memória.

> *(...) type design involves creating all the signs necessary for the composition of legible language on screen, paper, or any other surface, such as walls or windows. All the strokes and parts, and all the signs built from these, function together as an entity; it has been said that the aim of the type designer is 'to create a beautiful group of letters and not a group of beautiful letters'.*
>
> UNGER, 2018:99

Legibility é conseguida através de um grupo de letras que, como Unger refere, funcionem como uma identidade onde todas as partes das letras e todos os outros símbolos tipográficos estejam bem organizados, de acordo com os atributos inerentes — estrutura, padrão, textura, consistência e coerência, ritmo e espaço (UNGER, 2018).

> *In the 1990s, design had its famous 'legibility wars'. Graphic designers discovered that the new computer technology made it easy to manipulate type and to layer text upon text, and the text with imagery.*
>
> SHAUGHNESSY, 2009:175

David Carson foi um dos primeiros a usufruir destas novas tecnologias. Marcou o design gráfico dos anos 1990, por adotar práticas que iam contra os princípios de *legibility* e *readability.* Muito ligado à cultura do surf, Carson emergiu como diretor de arte da revista *Beach Culture,* mas foi na revista *Ray Gun* que ficou internacionalmente conhecido pelo seu trabalho. Numa abordagem mais liberal e experimental ao design, muitos dos seus trabalhos apresentam textos cortados sem entrelinha e letras recortadas praticamente ilegíveis (Figura 13).

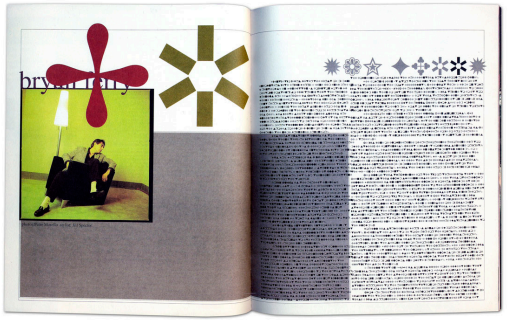


Figura 13
Páginas de revista Ray Gun compostas por David Carson, que por achar o artigo de Bryan Ferry aborrecido usou a fonte Zapf Dingbats, um tipo de letra, que é composto apenas por símbolos, em vez de letras.

Numa entrevista ao jornal *The Australian* em 2014, Carson afirma: «*communication begins way before any actual reading*» e é tendo isto em conta, que procura sempre atrair as pessoas, não apenas para ler o livro ou revista, mas principalmente para suscitar o seu interesse. Para o fazer, recorre muitas das vezes ao uso de citações incomuns e ao design, começando sempre por ler o artigo ou material que seja, e desta forma fazer passar a sua interpretação no trabalho final. Carson influenciou fortemente as gerações jovens da época, e a tipografia contemporânea com o teu trabalho exuberante e fora do comum (PAWLE, 2014; SHAUGHNESSY, 2009).

Legibility não é apenas afetada pelas características da letra, mas também — e não menos importante — pelo espaço entre elas e à sua volta. Para controlar estes espaços existe o *kerning* que muitas vezes é confundido com o *tracking*. *Kerning* consiste em ajustar (aumentar ou diminuir) o espaço entre pares de letras específicos, para assegurar uma aparência uniforme e consistente, de forma a que os olhos do leitor não sejam distraídos pelas brechas ou colisões de espaços presentes nas palavras — por exemplo, nas palavras Washington e Avenue, os pares Wa e Av, necessitam de *kerning* (Figura 14). *Tracking* refere-se ao controlo de todos os espaços entre letras num dado bloco de texto, deste modo, quando é alterado, afeta todos estes espaços de forma igual (BRINGHURST, 2008; SHAUGHNESSY, 2009).

Kerning surge da época da composição manual com os tipos móveis de metal, em que algumas letras — tipicamente as letras l e f — excediam o bloco de metal, criando combinações de letras desagradáveis. Com a introdução do computador, na composição digital, o espacejamento nos tipos digitais é possível de ser refinado para além do que era possível com os tipos de metal, e para o alcançar, são utilizadas tabelas de *kerning*. Estas tabelas, têm como função, geralmente, retirar espaço numa série de combinações de letras predefinidas — por exemplo, nas combinações Av, Aw, Ay, 'A, A', L' ou nos casos em que a primeira letra é T, V, W ou Y e a segunda é, a,c,d,e,g,m,n,o,p,q,r,s,u,v,-w,x,y ou z. É claro que nem todas estas combinações ocorrem em português (ou inglês), mas uma boa tabela de *kerning* deve acomodar sem problemas palavras como Tchaikovsky, Tmolos, Tsimshian, Vázquez, Chateau d'Yquem e Ysaye (BRINGHURST, 2008; SHAUGHNESSY, 2009).

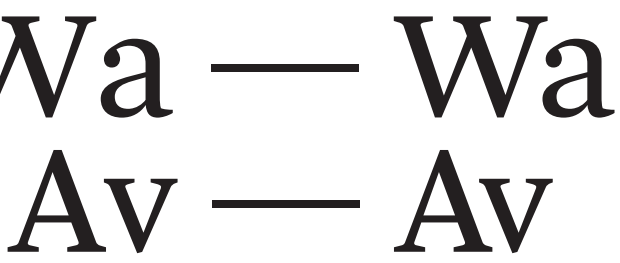


Figura 14
Diferenças entre pares de letras sem *kerning* aplicado (pares à esquerda) e pares de letras com *kerning*.

3.4. Classificação tipográfica

Em 2012, Indra Kupferschmid, escreveu um artigo para publicação na conferência *Research in Graphic Design*, no qual afirma a tendência existente para organizar tudo o aquilo que exista em grandes quantidades, de forma a poder referenciar e encontrar de novo mais facilmente. O mundo da tipografia não é exceção, e revela-se uma necessidade (KUPFERSCHRIFT, 2012; TRACY, 1971).

> *The categorisation of typeforms grew out of a changing climate in production, when, during the c19th, printers experienced a broadening in the range at their disposal.*
>
> DIXON, 2002

O processo de classificar tipos de letra é relativamente novo, pois nos primeiros 400 anos da história da tipografia, os tipos de letras eram chamados de acordo com o seu tamanho — *Paragon, Great Primer, Nonparaille*. Na hipótese de existir mais que uma versão romana do mesmo tipo de letra, esta era numerada, por exemplo *Great Primer Roman No.2*. Entretanto surgiu a revolução industrial e consigo trouxe o desejo por tipos de letra mais atraentes. Por consequência, cresceu o número de variações de tipos de letra e com eles a necessidade de lhes dar nome de forma a serem divulgados e facilitar a comunicação entre os impressores e os clientes (DIXON, 2002; KUPFERSCHRIFT, 2012).

Desta forma, as *type foundries* começaram a inventaram os nomes para descrever os novos estilos: *Egyptian, Gothic e Grotesque* para os tipos de letra sem serifa*; Ionic, Doric e Antique* para os tipos *slab-serifs*. No entanto, não existia consistência dos termos utilizados pelas *type foundries*, por exemplo, o termo *Egyptian* integrava o grupo dos tipos sem serifa, ainda que hoje seja mais associado ao grupo dos tipos *slab-serifs* (DIXON, 2002; KUPFERSCHRIFT, 2012).

Assim, começaram a aparecer os primeiros sistemas com o propósito de solucionar esta ambiguidade, e apresentar classificações que pudessem ser aplicadas universalmente.

Em 1921, Francis Thibaudeau, propôs uns dos primeiros sistemas, que se baseava apenas na forma das serifas (Figura 15), assim como Aldo Novarese fez em 1964. O sistema constituído por quatro classes — *Elzevir* (serifas triangulares), *Didot* (serifas filiformes), *Egyptienne* (serifas quadrangulares) e *Antiques* (sem serifa) — apesar de não ser muito completo veio trazer mais clareza na forma de distinguir os tipos de letra para a época em que foi apresentado, pois a serifa era das características mais facilmente distinguível nos tipos de letra existentes (COSTA, 2013; KUPFERSCHRIFT, 2012).

> *Vox's proposals provided a focus for classificatory ideas like no others before (...)*
>
> DIXON, 2002

Figura 15
Uma das primeiras tentativas de classificação tipográfica por Francis Thibaudeau.

Posteriormente, em 1954, Maximilien Vox parte do sistema Thibaudeau para desenvolver o seu próprio modelo, sendo bastante sofisticado para a época afirma Catherine Dixon. Em adição ao sistema Thibaudeau, Vox repartiu a classe Elzevir em três grupos — Humanes, Garaldes e Réales — tendo em conta a contribuição dos tipógrafos mais icónicos para a atribuição dos termos a cada grupo — por exemplo, o termo *Didone* provem dos tipos de letra Didot e Bodoni (COSTA, 2013; KUPFERSCHRIFT, 2012).

> *This system, which crystalized contemporary discussion of the issue, proved — despite a certain whimsy in its nomenclature — to be a more workable tool than anything that had been conceived of before.*
>
> KINROSS, 2004:97

Vox procurou abordar as semelhanças entre os tipos de letra e também a possibilidade de se diferenciarem individualmente. Assim, apresenta a classificação Vox que contém nove classes — *Humanes, Garaldes, Réales, Didones, Mécanes, Linéales, Incises, Manuaires e Scriptes.* (COSTA, 2013; DIXON, 2002)

Em 1962, a *Association Typographique Internationale* (ATypI) adotou a classificação de Vox como standard — *Vox-ATypI*—, já com as duas últimas classes adicionadas por Vox à sua classificação — *Fraktur* (*Black letter*) e *Non-Latin* (COSTA, 2013).

Para esta dissertação decidimos optar pela classificação *Vox-ATypI,* pois este sistema reflete uma análise histórica da tipografia, aos atributos formais das letras, aliado ao facto de ser o sistema em vigor internacionalmente. Decidimos dividir as classes em três grupos de acordo com as suas características, excluindo as *Non-Latin,* pois neste trabalho apenas iremos trabalhar com o alfabeto latino.

GRUPO A

Adobe Jenson Pro
Centaur MT Std
Perpetua Std

Na classe *Humanes,* é característico o contraste moderado entre traços horizontais e verticais, eixo humanista (oblíquo) consistente e ainda ascendentes com serifas abruptas. Podem ser facilmente reconhecidas pela característica inclinação no braço do "e" (COSTA, 2013; PARENTE, 2018).

Ex: Jenson, Véronèse, Centaur, Perpetua

Bembo Std
Adobe Garamond Pro
Adobe Caslon Pro
Sabon LT Std

A classe *Garaldes* deve o seu nome ao trabalho de Claude Garamond e Aldus Manutius.É possível encontrar traços dos movimentos Barroco e Maneirista, que atenuam traços da caligrafia, apresentando maior contraste. Ao contrário das humanistas, as fontes desta classe deixam de ter o braço do "e" inclinado e o eixo do "o" passa a ser só ligeiramente inclinado para a esquerda (COSTA, 2013; PARENTE, 2018).

Ex: Bembo, Garamond, Caslon, Sabon

Baskerville
Cochin Std
Times

A classe das *Réales* faz a ponte entre as *Garaldes* e *Didones, ou seja,* acolhe as fontes que não se encaixam em nenhuma das duas. São influenciadas pelo neoclassicismo e pelos ideais racionalistas, apresentando por consequência disto, o eixo totalmente vertical e alto contraste. Tendo como principal diferença das *Didones*, as serifas que não são horizontais (COSTA, 2013; PARENTE, 2018).

Ex: Baskerville, Cochin, Times

Didot LT Std
Bodoni Std

As *Didones* provêm do trabalho de Firmin Ditot e Giambattista Bodoni, inspirando muitos outros designers de tipos. Inspiradas nos ideais expressivos do Romantismo, estes tipos de letra apresentam eixo totalmente vertical, larguras uniformes e contraste extremo (COSTA, 2013; PARENTE, 2018).

Ex: Didot, Bodoni, Walbaum, Falstaff

As *Mécanes* surgem por consequência da revolução industrial no século XIX. Foram feitas experiências nunca antes feitas, de onde nasceram de tipos de letra com serifas quadrangulares e bastante grossas (COSTA, 2013; PARENTE, 2018).

Ex: Clarendon, Ionic, Rockwell

Clarendon LT Std
Rockwell

GRUPO B

Também no século XIX, começaram a surgir as *Lineáles,* associadas à máquina e ao modernismo, representavam todos os tipos de letra sem serifas, com maior ou menor contraste. Com o aparecimento da Helvetica e da Univers, ambas em 1957, os tipos de letra sem serifas ganharam maior relevância e reputação (COSTA, 2013; UNGER, 2018).

Ex: Helvetica, Akzidenz-Grotesk, Univers, Futura

Helvetica
Akzidenz-Grotesk BQ
Univers LT Std
Futura Std

GRUPO C

As *Incises* demonstram traços característicos das inscrições em lápides — utilizando um cinzel ou um instrumento similar —, por este motivo, possuem serifas pequenas e afiadas ou com hastes afuniladas (COSTA, 2013; UNGER, 2018).

Ex: Optima, Copperlate, Gothic

Optima
COPPERPLATE

As *Manuaires* representam os tipos de letra que são baseados no desenho feito à mão com o punho sobre a mesa — utilizando caneta, pincel ou qualquer outro instrumento de desenho — com o propósito de serem usados em títulos e não em texto corrido(COSTA, 2013; UNGER, 2018).

Ex: Banco

BANCO STD

A classe das *Scriptes* representa os tipos de letra que imitam o manuscrito rápido — caligráfico ou comum — realizado com o punho levantado. Encaixam-se nesta classe, apenas os tipos em que as letras se conectam, caso contrário dizem respeito às *Manuaires* (COSTA, 2013).

Ex: Mistral, Zapfino, Arrighi

Mistral
Zapfino

As *Fraktur* representam o início da impressão na Europa, e são essencialmente letras desenhadas com uma caneta de ponta larga. Apresentam tipicamente caracteres quebrados e formas pontiagudas e redondas (COSTA, 2013).

Ex: Fraktur, Berliner Fraktur

Fette Fraktur LT Std

Até ao dia de hoje foram muitos os sistemas de classificação que foram surgindo, abordando novos paradigmas, contextos, culturas e movimentos artísticos. Com a intenção de trazer ordem e clareza à enorme variedade de tipos de letra existentes, muitas das vezes acabam por não o conseguir de forma exímia.

All these systems work to a certain extent, but all leave much to be desired. They are neither good science nor good history.

BRINGHURST, 2008

As classificações são, em grande parte, mais teóricas do que práticas acabando por ser pouco abrangentes e complicadas, mais ainda para quem não estuda tipografia. O sistema adotado não é exceção, e Dixon critica o facto deste sistema dar vantagem aos tipos Romanos em relação aos tipos *display* — que cada vez mais pouco se diferenciam dos tipos para texto. Aponta ainda, o nível de inconsistência descritivo entre as classes. Além disso, a nomenclatura tem sido um dos mais controversos temas na área do design de tipos, incentivando a novas pesquisas e revisões dos termos existentes, para que seja possível trazer consistência e clareza aos mesmos (DIXON, 2002; UNGER, 2018).

Desta forma, são cada vez mais as tentativas de solucionar estes paradigmas, com classificações inovadoras e mais abrangentes. Dixon apresenta a sua abordagem aos 550 anos de design de tipos num sistema descritivo capaz de abranger os tipos contemporâneos na prática do design antecedente, e com a finalidade de facilitar a comunicação desta nova estrutura a utilizadores com muito ou pouco conhecimento sobre tipos de letra.

Mas assim como todos os outros sistemas, este não deve ser considerado como completo e deverá estar em constante atualização para corresponder à prática contemporânea (DIXON, 2002).

Trabalho Relacionado

Neste capítulo são apresentados sistemas *web* similares à componente prática que irá ser realizada. São expostas as principais funcionalidades e também uma análise crítica das mesmas. São também expostos trabalhos e experimentações realizadas no âmbito da criação de tipos de letra, quer sejam mais manuais ou generativas, que de alguma forma podem influenciar as nossas abordagens na componente prática.

Antes de prosseguir, é necessário perceber o conceito de fontes variáveis e fontes paramétricas assim como as suas diferenças. As fontes variáveis são as que permitem que um único ficheiro de uma fonte se comporte como múltiplas fontes, com uma infinita variedade de possíveis pesos, larguras e outros atributos. Estas variantes, são definidas pelo designer da fonte. Por outro lado, as fontes paramétricas são definidas de forma programática. Os parâmetros — por exemplo, a altura-x, largura do traço e largura da letra — são parâmetros numéricos que podem ser modificados e por consequência uma nova fonte é gerada de acordo com esses parâmetros (BRATH, 2017; RIECHERS, 2018).

4.1. Sistemas *Web*

Ntype é uma ferramenta *web* de comunicação tipográfica 4D. Foi desenvolvida por Kevin Zweerink em 2015, e consiste na exploração de extrusões de *letterforms* de acordo com os parâmetros dados pelo utilizador — eixos de rotação, velocidade rotação, comprimento do rasto e desenhar a forma, o rasto ou ambos (Figura 16). Utiliza WebGL e Three.js para renderizar as *letterforms* e opentype.js para guardar as mesmas como fonte do tipo OTF e fazer *download.* Permite ainda partilhar o estado das rotações, por via URL (ZWEERINK, 2015A, 2015B).

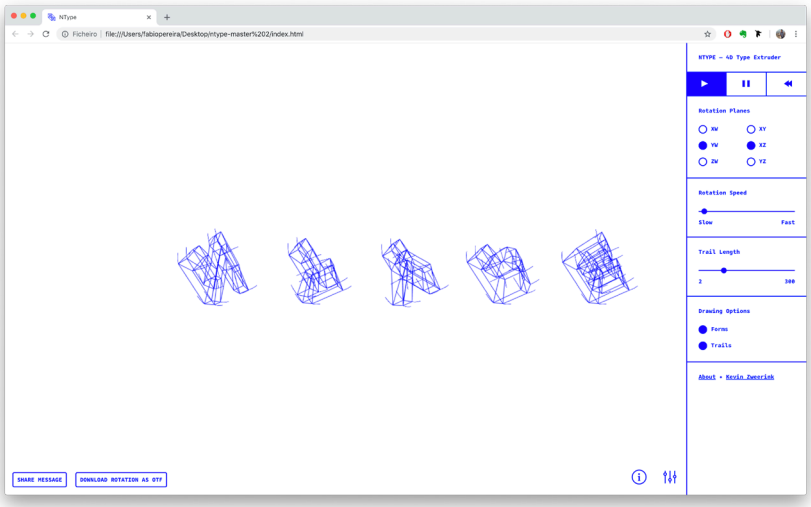


Figura 16
Sistema web, Ntype, desenvolvido por Kevin Zweerink em 2015.

Esta ferramenta explora a visualização da *letterforms* em 4D, aspeto que normalmente é pouco trabalhado na área da tipografia. Fornece uma extensa variedade de controladores que afetam a renderização

das *letterforms,* o que torna a ferramenta bastante dinâmica e interessante para o utilizador explorar e brincar com os parâmetros, obtendo resultados a seu gosto. A partilha dos resultados via URL, ou download como fonte do tipo OTF, dão outra dimensão à ferramenta, uma vez que, podem ser utilizados noutros meios — por exemplo, no *Adobe InDesign*. Apresenta uma interface bastante clara, e deste modo, torna explícito ao utilizador que funcionalidades tem a seu dispor. No entanto, não é muito óbvio que existe a opção de mudar as *letterforms* a serem renderizadas através do teclado.

Phase é um sistema que explora o conceito de tipos de letra generativos, concetualizado por Elias Hanzer e desenvolvido por Florian Zia (Figura 17). Consiste na escolha de um módulo, entre os vários à escolha, e na posterior parametrização das características associadas. Este sistema utiliza a tecnologia de fontes variáveis, podendo ser manipulado em tempo real — pelos parâmetros — e também pelo *input* de som, o que possibilita um infinito número de formas. Permite ainda, alterar os valores dos parâmetros de forma aleatória e exportar a fonte como ficheiro TTF (Figura 18) (HANZER, N.D.).

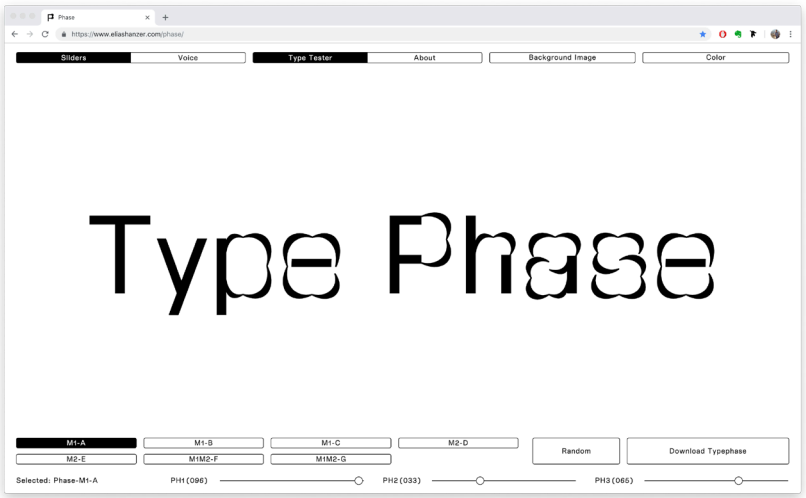


Figura 17
Sistema web, Phase, desenvolvido por Elias Hanzer e Florian Zia.

Este sistema alia, de uma forma bem conseguida, o processo generativo e a criação de tipos de letra, pois proporciona ao utilizador a capacidade de "desenhar" a sua própria fonte tendo apenas de controlar valores, que o sistema aplica de forma automática. No entanto, os parâmetros dados ao utilizador para poder experimentar, são iguais para qualquer um dos módulos escolhidos, e também não têm qualquer tipo de identificação nem explicação, relativamente às características que alteram nas letras. Ainda que as alterações sejam feitas em tempo real, o utilizador pode ficar frustrado por não perceber o que está a modificar — algo que poderia ser facilmente ser resolvido se os nomes dos parâmetros fossem ao encontro dos detalhes que al-

Figura 18
Fonte do formato TTF, obtida através do sistema Phase.

teram. Apesar disto, possui um botão para alterar todos os parâmetros incluindo o módulo de forma aleatória, algo que é bom quando o utilizador não quer resultados com grande precisão. A possibilidade de exportar a fonte, torna esta ferramenta ainda mais completa e dinâmica.

Metaflop é uma aplicação *web* para a criação personalizada de fontes, desenvolvida por Marco Müller e Alexis Reigel (Figura 19). Consiste na escolha de uma das fontes predefinidas, e na posterior parametrização das características anatómicas que lhe são inerentes —para este efeito utiliza a linguagem *metafont.* Dá acesso a um esquema dos parâmetros, assim como à pré-visualização do glifo escolhido, do gráfico de todos os glifos e de uma área onde é possível escrever texto personalizável. Permite alterar os valores dos parâmetros de forma aleatória (*flop it*), voltar ao estado anterior dos valores (*undo*), voltar ao estado inicial (*reset*) e ainda escolher a forma de alterar os valores, quer por *sliders* ou *inupt* numéricos. Possibilita também a exportação da fonte criada como ficheiro OTF ou como ficheiro WOFF (Figura 20) e a partilha da mesma via Twitter, Facebook, email e URL (REIGEL & MÜLLER, 2012).

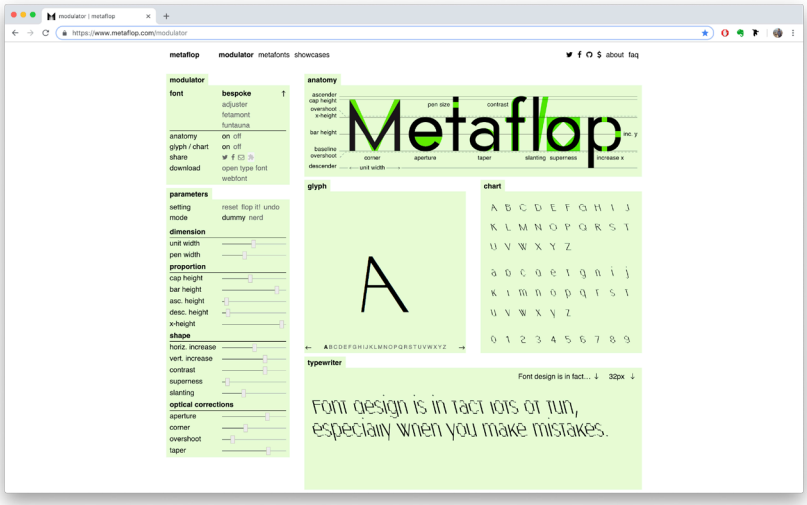


Figura 19
Sistema web, Metaflop, desenvolvido por Marco Müller e Alexis Reigel.

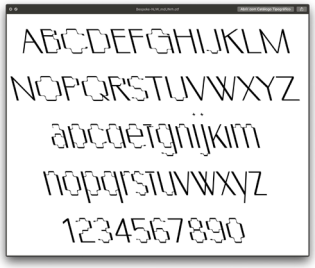


Figura 20
Fonte do formato OTF, obtida através do sistema Metaflop.

A aplicação possibilita uma grande variedade de resultados sem que seja necessário lidar com programação, o que é bastante vantajoso utilizador pois este, apenas têm de se preocupar com o aspeto visual que deseja para a fonte. O facto de fornecer um esquema dos parâmetros, ajuda o utilizador a ter mais consciência do que está a modificar quando altera os valores, assim como o facto de todos os parâmetros terem um nome descritivo das características que alteram, o que torna a interface bastante clara. A visualização em tempo real das modificações e os botões *flop it*, *undo* e *reset* tornam a aplicação mais dinâmica e interessante no que diz respeito à experimentação — que passa disso quando é feito o *download* da fonte. Aliados a todos estes fatores positivos, vem o facto da aplicação ser *open source,* que é bastante vantajoso para a evolução da mesma ou de outras aplicações.

Prototypo é uma aplicação que utiliza a tecnologia das fontes paramétricas para criação de fontes com inúmeras variantes. Consiste na escolha de uma das fontes modelo, e na posterior modificação das características — como altura-x, largura da letra, contraste, eixo, serifas, entre outras — com um conjunto de *sliders* que representam estes parâmetros e a pré-visualização das modificações em tempo real (Figura 21). Posteriormente ainda permite editar os glifos individualmente para adicionar mais detalhe ou corrigir imperfeições. Fornece a possibilidade de criar vários pesos e ainda é possível exportar a fonte, no entanto é necessário pagar uma subscrição para o fazer (MATHEY, 2009).

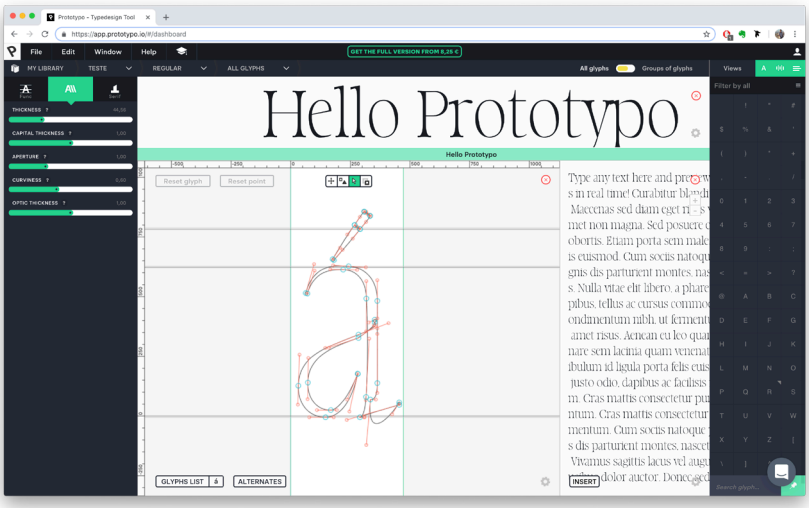


Figura 21
Sistema web, Prototypo. Escolha das características anatómicas da letra iniciais para a fonte.

Esta aplicação é bastante complexa e permite uma grande variedade de funcionalidades ao utilizador, o que pode tornar a interface da mesma um pouco confusa. Da mesma forma que as outras aplicações, esta também possibilita uma grande variedade de resultados sem que seja necessário lidar com programação ou necessário ter conhecimentos em tipografia. A fonte criada contém, para além das letras, os sinais de pontuação necessários para que seja viável, o que é uma grande vantagem em relação às aplicações anteriores. A possibilidade de editar os glifos individualmente é algo favorável, pois permite que a fonte seja totalmente do gosto do utilizador. A opção de criar vários pesos aumenta ainda mais a qualidade desta aplicação, pois o utilizador consegue criar todas as variações que necessita e já não precisa de recorrer a outras fontes. Também é possível agrupar glifos, e apenas aplicar alterações a estes.

4.2. Tipos de letra experimentais

Com a chegada da revolução industrial no século XIX, cresceu a procura por novos tipos de letra, mais atraentes e extravagantes, para colmatar as necessidades do consumo em massa e da publicidade. Rapida-

mente nasceram tipos de letra que resultavam da distorção de alguns dos elementos anatómicos das letras clássicas — por exemplo, peso, haste, serifas, ângulos, ascendentes, descendentes —, nomeadamente as serifas, que deixaram de ser um acabamento para dar vez a uma estrutura arquitetónica independente (Figura 22). Por consequência desta prática, a relação entre as letras fonte tornou-se mais importante do que a identidade das letras individualmente (LUPTON, 2004).

No entanto, nem todos os designers iam ao encontro a esta ideologia ligada à industrialização e consideravam a distorção do alfabeto algo grosseiro e imoral. Um deles foi Edward Johnston, que em 1906 alertou para os perigos do exagero na criação de novos tipos de letra e reanimou a procura por um alfabeto essencial inspirando-se no Renascimento e na idade Média, onde encontrava letras puras e não corrompidas (Figura 23) (LUPTON, 2004).

Figura 22
À esquerda. Uma dúzia de fontes é utilizada para a composição de um cartaz tipográfico em 1875, fazendo promoção a um cruzeiro a vapor.

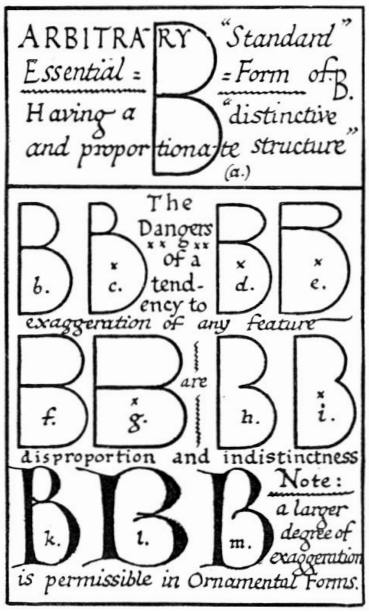


Figura 23
À direita. Diagrama de caracteres essenciais de Edward Johnston criado em 1906.

Figura 24
Vilmos Huszár desenhou o logotipo para a revista De Stijl em 1917.

Muitos foram os artistas da vanguarda do início do século XX que rejeitaram as formas históricas e adotaram uma posição de crítica. Alguns destes artistas, começaram a realizaram experiências olhando para o alfabeto como um sistema de relações abstratas. Exemplo disto são, os membros do grupo *De Stijl* na Holanda que reduziram o alfabeto a elementos perpendiculares (Figura 24).

Assim como, em 1925, surgiu na Bauhaus o interesse por um alfabeto universal constituído por formas geométricas básicas — círculo, quadrado e triângulo — que consideravam de uma linguagem universal da visão. Sendo os exemplos mais conhecidos, os alfabetos criados por Herbert Bayer (Figura 25) e Josef Albers (KINROSS, 2004; LUPTON, 2004).

Cativado pelas obsessões da vanguarda, Paul Renner em 1927, desenhou a Futura para a *Bauer type foundry* (Figura 26). Uma fonte

geométrica com algumas variações subtis nos traços, curvas e proporções. Esta satisfez o desejo pela geometria, assim como, por um tipo de letra que ficava bem em texto corrido, numa grande variedade de pesos. Desta forma, veio a ser considerada pelos designers de tipos da época como a mais satisfatória das fontes sem serifas (KINROSS, 2004; LUPTON, 2004).

Figura 25
Alfabeto criado por Herbert Bayer na Bauhaus em 1925, denominado de Universal.

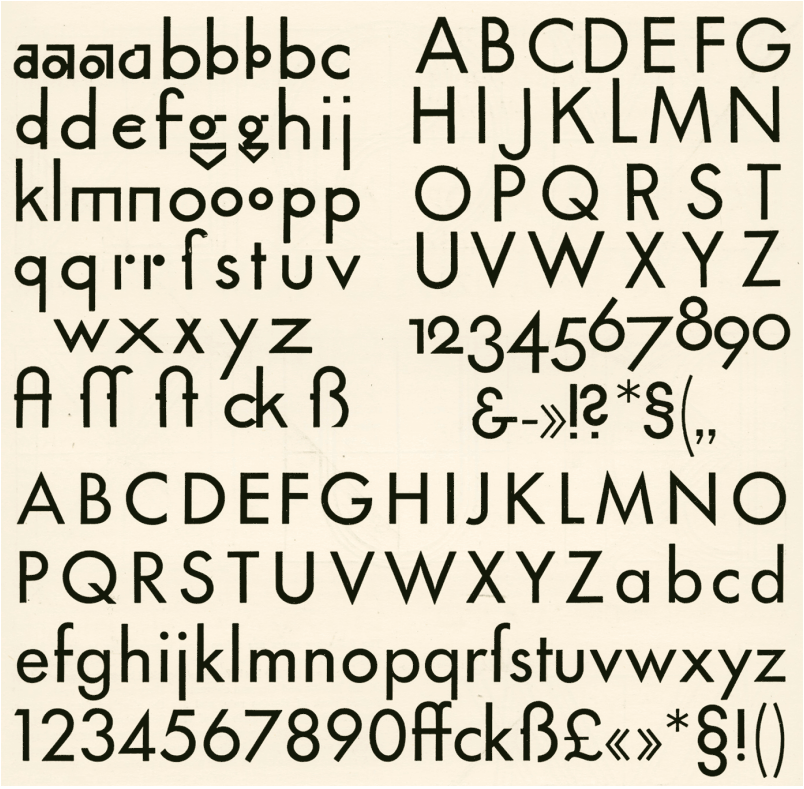


Figura 26
Futura desenhada por Paul Renner em 1927. As duas primeiras ilustrações mostram os primeiros desenhos feitos para a Futura. A terceira ilustração mostra o desenho final.

Em resposta à rápida ascensão da comunicação eletrônica, o designer holandês Wim Crouwel, apresentou em 1967 o "novo alfabeto", no qual as letras eram apenas construídas com linhas retas, com a finalidade de obter a melhor nas telas de vídeo (Figura 27). Desta forma, veio adaptar o seu processo de design às novas tecnologias e não o oposto (LUPTON, 2004).

Figura 27
New Alphabet, desenhado por Wim Crouwel em 1967.

Por volta de 1980, o aparecimento de computadores pessoais e impressoras de baixa resolução vieram proporcionar as ferramentas ideais para novas e inovadoras experiências no campo da tipografia. Em 1985, Zuzana Licko começou a explorar estas novas ferramentas, explorando o pixel como módulo — fontes *bitmap* — para construir novos tipos de letra que se ajustavam aos ecrãs da época assim como às impressoras. Juntamente com o seu marido, Rudy Vanderlands, fundaram a revista *Emigre,* e por este motivo, ficaram conhecidos como pioneiros da tecnologia no mundo tipográfico (LUPTON, 2004).

Com introdução das tecnologias no processo de design por volta dos anos 1990, cresceu a insatisfação de muitos designers com as letras limpas e puras e então deu-se início à exploração mais abstrata e rude. Exemplos disso, são os trabalhos expostos de seguida.

Em 1990, Barry Deck desenhou a fonte *Template Gothic*, enquanto aluno de Edward Fella (Figura 28). Baseada em letras *stencil* feitas de plástico (da lavandaria do seu bairro), apresenta um processo ao mes-

mo tempo mecânico e manual. Foi lançada comercialmente pela *Emigre Fonts* e desta forma, tornou-se numa das mais importantes fontes digitais dos anos 90 (LUPTON, 2004).

Figura 28
Template Ghotic, desenhada por Barry Deck em 1990.

Também em 1990, P.Scott Makela desenhou a fonte *Dead History* (Figura 29), misturando duas fontes existentes, Centenial e a VAG Rounded, numa abordagem de apropriação já existente em outras áreas artísticas — por exemplo, na arte e na música. Ao fazer isto, Makela pretendia aludir à importância da história no processo da inovação tipográfica (LUPTON, 2004).

Figura 29
Dead History, desenhada por P.Scott Makela em 1990.

Ainda nos anos 1990, os designers Erik van Blokland e Just van Rossum descobriram uma forma de modificar o código das fontes *PostScript,* e por consequência criaram a fonte Beowolf (Figura 30). Quando enviada para imprimir, esta altera, de forma aleatória, os pontos de cada letra acabando por ficar com uma aparência deformada — que nunca é igual. Com o nome original de "*RandomFont*", a fonte chamou a atenção da *FontFont*, que acabou por a comercializar já com o nome *FF Beowolf* — chegando a ser aplicada em cartazes, revistas, capas de CDs, logotipos, entre outros. Mas esta técnica acabaria por passar a ser obsoleta, pois as impressoras começaram a ignorar as deformações, com as novas *drivers* e sistemas operativos. No entanto com o aparecimento da tecnologia OpenType, deu uma nova esperança, e utilizando aleatoriedade pré-programada conseguiram dar novas possibilidades de variações à fonte. Em 2011, veio a ser selecionada, juntamente com mais 22 projetos, para a coleção permanente do Museu de Arte Moderna de Nova York, e fez parte da instalação "*Standard Deviations*" (FONTSHOP, N.D.).

Juntamente com o aparecimento do computador pessoal, surgiram linguagens de programação que transformaram o paradigma da prática do design gráfico e do desenho de tipos de letra. Uma delas foi a linguagem de programação Python. Just van Rossum, irmão do criador do Python (Guido Van Rossum) explica em entrevista à *TypeThursday* de que forma o Python entrou no mundo do design de tipos e a sua importância no campo.

« I just want to get some things done in a way that allows me to focus on the problem instead of on the difficulties of programming.» afirma Just van Rossum numa entrevista à *TypeThursday* em 2016.

Just van Rossum pretendia realizar experiências no desenho de tipos através de programação, no entanto as linguagens de programação tradicionais como C e Pascal eram relativamente difíceis de trabalhar para este propósito. Desta forma, o seu irmão introduziu-o a uma nova linguagem, Python, com uma sintaxe bastante mais simples em relação às outras linguagens de programação da altura. No final dos anos 1990, em colaboração com Erik van Blockland e Petr van Blockland, desenvolveram a aplicação *RoboFog*, que era uma versão híbrida do *Fontographer 3.5* com *script* de Python adicionado. Rapidamente o Python ficou a linguagem predominante no campo do desenho de tipos, começando a ser utilizado em programas da Adobe e FontLab (TYPETHURSDAY, 2016).

Esta linguagem possibilitava dois usos diferentes no desenho de tipos, por um lado, a produção técnica das fontes, ou seja, ajudar a preparação das fontes para uso comercial. Por outro lado, viabilizava a exploração visual e execução visual de experiências que poderiam ser demasiado complexas ou demorosas para serem feitas manualmente (TYPETHURSDAY, 2016).

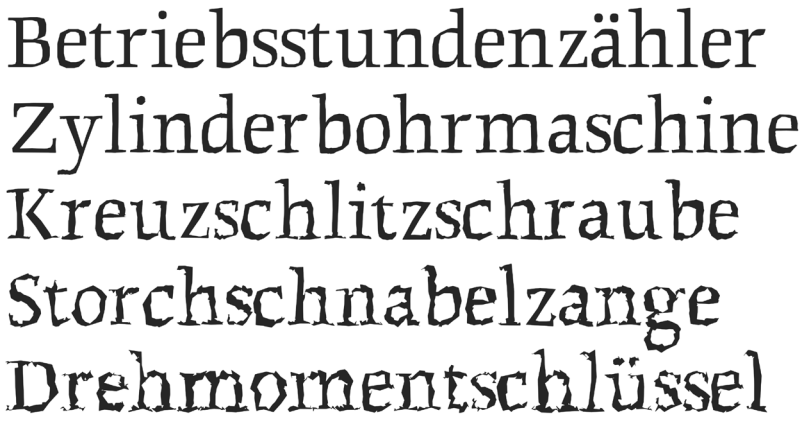


Figura 30
Pesos da FF Beowolf, consoante o grau de variações.

Em 2008, na Universidade de Ciências Aplicadas de Mainz, teve lugar o curso "*Generative Typography*" orientado por Philipp Pape e Florian Jenett, no design e na informática, respetivamente. Deste curso resultaram várias aplicações e fontes experimentais desenvolvidas pelos alunos: *Bastard, Blast, Broken Grid, Carve LCD, Elien, Fontmix, Irratio, PDE Lubanoise, Pong,* Zwirn e WDCE (PAPE & JENETT, N.D.).

A aplicação *Bastard,* foi desenvolvida por Tobias Tschense, utilizando *Processing* e a biblioteca *Geomerative* — que facilita o acesso aos pontos que formam as letras (Figura 31). Consiste na criação de tipos

generativos, a partir de fragmentos de fontes muito diferentes — mas com transições correspondentes — o que resulta em tipos sempre distintos (MARXER, N.D.; TSCHENSE, N.D.).

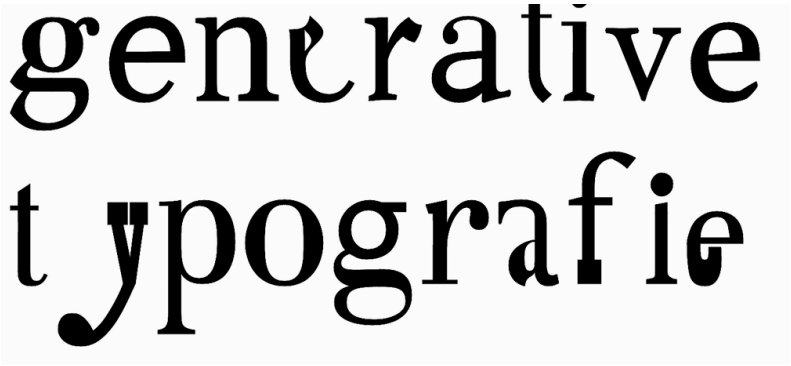


Figura 31
Bastard, criada por Tobias Tschense em 2008.

Blast é uma fonte generativa que reage ao som, desenvolvida por Denis Klein, utilizando *Processing* e uma biblioteca de análise de som (Figura 32). A música é analisada em tempo real e os dados obtidos pela biblioteca são diretamente representados na forma visual da fonte — forma e espessura. Assim, é possível visualizar na fonte, algumas das características da música — por exemplo, os graves, médios e agudos (KLEIN, N.D.).

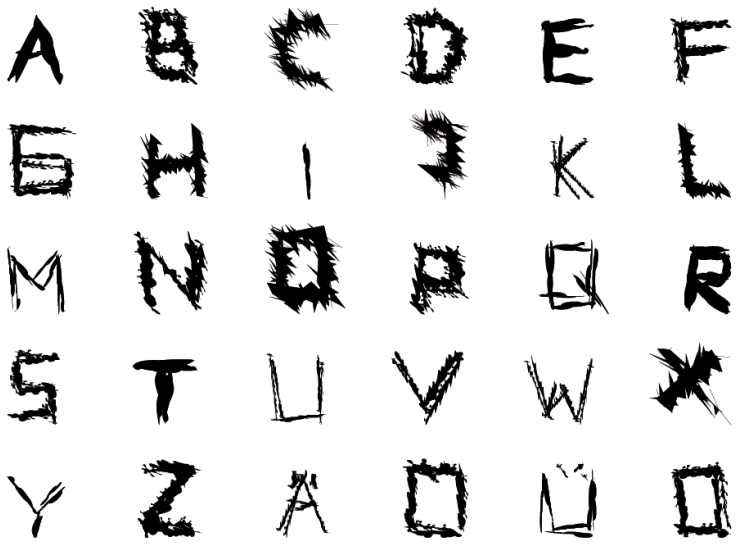


Figura 32
Blast, criada por Denis Klein em 2008.

Broken Grid é uma fonte generativa desenvolvida por Nils Holland-Cunz, também utilizando *Processing* (Figura 33). A fonte é criada a partir de uma existente, da qual, foram utilizados os pontos de âncora para manipulação das curvas de *Bézier*, alterando assim a sua forma radicalmente (HOLLAND-CUNZ, N.D.).

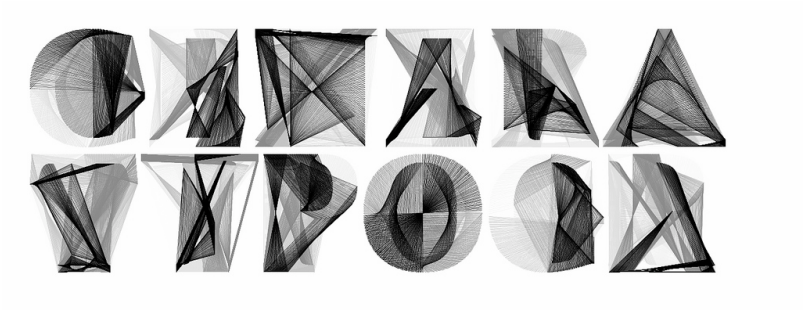


Figura 33
Broken Grid, criada por Nils Holland-Cunz em 2008.

Na Figura 34 é possível observar o sistema *CarveLCD,* desenvolvido por Alex Balzien em Processing. Consiste na geração de formas completamente novas de caracteres, modificando alguns dos parâmetros fornecidos (BALZIEN, N.D.).

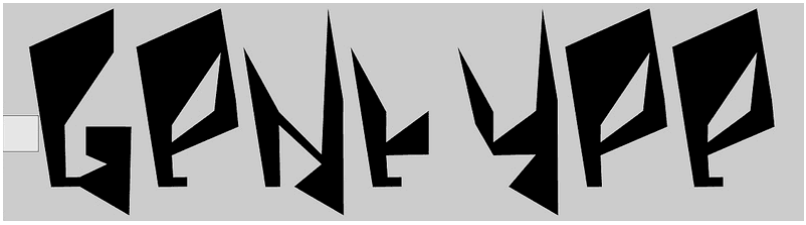


Figura 34
CarveLCD, criada por Alex Balzien em 2008.

Elien é uma fonte monospace generativa desenvolvida num sistema criado por Tatevik Aghabayan, em *Processing* com recurso a *Metaballs* (Figura 35). Estas são objetos que podem conter até cinco níveis de círculos dispostos entre si, e criam transições quando a distância entre eles é reduzida. Desta forma, as *metaballs* são distribuídas pelo esqueleto das letras, formando conexões entre os círculos do mesmo nível, e também entre os círculos de letras vizinhas. O tamanho dos círculos é calculado de forma aleatória. O sistema fornece um conjunto de parâmetros que influenciam a forma visual das *metaballs*, permitindo variações ilimitadas: o espacejamento, que influencia a distancia das letras e por sua vez o números de ligações entre letras vizinhas caso a distância seja pequena; a densidade influencia o número de *metaballs* no esqueleto; o contraste, que influencia a diferença dos tamanhos das *metaballs*; e os níveis influenciam o número de círculos em cada *metaball*. A fonte gerada contém 39 glifos, incluindo alguns sinais de pontuação (AGHABAYAN, N.D.).

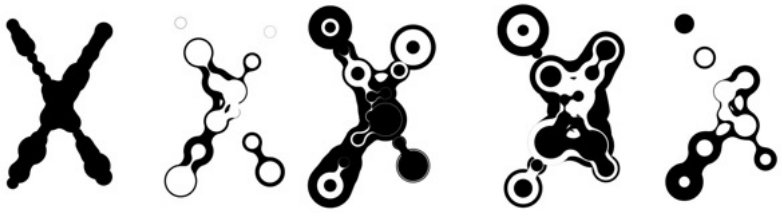


Figura 35
Elien, criada por Tatevik Aghabayan em 2008.

A aplicação *Fontmixer*, foi desenvolvida por Stefanie Oppenhäuser em *Processing*, e consiste na mistura de três fontes escolhidas pelo utilizador e posterior subtração ou adição entre elas (Figura 36). O utilizador, possui o controlo onde são feitas estas operações e desta forma os resultados são sempre diferentes uns dos outros (OPPENHÄUSER, N.D.).

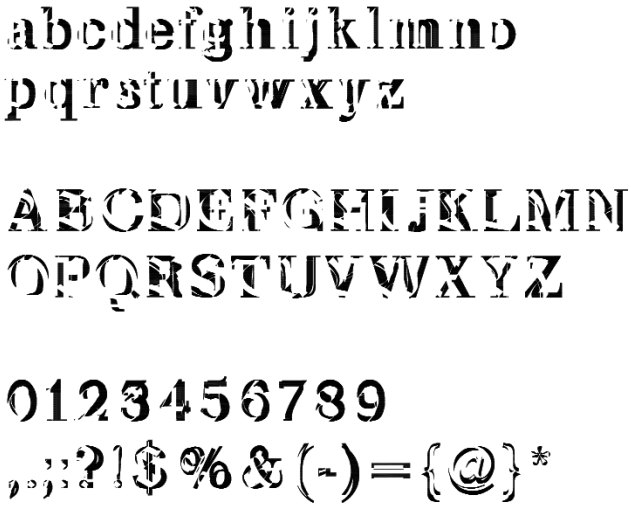


Figura 36
Fontmixer, criada por Stefanie Oppenhäuser em 2008.

A *Irratio* é um tipo de letra generativo desenvolvido por Ingo Reinheimer também em *Processing* (Figura 37). Baseada num tipo de letra com formas simples e sem serifas, foi construída através da ligação, com curvas de *Bézier*, entre pontos de âncora sucessivos, ou seja, é feita a ligação entre o primeiro ponto e o terceiro, entre o segundo e o quarto, e assim sucessivamente, até percorrer todos os pontos da letra (REINHEIMER, N.D.).

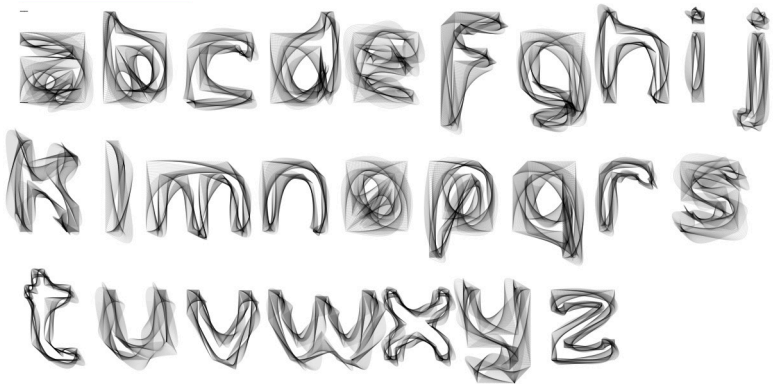


Figura 37
Irratio, criada por Ingo Reinheimer em 2008.

PDE Lubanoise é um sistema desenvolvido por Il-Ho Jung, que consiste na divisão da fonte Lubalin em diferentes números de pontos e na posterior parametrização dos mesmos (Figura 38). O utilizador pode alterar o número destes pontos, o diâmetro e ainda a direção do movimento (JUNG, N.D.).

Figura 38
PDE Lubanoise, criada por Il-Ho Jung em 2008.

Pong é uma fonte generativa, desenvolvida por Kersten Stahl (Figura 39). É criada a partir de uma bola que deambula dentro dos contornos das letras e deixa um rasto — desenha uma linha. Desta forma, o utilizador tem a possibilidade de alterar a grossura da linha, a cor ou serem representadas por curvas. O nome da fonte é uma referência ao jogo que tem o mesmo nome e do qual é feita a metáfora para este projeto (STAHL, N.D.).

Figura 39
Pong, criada por Kersten Stahl em 2008.

Zwirn é uma ferramenta desenvolvida por Lisa Reimann em *Processing*, que gera novos tipos de letra a partir de qualquer fonte no formato TTF (Figura 40). A aplicação desenha os pontos que não são visíveis no contorno das letras e desta forma obtém um entrelaçamento entre todas as letras, contando com vários parâmetros que podem influenciar o aspeto visual da fonte (REIMANN, N.D.).

Figura 40
Zwirn, criada por Lisa Reimann em 2008.

WDCE é a abreviatura para "*words do not come easy*", uma fonte criada por Christoph Köhler com recurso ao *Processing* (Figura 41). Esta é apenas composta por gradientes, que podem ser alterados a partir de um conjunto de parâmetros, como os ângulos, rotações, segmentos, refleções e sobreposições — obtendo sempre resultados diferentes (KÖHLER, N.D.).

Figura 41
WDCE, criada por Christoph Köhler em 2008.

Typegalapos é um *software* desenvolvido por Ann Chen e Danne Woo em *Processing* (Figura 42). A aplicação utiliza algoritmos evolucionários para transformar tipos de letra existentes em novas versões, mas generativos. Este projeto surge devido grande interesse de ambos por tipografia, design evolutivo e algoritmos genéticos (CHEN, N.D.; WOO, N.D.).

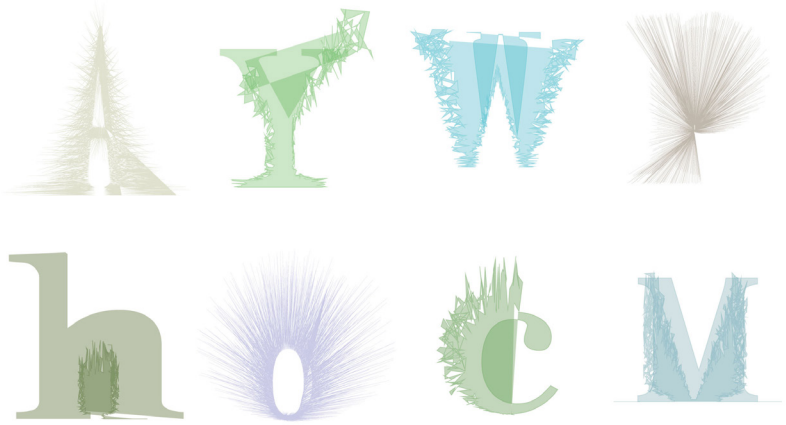


Figura 42
Typegalapos, criada por Ann Chen e Danne Woo.

Evotype é um sistema generativo para o desenho de tipos de letra baseado num algoritmo evolucionário, desenvolvido por Tiago Martins, João Correia, Ernesto Costa e Penousal Machado. Este projeto é baseado na ideia de um *stencil* capaz de gerar todas glifos do alfabeto de forma coerente, conhecido como *PDU (Plaque Découpée Universelle),* desenvolvido por Joseph A. David em 1876 (Figura 43). O utilizador pode escolher as formas para gerar os glifos, que durante o processo de evolução, sofrem várias alterações para alcançar o melhor resultado (Figura 44). O sistema demonstra capacidade de gerar uma ampla variedade de glifos alternativos, o que amplia os limites entre *legibility* e expressividade (MARTINS, CORREIA, COSTA, & MACHADO, N.D.).

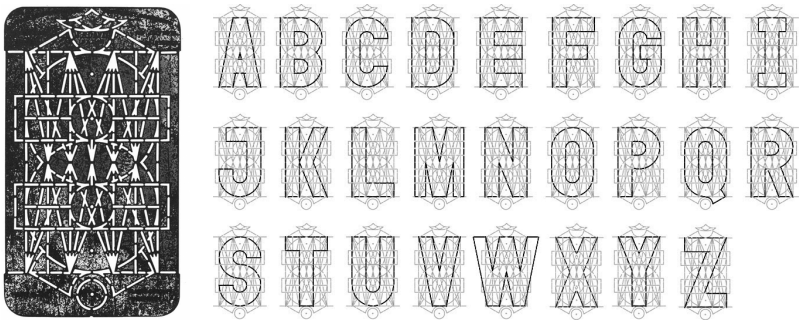


Figura 43
Plaque Découpée Universelle (PDU), Joseph A. David, 1876.

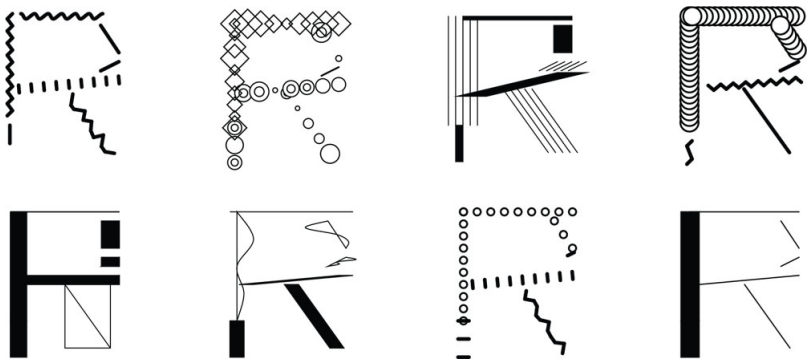


Figura 44
Evotype. Elementos de um stencil evoluído substituído por diferentes conjuntos de formas.

Em 2009, Michael Flückiger e Nicolas Kunz desenvolveram, para a tese de licenciatura, um projeto sobre tipos de letra dinâmicos denominado *Laika.* A dupla acreditava que os tipos de letra que não têm como finalidade ser impressos, deveriam ser dinâmicos para corresponder aos meios dinâmicos em que eram utilizados. Desta forma, criaram um sistema no qual um tipo de letra é capaz de alterar a sua forma e aparência a qualquer momento reagindo a um amplo conjunto de *inputs* — peso, contraste, serifas e o eixo —, em tempo real (Figura 45). O potencial desta fonte ficou demonstrado no amplo conjunto de protótipos em que podia ser utilizada, como por exemplo, receber os dados da meteorologia de uma certa cidade e ajustar a sua forma de acordo com eles, para o preço do mercado de ações ou ainda para escrita emocional (FLÜCKIGER & KUNZ, N.D., 2009)

Figura 45
Laika a reagir em tempo real a dados. À esquerda a dados meteorológicos. À direita ao peso.

Gothic Lab é um projeto iniciado em 2016, pelo designer generativo Ivan Murit e a Production Type — agência fontes digitais. Este projeto explora o potencial de padrões gerados por algoritmos (Figura 46). Murit construiu vários sistemas misturando padrões orgânicos (penas, impressões digitais ou vermes) e mecanizados tendo a aleatoriedade como parâmetro. Para este efeito, primeiro tiveram de escolher um

tipo de letra capaz de sofrer tais alterações, Antique Gothic, acabou por ser a mais apropriada. Após a escolha, adicionaram cinco novos estilos à fonte, sendo um deles o estilo *Black Rounded,* proporcionado superfície suficiente para os aplicar os padrões.

Um dos desafios neste projeto, foi a intenção de eliminar quaisquer vestígios de repetição nos padrões, evitando algoritmos previsíveis. Assim, utilizaram como *input*, para os algoritmos, a aleatoriedade presente em elementos orgânicos, como a humidade ou ruído branco. O uso de um controlador MIDI, facilitou o controlo de todos os *inputs*. Segundo os designers, um dos aspetos mais difícil de conseguir foi a influência da *letterform* no padrão, e não funcionar apenas como linha delimitadora de qualquer um dos padrões criados. Ainda assim, conseguiram esta característica em alguns dos resultados finais — por exemplo, *Gecko* e *Tigre* (LEVÉE & MURIT, N.D.).

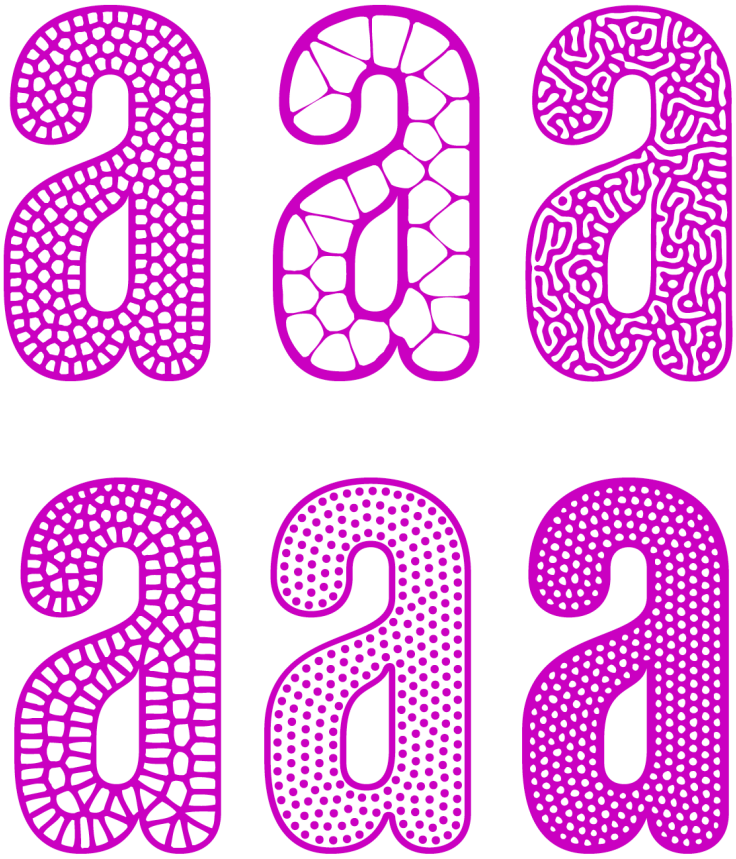


Figura 46
Diferentes versões da letra "a" pelas fontes produzidas pelo Gothic Lab.

Post Bitmap Scripter, é um projeto que consiste em fazer versões abstratas e reduzidas de *bitmap* ou fontes de pixéis e posteriormente convertê-las em linhas (Figura 47). Este projeto tem como objetivo recriar fontes originais através deste processo (TOOLS, 2007).

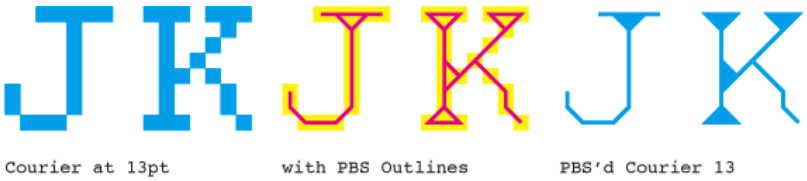


Figura 47
Post Bitmap Scripter, processo de transformação da fonte Courier em linhas.

O projeto *Letter Spirit*, desenvolvido por Gary McGraw e John Rehling, consiste na tentativa de modelar aspetos centrais da perceção humana de alto nível e da criatividade num computador (Figura 48). Deste modo, optaram por utilizar o processo de construção de um tipo de letra para simular estes aspetos no computador. Assim, o objetivo é perceber como é que as 26 letras minúsculas do alfabeto romano podem ser renderizadas em diferentes estilos, mas mantendo coerência. Para isto, abordam dois aspetos importantes da letra: a coesão categórica possuída por letras relativas a uma determinada categoria, por exemplo ao "a"; e a coesão estilística possuída por letras relativas a um determinado estilo, por exemplo, "Helvetica". Utilizando um pequeno conjunto de letras iniciais com um determinado estilo, o programa procura criar as restantes letras de maneira a que todas tenham o mesmo estilo (McGraw & Rehling, n.d.).

Figura 48
Letter Spirit, várias fontes desenhadas na grelha por vários utilizadores.

Typographic Music é um projeto que reflete um processo de investigações em tipografia generativa feitas pela designer gráfica Dina Silanteva (Figura 49). Consiste numa grelha básica e três formas geométricas simples — círculo, octógono e quadrado —, para a construção de letras. O projeto é aplicado à música e desta forma os parâmetros de construção das letras são alterados por ela. As formas correspondem ao tipo de som: círculo ao som orgânico; octógono ao som analógico; e o quadrado ao som digital. As características das formas, correspondem a características da música, por exemplo, o raio à duração do som e a saturação ao volume (SILANTEVA, N.D.).

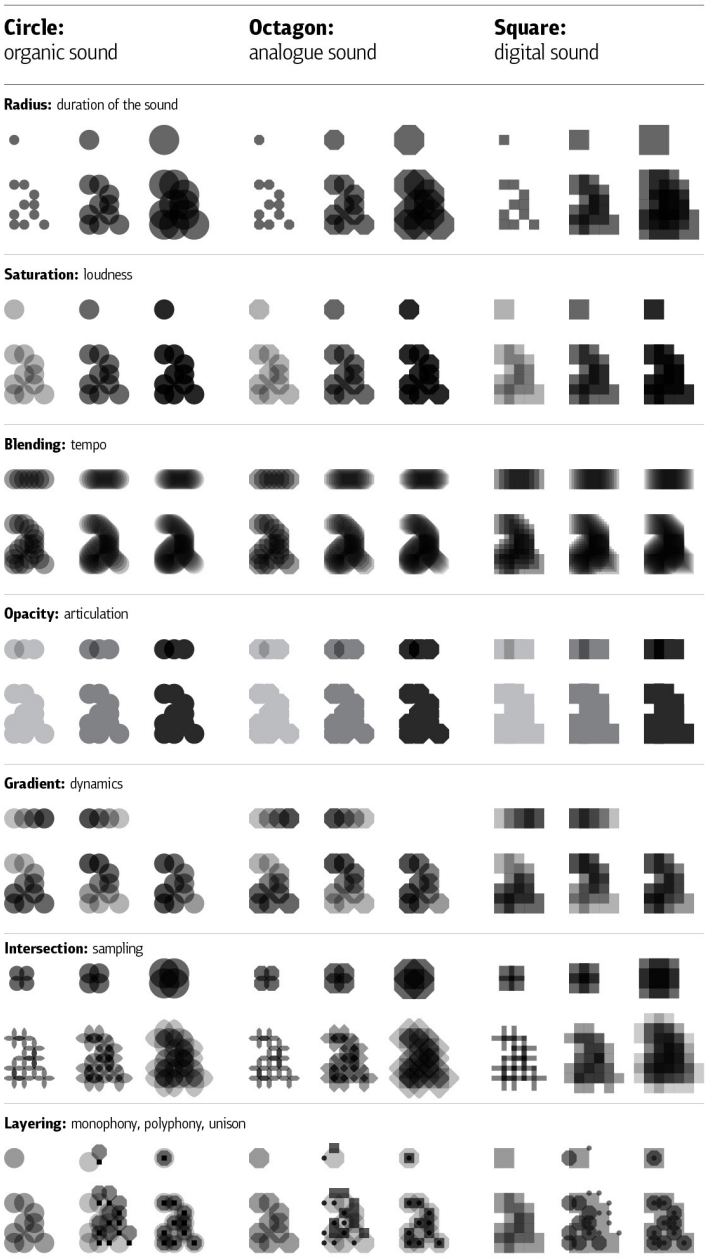


Figura 49
Typographic Music, vários parâmetros utilizados na construção de tipos de letra.

Em 2018, Jéssica Parente desenvolveu um sistema que explorou a possibilidade de gerar tipos de letra a partir da combinação de camadas de diversos tipos de letra (Figura 50) — elementos da anatomia da letra. Este sistema computacional consiste na separação dos diferentes elementos da anatomia da letra em camadas e na posterior combinação e aplicação de um método de preenchimento dos mesmos, utilizando formas como o círculo, quadrado e triângulo (PARENTE, 2018).

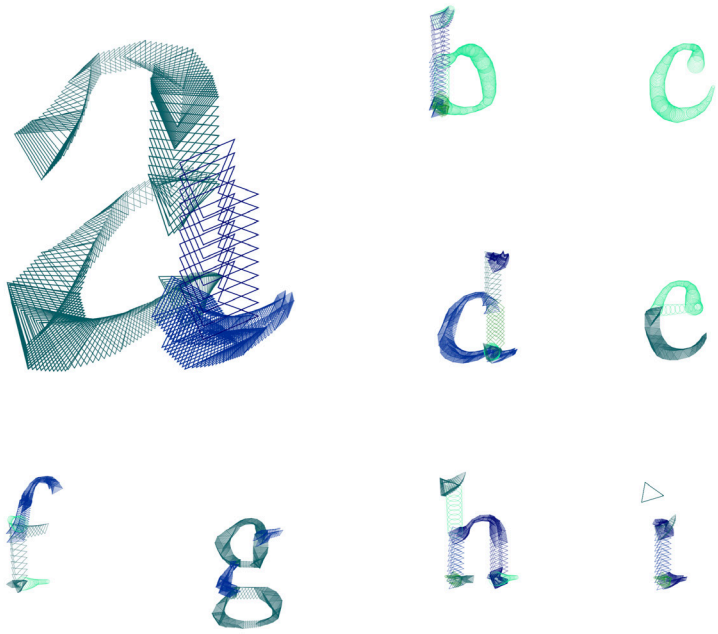


Figura 50
Tipo de letra gerado pelo sistema desenvolvido por Jéssica Parente em 2018.

Typeface é um projeto que gera tipos de letra com base em expressões faciais, desenvolvido por Mary Huang em 2010 (Figura 51). O sistema utiliza *Processing* para a interface da aplicação e *OpenCV* para a deteção dos movimentos faciais que posteriormente são traduzidos em variáveis que afetam as características das letras — como a altura-x e o eixo —, desta forma, possibilita infinitas variações (HUANG, 2010).

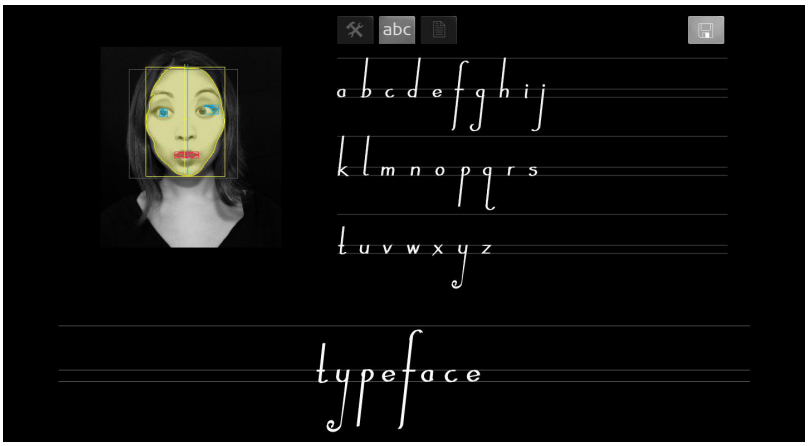


Figura 51
Interface do sistema Typeface.

Normaltype é uma aplicação de geração de tipos de letra *display*, desenvolvida pelo grupo criativo *Normals.* Inicialmente não existem formas de letras, apenas um esqueleto e pontos (Figura 52). A aplicação fornece um conjunto de parâmetros para modificar as características das letras geradas, assim como a possibilidade de criar animações e exportar como um *GIF* animado (NORMALS, N.D.).

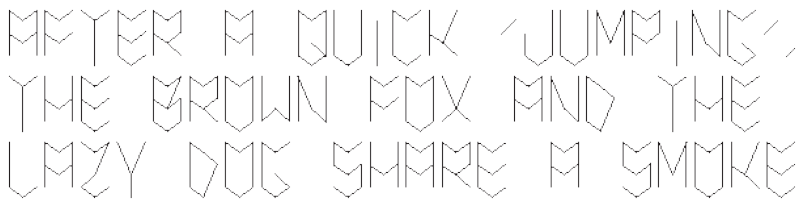


Figura 52
Tipo de letra gerado pelo sistema Normaltype.

Growing Data é um projeto de investigação que examina como processos reais e estruturas podem ser utilizadas para criar novas formas alternativas de visualização de informação (Figura 53).

As estratégias generativas em particular levam a criação de padrões visuais e estruturas que o cérebro humano facilmente interpreta numa imagem só. O projeto explora o crescimento de plantas virtuais, neste caso aplicadas a tipos de letra, explorando as possibilidades de visualizar a qualidade do ar nas grandes cidades. Estas plantas têm propriedades — vida útil, densidade e velocidade de crescimento —, que são influenciadas por fatores externos. Deste modo, são criadas visualizações únicas e sempre diferentes (ONFORMATIVE, 2011).

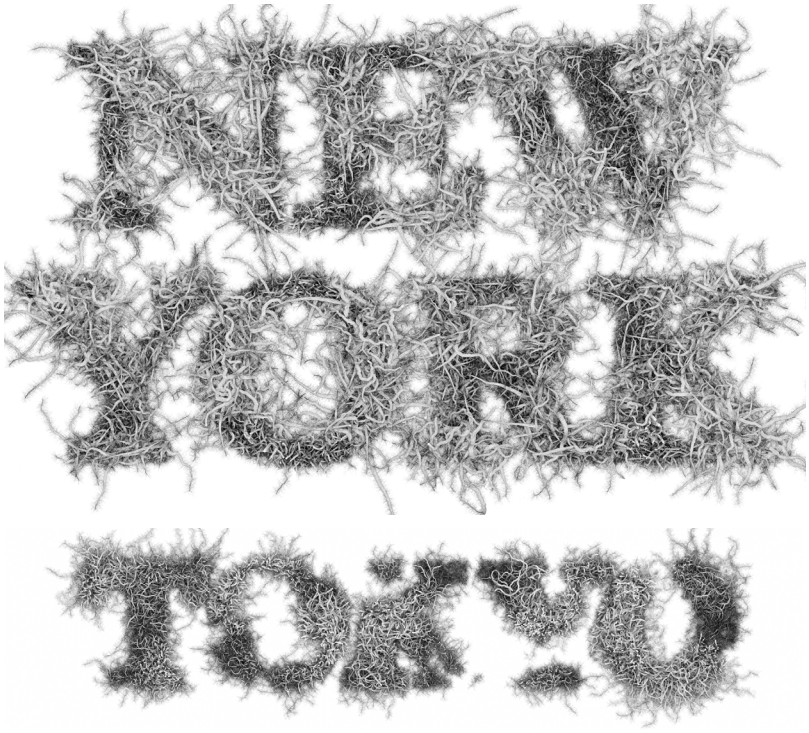
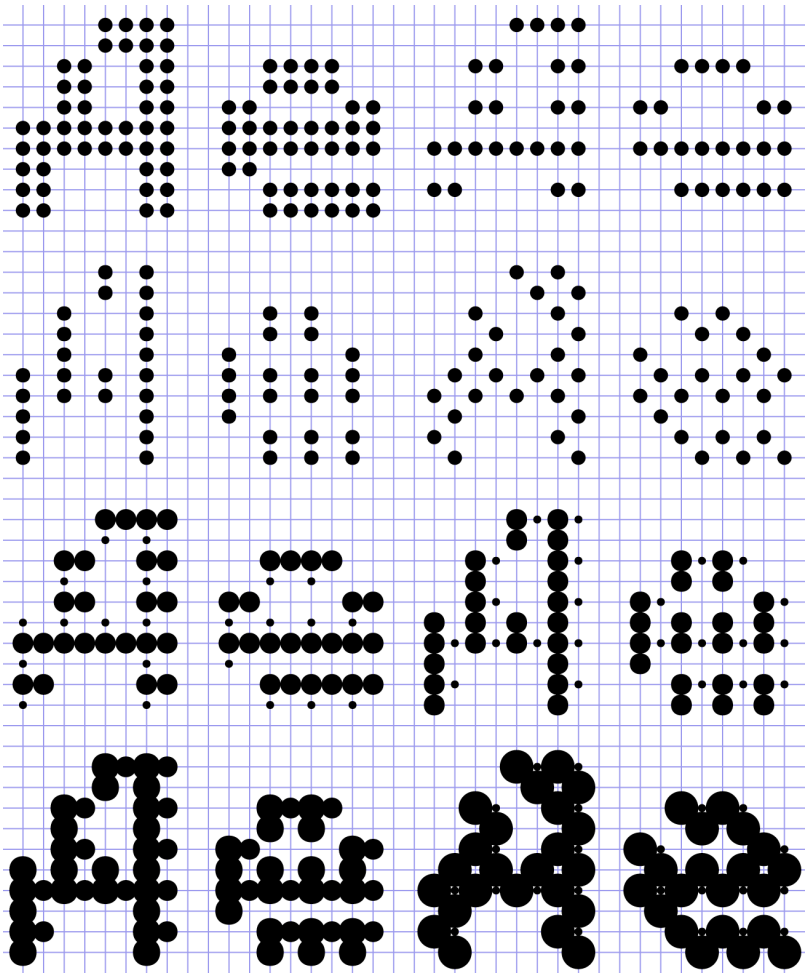


Figura 53
Tipo de letra gerado no projeto de investigação Growing Data.

Em 2010, Paul McNeil e Hamish Muir formaram uma dupla (MuirMcNeil) focada na exploração de sistemas paramétricos para solucionar problemas da comunicação visual. Na exploração destas novas abordagens, realizaram trabalhos inovadores em várias áreas, nomeadamente, na tipografia com os projetos: *TwoPoint*, *TwoPlus, ThreePoint, ThreeSix e Intersect* (MCNEIL & MUIR, N.D.-B).

TwoPoint é um sistema de tipos de letra *monospace* que explora o uso de permutações progressivas para gerar formas tipográficas. A fonte permite quatro estilos variantes e cada um tem seis pesos. Os pesos são incrementados sem alterar nenhum posicionamento (Figura 54). Em vez disso, os diâmetros dos pontos individuais dentro de cada letra mudam progressivamente (MCMEIL & MUIR, N.D.-F).

Figura 54
Versões da fonte TwoPoint, desenhada por MuirMcNeil.



TwoPlus é o desenvolvimento do sistema modular *TwoPoint.* Consiste numa coleção de sete fontes monospace, que englobam 48 versões de pesos no total (Figura 55). O projeto foi desenvolvido para ser aplicado na identidade da pós-graduação de 2015 no *London College of Communication* e posteriormente na identidade da *TypeCon* 2016. Nes-

te sistema, são exploradas várias componentes — contornos, cores, texturas, padrões e transparências — algumas delas partilhas com o sistema *TwoPoint,* por exemplo, a grelha que utilizada para construção das letras (MCNEIL & MUIR, N.D.-E).

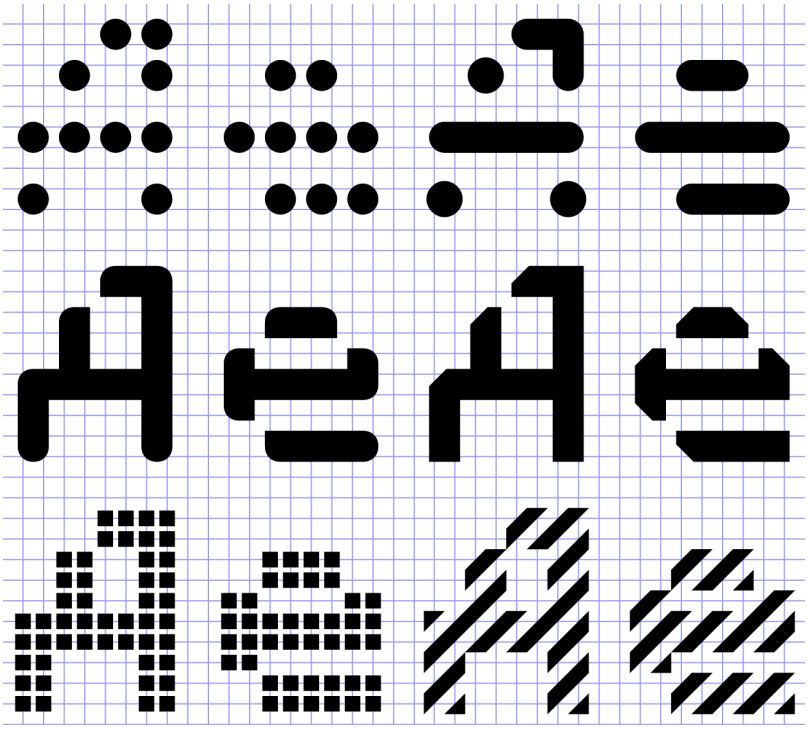


Figura 55
Versões da fonte TwoPlus, desenhada por MuirMcNeil.

ThreePoint é um sistema de tipos de letra tridimensionais. As letras são construídas por círculos dispostos numa grelha e podem ser projetados em quatro vistas ortográficas: superior e inferior direita; e superior e inferior esquerda (Figura 56). Cada uma destas projeções tem ainda quatro versões (pesos), nas quais são alteradas o tamanho dos círculos (MCNEIL & MUIR, N.D.-C).

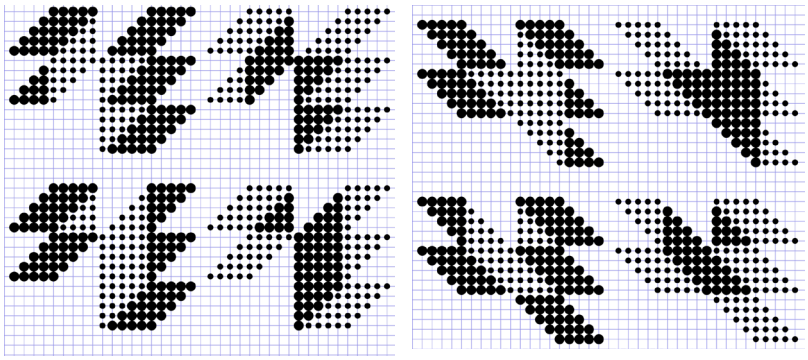
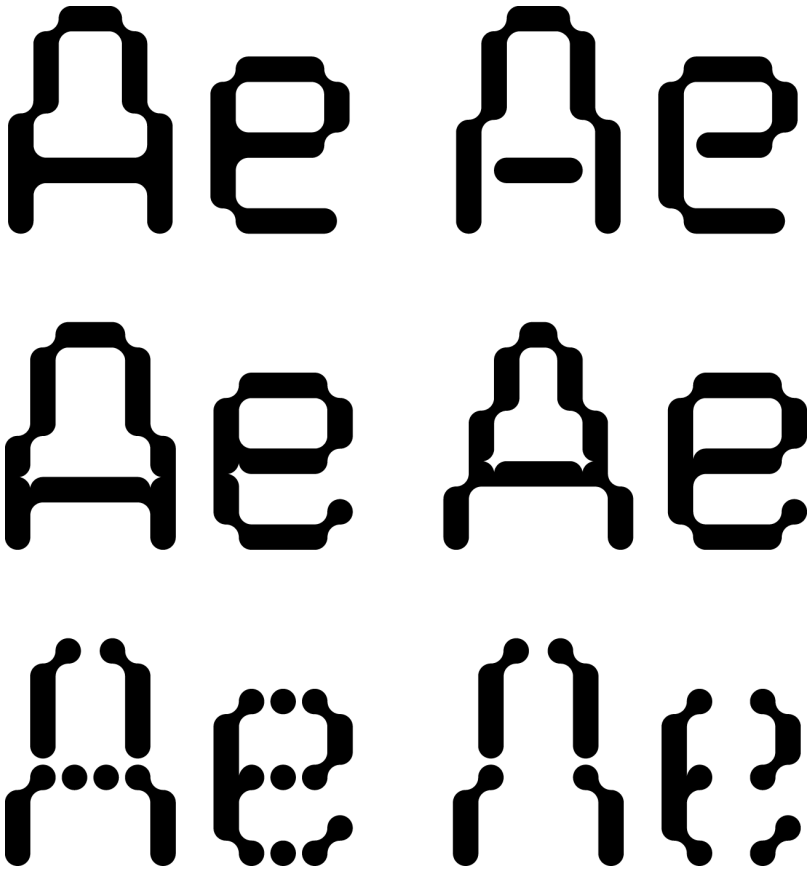


Figura 56
Versões da fonte ThreePoint, desenhada por MuirMcNeil.

ThreeSix é um sistema geométrico constituído por seis tipos de letra, cada um com oito pesos (Figura 57). Neste sistema são exploradas questões como a *legibility* e *readability*, no processo de construção das fontes geométricas, quer para o uso em texto corrido ou apenas *display*. Cada um dos seis tipos de letra pode sofrer modificações de acordo com cinco parâmetros: contorno, que define a forma das letras individuais; modulação do traço, que define o contraste entre traços horizontais e traços verticais; junções, que operam os efeitos óticos nas interseções dos traços; peso, que representa a acumulação progressiva de densidade nos esqueletos das letras; e espacejamento entre as letras. Em 2011, este projeto ganhou o *Premier Award*, da *International Society of Typographic Designers* (MCNEIL & MUIR, N.D.-D).

Figura 57
Versões da fonte ThreeSix, desenhada por MuirMcNeil.



Intersect é um sistema geração de tipos de letra que explora a relação da forma das letras e o espaço vazio à volta das mesmas, ao contrário dos tipos tradicionais que apenas apresentam um contraste binário — preto e branco, forma e contra forma. Desta forma, o sistema emula faixas sucessivas de formas à volta dos esqueletos das letras, possibilitando inúmeras variações (Figura 58). Utilizando um *software* de paginação, o designer pode explorar a cor, tonalidade e transparências do sistema (MCNEIL & MUIR, N.D.-A).

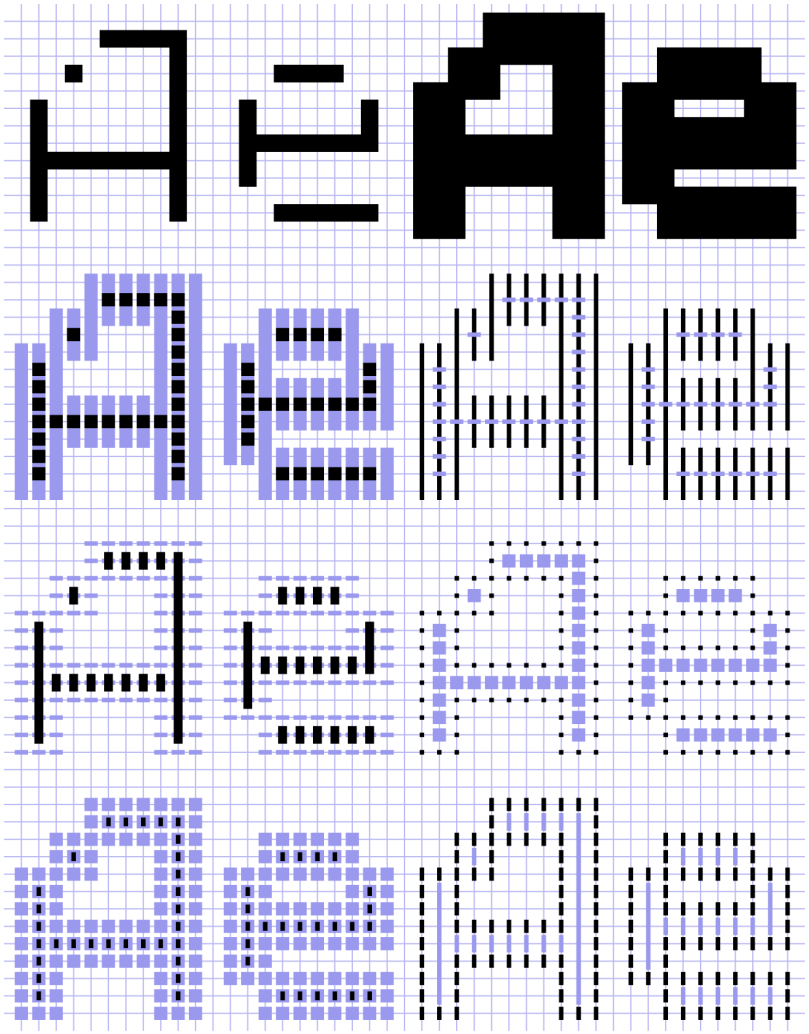


Figura 58
Versões da fonte Intersect, desenhada por MuirMcNeil.

Em 2013, Catarina Maçãs desenvolveu um tipo de letra generativo capaz de representar valores emocionais (aversão, felicidade, medo, raiva, surpresa e tristeza), através do conteúdo de um texto (Figura 59). Este tipo de letra absorve as características do texto acabando por representar o conteúdo textual e assim fornece ao utilizador uma representação do tipo de emoção presente no texto. Neste trabalho foi dada importância à representação das emoções e a distinção entre elas, sendo objetivo a interpretação individual de cada utilizador de acordo com a sua cultura visual e vivências.

Foi também desenvolvida uma aplicação que permite a interação do utilizador com a geração do tipo de letra, onde é possível submeter um texto, ao qual é feita a analise emocional e posteriormente é gerado um PDF, com o texto composto no tipo de letra gerado, e um ficheiro TTF — o tipo de letra gerado (MAÇÃS, 2013).

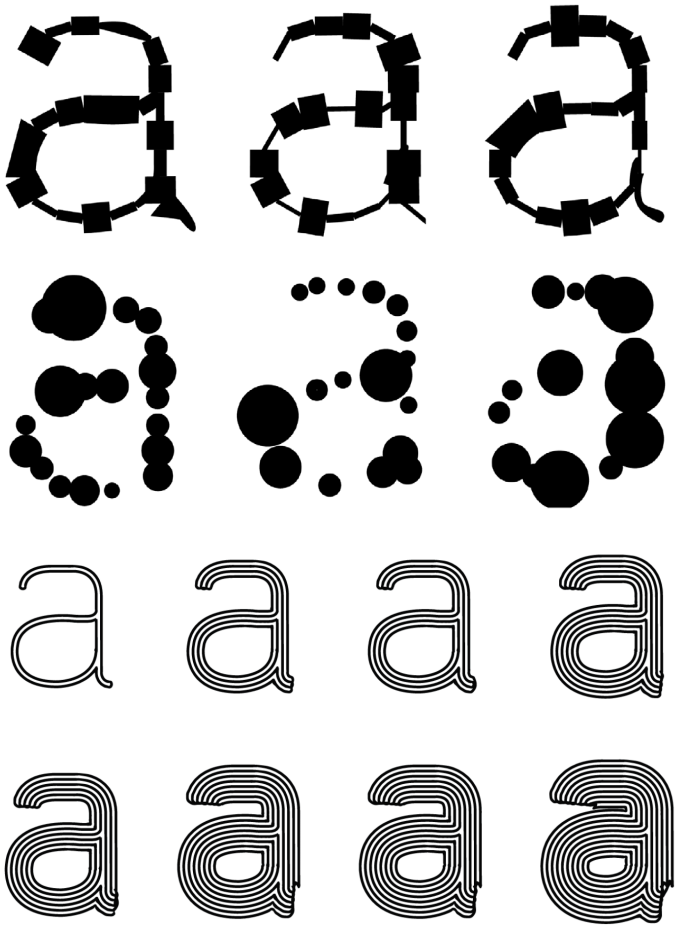


Figura 59
Tipos de letra gerados pelo sistema desenvolvido por Catarina Maçãs em 2013.

5 — Projeto Prático

064 — 125

Neste capítulo é apresentado todo trabalho desenvolvido para a componente prática desta dissertação. Em primeiro lugar, são apresentados o conceito e as funcionalidades para o sistema desenvolvido, assim como o trabalho preliminar. De seguida, descrevemos de que forma construímos este sistema, assim como o seu funcionamento. São também expostos e analisados os testes com utilizadores realizados ao sistema. Por fim, fazemos uma demonstração das possíveis aplicações do sistema *web*.

5.1. Conceito

Na componente prática desta dissertação é desenvolvido um sistema para a *web* que permite a geração algorítmica de glifos a partir de esqueletos tipográficos — formas essenciais das letras, segundo Edward Johnston — e, por consequência, a criação de tipos de letra baseados nos glifos gerados. Estes últimos, são criados a partir do preenchimento do esqueleto tipográfico através de diversas técnicas de desenho tendo em conta não só, mas também, abordagens de trabalhos relacionados. O sistema recebe dois *inputs:* (i) o esqueleto tipográfico de um tipo de letra estruturado num ficheiro JSON e (ii) uma técnica de desenho. Posteriormente, o sistema gera como *output* uma fonte em formato OTF. A Figura 60 explica o funcionamento do sistema desenvolvido.

O utilizador pode configurar um conjunto de parâmetros que é inerente à técnica de desenho escolhida e desta forma conseguir uma grande variedade de glifos, de acordo com as suas preferências visuais. Cada técnica de desenho proporciona um estilo visual único aos glifos gerados. O sistema permite ao utilizador visualizar em tempo real os glifos resultantes e, assim, dar *feedback* visual relativamente ao impacto de cada parâmetro no desenho dos glifos. O texto da pré-visualização dos glifos pode ser modificado a qualquer momento, permitindo ao utilizador experimentar os glifos com diferentes textos.

A configuração destes parâmetros pode também ser conseguida de forma automática e aleatória. Para tal, o sistema inclui uma opção para atribuir um valor aleatório a cada parâmetro e, assim, criar glifos com propriedades diversificadas e inesperadas. A exploração aleatória destes valores pode levar a resultados muito díspares da forma inicial. Para que o utilizador tenha a possibilidade de voltar à forma inicial, a qualquer momento, existe uma opção que atribui aos parâmetros os seus valores predefinidos. Existe ainda a possibilidade do sistema receber o som do microfone como *input* para os parâmetros, o que torna possível gerar formas reativas ao som em tempo real e criar uma vasta diversidade para o aspeto visual dos glifos.

Para além do utilizador poder exportar a fonte gerada pelo sistema no formato OTF, também pode exportar e importar configurações dos parâmetros via URL, possibilitando a partilha e réplica de glifos gerados pelo sistema. Pode ainda exportar o texto que está a ser visualizado em formato SVG e adicionar a fonte gerada à galeria do sistema *web*.

Para desenvolver este projeto utilizamos maioritariamente JavaScript, devido às qualidades que oferece para o desenvolvimento na *web*. Outras tecnologias importantes são: HTML, CSS, PHP, e as bibliotecas Skelefont (Java), Opentype.js (JavaScript), Javascript Clipper (JavaScript) e FileSaver.js (JavaScript). A importância de cada uma destas tecnologias será abordada mais adiante, com maior detalhe, assim como o funcionamento das funcionalidades e da configuração dos parâmetros. O sistema *web* desenvolvido encontra-se alojado num servidor, com o seguinte URL: cdv.dei.uc.pt/2019/letterspecies.

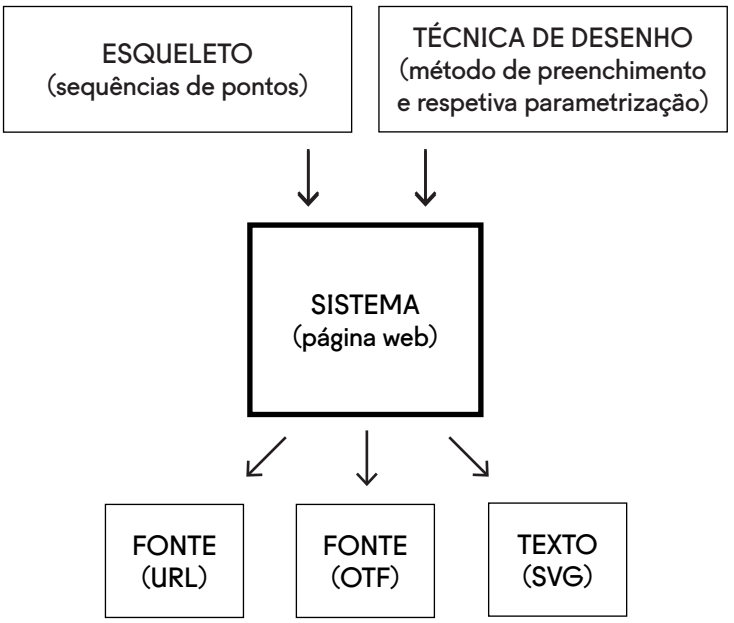


Figura 60
Arquitetura do sistema *web*.

5.2. Desenvolvimento

Esta fase do trabalho consistiu em vários passos que podem ser agrupados em quatro temas principais: (i) Esqueleto Tipográfico, onde é explicado o processo de extração dos esqueletos tipográficos; (ii) Protótipo Funcional, que teve como propósito conseguir importar os esqueletos tipográficos para o sistema *web*, conseguir desenhar glifos com base nos pontos do esqueleto tipográfico recebido e conseguir exportar os glifos gerados no formato OTF; (iii) Técnicas de Desenho, que consistiu em explorar e desenvolver as técnicas de desenho através de SVG; e por último (iv) Interface Gráfica do Utilizador, tendo como propósito a implementação da interface do sistema *web* e o desenvolvimento de todas as funcionalidades.

Esqueleto Tipográfico

O esqueleto tipográfico de um glifo pode ser visto como sua estrutura principal. Contém as características tipográficas essenciais como a relação entre os elementos anatômicos (por exemplo, ascendentes, descendentes, diagonais, linhas verticais e horizontais) e métricas (por

exemplo, altura-x, linha de base, altura total e largura). Em termos técnicos, estes esqueletos tipográficos consistem em sequências de pontos bidimensionais definindo as linhas que estão localizadas no centro da forma do glifo. Para obter os esqueletos tipográficos, começámos por perceber o funcionamento, possibilidades e limitações da biblioteca Skelefont. Esta é uma biblioteca para *Processing* (Java) desenvolvida por Tiago Martins, em 2018, que possibilita a transformação de formas de letras em esqueletos tipográficos. Para além das coordenadas X e Y, a cada ponto do esqueleto tipográfico é também associado um valor correspondente ao raio do círculo que permite replicar, com algum erro, o glifo original (Figura 61). Após o cálculo, estes valores são guardados de forma estruturada num ficheiro JSON. Todos os esqueletos tipográficos utilizados são exportados de *type foundries open source*, como por exemplo, a Google Fonts e Velvetyne Type Foundry.

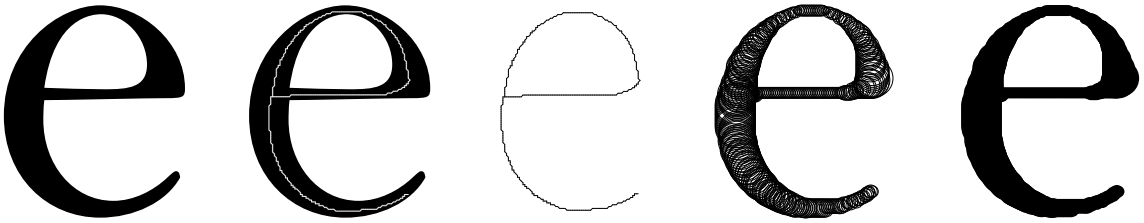
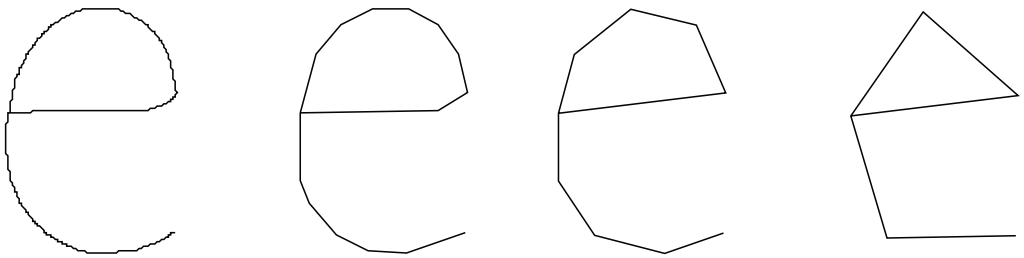


Figura 61
Processo de transformação da letra "e" em esqueleto com a biblioteca Skelefont, e posterior preenchimento através de círculos com raio recebido do esqueleto.

Os esqueletos tipográficos extraídos são formados por um grande número de pontos e a curva resultante da ligação entre eles contém pontos muito próximos. Alguns desses pontos são redundantes para o reconhecimento da estrutura do glifo e, portanto, desnecessários, por exemplo, pontos colineares. Outros pontos acrescentam imperfeições às linhas dos esqueletos. De forma a simplificar e reduzir o ruído nos esqueletos tipográficos, aplicamos o algoritmo *Ramer-Douglas-Peucker* (Wu & Marquez, 2003). Este algoritmo decompõe uma dada curva formada por segmentos de reta numa curva similar mas com menos segmentos. Assim sendo, a curva simplificada é definida pelos pontos mais importantes que definiam a curva original. O grau de diferença entre as curvas é controlado por um parâmetro, geralmente chamado *epsilon*, que define a distância máxima entre os pontos originais e os pontos simplificados. Na Figura 62 é possível observar que variando o valor de *epsilon*, é possível obter esqueletos tipográficos com diferentes níveis de detalhe.

Figura 62
Processo de redução de pontos ao esquelto tipográfico.



Protótipo Funcional

De seguida, exportamos um conjunto de esqueletos para implementar a funcionalidade de importar os esqueletos para o sistema *web*. Para isto, primeiro tentamos utilizar a biblioteca p5.js — que facilita a implementação de código *Processing* na *web*. No entanto, esta abordagem não facilitava a manipulação dos ficheiros relativos aos esqueletos tipográficos. Desta forma, optámos por utilizar uma abordagem de carregamento dos ficheiros JSON através de pedidos AJAX, pois esta oferece maior flexibilidade no processamento dos ficheiros minimizando o risco de surgirem problemas. Para isto, é utilizado o objeto *XMLHttpRequest*, que possibilita a solicitação do ficheiro JSON ao servidor. Na resposta ao pedido, é carregado, de forma assíncrona, o conteúdo do ficheiro JSON para que possa ser utilizado na técnica de desenho e por sua vez na geração dos glifos. Este pedido é realizado apenas quando o utilizador escolhe um esqueleto tipográfico de forma a otimizar a interação e fluidez do sistema (Figura 63).

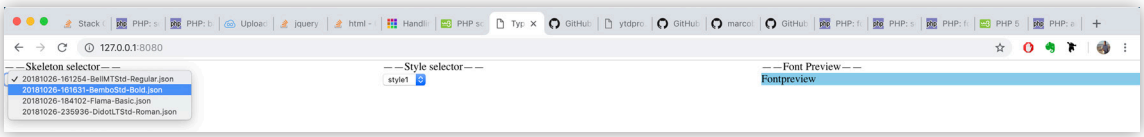


Figura 63
Importação dos esqueletos tipográficos para o sistema web através de pedidos AJAX.

Após concluir a parte da importação dos esqueletos com sucesso, os passos seguintes foram (i) desenhar glifos com base nos pontos do esqueleto recebido e (ii) exportar os mesmos para ficheiro OTF. Na Figura 64 é possível observar a primeira tentativa de desenho dos glifos (nos quadrados mais à esquerda) e a pré-visualização da fonte criada com a biblioteca Opentype.js (mais à direita). Este processo de criação da fonte, consiste na criação de um *path* para cada caractere com os métodos fornecidos pelo Opentype.js — *Move To (M), Line To (L), Curve To (C), Quad To (Q)* e *Close (Z)*. De seguida, estes *paths* são agrupados numa só fonte no formato OTF. Nesta fase, o desenho dos *paths* era baseada em círculos criados com curvas *Bézier*.

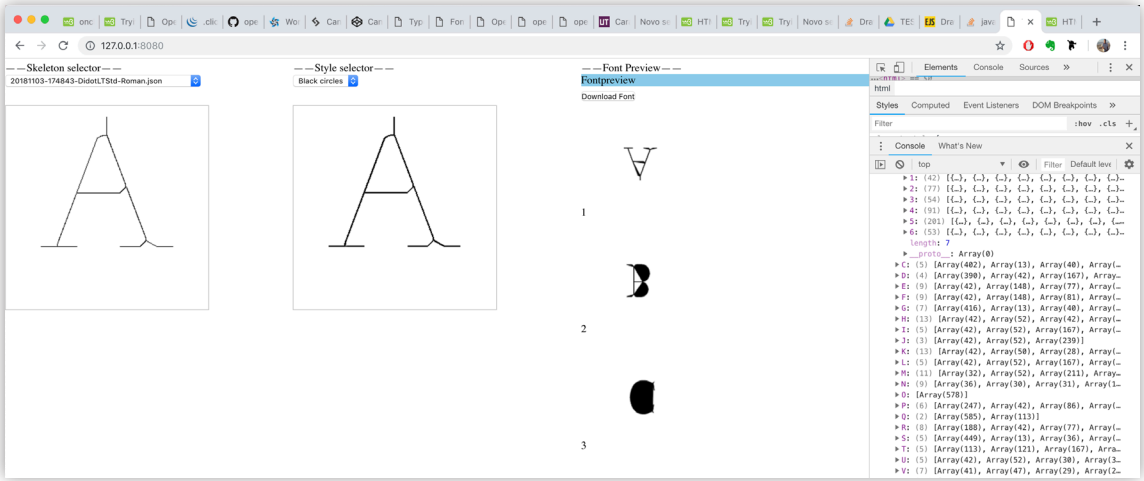


Figura 64
Primeira tentativa de desenho dos glifos pelo sistema.

O desenho dos glifos foi conseguido sem problemas, no entanto, as letras apresentavam algumas deformações e estavam invertidas verticalmente. Este problema estava relacionado com o sistema de coordenadas do OpenType ser diferente do sistema de coordenadas utilizado para gerar os esqueletos.

Deste modo, procedemos a uma tentativa de inverter as coordenadas verticais (Y) dos esqueletos, que acabou por resolver o problema das letras invertidas. No entanto as deformações continuavam presentes, como é possível observar na Figura 65.

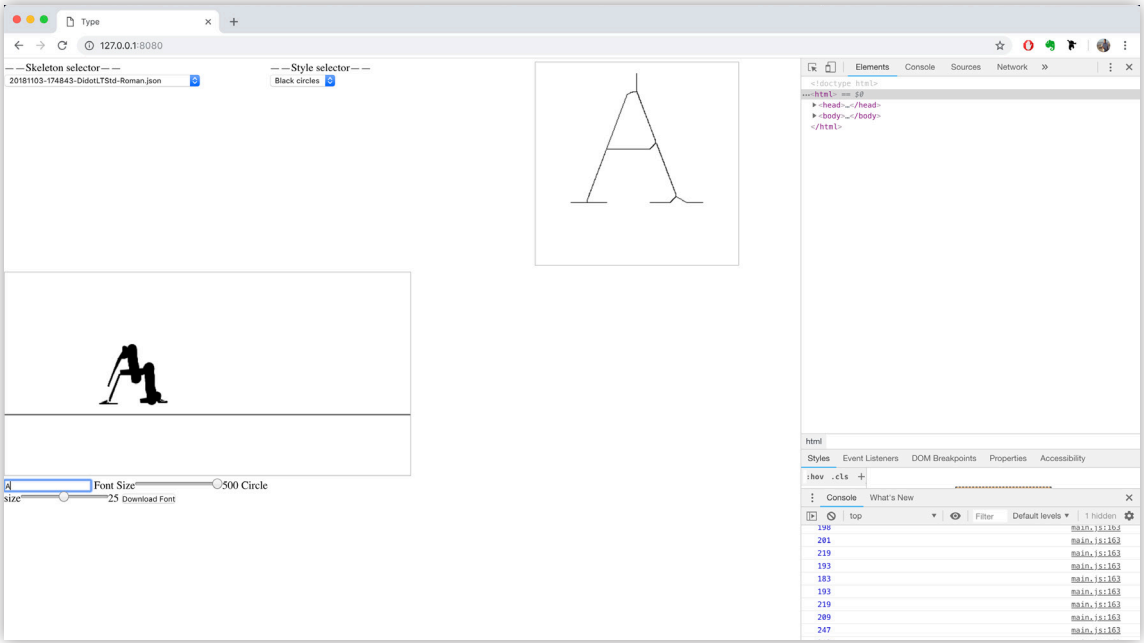


Figura 65
Sistema após a resolução do problema das letras invertidas.

As deformações eram visíveis quando eram utilizadas as dimensões associadas aos pontos do esqueleto (Figura 66). Quando as dimensões eram fixas, utilizando um raio de 10 pixéis, apenas algumas deformações eram visíveis, nomeadamente ligações que não deviam existir entre alguns dos pontos (Figura 67).

Após várias tentativas para tentar corrigir estas deformações, decidimos alterar a forma de desenhar os círculos que preenchiam o esqueleto. Optamos por criar os círculos fazendo a ligação, com linhas retas, de pontos circundantes ao ponto recebido do esqueleto, criados com recurso ao círculo trigonométrico. Desta forma conseguimos corrigir as deformações, alcançando resultados satisfatórios em comparação com a fonte original (Figura 68).

Após implementar as primeiras quatro funcionalidades conseguimos um protótipo funcional do sistema (Figura 69) com o qual era possível escolher o esqueleto tipográfico, escrever com os glifos gerados e exportar a fonte gerada. Foram feitos também alguns testes da fonte no *Adobe InDesign* (Figura 70) com o intuito de verificar o comporta-

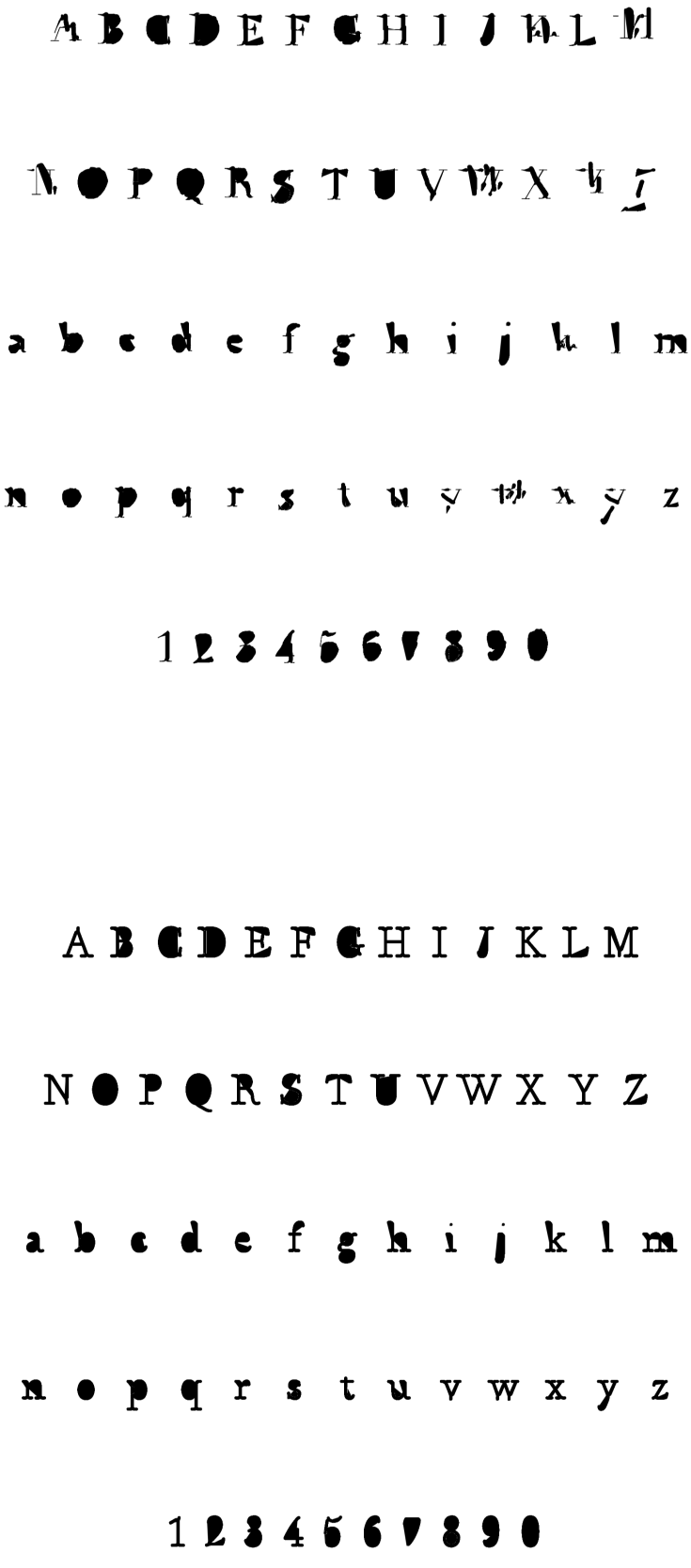


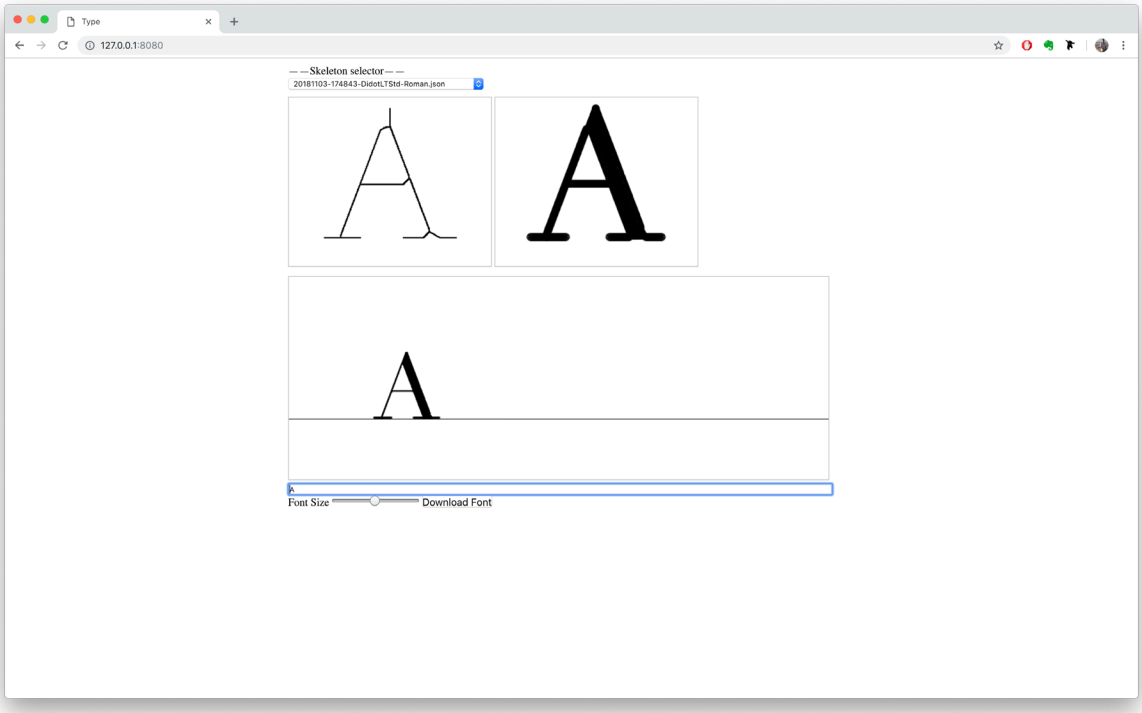
Figura 66 e 67
Diferenças entre glifos gerados com raio associado ao esqueleto, em cima, e glifos gerados com raio de 10 pixéis, em baixo.

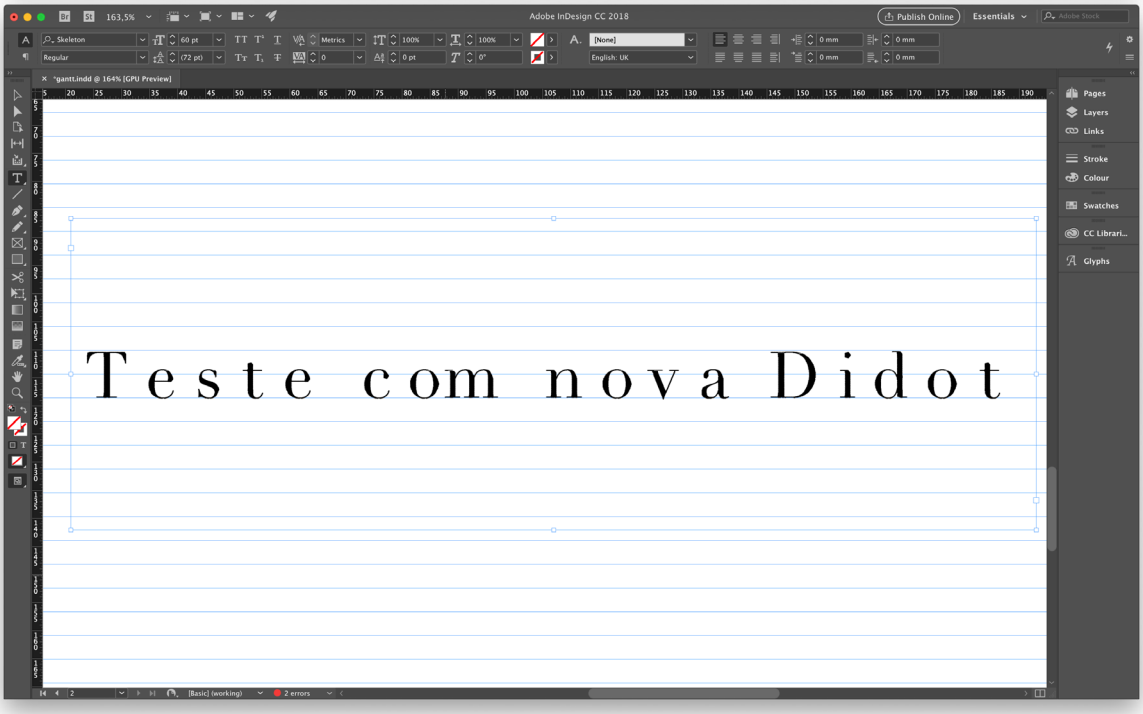
Figura 68
Em cima, fonte original (Didot LT Std). Em baixo, fonte gerada pelo sistema *web*.

mento da mesma num ambiente onde pode ser utilizada. Constatámos que existiam alguns problemas a nível de *kerning* e também no desempenho da fonte, ou seja, necessitava para além do tempo expectável para carregar e ser possível de usar. Ambos os problemas foram solucionados na seguinte fase do trabalho.

Figura 69
Em cima, protótipo funcional do sistema *web*.

Figura 70
Em baixo, teste de fonte gerada no sistema web no *Adobe InDesign*.





Técnicas de Desenho

O sistema desenvolvido cria glifos através do preenchimento de esqueletos tipográficos com técnicas de desenho. Uma técnica de desenho pode ser considerada como o traço que dá forma às linhas que constroem o esqueleto. Cada técnica de desenho fornece aos esqueletos tipográficos, e por sua vez, aos glifos, um estilo visual único. Além disso, cada técnica de desenho possui um conjunto de parâmetros que permitem ao utilizador controlar diferentes aspetos do algoritmo de desenho que implementa o preenchimento dos esqueletos. Desta forma, o utilizador pode alterar os parâmetros para obter resultados diferentes com o mesmo estilo visual. Em termos técnicos, cada técnica de desenho consiste num método de programação que desenha um glifo, dados os seguintes *inputs*: (i) os dados do esqueleto tipográfico selecionado (coordenadas x e y e raios dos pontos do esqueleto); e (ii) os valores dos parâmetros que controlam o algoritmo de desenho inerente. Implementamos nove técnicas de desenho, as quais são descritas nas subseções seguintes. Cada uma é baseada num conceito visual diferente e emprega uma combinação de processos de design e / ou formas geométricas. Posteriormente, mais técnicas de desenho podem ser facilmente adicionadas ao sistema.

1ª Técnica de desenho — *Calligraphy*

Após conseguir o protótipo funcional do sistema, o passo seguinte foi começar a desenvolver a primeira técnica de desenho. No entanto, a par com o problema da performance da fonte gerada, o sistema também apresentava alguma lentidão na tarefa de gerar a fonte sempre que era alterado o esqueleto tipográfico. Desta forma não podíamos avançar, pois tudo o que fosse implementado a seguir iria tornar o sistema ainda mais lento e por sua vez proporcionar uma má interação para o utilizador. Era portanto, necessário otimizar o código até então desenvolvido. Para tal, começamos por analisar as causas de ambos os problemas.

No que diz respeito ao sistema, de forma a renderizar o texto editável com os glifos gerados, estávamos a gerar uma fonte completa, ou seja, criar todos os glifos de novo, sempre que era alterado o esqueleto tipográfico. Isto aumentava drasticamente o número de processos a realizar. Em relação à fonte gerada, a má performance era provocada pelo enorme número de formas — neste caso, círculos — necessárias para recriar a fonte original. Desta forma, era essencial minimizar (i) o número de vezes que a fonte completa teria de ser gerada e (ii) o número de formas necessárias para o desenho dos glifos.

Posto isto, a solução encontrada foi colocar o sistema a realizar a geração da fonte completa apenas quando o utilizador decidir exportar a fonte, e para a renderização do texto, criar apenas os glifos presentes no texto editado pelo utilizador. De forma a implementar esta solução recorremos a SVG para renderizar os glifos. Esta escolha deveu-se em

grande parte às possibilidades que esta tecnologia oferece. Permite a renderização dos glifos, em tempo real, com um tempo de processamento associado relativamente baixo. Tendo em conta que a criação de formas em SVG é feita através dos métodos *moveto (M), lineto (L), curveto (C), quadratic Bézier curveto (Q)* e *closepath (Z)* — idênticos à biblioteca Opentype.js —, esta tecnologia facilita integração com a biblioteca e por sua vez facilita a geração da fonte.

Tendo sido solucionado o problema da performance, demos início a exploração da primeira técnica de desenho com uso de SVG. Esta técnica de desenho foi criada com a intenção de simular a escrita caligráfica. Para tal, analisamos uma caneta de ponta larga com o intuito de perceber as principais características necessárias para desenhar o traço caligráfico. Esta caneta não é flexível e por isso deixa um traço com largura fixa. Além disso, a caneta é normalmente utilizada num ângulo de 30 graus.

Assim sendo, utilizamos estas duas principais características como os argumentos para criar o método JavaScript juntamente com os pontos do esqueleto tipográfico. Deste modo, a nossa abordagem inicial consistiu em duplicar os pontos do esqueleto tipográfico recebido com uma certa direção, ou seja, ângulo e com um determinado comprimento. Tendo os dois conjuntos de pontos, era preciso determinar quais pertenciam ao contorno interior e exterior para os poder unir de maneira a criar apenas um conjunto de pontos, ou seja, uma única forma. Para tal, criamos um método que percorria todos os pontos e de forma sequencial era calculado o ângulo entre o ponto anterior e posterior. Caso este ângulo fosse inferior a 180 graus pertencia ao contorno interior, se fosse maior ou igual a 180 graus pertencia ao

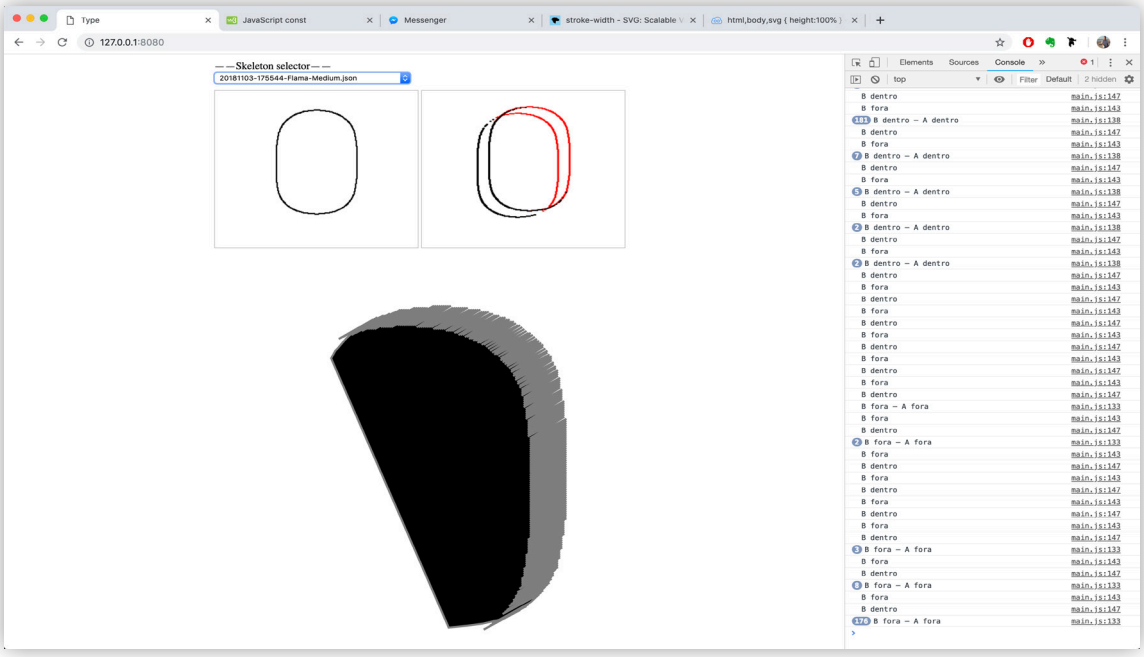


Figura 71
Primeira tentativa de implementação da abordagem inicial no glifo "o".

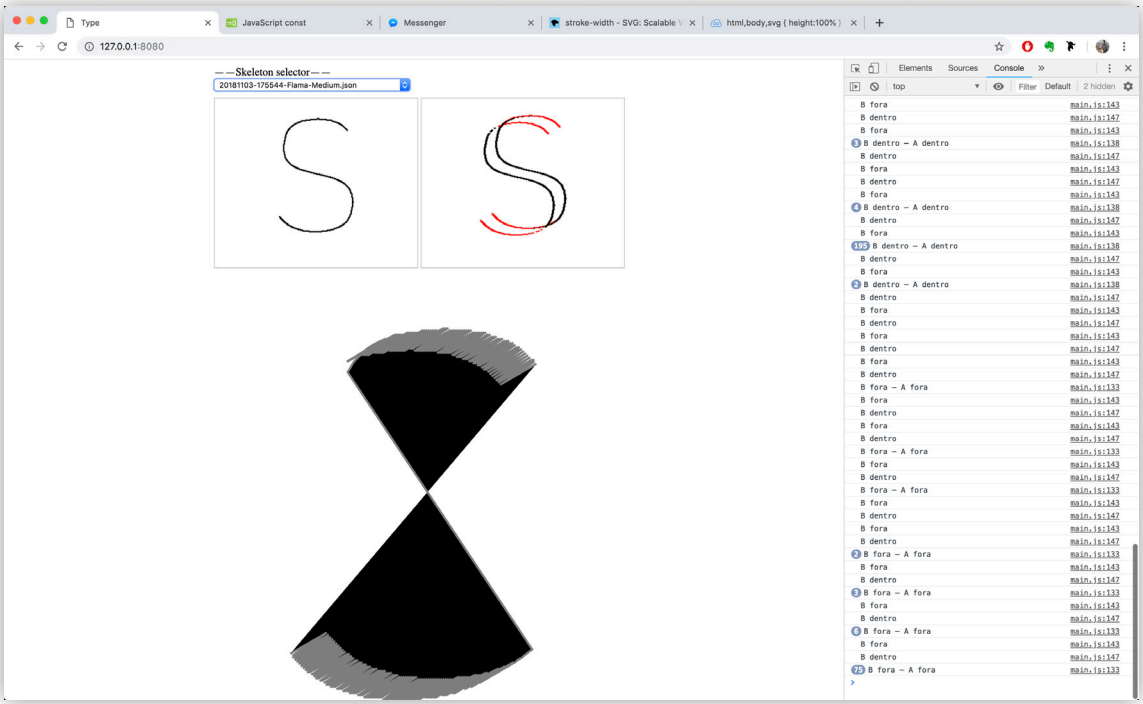


Figura 72
Primeira tentativa de implementação da abordagem inicial no glifo "s".

contorno exterior. Nas Figuras 71 e 72 é visível a primeira tentativa de implementar esta abordagem em dois glifos diferentes.

Após a implementação verificamos que existiam erros no desenho dos glifos, pois não era possível determinar com fidelidade a qual contorno os pontos pertenciam. E por sua vez, causava a incorreta ligação dos pontos. Depois de várias tentativas sem sucesso para corrigir este erro, decidimos experimentar outra abordagem.

Optamos por criar polígonos de forma sequencial por cada par de pontos e no final unir o conjunto de todos os polígonos de maneira a criar uma única forma (Figura 73). Para realizar a união dos polígonos é utilizada a biblioteca Javascript Clipper, que nos permite realizar um grande leque de operações com polígonos no formato SVG. Na Figura 74, é visível a primeira iteração desta abordagem e é possível observar que o glifo gerado apresenta uma única forma, o que revela o sucesso desta abordagem.

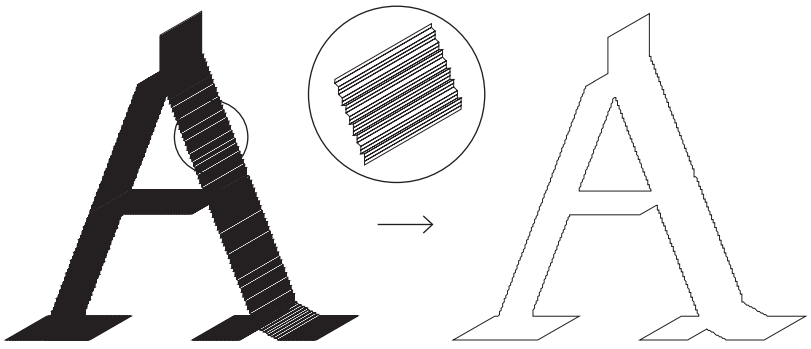


Figura 73
Esquema representativo do método utilizado para a união de polígonos

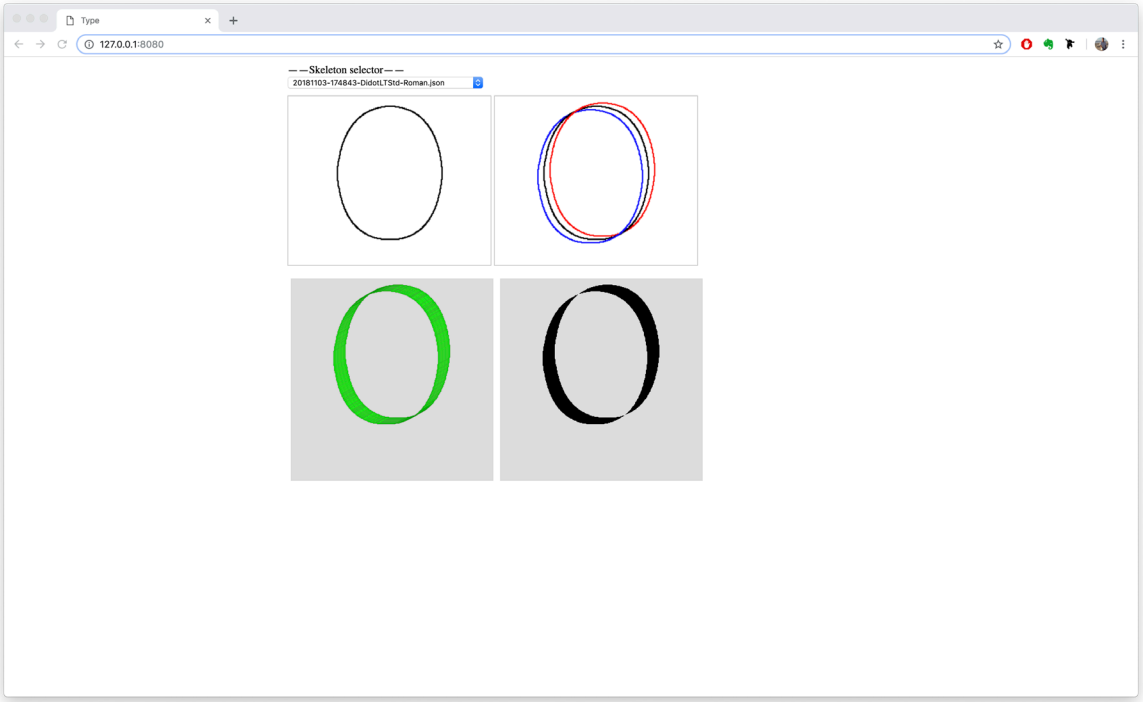


Figura 74
Sistema após a primeira iteração da segunda abordagem da técnica de desenho *calligraphy*.

Após conseguirmos desenhar os glifos com sucesso, era possível observar que ainda existiam algumas imperfeições nas partes onde os contornos, interior e exterior, se cruzavam. Para perceber qual era a origem das imperfeições exportamos a fonte (Figura 75) e analisamos em detalhe os glifos gerados. Percebemos que na extração ainda surgiam mais defeitos. No entanto, conseguimos identificar que a origem do problema era a ordem pela qual os pontos estavam a ser ligados. Sendo assim, procedemos à ordenação de todos os pontos antes de desenhar os glifos. Como é possível observar na Figura 76, conseguimos corrigir as imperfeições com sucesso e alcançar resultados satisfatórios no que diz respeito ao aspeto visual dos glifos, pois foi possível replicar o estilo visual caligráfico.

Realizámos ainda testes da fonte gerada no *Adobe InDesign* (Figura 77) de forma a testar a sua performance. Em comparação com os testes feitos com a primeira fonte exportada, verificamos uma redução substancial no tempo necessário para ser carregada e possível de usar. Utilizando SVG para a criação dos glifos, conseguimos otimizar a fonte gerada e por sua vez facilitar a sua utilização. No entanto, continuava a existir o problema do *kerning*, algo que será abordado mais adiante.

Depois de termos conseguido replicar com sucesso o estilo caligráfico, realizámos algumas explorações ainda nesta técnica de desenho com base na fonte gerada da Figura 75. Esta apresentava imperfeições que lhe davam um aspeto visual muito interessante. Tendo em conta que sabíamos que a causa das imperfeições era a ordem como os pontos eram conectados, exploramos várias possibilidades para criar

Figura 75 e 76
Em cima, fonte utilizada para detetar imperfeições.
Em Baixo, fonte após correção de imperfeições.

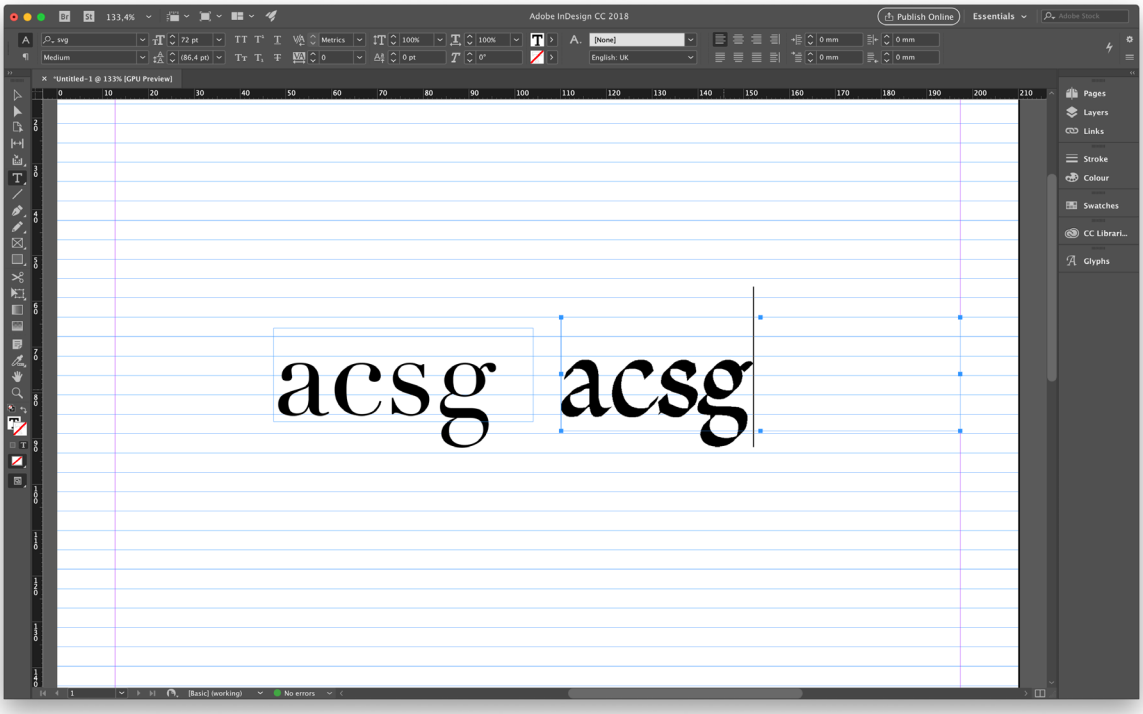
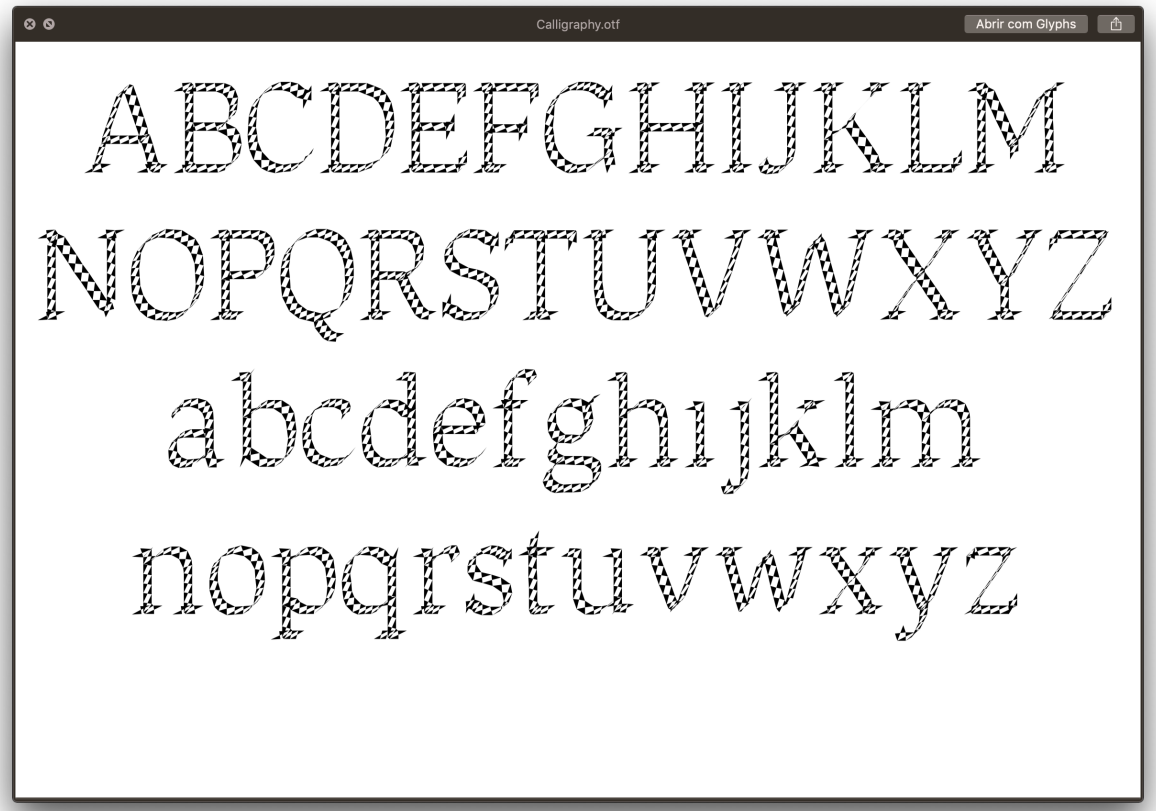


Figura 77
Teste de fonte gerada no *Adobe InDesign*.

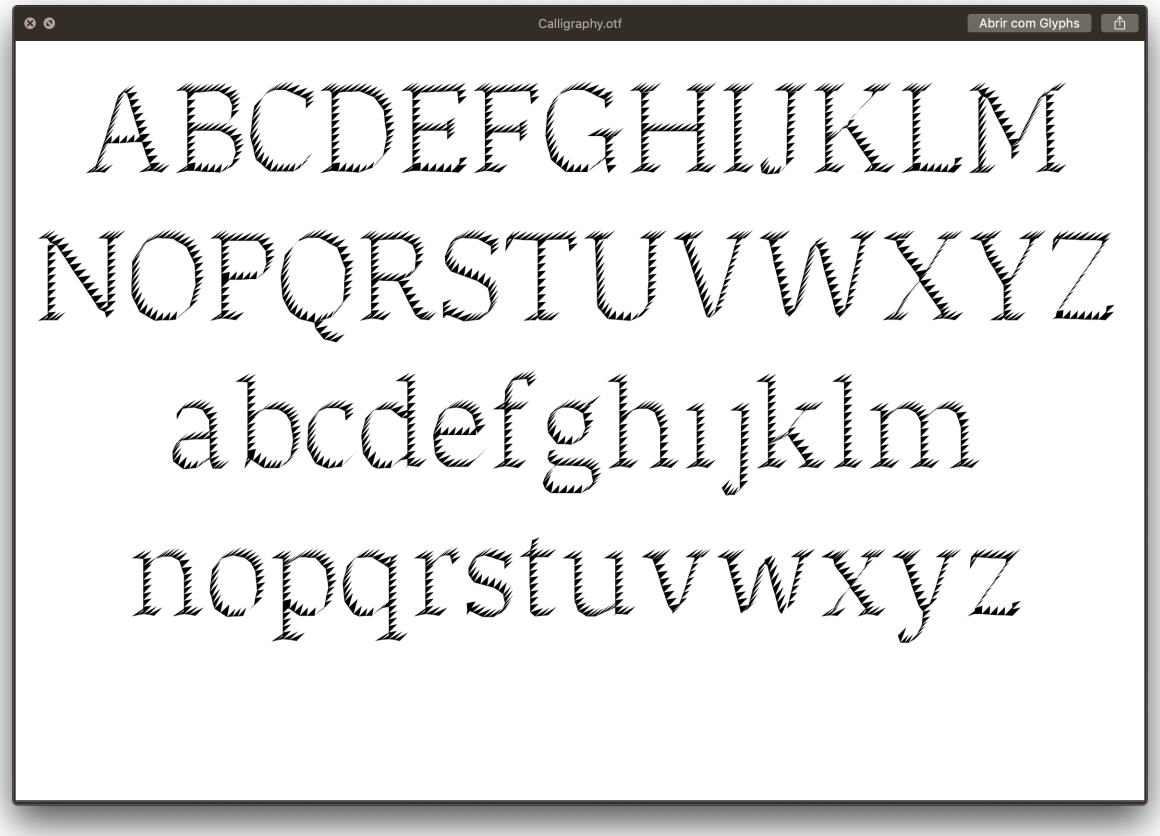
múltiplas variantes. No final, conseguimos obter três variantes do estilo principal; na Figura 78, vemos a primeira variante, na qual invertemos duas vezes a ordem dos pontos; na Figura 79, a segunda variante, alcançada após inverter a ordem apenas uma vez; e por fim, na Figura 80, a terceira variante, obtida através da remoção de um dos pontos que constrói o polígono. Com estas variantes do estilo principal conseguimos alcançar fontes com um padrão visualmente interessante, através da utilização de formas geométricas básicas. Nesta técnica de desenho, o utilizador pode controlar cinco parâmetros: o detalhe do esqueleto tipográfico; o ângulo do traço; o comprimento do traço; o comprimento relativo a uma forma geométrica individual e por fim, que tipo de forma geométrica é desenhada.

Figura 78, 79 e 80
À direita, de cima para baixo, primeira, segunda e terceira variantes do estilo caligráfico .

Do tempo utilizado para o desenvolvimento de todas as técnicas de desenho, esta foi a que demorou mais tempo. A razão para tal deve-se ao tempo empregue na procura da tecnologia certa para a criação dos glifos e também dos problemas que surgiram durante a implementação. Desta forma, esta técnica de desenho deu o mote para o desenvolvimento das restantes, pois já sabíamos que tecnologia usar e como criar o método para criar os glifos.


Calligraphy.otf
Abrir com Glyphs
ABCDEFGHIJKLM
NOPQRSTUVWXYZ
abcdefghijklm
nopqrstuvwxyz


Calligraphy.otf
Abrir com Glyphs
ABCDEFGHIJKLM
NOPQRSTUVWXYZ
abcdefghijklm
nopqrstuvwxyz



2ª Técnica de desenho — *Strips*

Para a exploração da segunda técnica de desenho decidimos basear-mo-nos na primeira, ou seja, partimos da mesma estrutura. Como tal, usamos o mesmo método de desenho, alterando apenas algumas configurações de forma a explorar a possibilidade de criar um padrão de listas. Para tal, optamos por não desenhar polígonos em todos os pontos recebidos do esqueleto tipográfico, mas sim apenas de dois em dois, de forma consecutiva.

A Figura 81, exibe o resultado da primeira implementação desta abordagem. É possível observar que o objetivo dos glifos terem um padrão de listas foi alcançado. No entanto, existiam muitas imperfeições, por exemplo, alguns dos traços eram maiores em relação a outros. Isto era provocado pelo facto de algumas das linhas do esqueleto tipográfico terem mais pontos que outras. Este problema só foi resolvido, após a fase de testes — esta fase será discutida mais à frente numa seção dedicada —, por sugestão de um dos utilizadores que testou o sistema *web*. Assim sendo, a solução passou por criar um método que transformasse uma linha com poucos pontos, numa linha com mais pontos. Dada uma linha construída apenas com dois pontos, é definido, pelo utilizador, o número de pontos a serem adicionados, que serão pos-

teriormente distribuídos numa nova linha. Este processo é realizado para cada conjunto de dois pontos, de forma sequencial, até percorrer todos pontos que definem a estrutura do esqueleto. Desta forma, foi possível uniformizar o padrão, ficando com um aspeto visual mais consistente, como é possível observar na Figura 82. Este método foi também implementado no desenvolvimento das técnicas de desenho seguintes. Nesta técnica de desenho, o utilizador pode controlar quatro parâmetros: o detalhe do esqueleto tipográfico; o ângulo do traço; o comprimento do traço; e o comprimento relativo às listas que são desenhadas.

Figura 81
Fonte gerada após a implementação da primeira abordagem da técnica de desenho *strips*.

Figura 82
Fonte gerada após a implementação da abordagem final da técnica de desenho *strips*.

3ª Técnica de desenho — *Pixels*

A terceira técnica de desenho foi baseada numa abordagem aos meios digitais. Para tal, exploramos o conceito visual de pixel por representar bem ambientes digitais. Este tipo de abordagens já tem sido exploradas em alguns trabalhos, nomeadamente no projeto *Post Bitmap Scripter* — ver subsecção 4.2 —, e de que certa forma influenciou a nossa abordagem. A implementação desta técnica de desenho também foi baseada na primeira técnica de desenho, na qual foi apenas adicionada uma funcionalidade para a redução de detalhe do traço com o intuito de produzir o efeito pixelizado.

As primeiras implementações desta abordagem revelaram alguns problemas, nomeadamente em termos de tamanho dos glifos, como é possível observar na Figura 83. Ao aplicarmos a redução de detalhe ao traço, também era reduzido o tamanho do traço, assim sendo, a solução passou por redimensionar o tamanho do traço após a redução de detalhe (Figura 84). Nesta técnica de desenho, o utilizador pode controlar apenas o ângulo e comprimento do traço.

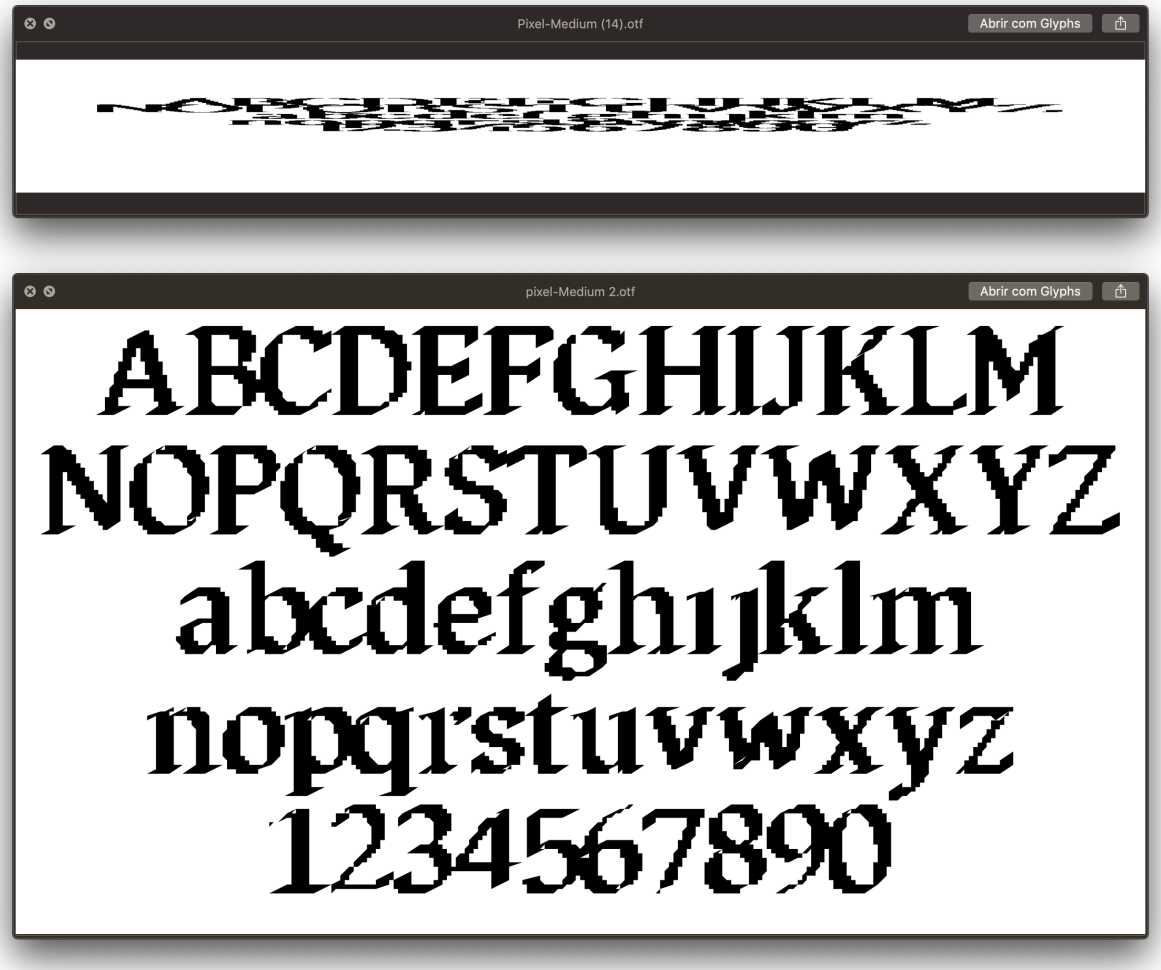


Figura 83 e 84
Em cima, a primeira implementação da técnica de desenho *pixels*. Em baixo, fonte gerada após correções.

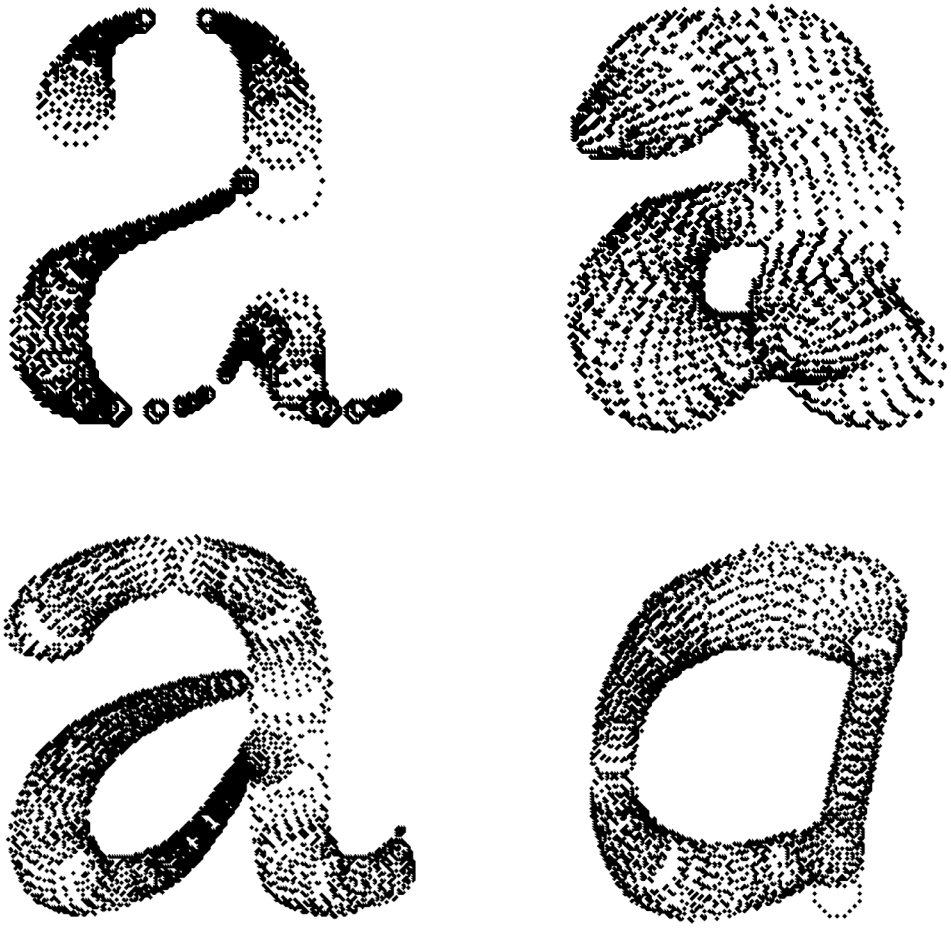
4ª Técnica de desenho — *Geometry*

Para esta técnica de desenho exploramos a hipótese de criar glifos com o contorno similar ao da fonte original. Contudo, pretendíamos criar formas mais abstratas do que as originais, razão pela qual nos baseámos no projeto *Gothic Lab* — ver subsecção 4.2. Desta forma, fizemos uso do raio recebido do esqueleto tipográfico para criar círculos nos quais estariam um certo número de retângulos.

Na Figura 85, são visíveis as primeiras explorações desta abordagem e na Figura 86, o resultado final. Decidimos utilizar o quadrado como forma geométrica, o que acabou por gerar glifos com um aspeto visualmente interessante.

Apesar dos glifos serem compostos por um grande número de quadrados, quando são agrupados em pequena distância formam uma forma abstrata, muito parecida com um cardume de peixes. Nesta técnica de desenho, o utilizador pode controlar três parâmetros: o detalhe do esqueleto tipográfico; o ângulo que determina a posição dos quadrados; o comprimento do traço, ou seja, o número de quadrados; e a distância pela qual os quadrados são posicionados.

Figura 85
Primeiras explorações para a técnica de desenho *Geometry*.



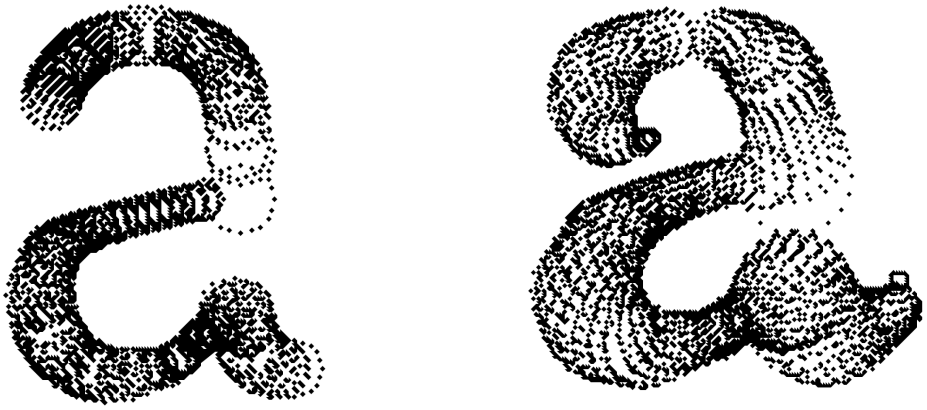
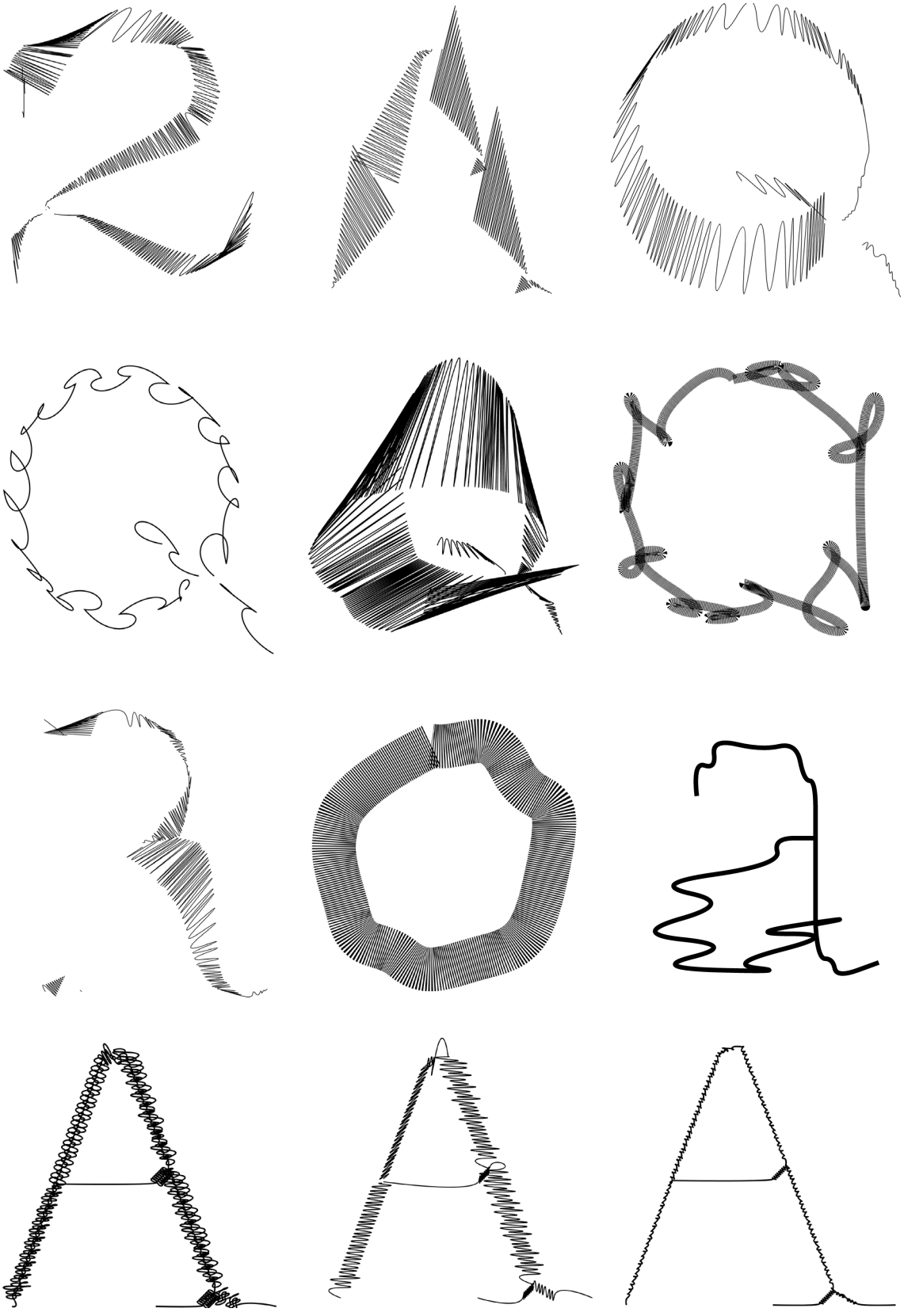


Figura 86
Em cima, fonte final gerada com a técnica de desenho *Geometry*.

5ª Técnica de desenho — *Curves*

Na abordagem à quinta técnica de desenho decidimos explorar o desenho dos glifos através de curvas de *Bézier* com o objetivo de conseguir formas menos lineares. A principal referência para esta abordagem foi o tipo de letra generativo *Irratio* — ver subsecção 4.2 —, pois utilizámos o mesmo princípio de construção dos glifos, ou seja, fazer a ligação das curvas de *Bézier* por pontos sucessivos do esqueleto tipográfico. Deste modo, começamos por aplicar esta abordagem em apenas alguns glifos (Figura 87).

Figura 87
À direita, primeiras explorações para a técnica de desenho *Curves*.



Como é possível observar na Figura 87, conseguimos alcançar resultados visualmente interessantes, através da repetição sequencial das curvas de *Bézier* nos pontos do esqueleto tipográfico. Os glifos tinham um padrão visualmente apelativo e era possível criar inúmeras variações para o mesmo glifo, o que tornava esta abordagem bastante visualmente dinâmica.

No entanto, apesar das tentativas, não conseguimos exportar os glifos como fonte OTF. Isto deve-se ao facto da biblioteca Opentype.js necessitar de um *path* fechado para gerar um glifo, e para gerar os glifos da Figura 87, era necessário o *path* estar aberto. Desta forma, optamos por tentar criar formas fechadas com as curvas de *Bézier* com o intuito de poder exportar a fonte posteriormente.

Na Figura 88, são visíveis as implementações desta segunda abordagem, na qual já foi possível exportar a fonte em formato OTF, com a geração dos glifos através de elipses. Nesta, são também visíveis algumas imperfeições, pois nem todos os traços apresentavam o mesmo comprimento. Estas imperfeições foram posteriormente corrigidas com o método de adicionar novos pontos nas linhas — anteriormente explicado —, a par da introdução de cinco novas formas, como é possível observar nas Figuras 89 a 94. Nesta técnica de desenho, o utilizador pode controlar quatro parâmetros: o detalhe do esqueleto tipográfico, o comprimento do traço, ou seja, das formas desenhadas e que tipo de forma é desenhada.

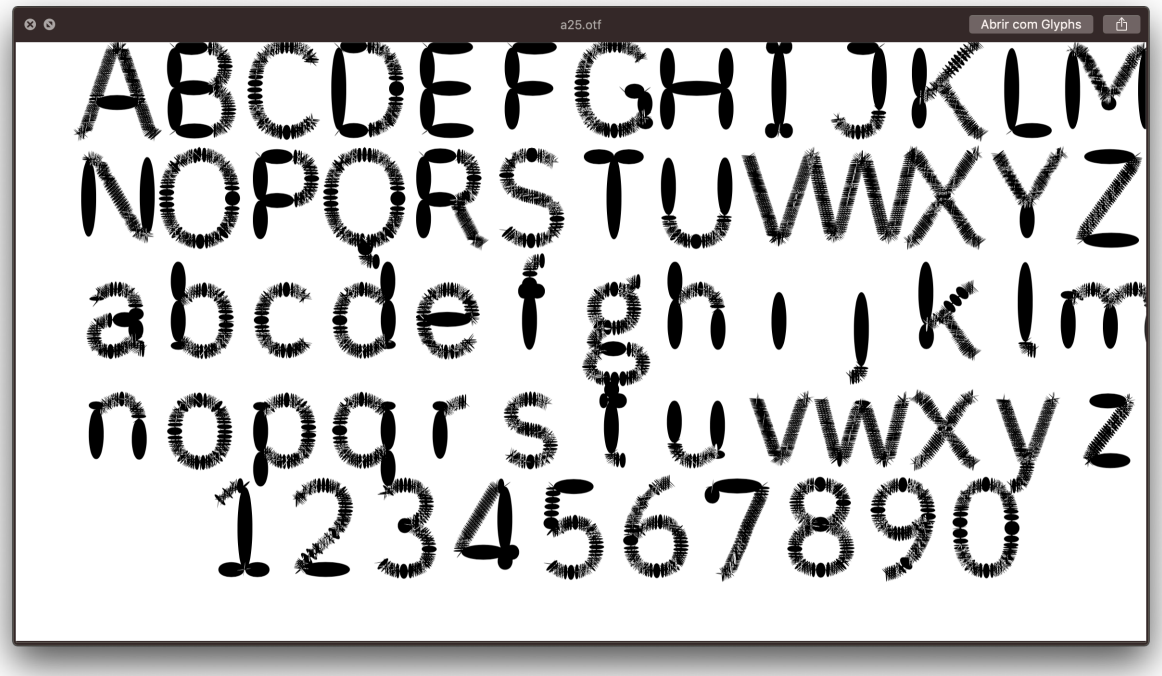


Figura 88
Primeira fonte gerada em formato OTF com a técnica de desenho *Curves*.

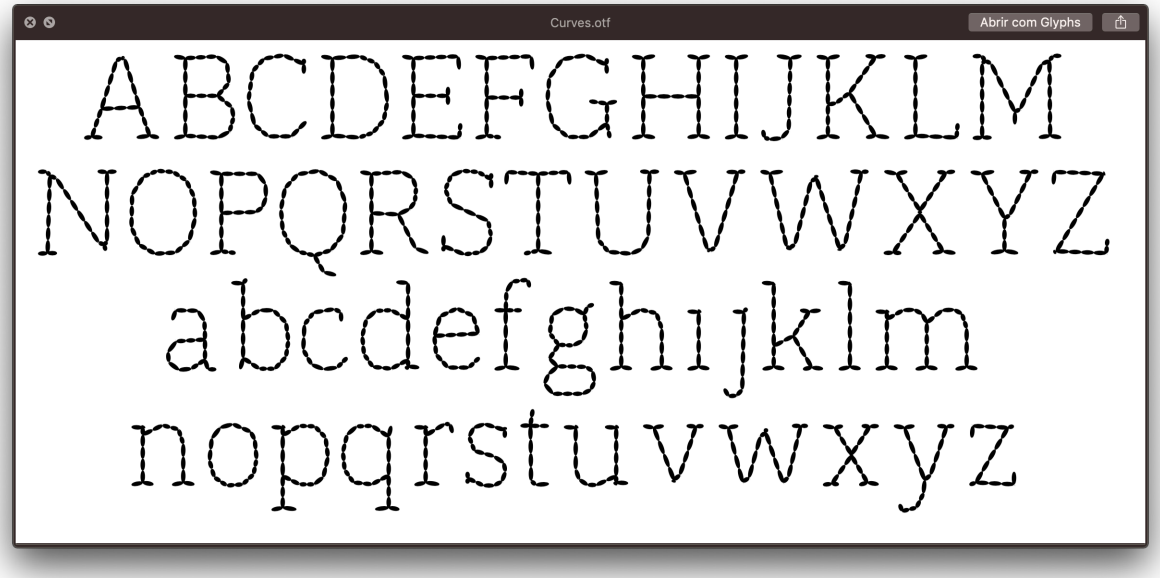
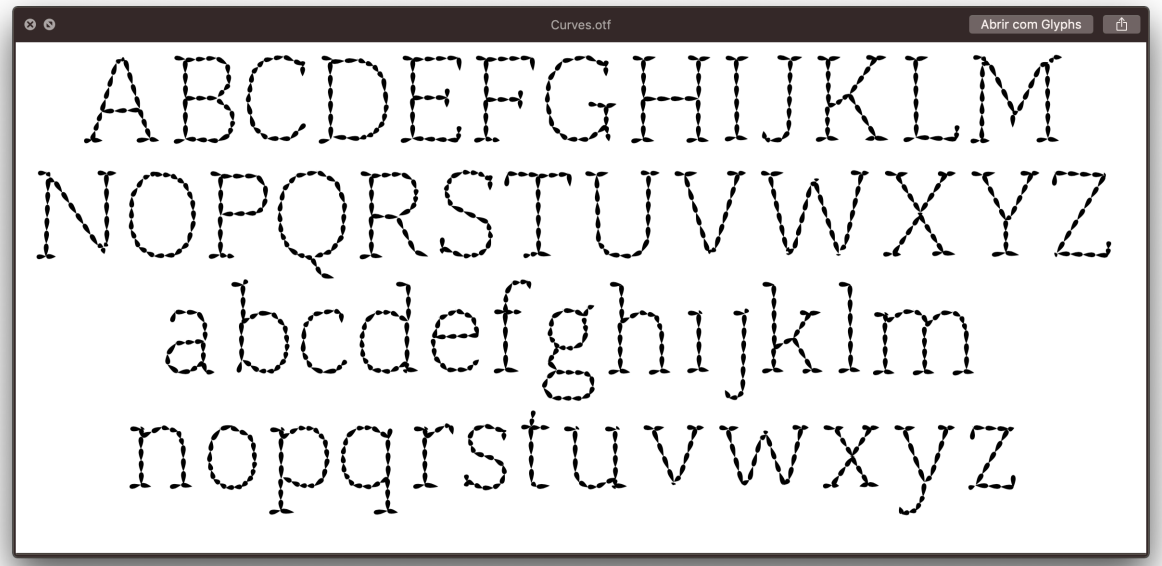


Figura 89 e 90
Variações possíveis com a técnica de desenho *Curves*.



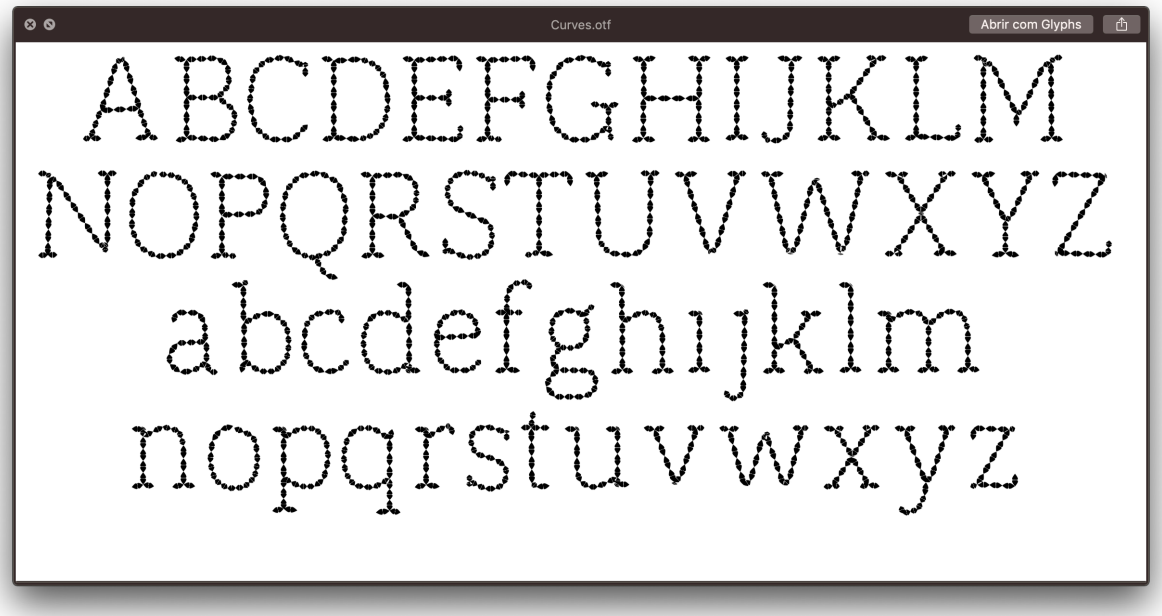


Figura 91 e 92
Variações possíveis com a técnica de desenho *Curves*.

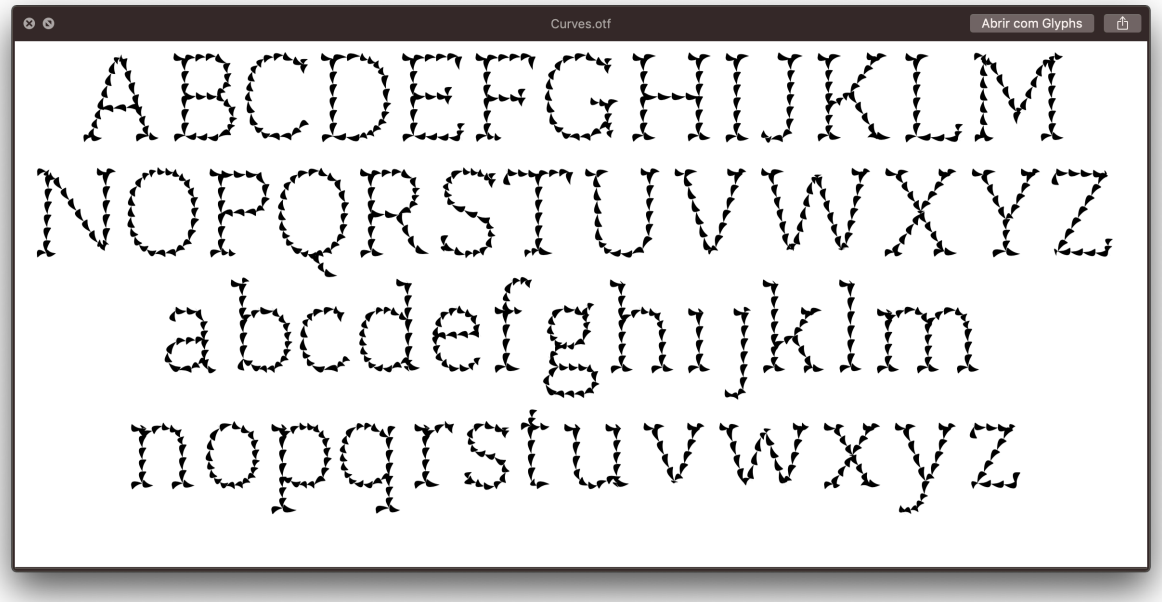


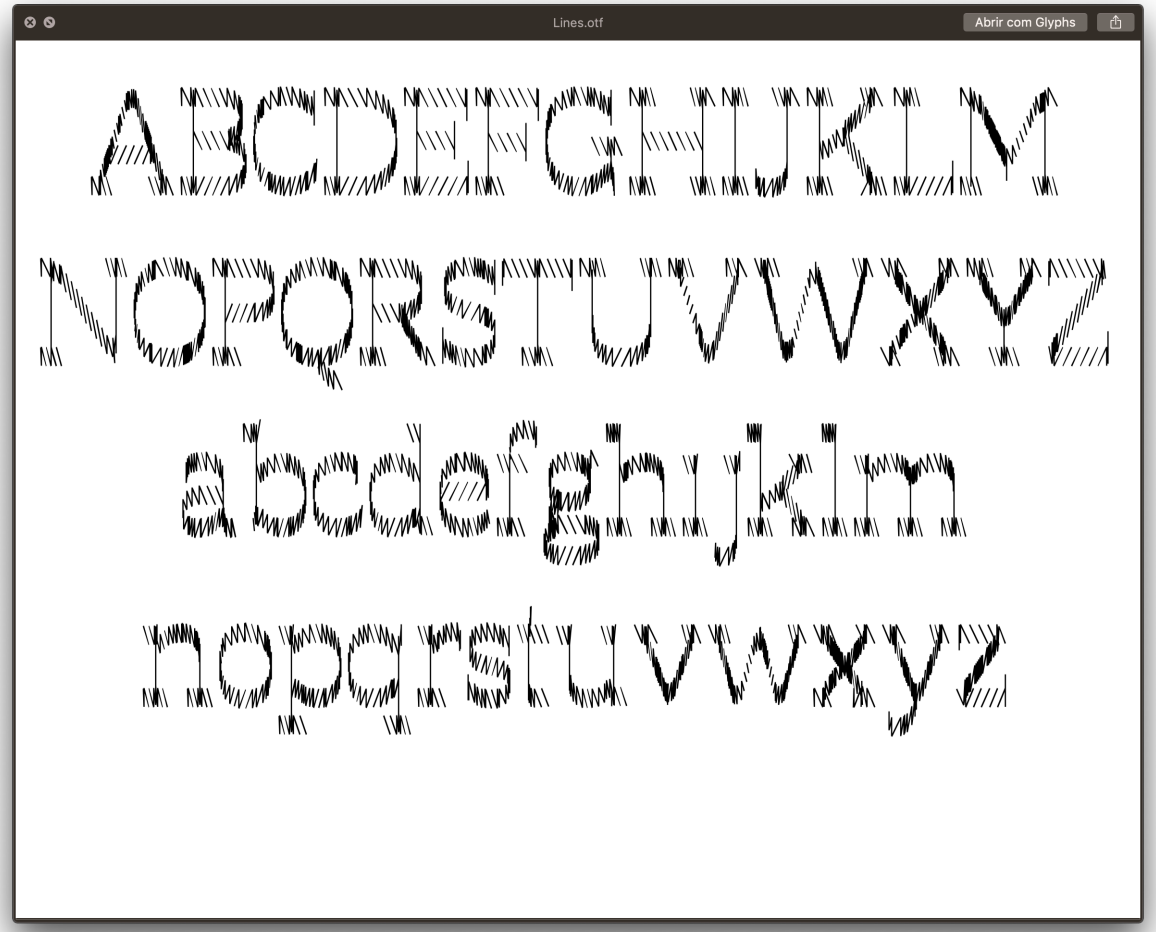
Figura 93 e 94
Variações possíveis com a técnica de desenho *Curves*.

6ª Técnica de desenho — *Lines*

A abordagem para a sexta técnica de desenho surgiu com a intenção de conseguir o efeito presente nas primeiras explorações da quinta técnica de desenho (Figura 87). Sabendo à partida que não conseguíamos exportar os glifos, pela forma como estavam a ser desenhados, optamos por tentar criar um efeito similar, mas com recurso a retângulos no lugar das curvas de *Bézier*.

A Figura 95, exibe a primeira iteração desta abordagem e como podemos observar, conseguimos alcançar um efeito similar apesar de possuir alguns defeitos. Estes defeitos eram provocados pelo nível de detalhe do esqueleto, desta forma na segunda iteração desta abordagem conseguimos corrigir os defeitos (Figura 96). Nos casos em que apenas existem dois pontos a definir a linha do esqueleto, é feita apenas uma ligação e por sua vez é criada um contraste muito grande com as zonas de mais linhas. Este efeito, não era previsto, no entanto, achamos visualmente interessante e optamos por definir com uma das variações possíveis. Nesta técnica de desenho, o utilizador pode controlar cinco parâmetros: o detalhe do esqueleto tipográfico, o ângulo do retângulo, o comprimento do retângulo, o número de retângulos e, por fim, a altura de cada retângulo.

Figura 95 e 96
Em baixo, primeira iteração com a técnica de desenho *Lines*. À direita, a iteração final.



7ª Técnica de desenho — *Noise*

A sétima técnica de desenho surgiu com a intenção de explorar uma abordagem capaz de gerar glifos com formas sempre aleatórias, ou seja, gerar sempre formas diferentes umas das outras. Para tal, baseámos esta abordagem principalmente na sexta técnica de desenho (*Lines*), e no trabalho *Growing Data* — ver subsecção 4.2 —, tirando partido do processo de criação dos glifos e da qualidade visual que os glifos apresentam, respetivamente. Desta forma, começámos por adicionar ao método de criação dos glifos da sexta técnica de desenho, um parâmetro referente ao ângulo pelo qual os retângulos seriam posicionados. De seguida definimos que este parâmetro tomaria sempre valores aleatórios dentro de um intervalo predefinido, com o intuito de gerar sempre formas diferentes e únicas.

Da Figura 97 a 99, são visíveis as primeiras iterações desta abordagem. Podemos observar, que o aspeto estava visualmente interessante, no entanto, precisava de ser refinado, pois existiam muitas imperfeições. Muitos traços verticais e horizontais dos glifos não apresentavam

qualquer semelhança com os traços diagonais, o que causava pouca coerência no glifo como forma individual e, por sua vez, no conjunto dos glifos como fonte. Também visível é que os glifos estavam com demasiados retângulos numa curta distância e estes estavam deformados. Desta forma procedemos à reordenação dos pontos dos retângulos, para corrigir as deformações e também a uma maior distanciação dos mesmos em termos de posicionamento para corrigir a sobreposição em demasia. Nesta iteração, alcançamos resultados satisfatórios (Figura 100), no entanto continuavam a existir as diferenças entre os traços verticais/horizontais e diagonais.

Figura 97 a 99
Primeiras iterações com a técnica de desenho *Noise*.

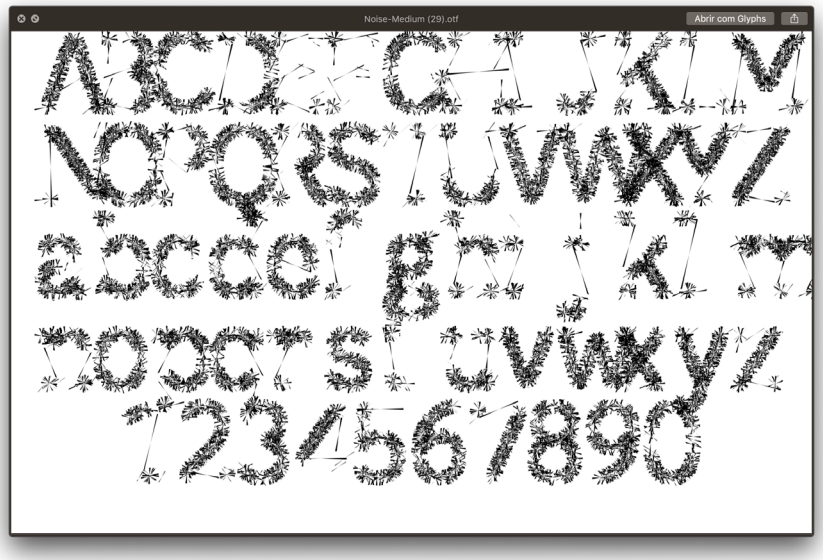


Figura 100
Primeira fonte razoavelmente satisfatória com a técnica de desenho *Noise*.

Este problema foi posteriormente corrigido com recurso ao método de adição de novos pontos nas linhas, que possibilitou desenhar retângulos nessas novas coordenadas, como é possível observar na Figura 101. Nesta técnica de desenho, o utilizador pode controlar três parâmetros: o comprimento do traço, a grossura do traço e, por fim, o nível de retângulos que compõem o traço.

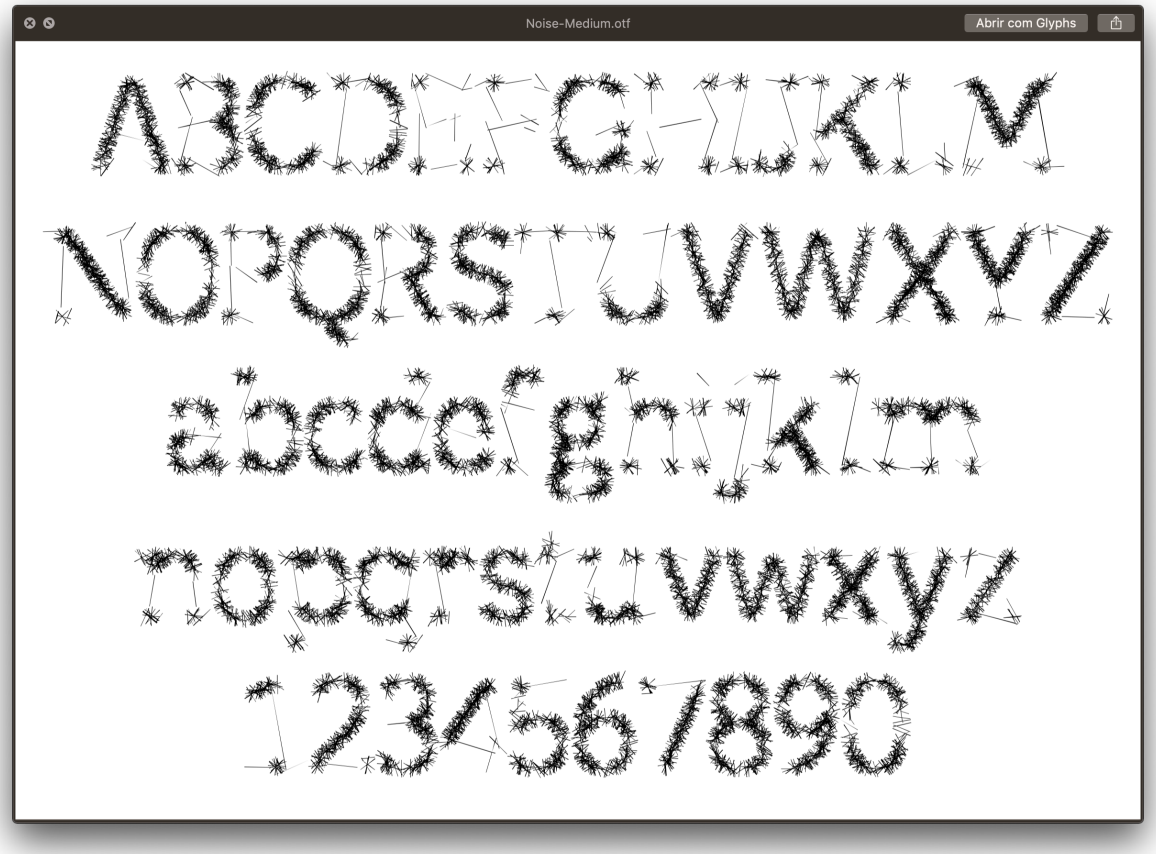


Figura 101
Fonte numa iteração final da técnica de desenho *Noise*.

8ª Técnica de desenho — *Dots*

Para a oitava técnica de desenho decidimos explorar uma abordagem com uma maior componente modular no processo de geração dos glifos. O trabalho devolvido por pelo estúdio MuirMcNeil, teve grande influência para a nossa abordagem, na medida em que trabalhos como *TwoPoint, ThreePoint* e *Intersect* — ver subsecção 4.2 — forma uma referência em termos de aspeto visual e em termos de possíveis variações. Desta forma, criamos uma técnica de desenho que por cada ponto do esqueleto tipográfico, desenha uma grelha de quadrados. Com esta grelha, é criado um efeito de duplicação através da repetição dos quadrados de forma alinhada.

Os primeiros resultados obtidos com esta abordagem apresentavam o efeito desejado, no entanto, continham muitos traços em que não eram desenhados quaisquer quadrados (Figura 102 a 105). Este defeito era provocado pela falta de pontos nas linhas verticais e horizontais do glifo, pois no caso dos traços diagonais isto já não acontecia como é possível verificar na fonte em itálico. Para a resolução deste problema, aplicamos o método de adicionar novos pontos nas linhas. Desta forma, conseguimos criar consistência por todos os traços que compõem o glifo e alcançar coerência entre todos os glifos (Figura 106). Nesta técnica de desenho, o utilizador pode controlar cinco pa-

râmetros: o detalhe do esqueleto tipográfico, o número de duplicação dos quadrados que forma a grelha, o ângulo que determina a posição dos quadrados, a distância pela qual são separados os quadrados e o número de quadrados que são desenhados.

Figura 102 a 105
Primeiras iterações com a técnica de desenho *Dots*.

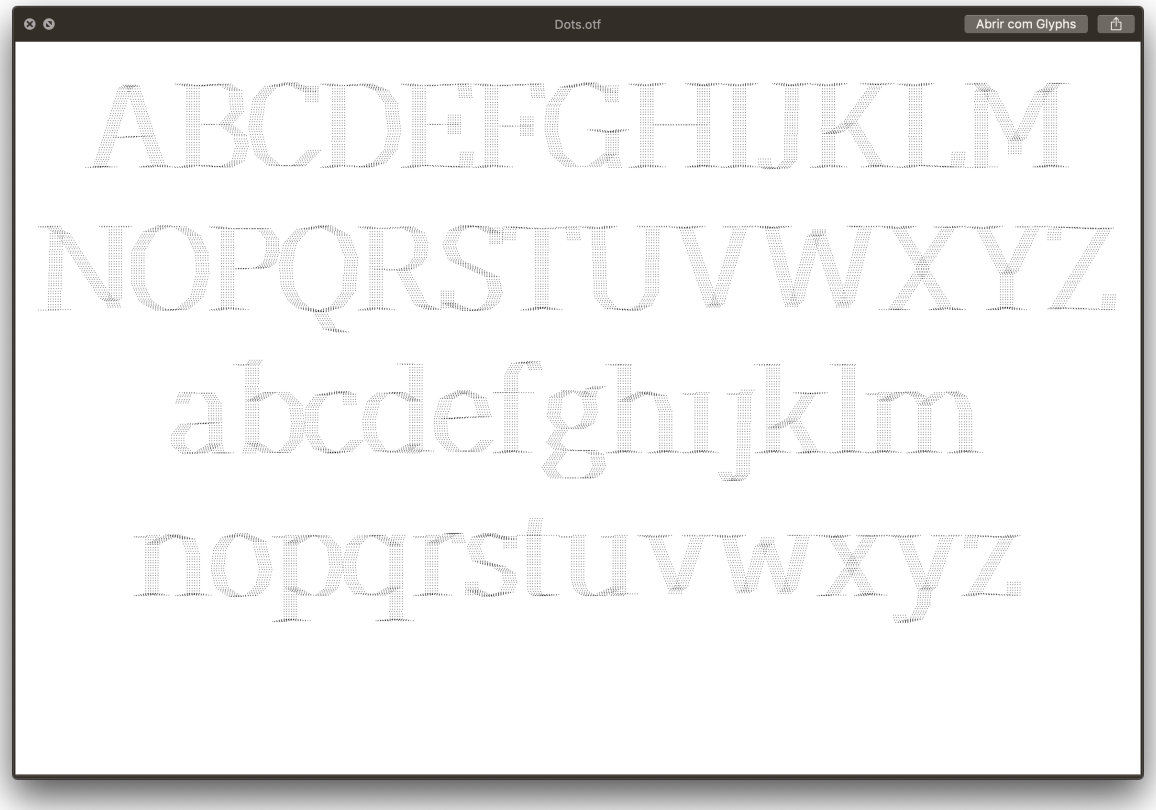


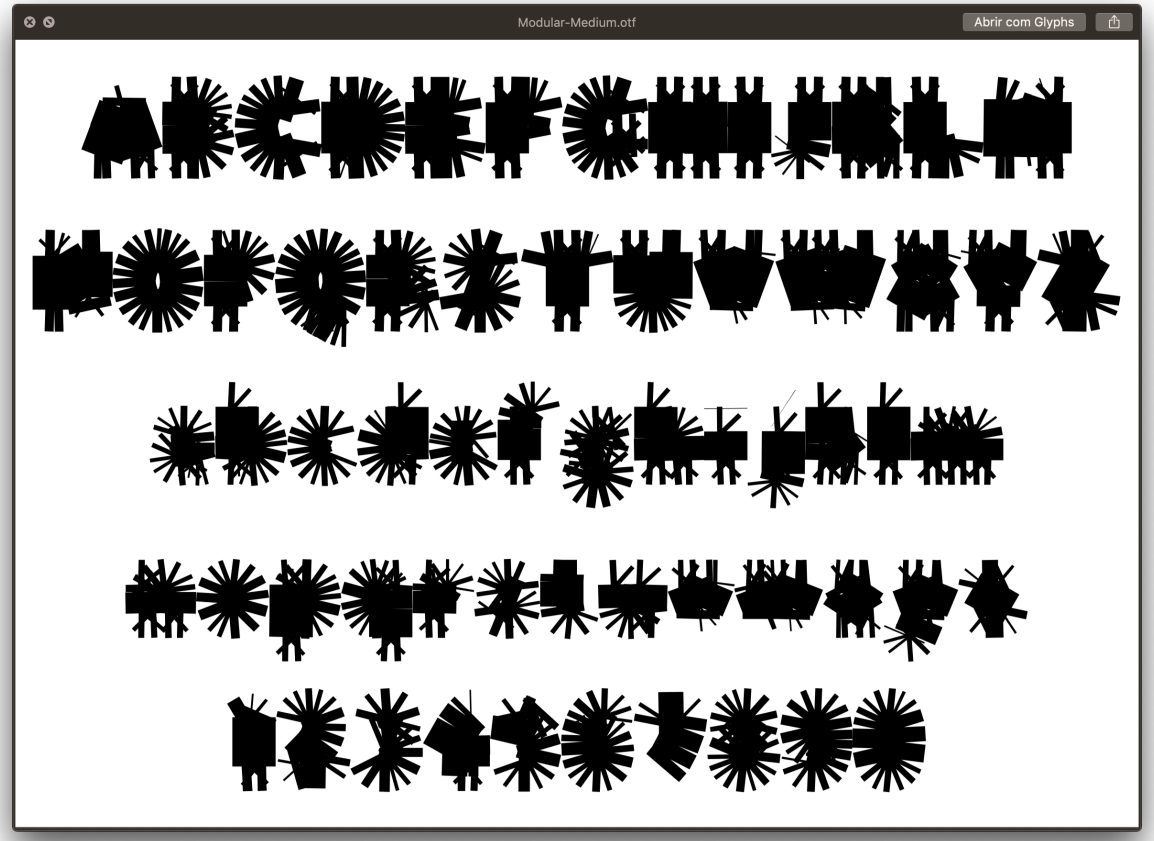
Figura 106
Em cima, fonte gerada na iteração final da técnica de desenho *Dots*.

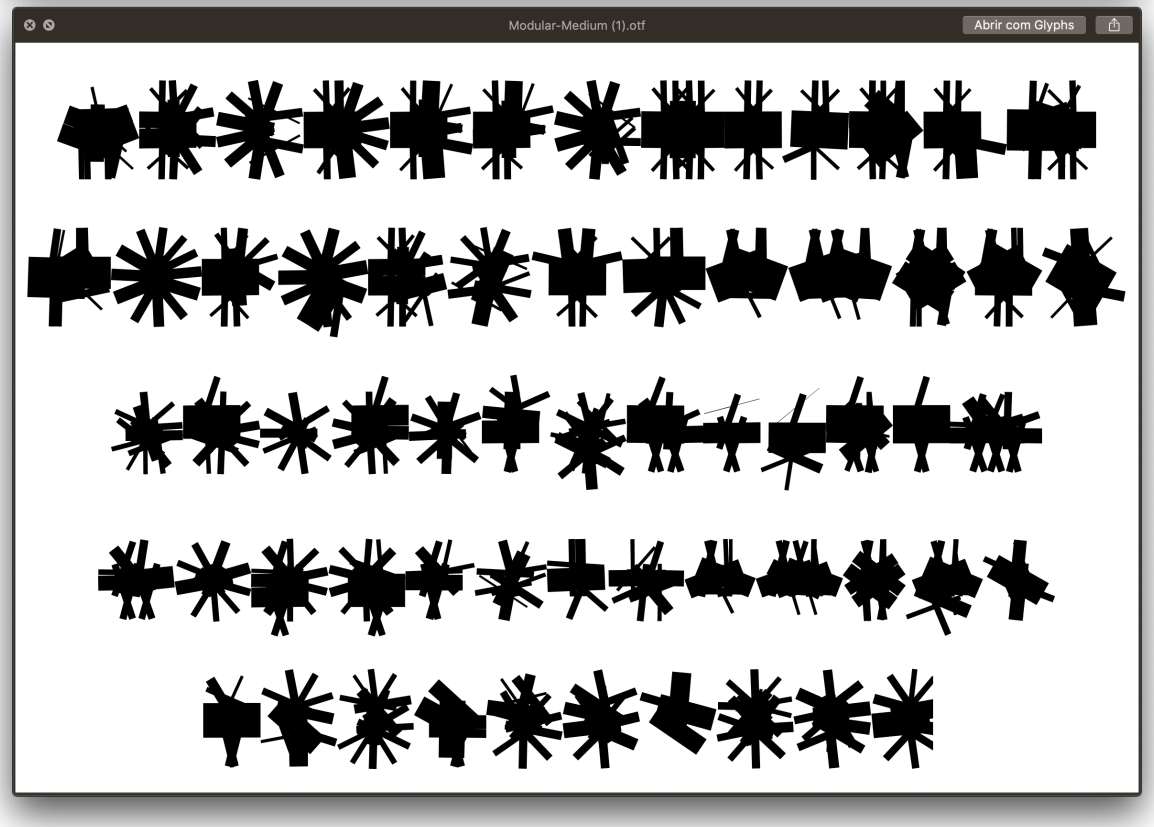
9ª Técnica de desenho — *Modular*

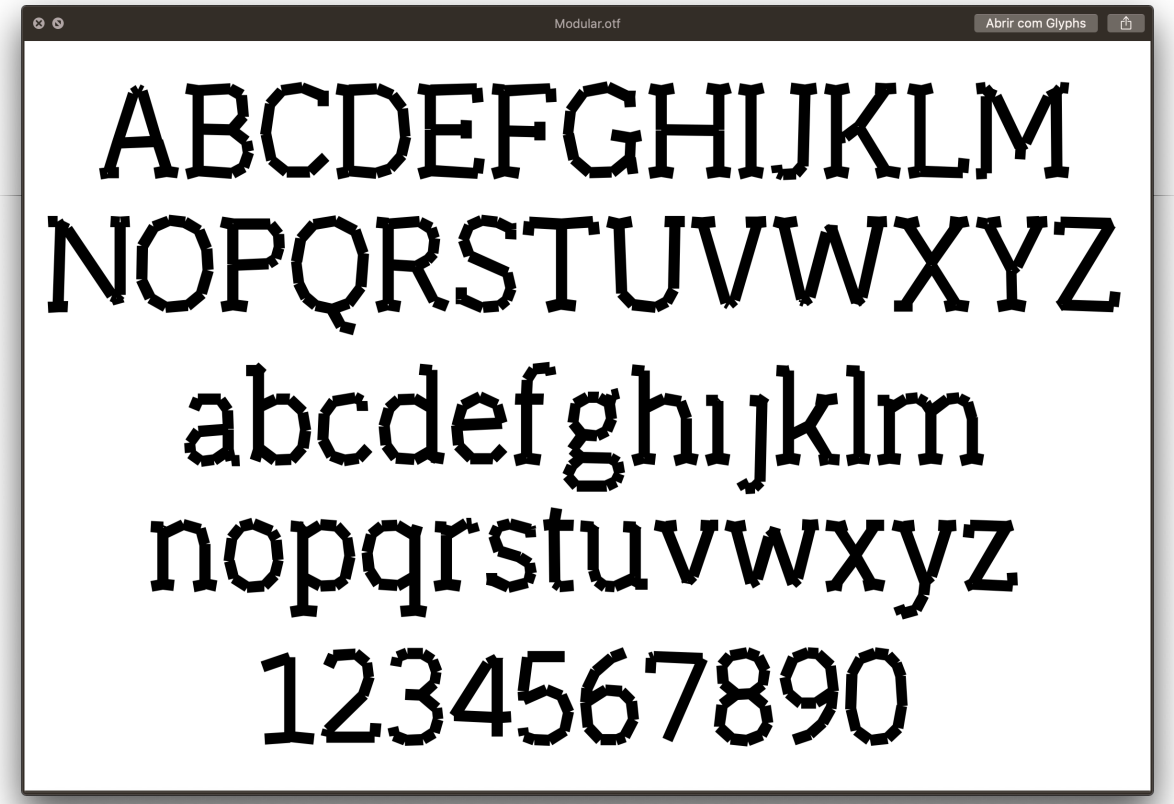
A par com a técnica de desenho *Dots*, esta técnica de desenho também surgiu com a intenção de adotar uma abordagem mais modular na criação dos glifos. No entanto, procurámos por basear esta abordagem em trabalhos que a criação de glifos apenas através de forma geométricas num sistema de relações abstratas. Exemplo disso, é o logotipo da revista *De Stijl (1917),* criado por Vilmos Huszár, no qual é explorada a criação de um alfabeto com recurso apenas a elementos geométricos perpendiculares — neste caso, retângulos. Tendo por base este trabalho, iniciamos a construção dos glifos também apenas com retângulos. Como é possível observar nas primeiras iterações desta abordagem (Figura 107 e 108), os glifos não apresentavam uma forma percetível devido ao grande número de retângulos sobrepostos a par da sua exagerada dimensão. Tendo isto em conta, procedemos à resolução destes dois defeitos e alcançamos com sucesso o efeito pretendido para os glifos. Foram ainda adicionadas posteriormente, três novas formas geométricas através da reordenação dos pontos que definiam o retângulo, conseguindo dar mais dinâmica e variedade a esta técnica de desenho (Figura 109 a 112). Nesta técnica de desenho, o utilizador pode controlar três parâmetros: o detalhe do esqueleto tipográfico, a grossura das formas geométricas e que tipo de forma pretende que seja desenhada.

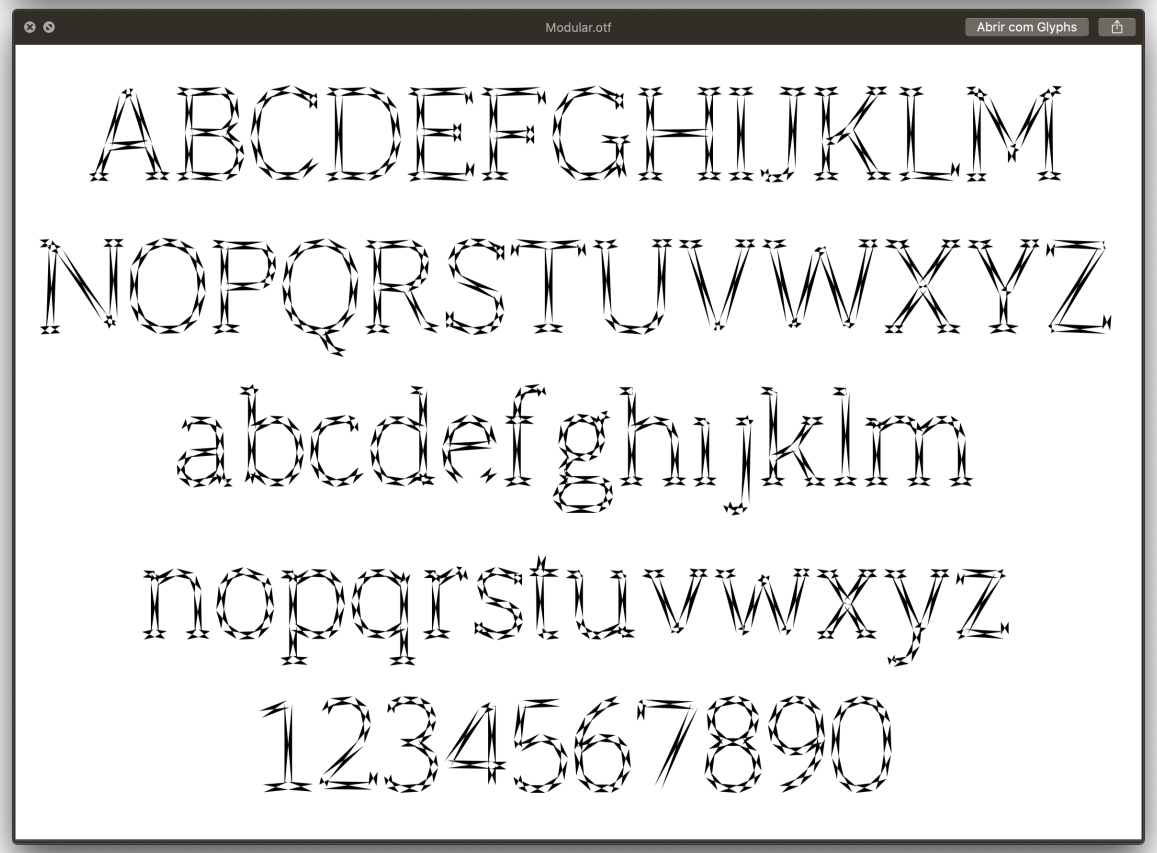
Figura 107 e 108
À direita, primeiras iterações com a técnica de desenho *Modular*.

Figura 109 e 112
Na páginas seguintes, variações da técnica de desenho *Modular*.

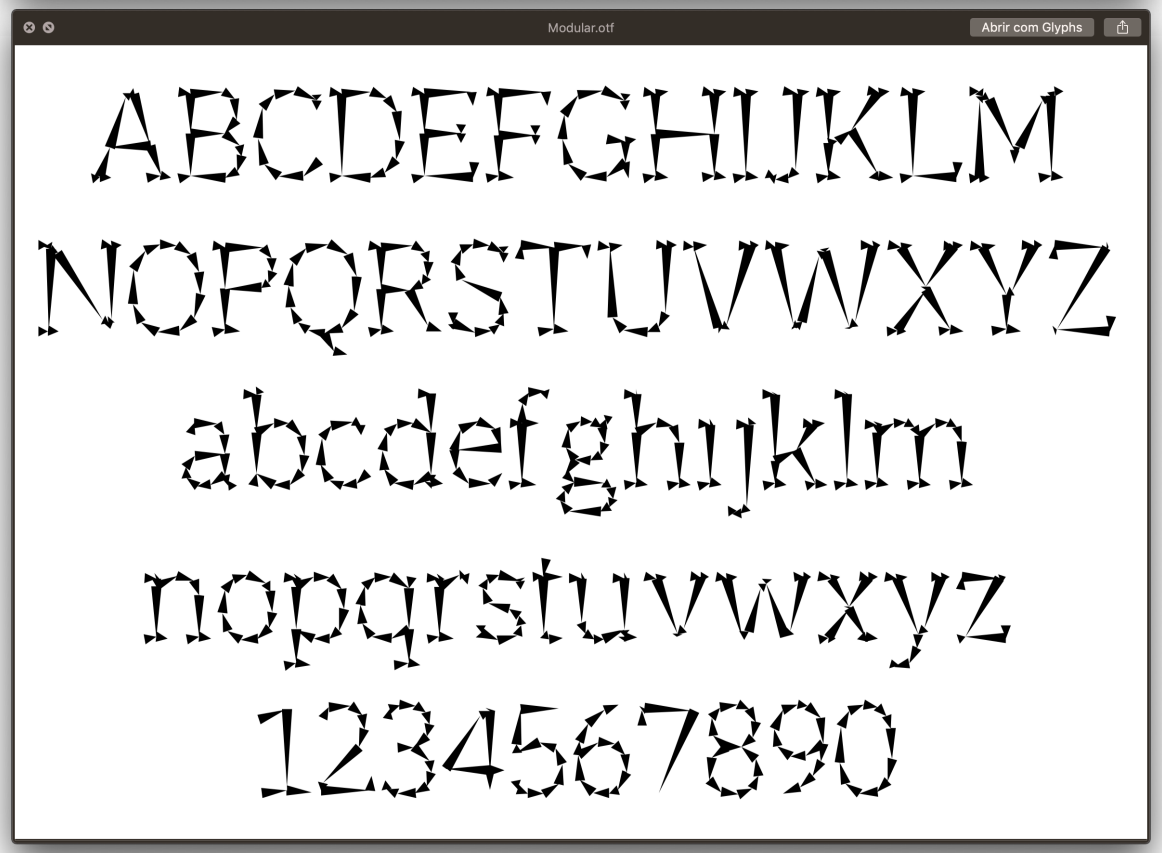

Modular-Medium.otf
Abrir com Glyphs


Modular-Medium (1).otf
Abrir com Glyphs


Modular.otf
Abrir com Glyphs
ABCDEFGHIJKLM
NOPQRSTUVWXYZ
abcdefghıjklm
nopqrstuvwxyz
1234567890


Modular.otf
Abrir com Glyphs
ABCDEFGHIJKLM
NOPQRSTUVWXYZ
abcdefghijklm
nopqrstuvwxyz
1234567890


Modular.otf
Abrir com Glyphs
ABCDEFGHIJKLM
NOPQRSTUVWXYZ
abcdefghijklm
nopqrstuvwxyz
1234567890


Modular.otf
Abrir com Glyphs
ABCDEFGHIJKLM
NOPQRSTUVWXYZ
abcdefghijklm
nopqrstuvwxyz
1234567890

Análise das técnicas de desenho

Depois de desenvolver as técnicas de desenho, podemos apresentar algumas observações gerais sobre os resultados. No que diz respeito à qualidade tipográfica dos glifos obtidos, é notável a alta transmissão de partes anatômicas da fonte original para a estrutura dos glifos gerados. Por exemplo, os glifos gerados a partir de um esqueleto tipográfico extraído de uma fonte serifada, vão também possuir serifas. O oposto acontece quando os glifos são gerados a partir de um esqueleto tipográfico extraído de uma fonte sem serifas, pois também não vão possuir serifas. Além disso, as principais relações entre os glifos são mantidas, por exemplo, a relação entre a altura-x, ascendentes e descendentes. Em termos de diversidade visual dos glifos criados, cada uma das técnicas de desenho apresenta uma abordagem diferente, e consequentemente, o conjunto de todas as técnicas de desenho é capaz de gerar uma vasta gama de glifos com formas distintas que, por sua vez, criam fontes com a mesma ampla gama de diversidade.

Avaliando as fontes geradas em termos de *legibility,* podemos afirmar que os glifos apresentam as características necessárias para a seu reconhecimento sem dificuldade. Mesmo em abordagens mais abstratas como são o caso das técnicas de desenho *Geometry* e *Noise,* os glifos que formam a fonte contêm as características, como estrutura, coerência e o padrão, necessárias para o seu reconhecimento.

No entanto, em termos de *Kerning,* as fontes geradas apresentam algumas imperfeições. Para que a fonte gerada tivesse os pares de *Kerning* corretos, seria necessário efetuar o demoroso processo de correção individual para cada par de glifos. Tendo em conta que a fonte é gerada em tempo real de acordo com as preferências do utilizador, não foi possível automatizar este processo de correção. A solução encontrada foi aplicar uma medida igual para todos os pares de *kerning*, desta forma, as fontes geradas têm valores predefinidos que posteriormente o utilizador pode refinar utilizando *software* de edição de fontes, por exemplo, o *Glyphs*.

5.2.4. Interface gráfica do utilizador

Nesta fase de desenvolvimento do projeto tínhamos a base do sistema *web* a funcionar e o próximo passo era agrupar as funcionalidades já desenvolvidas e criar as restantes necessárias de forma a construir como uma única ferramenta. Deste modo, demos início ao desenvolvimento da interface gráfica para o sistema *web*. Começamos por realizar prototipagem do *website* (Figura 113 a 115) antes de passar à sua implementação, com o intuito de perceber quais eram os requisitos gráficos referentes às funcionalidades do sistema.

Nas Figuras 116 e 117, é possível observar a implementação dos protótipos realizados, já com algumas alterações relativas à disposição de alguns botões. Apesar da interface ser simples e clara, após algum uso da nossa parte começávamos a sentir algumas dificuldades em

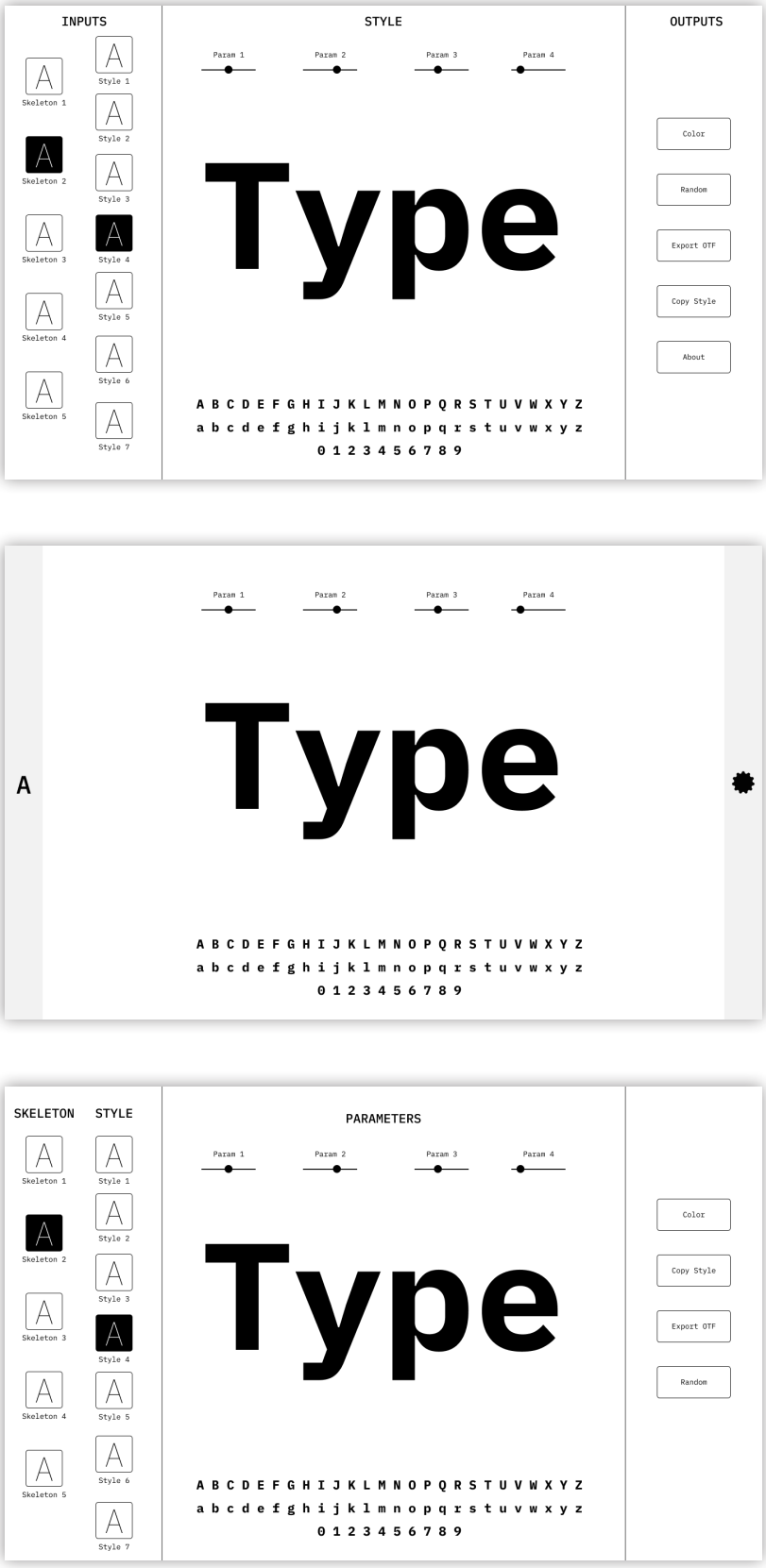
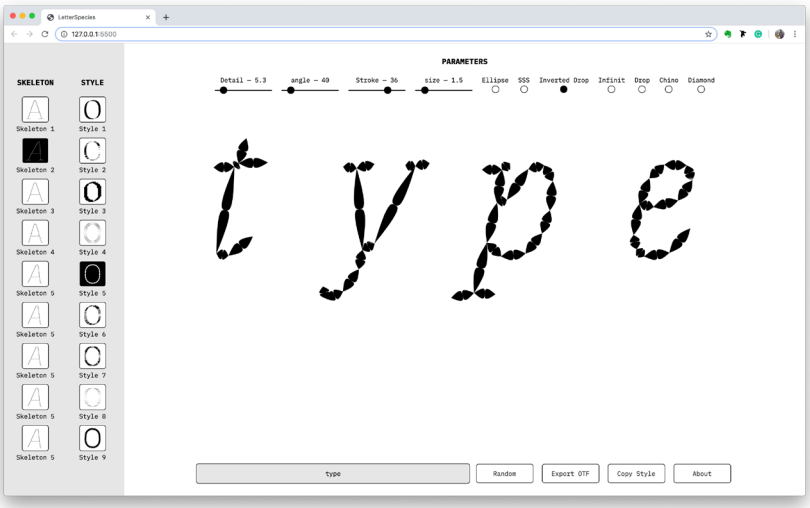


Figura 113 e 115
Protótipos realizados antes da implementação do sistema *web*.

certos tipos de interação. Por exemplo, o menu de escolha do esqueleto tipográfico foi desenhado com a intenção de fazer uma pequena pré-visualização de cada um dos esqueletos possíveis para a escolha do utilizador, assim como o menu das técnicas de desenho foi desenhado com o propósito de mostrar as diferenças entre cada técnica de desenho. No entanto, apesar da intenção ser ajudar o utilizador, as diferenças nem sempre eram percetíveis. Além disto, era necessária uma divisão para a visualização de cada esqueleto tipográfico e técnica de desenho, o que não facilitaria a posterior adição de novas opções. Desta forma, optamos redesenhar a interface e desistir de visualização de uma divisão por cada opção e criar apenas um menu *drop-down* com todas as opções.



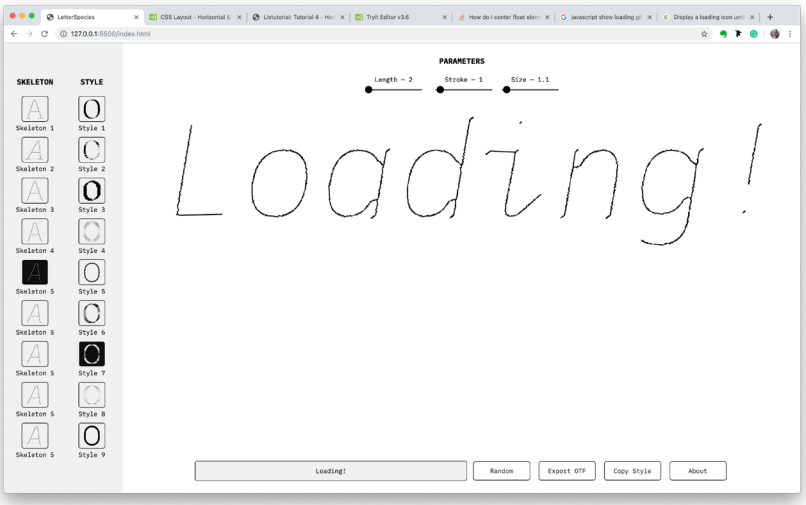


Figura 116 e 117
Primeira implementação do sistema *web* com base nos protótipos realizados.

A Figura 118, apresenta a interface gráfica final, após as correções e modificações realizadas com a finalidade de proporcionar a melhor interação ao utilizador. Nesta interface, o utilizador tem o controlo so-

bre todos os parâmetros de configuração visual dos glifos, assim como as opções para exportar a fonte gerada. Além da barra principal, que possui apenas funcionalidades que não afetam a geração dos glifos, o sistema *web* consiste em três seções principais: o painel de configurações no lado esquerdo, a área de pré-visualização no meio e o painel de exportação no lado direito. Esta divisão por secções é feita desta forma, com o intuito de ficar coerente com o funcionamento do sistema. O funcionamento de cada uma das secções é explicada detalhadamente nas subsecções seguintes.

Figura 118
Interface do sistema *web* na fase final de implementação.

Painel de Configuração

No painel de configuração, estão presentes todos os parâmetros que influenciam a criação e visualização dos glifos (Figura 119). O primeiro menu *drop-down, Skeleton,* possui um conjunto de esqueletos tipográficos pré-calculados que o utilizador pode escolher. O menu *drop-down* seguinte, *Drawing Technique*, contém o conjunto das técnicas de desenho desenvolvidas, bem como os parâmetros que são inerentes a cada uma. Estes parâmetros são apresentados como *input sliders* que são referentes às características dos glifos que o utilizador pode variar no processo de criação dos glifos.

Com o propósito de ajudar o utilizador na exploração das variações que os glifos podem tomar, decidimos implementar três funcionalidades: (i) botão *Random Values*, que permite a possibilidade de definir os valores de todos os parâmetros de forma aleatória, dentro da respetiva escala, com o intuito de obter resultados visuais surpreen-

dentes e inesperados ; (ii) botão *Default Values,* que permite a possibilidade de restaurar os valores de todos os parâmetros aos seus valores predefinidos; e por último, (iii) o botão de *Audio Input On*, que permite a possibilidade de receber o som do microfone e mapear os valores do som recebido para os valores dos parâmetros, abrindo espaço para uma imensa variedade de formas para a criação dos glifos.

Assim sendo, o utilizador tem ao seu dispor um conjunto de funcionalidades que lhe permitem explorar sem medo de cometer erros, as diversas variações que os glifos podem tomar.

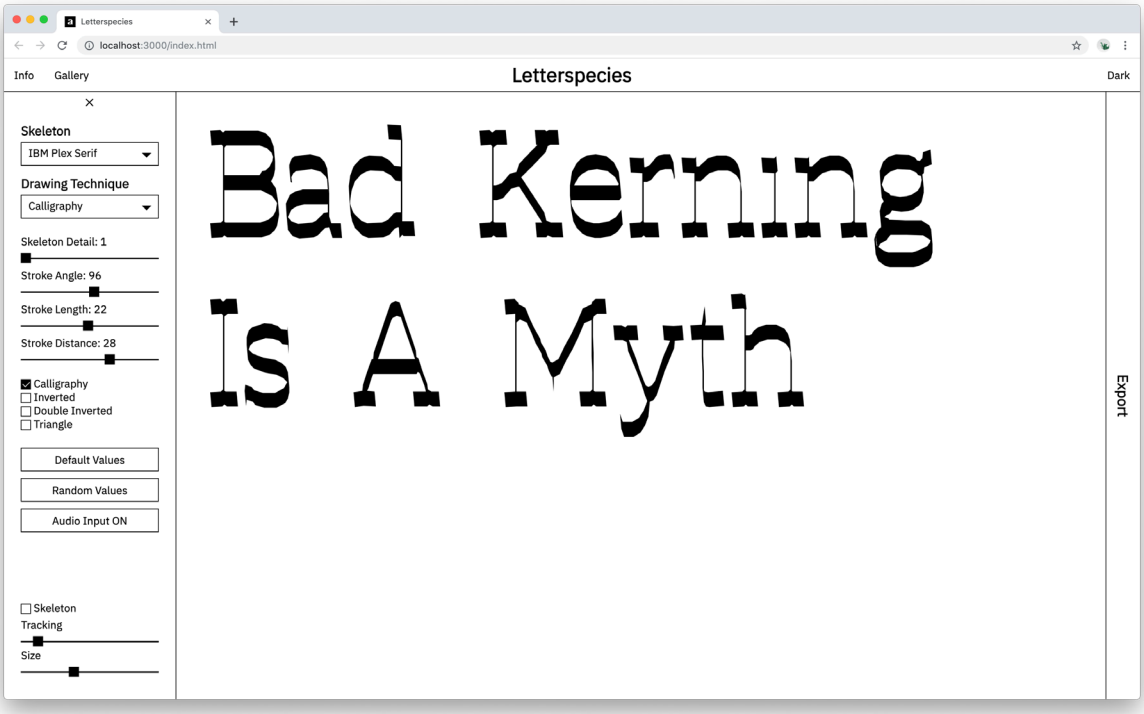


Figura 119
Interface do sistema *web* com painel de configuração aberto.

Área de pré-visualização

A área de pré-visualização surgiu com a necessidade de fornecer ao utilizador *feedback* das alterações que realiza nos parâmetros de configuração dos glifos (Figura 120). Deste modo, o seu objetivo é proporcionar uma interação fluida através da geração e renderização em tempo real dos glifos com os valores definidos pelo utilizador. Através desta interação o utilizador tem o *feedback* visual sobre o impacto que cada parâmetro tem no aspeto visual do glifo e ganha conhecimento suficiente para modificar os valores até encontrar o resultado que mais lhe agrade. Além disto, também é dada a possibilidade ao utilizador de personalizar o texto que está a ser renderizado. Este texto, é ajustável através do seu tamanho e do *tracking*. A área de pré-visualização permite ainda visualizar o esqueleto tipográfico escolhido, possibilitando ao utilizador visualizar a relação que existem entre o esqueleto tipográfico e a técnica de desenho aplicada.

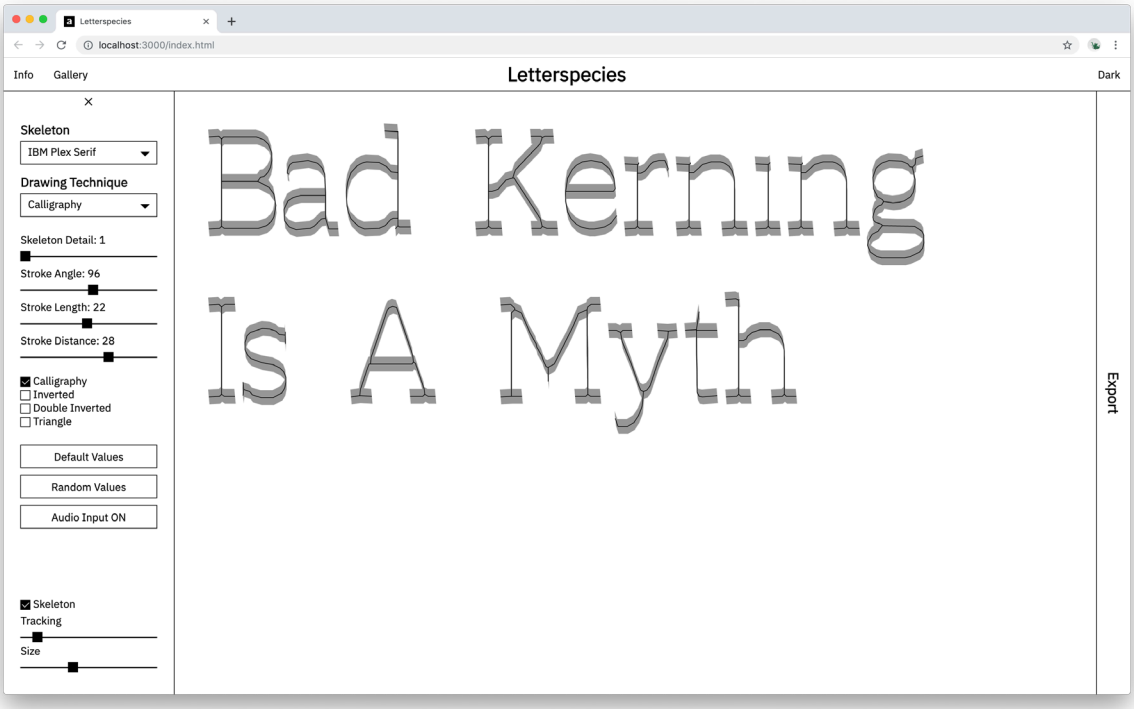


Figura 120
Interface do sistema *web* com opção de mostrar esqueleto tipográfico ativa.

Painel de Exportação

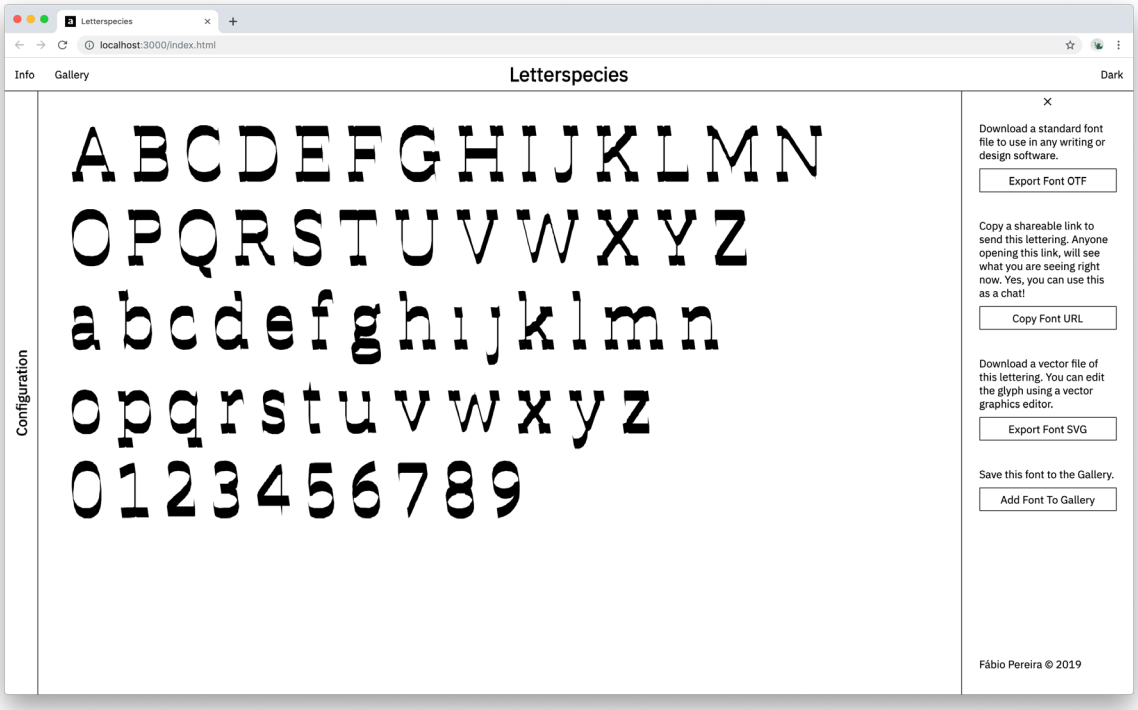
No painel de exportação, é dado ao utilizador todas as opções para exportar a fonte que criou. Para este efeito, é possível exportar a fonte no formato OTF, partilhar a fonte via URL, exportar o texto renderizado na área de pré-visualização em formato SVG e ainda é possível adicionar a fonte a uma galeria (Figura 121). Para exportar a fonte em formato OTF, o sistema utiliza SVG com a integração da biblioteca Opentype.js.

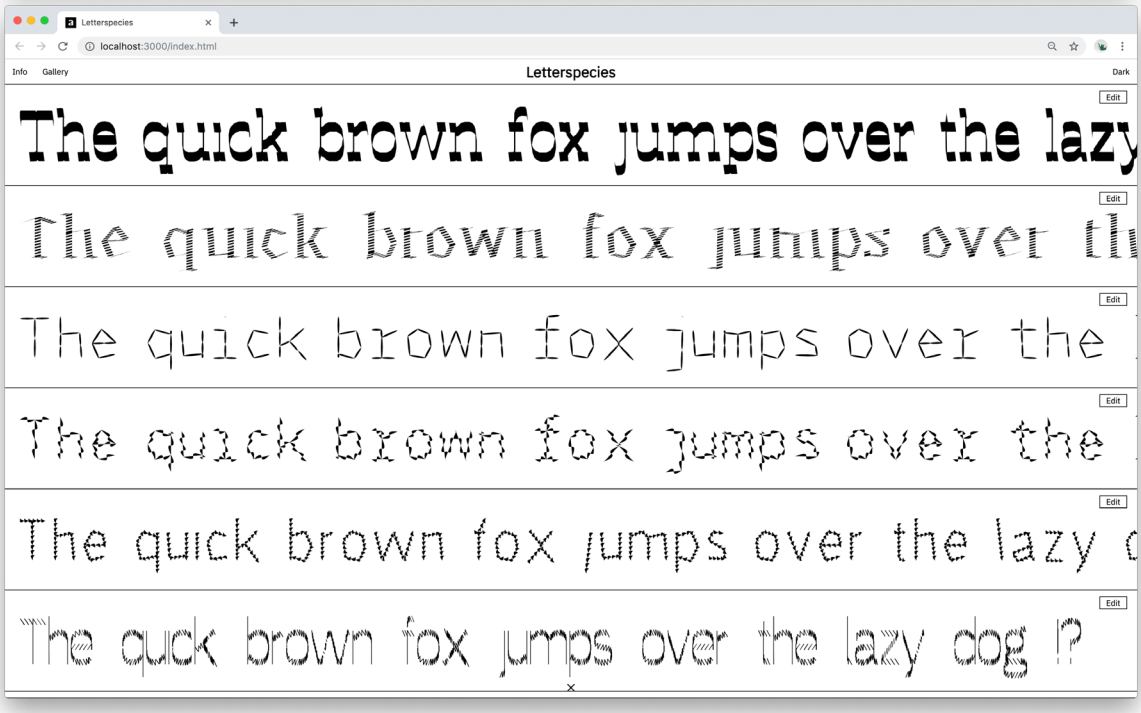
Realizámos uma pesquisa para definir qual a licença de distribuição para as fontes geradas pelo sistema *web* desenvolvido. Tendo em conta que os esqueletos tipográficos são exportados apenas de fontes *open source*, optámos por definir que as fontes geradas fossem distribuídas também segundo a mesma licença. Acreditamos que a atribuição desta licença às fontes valoriza os resultados do sistema, pois permite que estes sejam partilhadas e refinadas pelos seus utilizadores finais. Deste modo, as fontes geradas pelo sistema são *open source*, sob a licença *SIL Open Font Licence*, pelo que, o utilizador pode usar em qualquer tipo de projeto, comercial ou não, e instalar em qualquer sistema operativo.

A opção de copiar o estado atual da fonte via URL, proporciona ao utilizador a possibilidade de partilhar o seu resultado e por sua vez facilita a disseminação de fontes geradas no sistema *web*. O ficheiro SVG exportado do texto na área de pré-visualização pode ser usado em qualquer tipo projeto ou arte, manipulando-o através de qualquer programa para edição de vetores.

Por último, e não menos importante, a funcionalidade de adicionar a fonte gerada à galeria do sistema *web*, possibilita criar uma vasta "biblioteca" das fontes geradas pelos utilizadores, que por sua vez demonstra a diversidade das fontes (Figura 122). Cada fonte exibida na galeria, contém um botão que redireciona para uma nova página do sistema *web* com a configuração da fonte.

Figura 121 e 122
Em cima,interface do sistema *web* com painel de exportação aberto. Em baixo, interface do sistema *web* com galeria aberta.





5.3. Testes e Aplicações

Tendo sido finalizado o desenvolvimento do sistema *web*, era necessário testá-lo com utilizadores de forma a recolher *feedback* relativo à interação com o sistema e em relação às fontes geradas, com o intuito de aprimorar o sistema *web* e/ou corrigir possíveis erros.

Após a fase de testes, foi também necessário idealizar possíveis situações em que as fontes geradas no sistema *web* pudessem ser aplicadas. Desta modo, nesta seção é descrita a fase de testes assim como uma análise feita dos mesmos, e ainda são apresentadas possíveis aplicações das fontes geradas.

Testes com utilizadores

Antes de dar início à fase de testes preparámos uma versão estável do sistema *web*, na qual estavam todas as funcionalidades que pretendíamos testar de forma a obter *feedback* fidedigno por parte dos utilizadores. De seguida, definimos os objetivos que pretendíamos alcançar através da resposta a um questionário e de possíveis comentários por parte dos utilizadores. Assim sendo, os testes foram realizados tendo em conta os seguintes objetivos: (i) compreensão do funcionamento do sistema *web* por parte do utilizador; (ii) interface visualmente apelativa; (iii) fácil uso do sistema *web*; (iv) observação de diversidade visual das fontes geradas; e por fim, (v) qualidade das fontes geradas. Após a definição dos objetivos, estruturámos uma lista de tarefas que o utilizador teria de cumprir. As tarefas presentes na lista foram definidas numa ordem específica para que o utilizador pudesse interagir com as funcionalidades do sistema *web* que estávamos a avaliar, mas também para garantir que estivesse preparado para responder ao questionário. A lista de tarefas foi também pensada de forma a não ser necessário muito tempo para o utilizador conseguir completar todas as tarefas. A lista de tarefas é apresentada na Tabela 1.

Tabela 1
Tarefas definidas para os testes.

Nº Tarefa	Descrição da Tarefa
T1	Encontre e leia a informação sobre o sistema.
T2	Oculte a informação sobre o sistema.
T3	Altere o tema de cores e mantenha o que for mais agradável.
T4	Encontre as configurações da fonte.
T5	Escolha um esqueleto tipográfico até encontrar um que goste.
T6	Escolha uma técnica de desenho até encontrar uma que goste.
T7	Modifique os parâmetros relativos à técnica de desenho escolhida até que o aspeto visual da fonte seja do seu agrado (pode utilizar as funcionalidades *random values* e *default values*).
T8	Modifique o texto que está a ser visualizado.
T9	Visualize as linhas do esqueleto tipográfico sobre as letras.
T10	Altere o *tracking* do *lettering*.
T11	Altere o tamanho do *lettering*.
T12	Exporte a fonte para ficheiro OTF.

Assim sendo, realizámos os testes de forma presencial com 15 utilizadores, certificando sempre que todos realizavam o teste nas mesmas condições. Da amostra dos 15 utilizadores, 12 são estudantes e professores de Design e Multimédia, 1 estudante de Direito, 1 estudante de Bioquímica, e 1 professor de Informática.

Em cada teste, fizemos uma breve explicação de sua finalidade e pedimos a cada utilizador que lesse a lista de tarefas que tinha de completar. No final do teste, pedimos a cada utilizador que preenchesse o questionário relativo às tarefas que tinha completado. Este questionário era finalizado com sugestões/comentários do utilizador relativamente ao sistema web. O questionário e respetivas respostas são apresentadas no Anexo A. As respostas numéricas são compreendidas num intervalo de 1 a 5, em que 1 corresponde a "discordo plenamente" e 5 corresponde a "concordo plenamente".

E ainda, existem duas questões de resposta binária, "sim" ou "não", no caso da resposta ser "não", é questionada a razão para essa escolha (Tabela 3). De todas as respostas, apenas houve um "não", no qual a razão foi «Não percebi que era para mudar o tema de cores, pensava que era para mudar a cor do tipo de letra», pelo que, associamos esta resposta à forma como a tarefa T3 tinha sido definida e não ao funcionamento do sistema.

A Tabela 2 apresenta a lista de questões, de resposta numérica, juntamente com a média e a mediana associadas a cada resposta. Calculamos estas medidas de forma a poder analisar a tendência dos resultados e quão enviesados são os dados.

Tabela 2
Questões e correspondentes média e mediana relativas às respostas.

Nº Questão	Questão	Média	Mediana
Q1	Entendi o funcionamento do sistema.	4.8	5
Q2	Entendi o que é um esqueleto tipográfico.	5	5
Q3	Entendi o modo como é dada forma ao esqueleto tipográfico.	4.46	5
Q4	Gostei do aspeto visual da interface.	4.8	5
Q5	Achei a interface fácil de utilizar.	4.73	5
Q6	Achei que os letterings gerados são visualmente diversificados.	4.86	5
Q7	Achei que a variação do esqueleto tipográfico (mantendo a técnica de desenho) resulta em letterings visualmente diversificados.	4.26	4
Q8	Achei que a variação da técnica de desenho (mantendo o esqueleto tipográfico) resulta em letterings visualmente diversificados.	4.66	5
Q9	Gostei do aspeto visual dos letterings gerados.	4.8	5
Q10	Foi fácil ler o texto no lettering.	4	4

Questão	Sim	Não
Entendi o resultado/impacto de cada funcionalidade.	14	1
Utilizaria uma fonte gerada pelo sistema num trabalho.	15	0

Tabela 3
Questões de resposta binária

Analisando os resultados, podemos dizer que os objetivos definidos antes dos testes foram alcançados. No que diz respeito ao sistema, a maioria dos utilizadores achou fácil de usar e visualmente atrativo. Relativamente à diversidade dos glifos gerados, os utilizadores responderam à questão Q6, com uma média de 4,86, confirmando a diversidade visual existente nos glifos/fontes gerados.

No entanto, relativamente à *legibility,* os utilizadores encontráram algumas dificuldades na leitura dos *letterings* com os glifos gerados. Isto deve-se ao facto de que algumas configurações dos parâmetros das técnicas de desenho podem gerar glifos com formas bastante incomuns para os glifos, o que dificulta o reconhecimento, por parte do utilizador, desses mesmos glifos.

O *feedback* que recebemos da seção de comentários/sugestões ajudou a melhorar o sistema *web*, sendo que alguns erros foram encontrados pelos utilizadores e posteriormente corrigidos. Por exemplo, a maioria dos comentários estava relacionada com os menus *drop-down* para escolher o esqueleto tipográfico e a técnica de desenho. A razão para isto, deve-se ao facto de quando os menus eram abertos, não havia *feedback* visual indicativo da opção selecionada, e também, quando era escolhida uma opção o menu fechava-se automaticamente, causando alguma dificuldade e frustração no processo de exploração. Outros comentários eram relacionados com: (i) a inconsistência de cores no menu de informação em comparação com o aspeto visual geral; (ii) algumas imperfeições nas técnicas de desenho, relacionadas com os traços horizontais e verticais; e por fim, (iii) a área de pré-visualização deveria permitir escrever texto em várias linhas.

De todos os comentários/sugestões feitos pelos utilizadores, identificámos os mais relevantes de forma a serem implementados no sistema *web* com o intuito de melhorá-lo. Tendo isto em conta, numa observação geral, percebemos que a maioria dos utilizadores estavam entusiasmados ao interagir com o sistema *web* e satisfeitos com as fontes geradas.

Aplicação de resultados

Sendo um dos principais objetivos do sistema web, auxiliar a prática dos designers gráficos e de tipos de letra — público alvo —, fazia sentido explorar possíveis cenários em que estes possam utilizar a ferramenta para esse propósito. Deste modo, nesta secção, são apresentadas possíveis aplicações para fontes geradas pelo sistema. No Anexo B, encontra-se um vasto leque de fontes previamente geradas.

Para realizar as aplicações de fontes geradas em composição de

texto, foi utilizado o sistema *Scripted Pages* desenvolvido por Diogo Ferreira, em 2019, no contexto de dissertação de mestrado em Design e Multimédia, na Universidade de Coimbra. Este sistema consiste na geração algorítmica de *layouts* de publicações impressas. Em suma, o sistema permite importar um ficheiro *Microsoft Word,* com conteúdo textual e/ou imagético, no programa *Adobe InDesign*, e de seguida gera um *layout* para esse conteúdo. A composição do conteúdo é feita automaticamente seguindo um conjunto de regras da macro e microtipografia. O uso deste sistema facilitou o processo de experimentação na aplicação das fontes geradas.

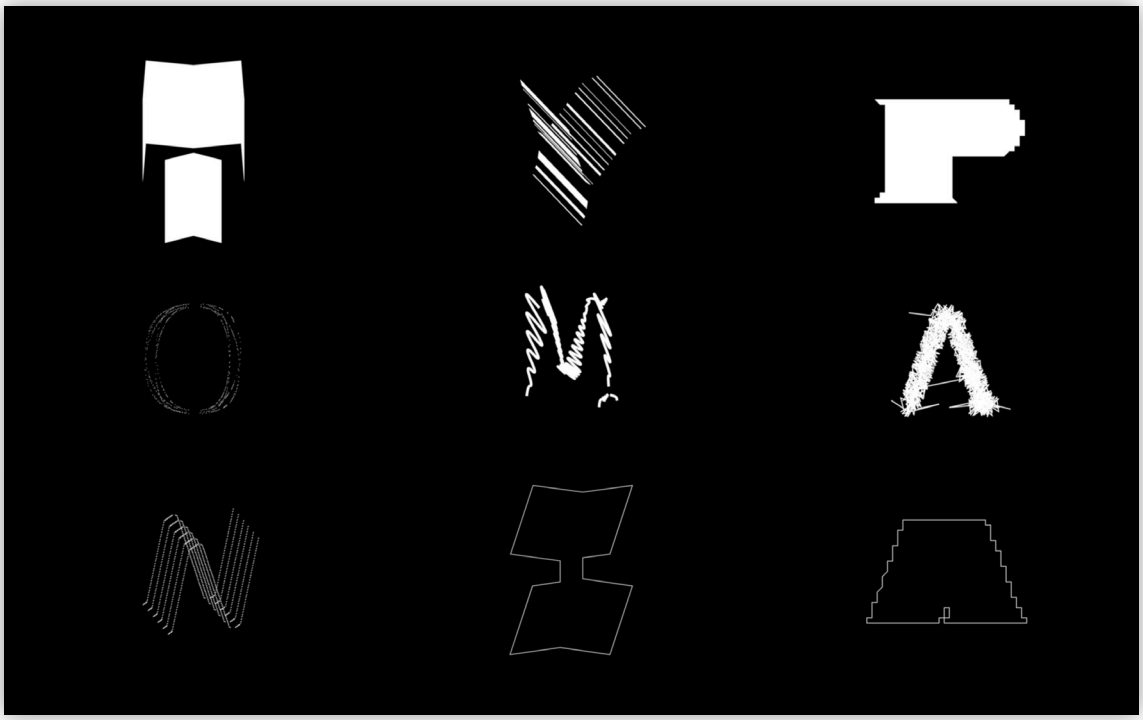
Analisando os conteúdos produzidos, é possível afirmar que existem fontes mais apropriadas para determinadas finalidades do que outras. Por exemplo, uma fonte gerada com a técnica de desenho *Calligraphy* (Figura 135)*,* é mais adequada para compor texto corrido, do que uma fonte gerada pela técnica de desenho *Lines* (Figura 129)*.* Isto deve-se às características anatómicas e visuais da fonte serem muito distintas. Comparando ambas as fontes, é possível verificar que, a fonte gerada com a técnica de desenho *Calligraphy* tem maior *legibility* que a fonte gerada com a técnica de desenho *Lines,* tornando-a menos cansativa para leitura de textos longos. Em oposição, a fonte gerada com a técnica de desenho *Lines* tem maior impacto quando aplicada na composição de um cartaz, pois chama mais facilmente a atenção.

Nas Figuras 123 e 124, são visíveis alguns *frames* de um vídeo produzido para o concurso de vídeo tipográfico *Typomania* 2019 (en.typomania.ru)*.* Este vídeo demonstra a capacidade do sistema apresentado em gerar glifos que reagem visualmente em tempo real a som. É possível aceder ao vídeo pelo seguinte URL: cdv.dei.uc.pt/2019/letterspecies/letterforms-sound-reactive.mov.

5.4 Disseminação

Na fase de desenvolvimento das técincas de desenho, surgiu a oportunidade de participar no projeto tipográfico *36 Days Of Type* (36daysoftype.com)*,* que consiste em criar uma interpretação individual das letras e dígitos presentes no alfabeto romano, e partilhar através do *Instagram* utilizando a *hashtag* "#36daysoftype"*.* Deste modo, foi criada uma página de *Instagram* (@letterspecies)*,* com o intuito de publicar as interpretações criadas utilizando o sistema apresentado. A participação neste projeto, ajudou no processo de exploração de técnicas de desenho bem como variações das mesmas.

Durante o desenvolvimento do projeto prático, foram escritos e submetidos dois artigos científicos para as conferências *Artech 2019* (2019.artech-international.org) e *10º Encontro de Tipografia* (10et.esad.pt)*.* Estes artigos retratam fases distintas deste trabalho. A quando da escrita deste texto, sabemos que o artigo submetido na conferência *Artech 2019* foi aceite e será apresentado em outubro de 2019. O artigo para o *10º Encontro de Tipografia* ainda se encontra na fase de avaliação.



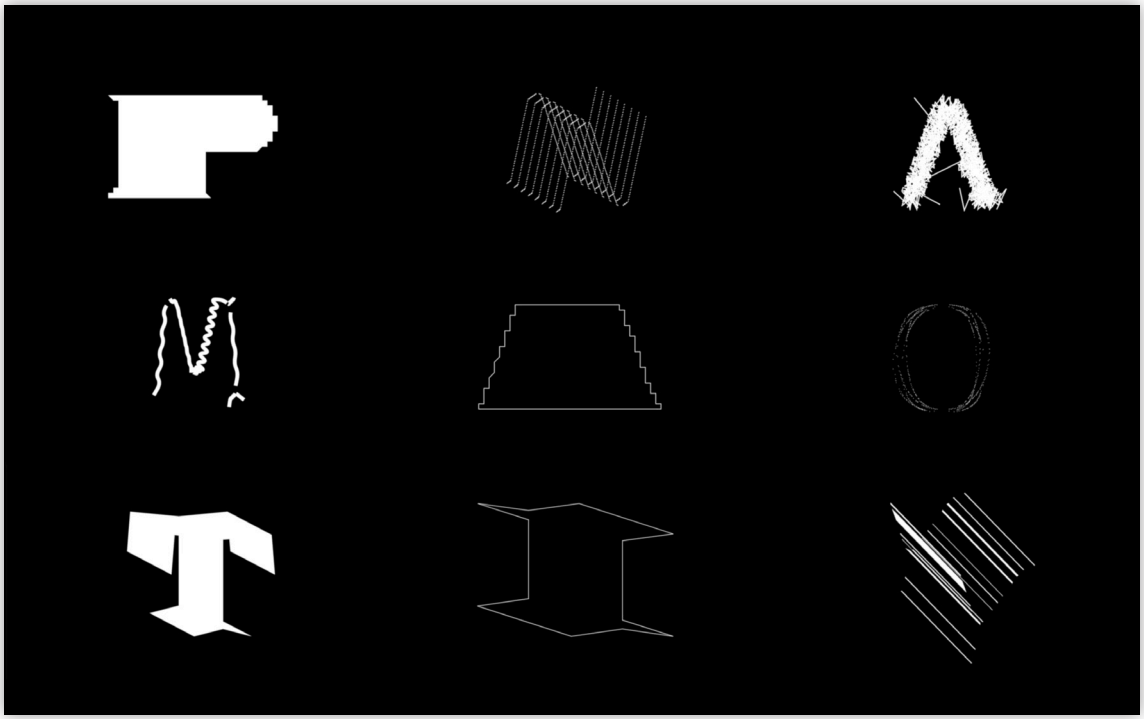


Figura 123 e 124
Frames de vídeo produzido para *Typomania* 2019.

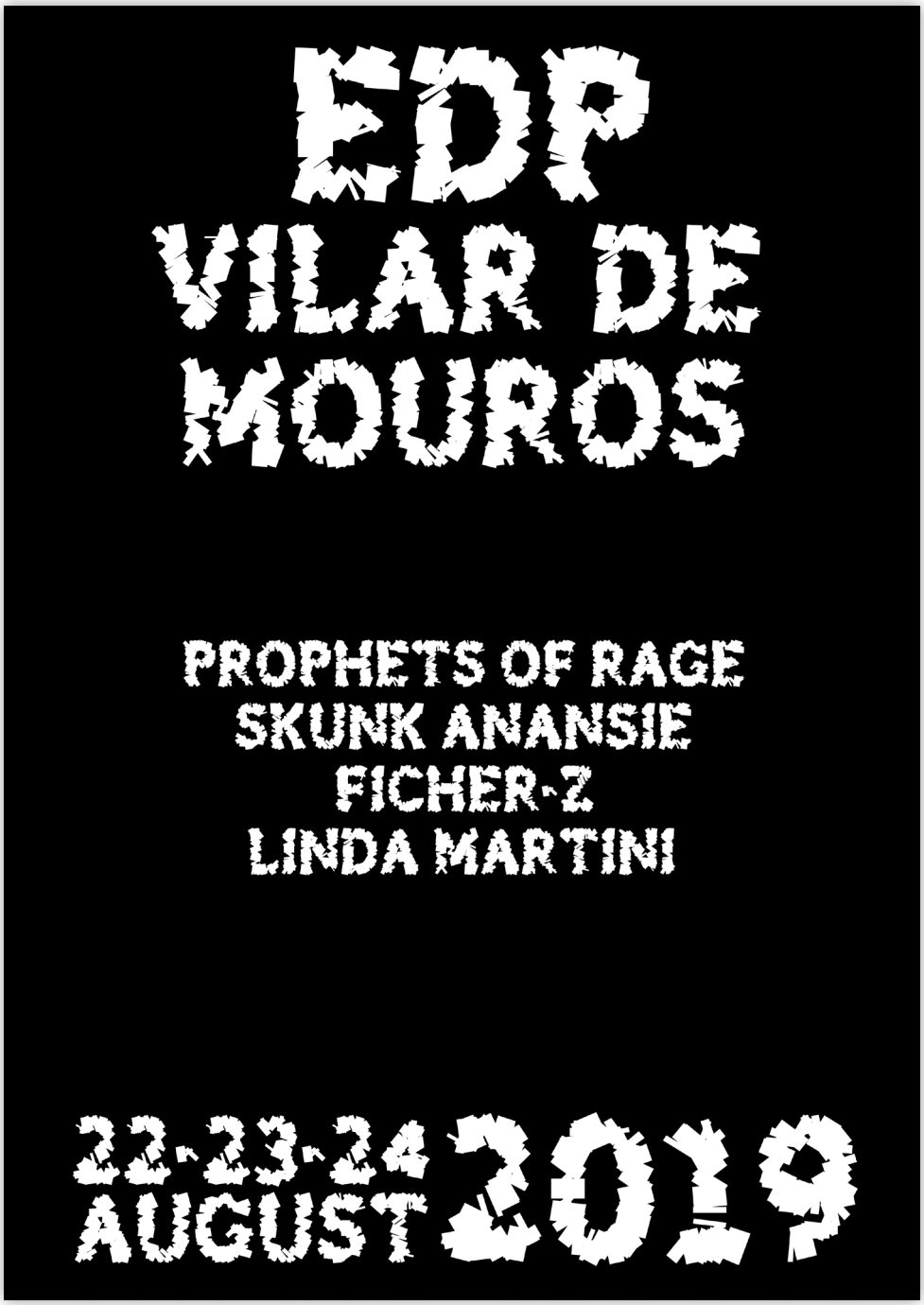


Figura 125
Mockup com possível aplicação de uma fonte gerada pelo sistema em cartaz.

Figura 126
Mockup com possível aplicação de uma fonte gerada pelo sistema em cartaz.

Figura 127
Mockup com possível aplicação de uma fonte gerada pelo sistema em cartaz.

Figura 128
Mockup com possível aplicação de uma fonte gerada pelo sistema em cartaz.

Figura 129
Mockup com possível aplicação de uma fonte gerada pelo sistema em cartaz.

Figura 130
Mockup com possível aplicação de uma fonte gerada pelo sistema em cartaz.

Figura 131
Mockup com possível aplicação de uma fonte gerada pelo sistema em cartaz.

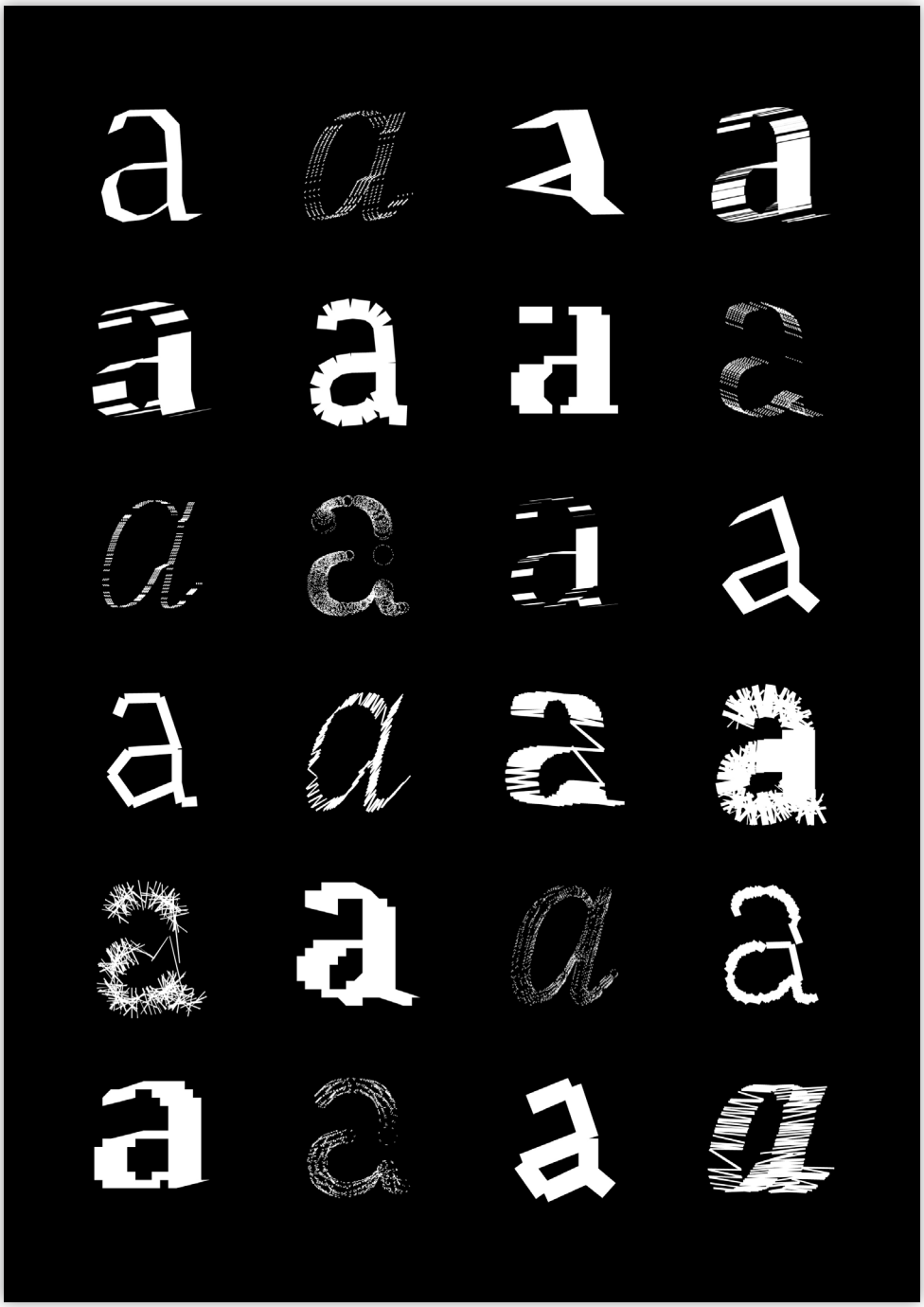


Figura 132
Mockup com possível aplicação de uma fonte gerada pelo sistema em cartaz.

Figura 133
Mockup com possível aplicação de uma fonte gerada pelo sistema em composição de texto.

The cycle changes—conceive an idea, the computer generates form alternatives. We evaluate and select. The seams are more apparent. Time is abbreviated. The realm of possibility expands. Imagine we have any typeface at our disposal on an interactive computer system. We can use simple directions to effect change by means of asking for certain operations to be performed on the type; we can fragment, distort, incline, change weight, etc. Consider all the permutations of a logotype we can easily run through. We are limited only by our imagination. We conceive of a visual game — the computer helps us play it out in great detail. Will we find new, more elegant form? It is too soon to tell.

SHARON POGGENPOHL
"Design Research, Building a Culture from Scratch", 2010

Figura 134
Mockup com possível aplicação de uma fonte gerada pelo sistema em composição de texto.

CONTOS 144

III

Então começaram, para nossos Pais, os dias abomináveis do Paraíso

O seu constante e desesperado esfôrço foi sobreviver—no meio duma Natureza que, sem cessar e furiosamente, tramava a sua destruição E Adão e Eva passaram êsses tempos, que os poemas Semíticos celebram como Inefáveis—sempre a tremer, sempre a ganir, sempre a fugir! A terra ainda não era uma obra perfeita e a Divina Energia, que a andava compondo, incessantemente a emendava, numa tam móbil inspiração, que em sítio coberto ao alvorecer por uma floresta, à noite se espelhava uma lagôa onde a Lua, já doente, vinha estudar a sua palidez Quantas vezes nossos Pais, repousando no pendor de um outeiro inocente, entre o serpol e o rosmaninho (Adão com a face deitada sôbre a côxa de Eva, Eva com dedos ágeis catando o pêlo de Adão) foram sacudidos pela encosta amena como por um dorso irritado, e rolaram, embrulhados, entre o ribombo, e a labareda, e a fumarada, e a cinza quente do vulcão que Jeová improvisara! Quantas noites escaparam, uivando, dalguma abrigada caverna, quando já sôbre ela corria um grande mar inchado que bramava, se desenrolava, ficava fervendo entre as rochas, com negras focas mortas a boiar Ou então era o chão, o chão seguro, já social e fertilizado para as searas socráveis, que de repente rugia como uma fera, escancarava uma insondável goela, e tragava rebanhos, prados, nascentes, benéficos cedros com todas as rôlas que na sua rama arrulhavam

Depois eram as chuvas, as longas chuvas Edénicas, desabando em jorros clamorosos, durante alagados dias, durante torrentosas noites, tam desabalada-

Figura 135
Mockup com possível aplicação de uma fonte gerada pelo sistema em composição de texto.

mente que do Paraíso, vasto charco barrento, apenas apareciam as pontas do arvoredo afogado, e os cimos dos montes atulhados de bichos transidos que bramiam no terror das águas soltas E nossos Pais, refugiados nalguma erguida fraga, gemiam lamentavelmente, com regatos a escorrer dos ombros, com ribeiras a escorrer dos pés, como se o barro novo de que Jeová os fizera se andasse já desfazendo

E mais terríficas eram as estiagens Oh! o incomparável tormento das sêcas no Paraíso! Lentos dias tristes, após lentos dias tristes, a imensa brasa do sol candente coriscava furiosamente num céu côr de cobre, em que o ar baço e grosso crepitava e arfava Os montes estalavam, gretados e as planícies desapareciam sob uma denegrida camada de fios retorcidos, ennovelados, rijos como arames, que eram os restos das verdes pastagens Toda a tisnada folhagem rolava nos ventos abrasados, com rugidora restolhada O leito dos rios chupados tinha a rigidez de ferro fundido O musgo escorregava das rochas, como uma pele sêca que se despega descobrindo largos ossos Cada noite um bosque ardia, fogueira estralejante, de lenha ressequida, escaldando mais a abóbada do forno inclemente Todo o Éden andava coberto das revoadas de abutres e corvos, porque, com tanto animal morto de fome e de sêde, abundava a carne pôdre No rio, a água que restava mal corria, empoçada pela massa fervilhante de cobras, rãs, lontras, tartarugas, refugiadas naquele derradeiro veio, lodoso e todo morno E nossos Pais veneráveis, com as magras costelas a arquejar contra o pêlo crestado, a língua pendida e mais dura que cortiça, erravam de fonte em fonte, a sorver desesperadamente alguma gota que ainda brotasse, gota rara, que assobiava, ao caír, sobre as lages esbraseadas

E assim Adão e Eva, fugindo do Fogo, fugindo da

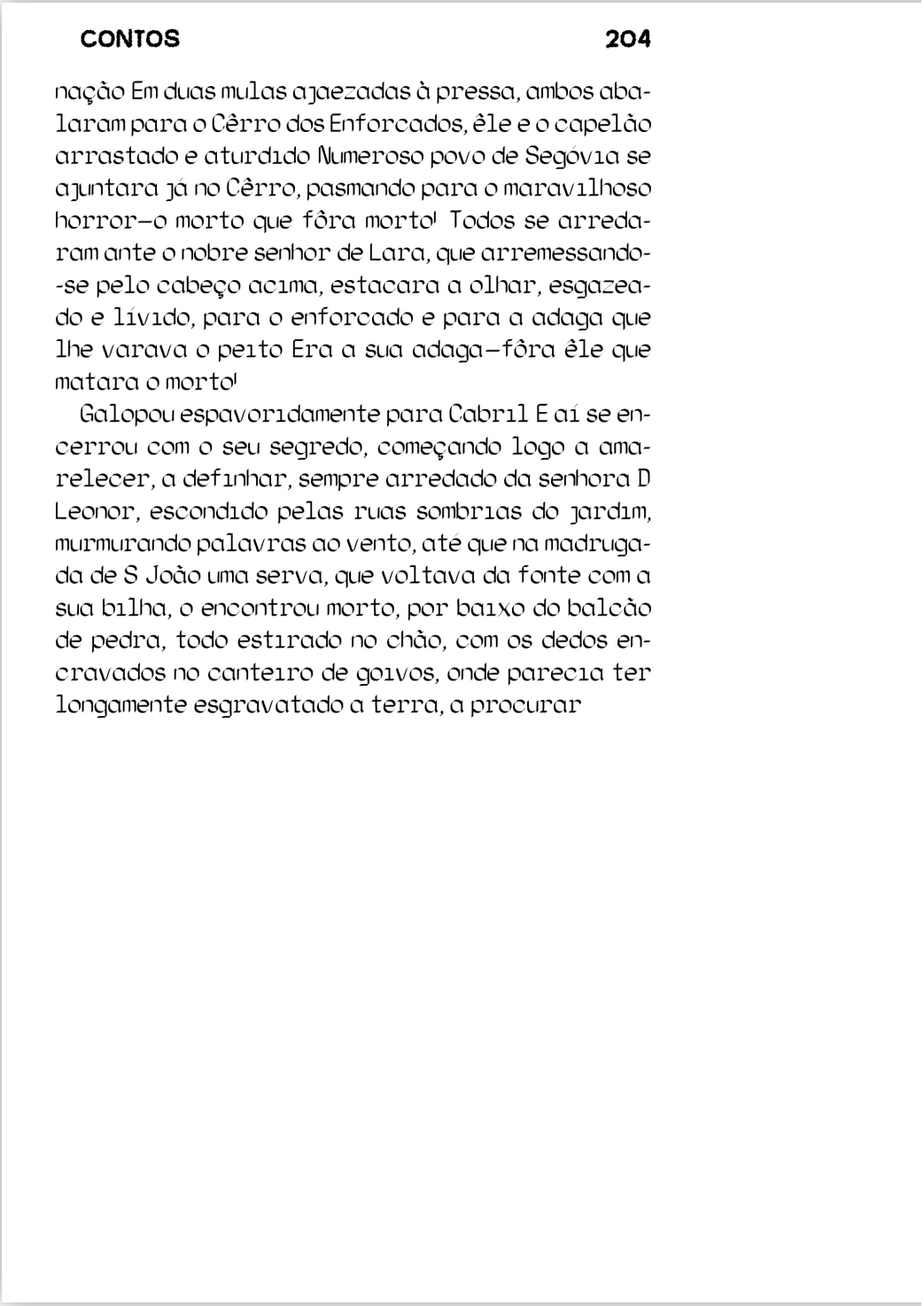


Figura 136
Mockup com possível aplicação de uma fonte gerada pelo sistema em composição de texto.

V

Para fugir a tam lamentáveis memórias, a senhora D Leonor, herdeira de todos os bens da casa de Lara, recolheu ao seu palácio de Segóvia Mas como agora sabia que o senhor D Rui de Cardenas escapara miraculosamente à emboscada de Cabril, e como cada manhã, espreitando de entre as gelosias, meio cerradas, o seguia, com olhos que se não fartavam e se humedeciam, quando êle cruzava o adro para entrar na igreja, não quis ela, com receio das pressas e impaciências do seu coração, visitar a Senhora do Pilar emquanto durasse o seu luto Depois, uma manhã de domingo, quando, em vez de crepes negros, se poude cobrir de sêdas roxas, desceu a escadaria do seu palácio, pálida de uma emoção nova e divina, pisou as lages do adro, transpôs as portas da igreja D Rui de Cardenas estava ajoelhado diante do altar, onde depusera o seu ramo votivo de cravos amarelos e brancos Ao rumor das sêdas finas, ergueu os olhos com uma esperança muito pura e toda feita de graça celeste, como se um anjo o chamasse D Leonor ajoelhou, com o peito a arfar, tam pálida e tam feliz que a cera das tochas não era mais pálida, nem mais felizes as andorinhas que batiam as asas livres pelas ogivas da vélha igreja

Ante êsse altar, e de joelhos nessas lages, foram eles casados pelo bispo de Segóvia, D Martinho, no outono do ano da Graça de 1475, sendo já reis de Castela Isabel e Fernando, muito fortes e muito católicos, por quem Deus operou grandes feitos sôbre a terra e sôbre o mar

6 — Conclusão

O trabalho apresentado teve início com uma investigação e análise bibliográfica no contexto da tipografia e de processos algorítmicos e generativos aplicados ao design de tipos. Esta investigação foi importante para ganhar os conhecimentos necessários para aplicar no desenvolvimento do projeto prático. Neste sentido, foi relevante o estudo da anatomia da letra, para ter ideia das possibilidades de variações que podiam ser exploradas, assim como aprofundar os conhecimentos em diferentes aspectos da tipografia, como a *legibility* e o *kerning*.

A pesquisa sobre a classificação tipográfica foi igualmente relevante para entender a evolução dos paradigmas da tipografia desde a sua origem até à prática contemporânea. Procurámos ainda perceber a influência das novas tecnologias na área do design gráfico e da tipografia e de que forma vieram revolucionar o processo criativo na prática do design. Concluímos esta investigação com uma análise de trabalhos realizados no campo do design de tipos através de técnicas manuais e generativas, de modo a entender o que já tinha sido feito e de que forma poderia influenciar a nossa abordagem para o projeto prático. Após o estudo bibliográfico, iniciámos a investigação prática no desenho generativo de tipos de letra. Foi desenvolvido um sistema computacional *web* capaz de gerar glifos a partir de esqueletos tipográficos. Estes glifos são o resultado do preenchimento de um esqueleto tipográfico por uma das várias técnicas de desenho existentes. Os esqueletos tipográficos são extraídos de diferentes fontes *open-source*. O sistema *web* possui uma interface gráfica que fornece ao utilizador todas as funcionalidades necessárias para a configuração, pré-visualização e exportação da fonte criada por ele.

Ao permitir que o utilizador ajuste os parâmetros referentes à geração dos glifos, o sistema *web* possibilita e encoraja a experimentação no âmbito de design gráfico e design de tipos. O utilizador é capaz de gerar glifos incomuns que ampliam os limites existentes entre *legibility* e expressividade. Consideramos que o sistema tem potencial para servir de ferramenta auxiliar em diferentes tipos de projetos de design gráfico. Possibilita também o auxílio na produção de fontes personalizadas que fornece um meio para expressar um conceito ou representar dados. Por exemplo, através do mapeamento dos parâmetros das técnicas de desenho para dados externos, como som ou temperatura, o sistema *web* é capaz de gerar fontes que se adaptam dinamicamente nos contextos que são aplicadas. Para além disto, esta funcionalidade proporciona a criação de identidades dinâmicas, uma prática crescente no contexto do design gráfico. O sistema *web* tem ainda potencial no âmbito de design *on-demand,* tendo em conta que permite a criação de fontes exclusivas, caracterizadas por exemplo pela aleatoriedade. A capacidade adaptativa dos glifos gerados comprova a capacidade do sistema *web* desenvolvido de gerar fontes reativas, atendendo às necessidades e oportunidades criadas pelos novos *media*.

O principal objetivo desta dissertação consistia em desenvolver

um sistema algorítmico *web* capaz de gerar glifos a partir de esqueletos extraídos de tipos de letra existentes. Uma vez finalizado o projeto e considerando para o sistema que foi desenvolvido, podemos afirmar que este responde ao objetivo referido. Era também um objetivo tornar este sistema *web* numa ferramenta para o auxílio na prática de design gráfico e design de tipos. Tendo em conta o *feedback* relativo às aplicações do sistema *web* recebido por parte dos utilizadores na fase de testes, podemos afirmar que o sistema *web* apresenta aplicabilidade em diversas áreas do design gráfico, por exemplo, identidades visuais, cartazes, composição de texto e produção de animações tipográficas.

Dado o limite temporal existente para a realização da dissertação, foi necessário cessar o desenvolvimento do sistema *web* numa versão estável, completa com todas as funcionalidades propostas. No entanto, existe espaço para desenvolver novas funcionalidades e minimizar possíveis limitações. Desta forma, o trabalho futuro passará por: (i) desenvolver mais técnicas de desenho e corrigir possíveis imperfeições nas existentes; (ii) implementar um processo de cálculo automático para os pares de *kerning*; (iii) melhorar o processo de composição do texto na área de pré-visualização; (iv) testar o sistema *web* para a geração de glifos de caracteres *Non-Latin*; (v) criar a funcionalidade para exportar o texto que é pré-visualizado nos formatos JPEG e PNG; (vi) integrar a tecnologia de fontes variáveis no sistema de forma a tornar possível o controlo dos parâmetros referentes às técnicas de desenho em *software* que suporta esta tecnologia; e por último, (vii) criar um modo de apresentação no qual é feita uma demonstração das fontes geradas pelo sistema.

7 — Bibliografia

Aghabayan, T. (n.d.). Gestalten mit Code, FH Mainz. Retrieved January 18, 2019, from http://generative-typografie.de/generativetypografie/elien/

Armstrong, H. (2016). *Digital Design Theory: Readings From The Field*. Princeton Architectural Press.

Balzien, A. (n.d.). Gestalten mit Code, FH Mainz. Retrieved January 18, 2019, from http://generative-typografie.de/generativetypografie/carve-lcd/

Brath, R. (2017). Variable Fonts vs. Parametric Fonts and Data Visualization. Retrieved January 17, 2019, from https://richardbrath.wordpress.com/2017/08/26/variable-fonts-vs-parametric-fonts-and-data-visualization/

Bringhurst, R. (2008). *Elementos do Estilo Tipográfico*. COSAC NAIFY.

Chen, A. (n.d.). Galapagos Evolutionary Type Design. Retrieved January 18, 2019, from http://annhchen.com/filter/design/Galapagos

Costa, C. (2013). *Organizador de tipos de letra*. Universidade de Coimbra.

Dixon, C. (2002). *Typeface classification*. London.

Dubberly, H. (2004). *How do you design? A compendium of models*.

Flückiger, M., & Kunz, N. (n.d.). LAIKA – a dynamic typeface. Retrieved January 18, 2019, from http://www.laikafont.ch/

Flückiger, M., & Kunz, N. (2009). LAIKA. Retrieved January 18, 2019, from https://vimeo.com/6993808

Fontshop. (n.d.). FF Beowolf. Retrieved January 17, 2019, from https://www.fontshop.com/families/ff-beowolf

Hanzer, E. (n.d.). Phase. Retrieved January 15, 2019, from https://www.eliashanzer.com/phase/

Holland-Cunz, N. (n.d.). Gestalten mit Code, FH Mainz. Retrieved January 18, 2019, from http://generative-typografie.de/generativetypografie/broken-grid/

Huang, M. (2010). Typeface. Retrieved January 19, 2019, from http://mary-huang.com/portfolio/typeface/

Jung, I.-H. (n.d.). Gestalten mit Code, FH Mainz. Retrieved January 18, 2019, from http://generative-typografie.de/generativetypografie/pde-lubanoise/

Kinross, R. (2004). *Modern Typography: an essay in critical history*. Hyphen Press.

Klein, D. (n.d.). Gestalten mit Code, FH Mainz. Retrieved January 18, 2019, from http://generative-typografie.de/generativetypografie/blast/

Köhler, C. (n.d.). Gestalten mit Code, FH Mainz. Retrieved January 18, 2019, from http://generative-typografie.de/generativetypografie/wdce/

Kupferschrift, I. (2012). Type classifications are useful, but the common ones are not. Retrieved December 9, 2018, from http://kupferschrift.de/cms/2012/03/on-classifications

Levée, J.-B., & Murit, I. (n.d.). New LAB font: Gothic Lab. Retrieved December 9, 2018, from https://www.productiontype.com/news/new_lab_font_gothic_lab

Lupton, E. (2004). *Pensar com tipos: guia para designer, escritores, editores e estudantes*.

Maçãs, C. (2013). *Comportamentos da Tipografia Generativa: Uma Proposta para um Tipo Generativo*. Universidade de Coimbra.

Martin, L. (2018). Font of all knowledge? Researchers develop typeface they say can boost memory. Retrieved January 12, 2019, from https://www.theguardian.com/artanddesign/2018/oct/04/font-of-all-knowledge-researchers-develop-typeface-they-say-can-boost-memory?fbclid=IwAR2GqHXXGiv_yDQm-YNSNQmH-JUMUXSvCL1YOTnqFJgYJzvO5rUg4jgy-DaY

Martins, T., Correia, J. N., Costa, E., & Machado, P. (n.d.). Evotype: Towards the Evolution of Type Stencils. Retrieved January 18, 2019, from https://cdv.dei.uc.pt/evotype/

Marxer, R. (n.d.). Geomerative. Retrieved January 18, 2019, from http://www.ricardmarxer.com/geomerative/

Mathey, Y. (2009). Prototypo. Retrieved September 20, 2001, from https://www.prototypo.io/

McGraw, G., & Rehling, J. (n.d.). The Letter Spirit project. Retrieved January 19, 2019, from http://goosie.cogsci.indiana.edu/farg/mcgrawg/lspirit.html

McNeil, P. (2017). *The Visual History of Type*. London: Laurence King Publishing Ltd.

McNeil, P., & Muir, H. (n.d.-a). Intersect Typefaces. Retrieved January 19, 2019, from http://www.muirmcneil.com/project/intersect-2/?section=about

McNeil, P., & Muir, H. (n.d.-b). MuirMcNeil. Retrieved December 23, 2018, from http://www.muirmcneil.com/about/

McNeil, P., & Muir, H. (n.d.-c). ThreePoint Typefaces. Retrieved January 19, 2019, from http://www.muirmcneil.com/project/threepoint/?section=about

McNeil, P., & Muir, H. (n.d.-d). ThreeSix Typefaces. Retrieved January 19, 2019, from http://www.muirmcneil.com/project/threesix-3/?section=about

McNeil, P., & Muir, H. (n.d.-e). TwoPlus Typefaces. Retrieved January 19, 2019, from http://www.muirmcneil.com/project/twoplus/

McNeil, P., & Muir, H. (n.d.-f). TwoPoint Typefaces. Retrieved January 19, 2019, from http://www.muirmcneil.com/project/twopoint/?section=about

Normals. (n.d.). Normaltype. Retrieved January 19, 2019, from http://normalfutu.re/esthetics-of-variability/normaltypev154/

Onformative. (2011). Growing Data. Retrieved January 19, 2019, from https://www.onformative.com/work/growing-data

Oppenhäuser, S. (n.d.). Gestalten mit Code, FH Mainz. Retrieved January 18, 2019, from http://generative-typografie.de/generativetypografie/fontmix/

Pape, P., & Jenett, F. (n.d.). Gestalten mit Code, FH Mainz. Retrieved January 18, 2019, from http://generative-typografie.de/generativetypografie/about/

Parente, J. (2018). *Desenho Generativo de Tipos de Letra*.

Pawle, F. (2014). Graphic designer David Carson, of Surfer and Ray Gun fame, on 'self-indulgence.' Retrieved January 12, 2019, from https://www.theaustralian.com.au/executive-living/home-design/graphic-designer-david-carson-of-surfer-and-ray-gun-fame-on-selfindulgence/story-fngmet9f-1227079205648

Poggenpohl. (1983). Creativity and Technology. *STA Design Journal*, 14–15.

Rayner, K., Pollatsek, A., Ashby, J., & Clifton, C. J. (2012). *Psychology of reading (2nd ed.).*

Reigel, A., & Müller, M. (2012). Metaflop. Retrieved January 16, 2019, from https://www.metaflop.com/

Reimann, L. (n.d.). Gestalten mit Code, FH Mainz. Retrieved January 18, 2019, from http://generative-typografie.de/generativetypografie/zwirn/

Reinheimer, I. (n.d.). Gestalten mit Code, FH Mainz. Retrieved January 18, 2019, from http://generative-typografie.de/generativetypografie/irratio/

Riechers, A. (2018). What's the Difference Between Variable and Parametric Fonts? Retrieved January 17, 2019, from https://eyeondesign.aiga.org/parametric-and-variable-typeface-systems-shape-shifters-for-letterforms/

Shaughnessy, A. (2009). *Graphic Design: A User's Manual.* London: Laurence King Publishing Ltd.

Silanteva, D. (n.d.). Typographic Music. Retrieved January 19, 2019, from http://www.ddina.com/index.php?/2011/typographic-music/2/

Stahl, K. (n.d.). Gestalten mit Code, FH Mainz. Retrieved January 18, 2019, from http://generative-typografie.de/generativetypografie/pong/

Tools, D. (2007). Post Bitmap Scripter. Retrieved January 18, 2019, from http://www.digital-tools-blog.com/blog/44-post-bitmap-scripter

Tracy, W. (1971). Type Design Classification. *Visible Language*, 59–66.

Tschense, T. (n.d.). Gestalten mit Code, FH Mainz. Retrieved January 18, 2019, from http://generative-typografie.de/generativetypografie/bastard/

TypeThursday. (2016). Learning Python Makes You A Better Designer: An Interview with Just van Rossum. Retrieved February 25, 2019, from https://medium.com/type-thursday/learning-python-makes-you-a-better-designer-an-interview-with-just-van-rossum-8d4758c192d8

Unger, G. (2018). *Theory of Type Design*. nai010 publishers.

Woo, D. (n.d.). Type Galapagos. Retrieved January 18, 2019, from http://dannewoo.com/projects/typegalapagos.html

Wu, S.-., & Marquez, M. R. G. (2003). A non-self-intersection Douglas-Peucker algorithm. In *16th Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI 2003)* (pp. 60–66). https://doi.org/10.1109/SIBGRA.2003.1240992

Zweerink, K. (2015a). ntype. Retrieved December 9, 2018, from https://github.com/kevinzweerink/ntype

Zweerink, K. (2015b). NType. Retrieved December 9, 2018, from https://experiments.withgoogle.com/ntype

Anexos

136 — 196

Anexo A

Respostas ao questionário.

Carimbo de data/hora	Entendi o funcionamento do sistema.	Entendi o que é um esqueleto tipográfico.	Entendi o modo como é dada forma ao esqueleto tipográfico.	Gostei do aspeto visual da interface.	Achei a interface fácil de utilizar.
2019/07/10 14:48:25	5	5	5	4	4
2019/07/10 15:02:33	4	5	3	5	3
2019/07/10 15:26:05	4	5	4	4	5
2019/07/10 15:50:18	5	5	5	5	5
2019/07/10 16:34:41	5	5	5	5	5
2019/07/10 18:29:21	5	5	5	5	5
2019/07/10 18:49:17	5	5	3	5	5
2019/07/10 19:05:46	4	5	5	4	4
2019/07/10 19:28:58	5	5	4	5	5
2019/07/10 19:44:43	5	5	3	5	5
2019/07/10 23:15:27	5	5	5	5	5
2019/07/10 23:50:39	5	5	5	5	5
2019/07/11 11:11:01	5	5	5	5	5
2019/07/11 13:46:00	5	5	5	5	5
2019/07/11 14:10:18	5	5	5	5	5
Média	4,8	5	4,466666667	4,8	4,733333333
Mediana	5	5	5	5	5

Achei que os letterings gerados são visualmente diversificados.	**Achei que a variação do esqueleto tipográfico (mantendo a técnica de desenho) resulta em letterings visualmente diversificados.**	**Achei que a variação da técnica de desenho (mantendo o esqueleto tipográfico) resulta em letterings visualmente diversificados.**	**Gostei do aspeto visual dos letterings gerados.**	**Foi fácil ler o texto no lettering.**
5	4	3	4	4
5	5	5	4	5
5	4	5	5	5
5	5	5	5	4
5	4	4	5	4
5	5	5	5	5
4	2	5	5	4
4	4	5	4	3
5	4	4	5	4
5	5	5	5	4
5	5	5	5	5
5	4	5	5	4
5	4	4	5	3
5	5	5	5	3
5	4	5	5	3
4,866666667	4,266666667	4,666666667	4,8	4
5	4	5	5	4

Carimbo de data/hora	**Escrever o que não gostou do aspeto visual da interface.**	**Entendi o resultado/impacto de cada funcionalidade.**
2019/07/10 14:48:25	—	Sim
2019/07/10 15:02:33	—	Sim
2019/07/10 15:26:05	—	Sim
2019/07/10 15:50:18	—	Sim
2019/07/10 16:34:41	Dar hipótese de translinear o texto, ou reduzir o tamanho da fonte automaticamente ao chegar ao fim da linha (este caso é opcional uma vez que o texto é um placeholder de teste e é possível alterar o seu tamanho no slider)	Sim
2019/07/10 18:29:21	—	Sim
2019/07/10 18:49:17	—	Sim
2019/07/10 19:05:46	—	Não
2019/07/10 19:28:58	—	Sim
2019/07/10 19:44:43	—	Sim
2019/07/10 23:15:27	—	Sim
2019/07/10 23:50:39	—	Sim
2019/07/11 11:11:01	falta um scroll	Sim
2019/07/11 13:46:00	falta algum feedback nas tarefas de selecção da fonte tipográfica	Sim
2019/07/11 14:10:18	—	Sim

Escrever qual funcionalidade/es não entendeu.	**Utilizaria uma fonte gerada pelo sistema num trabalho.**
—	Sim
—	Sim
—	Sim
—	Sim
—	Sim
—	Sim
—	Sim
Não percebi que era para mudar o tema de cores, pensava que era para mudar a cor do tipo de letra	Sim
—	Sim
—	Sim
—	Sim
—	Sim
—	Sim
—	Sim
—	Sim

Carimbo de data/hora	Em que caso/os utilizaria uma fonte gerada pelo sistema?
2019/07/10 14:48:25	Cartazes
2019/07/10 15:02:33	Trabalho para a faculdade
2019/07/10 15:26:05	Cartazes, posters
2019/07/10 15:50:18	Para produção de animações
2019/07/10 16:34:41	Identidades de utilização única ou cartazes de eventos
2019/07/10 18:29:21	criação de uma identidade ou um lettering para um cartaz
2019/07/10 18:49:17	Num cartaz mas não para identificar uma marca.
2019/07/10 19:05:46	Identidade visual
2019/07/10 19:28:58	Títulos arriscados
2019/07/10 19:44:43	Posters, Identidade visual, para logotipos. Em alguns casos para redigir textos mais informais ou de forma experimental
2019/07/10 23:15:27	Poster, logotipo
2019/07/10 23:50:39	Em Posters. Como parte de uma identidade. Como Logotipo.
2019/07/11 11:11:01	cartazes e web
2019/07/11 13:46:00	cartazes, títulos, identidades
2019/07/11 14:10:18	No design de um poster

Carimbo de data/hora	Comentários/Sugestões?
2019/07/10 14:48:25	1. Cores dos links no about não são coerentes com o restante website. 2. Ao abrir os dropdowns dos esqueletos e da técnica de desenho, devia estar selecionada a opção atual (sugestão: inverter cores)
2019/07/10 15:02:33	mudar a cor quando se carrega no botão do esqueleto
2019/07/10 15:26:05	design simples e eficaz
2019/07/10 15:50:18	Pré-visualização da fonte no menu das fontes.
2019/07/10 16:34:41	Ter um slider para mudar a cor do fundo do site
2019/07/10 18:29:21	—
2019/07/10 18:49:17	Mais formas de preencher. Talvez formas geradas automaticamente para as possibilidades serem infinitas
2019/07/10 19:05:46	Em algumas técnicas de desenho o tipo de traço não funciona tão bem em traços retos (horizontais e vertivais) como em curvos
2019/07/10 19:28:58	—
2019/07/10 19:44:43	A área de escrita ser mais editável, permitir multíplas linhas. Mais valores para a parte do size. Edição do esqueleto.
2019/07/10 23:15:27	Pré-visualizar os diferentes esqueletos sem carregar em cada um deles.
2019/07/10 23:50:39	O ícone "i" para a info poderia estar um pouco mais pesado para igualar o ícone de trocar de tema.
2019/07/11 11:11:01	dropdown é cansativo. poderia ser algo com setinhas
2019/07/11 13:46:00	—
2019/07/11 14:10:18	—

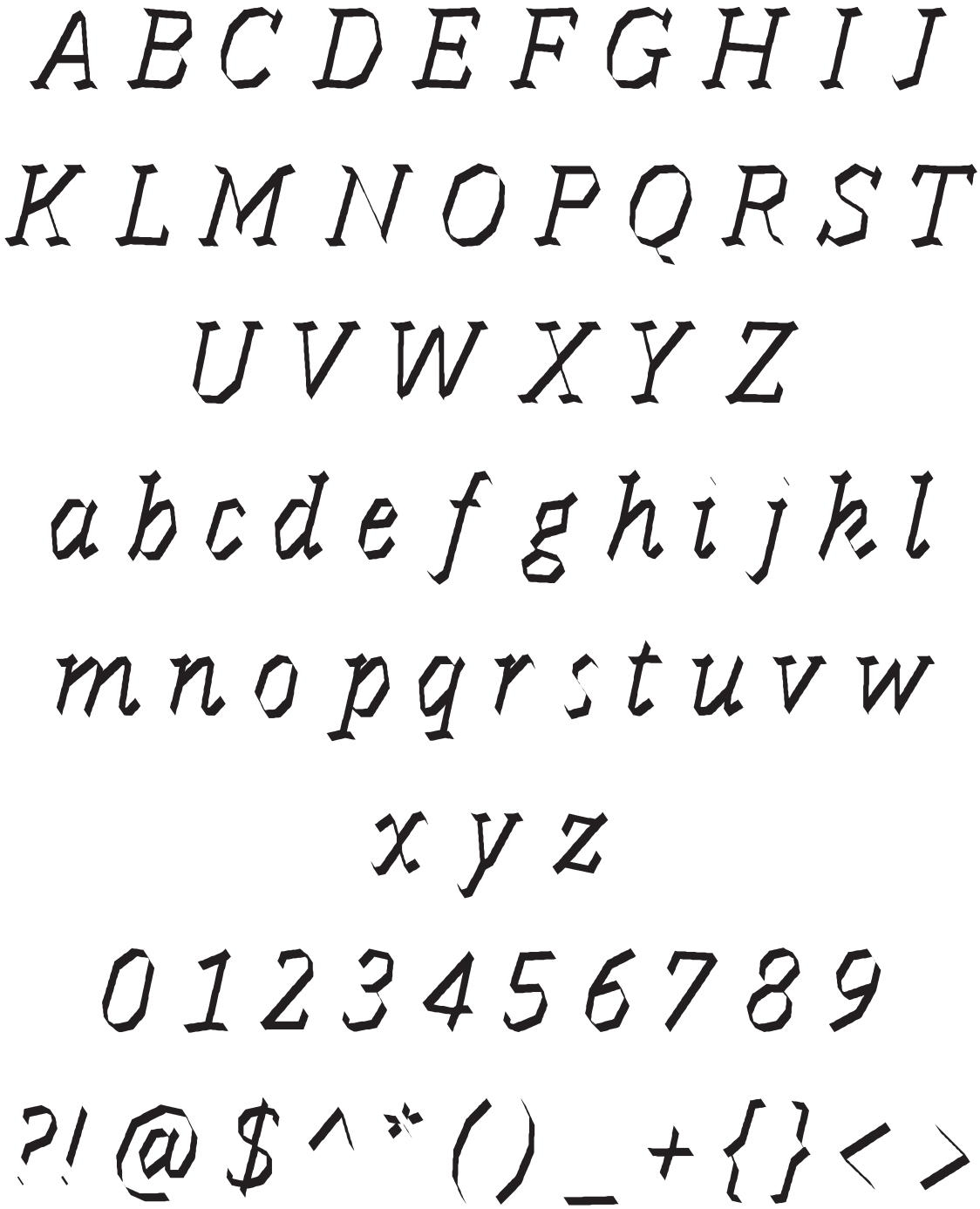
Anexo B

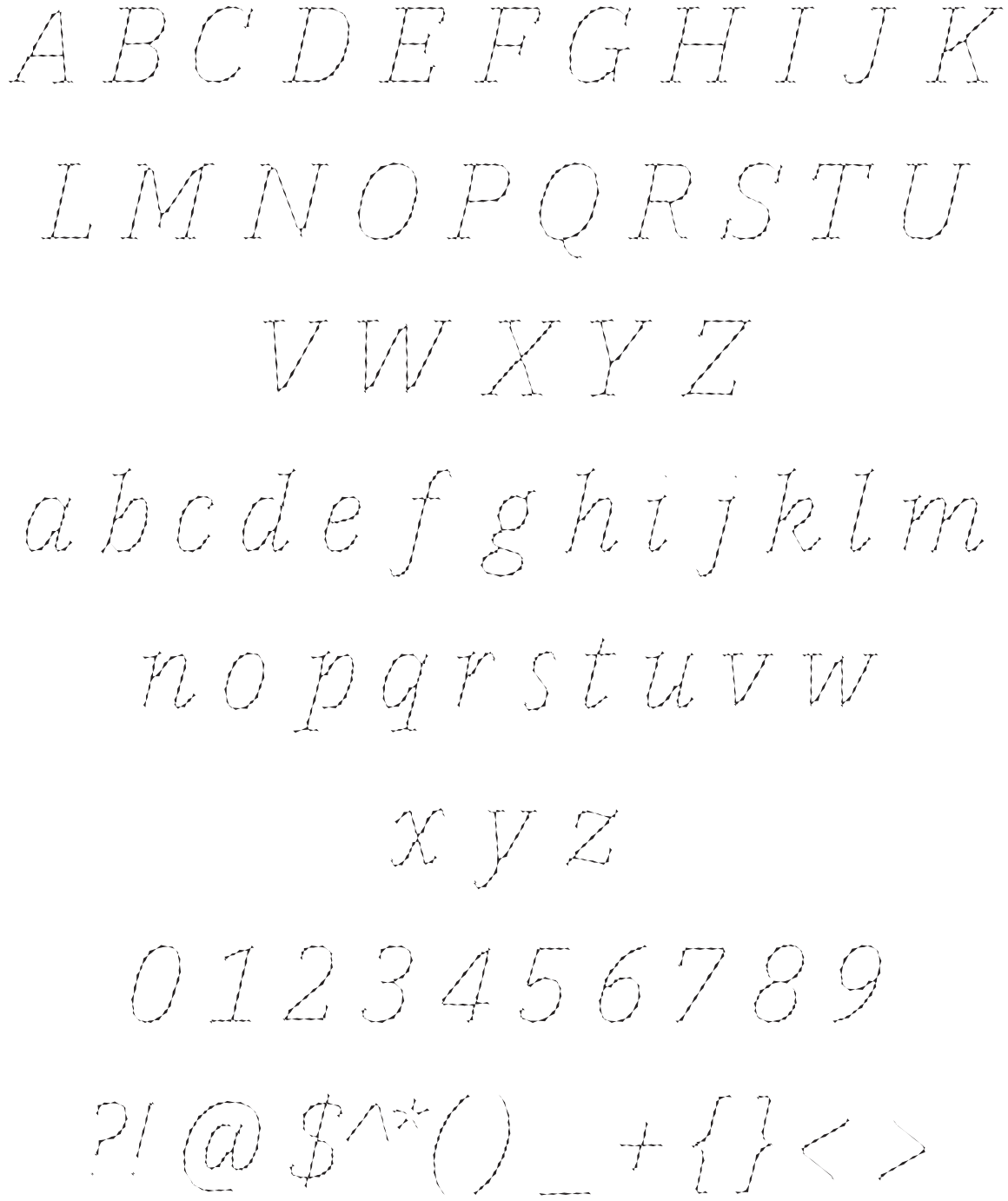
Exemplos de fontes geradas pelo sistema web.

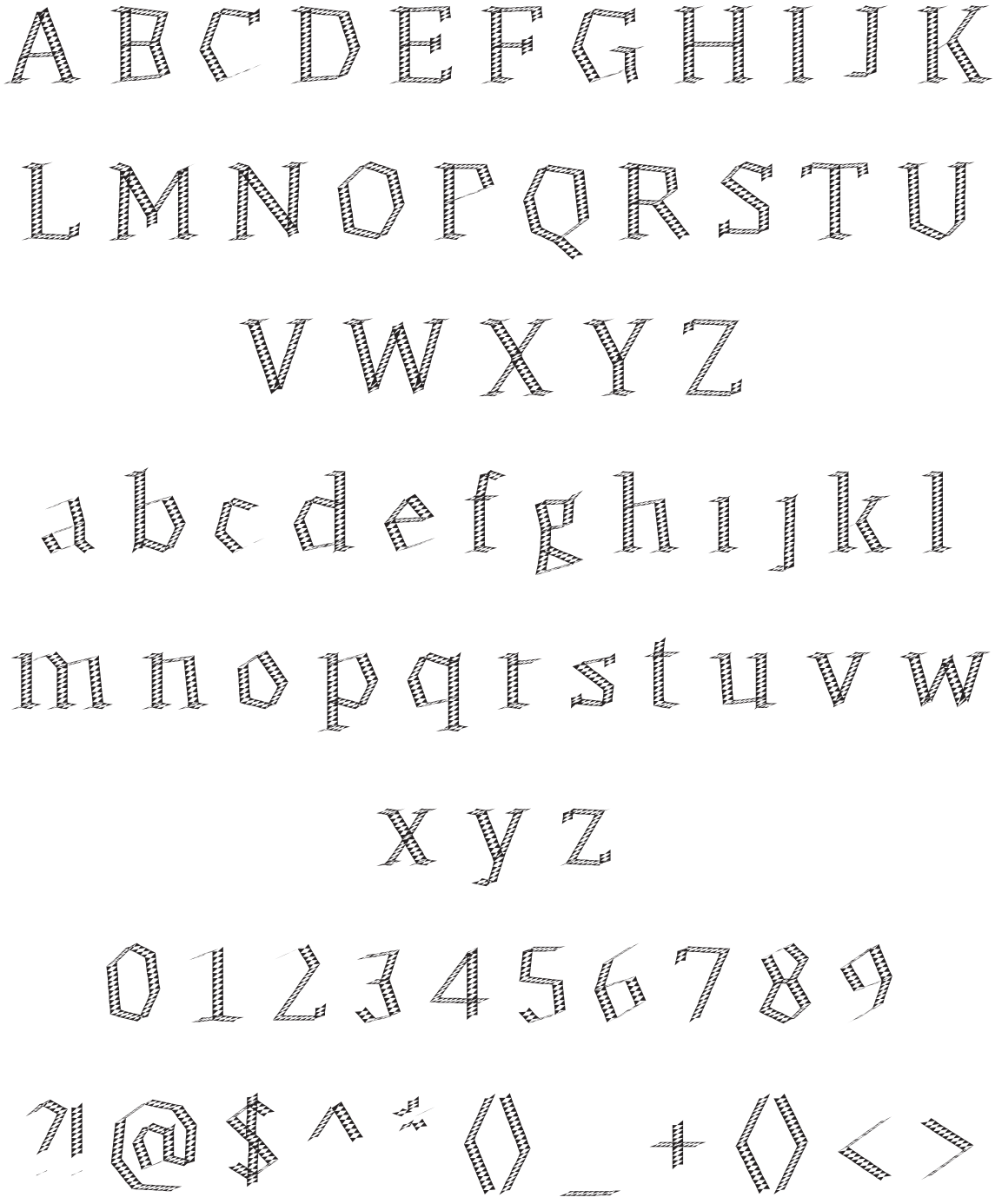

A B C D E F G H I J K
L M N O P Q R S T U V
W X Y Z
a b c d e f g h i j k l m
n o p q r s t u v w
x y z
0 1 2 3 4 5 6 7 8 9
? @ $ ^ * ( ) _ + { } < >


A B C D E F G H I J K
L M N O P Q R S T U V
W X Y Z
a b c d e f g h i j k l
m n o p q r s t u v w
x y z
0 1 2 3 4 5 6 7 8 9
? @ $ ^ * ( ) _ + { } < >

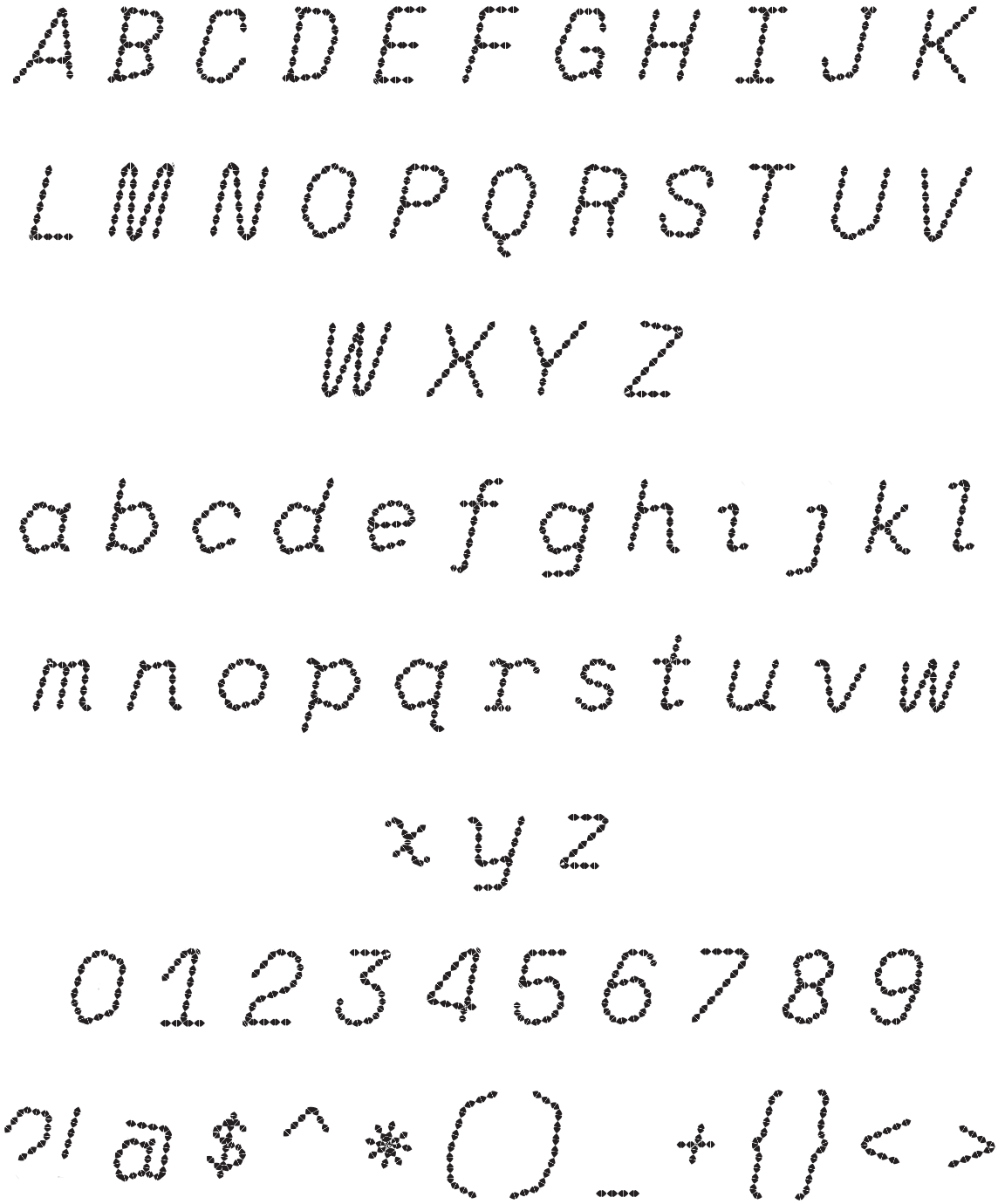

A B C D E F G H I J
K L M N O P Q R S
T U V W X Y Z
a b c d e f g h i j k l
m n o p q r s t u v w
x y z
0 1 2 3 4 5 6 7 8 9
? ! @ $ % ^ ( ) _ + { } < >

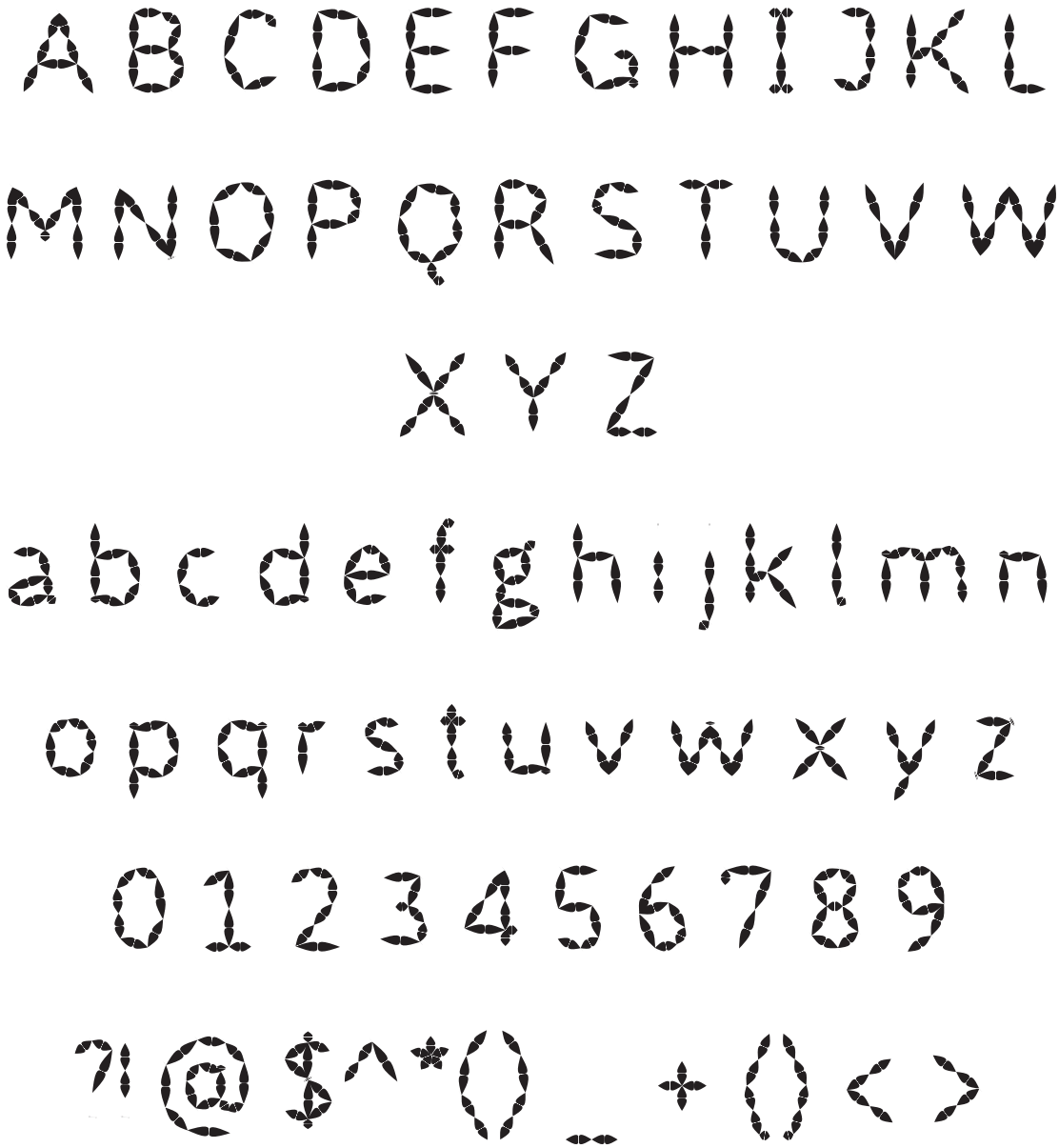

A B C D E F G H I J
K L M N O P Q R S T
U V W X Y Z
a b c d e f g h i j k l
m n o p q r s t u v w
x y z
0 1 2 3 4 5 6 7 8 9
?! @ $ ^ * ( ) _ + { } < >

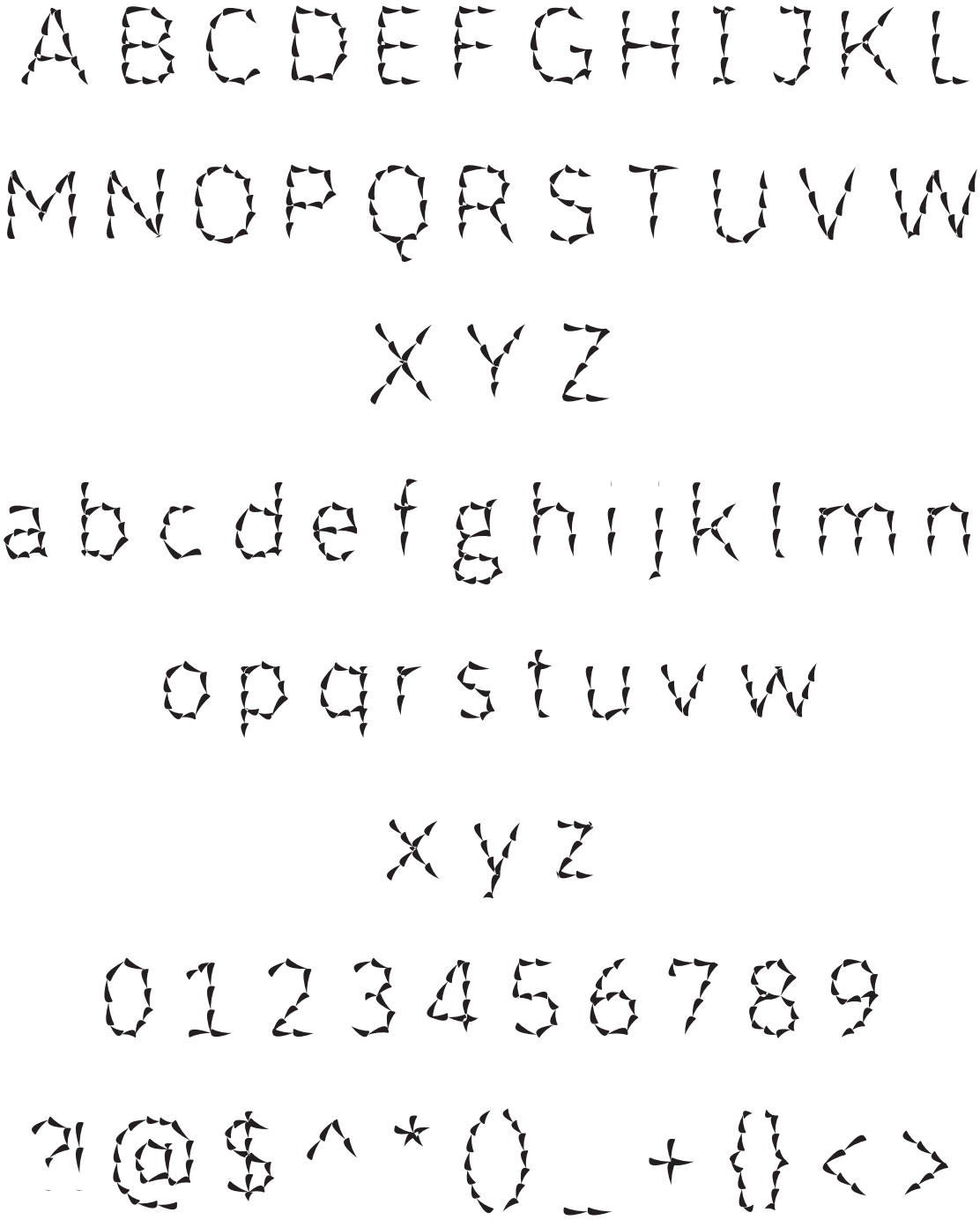

A B C D E F G H I J K
L M N O P Q R S T U
V W X Y Z
a b c d e f g h i j k l m
n o p q r s t u v w
x y z
0 1 2 3 4 5 6 7 8 9
?! @ $^*( ) _ + { } < >

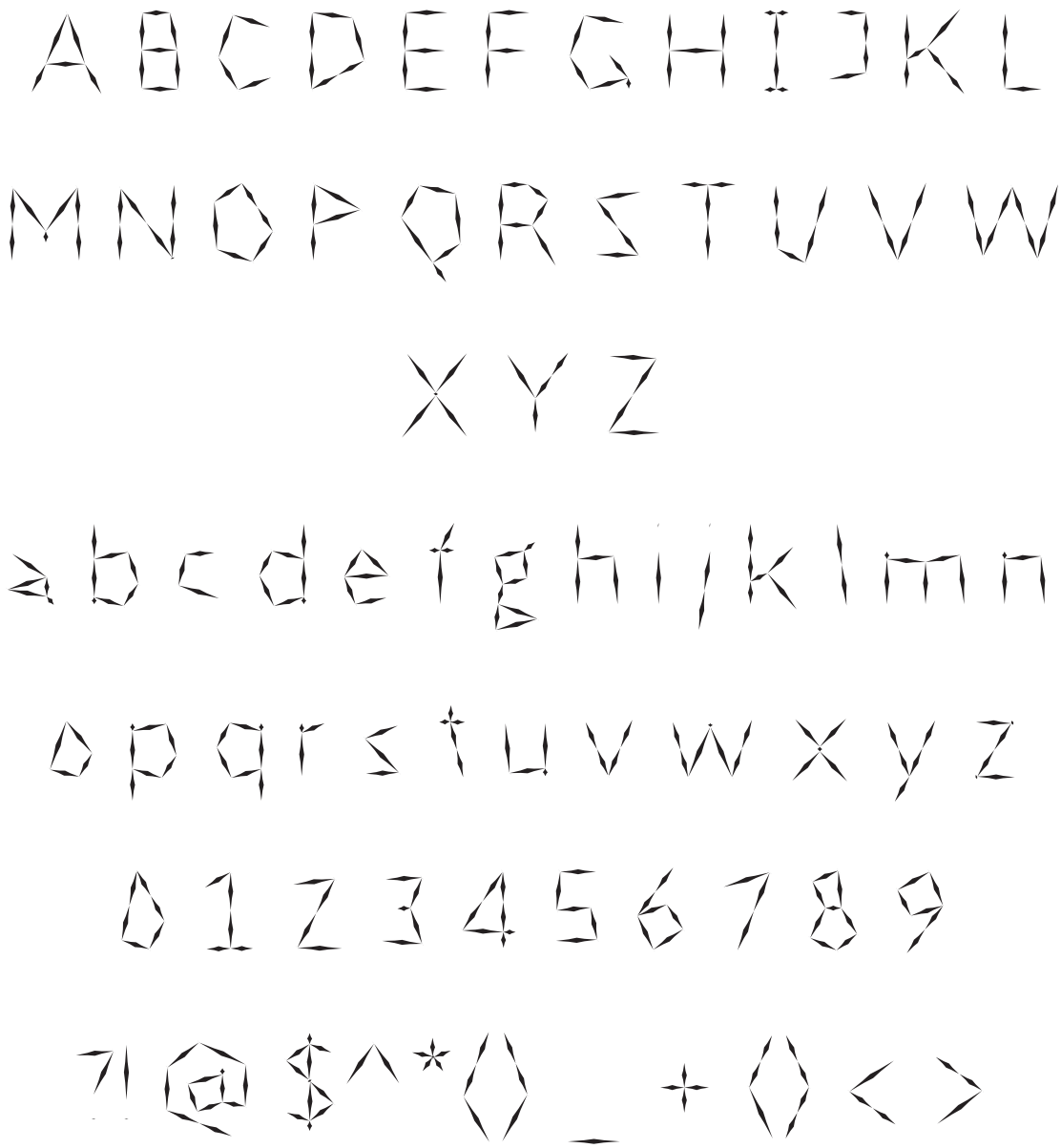

A B C D E F G H I J K
L M N O P Q R S T U
V W X Y Z
a b c d e f g h i j k l
m n o p q r s t u v w
x y z
0 1 2 3 4 5 6 7 8 9
! @ $ ^ * ( ) _ + { } < >


A B C D E F G H I J K L
M N O P Q R S T U V W
X Y Z
a b c d e f g h i j k l m n
o p q r s t u v w x y z
0 1 2 3 4 5 6 7 8 9
?! @ $ ^ * ( ) _ + { } < >


A B C D E F G H I J K
L M N O P Q R S T U V
W X Y Z
a b c d e f g h i j k l
m n o p q r s t u v w
x y z
0 1 2 3 4 5 6 7 8 9
? ! @ $ ^ * ( ) _ + { } < >

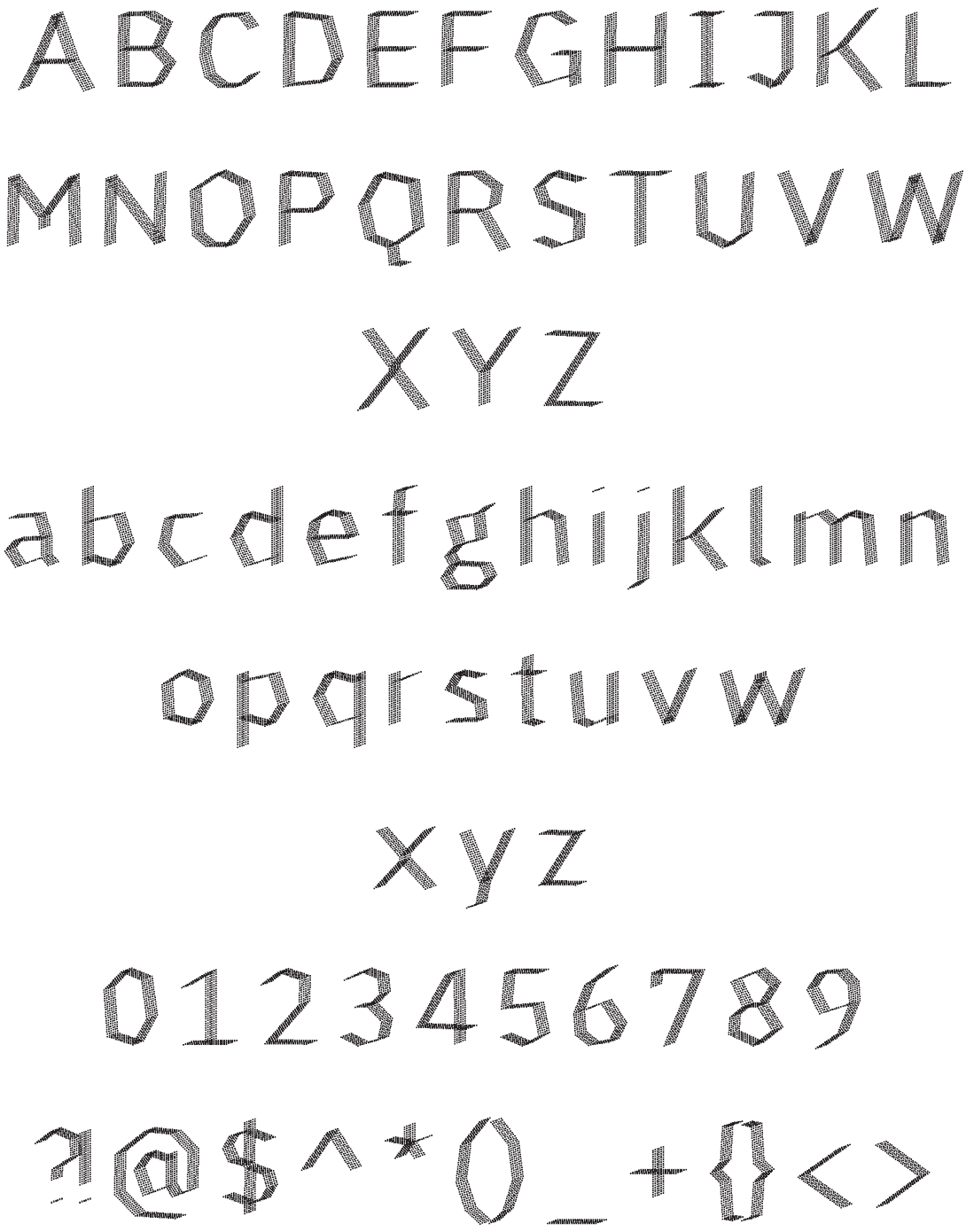

A B C D E F G H I J K L
M N O P Q R S T U V W
X Y Z
a b c d e f g h i j k l m n
o p q r s t u v w x y z
0 1 2 3 4 5 6 7 8 9
¡ @ $ ^ * ( ) _ + { } < >


A B C D E F G H I J K L
M N O P Q R S T U V W
X Y Z
a b c d e f g h i j k l m n
o p q r s t u v w
x y z
0 1 2 3 4 5 6 7 8 9
? ! @ $ ^ * ( ) _ + { } < >

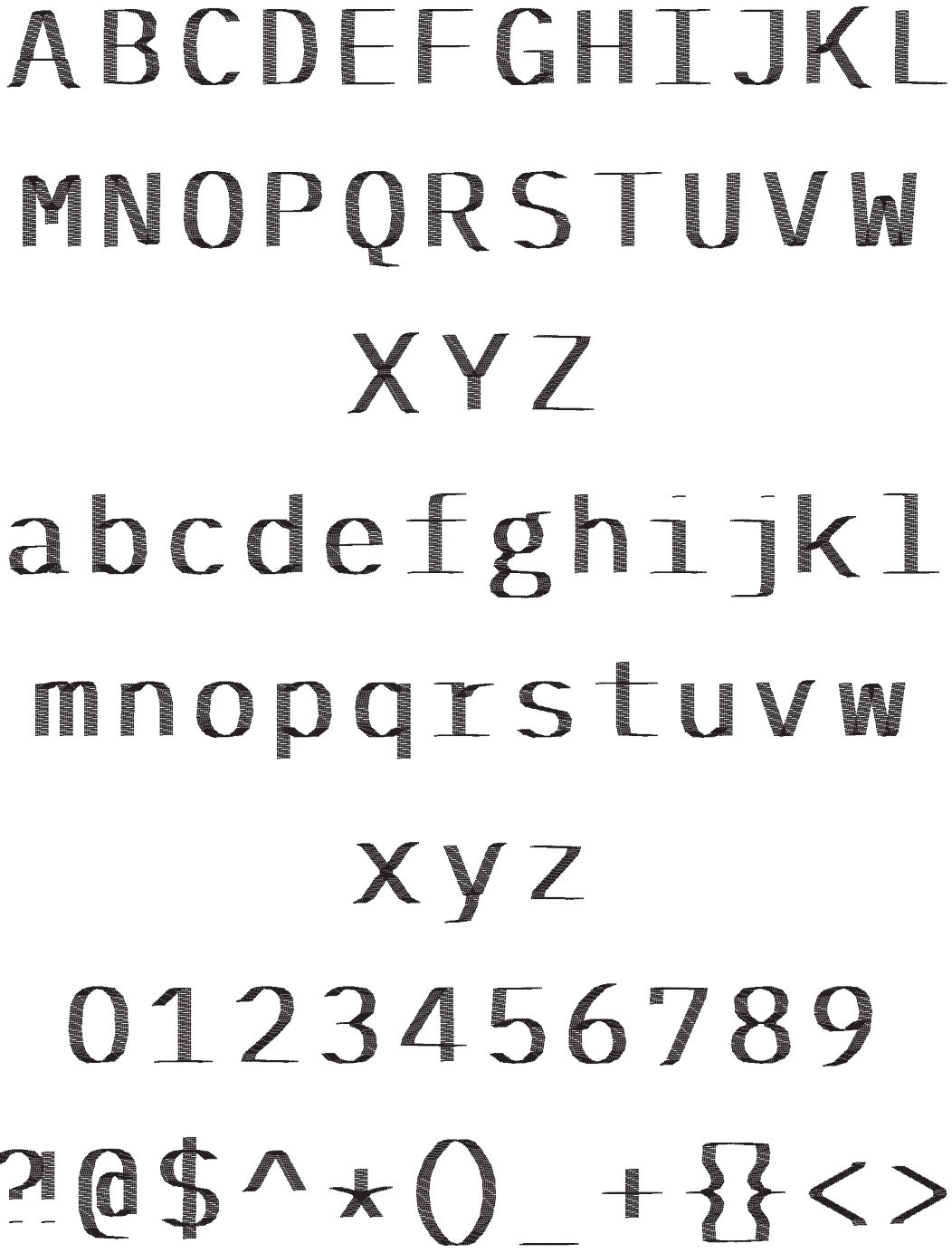

A B C D E F G H I J K L
M N O P Q R S T U V W
X Y Z
a b c d e f g h i j k l m n
o p q r s t u v w x y z
0 1 2 3 4 5 6 7 8 9
! @ $ ^ * ( ) _ + { } < >

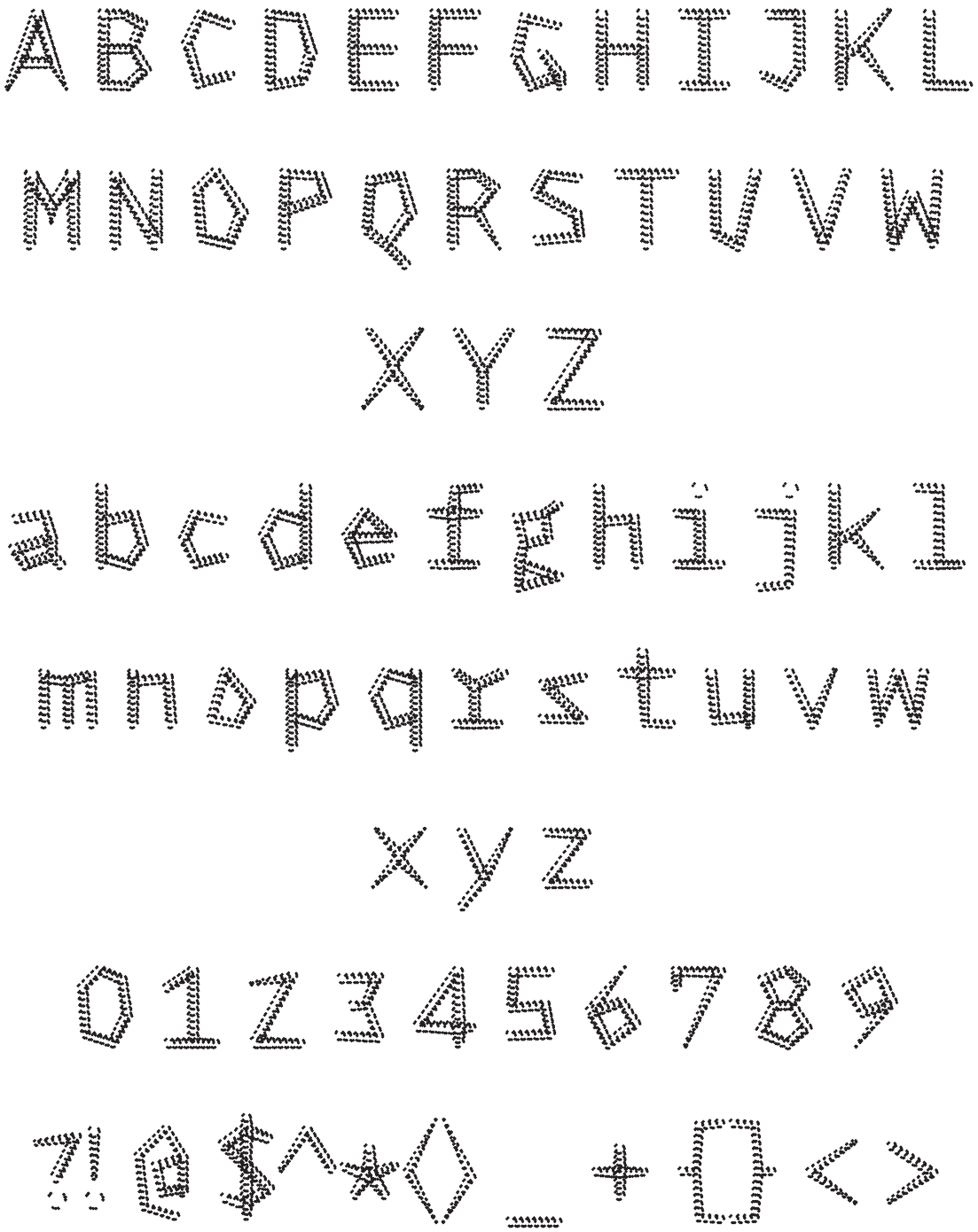

A B C D E F G H I J K L
M N O P Q R S T U V W
X Y Z
a b c d e f g h i j k l m n
o p q r s t u v w
x y z
0 1 2 3 4 5 6 7 8 9
? ! @ $ ^ * ( ) _ + { } < >


A B C D E F G H I J K
L M N O P Q R S T U V
W X Y Z
a b c d e f g h i j k l
m n o p q r s t u v w
x y z
0 1 2 3 4 5 6 7 8 9
! @ $ ^ + ( ) _ + { } < >

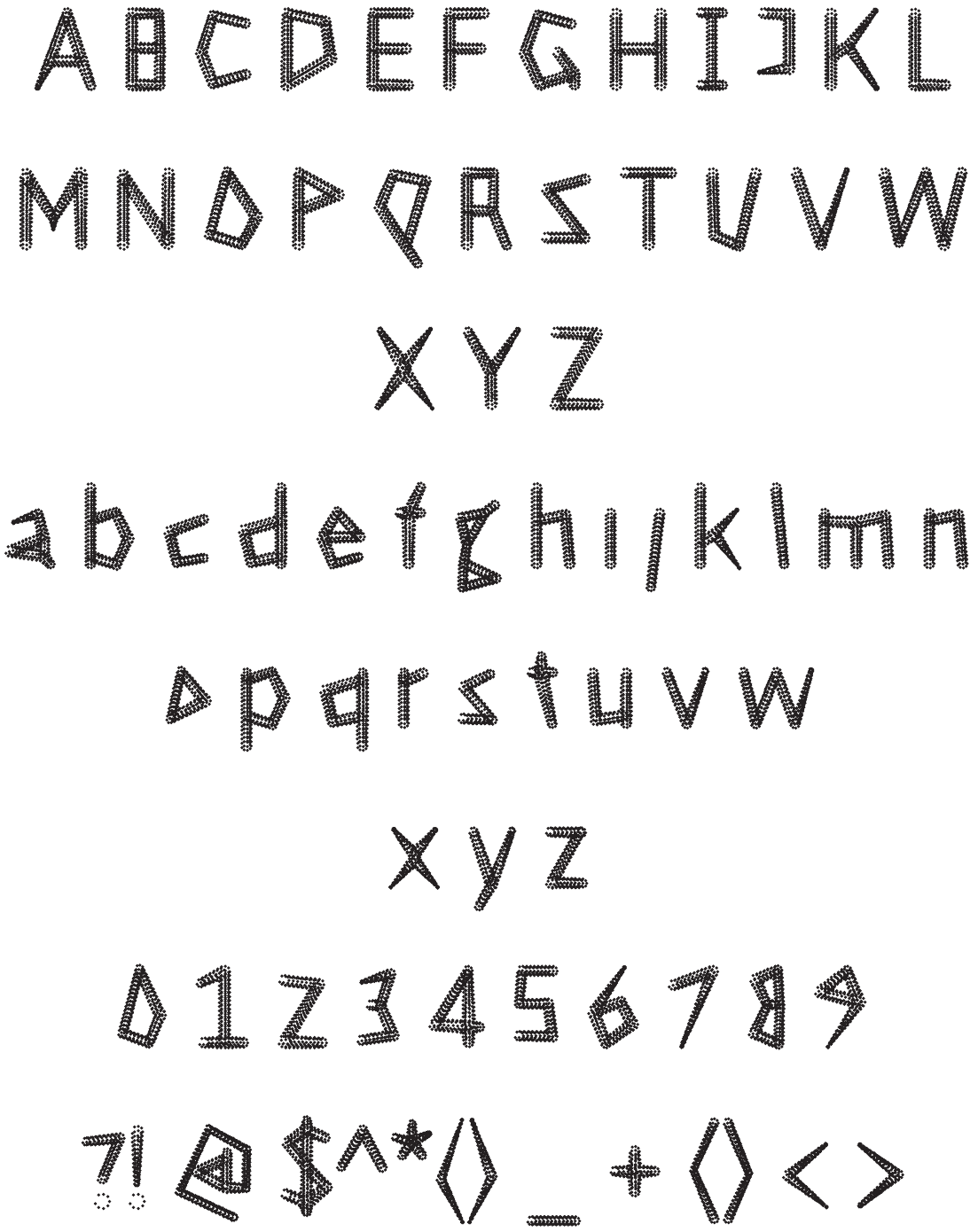

A B C D E F G H I J K L
M N O P Q R S T U V W
X Y Z
a b c d e f g h i j k l m n
o p q r s t u v w
x y z
0 1 2 3 4 5 6 7 8 9
! @ $ ^ * ( ) _ + { } < >

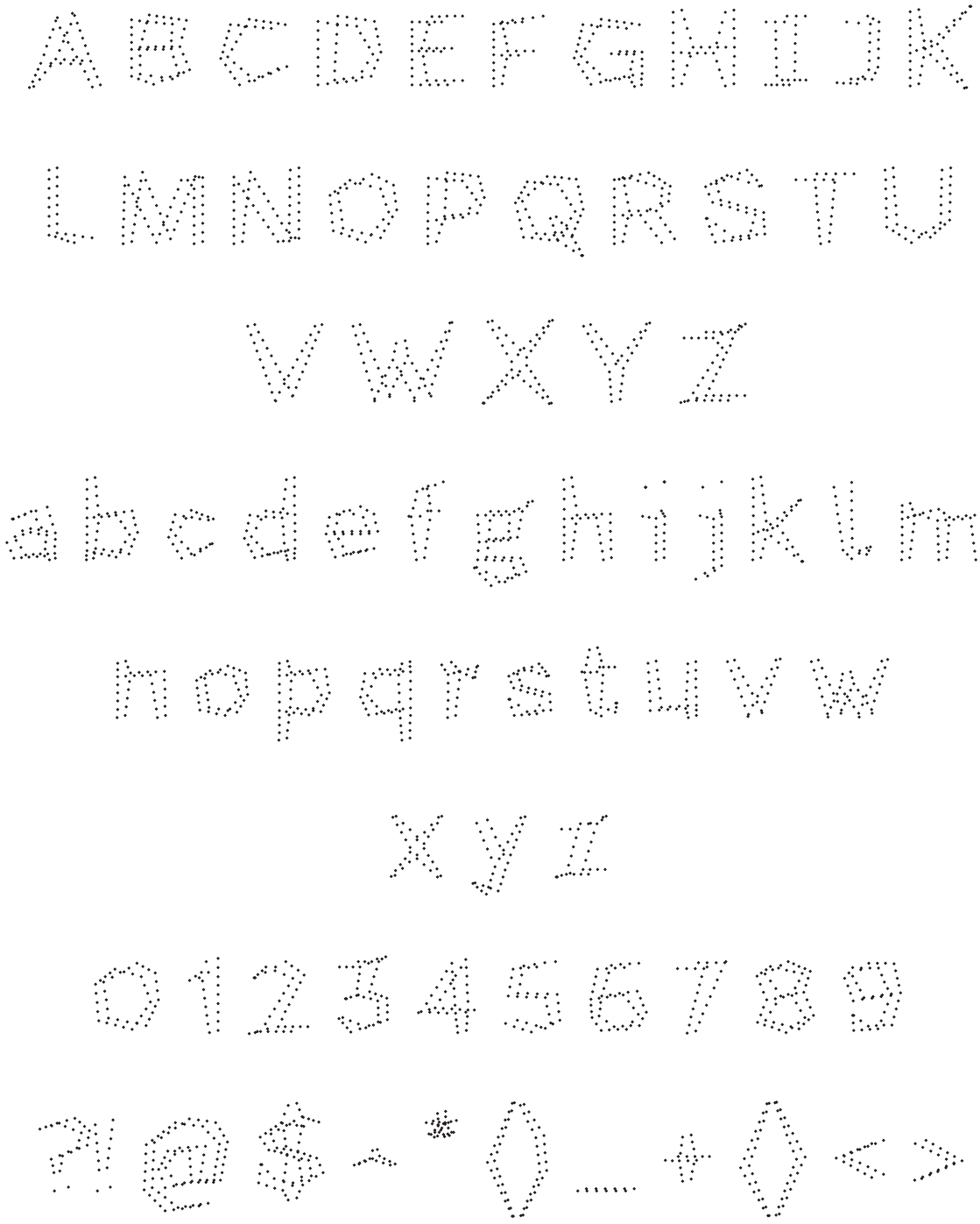

A B C D E F G H I J K
L M N O P Q R S T U V
W X Y Z
a b c d e f g h i j k
l m n o p q r s t u v
w x y z
0 1 2 3 4 5 6 7 8 9
! @ $ ^ * ( ) _ + { } < >


ABCDEFGHIJKL
MNOPQRSTUVW
XYZ
abcdefghijkl
mnopqrstuvw
xyz
0123456789
?!@$^*()_+{}<>


A B C D E F G H I J K L
M N O P Q R S T U V W
X Y Z
a b c d e f g h i j k l
m n o p q r s t u v w
x y z
0 1 2 3 4 5 6 7 8 9
? ! @ $ ^ * ( ) _ + - = < >


A B C D E F G H I J K L M
N O P Q R S T U V W
X Y Z
a b c d e f g h i j k l m n o
p q r s t u v w
x y z
0 1 2 3 4 5 6 7 8 9
? ! @ $ ^ " ( ) _ + [ ] < >

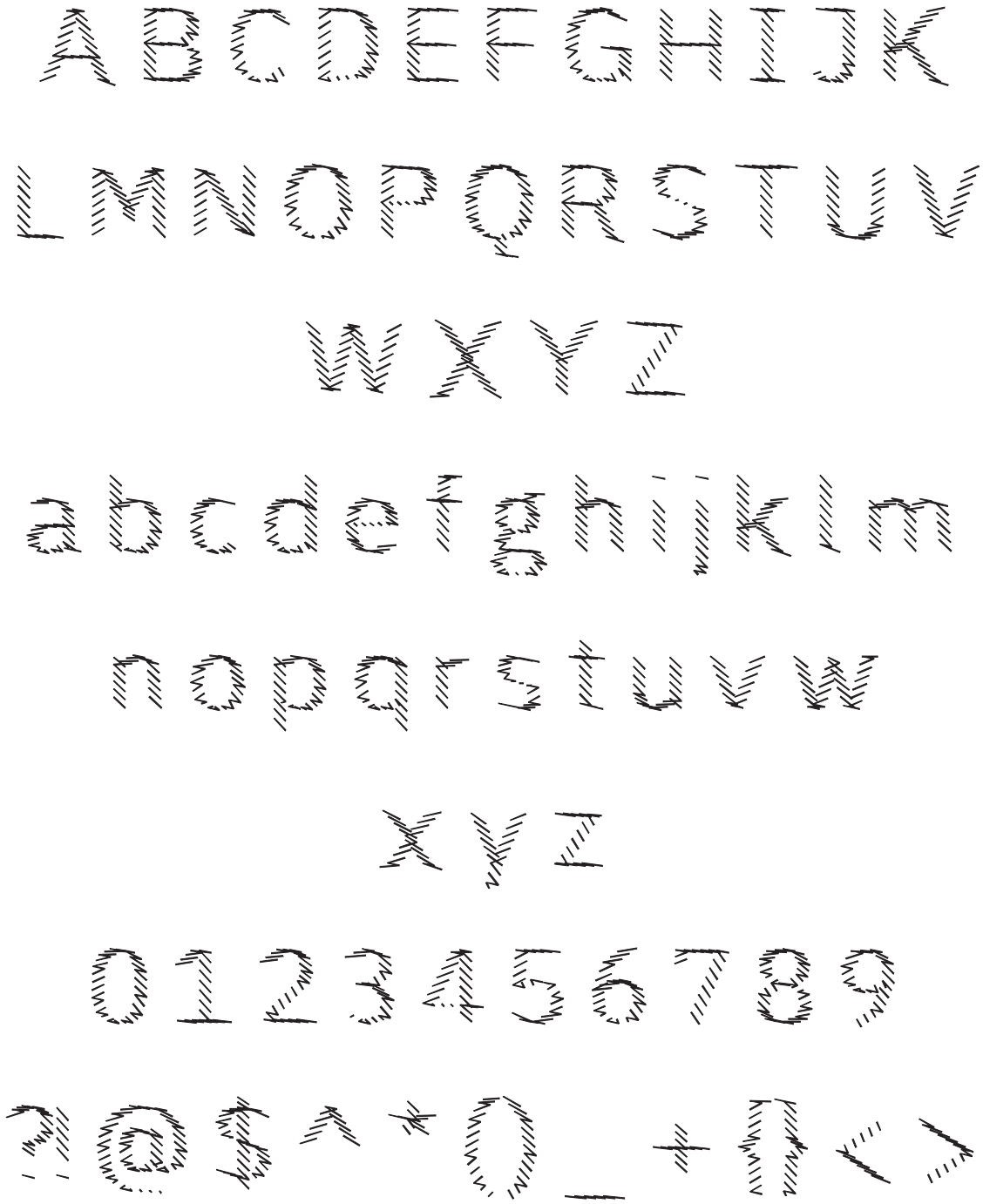

A B C D E F G H I J K L
M N O P Q R S T U V W
X Y Z
a b c d e f g h i j k l m n
o p q r s t u v w
x y z
0 1 2 3 4 5 6 7 8 9
? ! @ $ ^ * ( ) _ + { } < >

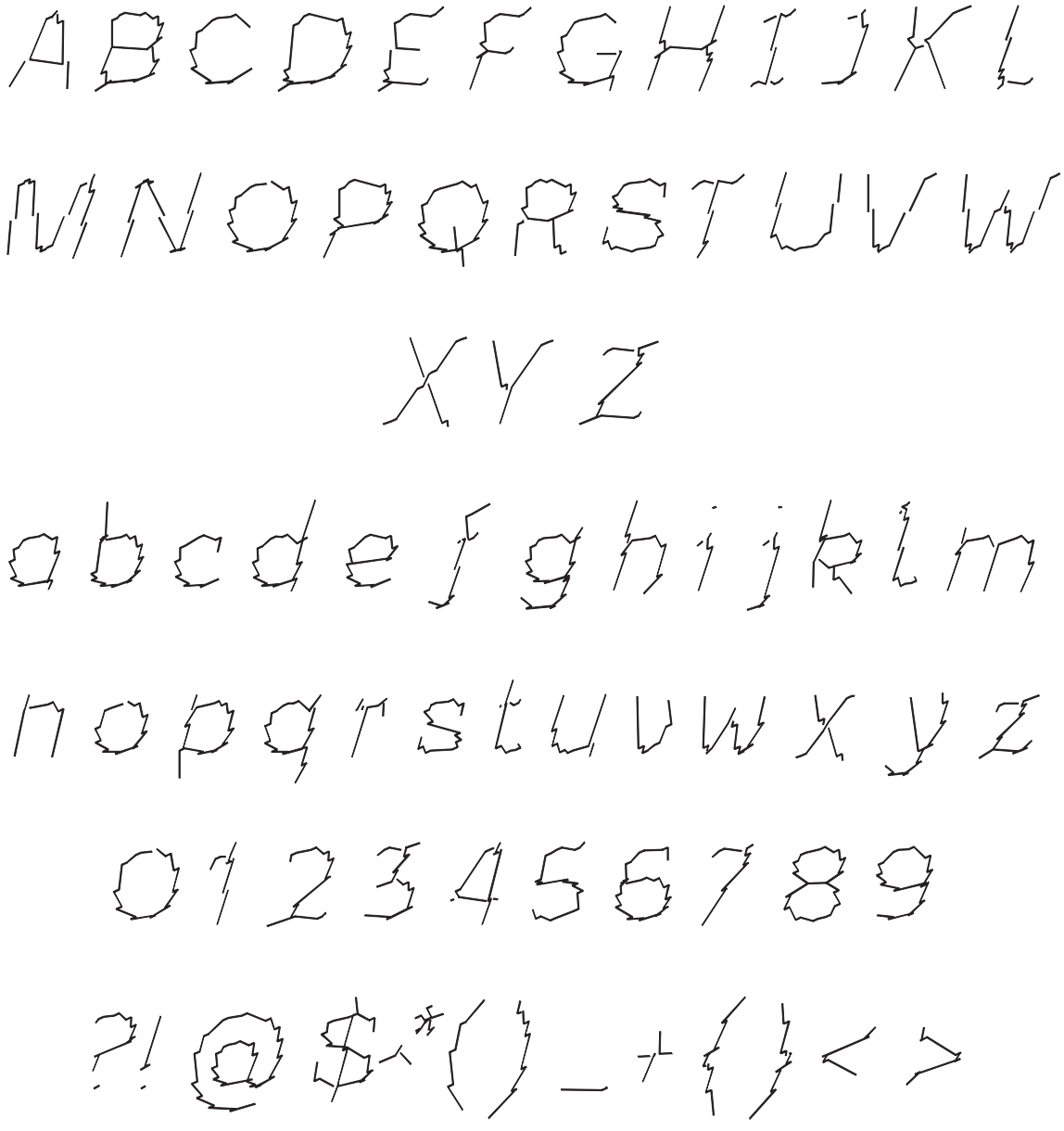

A B C D E F G H I J K
L M N O P Q R S T U
V W X Y Z
a b c d e f g h i j k l m
n o p q r s t u v w
x y z
0 1 2 3 4 5 6 7 8 9
? ! @ $ - * ( ) _ + { } < >

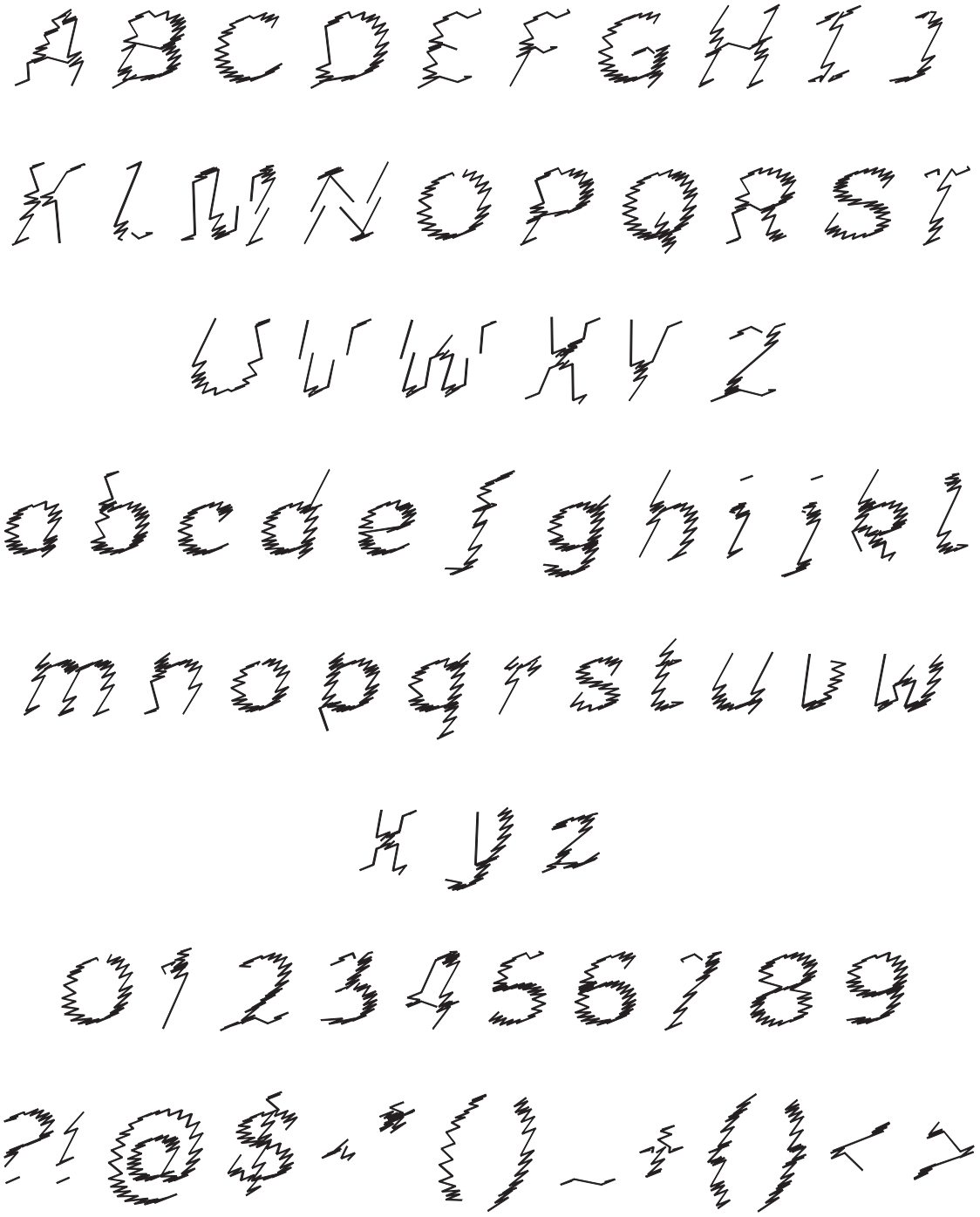

A B C D E F G H I J K L
M N O P Q R S T U V
W X Y Z
a b c d e f g h i j k l m
n o p q r s t u v w
x y z
0 1 2 3 4 5 6 7 8 9
? ! @ $ - * ( ) _ + { } < >


A B C D E F G H I J K
L M N O P Q R S T U
V W X Y Z
a b c d e f g h i j k l m
n o p q r s t u v w
x y z
0 1 2 3 4 5 6 7 8 9
? ! @ $ - * ( ) _ + { } < >

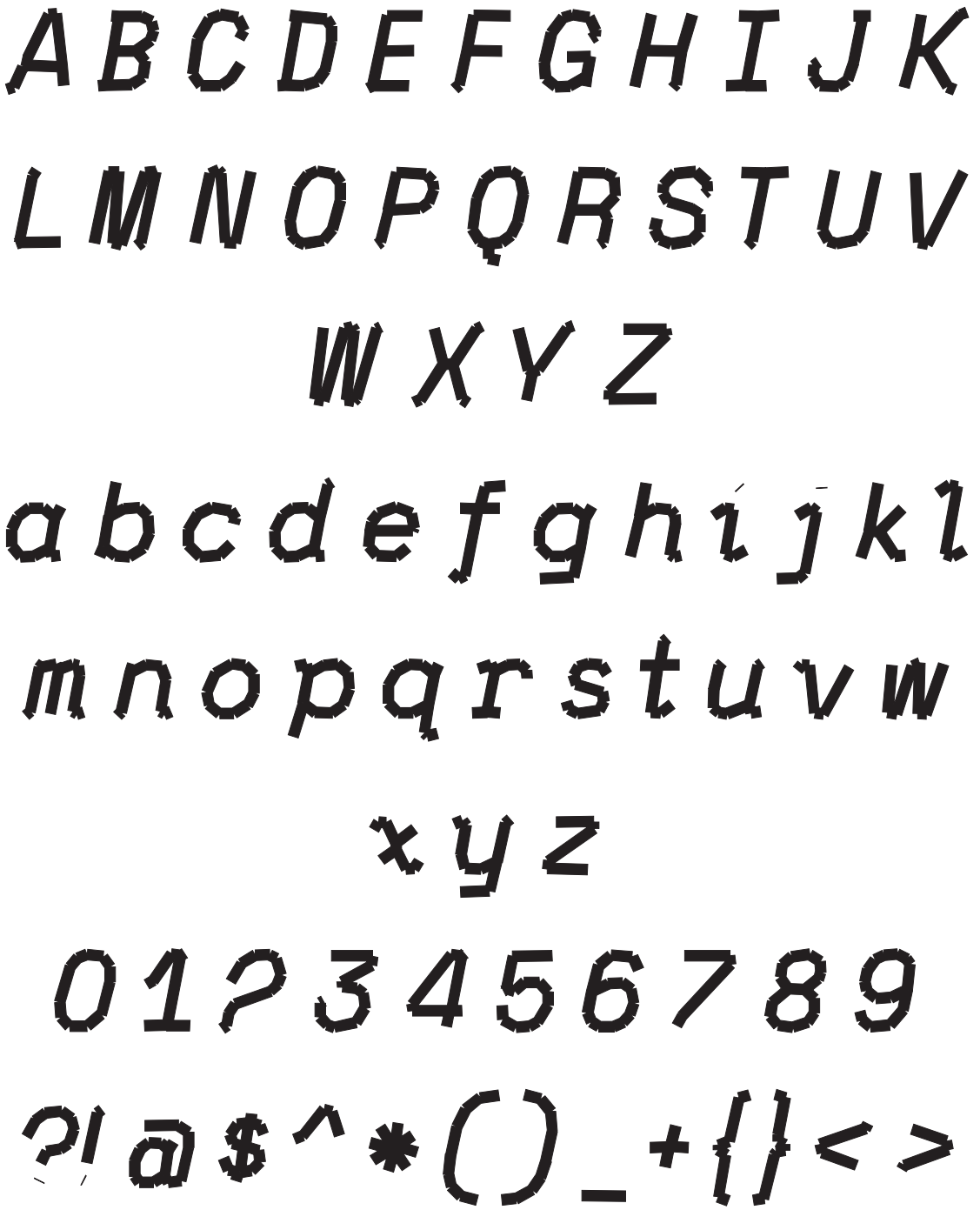

A B C D E F G H I J K L M N
O P Q R S T U V W X Y Z
a b c d e f g h i j k l m n o
p q r s t u v w x y z
0 1 2 3 4 5 6 7 8 9
? ! @ $ ^ * ( ) _ + { } < >


ABCDEFGHIJK
LMNOPQRSTUV
WXYZ
abcdefghijklm
nopqrstuvw
xyz
0123456789
!@£$%^&*()_+[]{};:

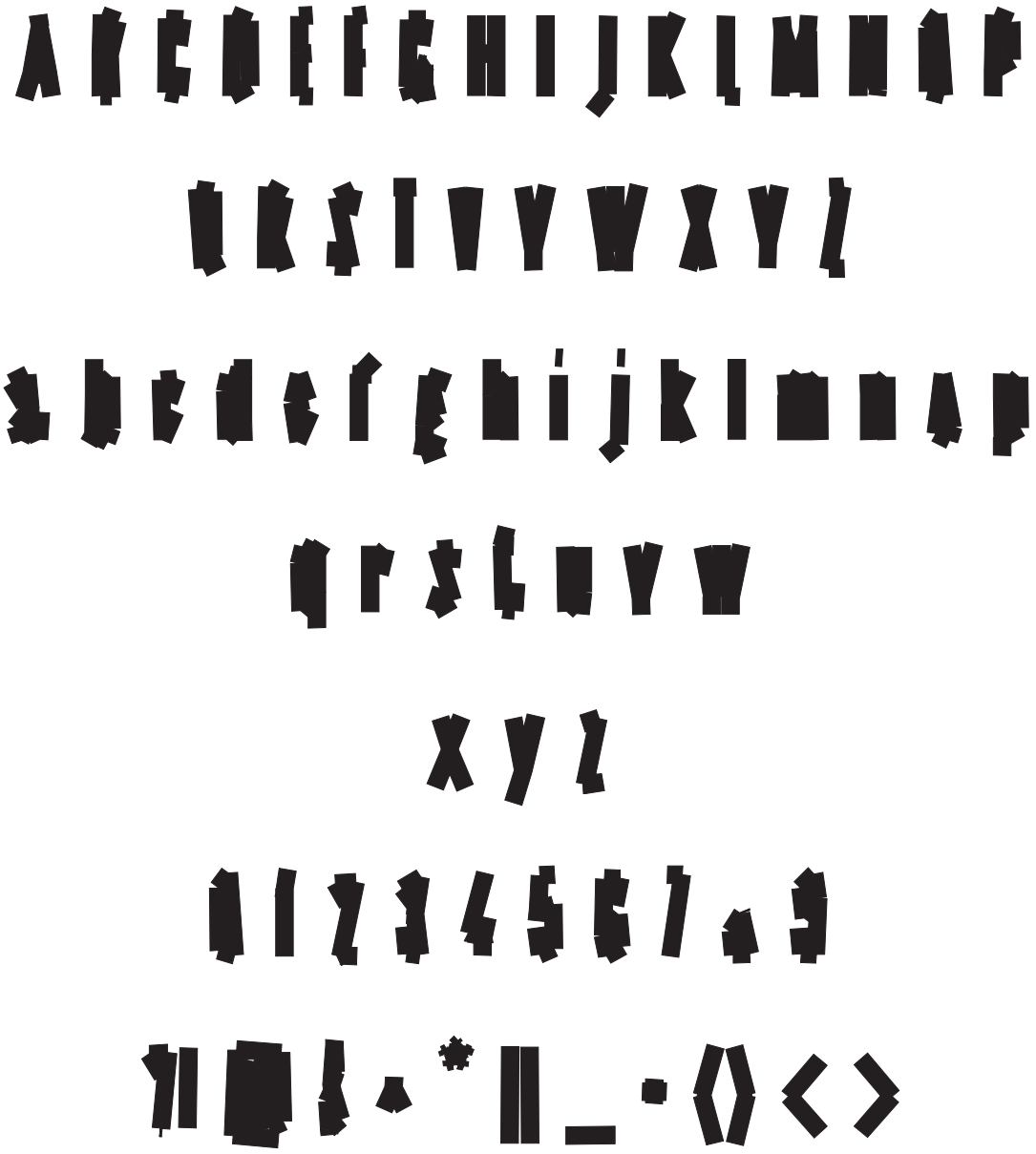




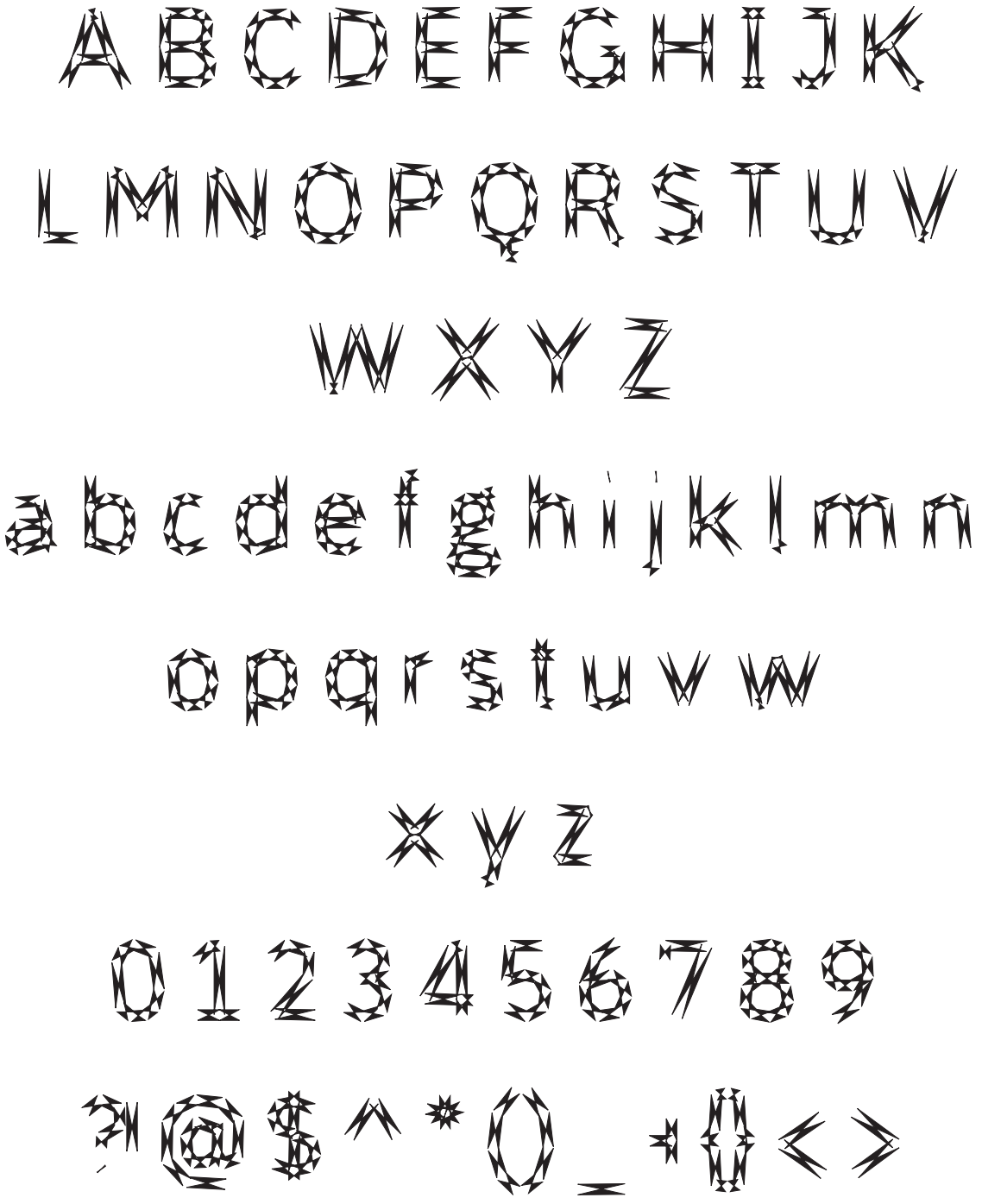

ABCDEFGHIJKLM
NOPQRSTUVWXYZ
abcdefghijklm
nopqrstuvwxyz
0123456789
?!a$^*()_+{}<>


A B C D E F G H I J K
L M N O P Q R S T U V
W X Y Z
a b c d e f g h i j k l
m n o p q r s t u v w
x y z
0 1 2 3 4 5 6 7 8 9
?! @ $ ^ * ( ) _ + { } < >


A B C D E F G H I J K
L M N O P Q R S T U V
W X Y Z
a b c d e f g h i j k
l m n o p q r s t u v
w x y z
0 1 2 3 4 5 6 7 8 9
? ! @ $ ^ • ( ) _ + { } < >




A B C D E F G H I J K
L M N O P Q R S T U
V W X Y Z
a b c d e f g h i j k l
m n o p q r s t u v w
x y z
0 1 2 3 4 5 6 7 8 9
? ! @ $ * ( ) _ + { } < >


A B C D E F G H I J K
L M N O P Q R S T U V
W X Y Z
a b c d e f g h i j k l m n
o p q r s t u v w
x y z
0 1 2 3 4 5 6 7 8 9
? @ $ ^ * ( ) _ + { } < >


A B C D E F G H I J K L M
N O P Q R S T U V W
X Y Z
a b c d e f g h i j k l m n
o p q r s t u v w x y z
0 1 2 3 4 5 6 7 8 9
?! @ $ ^ * ( ) _ + { } < >

A B C D E F G H I J K L

M N O P Q R S T U V W

X Y Z

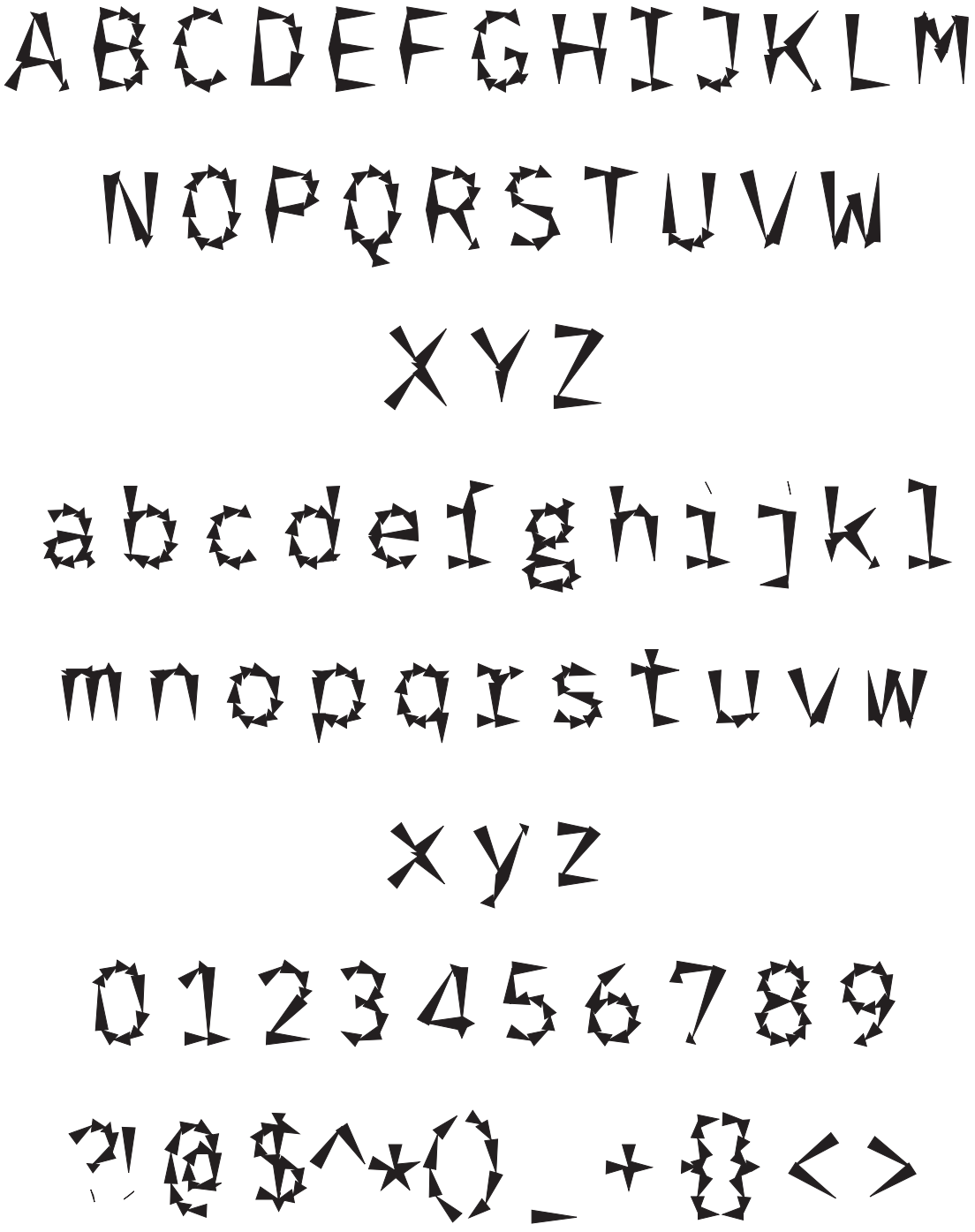
a b c d e f g h i j k l m

n o p q r s t u v w

x y z

0 1 2 3 4 5 6 7 8 9

?! @ $ ^ * ( ) _ + { } < >


A B C D E F G H I J K L M
N O P Q R S T U V W
X Y Z
a b c d e f g h i j k l
m n o p q r s t u v w
x y z
0 1 2 3 4 5 6 7 8 9
?! @ $ ^ * ( ) _ + { } < >

A B C D E F G H I J K L

M N O P Q R S T U V W

X Y Z

a b c d e f g h i j k l

m n o p q r s t u v w

x y z

0 1 2 3 4 5 6 7 8 9

! @ $ ^ * ( ) _ + { } < >


A B C D E F G H I J K L
M N O P Q R S T U V
W X Y Z
a b c d e f g h i j k l m
n o p q r s t u v w
x y z
0 1 2 3 4 5 6 7 8 9
? ! @ $ - * ( ) _ + { } < >

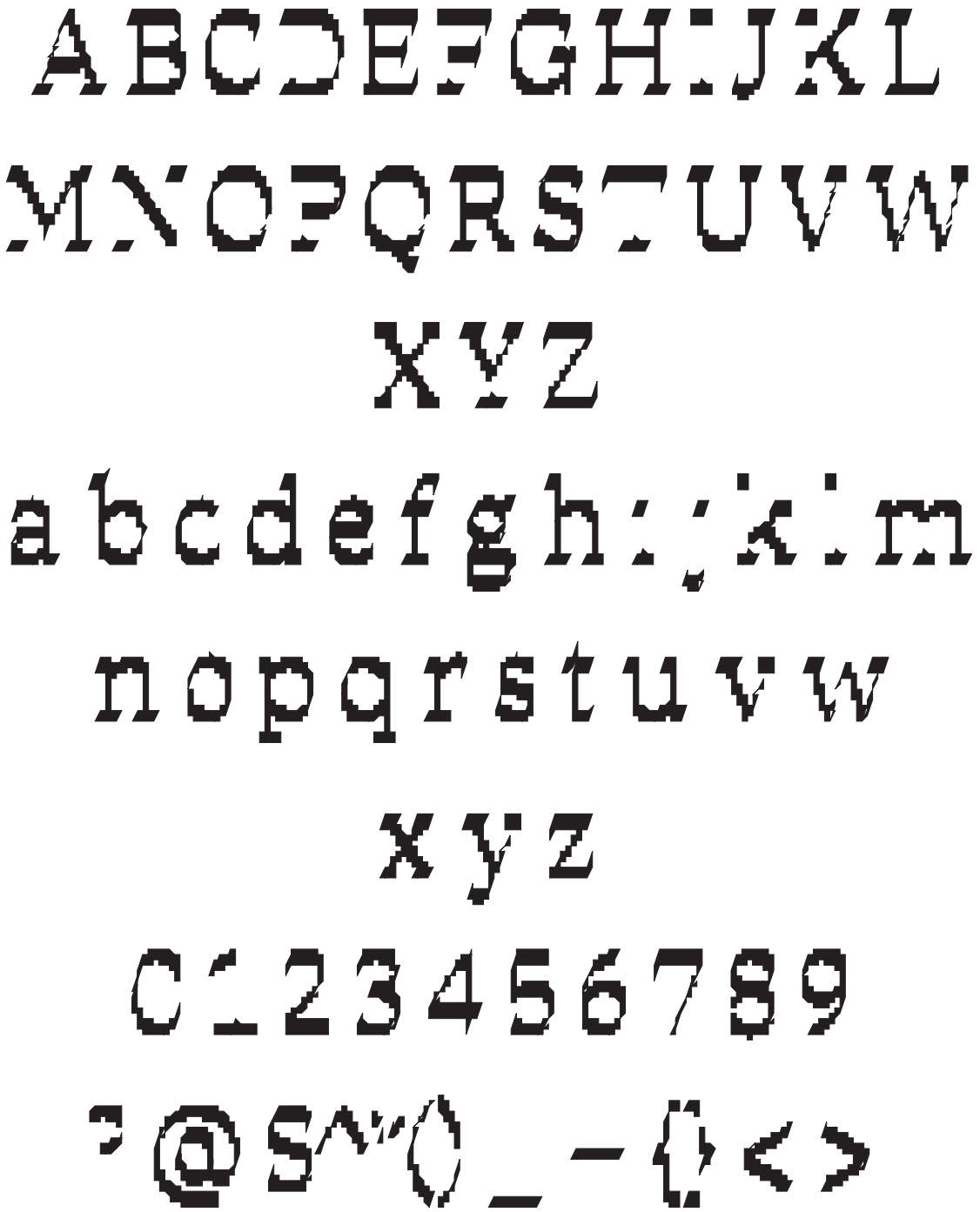

ABCDEFGHIJ
KLMNOPQRS
TUVWXYZ
abcdefghijk
lmnopqrstuv
wxyz
0123456789
?!@$*()_+{}<>


ABCDEFGHIJKLMN
OPQRSTUVWXYZ
abcdefghijkl
mnopqrstuvw
xyz
0123456789
?!@$^*()_+{}<>


ABCDEFGHIJKLMN
OPQRSTUVWXYZ
abcdefghijklm
nopqrstuvwxyz
0123456789
?!@$^*()_+{}<>

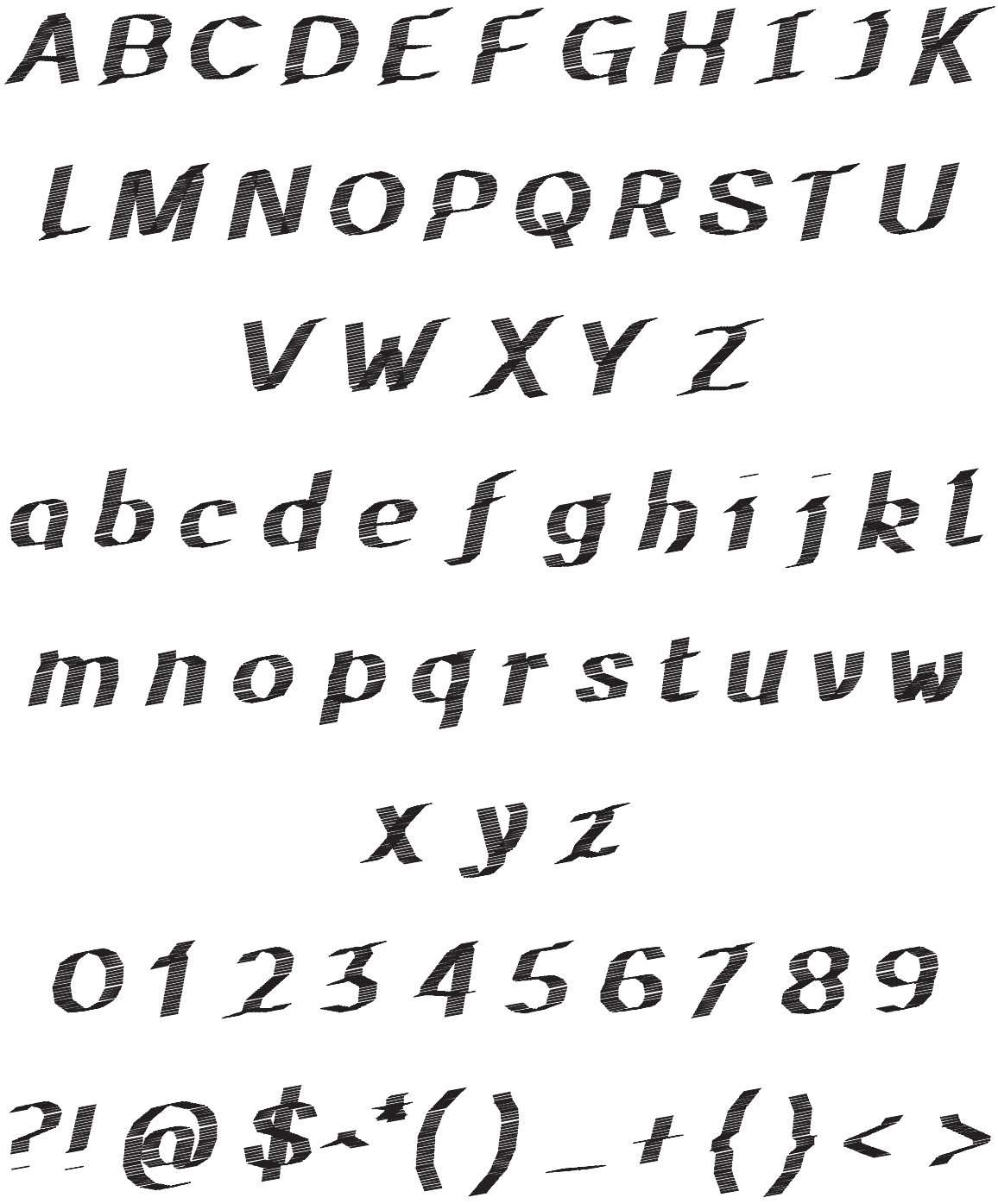

ABCDEFGHIJKL
MNOPQRSTUVW
XYZ
abcdefghijkl
mnopqrstuvw
xyz
0123456789
?!@$^*()_+{}<>

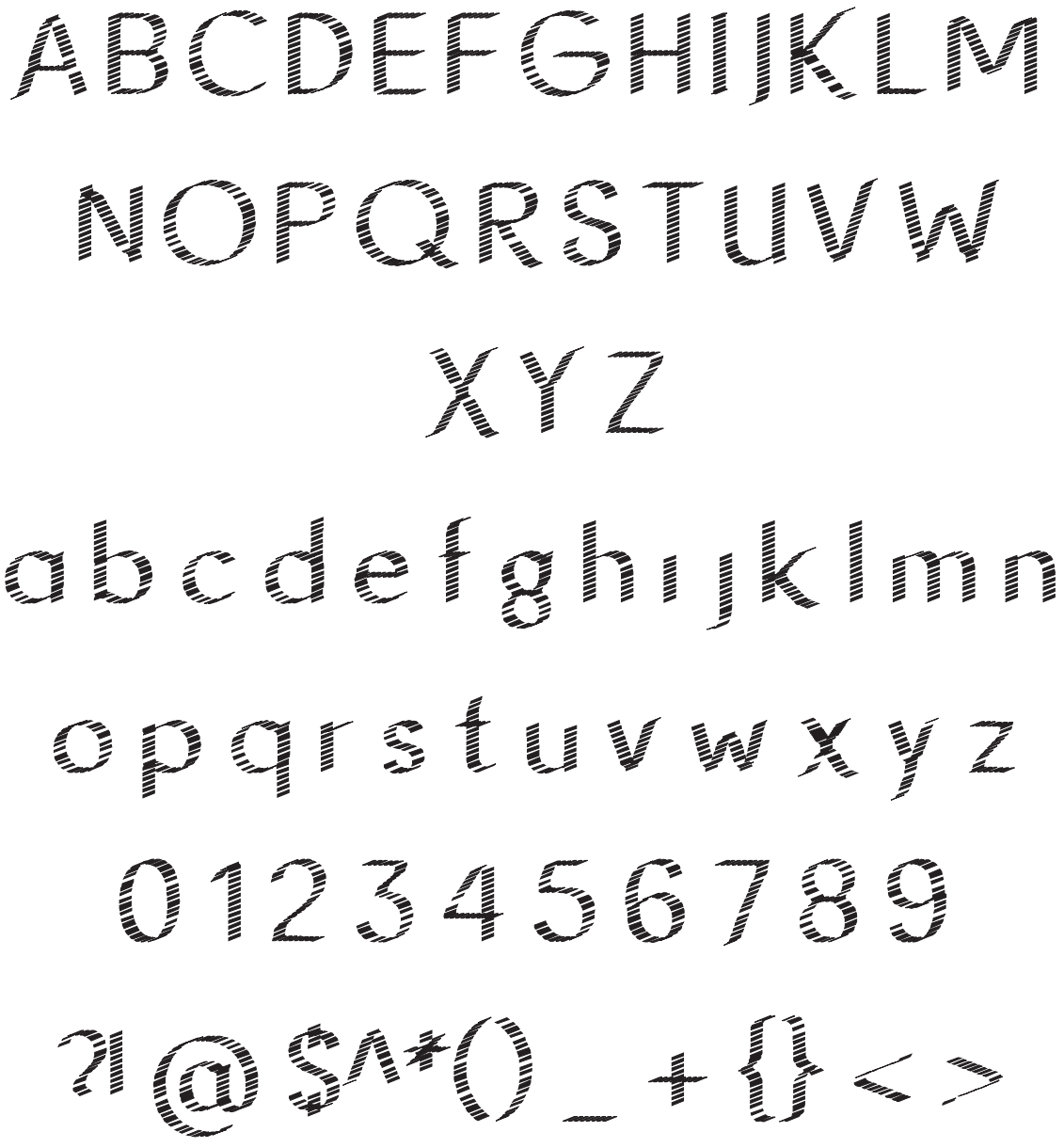

ABCDEFGHIJKL
MNOPQRSTUV
WXYZ
abcdefghijkl
mnopqrstuvw
xyz
0123456789
?!@$*()[]{}<>

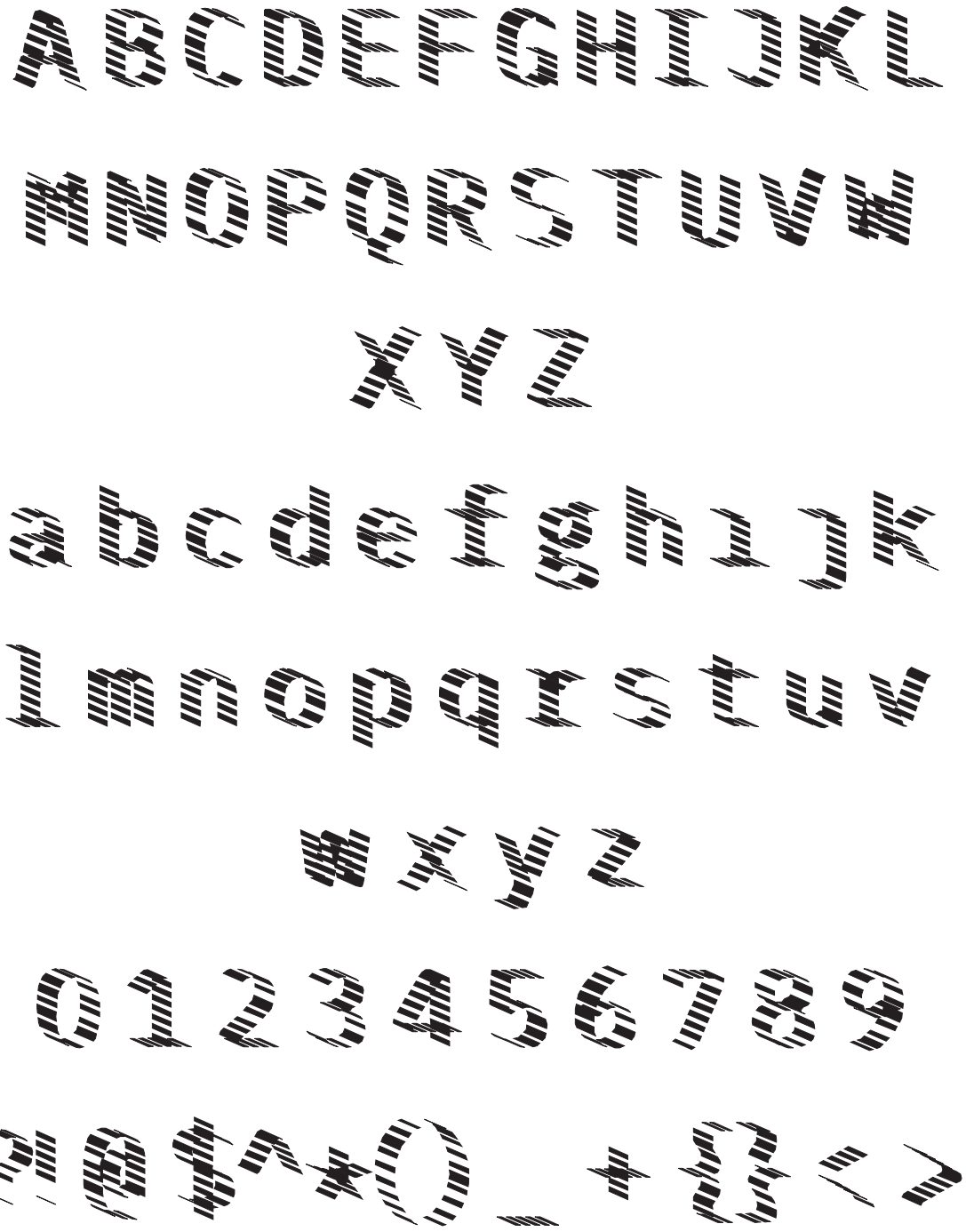

ABCDEFGHIJKL
MNOPQRSTUVW
XYZ
abcdefghijklm
nopqrstuvw
xyz
0123456789
?@$%()_-{}<>


ABCDEFGHIJKLM
NOPQRSTUVWXYZ
abcdefghıjkl
mnopqrstuvw
xyz
0123456789
?!@$^*()_+{}<>


A B C D E F G H I J K L M N O P Q
R S T U V W X Y Z
a b c d e f g h i j k l m n o p q
r s t u v w x y z
0 1 2 3 4 5 6 7 8 9
? ! @ $. * ( ) _ + { } ( )


A B C D E F G H I J K
L M N O P Q R S T U
V W X Y Z
a b c d e f g h i j k l
m n o p q r s t u v w
x y z
0 1 2 3 4 5 6 7 8 9
?! @ $ - * ( ) _ + { } < >


ABCDEFGHIJKLM
NOPQRSTUVW
XYZ
abcdefghıjklmn
opqrstuvwxyz
0123456789
?! @ $^*() _ + {} < >


A B C D E F G H I J K L
M N O P Q R S T U V W
X Y Z
a b c d e f g h i j k
l m n o p q r s t u v
w x y z
0 1 2 3 4 5 6 7 8 9
! @ $ ^ * ( ) _ + { } < >