

Proposta de Gcode baseado em imagem BMP para impressão 3D em meio Cerâmico

BMP based GCode Communication proposal for 3D Printing on Ceramic Material

RESUMO

Este trabalho se insere em um projeto que visa desenvolver um sistema para impressão 3D de material poroso em cerâmica pelo princípio de jateamento de aglutinante (3DP). Um dos requerimentos desejáveis era que, a partir dos dados levantados fosse possível obter uma alternativa aberta e flexível em relação ao controle de atuadores no processo de impressão 3DP. Com base neste levantamento e na identificação das necessidades do sistema proposto, propõe-se o desenvolvimento de um padrão de comunicação GCode a partir de imagens BMP. Este formato deve simular a impressão por varredura utilizando cartuchos jatos de tinta, utilizando o Marlin como firmware neste aparato, além de permitir a sua aplicação em projetos distintos, como o reaproveitamento de impressoras 3DP defeituosas. O resultado do padrão de comunicação obtido contempla a codificação das principais informações para processar uma imagem BMP.

PALAVRAS-CHAVE: Impressão 3D. Manufatura Aditiva. Freeware.

ABSTRACT

This work is part of a project that aims to develop a system for 3D printing of porous ceramic material by the binder jetting principle (3DP). One of the desirable requirements was that, based on the collected data, it would be possible to obtain an open and flexible alternative to control the 3DP process. Based on this, and on the identification of the proposed system needs, it is proposed to develop a GCode communication format based on BMP images. This format should simulate a raster printing using inkjet cartridges, using Marlin as its firmware in this device, allowing its application on different projects, including the reuse of defective 3DP printers. The result of the obtained format contemplates the encoding of the main information to process a BMP image.

KEYWORDS: 3D Printing. Additive Manufacturing. Freeware.

INTRODUÇÃO

Kayque Xavier Scapolan
Kayque03ii@gmail.com
Universidade Tecnológica
Federal do Paraná, Curitiba,
Paraná, Brasil

Phd Neri Volpato
nvolpato@utfpr.edu.br
Universidade Tecnológica
Federal do Paraná, Curitiba,
Paraná, Brasil

Recebido:

Aprovado:

Direito autorial: Este trabalho está licenciado sob os termos da Licença Creative Commons-Atribuição 4.0 Internacional.



4 EDUCAÇÃO DE QUALIDADE



9 INDÚSTRIA, INOVAÇÃO E INFRAESTRUTURA



17 PARCERIAS E MEIOS DE IMPLEMENTAÇÃO



Nos processos denominados de manufatura aditiva (AM – Additive Manufacturing) ou de impressão 3D, as peças são produzidas através da adição de material, camada a camada, sem utilizar ferramentas externas, mudança de orientação da peça, moldes ou dispositivos de fixação e com pouca interferência humana (VOLPATO, 2017).

O processo de impressão por meio de jateamento de aglutinante (3DP) funciona através da deposição de um líquido aglutinante sobre um leito de pó geralmente cerâmico, de forma sequencial por meio de camadas, para que obtenhamos uma peça impressa.

Este princípio de Impressão 3D requer que o *firmware* receba instruções de cada camada de uma peça por meio de imagens tipo BMP. A partir destas imagens, as informações necessárias para realizar a varredura de um cartucho *inkjet* (cartuchos de jato de tinta, seja este térmico, ou piezoelétrico), são obtidas. Não foi encontrado um padrão de GCode específico para esta aplicação. Neste sentido este trabalho visa propor um GCode que contemple tais necessidades.

METODOLOGIA

Foi realizada a análise de alguns projetos semelhantes já existentes sendo que o projeto que mais se aproximou da proposta foi o projeto do grupo ytec3d (YTEC3D, 2015) com a impressora denominada de Oásis 3DP. Outro projeto que foi de grande auxílio foi a PlanB. Estes projetos são impressoras 3DP feitas inteiramente de maneira open source (projeto de código aberto). O seu princípio de funcionamento é bastante documentado, permitindo a sua ampla utilização em outros projetos. Entretanto, um problema encontrado é que ambos os projetos utilizam firmwares escritos especificamente para eles, o que limita a flexibilidade do projeto (YTEC3D, 2015).

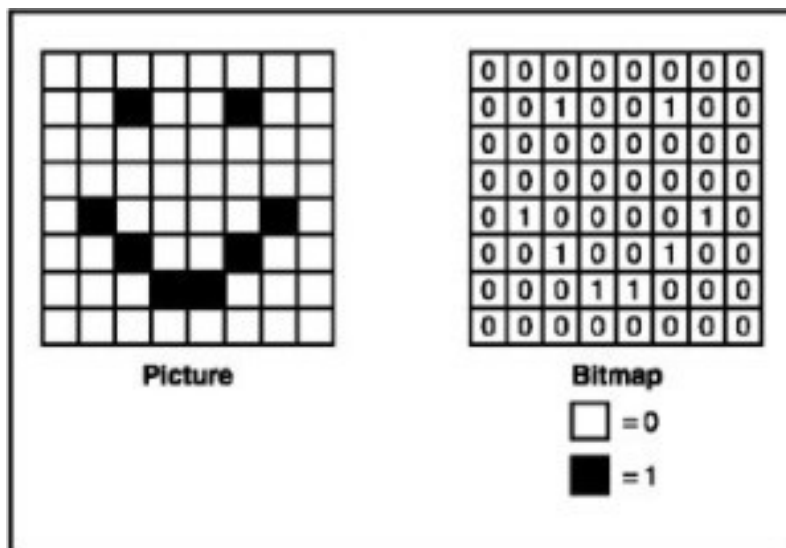
Visando deixar o sistema mais flexível, foi estudado a possibilidade de se utilizar o firmware open-source Marlin (que funciona por meio de GCode), que é amplamente empregado em impressoras 3D. Entretanto, em seu estado original, este não oferece suporte para controlarmos um cartucho de impressora. Algo notado no projeto da impressora PlanB foi a forma que é feita a interface entre o cartucho e o Arduino, realizada através de um Shield, chamado Inkshield (LEWIS, 2012). Este utiliza CI's multiplexadores para comunicar o cartucho HP c6602 com o micro controlador responsável por executar o código embarcado. O projeto Inkshield é bastante popular e já inspirou inúmeros projetos distintos, tais como impressoras inkjet abertas, e bioimpressoras. Tendo como base a Inkshield, surgiu a revisão do firmware Marlin feita por Steve Kelly (KELLY, 2013), que visa justamente adicionar tal funcionalidade ao firmware.

Convém mencionar que o princípio de funcionamento de uma impressora de extrusão de material (tipo FDM – Fused Deposition Modeling) é distinto ao da impressora por jateamento de aglutinante (popularmente conhecida como impressão 3DP). Enquanto uma funciona com base de movimentação vetorial, e a outra funciona por meio de varredura de uma determinada área. Para a melhor resolução se considera que um bico injetor do cartucho equivale a um pixel de

uma imagem (TYLER, 2014). No projeto em questão, foi utilizado um cartucho modelo HP c6602 que possui 12 injetores (nozzle), o que indica que ele é capaz de varrer 12 linhas de uma vez só (WOODFORD, 2018). Uma imagem não é interpretada diretamente pelo Marlin, uma vez que este somente reconheça arquivos GCode, para isto é necessário um código GCode que indique para a impressora como fazer a varredura de uma imagem do tipo bitmap.

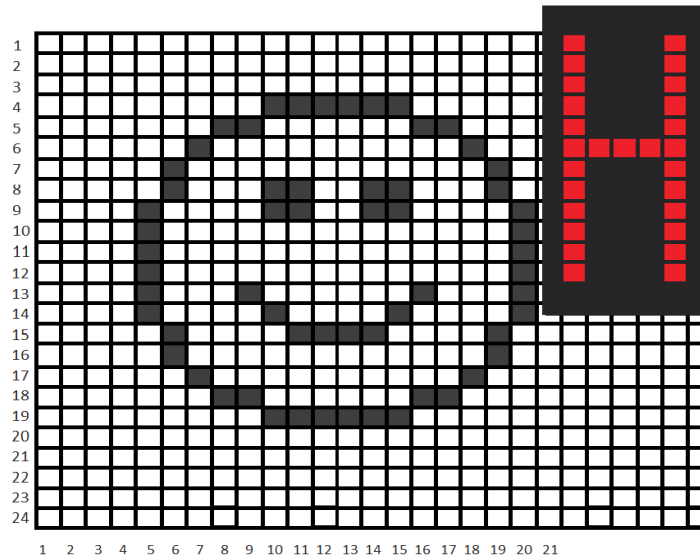
A Figura 1 apresenta uma imagem BMP do tipo binário (abaixo na Figura 1). Esta é uma imagem monocromática que vai indicar justamente quais pixels da imagem terão, ou não um dado (presença ou não de fluido aglutinante). Dessa forma, para que uma imagem seja enviada para o firmware, é então necessário segmentá-la em arrays de pixels correspondente aos cabeçotes do cartucho. Criase assim, um arquivo GCode que coordenará a movimentação da cabeça de impressão ao mesmo tempo que coordenará o acionamento dos bicos por meio do comando M700, adicionado na modificação do firmware Marlin. A geração e segmentação da imagem está sendo paralelamente desenvolvida pelo grupo responsável pelo software RP3 que visa justamente fazer os tratamentos dessas imagens BPM e exportar um arquivo GCode referente a todas as camadas da peça, permitindo também que ajustes de parâmetros sejam realizados.

Figura 1: Demonstração de como uma imagem BMP binária é interpretada.



Fonte:
<https://mitchellkscscomputing.wordpress.com/2015/10/21/how-bitmap-images-are-represented-in-binary/>

Figura 2: Exemplo de uma imagem BPM que poder ser impresso em uma varredura do cabeçote jateador



Fonte: Aatoria Própria

A Figura 2 representa um exemplo do que pode ser impresso em faixas correspondentes à área de varredura do cartucho. No caso específico do cartucho HP c6602, este possui um *array* de 12 injetores, pelo comprimento desejado da impressão. Os pontos vermelhos representam os injetores do cartucho. Sabendo que a resolução deste cartucho é de 96 dpi (*dots per inch*) isso equivale a passos de 265 micrón. Para manter a proporcionalidade, as movimentações devem ser múltiplos deste número.

Para que o *firmware* apresente um resultado semelhante ao da Figura 2 apresentada ele espera receber um comando neste seguinte formato:

```
M700 – P<número_do_cartucho> S<endereço_de_bits>
```

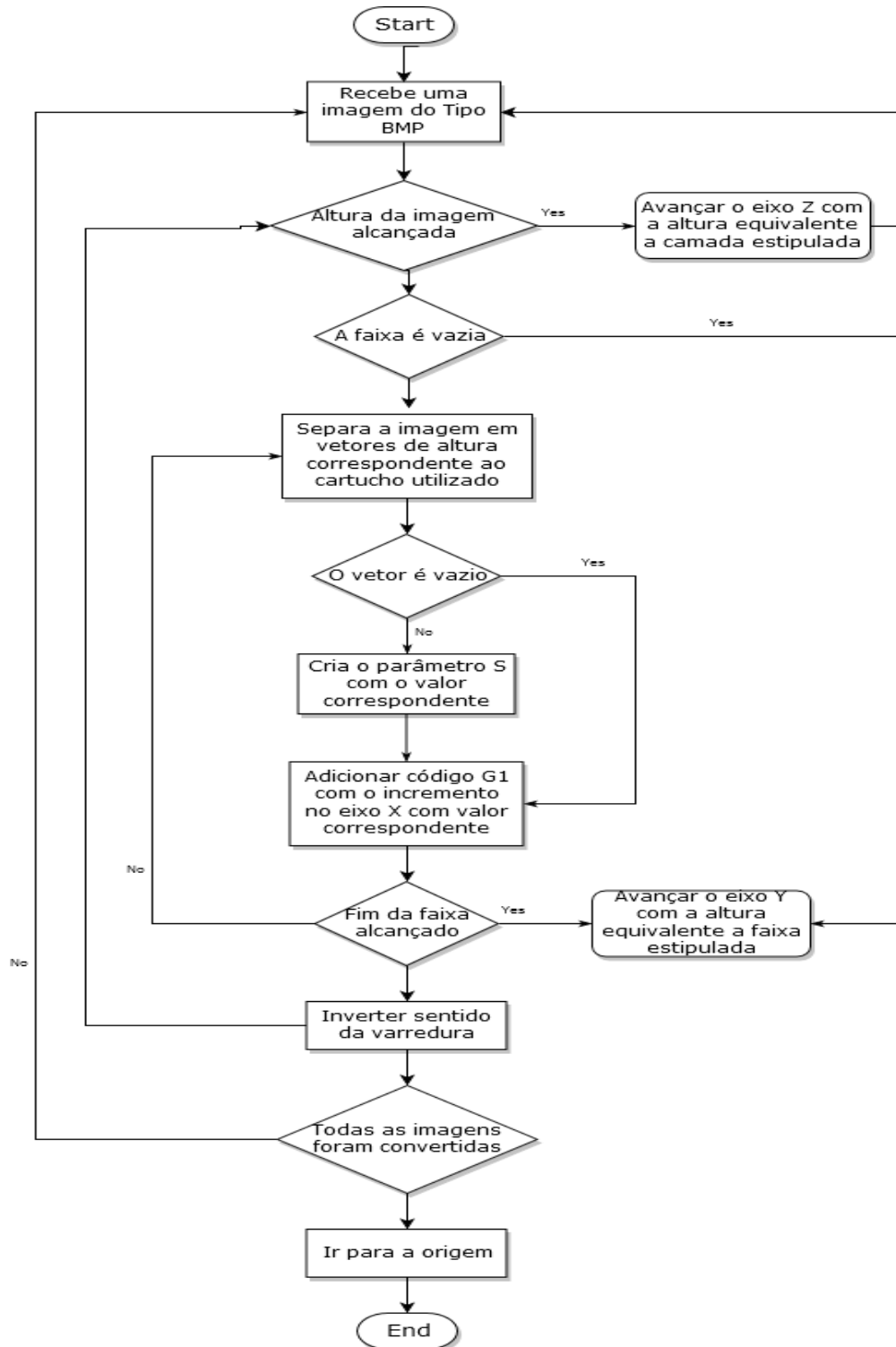
Onde P equivale ao número identificador do cartucho utilizado, uma vez que essa iteração permite o uso de vários cartuchos (por exemplo no uso de impressão colorida). No caso de utilizar somente 1 cartucho, o valor padrão para P, é zero. A variável S corresponde ao valor que vai determinar quais bicos serão acionados. Ele é um número decimal que será convertido para um binário de base 16, que mapeará os bicos acionados. Por exemplo, se o valor de S for igual a 15, isso significa que os quatro primeiros injetores serão acionados, pois convertendo 15 em valor binário obtemos “000000001111”.

RESULTADOS E DISCUSSÕES

No caso da Figura 2 para que consigamos imprimir o “smiley” na ilustração, considerando que ela esteja a 20 mm da borda tanto superior quanto inferior, o *firmware* espera os seguintes comandos em Gcode:

- | | |
|---|---|
| 1. M700 P0 S15 //coluna 5 da figura | 49. G1 X23.71 Y23.2 Z0.4 |
| 2. M400 //limpa o buffer | 50. M700 P0 S3072 //coluna 20 da figura |
| 3. G1 X20.265 Y20 Z0.4 //movimenta o cartucho | 51. M400 |
| 4. M700 P0 S48 //coluna 6 da figura | 52. G1 X23.445 Y23.2 Z0.4 |
| 5. M400 | 53. M700 P0 S768 //coluna 19 da figura |
| 6. G1 X20.53 Y20 Z0.4 | 54. M400 |
| 7. M700 P0 S64 //coluna 7 da figura | 55. G1 X23.18 Y23.2 Z0.4 |
| 8. M400 | 56. M700 P0 S128 //coluna 18 da figura |
| 9. G1 X20.795 Y20 Z0.4 | 57. M400 |
| 10. M700 P0 S128 //coluna 8 da figura | 58. G1 X22.915 Y23.2 Z0.4 |
| 11. M400 | 59. M700 P0 S64 //coluna 17 da figura |
| 12. G1 X21.06 Y20 Z0.4 | 60. M400 |
| 13. M700 P0 S128 //coluna 9 da figura | 61. G1 X22.65 Y23.2 Z0.4 |
| 14. M400 | 62. M700 P0 S2112 //coluna 16 da figura |
| 15. G1 X21.326 Y20 Z0.4 | 63. M400 |
| 16. M700 P0 S280 //coluna 10 da figura | 64. G1 X22.385 Y23.2 Z0.4 |
| 17. M400 | 65. M700 P0 S1056 //coluna 15 da figura |
| 18. G1 X21.325 Y20 Z0.4 | 66. M400 |
| 19. M700 P0 S280 //coluna 11 da figura | 67. G1 X22.12 Y23.2 Z0.4 |
| 20. M400 | 68. M700 P0 S544 //coluna 14 da figura |
| 21. G1 X21.59 Y20 Z0.4 | 69. M400 |
| 22. M700 P0 S256 //coluna 12 da figura | 70. G1 X21.855 Y23.2 Z0.4 |
| 23. M400 | 71. M700 P0 S2544 //coluna 13 da figura |
| 24. G1 X21.855 Y20 Z0.4 | 72. M400 |
| 25. M700 P0 S256 //coluna 13 da figura | 73. G1 X21.59 Y23.2 Z0.4 |
| 26. M400 | 74. M700 P0 S544 //coluna 12 da figura |
| 27. G1 X22.12 Y20 Z0.4 | 75. M400 |
| 28. M700 P0 S280 //coluna 14 da figura | 76. G1 X21.325 Y23.2 Z0.4 |
| 29. M400 | 77. M700 P0 S544 //coluna 11 da figura |
| 30. G1 X22.385 Y20 Z0.4 | 78. M400 |
| 31. M700 P0 S280 //coluna 15 da figura | 79. G1 X21.326 Y23.2 Z0.4 |
| 32. M400 | 80. M700 P0 S1056 //coluna 10 da figura |
| 33. G1 X22.65 Y20 Z0.4 | 81. M400 |
| 34. M700 P0 S128 //coluna 16 da figura | 82. G1 X21.06 Y23.2 Z0.4 |
| 35. M400 | 83. M700 P0 S2112 //coluna 9 da figura |
| 36. G1 X22.915 Y20 Z0.4 | 84. M400 |
| 37. M700 P0 S128 //coluna 17 da figura | 85. G1 X20.795 Y23.2 Z0.4 |
| 38. M400 | 86. M700 P0 S64 //coluna 8 da figura |
| 39. G1 X23.18 Y20 Z0.4 | 87. M400 |
| 40. M700 P0 S64 //coluna 18 da figura | 88. G1 X20.53 Y23.2 Z0.4 |
| 41. M400 | 89. M700 P0 S128 //coluna 7 da figura |
| 42. G1 X23.445 Y20 Z0.4 | 90. M400 |
| 43. M700 P0 S48 //coluna 19 da figura | 91. G1 X20.265 Y23.2 Z0.4 |
| 44. M400 | 92. M700 P0 S768 //coluna 6 da figura |
| 45. G1 X23.71 Y20 Z0.4 | 93. M400 |
| 46. M700 P0 S15 //coluna 20 da figura | 94. G1 X20 Y23.2 Z0.4 |
| 47. M400 | 95. M700 P0 S3072 //coluna 5 da figura |
| 48. G1 X25 Y23.2 Z0.4 //movimenta o cartucho | 96. M400 |

Figura 3: Fluxograma que descreve os requerimentos para a elaboração de um código Gcode a partir de diversas imagens do tipo BMP



Fonte: Autoria Própria

O código Gcode acima foi elaborado respeitando as condições preestabelecidas pelo fluxograma apresentado na Figura 3 acima. Tal lógica representa basicamente uma varredura matricial, com uma filtragem que verifica a presença ou não de uma linha ou coluna completamente nula, com a finalidade de ignorá-la, fazendo com que o cabeçote somente varra as partes que possuam ao menos um pixel em sua coluna, otimizando assim todo o processo.

No código Gcode, a distância no eixo X vai sendo incrementado em 0.265mm que é a resolução mínima do cartucho equivalente em milímetros, e uma vez que o array foi completado o eixo Y terá que se movimentar 3.18mm que equivale a espessura de uma linha completa de 12 pontos. Como pode ser verificada no código Gcode, para fins de otimização de trajetória, os sentidos de movimentação do eixo X foram intercalados. Isso significa que enquanto que no primeiro array, o cabeçote se movimentou da esquerda para a direita. Após terminado, ele ajustou sua posição Y, e fez o segundo array da direita para a esquerda, em sentido de zigue-zague.

Entretanto, durante a impressão de uma peça tridimensional, o que mudará efetivamente é a quantidade de vezes que este processo será repetido. Isso porque a peça em questão será fatiada e cada fatia dessa peça será exportada como uma imagem BMP distinta, gerando um único código Gcode, com vários processos de varredura. A distinção é que ao término de uma camada a impressora efetuara todo o processo de realimentação de pó, ajustará a medida no eixo Z de acordo com a distância previamente estabelecida, e repetirá toda a movimentação para as camadas posteriores.

CONCLUSÕES

Neste trabalho foi proposto um padrão para transformar uma imagem BMP em uma série de instruções em um formato GCode que permitem o processamento através do princípio de jateamento de aglutinante.

Os aspectos referentes ao posicionamento do cabeçote no plano XY e a discretização da imagem em faixas de varredura foram detalhados e estão prontos para serem analisados.

É importante frisar que tanto o equipamento quanto a parte de discretização de imagens ainda não foram finalizados. Desta forma, o padrão proposto não pode ser testado efetivamente. Na continuidade do projeto esta implementação será realizada e testada.

Além do firmware Marlin, este projeto também faz uso de outro projeto aberto, a Inkshild, que apesar de limitado atende majoritariamente os requisitos da proposta de criar um aparato de impressão 3D. Ainda não houve um projeto documentado que mostrasse a interação entre a plataforma Inkshield e o firmware Marlin. Apesar de ser teoricamente possível, e ser uma possibilidade explorada por entusiastas na área, se tal proposta se provar possível, este processo pode se tornar uma forma fácil, flexível e intuitiva de se construir uma impressora 3DP aberta.

AGRADECIMENTOS

Agradeço ao professor Neri Volpato pela oportunidade de estar neste projeto e pela orientação no projeto regido pelo centro de inovação tecnológica, a todo o pessoal que faz parte do Núcleo de Manufatura Aditiva e Ferramental (NUFER) aos quais pude aprimorar meus conhecimentos, auxiliando e aprendendo por meio de experiência.

REFERÊNCIAS

VOLPATO, N. Processos de AM por extrusão de material. In: VOLPATO, N. (Editor). Manufatura Aditiva: Tecnologias e aplicações da impressão 3D. São Paulo: Blucher, 2017.

TYLER, Harrison. Open Source Possibilities for Inkjet 3D Printing. In: ADVANCED MANUFACTURING RESEARCH INSTITUTE, 2014, Rice University. **HP C6602 Characterization and Application** [...]. [S. l.: s. n.], 2014. Disponível em: <https://static1.squarespace.com/static/576574baf5e23128b321942a/t/57743efa9de4bbb70589c1a2/1467236300874/Open+Source+Possibilities+for+Inkjet+3D+Printing+>. Acesso em: 15 jun. 2019.

LEWIS, Nicholas C. **Inkshield**. [S. l.], 2012. Disponível em: <http://nicholasclewis.com/projects/inkshield/>. Acesso em: 15 jun. 2019.

YTEC3D, Site. **Installing the firmware**. [S. l.], 2015. Disponível em: <http://ytec3d.com/plan-b-firmware/>. Acesso em: 15 jun. 2019.

KELLY, Steve. **Marlin 3D Printer Firmware**. [S. l.], 2013. Disponível em: <https://github.com/sjkelly/Marlin/tree/hpc6602>. Acesso em: 15 jun. 2019.

WOODFORD, Chris. **How Inkjet Printer Works**. [S. l.], 2018. Disponível em: <https://www.explainthatstuff.com/inkjetprinters.html> Acesso em: 2 jul. 2019.