EMO.ji

Uma nova linguagem de programação para gerações futuras

Sabrina SS
@Insper

Evolução Tabela Emoji

SoftBank 1997

Vendor: SoftBank
Version: 1997
Release Date: Nov. 1, 1997
Emojis: 90

First launched 1997-11-01 on the SkyWalker service which was launched in Japan on the DP-211SW mobile phone. May have been the first emoji set in existence. More about this set.

Unicode Version 1.1

Unicode 1.1 were pre-dated emoji inclus
characters primarily sourced from the Za

The following characters were added to U
status in the years 2010-2015.

- Smiling Face
- Frowning Face
- Skull and Crossbones
- Heavy Heart Exclamation
- Red Heart
- Victory Hand
- Index Pointing Up
- Writing Hand
- Hot Springs
- Airplane
- Hourglass Done
- Watch
- Sun
- Cloud
- Umbrella
- Snowflake
- Snowman
- Comet
- Spade Suit
- Heart Suit
- Diamond Suit
- Club Suit
- Chess Pawn
- Telephone
- Keyboard
- Envelope

Unicode Version 6.0

Released in October 2010, Unicode 6.0 was th
to support emoji.

Some characters in prior versions of Unicode v
emoji presentation. This release was the first t
purpose of compatability with emojis being use

- Grinning Face With Big Eyes
- Grinning Face With Smiling Eyes
- Beaming Face With Smiling Eyes
- Grinning Squinting Face
- Grinning Face With Sweat
- Face With Tears of Joy
- Winking Face
- Smiling Face With Smiling Eyes
- Smiling Face With Halo
- Smiling Face With Heart-Eyes
- Face Blowing a Kiss
- Kissing Face With Closed Eyes
- Face Savoring Food
- Winking Face With Tongue
- Squinting Face With Tongue
- Neutral Face
- Face Without Mouth
- Smirking Face
- Unamused Face
- Relieved Face
- Pensive Face
- Sleepy Face
- Face With Medical Mask
- Dizzy Face
- Smiling Face With Sunglasses
- Astonished Face

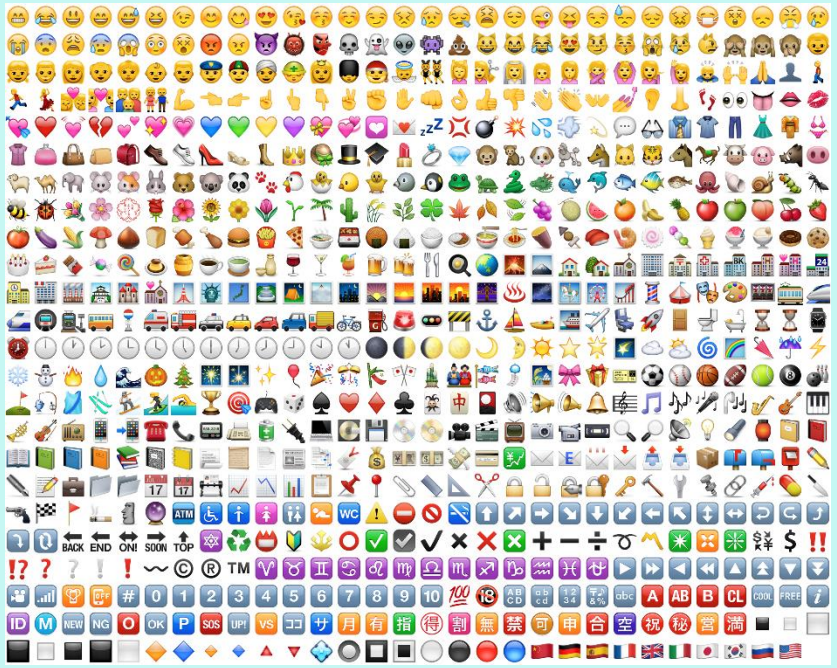
Unicode Version 10.0

Unicode 10.0 is the version of the Unicode S
new characters were included with this upda

Only new emoji code points are listed on thi
Emoji 5.0 which included all 239 emoji addit
gender.

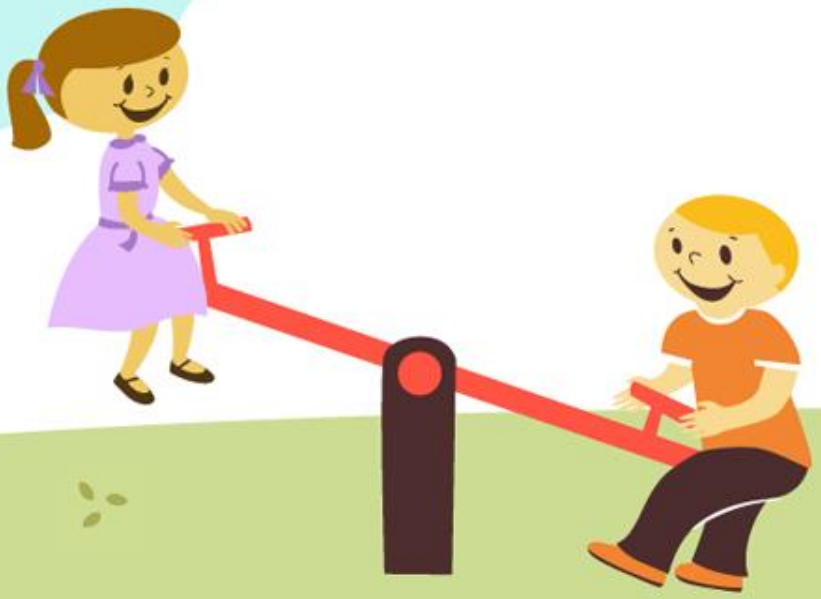
- Emoji 5.0 list
- Unicode 10.0.0 Specification

- Star-Struck
- Zany Face
- Face With Hand Over Mouth
- Shushing Face
- Face With Raised Eyebrow
- Face Vomiting
- Exploding Head
- Face With Monocle
- Face With Symbols on Mouth
- Orange Heart
- Love-You Gesture
- Palms Up Together
- Brain
- Child
- Person
- Man: Beard
- Older Person
- Woman With Headscarf
- Breast-Feeding
- Mage



Um crescimento de 9400% no numero de emojis entre 1997 e 2019

A comunicação por símbolos é natural ao ser humano e surgiu bem antes das palavras




While
Mientras
Enquanto
Pendant
Podczas

Var < 3

var_2

Como aumento do uso e número de emojis no vocabulário infantil, é de se esperar uma maior familiaridade deste público com os símbolos.

Dessa forma, uma linguagem universal, sem barreiras linguísticas, pode vir a nascer.

É óbvio que a transição é lenta, principalmente porque não existem emojis perfeitos para todas as ocasiões da linguagem de programação *ainda.* Contudo, tudo indica que virão a surgir.

Valor de Mercado: como vai funcionar a programação pelos usuários?

A ideia é que seja uma linguagem de programação de fácil acesso em celulares, tablets e aparelhos com teclado nativo de Emoji. Uma vez que a correção automática de texto e a sugestão de mensagens já usam em consideração os emojis em seu comportamento, fica ainda mais rápido para se programar assim pelos smartphones.

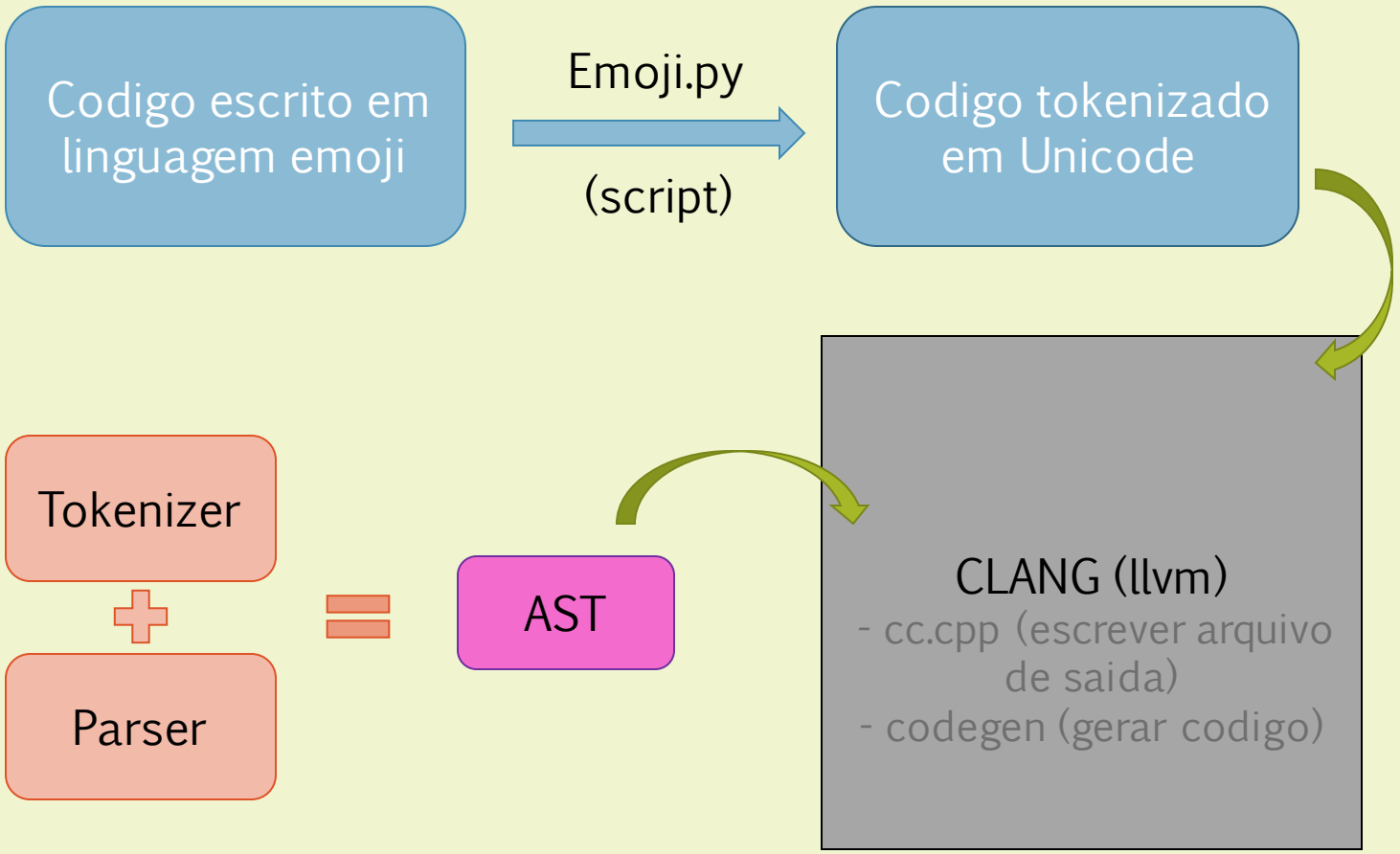
Mas, principalmente, deve-se olhar para um futuro onde a programação é muito mais intrínseca no dia a dia de todas as pessoas, onde mandar um e-mail é quase tão banal quando escrever códigos em seu bolso ou enquanto espera o ônibus.

Neste paradigma, a facilidade de programar pelo celular somado há uma geração habituada pelos emojis cria o ambiente perfeito para EMO.ji

Parte Técnica:

- Ferramentas
- Geração de Código
- Compilação

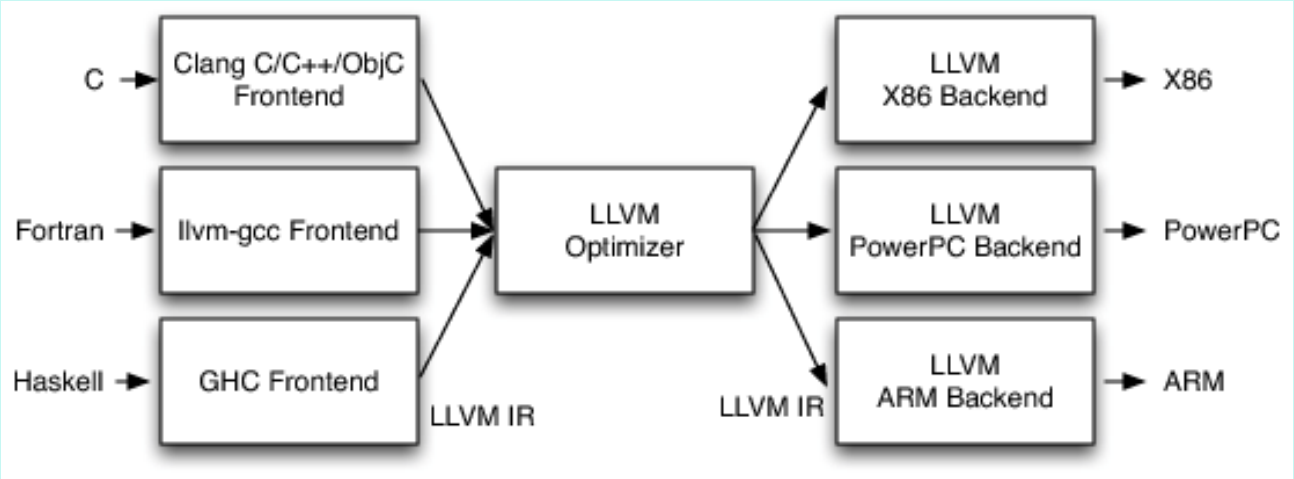
Etapas de Compilação



Requisitos

- Flex
- Bison
- C Lang v3.9
- Python 3
  - Emoji module

Parte Técnica:



- O Frontend parseia a linguagem original e retorna código em Intermediate Representation (IR) do llvm.
- O *otimizador* faz uma otimização magica - como uma caixa preta - como remover código lixo e otimiza variáveis.
- O backend pega o IR e produz código de máquina para uma CPU específica

Parte Técnica:

Antes de prosseguir com os exemplos, eles foram rodados com lli, uma ferramenta do llvm que já interpreta o código IR e o executa.

O código IR é como um assembly, então foi escolhido não os mostrar

```
main(){

    ("%s\n","FizzBuzz Example:\n" );

    i;
    tres;
    cinco;

    (i=1; i<=100; i++)
    {
        tres = (i - (i /  3 * 3));
        cinco = (i - (i /  5 * 5));


        ((tres == 0) & (cinco == 0)){
            ("number = %d FizzBuzz\n", i);
    }
        (tres == 0){
            ("number = %d Fizz\n", i);
    }
        (cinco == 0){
            ("number = %d Buzz\n", i);
    }
        {
            ("number = %d\n",i);
    }
}

    0;
}
```

```
main(){

    ("%s\n","FizzBuzz Example:\n" );

    i;
    tres;
    cinco;

    (i=1; i<=100; i++)
    {
        tres = (i - (i / 3 * 3));
        cinco = (i - (i / 5 * 5));


        ((tres == 0) & (cinco == 0)){
            ("number = %d FizzBuzz\n", i);
        }
        (tres == 0){
            ("number = %d Fizz\n", i);
        }
        (cinco == 0){
            ("number = %d Buzz\n", i);
        }
        {
            ("number = %d\n",i);
        }
    }

    0;
}
```

```
FizzBuzz Example:

number = 1
number = 2
number = 3 Fizz
number = 4
number = 5 Buzz
number = 6 Fizz
number = 7
number = 8
number = 9 Fizz
number = 10 Buzz
number = 11
number = 12 Fizz
number = 13
number = 14
number = 15 FizzBuzz
number = 16
number = 17
number = 18 Fizz
number = 19
number = 20 Buzz
number = 21 Fizz
number = 22
number = 23
number = 24 Fizz
number = 25 Buzz
number = 26
number = 27 Fizz
number = 28
number = 29
number = 30 FizzBuzz
number = 31
number = 32
number = 33 Fizz
number = 34
number = 35 Buzz
number = 36 Fizz
number = 37
number = 38
number = 39 Fizz
number = 40 Buzz
number = 41
number = 42 Fizz
number = 43
number = 44
number = 45 FizzBuzz
number = 46
number = 47
number = 48 Fizz
number = 49
number = 50 Buzz
number = 51 Fizz
```

```
number = 52
number = 53
number = 54 Fizz
number = 55 Buzz
number = 56
number = 57 Fizz
number = 58
number = 59
number = 60 FizzBuzz
number = 61
number = 62
number = 63 Fizz
number = 64
number = 65 Buzz
number = 66 Fizz
number = 67
number = 68
number = 69 Fizz
number = 70 Buzz
number = 71
number = 72 Fizz
number = 73
number = 74
number = 75 FizzBuzz
number = 76
number = 77
number = 78 Fizz
number = 79
number = 80 Buzz
number = 81 Fizz
number = 82
number = 83
number = 84 Fizz
number = 85 Buzz
number = 86
number = 87 Fizz
number = 88
number = 89
number = 90 FizzBuzz
number = 91
number = 92
number = 93 Fizz
number = 94
number = 95 Buzz
number = 96 Fizz
number = 97
number = 98
number = 99 Fizz
number = 100 Buzz
sabrina@sabrina-Inspi
```

```
main(){

    ("%s\n","Fibonacci:\n" );

    n = 6;

    first = 0;
    second = 1;
    next = 0;
    c = 0;

    (c = 0; c < n; c++){
        (c <= 1){
            next = c;
        }
         {
            next = first + second;
            first = second;
            second = next;
        }
        ("%d\n", next);
    }

    0;
}
```

```
Fibonacci:

0
1
1
2
3
5
sabrina@sabrina-Inspiron-15-5578:~/GitHub/EMO-ji$
```

```
main(){

    ("%s\n","Factorial Example:\n" );


    i;
    fact = 1;
   i = 5;
    n = i;


/*while loop will calculate factorial*/
    (i>=1){
        fact=fact*i;
        i--;
    }
        ("The factorial of given number %d is %d\n",n,fact);
        0;
}
```

```
Factorial Example:

The factorial of given number 5 is 120
sabrina@sabrina-Inspiron-15-5578:~/GitHub/EMO-ji$
```

```
🔢 soma(🔢 a, 🔢 b);
🔢 sub(🔢 a, 🔢 b);
🔢 mult(🔢 a, 🔢 b);
🔢 div(🔢 a, 🔢 b);

🔢 main(){

    🖨("%s\n","Function Example:\n" );

    🖨("Soma(2,3) is: %d\n", soma(2,3));
    🖨("Sub(2,3) is: %d\n", sub(2,3));
    🖨("Mult(2,3) is: %d\n", mult(2,3));
    🖨("Div(10,2) is: %d\n", div(10,2));

    🕹 0;
}

🔢 soma(🔢 a, 🔢 b){
    🕹 a+b;
}

🔢 sub(🔢 a, 🔢 b){
    🕹 a-b;
}

🔢 mult(🔢 a, 🔢 b){
    🕹 a*b;
}

🔢 div(🔢 a, 🔢 b){
    🕹 a/b;
}
```

```
Function Example:

Soma(2,3) is: 5
Sub(2,3) is: -1
Mult(2,3) is: 6
Div(10,2) is: 5
```

Conclusão

Espera-se que a linguagem EMO.ji seja ao menos destinada à fins educativos.

Abstraindo maiores blocos de código em símbolos e emojis, as crianças podem ter contato com a programação desde cedo, e quem sabe essa não seja a solução para um dia resolverem p=np?

Emojis estão em todo lugar, inclusive no powerpoint --->

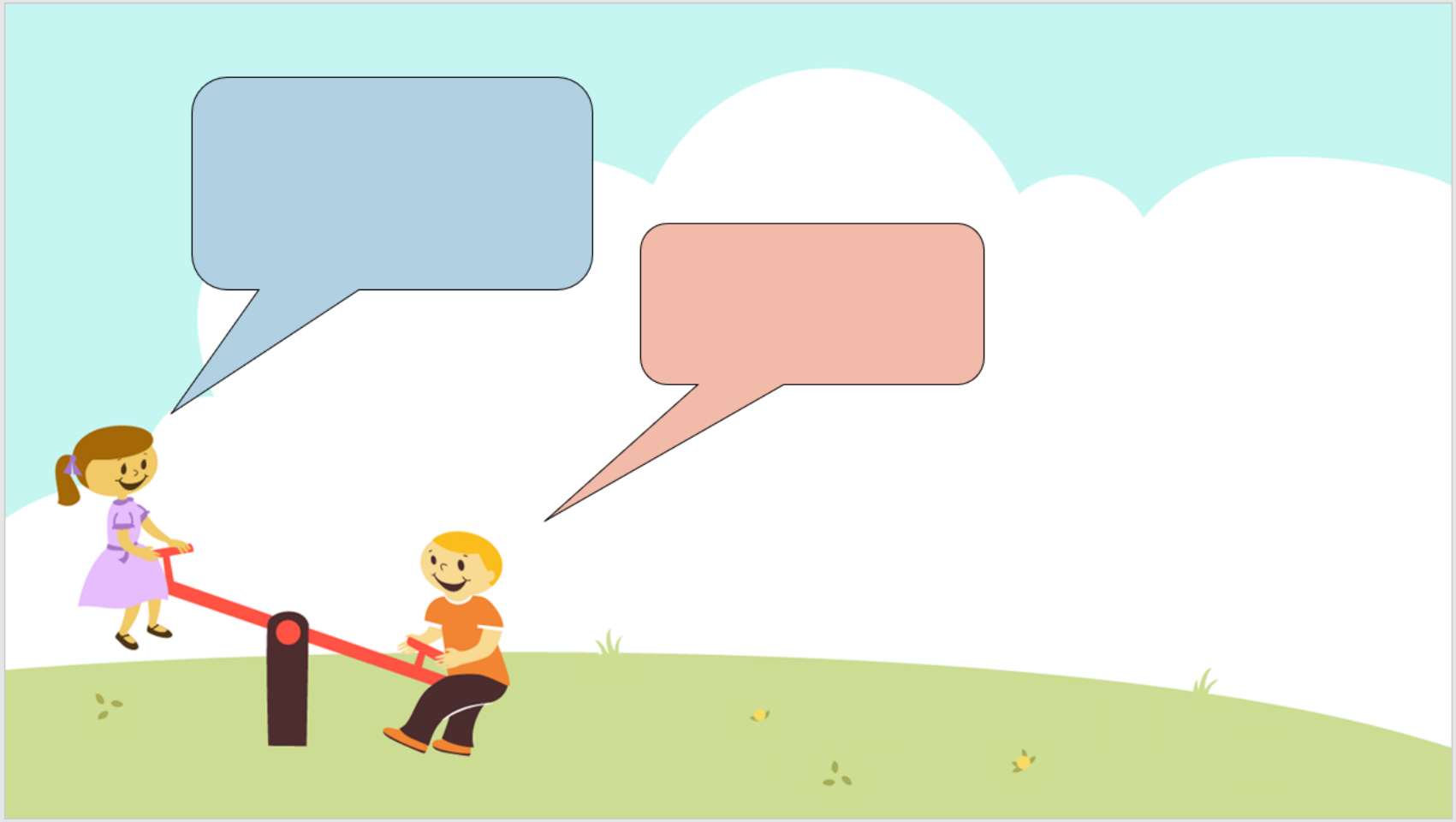
Salvamento Automático | EMOJI | sabrina sampaio simão

Arquivo | Página Inicial | Inserir | Design | Transições | Animações | Apresentação de Slides | Revisão | Exibir | Ajuda | Diga-me o que você deseja fazer | Compartilhar

Novo Slide | Tabela | Imagens | Imagens Online | Instantâneo | Álbum de Fotografias | Formas | Ícones | Modelos 3D | SmartArt | Gráfico | Obter Suplementos | Meus Suplementos | Zoom | Link | Ação | Comentário | Caixa de Texto | Cabeçalho e Rodapé | WordArt | Data e Hora | Número do Slide | Objeto | Equação | Símbolo | Vídeo | Áudio | Gravação de Tela | Emoji Keyboard

Slides | Tabelas | Imagens | Ilustrações | Suplementos | Links | Comentários | Texto | Símbolos | Mídia | Emoji

EMOJI POSSUI TECLADO PADRAO NO POWERPOINT



Emoji Keyboard

Search Emoji

Smileys & People

White Smiling Face

Skin Tone:

TEXT | XS 16px | S 32px | M 64px | L 128px | XL 512px | XXL 1024px

User Manual | Attribution | About | RATE

Slide 3 de 7 | Português (Brasil) | Anotações | 99%

Digite aqui para pesquisar | POR | 18:31 | 21/04/2019