
BIENVENIDOS AL
BOOT CAMP:
INTRODUCCIÓN A LAS
DAPPS EN SOLANA
POR:
HEAVY DUTY
BUILDERS

ESTRUCTURA DEL BOOT CAMP:

- 4 semanas. (Del 05 febrero al 01 marzo)
- 1 clase por semana. (Lunes 19:00hrs UTC)
- 1 reto semanal.
- 1 consultoría a la semana. (Miércoles 19:00hrs UTC)
- discusión del reto semanal. (Viernes 19:00hrs UTC)
- conoce a un builder semanal. (Jueves 19:00hrs UTC)

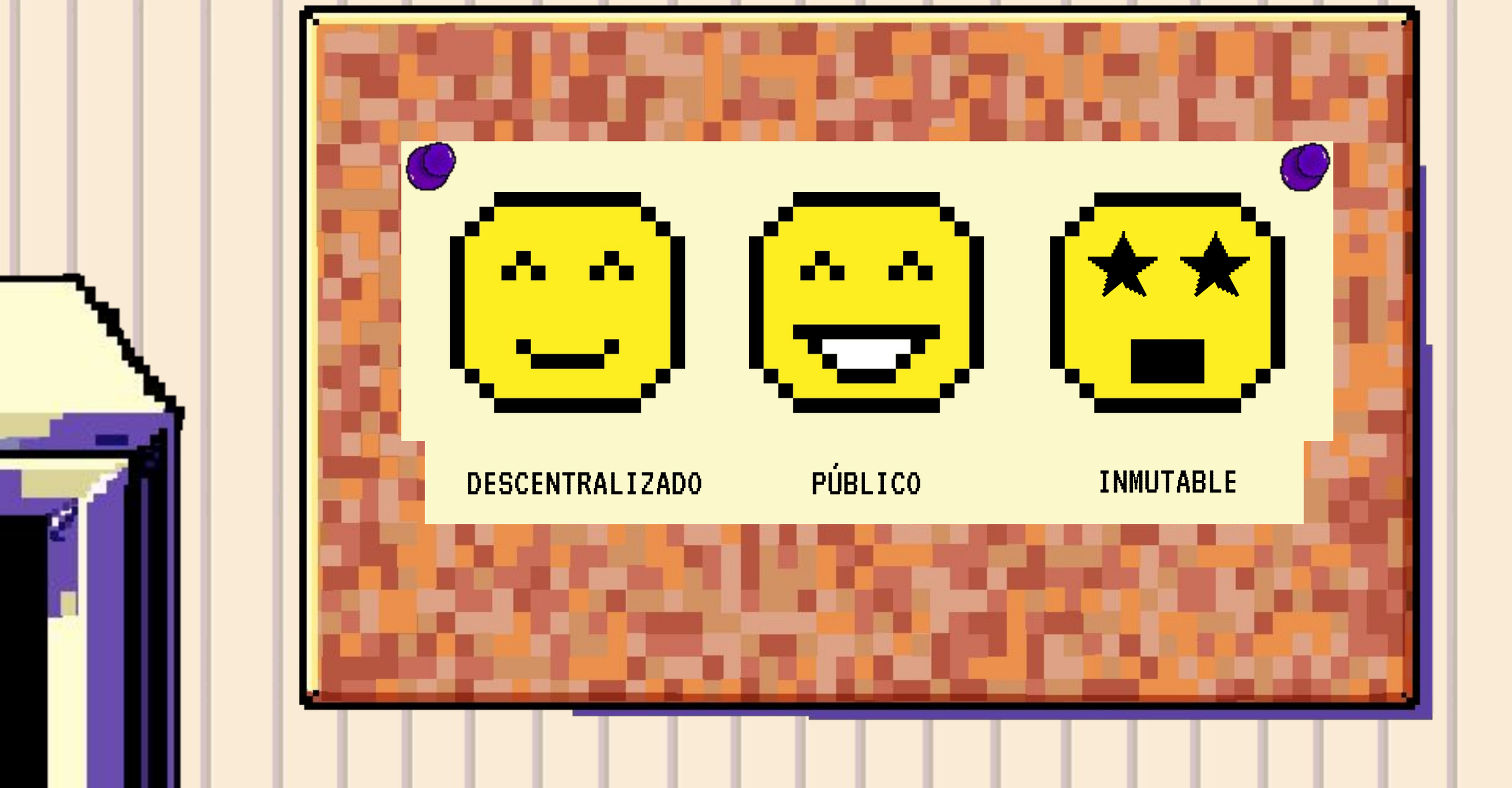
Premios NFT asociados a cada desafío. Habrá recompensas para todos los que completen todos los desafíos.

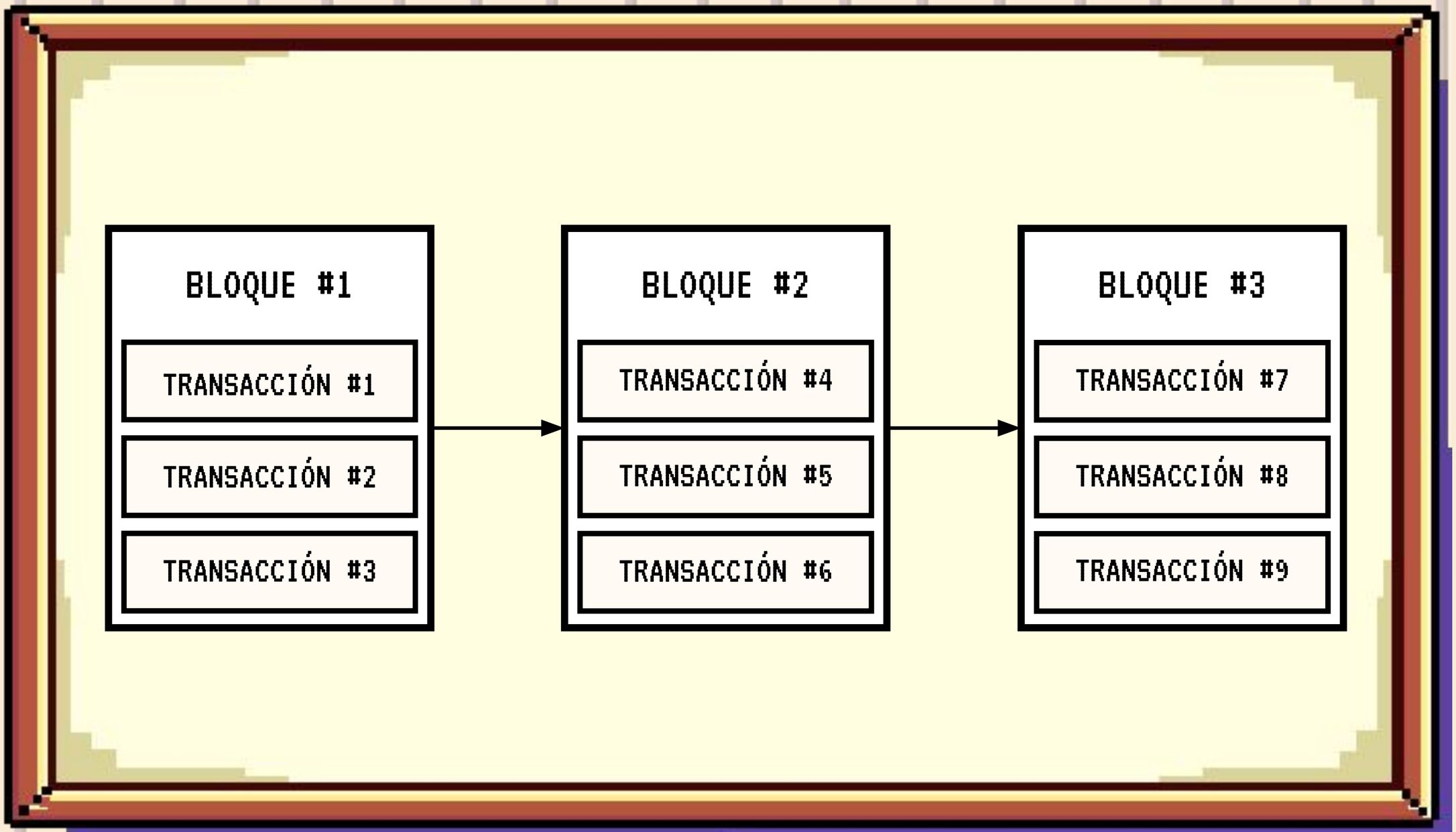
AGENDA DE LA CLASE #1

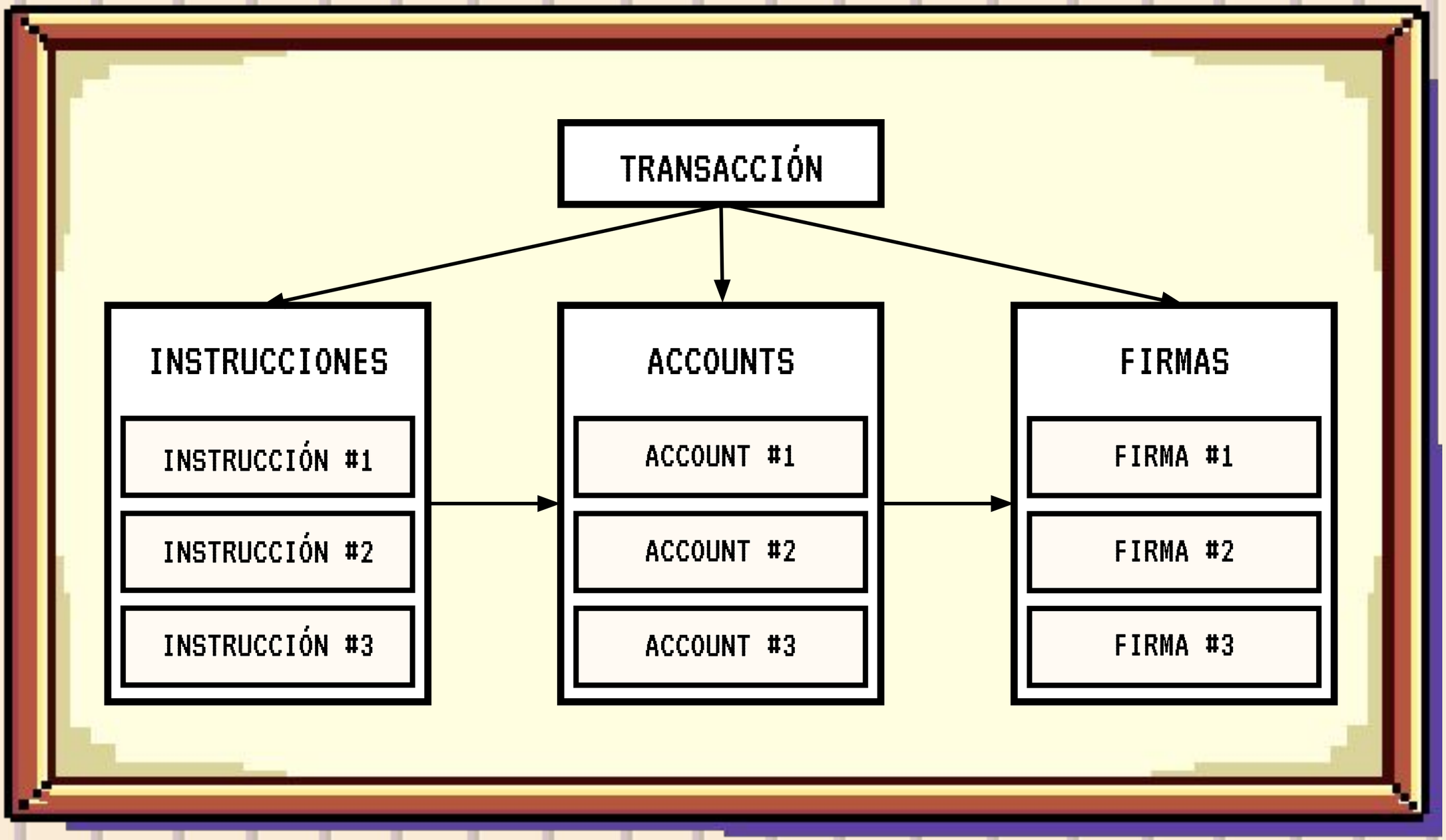
- ¿Qué es Blockchain?
- ¿Qué es una dApp?
- Desarrollando aplicaciones web.
- HTML.
- CSS.
- Javascript.
- Angular.
- Material Design.
- Tailwind CSS.
- Monorepos y Nx.

BLOCKCHAIN

BASE DE DATOS

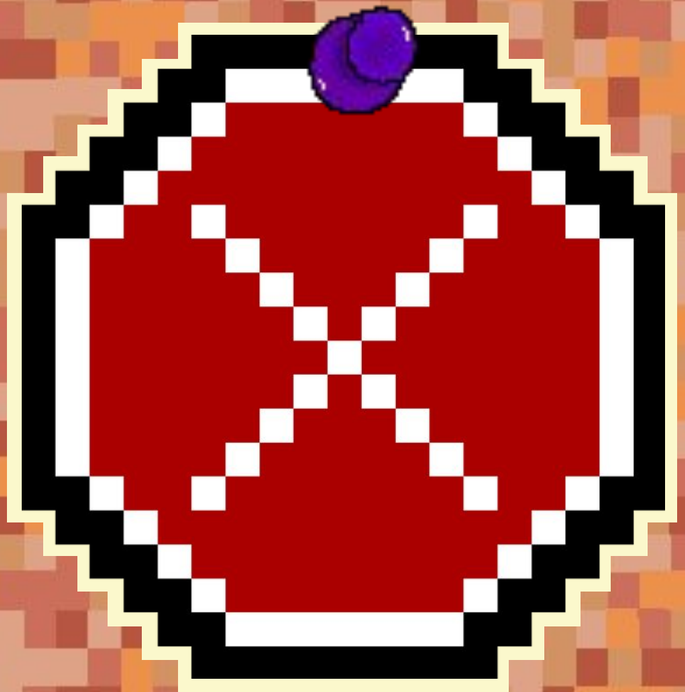

DESCENTRALIZADO
PÚBLICO
INMUTABLE


BLOQUE #1
TRANSACCIÓN #1
TRANSACCIÓN #2
TRANSACCIÓN #3
BLOQUE #2
TRANSACCIÓN #4
TRANSACCIÓN #5
TRANSACCIÓN #6
BLOQUE #3
TRANSACCIÓN #7
TRANSACCIÓN #8
TRANSACCIÓN #9


TRANSACCIÓN
INSTRUCCIONES
INSTRUCCIÓN #1
INSTRUCCIÓN #2
INSTRUCCIÓN #3
ACCOUNTS
ACCOUNT #1
ACCOUNT #2
ACCOUNT #3
FIRMAS
FIRMA #1
FIRMA #2
FIRMA #3

SCAMMER ENVIANDO UNA TRANSACCIÓN FALSA:

LOS VALIDADORES:



QUIZ #1

¿Cuál de las siguientes opciones es una característica de la blockchain?

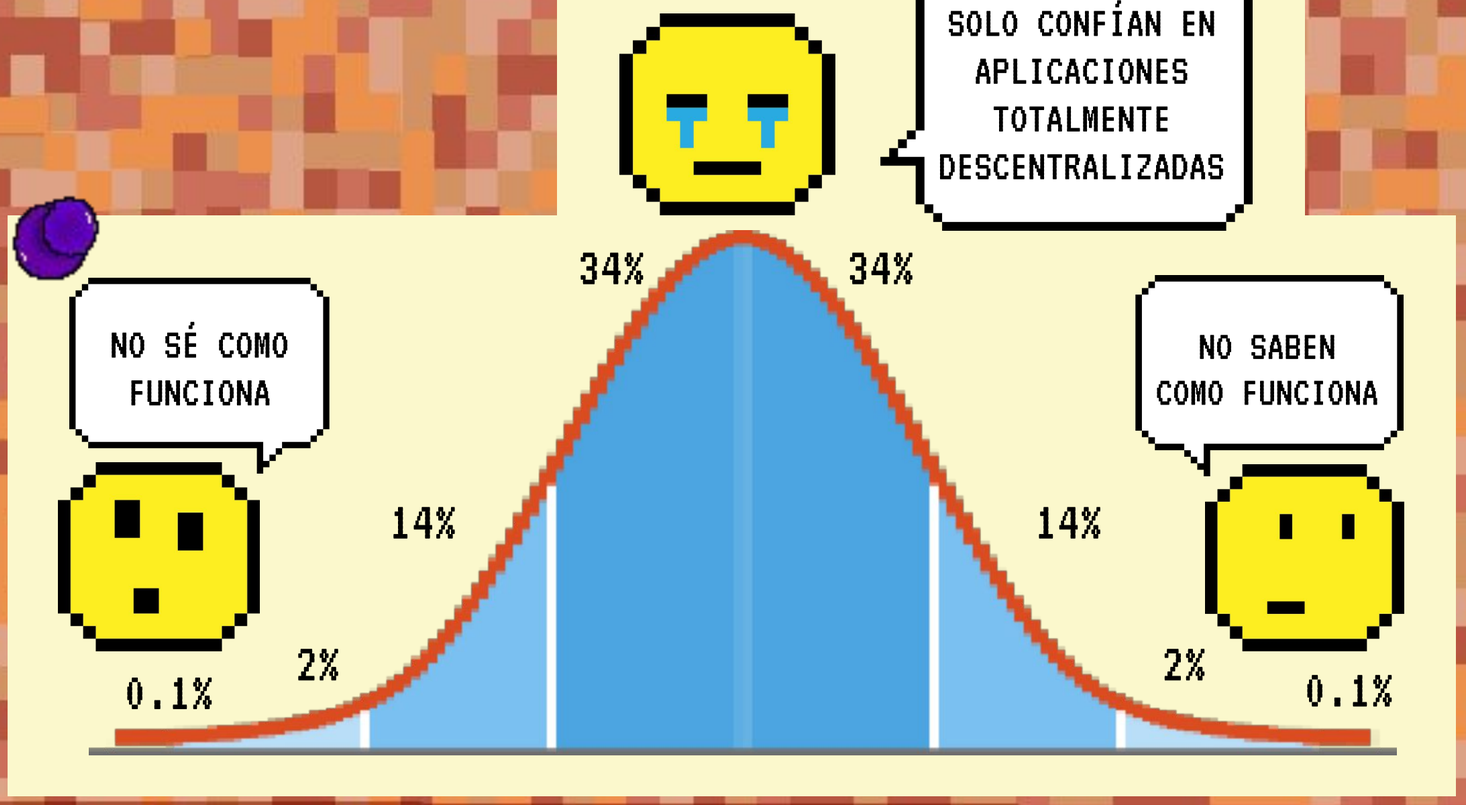
a. Centralizada.

b. Pública.

c. Mutable.

d. Insegura.

DISEÑO DE LA APLICACIÓN



CSS

HTML

JAVASCRIPT

DOM

4

11

WIKIPEDIA
La enciclopedia libre

Portada Discusión Herramientas

Bienvenidos a Wikipedia,

la enciclopedia de contenido libre
que todos pueden editar.

1 927 323 artículos en español.

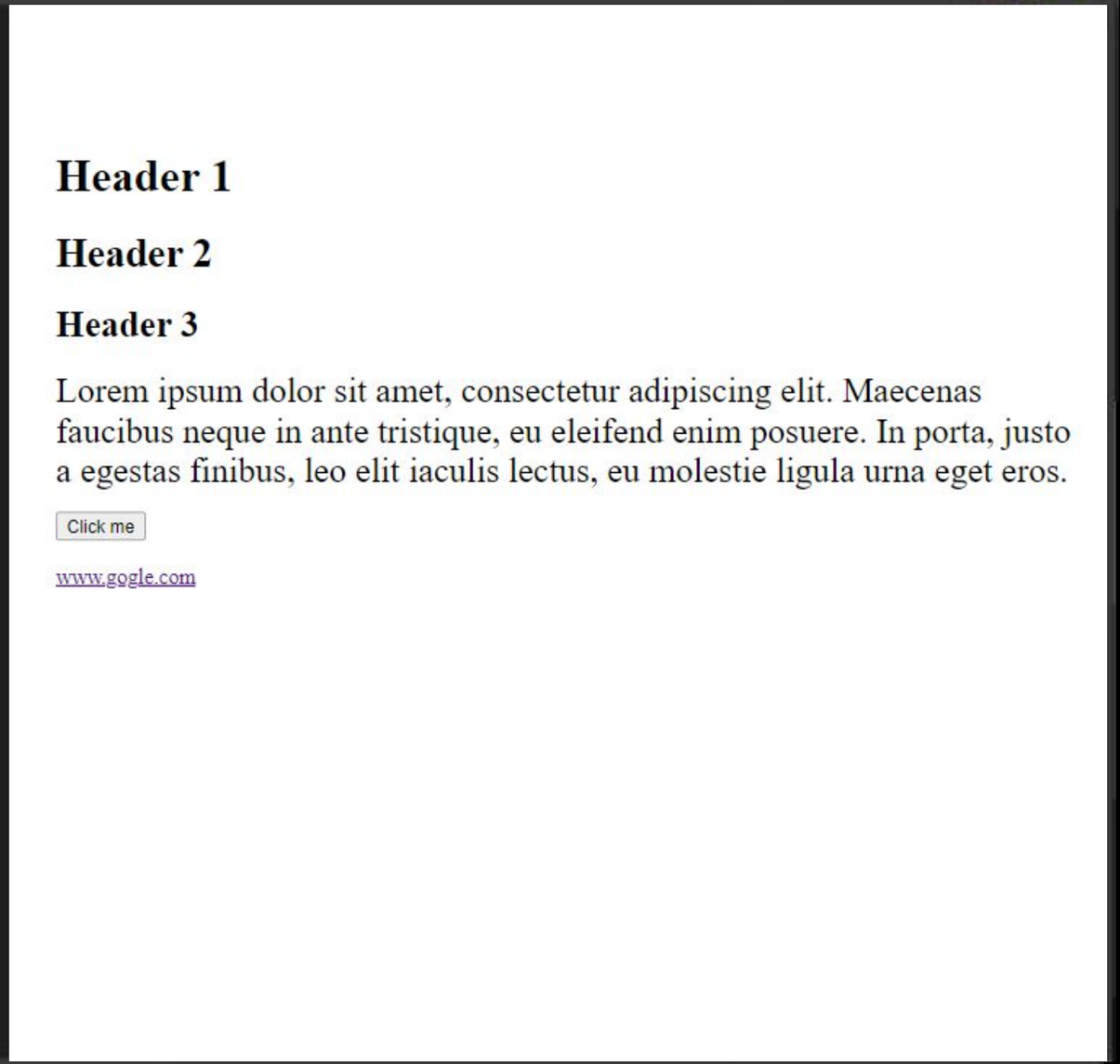
Artículo destacado

Segundo arbitraje de Viena

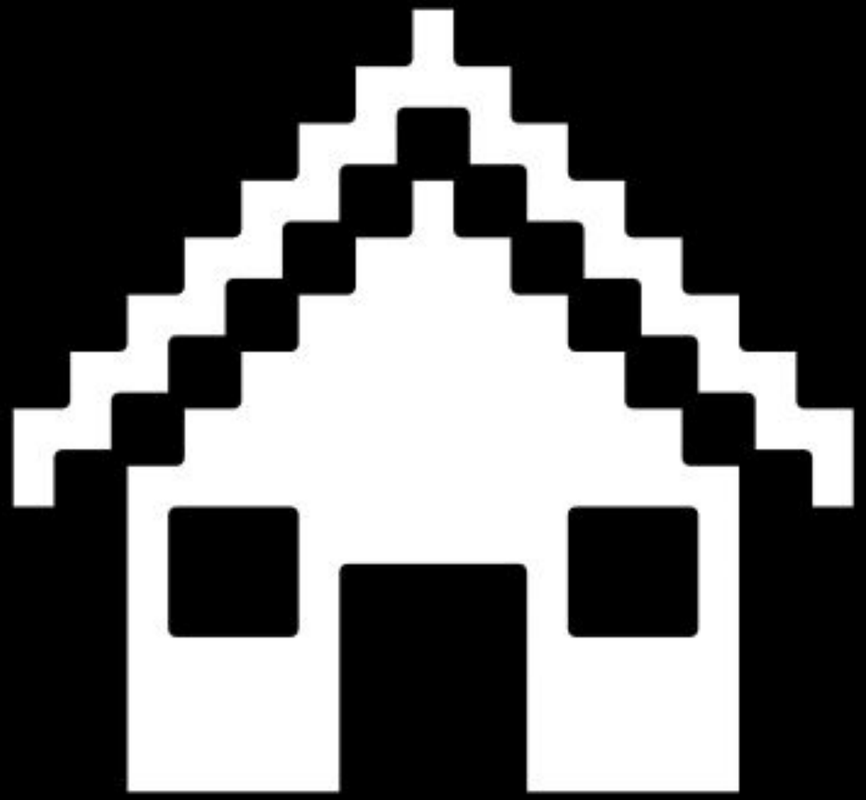
El **Segundo arbitraje de Viena** consistió en un acuerdo territorial alcanzado por mediación y presión alemana entre Hungría y Rumanía que dividió entre ellas la región de Transilvania, perdida por la primera tras la Primera Guerra Mundial en el Tratado de Trianon y que había pertenecido a

main.html

```html
<h1> Header 1 </h1>
<h2> Header 2 </h2>
<h3> Header 3 </h3>

<p>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit.
    Maecenas faucibus neque in ante tristique,
    eu eleifend enim posuere.
    In porta, justo a egestas finibus,
    leo elit iaculis lectus,
    eu molestie ligula urna eget eros.
</p>

<button> Click me </button>

<a href="https:www.google.com"> www.gogle.com </a>
```



HTML



HTML + CCS

main.css

```css
.text {
    font-size: 16px;
}
```

¡Hola Mundo!

main.css

```css
.text {
    font-size: 16px;
    color: red;
}
```

¡Hola Mundo!

main.css

```css
.text {
    font-size: 16px;
    color: red;
    font-style: italic;
}
```

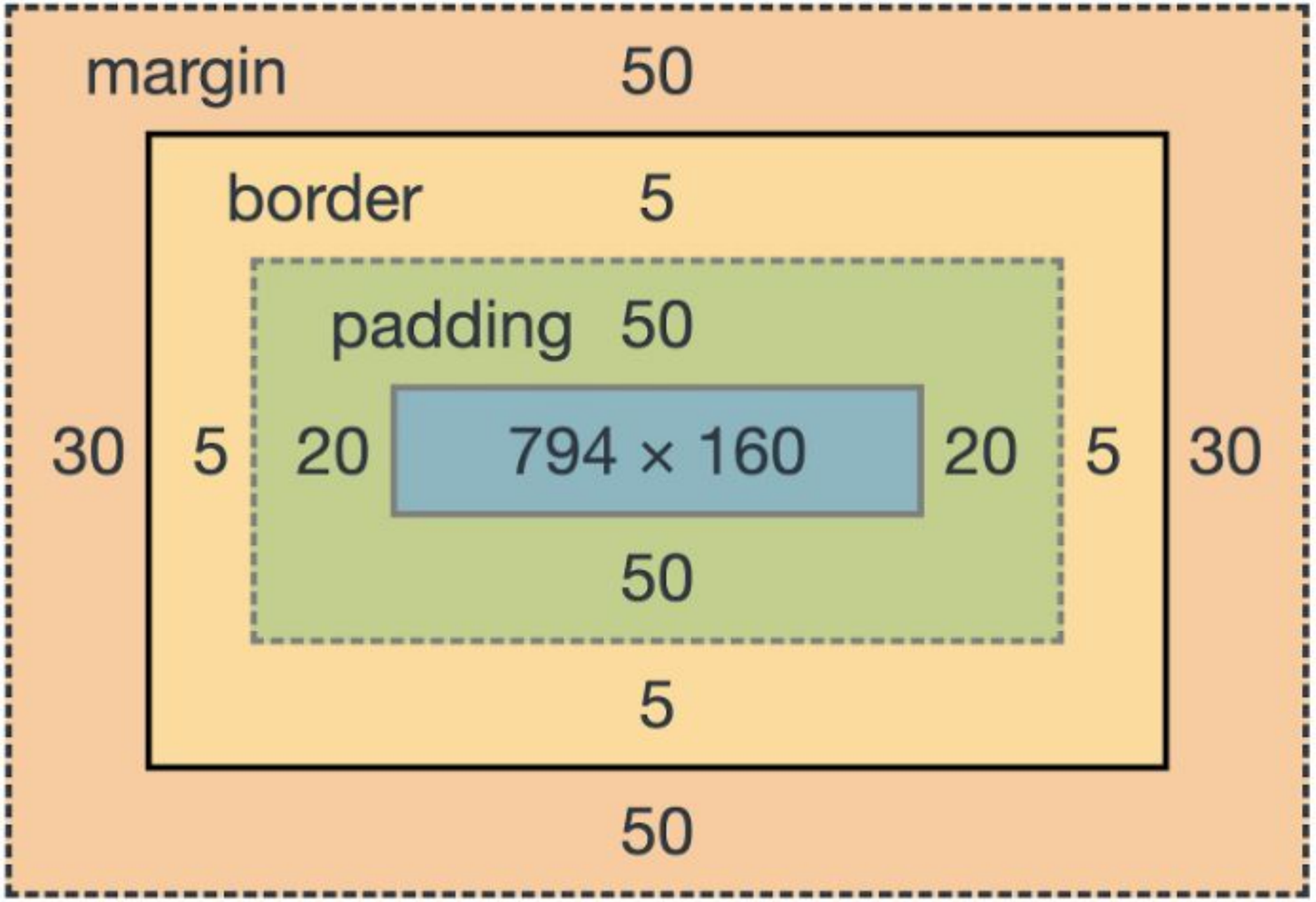
¡Hola Mundo!

main.css

```css
.text {
    font-size: 48px;
    color: red;
    font-style: italic;
}
```

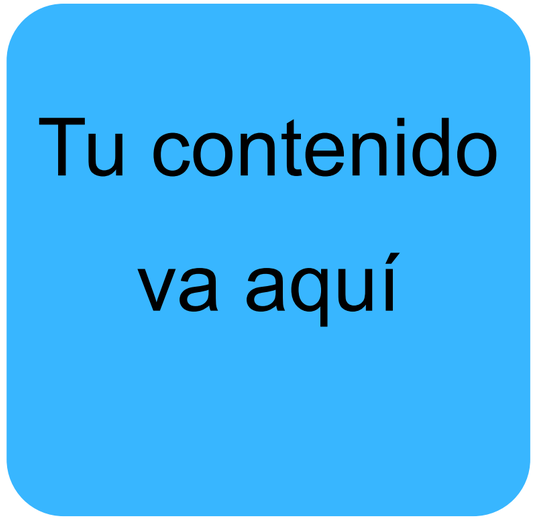
¡Hola

Mundo!


margin
50
border
5
padding
50
30
5
20
794 × 160
20
5
30
50
5
50

main.css

```css
.card {
    background-color: blue;
}
```

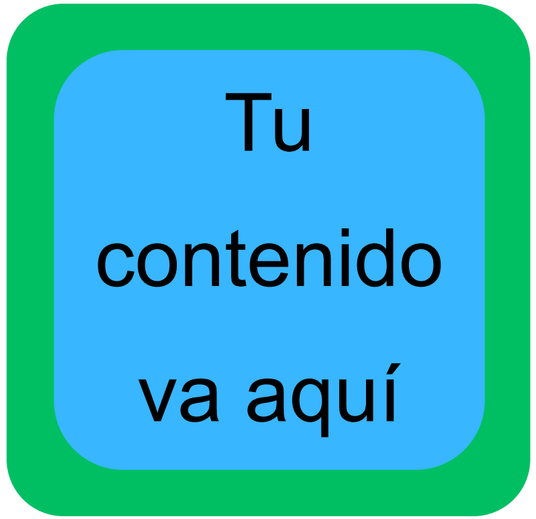


main.css

```css
.card {
    background-color: blue;
    padding: 4px;
}
```

main.css

```css
.card {
    background-color: blue;
    padding: 16px;
}
```



main.css

```css
.card {
    background-color: blue;
    padding: 16px;
    border: 2px solid black;
    margin: 8px 8px 8px 8px;
}
```

QUIZ #2

¿Cuál de las siguientes opciones es parte de los elementos de la web?

a. Blockchain.
b. Validadores.
c. HTML y CSS.
d. Wallets.

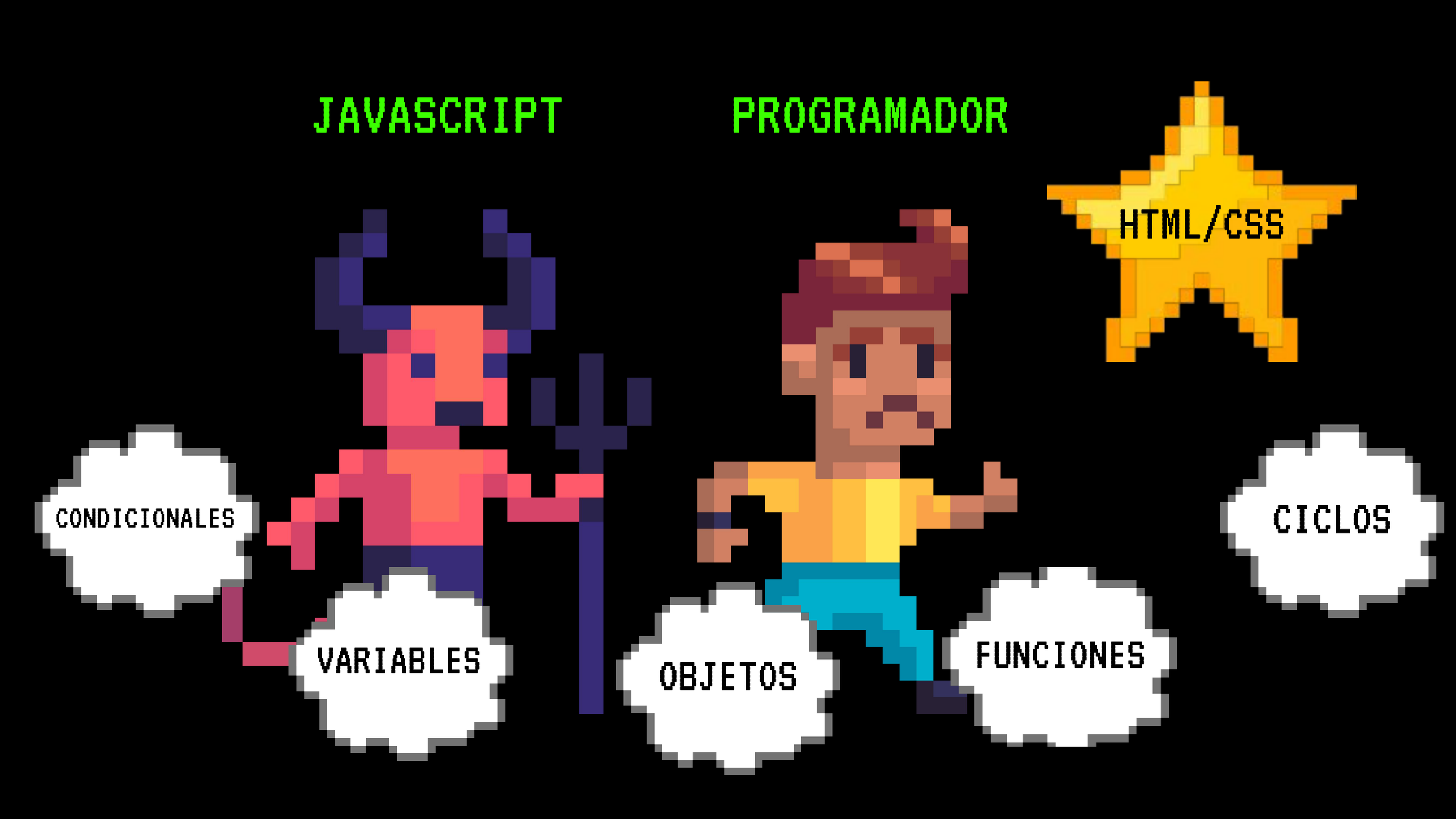
PROGRAMADOR

JAVASCRIPT

PROGRAMADOR

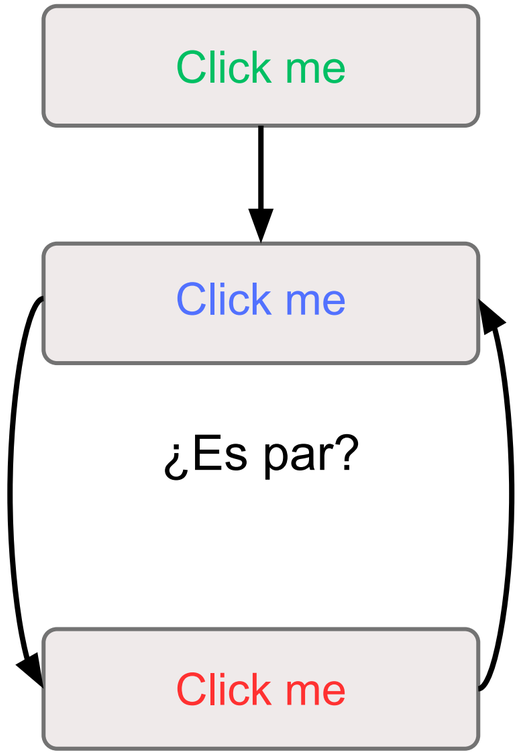
JAVASCRIPT

PROGRAMADOR



main.html

```html
<button id="button-1">Click me</button>

<script>
  let count = 0;
  const buttonElement = document.getElementById('button-1');
  buttonElement.style = 'color: green;';

  buttonElement.addEventListener('click', () => {
    if (count % 2 === 0) {
      buttonElement.style = 'color: blue;';
    } else {
      buttonElement.style = 'color: red;';
    }

    count++;
  });
</script>
```

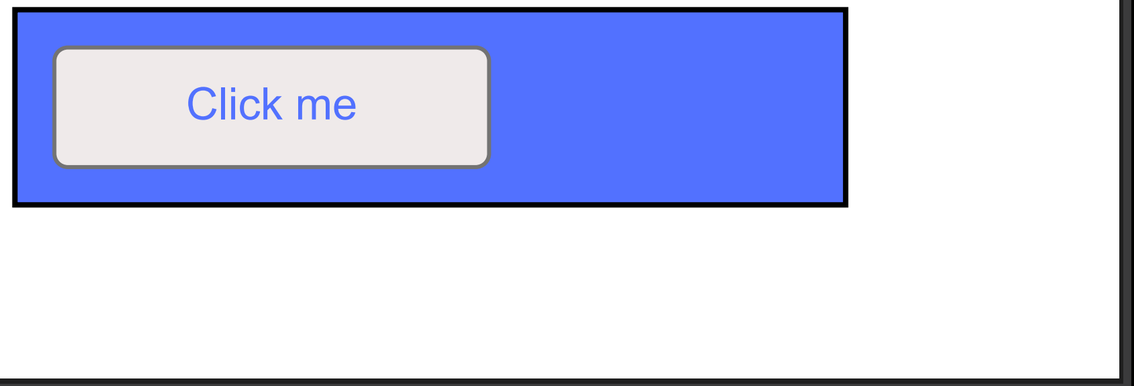


USAR HTML,
CSS Y JS

USAR UN
FRAMEWORK

```ts
import { NgStyle } from '@angular/common';
import { Component } from '@angular/core';

@Component({
  selector: 'app-thing',
  template: `
    <div class="container">
      <button
        [ngStyle]="{ color: color }"
        (click)="onClick()">
        Click me
      </button>
    </div>
  `,
  styles: [
    `
      .container {
        background-color: blue;
        padding: 16px;
        border: 2px solid black;
        margin: 8px 8px 8px 8px;
      }
    `,
  ],
  standalone: true,
  imports: [NgStyle],
})
export class ThingComponent {
  color = 'green';
  count = 0;

  onClick() {
    if (this.count % 2 === 0) {
      this.color = 'blue';
    } else {
      this.color = 'red';
    }

    this.count++;
  }
}
```

```ts
import { Component } from '@angular/core';
import { ThingComponent } from './component';

...
@Component({
  standalone: true,
  imports: [ThingComponent],
  selector: 'app-root',
  template: `
    <app-thing></app-thing>
  `,
})
export class AppComponent {}
```



```ts
import {
  Directive,
  ElementRef,
  HostListener,
  OnInit,
  Renderer2,
  inject,
} from '@angular/core';

@Directive({ selector: '[appColor]', standalone: true })
export class ColorDirective implements OnInit {
  private readonly _renderer2 = inject(Renderer2);
  private readonly _elementRef = inject(ElementRef);

  private _color = 'green';
  private _count = 0;

  private _setColor() {
    this._renderer2.setStyle(
      this._elementRef.nativeElement,
      'color',
      this._color
    );
  }

  ngOnInit() {
    this._setColor();
  }

  @HostListener('click') onClick() {
    if (this._count % 2 === 0) {
      this._color = 'blue';
    } else {
      this._color = 'red';
    }

    this._count++;
    this._setColor();
  }
}
```

```ts
import { Component } from '@angular/core';
import { ColorDirective } from './directive';

@Component({
  selector: 'app-thing',
  template: `
    <div class="container">
      <button appColor>Click me</button>
    </div>
  `,
  styles: [
    `
      .container {
        background-color: blue;
        padding: 16px;
        border: 2px solid black;
        margin: 8px 8px 8px 8px;
      }
    `,
  ],
  standalone: true,
  imports: [ColorDirective],
})
export class ThingComponent {}
```

```ts
@Component({
  standalone: true,
  selector: 'app-first',
  template: ``,
})
export class FirstComponent {
  doThings() {
    // thing 1
    // thing 2
    // thing 3
    // thing 4
    // thing 5
  }
}
```

```ts
@Component({
  standalone: true,
  selector: 'app-second',
  template: ``,
})
export class SecondComponent {
  doThings() {
    // thing 1
    // thing 2
    // thing 3
    // thing 4
    // thing 5
  }
}
```

```ts
import { Injectable } from '@angular/core';

@Injectable({ providedIn: 'root' })
export class ThingsService {
  doThings() {
    // thing 1
    // thing 2
    // thing 3
    // thing 4
    // thing 5
  }
}
```

```ts
@Component({
  standalone: true,
  selector: 'app-first',
  template: ``,
})
export class FirstComponent {
  private readonly _thingsService = inject(ThingsService);

  doThings() {
    this._thingsService.doThings();
  }
}
```

```ts
@Component({
  standalone: true,
  selector: 'app-second',
  template: ``,
})
export class SecondComponent {
  private readonly _thingsService = inject(ThingsService);

  doThings() {
    this._thingsService.doThings();
  }
}
```

```ts
import { Component, OnInit } from '@angular/core';

@Component({
  standalone: true,
  selector: 'app-container',
  template: ` <p>{{ dateTimeString }}</p> `,
})
export class ContainerComponent implements OnInit {
  dateTimeString = ''

  ngOnInit() {
    const now = Date.now();
    const date = new Date(now);
    const dateString = date.toLocaleDateString();
    const timeString = date.toLocaleTimeString();

    this. dateTimeString = `${dateString} ${timeString}`;
  }
}
```

1/20/2024 7:26:25 PM

```ts
import { Pipe, PipeTransform } from '@angular/core';

@Pipe({
  name: 'appDate',
  standalone: true
})
export class DatePipe implements PipeTransform {
  transform(value: number): string {
    const date = new Date(value);
    const dateString = date.toLocaleDateString();
    const timeString = date.toLocaleTimeString();

    return `${dateString} ${timeString}`;
  }
}
```

```ts
import { Component } from '@angular/core';
import { DatePipe } from './date.pipe';

@Component({
  standalone: true,
  selector: 'app-container',
  template: ` <p>{{ now | appDate }}</p> `,
  imports: [DatePipe],
})
export class ContainerComponent {
  now = Date.now();
}
```

1/20/2024 7:26:25 PM

QUIZ #3

¿Cuál de las siguientes opciones no es parte de Angular?

a. Componentes.
b. Servicios.
c. Directivas.
d. Smart Contracts.

Ustedes siempre actúan como si fuesen mejor que yo

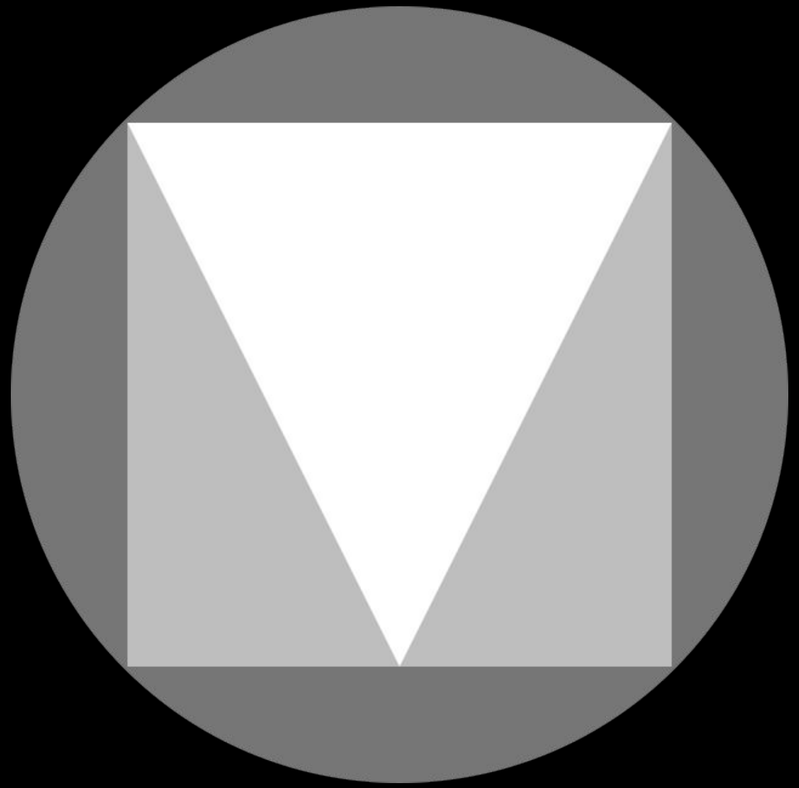
Click me

Primary

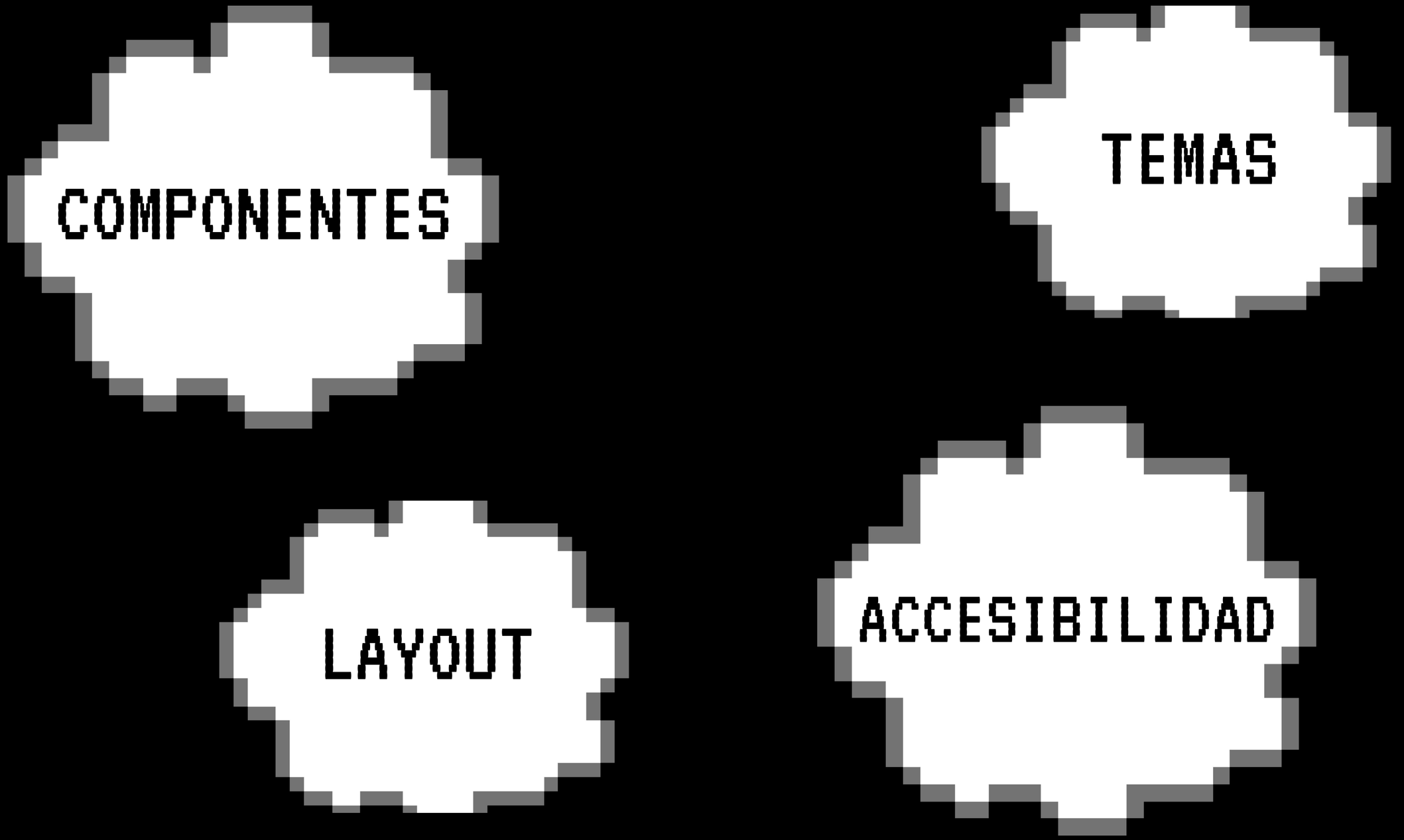
Primary

Submit

MATERIAL
DESIGN



ANGULAR
MATERIAL


COMPONENTES
TEMAS
LAYOUT
ACCESIBILIDAD

QUIZ #4

¿Cuál de las siguientes opciones es una razón para usar un Design System existente?

a. Procesar más transacciones.
b. Hacer aplicaciones más eficientes.
c. Hacer aplicaciones bonitas en poco tiempo.
d. Diseñar cosas únicas.

CSS

TAILWIND CSS

```ts
import { Component } from '@angular/core';
import { ColorDirective } from './color.directive';

@Component({
  selector: 'app-thing',
  template: `
    <div class="container">
      <button appColor>Click me</button>
    </div>
  `,
  styles: [
    `
      .container {
        background-color: blue;
        padding: 16px;
        border: 2px solid black;
        margin: 8px 8px 8px 8px;
      }
    `,
  ],
  standalone: true,
  imports: [ColorDirective],
})
export class ThingComponent {}
```

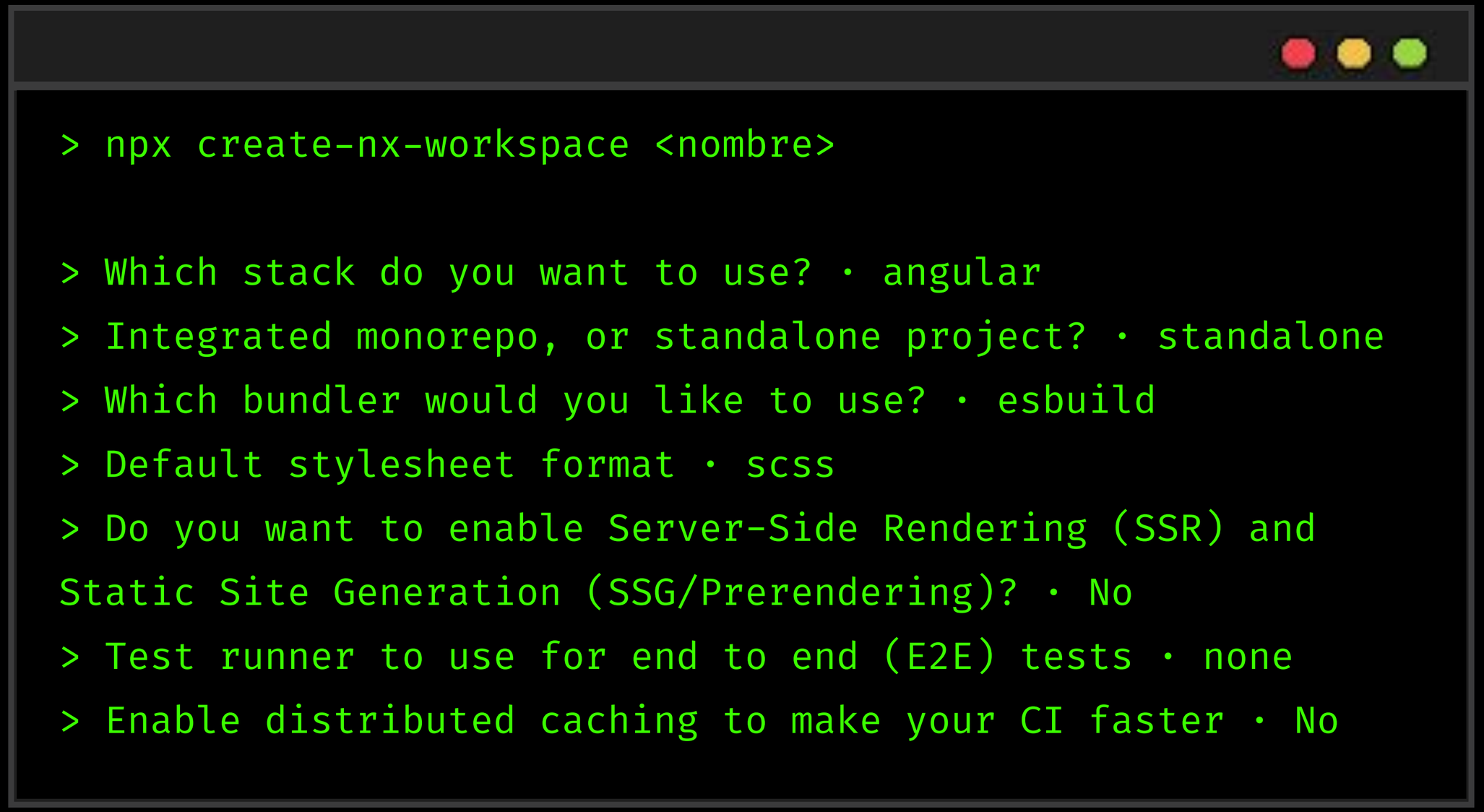
```ts
import { Component } from '@angular/core';
import { ColorDirective } from './color.directive';

@Component({
  selector: 'app-thing',
  template: `
    <div class="bg-blue-500 p-4 m-2 border-2 border-solid border-black">
      <button appColor>Click me</button>
    </div>
  `,
  standalone: true,
  imports: [ColorDirective],
})
export class ThingComponent {}
```

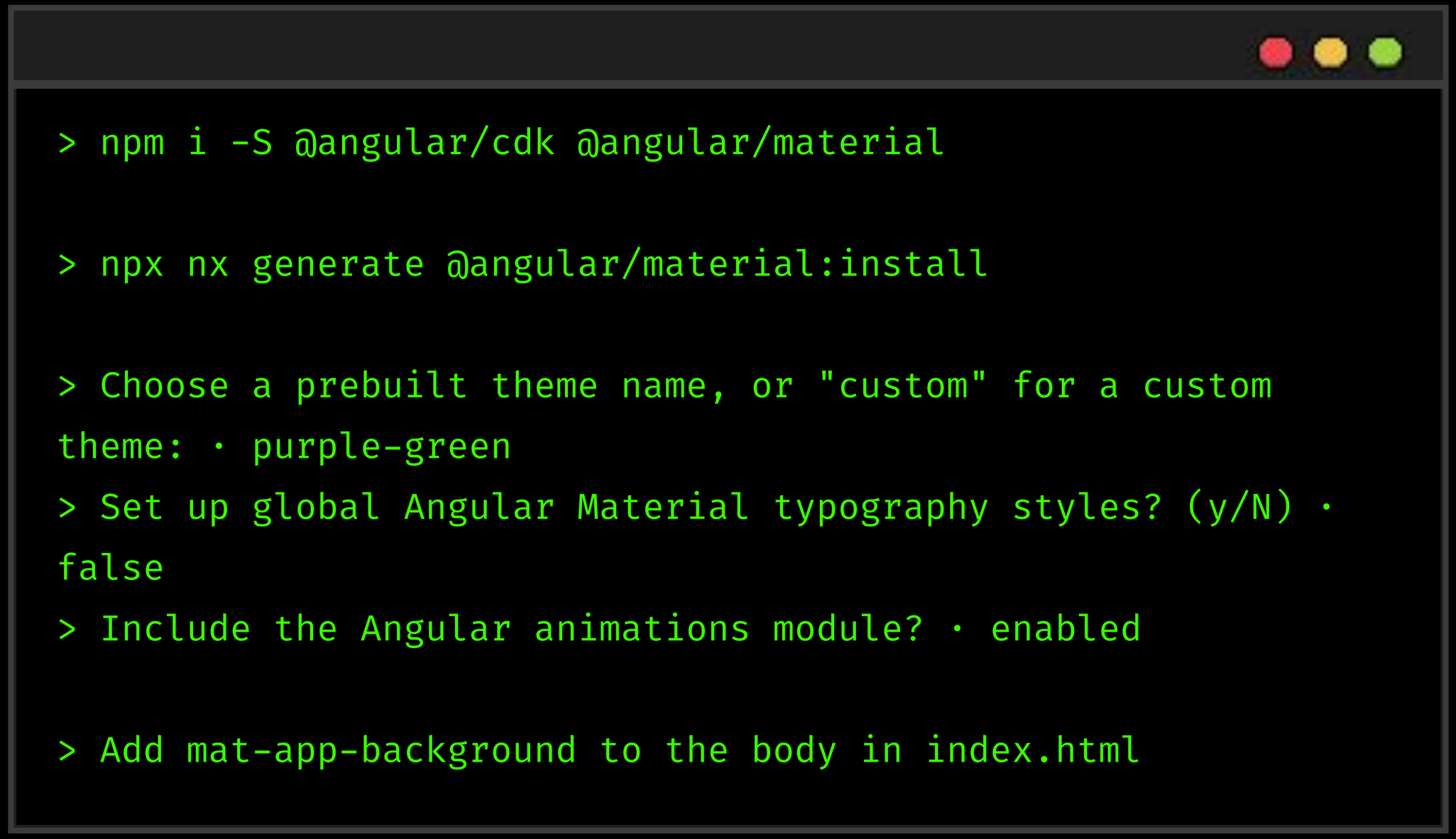
QUIZ #5

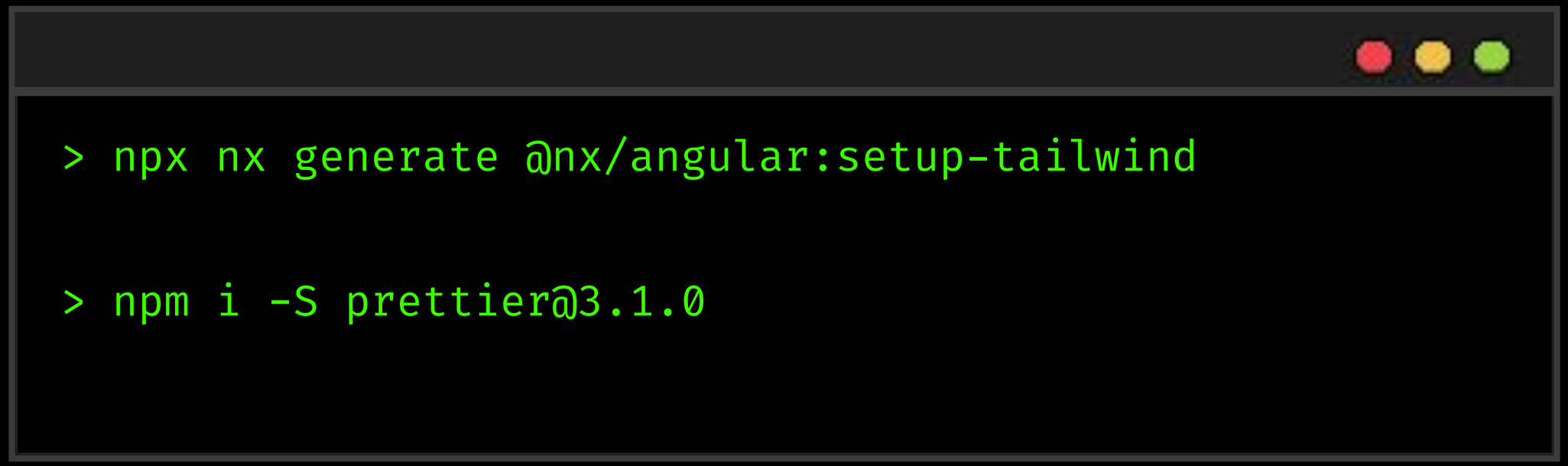
¿Cuál de las siguientes opciones agrega un padding de 16px en TailwindCSS?

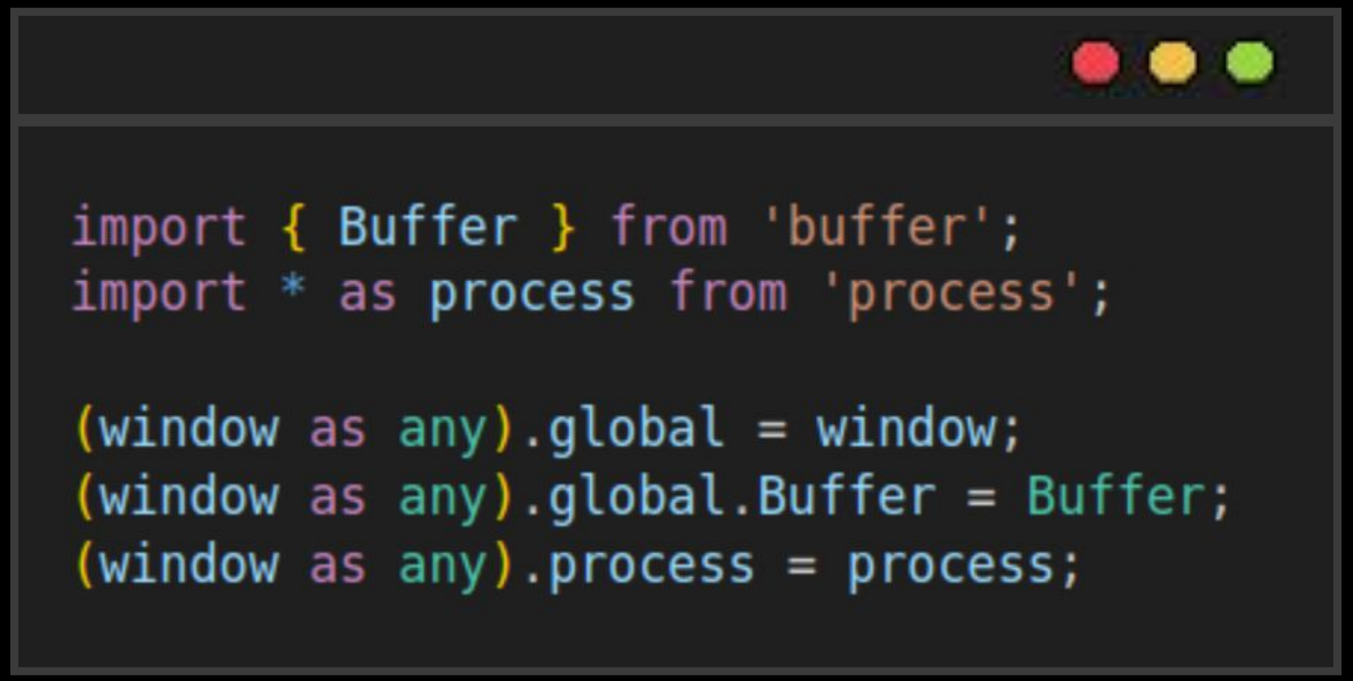
a. `padding-16`.
b. `p-16px`.
c. `p-4`.
d. `padding-16px`.

CUANDO USAS MONOREPOS


> npx create-nx-workspace <nombre>
> Which stack do you want to use? · angular
> Integrated monorepo, or standalone project? · standalone
> Which bundler would you like to use? · esbuild
> Default stylesheet format · scss
> Do you want to enable Server-Side Rendering (SSR) and
Static Site Generation (SSG/Prerendering)? · No
> Test runner to use for end to end (E2E) tests · none
> Enable distributed caching to make your CI faster · No


> npm i -S @angular/cdk @angular/material
> npx nx generate @angular/material:install
> Choose a prebuilt theme name, or "custom" for a custom theme: · purple-green
> Set up global Angular Material typography styles? (y/N) · false
> Include the Angular animations module? · enabled
> Add mat-app-background to the body in index.html


> npx nx generate @nx/angular:setup-tailwind
> npm i -S prettier@3.1.0



```
import { Buffer } from 'buffer';
import * as process from 'process';

(window as any).global = window;
(window as any).global.Buffer = Buffer;
(window as any).process = process;
```

Agregar polyfills en:

- project.json
- targets.build.options.polyfills
- tsconfig.app.json files

RETO #1

- Crear un workspace Nx standalone Angular.
- Configurar Angular Material.
- Configurar Tailwind CSS.
- Configurar polyfills para el Buffer.

¡SUERTE!