

## Simulando el próximo apagón: ¿Podremos evitarlo?

(En referencia al 28 de abril de 2025, cuando España vivió tensiones eléctricas en su red)

---

### Contextualización y Justificación

El alumnado parte del contexto real de las tensiones en la red eléctrica española ocurridas el 28 de abril de 2025, cuando, por una combinación de alta demanda y fallos en la generación, se estuvo a punto de provocar un apagón generalizado. Esta situación es el punto de partida para desarrollar un simulador web interactivo que permite ajustar generación y demanda, y visualizar el estado de la red en tiempo real.

Se busca no solo trabajar programación web (HTML, CSS, JavaScript), sino también integrar competencias digitales críticas, pensamiento sistémico y conciencia sobre sostenibilidad energética.

---

### Objetivos de Aprendizaje

#### Competencias clave:

- Competencia digital
- Competencia matemática y en ciencia, tecnología e ingeniería (STEM)
- Competencia para aprender a aprender
- Competencia social y cívica

#### Criterios de evaluación (adaptados del currículo de programación):

- Diseñar aplicaciones interactivas que simulen fenómenos reales.
  - Integrar estructuras dinámicas y visualización de datos en la interfaz.
  - Aplicar buenas prácticas de codificación y diseño responsivo.
  - Documentar y exponer el proyecto de manera clara, incluyendo análisis crítico de resultados.
-

## Desarrollo de la Situación de Aprendizaje

### Activación del conocimiento previo

- **Dinámica inicial:** ¿Qué ocurrió el 28 de abril de 2025? Análisis breve de noticias, gráficos y datos reales. Se pide a los alumnos identificar:
  - ¿Qué factores provocaron la tensión eléctrica?
  - ¿Qué variables se podrían controlar para evitarlo?
- **Exploración técnica:** Pequeña demostración del simulador ya desarrollado (el código proporcionado) para que entiendan qué funcionalidades tiene y qué aspectos podrían ampliarse.

### Planteamiento del reto

#### Pregunta guía:

*¿Serías capaz de programar un simulador que ayude a entender y prevenir un colapso eléctrico en tiempo real?*

El alumnado debe:

- Comprender el funcionamiento del código base.
- Ampliar funcionalidades propuestas (p. ej.: introducir eólica, baterías, predicción automática, alarmas sonoras, gráficos históricos).
- Mejorar el diseño visual y hacerlo adaptable a móviles.

### Desarrollo de las actividades

Semana	Actividad	Metodología
1	Lectura del código base + análisis de su estructura + detección de mejoras	Aprendizaje cooperativo
2	Talleres de refactorización: dividir en componentes, modularizar funciones, limpiar código	Programación en parejas
3	Diseño UX/UI: creación de prototipos visuales, mejora de interacción	Design Thinking
4	Desarrollo de nuevas funcionalidades + pruebas	Aprendizaje basado en proyectos (ABP)
5	Preparación de la presentación pública del proyecto + creación de documentación	Aprendizaje cooperativo
6	Exposición final ante compañeros, otros docentes o incluso personal del centro	Exposición pública

### *Producto final*

- **Simulador mejorado** presentado como proyecto web interactivo.
  - Documentación técnica explicativa.
  - Video breve (2-3 min) mostrando cómo funciona y qué aprendizajes han sacado del proyecto.
  - Reflexión grupal sobre los retos energéticos y tecnológicos.
- 

## **Evaluación**

---

### **Formativa (durante el proceso):**

- Rúbricas para evaluar el trabajo en equipo y la calidad del código.
- Rondas de feedback entre grupos.
- Uso de checklists semanales para autoevaluación.

### **Final:**

- Presentación del producto final usando una rúbrica con criterios como:
  - Funcionalidad técnica.
  - Innovación en mejoras añadidas.
  - Calidad estética y experiencia de usuario.
  - Claridad de la documentación y exposición.

### **Evaluación alternativa:**

- Portafolio digital por grupo con capturas de código, prototipos, pruebas y reflexiones.
- Diario de aprendizaje individual: ¿qué aprendí yo durante este proceso?

### **Evaluación del impacto:**

- Encuesta de motivación y satisfacción al alumnado.
  - Revisión grupal sobre qué ha funcionado bien y qué mejorarían.
- 

## **Recursos y materiales**

- Código base (HTML/CSS/JS).
- Ordenadores con editores de código (Visual Studio Code, por ejemplo).
- Herramientas de prototipado (Figma, Canva).

- Repositorio colaborativo (GitHub o similar).
  - Documentación de buenas prácticas en JavaScript.
  - Videos explicativos sobre redes eléctricas y sostenibilidad.
- 

### Adaptaciones y Diversificación

- **Dislexia/TDAH:** materiales visuales simplificados, uso de colores contrastados, tiempos de entrega flexibles.
  - **Altas capacidades:** retos adicionales como conexión con APIs externas, simulaciones predictivas, integración con datos reales.
  - **Tareas multinivel:** cada grupo puede elegir entre mejoras básicas (diseño, corrección de bugs) o complejas (nuevas funcionalidades, optimización).
- 

### Estrategias de Motivación

- Narrativa gamificada: el alumnado se convierte en “ingenieros de red” salvando a España del próximo apagón.
  - Retos sorpresa semanales con insignias (p. ej., mejor interfaz, código más limpio).
  - Relación con contextos reales: entrevistas (grabadas) a técnicos eléctricos, análisis de casos reales.
-

## Bloque 1: Comprender el código base

### 1 Exploración guiada

- Abre el código HTML, CSS y JavaScript en tu editor (recomendado: Visual Studio Code).
- Usa etiquetas de colores (comentarios) para marcar:
  - Estructura HTML principal.
  - Bloques de estilos CSS.
  - Bloques funcionales de JavaScript.

### 2 Mapa de componentes

- Dibuja (en papel o digital) un esquema que relacione los controles, los elementos visuales (contenedores) y las funciones de JavaScript.

### 3 Prueba y observa

- Ejecuta el simulador en el navegador.
- Cambia manualmente valores en los deslizadores y observa cómo afectan al balance y la frecuencia.

### 4 Documentación interna

- Añade comentarios explicativos en el código (mínimo 10 líneas) describiendo para qué sirve cada sección.
- 

## Bloque 2: Mejorar el simulador

### 1 Modificar estilos

- Cambia los colores principales usando las variables CSS (`:root`) para personalizarlo.
- Mejora la visualización en móviles con media queries.

### 2 Agregar nuevas funcionalidades

- Añade un nuevo control deslizante para generación eólica.
- Programa el cálculo del balance incluyendo esta nueva fuente.
- Haz que las líneas SVG cambien de color según su estado.

### 3 Mejorar la experiencia de usuario

- Añade animaciones suaves (p. ej., transiciones al cambiar estados).

- Introduce iconos adicionales o gráficos.

### Bloque 3: Desafíos avanzados (para alumnado que va más rápido)

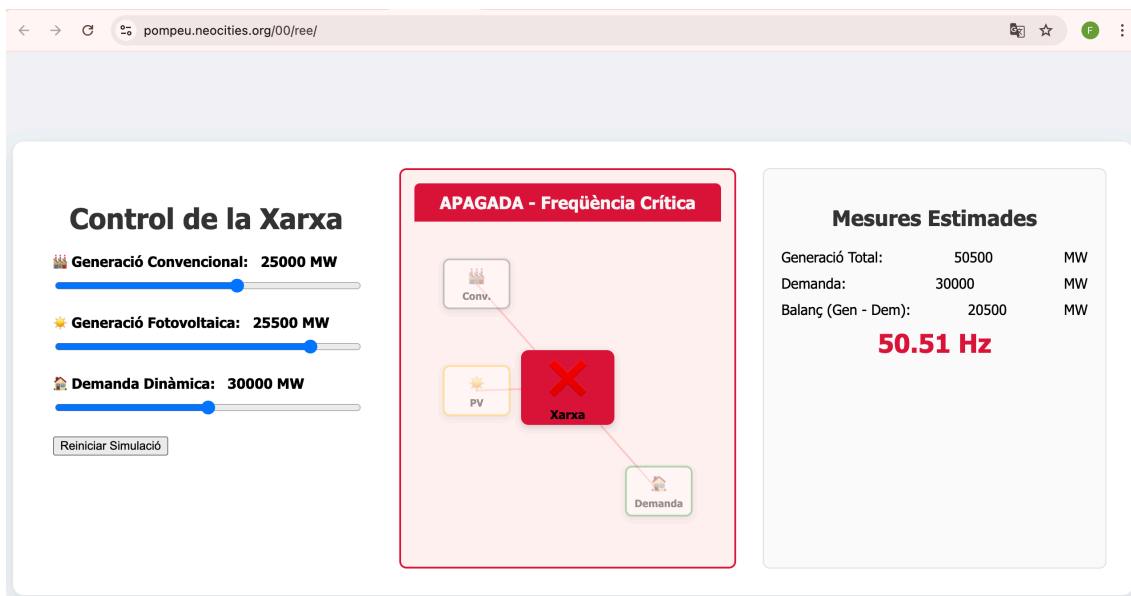
Conecta los controles a un gráfico de líneas que muestre la evolución en el tiempo (usa librerías como Chart.js).

Añade un modo automático donde la demanda fluctúe sola y el alumnado deba reaccionar en tiempo real.

Implementa un sistema de puntuación: ¿cuánto tiempo pueden mantener la red en estado normal?

## CODIGO FUENTE INICIAL:

<https://pompeu.neocities.org/00/ree/>



```
<!DOCTYPE html>
<html lang="ca">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Simulador Avançat Xarxa Elèctrica - Demanda Dinàmica amb Línies</title>
<style>
```

```
:root {
  --color-normal: #2e8b57;
  --color-warning: #ffa500;
  --color-shutdown: #dc143c;
  --color-line-normal: #77dd77;
  --color-line-warning: #ffcc00;
  --color-line-shutdown: #ff6961;
  --color-bg: #f0f4f8;
  --color-container-bg: #ffffff;
  --color-text-dark: #333333;
  --color-text-light: #f8f8f8;
  --color-disabled: #aaaaaa;
}
body {
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
  display: flex;
  justify-content: center;
  align-items: center;
  min-height: 100vh;
  margin: 0;
  background-color: var(--color-bg);
}
.main-container {
  background: var(--color-container-bg);
  padding: 30px;
  border-radius: 12px;
  box-shadow: 0 6px 20px rgba(0,0,0,0.1);
  display: flex;
  gap: 30px;
  flex-wrap: wrap;
  max-width: 1200px;
  width: 100%;
}
.controls, .visualization-container, .values {
  flex: 1;
  min-width: 320px;
  padding: 15px;
  box-sizing: border-box;
}
```

```
h1, h2 { text-align: center; color: var(--color-text-dark); margin-bottom: 20px; }
.control-group { margin-bottom: 20px; }
label { display: block; margin-bottom: 6px; font-weight: 600; }
input[type="range"] { width: 100%; cursor: pointer; }
.value-display { font-weight: bold; min-width: 60px; display: inline-block; text-align: right; }
.visualization-container {
border: 2px solid var(--color-disabled);
border-radius: 8px;
position: relative;
background: #f8f9fa;
display: flex;
flex-direction: column;
height: 450px;
transition: border-color 0.5s ease;
}
.status-banner {
padding: 10px;
text-align: center;
font-size: 1.2em;
font-weight: bold;
color: var(--color-text-light);
border-top-left-radius: 6px;
border-top-right-radius: 6px;
}
.grid-map {
flex-grow: 1;
position: relative;
padding: 20px;
overflow: hidden;
}
.grid-map svg {
position: absolute; top:0; left:0;
width:100%; height:100%;
z-index:1;
pointer-events: none;
}
.svg-power-line {
stroke: var(--color-line-normal);
```

```
stroke-width: 5;
stroke-linecap: round;
transition: stroke 0.5s ease, opacity 0.5s ease, stroke-width 0.5s ease;
}
.grid-element {
position: absolute;
width: 80px; height: 60px;
border-radius: 8px;
display: flex;
flex-direction: column;
align-items: center;
justify-content: center;
background: var(--color-container-bg);
border: 2px solid var(--color-disabled);
z-index: 2;
font-size: 0.75em;
font-weight: bold;
box-shadow: 0 3px 6px rgba(0,0,0,0.15);
padding: 5px;
box-sizing: border-box;
transition: opacity 0.5s, transform 0.3s;
}
#conventional-gen { top: 40px; left: 30px; border-color: #6c757d; }
#pv-gen { top: 160px; left: 30px; border-color: #ffc107; }
#demand-load { bottom: 40px; right: 30px; border-color: #28a745; }
#grid-core {
top: 50%; left: 50%; transform: translate(-50%,-50%);
width: 100px; height: 80px; border-color: #007bff;
font-size: 0.8em;
}
.icon { font-size: 1.6em; margin-bottom: 3px; }
#conventional-gen .icon::before { content: '🏠'; }
#pv-gen .icon::before { content: '☀️'; }
#demand-load .icon::before { content: '🏠'; }
#grid-core .icon::before { content: '⚡'; font-size: 2em; }
/* Estils d'estat */
.status-normal { border-color: var(--color-normal); }
.status-normal .status-banner { background: var(--color-normal); }
```

```
.status-normal .svg-power-line { stroke: var(--color-line-normal); opacity:1; stroke-
width:5; }

.status-warning { border-color: var(--color-warning); }
.status-warning .status-banner { background: var(--color-warning); }
.status-warning .svg-power-line { stroke: var(--color-line-warning); opacity:1; stroke-
width:6; }

.status-shutdown { border-color: var(--color-shutdown); background:#fff0f0; }
.status-shutdown .status-banner { background: var(--color-shutdown); }
.status-shutdown .svg-power-line { stroke: var(--color-line-shutdown); opacity:0.3;
stroke-width:2; }

.status-shutdown .grid-element { opacity:0.4; transform: scale(0.95); }
.status-shutdown #grid-core { transform: translate(-50%,-50%) scale(1.05);
background:var(--color-shutdown); border-color:var(--color-shutdown); opacity:1; }
.status-shutdown #grid-core .icon::before { content:'✖'; color: var(--color-text-light);
}

.values { border:1px solid #ddd; padding:20px; border-radius:8px;
background:#fdfdfd; }
.values p { display:flex; justify-content:space-between; margin:10px 0; }
#frequency-display { text-align:center; font-size:1.8em; margin-top:10px; }
.freq-normal { color: var(--color-normal); }
.freq-warning { color: var(--color-warning); }
.freq-shutdown { color: var(--color-shutdown); font-weight:bold; }
</style>
</head>
<body>
<div class="main-container">
<div class="controls">
<h1>Control de la Xarxa</h1>
<div class="control-group">
<label for="conventionalGen">🏠 Generació Convencional: <span class="value-
display" id="conventionalVal">0</span> MW</label>
<input type="range" id="conventionalGen" min="10000" max="35000" step="500"
value="25000">
</div>
<div class="control-group">
<label for="pvGen">☀️ Generació Fotovoltaica: <span class="value-display"
id="pvVal">0</span> MW</label>
<input type="range" id="pvGen" min="0" max="30000" step="500" value="5000">
</div>
<div class="control-group">
```

```

<label for="demandSlider">🏠 Demanda Dinàmica: <span class="value-display"
id="demandVal">30000</span> MW</label>
<input type="range" id="demandSlider" min="20000" max="40000" step="500"
value="30000">
</div>
<button id="resetBtn" disabled>Reiniciar Simulació</button>
</div>
<div class="visualization-container status-normal" id="visualizationContainer">
<div class="status-banner" id="statusBanner">Estat: Normal</div>
<div class="grid-map" id="gridMap">
<svg>
<line id="svg-line-conv" class="svg-power-line" x1="0" y1="0" x2="0" y2="0" />
<line id="svg-line-pv" class="svg-power-line" x1="0" y1="0" x2="0" y2="0" />
<line id="svg-line-demand" class="svg-power-line" x1="0" y1="0" x2="0" y2="0" />
</nsvg>
<div class="grid-element" id="conventional-gen"><span
class="icon"></span>Conv.</div>
<div class="grid-element" id="pv-gen"><span class="icon"></span>PV</div>
<div class="grid-element" id="grid-core"><span class="icon"></span>Xarxa</div>
<div class="grid-element" id="demand-load"><span
class="icon"></span>Demanda</div>
</div>
</div>
<div class="values">
<h2>Mesures Estimades</h2>
<p><span>Generació Total:</span><span id="totalGenVal">0</span> MW</p>
<p><span>Demanda:</span><span id="demandValDisplay">0</span> MW</p>
<p><span>Balanç (Gen - Dem):</span><span id="balanceVal">0</span> MW</p>
<div id="frequency-display"><span id="frequencyVal">50.00</span> Hz</div>
</div>
</div>
<script>
// Constants
const BASE_FREQ = 50.0;
const FREQ_SENS = 0.000025;
const NORMAL_MIN = 49.85, NORMAL_MAX = 50.15;
const WARN_MIN = 49.70, WARN_MAX = 50.30;
const SHUT_MIN = 49.50, SHUT_MAX = 50.50;
const DEF_CONV = 25000, DEF_PV = 5000, DEF_DEM = 30000;
// Elements

```

```
const convSlider = document.getElementById('conventionalGen');
const pvSlider = document.getElementById('pvGen');
const demSlider = document.getElementById('demandSlider');
const convVal = document.getElementById('conventionalVal');
const pvVal = document.getElementById('pvVal');
const demVal = document.getElementById('demandVal');
const totalVal = document.getElementById('totalGenVal');
const demDisp = document.getElementById('demandValDisplay');
const balanceVal = document.getElementById('balanceVal');
const freqVal = document.getElementById('frequencyVal');
const freqDisp = document.getElementById('frequency-display');
const visCont = document.getElementById('visualizationContainer');
const statusBan = document.getElementById('statusBanner');
const resetBtn = document.getElementById('resetBtn');
const gridMap = document.getElementById('gridMap');
const convEl = document.getElementById('conventional-gen');
const pvEl = document.getElementById('pv-gen');
const coreEl = document.getElementById('grid-core');
const demEl = document.getElementById('demand-load');
const lineConv = document.getElementById('svg-line-conv');
const linePv = document.getElementById('svg-line-pv');
const lineDem = document.getElementById('svg-line-demand');

function getCenter(el) {
  const mapR = gridMap.getBoundingClientRect();
  const r = el.getBoundingClientRect();
  return {
    x: r.left - mapR.left + r.width/2,
    y: r.top - mapR.top + r.height/2
  };
}

function updateLines() {
  const c = getCenter(convEl), p = getCenter(pvEl), o = getCenter(coreEl), d =
  getCenter(demEl);
  lineConv.setAttribute('x1', c.x); lineConv.setAttribute('y1', c.y);
  lineConv.setAttribute('x2', o.x); lineConv.setAttribute('y2', o.y);
  linePv.setAttribute('x1', p.x); linePv.setAttribute('y1', p.y);
  linePv.setAttribute('x2', o.x); linePv.setAttribute('y2', o.y);
  lineDem.setAttribute('x1', o.x); lineDem.setAttribute('y1', o.y);
  lineDem.setAttribute('x2', d.x); lineDem.setAttribute('y2', d.y);
}
```

```
}  
  
function updateSim() {  
  const conv = +convSlider.value;  
  const pv = +pvSlider.value;  
  const dem = +demSlider.value;  
  const total = conv + pv;  
  const bal = total - dem;  
  const freq = Math.max(49, Math.min(51, BASE_FREQ + bal * FREQ_SENS));  
  convVal.textContent = conv;  
  pvVal.textContent = pv;  
  demVal.textContent = dem;  
  totalVal.textContent = total;  
  demDisp.textContent = dem;  
  balanceVal.textContent = bal;  
  freqVal.textContent = freq.toFixed(2);  
  
  let cls = 'status-normal', msg = 'Estat: Normal';  
  if (freq < SHUT_MIN || freq > SHUT_MAX) {  
    cls = 'status-shutdown'; msg = 'APAGADA - Freqüència Crítica'; resetBtn.disabled =  
    false;  
  } else if (freq < WARN_MIN || freq > WARN_MAX) {  
    cls = 'status-warning'; msg = 'ALERTA - Desviació Freqüència'; resetBtn.disabled =  
    true;  
  } else {  
    resetBtn.disabled = true;  
  }  
  
  visCont.className = `visualization-container ${cls}`;  
  statusBan.textContent = msg;  
  freqDisp.className = cls.replace('status-', 'freq-');  
  updateLines();  
}  
  
function resetSim() {  
  convSlider.value = DEF_CONV;  
  pvSlider.value = DEF_PV;  
  demSlider.value = DEF_DEM;  
  resetBtn.disabled = true;  
  updateSim();  
}  
  
[convSlider, pvSlider, demSlider].forEach(s => s.addEventListener('input',  
  updateSim));
```

```
resetBtn.addEventListener('click', resetSim);  
window.addEventListener('resize', updateLines);  
document.addEventListener('DOMContentLoaded', resetSim);  
</script>  
</body>  
</html>
```