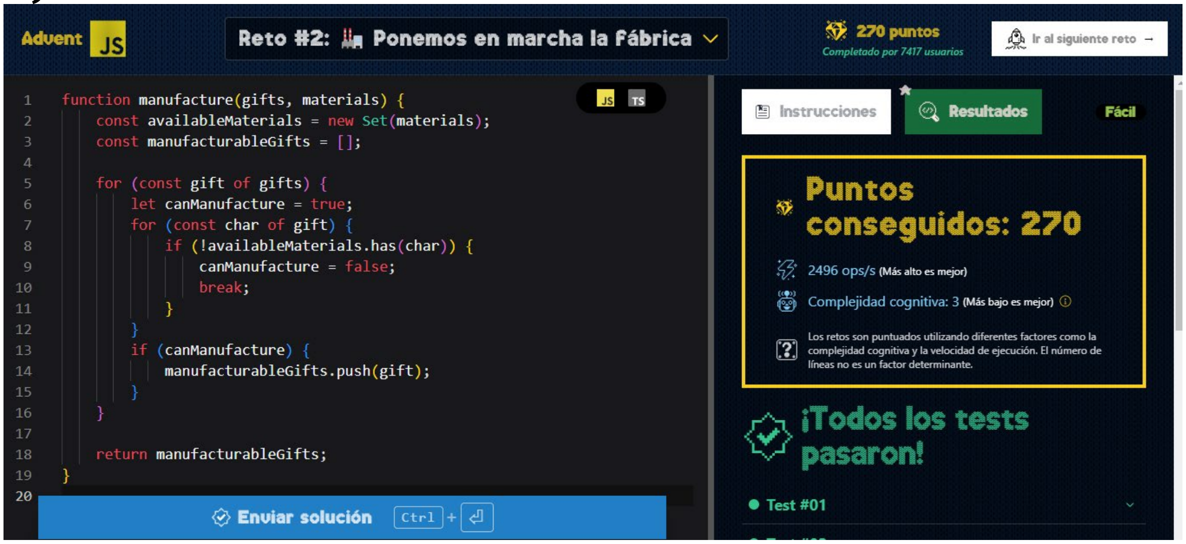
Advent JS

Retos y regalos de programación navideños

Ejercicio 2



```
function manufacture(gifts, materials) {
  const availableMaterials = new Set(materials);
  const manufacturableGifts = [];

  for (const gift of gifts) {
    let canManufacture = true;
    for (const char of gift) {
      if (!availableMaterials.has(char)) {
        canManufacture = false;
        break;
      }
    }
    if (canManufacture) {
      manufacturableGifts.push(gift);
    }
  }

  return manufacturableGifts;
}
```

Puntos conseguidos: 270

2496 ops/s (Más alto es mejor)

Complejidad cognitiva: 3 (Más bajo es mejor)

Los retos son puntuados utilizando diferentes factores como la complejidad cognitiva y la velocidad de ejecución. El número de líneas no es un factor determinante.

¡Todos los tests pasaron!

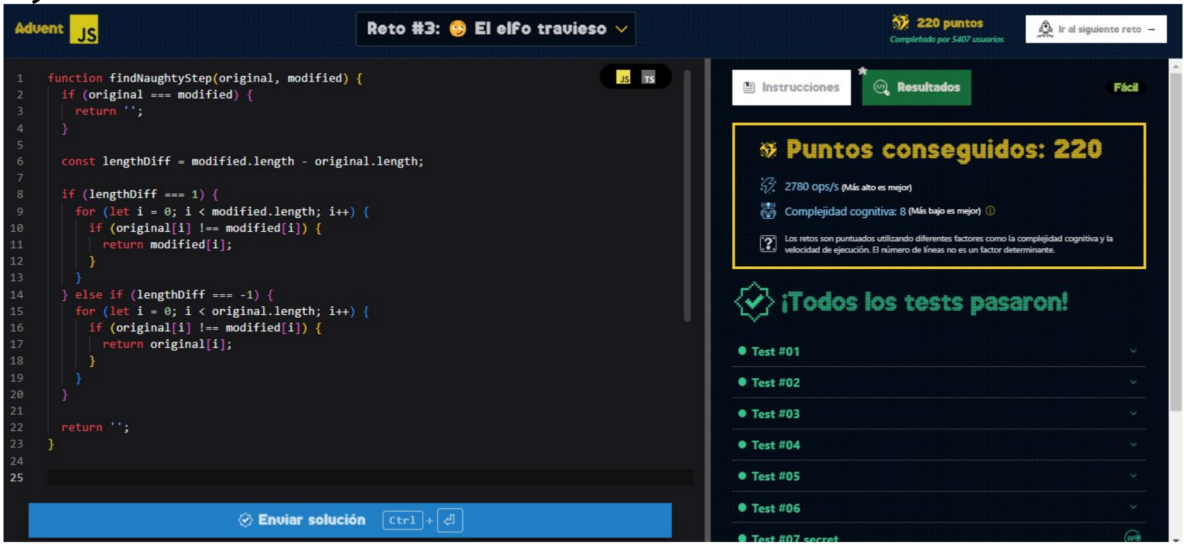
Código JS:

```
function manufacture(gifts, materials) {
const availableMaterials = new Set(materials);
const manufacturableGifts = [];

for (const gift of gifts) {
    let canManufacture = true;
    for (const char of gift) {
        if (!availableMaterials.has(char)) {
            canManufacture = false;
            break;
        }
    }
    if (canManufacture) {
        manufacturableGifts.push(gift);
        }
    }

    return manufacturableGifts;
}
```

Ejercicio 3



```
function findNaughtyStep(original, modified) {
  if (original === modified) {
    return '';
  }

  const lengthDiff = modified.length - original.length;

  if (lengthDiff === 1) {
    for (let i = 0; i < modified.length; i++) {
      if (original[i] !== modified[i]) {
        return modified[i];
      }
    }
  } else if (lengthDiff === -1) {
    for (let i = 0; i < original.length; i++) {
      if (original[i] !== modified[i]) {
        return original[i];
      }
    }
  }

  return '';
}
```

Puntos conseguidos: 220

2780 ops/s (Más alto es mejor)

Complejidad cognitiva: 8 (Más bajo es mejor)

Los retos son puntuados utilizando diferentes factores como la complejidad cognitiva y la velocidad de ejecución. El número de líneas no es un factor determinante.

¡Todos los tests pasaron!

- Test #01
- Test #02
- Test #03
- Test #04
- Test #05
- Test #06
- Test #07 secret

Código JS:

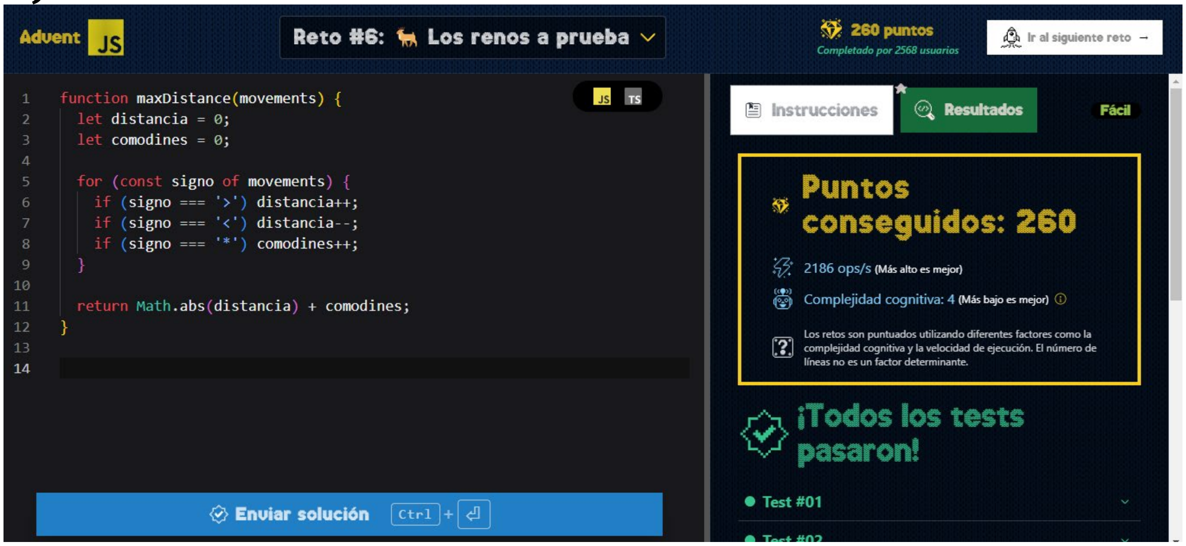
```
function findNaughtyStep(original, modified) {
if (original === modified) {
  return '';
}

const lengthDiff = modified.length - original.length;

if (lengthDiff === 1) {
  for (let i = 0; i < modified.length; i++) {
    if (original[i] !== modified[i]) {
      return modified[i];
    }
  }
  } else if (lengthDiff === -1) {
    for (let i = 0; i < original.length; i++) {
      if (original[i] !== modified[i]) {
        return original[i];
      }
    }
  }

  return '';
}
```

Ejercicio 6



```
function maxDistance(movements) {
  let distancia = 0;
  let comodines = 0;

  for (const signo of movements) {
    if (signo === '>') distancia++;
    if (signo === '<') distancia--;
    if (signo === '*') comodines++;
  }

  return Math.abs(distancia) + comodines;
}
```

Puntos conseguidos: 260

2186 ops/s (Más alto es mejor)

Complejidad cognitiva: 4 (Más bajo es mejor)

Los retos son puntuados utilizando diferentes factores como la complejidad cognitiva y la velocidad de ejecución. El número de líneas no es un factor determinante.

¡Todos los tests pasaron!

- Test #01
- Test #02

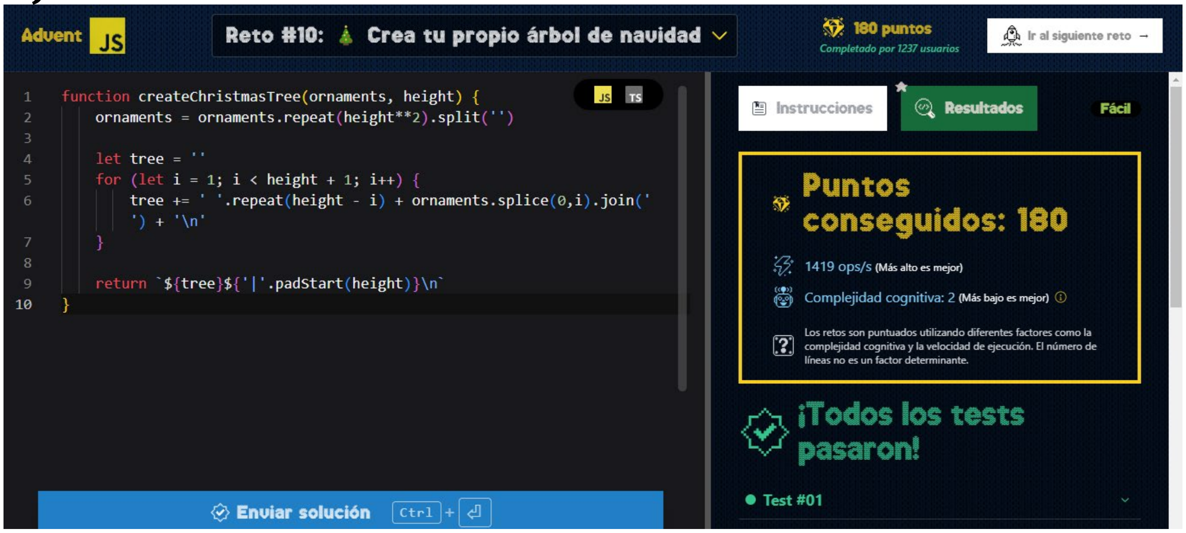
Código JS:

```
function maxDistance(movements) {
  let distancia = 0;
  let comodines = 0;

  for (const signo of movements) {
    if (signo === '>') distancia++;
    if (signo === '<') distancia--;
    if (signo === '*') comodines++;
  }

  return Math.abs(distancia) + comodines;
}
```

Ejercicio 10



```
function createChristmasTree(ornaments, height) {
    ornaments = ornaments.repeat(height**2).split('')

    let tree = ''
    for (let i = 1; i < height + 1; i++) {
        tree += ' '.repeat(height - i) + ornaments.splice(0,i).join('
        ') + '\n'
    }

    return `${tree}${'|'.padStart(height)}\n`
}
```

Puntos conseguidos: 180

1419 ops/s (Más alto es mejor)

Complejidad cognitiva: 2 (Más bajo es mejor)

Los retos son puntuados utilizando diferentes factores como la complejidad cognitiva y la velocidad de ejecución. El número de líneas no es un factor determinante.

¡Todos los tests pasaron!

Código:

```
function createChristmasTree(ornaments, height) {
    ornaments = ornaments.repeat(height**2).split('')

    let tree = ''
    for (let i = 1; i < height + 1; i++) {
        tree += ' '.repeat(height - i) +
ornaments.splice(0,i).join(' ') + '\n'
    }

    return `${tree}${'|'.padStart(height)}\n`
}
```