



**UNIVERSIDAD DE LAS FUERZAS ARMADAS - ESPE**  
**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN**  
**Pruebas de Software**

Integrantes: Mateo Sosa

NRC: 27854

Fecha: February 10, 2026

## Contents

<b>CI (lógica + pruebas sobre funciones)</b>	<b>2</b>
Funciones obligatorias (en src/utils/) . . . . .	2
Casos de referencia que deben pasar en tests: . . . . .	3
Criterios de aceptación CI (NRC B) . . . . .	5
<b>CD (despliegue a Firebase con página estática)</b>	<b>7</b>
Criterios de aceptación CD . . . . .	8

# CI (lógica + pruebas sobre funciones)

## Funciones obligatorias (en src/utils/)

### 1. calcWeightedGrade(items)

Calcula una nota final ponderada a partir de componentes con peso.

- items: arreglo de objetos {score: número 0–100, weight: número 0–1}.
- Suma de weight = 1 con tolerancia  $\pm 0.001$ ; si no, RangeError.
- Devuelve nota 0–100 con 2 decimales.
- Validar tipos/rangos; lanzar TypeError/RangeError según corresponda.



```
1 function calcWeightedGrade(items) {
2     // Mateo Sosa
3     let totalWeight = 0;
4     let sum = 0;
5
6     try {
7         for (let i of items) {
8             totalWeight += i.weight;
9         }
10
11        for (let i of items) {
12            sum += i.score * (i.weight / totalWeight);
13        }
14    } catch {
15        throw TypeError('items debe ser un Array');
16    }
17
18    return sum;
19 }
```

Figure 1: Función de Cálculo por Peso de Notas

### 2. percentile(p, values)

Devuelve el percentil p de una lista de números usando el método de rango más cercano (sin interpolación).

- p en  $[0,100]$ ; values: arreglo de números, longitud  $\geq 1$ .
- Ordena ascendente; rank =  $\text{ceil}(p/100 \times N)$  con indexación 1..N.
- Regla explícita para bordes: si  $p = 0 \rightarrow$  devuelve mínimo; si  $p = 100 \rightarrow$  máximo.
- Resultado con 2 decimales; validar tipos/rangos; lanzar errores apropiados.



```
1  function percentile(p, values) {
2    // Mateo Sosa
3    if (p > 100 || p < 0) { // Máximo 100%
4      throw RangeError('El porcentaje debe existir entre 0 y 100');
5    }
6
7    try {
8      const percentage = 1 / values.length;
9      const i = Math.ceil(p * percentage / 100);
10
11     return values[i];
12   } catch {
13     throw TypeError('values debe ser un Array');
14   }
15 }
```

Figure 2: Función de Cálculo de Percentil

## Casos de referencia que deben pasar en tests:

- calcWeightedGrade([{"score":80,"weight":0.4}, {"score":90,"weight":0.6}]) → 86.00



```
1  test('Calcular Nota por Peso', () => {
2    // Mateo Sosa
3    const items = [
4      {
5        score: 80,
6        weight: 0.4
7      },
8      {
9        score: 90,
10       weight: 0.6
11     }
12   ];
13
14   expect(calcWeightedGrade(items)).toBe(86);
15 });
```

Figure 3: Prueba para la Función *calcWeightedGrade*

```

● msosa@fedora:~/Documents/ESPE/Pruebas/P3/examen$ npm test

> examen@1.0.0 test
> jest

PASS  test/utils.test.js
  ✓ Calcular Nota por Peso (3 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        0.421 s, estimated 1 s
Ran all test suites.

```

Figure 4: Ejecución de Prueba *calcWeightedGrade*

- percentile(0,[1,2,3]) → 1.00 ; percentile(100,[1,2,3]) → 3.00
- percentile(50,[1,2,3,4]) → 2.00 (nearest-rank)



```

1  test('Calcular Percentiles Lista 3 Elementos', () => {
2    // Mateo Sosa
3    const data = [1, 2, 3];
4    const p = 0;
5
6    expect(percentile(p, data)).toBe(1);
7  });
8
9  test('Calcular Percentiles Lista 4 Elementos', () => {
10   // Mateo Sosa
11   const data = [1, 2, 3, 4];
12   const p = 50;
13
14   expect(percentile(p, data)).toBe(2);
15 });
16
17 test('Calcular Percentiles Incorrectos', () => {
18   expect(percentile(150, [1, 2, 3])).toError();
19 });

```

Figure 5: Pruebas para la Función *percentile*

```

@ msosa@fedora:~/Documents/ESPE/Pruebas/P3/examen$ npm test
> examen@1.0.0 test
> jest

 FAIL | test/utils.test.js
  ✓ Calcular Percentiles Lista 3 Elementos (4 ms)
  ✓ Calcular Percentiles Lista 4 Elementos (1 ms)
  ✘ Calcular Percentiles Incorrectos (1 ms)

  • Calcular Percentiles Incorrectos

    RangeError: El porcentaje debe existir entre 0 y 100

    27 |   // Mateo Sosa
    28 |   if (p > 100 || p < 0) { // Máximo 100%
    > 29 |     throw RangeError('El porcentaje debe existir entre 0 y 100');
        |     ^
    30 |   }
    31 |
    32 |   try {

      at RangeError (src/utils.js:29:11)
      at Object.percentile (test/utils.test.js:36:10)

Test Suites: 1 failed, 1 total
Tests:       1 failed, 2 passed, 3 total
Snapshots:  0 total
Time:        0.459 s, estimated 1 s
Ran all test suites.

```

Figure 6: Ejecución de Prueba *percentile*

## Criterios de aceptación CI (NRC B)



```

1  export default [
2    {
3      files: ['**/*.js'],
4      languageOptions: {
5        ecmaVersion: 'latest',
6        sourceType: 'module',
7      },
8      rules: {
9        semi: ['error', 'always'],
10       quotes: ['error', 'single'],
11     }
12   }
13 ];
14

```

Figure 7: Configuración de ESLint

- CI corre en push/PR a main.



Figure 8: Ejecución de CI en GitHub Actions

- Lint sin errores.

```
● msosa@fedora:~/Documents/ESPE/Pruebas/P3/examen$ npm run lint
> examen@1.0.0 lint
> eslint

(node:27128) [MODULE_TYPELESS_PACKAGE_JSON] Warning: Module type of file:///home/msosa/Documents/ESPE/Pruebas/P3/examen/es.
se as CommonJS.
Reparising as ES module because module syntax was detected. This incurs a performance overhead.
To eliminate this warning, add "type": "module" to /home/msosa/Documents/ESPE/Pruebas/P3/examen/package.json.
(Use 'node --trace-warnings ...' to show where the warning was created)
○ msosa@fedora:~/Documents/ESPE/Pruebas/P3/examen$
```

Figure 9: Cobertura de Errores Tipográficos en Local

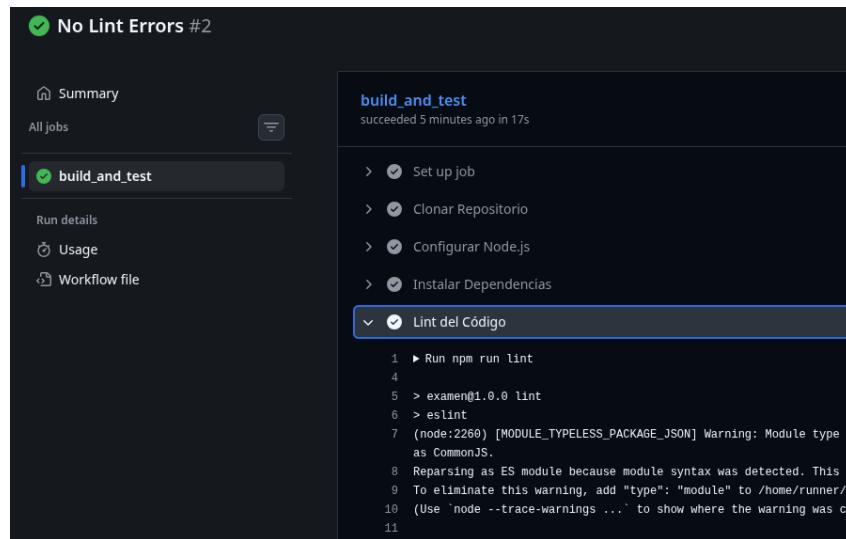


Figure 10: Cobertura de Errores Tipográficos en Remoto

- Revisión de Pruebas en GitHub Actions

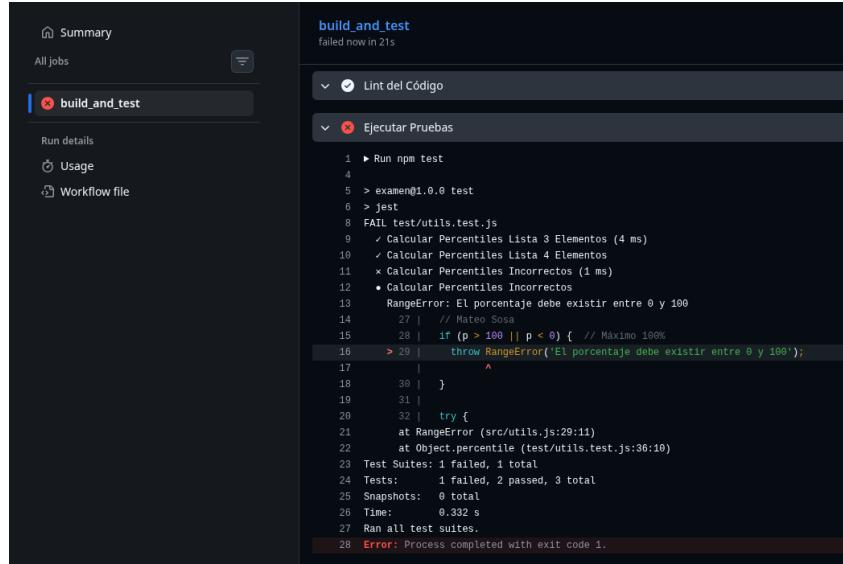


Figure 11: Ejecución de Pruebas en Workflow

## CD (despliegue a Firebase con página estática)

```

msosa@fedora:~/Documents/ESPE/Pruebas/P3/examen$ npm exec firebase init hosting
✓ Please select an option: Use an existing project
✓ Select a default Firebase project for this directory: examen-p3-879e0 (examen-p3)

== Hosting Setup

Your public directory is the folder (relative to your project directory) that
will contain Hosting assets to be uploaded with firebase deploy. If you
have a build process for your assets, use your build's output directory.

✓ What do you want to use as your public directory? public
✓ Configure as a single-page app (rewrite all urls to /index.html)? No
✓ Set up automatic builds and deploys with GitHub? Yes

i  Detected a .git folder at /home/msosa/Documents/ESPE/Pruebas/P3/examen
i  Authorizing with GitHub to upload your service account to a GitHub repository's secrets store.

Visit this URL on this device to log in:
https://github.com/login/oauth/authorize?client_id=89cf50f02ac6aaed3484&state=912326271&redirect_uri=http%3A

Waiting for authentication...

✓ Success! Logged into GitHub as MatSosa1

```

Figure 12: Comando de Inicialización de Proyecto en Firebase

### Página requerida

- Título: “Perfil — Notas & Estadística”.
- Contenido mínimo:
  - Nombre completo, NRC, correo institucional.
  - Un párrafo corto de presentación (3–5 líneas).
  - Una tabla sencilla con 3–4 asignaturas y calificaciones simuladas (no reales).
- Estilos básicos (sin frameworks):
  - Encabezado y secciones con bordes suaves; paleta de 2–3 colores; tipografía sans.

## Criterios de aceptación CD

- Preview visible en el PR (URL única del preview).

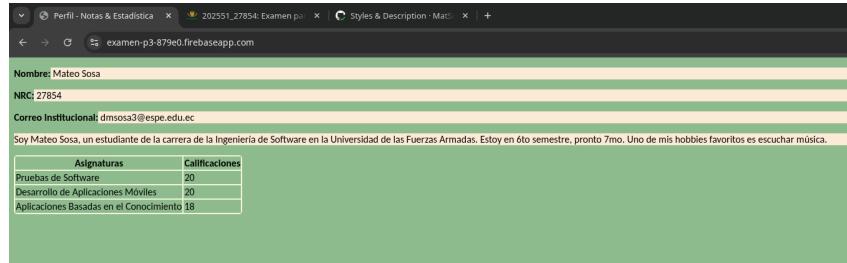


Figure 13: Página Web Hosteada en Firebase

El link de acceso es: <https://examen-p3-879e0.firebaseioapp.com/>

- Live al hacer merge a main (condicionado a CI verde).

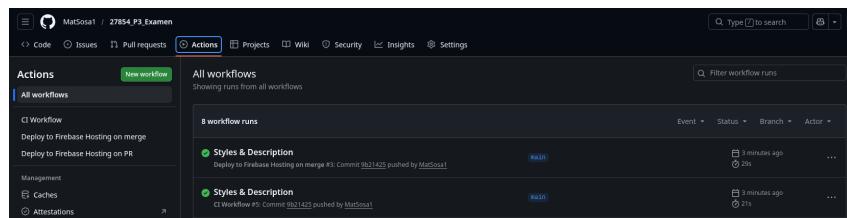


Figure 14: Comprobación de Workflows Correctos en GitHub Actions