

## Cotizador de créditos

### Repositorios

[cotizador-productos](#)

<https://github.com/funerhk/cotizador-productos.git>

[cotizador-plazos](#)

<https://github.com/funerhk/cotizador-plazos.git>

[cotizador-creditos](#)

<https://github.com/funerhk/cotizador-creditos.git>

[cotizador-web-app](#)

<https://github.com/funerhk/cotizador-web-app.git>

### Documentacion Swagger

[cotizador-productos](#)

<http://localhost:8080/swagger-ui/index.html>

[cotizador-plazos](#)

<http://localhost:8081/swagger-ui/index.html>

[cotizador-creditos](#)

<http://localhost:8082/swagger-ui/index.html>

## Flujo de cotización front

The screenshot displays the 'Cotizador-credito' web application. The browser address bar shows 'http://localhost:4200/products'. The application has a dark header with the title 'Cotizador-credito' and two navigation links: 'Agregar Productos' and 'Agregar Plazos'.

Below the header, there is a search bar with the placeholder text 'Buscar por nombre' and a 'buscar' button. The main content area is divided into three sections:

- Lista de productos:** A list of products with 'calcetines nike' selected and highlighted in blue. Other items include 'gorro cafe', 'sabrero rojo', 'mochila nike', and 'playera nike'.
- Detalle de producto:** A form for the selected product 'calcetines nike'. It includes fields for 'Nombre' (filled with 'calcetines nike'), 'Descripción' (filled with 'calcetines cortos blancos con negro'), and 'Precio' (filled with '50'). Below these fields are two buttons: 'Borrar' (red) and 'Modificar' (green).
- Lista de plazos:** A list of terms with '12 plazos' selected and highlighted in blue. Other items include '9 plazos' and '6 plazos'.

At the bottom, there is a 'Cotizacion' section showing the selected items and their values:

- SKU:** 664c2101899f31462e922a4b
- Nombre:** calcetines nike
- Precio:** 50
- Plazo:** 12

Borrar

Modificar

Lista de plazos

12 plazos

9 plazos

6 plazos

3 plazos

Cotizacion

SKU: 664c2101899f31462e922a4b

Nombre: calcetines nike

Precio: 50

Plazo: 12

Interes normal: 8.485833333333334

Interes puntual: 7.90125

Cotizar

**Bonus 3: ¿cómo harías una función en donde multipliques dos números dados sin USAR "\*" como operador?, es decir, sin USAR por ejemplo `let resultado = numero1 * numero2`**

```
function multiplicar(numero1, numero2) {
  let resultado = 0;
  for (let i = 0; i < numero2; i++) {
    resultado += numero1;
  }
  return resultado;
}
```

**Bonus 4: ¿Cuáles son los parámetros nombrados en JavaScript y para qué sirven?, da un ejemplo de cómo los usarías.**

Los parámetros nombrados en JavaScript son una característica que permite pasar argumentos a una función utilizando nombres específicos para cada parámetro, en lugar de depender del orden de los argumentos en la llamada a la función. Esto ofrece claridad y flexibilidad al invocar funciones con múltiples parámetros, ya que permite identificar explícitamente qué valor se asigna a qué parámetro.

```
function calcularAreaRectangulo({ base, altura }) {
  return base * altura;
}
```

```
const area = calcularAreaRectangulo({ base: 5, altura: 10 });
```

**Bonus 5: ¿Qué métodos de un array en JavaScript conoces?**

`push()`: Agrega uno o más elementos al final de un array y devuelve la nueva longitud del array.

```
const array = [1, 2, 3];
array.push(4, 5);
```

`pop()`: Elimina el último elemento de un array y lo devuelve.

```
const array = [1, 2, 3];  
const removedElement = array.pop();
```

shift(): Elimina el primer elemento de un array y lo devuelve.

```
const array = [1, 2, 3];  
const removedElement = array.shift();
```

unshift(): Agrega uno o más elementos al inicio de un array y devuelve la nueva longitud del array.

```
const array = [2, 3];  
array.unshift(0, 1);
```

splice(): Cambia el contenido de un array eliminando o reemplazando elementos existentes y/o agregando nuevos elementos.

```
const array = [1, 2, 3, 4, 5];  
array.splice(2, 1); // Elimina el elemento en la posición 2
```

slice(): Devuelve una copia superficial de una porción de un array en un nuevo array.

```
const array = [1, 2, 3, 4, 5];  
const newArray = array.slice(1, 3); // Devuelve [2, 3]
```

concat(): Retorna un nuevo array que combina el array original con otros arrays y/o valores.

```
const array1 = [1, 2];  
const array2 = [3, 4];  
const newArray = array1.concat(array2); // Devuelve [1, 2, 3, 4]
```

forEach(): Ejecuta una función por cada elemento del array.

```
const array = [1, 2, 3];  
1. array.forEach(item => console.log(item));
```