# RESULTADO DE CHALLENGE KOSMOS

**Nombre:** Ariel Fuentes García.

**Fecha:** 18 de febrero de 2025.

**Profesión:** Programador Web.

Tabla de contenido

[RESULTADO DE CHALLENGE KOSMOS 1](#_Toc190769074)

[Código completo 2](#_Toc190769075)

[Salida en consola por imagen 5](#_Toc190769076)

[Salida por consola en código 6](#_Toc190769077)

## Código completo

function calculoMotor(tipoDeNomina, fechaPrimerEmpleo, genero) {

  //\* Calcular la cantidad de meses transcurridos desde fechaPrimerEmpleo hasta hoy 18-02-2025

  const fechaActual = new Date();

  let años = fechaActual.getFullYear() - fechaPrimerEmpleo.getFullYear();

  let meses = fechaActual.getMonth() - fechaPrimerEmpleo.getMonth();

  let días = fechaActual.getDate() - fechaPrimerEmpleo.getDate();

  let totalMeses = años \* 12 + meses;

  if (días < 0) {

    totalMeses--;

  }

  let montoMinimoDeCredito, montoMáximoDeCredito;

  //\* convertir a minúsculas para la validación exacta

  genero = genero.toLowerCase();

  if (genero === 'm') {

    //\* Tablas para género masculino

    if (totalMeses < 26) {

      // [0-26)

      montoMinimoDeCredito = { A: 100,  B: 1000, C: 400, D: 400 }[tipoDeNomina];

      montoMáximoDeCredito = { A: 4900, B: 4700, C: 5000, D: 4400 }[tipoDeNomina];

    } else if (totalMeses === 27) {

      montoMinimoDeCredito = { A: 400,  B: 600,  C: 200, D: 300 }[tipoDeNomina];

      montoMáximoDeCredito = { A: 4700, B: 4400, C: 4700, D: 4700 }[tipoDeNomina];

    } else if (totalMeses === 28) {

      montoMinimoDeCredito = { A: 900,  B: 1000, C: 200, D: 500 }[tipoDeNomina];

      montoMáximoDeCredito = { A: 4600, B: 5000, C: 5000, D: 4300 }[tipoDeNomina];

    } else if (totalMeses === 29) {

      montoMinimoDeCredito = { A: 100,  B: 1000, C: 1000, D: 900 }[tipoDeNomina];

      montoMáximoDeCredito = { A: 4600, B: 4400, C: 4200, D: 4900 }[tipoDeNomina];

    } else if (totalMeses >= 30) {

      montoMinimoDeCredito = { A: 600,  B: 1000, C: 600, D: 1000 }[tipoDeNomina];

      montoMáximoDeCredito = { A: 4500, B: 4900, C: 4600, D: 4300 }[tipoDeNomina];

    }

  } else if (genero === 'f') {

    //\* Tablas para género femenino

    if (totalMeses < 24) {

      // [0-24)

      montoMinimoDeCredito = { A: 800,  B: 800,  C: 200, D: 500 }[tipoDeNomina];

      montoMáximoDeCredito = { A: 4000, B: 4700, C: 4600, D: 5000 }[tipoDeNomina];

    } else if (totalMeses === 25) {

      montoMinimoDeCredito = { A: 800,  B: 700,  C: 900,  D: 1000 }[tipoDeNomina];

      montoMáximoDeCredito = { A: 4200, B: 4200, C: 4900, D: 4900 }[tipoDeNomina];

    } else if (totalMeses === 26) {

      montoMinimoDeCredito = { A: 800,  B: 100,  C: 700,  D: 600 }[tipoDeNomina];

      montoMáximoDeCredito = { A: 4100, B: 4500, C: 4600, D: 4700 }[tipoDeNomina];

    } else if (totalMeses === 27) {

      montoMinimoDeCredito = { A: 600,  B: 600,  C: 800,  D: 400 }[tipoDeNomina];

      montoMáximoDeCredito = { A: 4200, B: 4300, C: 4700, D: 5000 }[tipoDeNomina];

    } else if (totalMeses >= 28) {

      montoMinimoDeCredito = { A: 200,  B: 700,  C: 100,  D: 700 }[tipoDeNomina];

      montoMáximoDeCredito = { A: 4500, B: 4400, C: 4000, D: 4300 }[tipoDeNomina];

    }

  } else {

    throw new Error("Género inválido. Use 'm' o 'f'.");

  }

  //\* Cálculo de la línea de crédito óptima:

  //TODO: p1 = montoMinimo + sqrt(montoMáximo - montoMinimo)

  let p1 = montoMinimoDeCredito + Math.sqrt(montoMáximoDeCredito - montoMinimoDeCredito);

  //TODO: p2 = montoMinimo + 0.0175 \* (montoMáximo - montoMinimo)

  let p2 = montoMinimoDeCredito + 0.0175 \* (montoMáximoDeCredito - montoMinimoDeCredito);

  let recomendacionLinea = Math.max(p1, p2);

  return {

    montoMinimo: montoMinimoDeCredito,

    montoMáximo: montoMáximoDeCredito,

    lineaOptimaDeCredito: recomendacionLinea

  };

}

//\* Conversión de fechas para la tabla de resultados

function parseFecha(fechaStr) {

  const [dia, mes, año] = fechaStr.split('/').map(Number);

  return new Date(año, mes - 1, dia);

}

//\* Datos de prueba de la tabla de ejemplo

const casosPrueba = [

  { tipoNomina: 'A', fecha: '12/06/2022', genero: 'f' },

  { tipoNomina: 'B', fecha: '30/12/1993', genero: 'f' },

  { tipoNomina: 'C', fecha: '19/09/2020', genero: 'm' },

  { tipoNomina: 'D', fecha: '15/01/2019', genero: 'm' },

];

casosPrueba.forEach(caso => {

  const resultado = calculoMotor(caso.tipoNomina, parseFecha(caso.fecha), caso.genero);

  console.log(`Tipo Nómina: ${caso.tipoNomina}, Fecha desde el primer Empleo: ${caso.fecha}, Género: ${caso.genero}`);

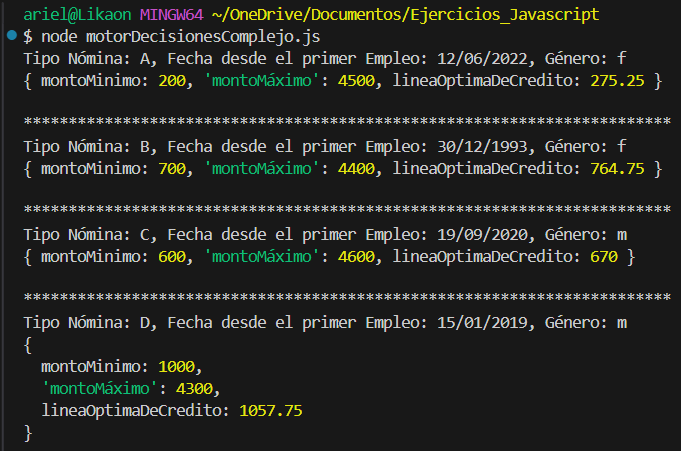
  console.log(resultado);

  console.log('                                                       ');

  console.log('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

});

## Salida en consola por imagen



## Salida por consola en código

Tipo Nómina: A, Fecha desde el primer Empleo: 12/06/2022, Género: f

{ montoMinimo: 200, 'montoMáximo': 4500, lineaOptimaDeCredito: 275.25 }

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Tipo Nómina: B, Fecha desde el primer Empleo: 30/12/1993, Género: f

{ montoMinimo: 700, 'montoMáximo': 4400, lineaOptimaDeCredito: 764.75 }

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Tipo Nómina: C, Fecha desde el primer Empleo: 19/09/2020, Género: m

{ montoMinimo: 600, 'montoMáximo': 4600, lineaOptimaDeCredito: 670 }

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Tipo Nómina: D, Fecha desde el primer Empleo: 15/01/2019, Género: m

{

montoMinimo: 1000,

'montoMáximo': 4300,

lineaOptimaDeCredito: 1057.75

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*