MANUAL JAVASCRIPT: ARREGLOS

Presentado por:

Johan Esteban Cuellar Silva

**Ficha:**

2899747

Análisis y Desarrollo de software.

**Instructor**:

Andres Moreno Collazos

SENA Centro de la Industria, la Empresa y los Servicios

Neiva - Huila

2024

**Tabla de contenido**

Nómina 3

Arreglo de información de empleados3

Funciones 9

Salario9

Subsidio de transporte9

Retención10

Salud11

Pensión11

Arl12

Deducibles 12

Pago total 13

Imprimir 13

Información de empleados, deducibles y sueldo13

Total de pagos 15

**NÓMINA**

En un arreglo contener objetos que representen la información sobre empleados de una empresa y cada objeto tenga propiedades.

* id
* nombres
* apellidos
* cargo
* valorDia
* diasTrabajados

A partir de la información de los empleados hacer una nómina en la que se calcule el subsidio de transporte, retención, salud, pensión y arl para cada empleado y calcular el pago total.

Se debe calcular el total a pagar de salud, pensión, arl y sueldos.

|  |  |
| --- | --- |
| **Nombre del arreglo: nomina** | |
| **Descripción:**  Arreglo que imprime la información de empleados | |
| numRegistros | **Tipo de variable:** Int |
| iterar | **Tipo de variable:** Int |
| id | **Tipo de variable:** Int |
| nombres | **Tipo de variable:** String |
| apellidos | **Tipo de variable:** String |
| cargo | **Tipo de variable:** String |
| valorDia | **Tipo de variable:** Int |
| diasTrabajados | **Tipo de variable:** Int |
| **Código**:  let numRegistros;  let iterar;  let nomina = [];  nomina = [      {          id: 8572036198,          nombres: 'Sofia',          apellidos: 'Johnson Martinez',          cargo: 'Gerente',          valorDia: 120000,          diasTrabajados: 12      },      {          id: 4321657890,          nombres: 'Liam',          apellidos: 'Rodriguez Wilson',          cargo: 'Analista',          valorDia: 100000,          diasTrabajados: 28      },      {          id: 9078563210,          nombres: 'Mia',          apellidos: 'Anderson Perez',          cargo: 'Director',          valorDia: 150000,          diasTrabajados: 5      },      {          id: 6543201987,          nombres: 'Mason',          apellidos: 'Smith Nguyen',          cargo: 'Supervisor',          valorDia: 80000,          diasTrabajados: 20      },      {          id: 1234567890,          nombres: 'Amelia',          apellidos: 'Lee Jackson',          cargo: 'Coordinador',          valorDia: 100000,          diasTrabajados: 3      },      {          id: 9876543210,          nombres: 'Benjamin',          apellidos: 'Gonzalez White',          cargo: 'Ingeniero',          valorDia: 150000,          diasTrabajados: 17      },      {          id: 2468013579,          nombres: 'Charlotte',          apellidos: 'Lopez Ramirez',          cargo: 'Jefe',          valorDia: 120000,          diasTrabajados: 25      },      {          id: 5802468139,          nombres: 'Dylan',          apellidos: 'Martinez Thompson',          cargo: 'Especialista',          valorDia: 100000,          diasTrabajados: 9      },      {          id: 3698521470,          nombres: 'Emma',          apellidos: 'Wilson Garcia',          cargo: 'Consultor',          valorDia: 150000,          diasTrabajados: 14      },      {          id: 7159364820,          nombres: 'Ethan',          apellidos: 'Perez Hernandez',          cargo: 'Jefe',          valorDia: 120000,          diasTrabajados: 14      }  ];  numRegistros = nomina.length;  for (let iterar = 0; iterar < numRegistros; iterar++){      console.log(nomina[iterar]);  }; | |
| **Consola**: | |

**Funciones**

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: salario(pdiasTrabajados, pvalorDia)** | | **Versión: 1.0** |
| **Descripción:**  Función que calcula el valor del salario | | |
| diasTrabajados | **Tipo de variable:** Int | |
| valorDia | **Tipo de variable:** Float | |
| totalSalario | **Tipo de variable:** Float | |
| **Código:**  function salario(pdiasTrabajados, pvalorDia){      let diasTrabajados = pdiasTrabajados;      let valorDia = pvalorDia;      let totalSalario;      totalSalario = diasTrabajados \* valorDia;      return totalSalario;  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: subsidioTra(pdiasTrabajados, pvalorDia)** | | **Versión: 1.0** |
| **Descripción:**  Función que calcula el valor del subsidio de transporte | | |
| diasTrabajados | **Tipo de variable:** Int | |
| valorDia | **Tipo de variable:** Float | |
| salarioMin | **Tipo de variable:** Int | |
| subsidioTrans | **Tipo de variable:** Int | |
| subsidioTransporte | **Tipo de variable:** Int | |
| **Código:**  function subsidioTra(pdiasTrabajados, pvalorDia){      let diasTrabajados = pdiasTrabajados;      let valorDia = pvalorDia;      let salarioMin = 1300000;      let subsidioTrans = salario(diasTrabajados, valorDia);      let subsidioTransporte;      if (subsidioTrans < 2 \* salarioMin){          subsidioTransporte = 120000;      }      else{          subsidioTransporte = 0;      }      return subsidioTransporte;  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: retencion(pdiasTrabajados, pvalorDia)** | | **Versión: 1.0** |
| **Descripción:**  Función que calcula el valor de la retencion | | |
| diasTrabajados | **Tipo de variable:** Int | |
| valorDia | **Tipo de variable:** Float | |
| salarioMin | **Tipo de variable:** Int | |
| valorRet | **Tipo de variable:** Float | |
| valorRetencion | **Tipo de variable:** Float | |
| **Código:**  function retencion(pdiasTrabajados, pvalorDia){      let diasTrabajados = pdiasTrabajados;      let valorDia = pvalorDia;      let salarioMin = 1300000;      let valorRet = salario(diasTrabajados, valorDia);      let valorRetencion;      if (valorRet > 12 \* salarioMin){          valorRetencion = valorRet \* 0.08;      }      else{          if (valorRet > 8 \* salarioMin){              valorRetencion = valorRet \* 0.04;          }          else{              if (valorRet > 6 \* salarioMin){                  valorRetencion = valorRet \* 0.02;              }              else{                  valorRetencion = 0;              }          }      }      return valorRetencion;  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: salud(pdiasTrabajados, pvalorDia)** | | **Versión: 1.0** |
| **Descripción:**  Función que calcula el valor de la salud | | |
| diasTrabajados | **Tipo de variable:** Int | |
| valorDia | **Tipo de variable:** Float | |
| pagoSalud | **Tipo de variable:** Float | |
| **Código:**  function salud(pdiasTrabajados, pvalorDia){      let diasTrabajados = pdiasTrabajados;      let valorDia = pvalorDia;      let pagoSalud;      pagoSalud = salario(diasTrabajados, valorDia) \* 0.12;      return pagoSalud;  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: pension(pdiasTrabajados, pvalorDia)** | | **Versión: 1.0** |
| **Descripción:**  Función que calcula el valor de la pensión | | |
| diasTrabajados | **Tipo de variable:** Int | |
| valorDia | **Tipo de variable:** Float | |
| pagoPension | **Tipo de variable:** Float | |
| **Código:**  function pension(pdiasTrabajados, pvalorDia){      let diasTrabajados = pdiasTrabajados;      let valorDia = pvalorDia;      let pagoPension;      pagoPension = salario(diasTrabajados, valorDia) \* 0.16;      return pagoPension;  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: arl(pdiasTrabajados, pvalorDia)** | | **Versión: 1.0** |
| **Descripción:**  Función que calcula el valor del arl | | |
| diasTrabajados | **Tipo de variable:** Int | |
| valorDia | **Tipo de variable:** Float | |
| pagoArl | **Tipo de variable:** Float | |
| **Código:**  function arl(pdiasTrabajados, pvalorDia){      let diasTrabajados = pdiasTrabajados;      let valorDia = pvalorDia;      let pagoArl;      pagoArl = salario(diasTrabajados, valorDia) \* 0.052;      return pagoArl;  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: deducibles(pdiasTrabajados, pvalorDia)** | | **Versión: 1.0** |
| **Descripción:**  Función que calcula el valor de los deducibles | | |
| diasTrabajados | **Tipo de variable:** Int | |
| valorDia | **Tipo de variable:** Float | |
| pagoDeducibles | **Tipo de variable:** Float | |
| **Código:**  function deducibles(pdiasTrabajados, pvalorDia){      let diasTrabajados = pdiasTrabajados;      let valorDia = pvalorDia;      let pagoDeducibles;        pagoDeducibles = salud(diasTrabajados, valorDia) + pension(diasTrabajados, valorDia) + arl(diasTrabajados, valorDia);      return pagoDeducibles;  } | | |

|  |  |  |
| --- | --- | --- |
| **Nombre de la función: pagoTotal(pdiasTrabajados, pvalorDia)** | | **Versión: 1.0** |
| **Descripción:**  Función que calcula el pago total de una persona | | |
| diasTrabajados | **Tipo de variable:** Int | |
| valorDia | **Tipo de variable:** Float | |
| pagoSueldo | **Tipo de variable:** Float | |
| **Código:**  function pagoTotal(pdiasTrabajados, pvalorDia){      let diasTrabajados = pdiasTrabajados;      let valorDia = pvalorDia;      let pagoSueldo;      pagoSueldo = salario(diasTrabajados, valorDia) + subsidioTra(diasTrabajados, valorDia) - retencion(diasTrabajados, valorDia) - deducibles(diasTrabajados, valorDia);      return pagoSueldo;  } | | |

**Imprimir**

|  |  |
| --- | --- |
| **Nombre del arreglo: tabla** | |
| **Descripción:**  Arreglo que imprime la información de empleados, junto a los deducibles y sueldo total de cada uno | |
| saludNom | **Tipo de variable:** Float |
| pensionNom | **Tipo de variable:** Float |
| arlNom | **Tipo de variable:** Float |
| retencionNom | **Tipo de variable:** Float |
| totalNom | **Tipo de variable:** Float |
| numRegistros | **Tipo de variable:** Int |
| iterar | **Tipo de variable:** Int |
| nomina[iterar].id | **Tipo de variable:** Int |
| nomina[iterar].nombres | **Tipo de variable:** String |
| nomina[iterar].cargo | **Tipo de variable:** String |
| **Código**:  let saludNom = 0;  let pensionNom = 0;  let arlNom = 0;  let retencionNom = 0;  let totalNom = 0;  let numRegistros;  let iterar;  let tabla = [];  for (let iterar = 0; iterar < numRegistros; iterar++){      tabla[iterar] = [          nomina[iterar].id,          nomina[iterar].nombres,          nomina[iterar].cargo,          salario(nomina[iterar].diasTrabajados, nomina[iterar].valorDia),          subsidioTra(nomina[iterar].diasTrabajados, nomina[iterar].valorDia),          salud(nomina[iterar].diasTrabajados, nomina[iterar].valorDia),          pension(nomina[iterar].diasTrabajados, nomina[iterar].valorDia),          arl(nomina[iterar].diasTrabajados, nomina[iterar].valorDia),          retencion(nomina[iterar].diasTrabajados, nomina[iterar].valorDia),          pagoTotal(nomina[iterar].diasTrabajados, nomina[iterar].valorDia),      ]      saludNom = saludNom + salud(nomina[iterar].diasTrabajados, nomina[iterar].valorDia);      pensionNom = pensionNom + pension(nomina[iterar].diasTrabajados, nomina[iterar].valorDia);      arlNom = arlNom + arl(nomina[iterar].diasTrabajados, nomina[iterar].valorDia);      retencionNom = retencionNom + retencion(nomina[iterar].diasTrabajados, nomina[iterar].valorDia);      totalNom = totalNom + pagoTotal(nomina[iterar].diasTrabajados, nomina[iterar].valorDia);  };  console.log(tabla); | |
| **Consola**: | |

|  |  |
| --- | --- |
| **Nombre del arreglo: totalPagos** | |
| **Descripción:**  Arreglo que imprime los totales a pagar en nómina | |
| saludNom | **Tipo de variable:** Float |
| saludNom | **Tipo de variable:** Float |
| pensionNom | **Tipo de variable:** Float |
| arlNom | **Tipo de variable:** Float |
| retencionNom | **Tipo de variable:** Float |
| totalNom | **Tipo de variable:** Float |
| numRegistros | **Tipo de variable:** Int |
| iterar | **Tipo de variable:** Int |
| nomina[iterar].id | **Tipo de variable:** Int |
| nomina[iterar].nombres | **Tipo de variable:** String |
| nomina[iterar].cargo | **Tipo de variable:** String |
| **Código**:  let totalPagos = [saludNom, pensionNom, arlNom, retencionNom, totalNom];  console.log(totalPagos); | |
| **Consola**: | |