

Para mis estudiantes del curso de JavaScript de Coder House. El presente material NO es responsabilidad de Coder House y NO tengo su consentimiento expreso para difundirlo como material propio del programa al que se inscribieron, por ello es que es un aporte de mi para ustedes, como material complementario con la intención de ofrecerles propuestas que considero les podrá ayudar a explorar los temas.

PUC / ASC KRUIOSKI, Saúl A.

## Ejemplos de sentencia condicional

### IF, IF-ELSE, IF-ELSE-IF

Todos y cada uno de los ejemplos abordan conceptos básicos y sencillos del manejo de condicionales, como propuesta complementaria a estos escenarios, te reto a adaptar cada uno de estos añadiendo más complejidad, como por ejemplo el uso de condicionales moleculares en vez de solo atómicas, para usar los conectivos lógicos AND (&&), OR (||) y NEGACIÓN (!). Además la posibilidad de usar los comparadores de relación, como == (igual), === (estrictamente igual), != (distinto), !== (estrictamente distinto), <, <=, >, >= (menor, menor o igual, mayor, mayor o igual) para crear estas proposiciones moleculares.

#### 1. Comprobar si un cliente posee saldo en su cuenta

Suponer que necesitamos validar el saldo de la cuenta bancaria de un cliente, si el cliente posee un saldo positivo (superior a cero) puede continuar con la transacción requerida.

#### Código fuente:

```
let saldo = parseFloat(prompt("¿Cuánto es el saldo de la cuenta del cliente?"));

if (saldo > 0) {
  alert("El usuario cuenta con saldo en la cuenta!");
}
```

**Explicación:** Si **saldo** es mayor que 0, el mensaje "El usuario cuenta con saldo en la cuenta!" se muestra en una ventana flotante. Si **saldo** fuera 0 o menor, no pasaría nada ya que no implementamos alternativas.

Para mis estudiantes del curso de JavaScript de Coder House. El presente material NO es responsabilidad de Coder House y NO tengo su consentimiento expreso para difundirlo como material propio del programa al que se inscribieron, por ello es que es un aporte de mi para ustedes, como material complementario con la intención de ofrecerles propuestas que considero les podrá ayudar a explorar los temas.

PUC / ASC KRUIOSKI, Saúl A.

## 2. Comprobación de edad de una persona para votar

Necesitamos que se verifique mediante la edad de una persona, si esta posee la edad suficiente como para poder votar en las elecciones locales.

### Código fuente:

```
let edad = parseInt(prompt("¿Cuál es la edad de la persona?"));

const edadMinima = 18;

if (edad >= edadMinima) {
    alert("La persona puede votar.");
} else {
    alert("La persona NO puede votar.");
}
```

**Explicación:** Si la **edad** de la persona es 18 o mayor, el mensaje “La persona puede votar.” se muestra en la pantalla. De lo contrario, se muestra un mensaje “La persona NO puede votar.”.

## 3. Mejoramos el caso del saldo en la cuenta

Este ejemplo introduce la estructura *e/se*, permitiendo que el código decida entre dos opciones. Mejoramos la lógica del ejemplo anterior para poder decidir con dos propuestas de flujos diferentes.

Para mis estudiantes del curso de JavaScript de Coder House. El presente material NO es responsabilidad de Coder House y NO tengo su consentimiento expreso para difundirlo como material propio del programa al que se inscribieron, por ello es que es un aporte de mi para ustedes, como material complementario con la intención de ofrecerles propuestas que considero les podrá ayudar a explorar los temas.

PUC / ASC KRUIOSKI, Saúl A.

### Código fuente:

```
let saldo = parseFloat(prompt("¿Cuánto es el saldo de la cuenta del cliente?"));

if (saldo > 0) {
  alert("El usuario cuenta con saldo en la cuenta!");
} else {
  alert("El usuario NO cuenta con saldo en la cuenta!");
}
```

**Explicación:** Si **saldo** es mayor que 0, el mensaje "El usuario cuenta con saldo en la cuenta!" se muestra en una ventana flotante. Si **saldo** fuera 0 o menor, se muestra el mensaje "El usuario NO cuenta con saldo en la cuenta!".

NOTA: Más adelante mejoraremos este algoritmo para hacerlo reutilizable y poder usarlo para diferentes tipos de validaciones.

### 4. Calcular costo de envío

Necesitamos calcular el costo de envío de un paquete, y este costo depende de la distancia recorrida desde el lugar de despacho al de recepción. Para ello distancias entre 0-500km el costo es de \$ 5.000, entre 500-1000km es de \$ 7.250 y superior a este es \$ 10.000.

### Código fuente:

```
const distanciaMinima = 500;
const distanciaMedia = 1000;

let distancia = parseFloat(prompt("¿Cuál es la distancia medida en kilometros?"));
```

Para mis estudiantes del curso de JavaScript de Coder House. El presente material NO es responsabilidad de Coder House y NO tengo su consentimiento expreso para difundirlo como material propio del programa al que se inscribieron, por ello es que es un aporte de mi para ustedes, como material complementario con la intención de ofrecerles propuestas que considero les podrá ayudar a explorar los temas.

PUC / ASC KRUIOSKI, Saúl A.

```
let costoDelEnvio;

if (distancia <= distanciaMinima) {
  // Lo usaremos para representar la distancia recorrida de hasta un máximo de
  500km
  costoDelEnvio = parseFloat(5000);
} else if (distancia <= distanciaMedia) {
  // Lo usaremos para representar una distancia recorrida superior a 500km y
  hasta 100km
  costoDelEnvio = parseFloat(7250);
} else {
  // Si la distancia recorrida supera los 1000km
  costoDelEnvio = parseFloat(10000);
}

alert("Costo del envío para " + distancia + " kilometros es $ " +
costoDelEnvio.toFixed(2));
```

**Explicación:** El programa ahora tiene tres opciones posibles. Evalúa si **distancia** es máximo de 500km define un **costo** de \$ 5.000, en cambio si se encuentra entre 500km y máximo 1000km el **costo** se configura en \$ 7.250, por lo contrario, si supera los 1000km el **costo** se establece en \$ 10.000.

## 5. Verificar si una tienda está abierta o cerrada según la hora actual

Este ejemplo usa una condición if para verificar si la hora actual está dentro del horario de atención de una tienda y dependiendo de ello nos arrojará un mensaje indicándonos esto.

Para mis estudiantes del curso de JavaScript de Coder House. El presente material NO es responsabilidad de Coder House y NO tengo su consentimiento expreso para difundirlo como material propio del programa al que se inscribieron, por ello es que es un aporte de mi para ustedes, como material complementario con la intención de ofrecerles propuestas que considero les podrá ayudar a explorar los temas.

PUC / ASC KRUIOSKI, Saúl A.

### **Código fuente:**

```
let horaActual = parseInt(prompt("Ingrese la hora actual expresada en el formato de 24hs. Es decir puede ingresar 00 a 24"));

const horaApertura = 9;
const horaCierre = 18;

if (horaActual >= horaApertura && horaActual <= horaCierre) {
    alert("La tienda está abierta.");
} else {
    alert("La tienda está cerrada.");
}
```

**Explicación:** Si la variable horaActual está entre las 9hs y las 18hs, se informa que la tienda está abierta; de lo contrario, informa que está cerrada.

### **6. Comprobar si una variable contiene texto**

Desarrollaremos una forma de identificar a nuestro usuario dentro de la aplicación, y si este está identificado (con algún nombre de usuario por ejemplo) le permitiremos acceso a algunas características del sitio que un usuario no registrado no puede acceder.

Para mis estudiantes del curso de JavaScript de Coder House. El presente material NO es responsabilidad de Coder House y NO tengo su consentimiento expreso para difundirlo como material propio del programa al que se inscribieron, por ello es que es un aporte de mi para ustedes, como material complementario con la intención de ofrecerles propuestas que considero les podrá ayudar a explorar los temas.

PUC / ASC KRUIOSKI, Saúl A.

### Código fuente:

```
let nombreUsuario = prompt("Nombre del usuario")

if (nombreUsuario !== "") {
  alert("Hola, " + nombreUsuario + "! Tienes acceso completo a la aplicación.");
} else {
  alert("Hola, desconocido! Su acceso es limitado, regístrese para una experiencia más amplia.");
}
```

**Explicación:** Si la variable **nombreUsuario** contiene el texto cualquiera (es decir *no está vacío*), se le permite el acceso; de lo contrario, se le da acceso limitado a las funcionalidades.

### 7. Determinar el nivel de descuento según el monto invertido en la compra

Requerimos proporcionar descuentos a aquellos compradores de una tienda online dependiendo del monto total de sus compras.

### Código fuente:

```
let montoTotal = parseFloat(prompt("Ingrese el monto total de la compra"));
let descuento;

const consumoMinimo = 5000;
const consumoMedio = 15000;
const consumoAlto = 50000;

if (montoTotal <= consumoMinimo) {
  descuento = 0;
```

Para mis estudiantes del curso de JavaScript de Coder House. El presente material NO es responsabilidad de Coder House y NO tengo su consentimiento expreso para difundirlo como material propio del programa al que se inscribieron, por ello es que es un aporte de mi para ustedes, como material complementario con la intención de ofrecerles propuestas que considero les podrá ayudar a explorar los temas.

PUC / ASC KRUIOSKI, Saúl A.

```
} else if (montoTotal <= consumoMedio) {  
    descuento = 5;  
} else if (montoTotal <= consumoAlto) {  
    descuento = 10;  
} else {  
    descuento = 15;  
}  
  
alert("Descuento otorgado " + descuento + "%");
```

**Explicación:** Si el cliente realiza una compra con un monto igual o inferior a \$ 5.000,00, no hay descuento. Hasta \$ 15.000,00, hay un 5% de descuento; hasta \$ 50.000,00, un 10%; y para montos superiores a este último, un 15%. Esto muestra cómo aplicar reglas de negocio de manera sencilla con if-else-if.

## 8. Otorgar incentivos

Suponer que en una fábrica hay incentivos otorgados a aquel personal que supera un umbral de producción, para este caso vamos a otorgar un incentivo de \$ 200.000,00 a aquellos individuos que superen la producción mínima de 30.000 unidades.

### Código fuente:

```
let producido = parseInt(prompt("Ingrese el número de unidades producidas"));  
let incentivo = 200000;  
  
const produccionMinima = 30000;  
  
if (producido > produccionMinima) {
```

Para mis estudiantes del curso de JavaScript de Coder House. El presente material NO es responsabilidad de Coder House y NO tengo su consentimiento expreso para difundirlo como material propio del programa al que se inscribieron, por ello es que es un aporte de mi para ustedes, como material complementario con la intención de ofrecerles propuestas que considero les podrá ayudar a explorar los temas.

PUC / ASC KRUJOSKI, Saúl A.

```
    alert("El personal recibe un incentivo de $ " + incentivo.toFixed(2) + " por la  
    producción de " + producido.toString() + " unidades.");  
}
```

**Explicación:** si la cantidad de unidades producidas por un personal supera al mínimo, que en este caso está configurado en 30.000 unidades, entonces recibirá un bono de incentivo de \$ 200.000,00, en caso contrario no recibe incentivo.

## 9. Moderador de lenguaje

Trabajamos en un sistema de chat de un juego en línea y debemos implementar un control del vocabulario dentro de las conversaciones entre los participantes, para comenzar un prototipo sencillo, vamos a considerar un mensaje y una palabra de un diccionario de palabras no aptas.

### Código fuente:

```
const frase = prompt("Ingrese la frase");  
const palabra = prompt("Palabra no permitida");  
  
const resultado =  
frase.trim().toLowerCase().includes(palabra.trim().toLowerCase());  
if (resultado) {  
    alert("El mensaje cuenta con vocabulario no permitido en la conversación!");  
}
```

**Explicación:** el usuario ingresa una frase que se supone es el mensaje que se enviará en caso de pasar el algoritmo de validación, lo siguiente es probar el algoritmo con una palabra que sabemos es una de la lista de no permitidas. Entonces, si la palabra prohibida se encuentra presente dentro de la frase se lo



Para mis estudiantes del curso de JavaScript de Coder House. El presente material NO es responsabilidad de Coder House y NO tengo su consentimiento expreso para difundirlo como material propio del programa al que se inscribieron, por ello es que es un aporte de mi para ustedes, como material complementario con la intención de ofrecerles propuestas que considero les podrá ayudar a explorar los temas.

PUC / ASC KRUIOSKI, Saúl A.

informaremos al usuario, podemos optar por mecanismos de penalización, de rankings, reemplazar la palabra por caracteres ilegibles, etc.

## 10. Autenticación de sistema informático

Uno de los casos más generales y el más útil en la mayoría de los sistemas, es la incorporación de validación de usuarios mediante la autenticación de credenciales para el ingreso. Por ello es que debemos contar con un algoritmo para validar nombre de usuario y contraseña ingresados por el usuario contra los almacenados en la base de datos.

### Código fuente:

```
const userNameProvided = prompt("Ingrese el nombre de usuario");
const passwordProvided = prompt("Ingrese la contraseña de acceso");
const savedPassword = "123456789";
const savedUserName = "Chaman";

if ((passwordProvided !== savedPassword) || (userNameProvided !==
savedUserName)) {
    alert("Las credenciales proporcionadas no son correctas");
}
```

**Explicación:** primero solicitamos al usuario su nombre de usuario (*nickname*) y la contraseña (*password*), y luego contrastamos estos sin aplicar cambios en lo absoluto (*no necesitamos hacer trim(), toLowerCase() u otros*) con los recuperados de la base de datos, aquí podemos abordar el algoritmo de diferentes maneras, en el ejemplo se decide ir por el caso donde alguno de estos dos no coincida e informar del error.

Para mis estudiantes del curso de JavaScript de Coder House. El presente material NO es responsabilidad de Coder House y NO tengo su consentimiento expreso para difundirlo como material propio del programa al que se inscribieron, por ello es que es un aporte de mi para ustedes, como material complementario con la intención de ofrecerles propuestas que considero les podrá ayudar a explorar los temas.

PUC / ASC KRUIOSKI, Saúl A.

## 11. Establecer tarifas de estacionamiento

Podemos imaginarnos trabajando para el municipio de la ciudad donde resides, y allí deciden incorporar la automatización del cobro de tarifas de estacionamiento mediante una app mobile. Para ello sabemos que si el tiempo de estacionamiento es inferior a 1 hora, se cobra una tarifa mínima; si es entre 1 y 3 horas, la tarifa por hora es un valor medio; y en el caso de superar las 3 horas, se aplicará una tarifa más elevada.

### Código fuente:

```
const horas = parseInt(prompt("Ingrese la cantidad de horas de
estacionamiento"));
const minimumAmount = 160;
const averageAmount = 240;
const higherAmount = 280;

let monto = 0;

if (horas < 1) {
  monto = minimumAmount;
} else if (horas > 1 && horas < 3) {
  monto = averageAmount * horas;
} else {
  monto = higherAmount * horas;
}

alert("El monto total de pago por el uso de estacionamiento es de $ " +
monto.toFixed(2));
```

Para mis estudiantes del curso de JavaScript de Coder House. El presente material NO es responsabilidad de Coder House y NO tengo su consentimiento expreso para difundirlo como material propio del programa al que se inscribieron, por ello es que es un aporte de mi para ustedes, como material complementario con la intención de ofrecerles propuestas que considero les podrá ayudar a explorar los temas.

PUC / ASC KRUIOSKI, Saúl A.

**Explicación:** el algoritmo presente puede ser optimizado y reducido en una expresión más pequeña, sin embargo para el nivel de detalle que manejamos en este momento preferimos mantener un grado de abstracción reducido. Si el usuario no llega a cumplir con al menos 1 hora de estacionamiento pagará la tarifa mínima, hasta 3 horas se multiplica la cantidad de horas por el monto medio y si supera las 3 horas, se multiplica por la tarifa máxima.

**NOTA:** Podemos a futuro mejorar este algoritmo para calcular el total, fraccionando la tarifa en los rangos definidos y calcular valores más “coherentes”.

## 12. Control de stock con alertas

Podemos encontrar el escenario donde debemos incorporar alertas al área de reposición de stock de una empresa. Para ello supongamos que debemos monitorear constantemente el manejo de mercadería entre el área de ventas y reposición, y tenemos parametrizado un umbral donde debemos alertar si es necesario reponer un producto y cuantas unidades son necesarias para alejarnos de este umbral.

### Código fuente:

```
let currentStock = 100;
const unitsSold = parseInt(prompt("Cantidad de unidades vendidas"));
currentStock = currentStock - unitsSold; // Actualización del stock
const threshold = 31; // Umbral
const replacementMargin = 20; // En cuántas unidades debemos superar para
evitar una alerta temprana

if (currentStock < threshold) {
  const quantity = (threshold - currentStock) + replacementMargin;
```

Para mis estudiantes del curso de JavaScript de Coder House. El presente material NO es responsabilidad de Coder House y NO tengo su consentimiento expreso para difundirlo como material propio del programa al que se inscribieron, por ello es que es un aporte de mi para ustedes, como material complementario con la intención de ofrecerles propuestas que considero les podrá ayudar a explorar los temas.

PUC / ASC KRUIOSKI, Saúl A.

```
alert("Es necesario reponer el producto en " + quantity.toString() + " unidades.");  
}
```

**Explicación:** en este caso el algoritmo no es complejo, si es un poco más complicado el cálculo de la cantidad a reponer. Comenzamos identificando la cantidad de unidades vendidas (*unitsSold*) y la cantidad de unidades disponibles para la venta (*currentStock*). Luego debemos saber con cuántas unidades nos quedamos luego de la venta, es decir debemos hacer una diferencia entre la cantidad actual - las unidades vendidas (*currentStock - unitsSold*), lo que nos dará como resultado una actualización de la cantidad actual (*currentStock*), ya sabemos cuantas unidades nos quedan para la venta, debemos comprobar si esta está por debajo del umbral (*threshold*). Si ocurre que estamos por debajo del umbral, se procederá a informar la cantidad de unidades a reponer, para ello debemos hacer el siguiente cálculo: identificar la cantidad de unidades necesarias para alcanzar al menos el umbral y evitar otra alerta (*threshold - currentStock*), luego a este número le debemos sumar una cantidad considerada para cada escenario, para evitar una alerta muy cercana a las operaciones diaria nuevamente y por ello le sumamos un margen (*replacementMargin*).

### 13. Control de temperatura en electrodomésticos

Imaginar que debemos programar placas controladoras para electrodomésticos con la capacidad de eficientizar consumos de electricidad, en este ejemplo intentaremos mejorar el mantenimiento de alimentos frescos dentro de un refrigerador, para ello sabemos que si la temperatura medida del interior supera los 8°C, el compresor debe activarse; si se encuentra entre 2°C y 8°C, se mantiene a baja capacidad; y de lo contrario si baja de 2°C, debe apagarse.

Para mis estudiantes del curso de JavaScript de Coder House. El presente material NO es responsabilidad de Coder House y NO tengo su consentimiento expreso para difundirlo como material propio del programa al que se inscribieron, por ello es que es un aporte de mi para ustedes, como material complementario con la intención de ofrecerles propuestas que considero les podrá ayudar a explorar los temas.

PUC / ASC KRUIOSKI, Saúl A.

### **Código fuente:**

```
const currentTemperature = parseFloat(prompt("Temperatura del interior del
refrigerador"));
const averageCapacity = 8;
const off = 2;

if (currentTemperature > averageCapacity) {
    alert("ENCENDER compresor a máxima capacidad.");
} else if (currentTemperature > off) {
    alert("MANTENER compresor a capacidad regular.");
} else {
    alert("APAGAR compresor");
}
```

**Explicación:** suponemos que hay un sistema que mide periódicamente la temperatura y que ejecuta el algoritmo, donde si la temperatura medida supera los 8°C se enciende el compresor al máximo de su capacidad, en cambio si se encuentra entre 2°C y 8°C se mantiene a capacidad mínima y si baja de 2°C debe apagarse.

### **14. Etiquetado de calificaciones**

Recordar cuando en la primaria las calificaciones no eran números sino etiquetas, donde si la calificación es menor a 5, se obtiene un "Reprobado"; mayor a 5 y menor a 8, "Suficiente"; y de 8 hasta el 10 obtenias un "Sobresaliente". bueno, tomaremos esta escala para “traducir” las calificaciones de estudiantes en etiquetas fácilmente identificables.

Para mis estudiantes del curso de JavaScript de Coder House. El presente material NO es responsabilidad de Coder House y NO tengo su consentimiento expreso para difundirlo como material propio del programa al que se inscribieron, por ello es que es un aporte de mi para ustedes, como material complementario con la intención de ofrecerles propuestas que considero les podrá ayudar a explorar los temas.

PUC / ASC KRUIOSKI, Saúl A.

### **Código fuente:**

```
const qualification = parseInt(prompt("Calificación del estudiante"));

if (qualification >= 8) {
    alert("SOBRESALIENTE");
} else if (qualification >= 5) {
    alert("SUFICIENTE");
} else {
    alert("REPROBADO");
}
```

**Explicación:** aquí podemos suponer que estamos recorriendo una lista con los nombres de los estudiantes y momos uno a una la calificación obtenida y la reemplazamos con el valor que devolveremos en el lugar del alert, por eso es que para un estudiante con menos de 5 puntos veríamos junto a su nombre la etiqueta “REPROBADO”, en aquellos que tengan entre 5 y menos de 8 veríamos “SUFICIENTE” y aquellos con 8 o más “SOBRESALIENTE”.

## **15. Domótica aplicada - Activación de luces inteligentes**

Igual que con el ejemplo del refrigerador, aquí podemos suponer que programamos una placa de circuitos para controlar las luces de la vivienda. Para esto necesitamos sensores instalados que miden la cantidad de luz ambiental, y con este valor desarrollar lo siguiente, si la luz ambiental está por debajo de 150lx (*lux*), se deben encender las luces; si está por encima de este umbral, las luces deben apagarse.

Para mis estudiantes del curso de JavaScript de Coder House. El presente material NO es responsabilidad de Coder House y NO tengo su consentimiento expreso para difundirlo como material propio del programa al que se inscribieron, por ello es que es un aporte de mi para ustedes, como material complementario con la intención de ofrecerles propuestas que considero les podrá ayudar a explorar los temas.

PUC / ASC KRUIOSKI, Saúl A.

### **Código fuente:**

```
const luxLevel = parseInt(prompt("Cuál es el nivel de luminosidad ambiental exterior medido en lux"));
const threshold = 150; // Umbral

if (luxLevel < threshold) {
    alert("Encender");
} else {
    alert("Apagar");
}
```

**NOTA:** Un lux es una unidad de medida de la iluminancia, que indica la cantidad de luz que llega a una superficie. Específicamente, un lux equivale a un lumen por metro cuadrado (lm/m<sup>2</sup>). Mide cuánta luz visible incide sobre una superficie en un área específica.

**Explicación:** según la investigación 150lx son suficientes en una zona residencial para considerar encender tanto las luces del jardín como las del interior de la vivienda, por ello es que, si el sensor de luminosidad ambiental instalado en el exterior de la vivienda arroja un valor por debajo de este umbral, se emitirá la señal de “Encendido” de lo contrario de “Apagado”.

### **16. Ajustes de consumo según control de niveles de batería**

Forma parte de un proyecto de desarrollo para el Sistema Operativo de un dispositivo móvil, y uno de los subsistemas es el control y gestión energética del dispositivo. Para esto debemos considerar el nivel de carga de la batería, si este está por debajo del 15%, se debe poner el dispositivo en modo ahorro de batería; si está entre 15% y 20%, se debe mantener una alerta de batería baja; si es superior no se debe hacer nada.

Para mis estudiantes del curso de JavaScript de Coder House. El presente material NO es responsabilidad de Coder House y NO tengo su consentimiento expreso para difundirlo como material propio del programa al que se inscribieron, por ello es que es un aporte de mi para ustedes, como material complementario con la intención de ofrecerles propuestas que considero les podrá ayudar a explorar los temas.

PUC / ASC KRUIOSKI, Saúl A.

### Código fuente:

```
const batteryLevel = parseInt(prompt("¿Cuál es el nivel de la batería del dispositivo?"));
const criticalLevel = 15;
const lowLevel = 20;

if (batteryLevel < criticalLevel) {
    alert("Poner el dispositivo en modo ahorro de energía.");
} else if (batteryLevel <= lowLevel) {
    alert("Alerta de batería baja!");
}
```

**Explicación:** suponemos que hay un subsistema que toma la medida de la batería cada cierto tiempo, pero a ejemplos prácticos nosotros ingresamos el valor a través del prompt. Luego evaluamos si este está por debajo del 15% se emite la señal de poner al dispositivo en modo ahorro, y si está entre 15% y 20% se emite una alerta de batería baja, si este valor se encuentra por encima de los 20% no se toman acciones.

## 17. Validación de permisos de acceso

Imaginar cómo en un documento online, cuando trabajas con diferentes colaboradores, cada uno de estos tiene un rol que le permite realizar ciertas acciones sobre él. Si el rol del usuario es "administrador", tiene acceso total; si es "editor", posee acceso parcial; si es "lector", solo puede ver contenido.



Para mis estudiantes del curso de JavaScript de Coder House. El presente material NO es responsabilidad de Coder House y NO tengo su consentimiento expreso para difundirlo como material propio del programa al que se inscribieron, por ello es que es un aporte de mi para ustedes, como material complementario con la intención de ofrecerles propuestas que considero les podrá ayudar a explorar los temas.

PUC / ASC KRUIOSKI, Saúl A.

### **Código fuente:**

```
const role = prompt("¿Cuál es el rol del usuario?");

if (role.trim().toLowerCase() === "administrador") {
  alert("Tienes acceso total.");
} else if (role.trim().toLowerCase() === "editor") {
  alert("Tienes acceso limitado");
} else if (role.trim().toLowerCase() === "lector") {
  alert("Solo puedes leer el contenido.");
} else {
  alert("Rol de usuario no identificado");
}
```

**Explicación:** si el rol del usuario recuperado es “administrador” entonces se emitirá la señal de control total sobre el documento, no tiene restricción alguna, por el contrario si es “editor”, podrá realizar algunas acciones y otras no, al final tenemos el último rol definido como “lector”, esté solo le permitirá leer el documento, sin embargo contemplamos la posibilidad de un rol que no se encuentre en la lista, por ello es que informaremos que el rol no está definido y por lo tanto no le daríamos ningún tipo de acceso al contenido o a las funcionalidades.

### **18. Gestión de prioridad en una fila de espera**

Nos encontramos regularmente con este tipo de situaciones, en filas de bancos por ejemplo, suponiendo que al momento de solicitar un turno a través del tótem de la entrada, el sistema puede identificar la edad del cliente y si posee condiciones que requieren incrementar su prioridad de atención. Para el algoritmo diremos que una persona tiene prioridad de paso si es mayor de 60 años o tiene condiciones particulares; de lo contrario, se mantiene en el orden regular.

Para mis estudiantes del curso de JavaScript de Coder House. El presente material NO es responsabilidad de Coder House y NO tengo su consentimiento expreso para difundirlo como material propio del programa al que se inscribieron, por ello es que es un aporte de mi para ustedes, como material complementario con la intención de ofrecerles propuestas que considero les podrá ayudar a explorar los temas.

PUC / ASC KRUIOSKI, Saúl A.

### **Código fuente:**

```
const age = parseInt(prompt("Edad de la persona"));
const hasParticularConditions = confirm("Posee condiciones particulares");
const ageLimit = 60;

if (age > ageLimit || hasParticularConditions) {
    alert("Poner a la persona en la lista de prioridad.");
} else {
    alert("Poner a la persona en la lista habitual.");
}
```

**Explicación:** cuando la persona se acerque al tótem de la entrada e ingrese su DNI, el sistema del banco la identificará, y entre la información recuperada sabrá su edad y si posee condiciones particulares que la vuelven una persona con prioridad dentro de la cola de espera para su atención. Por lo tanto, si la persona posee más de 60 años o bien posee condiciones particulares, o si pasan estas dos cosas juntas, entonces la persona es movida a la lista de prioridad, de lo contrario se la mueve a la lista de espera habitual.

### **19. Calcular monto de impuestos según ingresos declarados**

Simular el funcionamiento de una agencia de recaudación de impuestos, y para ello suponemos que un contribuyente con ingresos menores a \$100.000, no paga impuestos; entre \$ 100.000 y \$ 500.000, paga un 4.6%; más de \$500.000 y menos de 1.000.000, un 6% y por encima del \$ 1.000.000 un total de 8%. Se requiere devolver el monto total de impuestos a ser solicitados según el ingreso declarado.

Para mis estudiantes del curso de JavaScript de Coder House. El presente material NO es responsabilidad de Coder House y NO tengo su consentimiento expreso para difundirlo como material propio del programa al que se inscribieron, por ello es que es un aporte de mi para ustedes, como material complementario con la intención de ofrecerles propuestas que considero les podrá ayudar a explorar los temas.

PUC / ASC KRUIOSKI, Saúl A.

### Código fuente:

```
const amount = parseFloat(prompt("Ingrese el monto de ingresos mensual del contribuyente"));
const maxLimit = 1000000;
const mediumLimit = 500000;
const lowLimit = 100000;
let taxAmount = 0;

if (amount > maxLimit) {
    taxAmount = amount * 0.080;
} else if (amount > mediumLimit) {
    // ¿Por qué quí no puse el >= como en el caso siguiente?
    taxAmount = amount * 0.060;
} else if (amount >= lowLimit) {
    taxAmount = amount * 0.046;
}

alert("El monto de impuestos a abonar es $ " + taxAmount.toFixed(2) + ".");
```

**Explicación:** para simular el proceso, necesitaremos contar con el monto de ingresos, los límites que establecen los rangos y una variable para calcular y almacenar el valor de impuestos a abonar. Si el monto de ingresos supera el millón, entonces se calculará el impuesto, multiplicando el monto por 0.080 (este 0.080 es equivalente a escribir 8/100), si el monto supera el medio millón pero no alcanza el millón, entonces se calculará el impuesto con 6/100, luego si el monto es superior a \$ 100.000,00 y hasta el medio millón, se calcula el impuesto con 4.6/100, por el contrario si el monto no supera los \$ 100.000,00 entonces el impuesto se mantiene en cero como de inicializo.