

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 Arrays

El siguiente apartado toma algunos de los ejercicios propuestos en “*Ejemplos de sentencias Cíclicas e iterativas*”. Aquí el desafío es que aún no usamos objetos.

### 1.3 Lista de pendientes

Simular el proceso de carga de un array con 5 elementos, cada uno de estos elementos representa a una tarea que debes ejecutar. Una vez cargados todos los elementos, recorrer el array mostrando cada uno de los ítems de la lista por pantalla.

### 2.6 Calcular promedios

- a) Mejorar el algoritmo propuesto originalmente convirtiéndolo en una función que procese N calificaciones y devuelve un array con las notas almacenadas en él. Donde N es un número entero que representa a la cantidad de calificaciones a cargar.
- b) Luego, desarrollar una función que reciba dicho array y devuelve un número que representa al promedio de las calificaciones, el algoritmo de este proceso debe ser dinámico (no debe importar la cantidad de elementos que hay dentro del array).

### 3.7 Calcular gastos

Mejorar el algoritmo originalmente propuesto y permitarnos procesar un array que contiene todos los ingresos y egresos y poder devolver la suma del debe, el haber y el saldo de todas las cuentas. Recordemos que el debe y el haber son elementos de una partida doble en contabilidad, donde el debe son el incremento de activos y el haber el incremento de pasivos.

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.

#### **4.11 Simular consumo sobre débitos**

Plantear un algoritmo para el control sobre el consumo sobre una tarjeta de débito, donde vamos a permitir al usuario ingresar un consumo siempre que cuente con saldo disponible para consumirlo. Para ello se insertará dentro del array una entrada con el valor siempre que pueda consumirse este de la cuenta, para esto se debe saber en todo momento cuánto saldo tenemos disponible en la cuenta.

#### **5.18 Temperaturas**

- a) Continuando con el ejemplo original, vamos a realizar los mismos cálculos, pero esta vez de forma dinámica, ya que vamos a construir una función que reciba el array con las temperaturas y nos va a devolver un objeto con la spropiedades, promedio, la temperatura más baja, el día que ocurrió, la temperatura más alta y el día que ocurrió.
- b) Permitir obtener un array con un periodo de días específico, por ejemplo si quiero tener las temperaturas entre los días 10 y 20 de un registro.

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.

El siguiente apartado toma algunos de los ejercicios propuestos en “*Ejemplos de Objetos*”.

## 1. Somos estudiantes

Continuando con el ejemplo propuesto, vamos a mejorar la estructura de datos y funcionalidad de la clase original definida para el ejercicio y vamos a reemplazar la lista de carreras que era originalmente un string por un array que contendrá todas las carreras.

La clase redefinida deberá tener métodos para las siguientes operaciones:

- a) Agregar una nueva carrera a la lista.
- b) Comprobar si el alumno se encuentra inscripto a una carrera o no.
- c) Listar todas las carreras que posee el estudiante.

## 2. Vamos al curso de JavaScript

De la misma forma que el ejercicio anterior, vamos a redefinir la estructura y funcionalidad de la clase original, y reemplazamos la lista de temario que era un string, por una estructura de array de objetos.

La clase redefinida deberá tener las siguientes nuevas estructuras y métodos:

- a) Definir la estructura para el temario debe contener un título, el número de la clase, una descripción, y un array con la lista que describa los objetivos del temario.
- b) Permitir poder agregar temas que no sean repetidos al curso.
- c) Permitir eliminar un temario específico a partir de su título.
- d) Obtener una lista con todos los objetivos alcanzados durante un curso si se dieran todos los temas propuestos.
- e) Crear un algoritmo que aproveche todas estas nuevas características.

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.

### 3. Cuenta bancaria

- a) Redefinir la propuesta del objeto original para que este contenga un historial de movimientos, lo cual haremos con un array de objetos, para ello deberás definir una nueva estructura con los siguientes datos: fecha, hora, monto (positivo o negativo), descripción.
- b) Crear un algoritmo para crear una cuenta con sus datos y simular la carga de movimientos dentro de la cuenta.
- c) Complementando esto, la clase debe permitirnos obtener una lista de movimientos a partir de un periodo comprendido por una fecha de inicio y una fecha de fin.

### 4. Producto

- a) Definir el algoritmo que permita cargar de forma dinámica una serie de productos con sus correspondientes datos.
- b) Además, permitirnos crear un nuevo array a partir de la lista de productos y una categoría, el algoritmo debe ser implementado en una función y devolver en un nuevo array los productos que coincidan con la categoría proporcionada.
- c) Desarrollar una función que reciba la lista de productos, el nombre de un producto o código si prefieres, y un número, este número representa la cantidad de unidades que se deben sumar al stock actual del producto.

### 5. Enviemos un mensaje

Partir de la idea original, generar un algoritmo que reciba un array con palabras claves que ingrese un usuario, y una lista de mensajes. Vamos a simular la búsqueda de palabras claves dentro del asunto y el cuerpo de los mensajes.

El algoritmo debe tomar la lista de palabras y evaluar si alguna de estas existe dentro del asunto del mensaje o su contenido (*cuerpo del mensaje*), si esto ocurre

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.

se debe mostrar como resultado en un nuevo array que se genera a partir del original.

## 6. Recibo o facturación

Partiendo de la definición original de este objeto, vamos a añadir una propiedad que representará a los ítems de la factura, es decir los productos y su correspondiente detalle.

- a) Para conseguirlo debes definir una estructura que define al ítem de la factura como: nombre del producto, precio unitario, cantidad, porcentaje de descuento (un número entero entre 0 y 100) y valor del descuento en moneda.
- b) Crear el algoritmo necesario para cargar los items a la factura, proporcionando todos los datos necesarios.
- c) Crear un método en la clase del ítem de la factura que me permita calcular el subtotal del ítem (precio unitario por cantidad) menos el descuento si lo hay.
- d) Y con el método creado anteriormente, crear un método en la clase factura que calcule el total del recibo.

## 7. Cita médica

Crear los algoritmos necesarios para manejar un array con la lista de citas de pacientes del nosocomio:

- a) Cargar citas en la lista.
- b) Listar todas las citas por pantalla.
- c) Permitirnos eliminar una cita a partir del nombre de un paciente.
- d) Obtener una lista con todas las citas de un médico en particular.
- e) Trasladar todos los pacientes de un médico a otro.

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.

## **8. Agenda de Contactos**

- a) Modificar la definición original para que las etiquetas se representen con un array y crear un método dentro de la clase que permita obtener la lista de etiquetas unificadas en una cadena de texto.
- b) Suponer que nuestro celular tiene la posibilidad de listar contactos a partir del nombre de una o varias etiquetas, crear un método que permita obtener una lista de contactos en un nuevo array siempre que este contacto tenga al menos una de las etiquetas proporcionadas.

## **9. Biblioteca Personal**

- a) Crear un algoritmo que permita cargar un array con los objetos representados por libros.
- b) Permitirnos tomar el nombre de un autor y generar una nueva lista de libros en un nuevo array.
- c) Poder obtener listas combinadas de varios autores a partir de un algoritmo que reciba los nombres de los autores en un array y la lista de libros en otro array, se deben procesar y devolver un nuevo array con libros de los autores indicados.

## **10. Reserva en un Restaurante**

Suponer que uno de los ambientes del restaurante sufrió un evento que lo dejó inhabilitado para su uso, crear un algoritmo que me permita eliminar todas las reservas que coincidan con este ambiente.

## **11. Contenido de streaming**

Suponer que la interfaz de nuestra cuenta de streaming nos permite eliminar registros de una posición inicial a una final, escribir el algoritmo para llevar a cabo dicha tarea.

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.

## **14. Calendario de Eventos**

Redefinir la clase para que los invitados ahora sean un array con los nombres y eliminar la propiedad cantidad, ya que este lo calcularemos a partir de la lista de invitados.

- a) Generar un algoritmo que me permita buscar a un invitado dentro de la lista a partir de su nombre parcial si fuera necesario.
- b) Supongo que sabemos que un grupo de invitados canceló su asistencia al evento y para corregirlo, debemos implementar un método que nos permita recibir la lista de invitados que canceló y eliminarlos a todos estos de la lista de invitados original.

## **15. Menú de Restaurante**

Mejorar la definición del objeto original cambiando a los ingredientes y a la preparación por arrays.

- a) Crear un algoritmo que me permita obtener la lista de ingredientes en una sola cadena de texto para imprimirla como texto plano.
- b) Desarrollar un algoritmo que me permita obtener una lista de platos a partir de una lista de ingredientes, esto se debe procesar y devolver siempre que el plato contenga en su lista de ingredientes al menos uno de los ingredientes proporcionados al algoritmo.