Actividades unidad 3: Funciones Callback, Closure, HOF, Generadoras y expresiones regulares

Preguntas Teóricas

1. Funciones clausura (closures):

a) Explica con tus palabras qué es una función clausura en JavaScript.
b) ¿Por qué las variables dentro de una clausura continúan existiendo después de ejecutar la función que las define?
c) Cita un ejemplo donde sería útil usar una clausura.

2. Funciones callback:

a) Define qué es una función callback y su utilidad en programación.
b) ¿Cuál es la diferencia entre pasar una función como callback y llamarla directamente dentro de otra función?

3. Funciones de orden superior (HOF):

a) ¿Qué significa que una función sea de orden superior?
b) ¿Qué ventajas tienen las funciones de orden superior frente a funciones normales?

4. Funciones generadoras:

a) Explica qué es una función generadora y qué significa la palabra clave yield.
b) ¿Cómo se pueden obtener los valores de una función generadora?

5. Expresiones regulares:

a) ¿Qué es una expresión regular y para qué se utiliza?
b) Explica qué hacen los flags g, i y m.
c) ¿Cuál es la diferencia entre los métodos test() y exec() de un objeto RegExp?

Ejercicios Prácticos

6. Ejercicio 1: Clausuras

Dada la siguiente función:

```
const contador = (function() {
  let cuenta = 0;
  return function() {
    cuenta++;
    return cuenta;
  };
})();
```

Apartados:

a) Explica qué hace la función contador.

b) Muestra en consola tres llamadas consecutivas a contador() y explica el resultado.

c) Modifica la función para que acepte un valor inicial para cuenta.

7. Ejercicio 2: Callbacks

Dado el siguiente array:

```
let nombres = ["Ana", "Luis", "Marta"];
```

Apartados:

a) Crea una función modificarArray(array, callback) que añada un nombre al array y luego ejecute la función callback pasada como parámetro.

b) Llama a modificarArray y muestra en consola el tamaño del array dentro del callback.

c) Utiliza forEach para imprimir cada nombre del array después de modificarlo

8. Ejercicio 3: Funciones de orden superior (HOF)

a) Crea dos funciones: exito() que muestre "Tarea completada" y fallo() que muestre "Error en la tarea".

b) Crea una función de orden superior ejecutarTarea(callbackExito, callbackError) que tenga un 50% de probabilidades de ejecutar callbackExito y 50% de callbackError.

c) Llama varias veces a ejecutarTarea para observar diferentes resultados.

9. Ejercicio 4: Funciones generadoras

a) Crea una función generadora colores() que genere los valores "Rojo", "Verde" y "Azul".

b) Crea un objeto a partir de la función generadora y llama a next() varias veces, mostrando el resultado en consola.

c) Añade un bucle for...of para imprimir todos los valores de la función generadora.

10. Ejercicio 5: Expresiones regulares

Dado el siguiente texto de prueba (úsalo cuando sea necesario):

```
const texto = `
Pedro nació el 1998-07-21.
Sonia nació el 2003-11-05.
La fecha incorrecta es 21-07-1998.
Mi correo es ejemplo123@gmail.com
Teléfono: 678-123-456
Palabras: perro, pato, saco, Sapo, mesa, sol, pico, pera
`;
```

Apartado 1: Coincidencias básicas

a) Crea una expresión regular que compruebe si una palabra contiene exactamente 4 caracteres.
b) Modifica la expresión anterior para que solo acepte letras, no números ni símbolos.

c) Explica qué significa el carácter . dentro de una expresión regular.

Apartado 2: Anclas (^ y $)

a) Crea una expresión regular que valide que un texto empieza por la letra p.

b) Crea otra que valide que un texto termina en a.

c) Crea una expresión regular que valide palabras que empiezan por p y terminan por o.

d) Prueba las expresiones con varias palabras del texto dado.

Apartado 3: Rangos y alternativas

a) Crea una expresión regular que detecte palabras que empiecen por p o s.

b) Modifica la expresión para que además terminen por o o a.

c) Añade el flag necesario para que no distinga mayúsculas de minúsculas.

d) Explica el uso del operador |.

Apartado 4: Cuantificadores

a) Crea una expresión regular que detecte números de exactamente 2 dígitos.

b) Modifícala para que acepte números de 2 a 4 dígitos.

c) Crea una expresión regular que detecte la palabra "disparo" o "disparos" usando ?.

d) Explica la diferencia entre *, + y ?.

Apartado 5: Expresiones negadas

a) Crea una expresión regular que detecte cualquier carácter que NO sea un número.

b) Crea otra que detecte palabras que NO empiecen por vocal.

c) Explica el significado del símbolo ^ cuando se usa dentro de corchetes [].

Apartado 6: Fechas

a) Crea una expresión regular que detecte fechas en formato YYYY-MM-DD.

b) Usa el método test() para comprobar si la fecha "2025-12-15" es válida.

c) Usa el método exec() para extraer la fecha del texto dado.

d) Indica qué información devuelve exec() además del texto coincidente.

Apartado 7: Grupos de captura

a) Modifica la expresión de fecha para capturar por separado:

- **Año**
- **Mes**
- **Día**

b) Muestra en consola cada uno de los valores capturados.

c) Explica qué es un grupo de captura y para qué sirve.

Apartado 8: Flags

a) Crea una expresión regular con los flags g e i.

b) Explica qué hace cada uno de esos flags.

c) Usa la propiedad .flags para mostrar los flags activos.
d) Usa la propiedad .source para mostrar el patrón sin los delimitadores.

Apartado 9: lastIndex y búsquedas sucesivas

a) Crea una expresión regular con el flag g para buscar fechas en el texto.

b) Ejecuta exec() varias veces y observa cómo cambia lastIndex.

c) Explica para qué sirve la propiedad .lastIndex.

Apartado 10: Validaciones reales

a) Crea una expresión regular que valide un correo electrónico sencillo (usuario@dominio.com).

b) Crea una expresión regular que valide un número de teléfono con el formato XXX-XXX-XXX.

c) Explica las limitaciones de estas expresiones regulares.

Apartado 11: Comparación de métodos

a) Indica cuándo es mejor usar test() y cuándo exec().

b) Crea un ejemplo práctico donde test() sea suficiente.

c) Crea otro ejemplo donde sea necesario usar exec().

Apartado 12: Expresión final

Crea una única expresión regular que cumpla lo siguiente:

- **Empiece por letra**
- **Contenga solo letras y números**
- **Tenga entre 5 y 10 caracteres**
- **No distinga mayúsculas de minúsculas**

Explica paso a paso cómo funciona la expresión creada.