

# UDO2UT03

## Fundamentos Funciones

### Índice

Resultado de aprendizaje, contenido y criterios de evaluación.....	2
Instrucciones.....	4
Actividad “Economía doméstica con map y reduce” .....	4
Rúbrica.....	5

## Resultado de aprendizaje, contenido y criterios de evaluación

Resultado de aprendizaje
RA2. Selecciona las arquitecturas y tecnologías de programación sobre clientes web, identificando y analizando las capacidades y características de cada una.
Contenido
1 – Variables. 2 - Tipos de datos. Conversiones. 3 – Asignaciones. 4 – Operadores 5 - Comentarios al código. 6 – Sentencias. 7 – Decisiones. 8 – Bucles. 9 - Prueba y documentación del código.
Criterios de evaluación
b) Se han utilizado los distintos tipos de variables y operadores disponibles en el lenguaje. c) Se han identificado los ámbitos de utilización de las variables d) Se han reconocido y comprobado las peculiaridades del lenguaje respecto a las conversiones entre distintos tipos de datos. e) Se han utilizado mecanismos de decisión en la creación de bloques de sentencias. f) Se han utilizado bucles y se ha verificado su funcionamiento g) Se han añadido comentarios al código h) Se han utilizado herramientas y entornos para facilitar la programación, prueba y documentación del código

<b>Resultado de aprendizaje</b>
RA3. Escribe código, identificando y aplicando las funcionalidades aportadas por los objetos predefinidos del lenguaje.
<b>Contenido</b>
1 - Utilización de objetos. Objetos nativos del lenguaje. 2 - Interacción con el navegador. Objetos predefinidos asociados 3 - Generación de texto y elementos HTML desde código 6 - Interacción con el usuario. 8 - Depuración y documentación del código.
<b>Criterios de evaluación</b>
a) Se han identificado los objetos predefinidos del lenguaje e) Se han escrito sentencias que utilicen los objetos predefinidos del lenguaje para interactuar con el usuario h) Se ha depurado y documentado el código

<b>Resultado de aprendizaje</b>
R4. Programa código para clientes web analizando y utilizando estructuras definidas por el usuario
<b>Contenido</b>
1 - Funciones predefinidas del lenguaje 2 - Llamadas a funciones. Definición de funciones 3 - Matrices (arrays). 4 - Operaciones agregadas: filtrado, reducción y recolección
<b>Criterios de evaluación</b>
a) Se han clasificado y utilizado las funciones predefinidas del lenguaje b) Se han creado y utilizado funciones definidas por el usuario. c) Se han reconocido las características del lenguaje relativas a la creación y uso de matrices (arrays). d) Se han creado y utilizado matrices (arrays). e) Se han utilizado operaciones agregadas para el manejo de información almacenada en colecciones f) Se han reconocido las características de orientación a objetos del lenguaje g) Se ha creado código para definir la estructura de objetos. h) Se han creado métodos y propiedades i) Se ha creado código que haga uso de objetos definidos por el usuario j) Se han utilizado patrones de diseño de software k) Se ha depurado y documentado el código

## Instrucciones

- Desarrolla sobre este mismo documento lo que se pide en cada apartado.
- Genera el código y sube a la tarea el fichero .js.
- Nombra el fichero como TuNombre\_Apellido1\_Apellido2\_ UD02TE03.pdf
- Súbelo a la tarea de CAMPUS.

## Actividad “Economía doméstica con map y reduce”

A partir del documento **UD01DOC01 - Arquitecturas y Lenguajes de Programación en Clientes Web** realiza las siguiente actividad:

Gestionaremos una economía doméstica sencilla. Para ello:

- Crea uno array de ingreso y gastos con valores iniciales, con el formato del apunte: {fecha, concepto, importe}. Usa fecha (yyyy-mm-dd HH:mm:ss) como id.
- Crear un menú para gestionar:
  - Agregar nuevo ingreso o gasto
  - Editar un apunte (busca por fecha)
  - Borrar un apunte (busca por fecha)
  - Mostrar saldo
  - Mostrar total por categoría (ingreso o gasto)
  - Muestra resumen mensual dada una fecha yyyy-mm
- Calcular totales y métricas usando map y reduce.
- Ejecutable con Node. Sin librerías.

Crea el fichero economia.js.

### 1.- Realizar los cálculo (funciones). Uso de map y reduce:

- saldo() = sum(ingresos) – sum(gastos) con reduce.
- totalPorCategoría(tipo) → objeto {categoria: total} con reduce. tipo ingreso|gasto
- resumenMensual(yyyy\_mm) filtra por mes y devuelve {ingresos, gastos, saldo} usando reduce.
- lineasFormateadas() devuelve array de strings con map tipo "2025-10-05 | compra | gasto | super | -23.40€".

### 2.- Prueba a realizar:

- Crea al menos 5 apuntes variados.
- Editar apunte y modificarlo.
- Borrar apunte.
- Muestra saldo().
- Muestra totalPorCategoría("gasto").
- Muestra resumenMensual("2025-10").

# Rúbrica

Criterio	Excelente (10)	Bien (7)	Suficiente (5)	Insuficiente (0)
<b>RA2-b) Tipos de variables y operadores.</b>	Usa correctamente tipos, operadores aritméticos, de comparación y lógicos. Sin coerciones indeseadas. <b>(0,8)</b>	Algún uso subóptimo sin afectar resultados. <b>(0,56)</b>	Varios descuidos pero cálculos válidos. <b>(0,40)</b>	Errores que invalidan cálculos. <b>(0)</b>
<b>RA2-c) Ámbitos de utilización de las variables.</b>	Controla alcance y sombra. Evita globales y fugas. <b>(0,3)</b>	Algún global justificado. <b>(0,21)</b>	Mezcla alcances sin impacto grave. <b>(0,15)</b>	Uso incorrecto rompe ejecución. <b>(0)</b>
<b>RA2-d) Conversiones entre tipos.</b>	Normaliza importes y fechas con conversión explícita y segura. <b>(0,8)</b>	Conversión correcta con casos límite sin cubrir. <b>(0,56)</b>	Conversión parcial o frágil. <b>(0,40)</b>	Coerciones erróneas o NaN. <b>(0)</b>
<b>RA2-e) Decisiones en bloques de sentencias.</b>	Menú y validaciones robustas. Ramas bien cubiertas. <b>(0,8)</b>	Flujo correcto con validaciones parciales. <b>(0,56)</b>	Flujo básico. <b>(0,40)</b>	Decisiones incorrectas o ausentes. <b>(0)</b>
<b>RA2-f) Bucles.</b>	Iteraciones adecuadas y seguras cuando proceda. <b>(0,3)</b>	Menores ineficiencias. <b>(0,21)</b>	Ineficiencias claras. <b>(0,15)</b>	Itera mal o innecesario. <b>(0)</b>
<b>RA2-g) Comentarios al código.</b>	Comentarios breves y útiles por función y partes clave. <b>(0,3)</b>	Comentarios suficientes. <b>(0,21)</b>	Pocos o genéricos. <b>(0,15)</b>	Ausentes o confusos. <b>(0)</b>
<b>RA2-h) Herramientas y entornos.</b>	Scripts de ejecución/prueba. <b>(0,5)</b>	Uno de los dos en buen estado. <b>(0,35)</b>	Mención mínima. <b>(0,25)</b>	Nada aportado. <b>(0)</b>
<b>RA3-a) Objetos predefinidos del lenguaje.</b>	Usa Date, Intl.NumberFormat, Math según necesidad. <b>(0,4)</b>	Uso correcto sin justificar elección. <b>(0,28)</b>	Uso parcial o redundante. <b>(0,20)</b>	Mal uso o ausencia. <b>(0)</b>
<b>RA3-e) Interacción con el usuario.</b>	CLI con readline u otra entrada. Manejo de errores. <b>(0,4)</b>	Interacción correcta con fallos menores. <b>(0,28)</b>	Interacción básica sin control de errores. <b>(0,20)</b>	Interacción deficiente. <b>(0)</b>
<b>RA3-h) Depuración y documentación.</b>	Mensajes de error claros. <b>(0,4)</b>	Depuración básica o documentación parcial. <b>(0,28)</b>	Mínimos apuntes. <b>(0,20)</b>	Sin depuración. <b>(0)</b>
<b>R4-a) Funciones predefinidas del lenguaje.</b>	Emplea adecuadamente Array.*, Object.*, etc. <b>(0,3)</b>	Algún uso mejorable. <b>(0,21)</b>	Uso limitado. <b>(0,15)</b>	No aplica o aplica mal. <b>(0)</b>
<b>R4-b) Funciones definidas por el usuario.</b> (saldo, totalPorCategoria, resumenMensual, lineasFormateadas)	Las 4 funciones. Firmas claras. Probadas. <b>(1,2)</b>	3 funciones correctas. <b>(0,84)</b>	2 funciones operativas. <b>(0,60)</b>	≤1 función o fallan. <b>(0)</b>
<b>R4-c) Características de arrays.</b>	Modelo {fecha, concepto, tipo, categoria, importe} correcto. <b>(0,3)</b>	Modelo correcto con detalle menor. <b>(0,21)</b>	Modelo parcial. <b>(0,15)</b>	Modelo incorrecto. <b>(0)</b>
<b>R4-d) Creación y uso de arrays. CRUD por fecha.</b>	CRUD fiable por fecha como id. <b>(0,8)</b>	CRUD con fallos menores. <b>(0,56)</b>	CRUD incompleto. <b>(0,40)</b>	CRUD incorrecto o ausente. <b>(0)</b>
<b>R4-e) Operaciones agregadas. map y reduce.</b>	Totales y métricas implementadas con map/reduce de forma correcta y eficiente. <b>(2,0)</b>	Algún cálculo fuera de map/reduce. <b>(1,40)</b>	Mezcla bucles sin necesidad. <b>(1,00)</b>	No usa map/reduce. <b>(0)</b>

<b>Criterio</b>	<b>Excelente (10)</b>	<b>Bien (7)</b>	<b>Suficiente (5)</b>	<b>Insuficiente (0)</b>
<b>R4-k) Depuración y documentación del código.</b>	Tests o casos de prueba y notas de límites. <b>(0,4)</b>	Alguna evidencia de prueba. <b>(0,28)</b>	Prueba mínima. <b>(0,20)</b>	Sin pruebas. <b>(0)</b>

	Se retrasa más de 5 días	Se retrasa entre 4 y 5 días	Se retrasa entre 2 y 3 días	Se retrasa 1 día.	Entrega en plazo
Entrega tarea en plazo	- 5	- 3	-2	-1	0