

[DWEK] Tarea presencial | 3º trimestre

Instrucciones de entrega

- Se deberá entregar la tarea ANTES de la hora límite.
- Los archivos de código (.HTML o .JS) deben contener una cabecera con la información de autoría (nombre del autor).
- Se deberá realizar la entrega de un archivo .ZIP con solución a la tarea online.
- Se recomienda adjuntar un archivo PDF mostrando el correcto funcionamiento de cada actividad.

Crea una página web que permita la conversión entre diferentes tipos de monedas usando la API REST **vatcomply** (<https://www.vatcomply.com/documentation>).

Parte 1 | Consultas AJAX

Primero vamos a implementar la funcionalidad de realizar consultas a la API y realizar conversiones de divisas. Se deben tener en cuenta los siguientes puntos.

- Un desplegable para elegir la divisa de origen de la conversión y otro para la de destino. Las opciones del desplegable deben rellenarse mediante una petición a la API.
- En los desplegables debe mostrarse el nombre de la moneda y no el código (p. ej. Euro en lugar de EUR).
- Los inputs numéricos deben actualizar el símbolo de la moneda para que encaje con la seleccionada en el desplegable.
- Por defecto la moneda de origen debe ser EUR y la de destino USD.
- Un campo de texto para la introducción de caracteres numéricos. Si se introduce otro tipo de dato debe de dar error. Por defecto su valor debe ser de 10.
- Un campo donde se muestre el resultado de la conversión de divisa.
- La conversión debe realizarse automáticamente al cambiar el valor de los desplegables o el campo de texto.
- Si falla la conversión, por errores relativos a la comunicación con la API, se debe generar una alerta por pantalla avisando al usuario que el servicio está actualmente inactivo.

A1 - Conversiones

Moneda origen	Cantidad
<input type="text" value="Euro"/>	<input type="text" value="€ 10"/>
Moneda destino	Resultado
<input type="text" value="British Pound"/>	<input type="text" value="£ 8.56"/>

Resumen de los criterios de evaluación de la actividad 1 (RA7)

RA7.c) Se han utilizado los objetos relacionados.

RA7.d) Se han identificado sus propiedades y sus métodos.

RA7.e) Se ha utilizado comunicación asíncrona en la actualización dinámica del documento Web.

RA7.f) Se han utilizado distintos formatos en el envío y recepción de información.

Parte 2 | Historial de consultas

Segundo vamos a crear una tabla de historial de consultas. Esta debe tener los siguientes campos:

Fecha	Conversión	Valor inicial	Valor final
10/05/2022	Euro a US Dollar	10	10.55

Al lado del campo donde se muestra el resultado de la conversión debe haber un botón de “Añadir conversión al historial” (puede ponerse con un icono de añadir). Además, este deberá:

- El botón debe desactivarse mientras se hace una petición al servidor.
- Los nuevos registros deben añadirse en la parte superior de la tabla.

Cada fila de la tabla debe contar con un botón de borrar entrada del historial (puede ponerse con un icono de borrar).

Por último, cada fila también debe contar con botones para subir o bajar la entrada.

A1 - Conversiones

Moneda origen	Cantidad
Euro	€ 10
Moneda destino	Resultado
British Pound	£ 8.56

A2- Historial

Fecha	Conv.	Cantidad (org.)	Cantidad (des.)	Acciones
15:29:56 11/4/2022	EUR a GBP	10 €	8.56 £	▲ ▼ ✕
15:29:52 11/4/2022	EUR a JPY	10 €	1373.80 ¥	▲ ▼ ✕
15:29:48 11/4/2022	EUR a USD	10 €	10.55 \$	▲ ▼ ✕

Resumen de los criterios de evaluación de la actividad 2 (RA6)

RA6.c) Se ha creado y verificado un código que acceda a la estructura del documento.

RA6.d) Se han creado nuevos elementos de la estructura y modificado elementos ya existentes.

RA6.e) Se han asociado acciones a los eventos del modelo.

RA6.h) Se han independizado las tres facetas (contenido, aspecto y comportamiento), en aplicaciones Web.

Parte 3 | Compatibilidad entre navegadores

Se ha depurado el código, se garantiza que se evitan posibles errores (como introducir valores no validos en los inputs) y se comprueba el funcionamiento de tu código en al menos 3 navegadores diferentes.

Resumen de los criterios de evaluación de la actividad 2 (RA6 & RA7)

RA6.g) Se han programado aplicaciones Web de forma que funcionen en navegadores con diferentes implementaciones del modelo.

RA7.g) Se han programado aplicaciones Web asíncronas de forma que funcionen en diferentes navegadores.

RA7.i) Se han creado y depurado programas que utilicen estas librerías