

# U08-T01: Trajes típicos de Canarias

## Objetivo

El objetivo es crear un juego interactivo con la ayuda de jQueryUI y Toastr. El usuario ha de arrastrar cada una de las [imágenes de trajes típicos de Canarias a la isla asociada](#) . La estructura o reglas para el juego son las siguientes:

## Descripción

- El juego debe estar en una página Web que contenga un área de juego que no ocupe todo el alto y ancho de la Web
- El usuario debe elegir el nivel de dificultad tres botones: fácil, normal y experto (con jQueryUI)
- Inicialmente:
  - Debe haber entre 4 y 7 áreas (containers para las diferentes islas (containers) en el área de juego de manera aleatoria
  - El juego debe mostrar (a la vez y en un área reservada para ello) al usuario un número de imágenes de trajes típicos (draggables) al azar (del total de imágenes proporcionadas) en el área de juego:
    - Fácil: 4 imágenes
    - Medio: 7 imágenes
    - Experto: 8 imágenes
- Las imágenes de los trajes a ubicar deben corresponderse únicamente a las islas que hay en el nivel seleccionado.
- El usuario debe ser capaz de seleccionar una imagen de un traje y arrastrarla con el ratón sobre la imagen de la isla adecuada. No se pueden arrastrar elementos fuera del área de juego.
- Si el usuario deposita la imagen del traje en la isla adecuada, él/ella gana 1 punto y se avisará al usuario a través de notificaciones Toastr. En el caso de error, la puntuación será de 0.
- Si la imagen del traje se deposita de manera correcta se le aplicará algún estilo (por ejemplo: borde verde). En el caso de que lo haga de manera incorrecta, le aplicará otro estilo diferente (por ejemplo: borde rojo).
- Después de que todas las imágenes de trajes han sido arrastradas, se debe mostrar el resultado (por ejemplo: "Tienes 2 puntos) y pedir al usuario "¿Quieres jugar de nuevo? Si el usuario decide volver a jugar, se inicia el juego (y puede cambiar de nivel), de lo contrario el juego se termina. Para estos mensajes se han de usar ventanas modales con algún efecto a la entrada y salida (jQueryUI).

Autor de las imágenes: [Proyecto TSP \(IECI-EFC\)](#)

## Calificación

- Requisitos mínimos (70%):
  - Las funcionalidades desde el lado cliente están implementadas con jQuery en vez de con JS (Vanilla): máx. 0,5 pts
  - El usuario puede elegir el nivel de dificultad con tres botones: fácil, normal y experto: 0.5 pts
  - Al comenzar la partida:
    - Hay entre 4 y 7 áreas (containers para las diferentes islas (containers) en el área de juego de manera aleatoria: 0.5 pts
    - Se muestran (a la vez y en un área reservada para ello) al usuario un número de imágenes de trajes típicos al azar en el área de juego:

- Fácil: 4 imágenes
  - Medio: 7 imágenes
  - Experto: 8 imágenes
  - (0,5 ptos)
- Las imágenes de los trajes a ubicar deben corresponderse únicamente a las islas que hay en el nivel seleccionado (1 pto)
- El usuario debe ser capaz de seleccionar una imagen de un traje y arrastrarla con el ratón sobre la imagen de la isla adecuada. (1 pto)
- No se pueden arrastrar elementos fuera del área de juego. (0,5 ptos)
- Mientras se está arrastrando un traje, se controlan aspectos como: tipo de cursor, estilo,.....: (0.5 ptos)
- Si el usuario deposita la imagen del traje en la isla adecuada, él/ella gana un punto y se avisará al usuario a través de notificaciones Toastr. (0,3 ptos)
- Si la imagen del traje se deposita de manera correcta se le aplicará algún estilo (por ejemplo: borde verde). En el caso de que lo haga de manera incorrecta, le aplicará otro estilo diferente (por ejemplo: borde rojo). (0,5 ptos)
- Después de que todas las imágenes de trajes han sido arrastradas:
  - se muestra el resultado de manera correcta (0,5 ptos)
  - cuando se solicita al usuario si quiere jugar de nuevo, si decide volver a jugar, se inicia el juego y puede cambiar de nivel
  - para estos mensajes se usan ventanas modales con algún efecto a la entrada y salida (jQueryUI).(0,2 ptos)
- Buenas prácticas de programación (20%):
  - Sólo se valorará si se obtienen al menos 5 puntos de 7 en el apartado anterior.
  - Código mínimo imprescindible.
  - Código optimizado.
  - Comentarios necesarios y suficientes.
  - Variables y métodos con nombres significativos y autoexplicativos.
    - uso de ESLint (incluir al menos 3 capturas en las que se pueda distinguir que se usó): 0.5 ptos.
    - notación camelCase, funciones arrows, métodos map, filter, reduce,...: 0.5 ptos.
    - uso de JSDocs: 0.5 ptos.
    - código refactorizado: 0.5 ptos.
- Originalidad y creatividad (10%). Se valorarán positivamente las aportaciones creativas o innovadoras para enriquecer la aplicación:
  - Sólo se valorará si se obtienen al menos 7 puntos de 9 entre los dos apartados anteriores.
  - Se valorarán como máximo dos aportaciones.