

**UNIVERSIDAD POLITÉCNICA DE SINALOA**  
**Ingeniería en Tecnologías de la Información e Innovación Digital**  
**Asignatura: Programación Estructurada      Evaluación: Corte I**

**Nombre del Alumno:** \_\_\_\_\_

**Instrucciones generales:**

Se requiere desarrollar una aplicación web que consuma información desde dos APIs públicas relacionadas entre sí, integre los datos obtenidos y los represente dinámicamente en el sitio utilizando JavaScript y manipulación del DOM. La aplicación deberá demostrar que el estudiante es capaz de interpretar correctamente estructuras JSON, seleccionar únicamente la información relevante, almacenarla en variables para su reutilización y representarla en distintos formatos visuales sin realizar múltiples peticiones innecesarias.

- **API 1 (Usuario):** <https://randomuser.me/api/>
- **API 2 (País):** <https://restcountries.com>

La información del país deberá obtenerse utilizando el país proporcionado por la API del usuario, estableciendo así una relación lógica entre ambas fuentes.

**LA APLICACIÓN DEBERÁ CUMPLIR CON LOS SIGUIENTES PUNTOS:**

**1.- Botón de carga de información**

La página deberá incluir un botón que:

- Realice únicamente una vez las peticiones necesarias a ambas APIs.
- Obtenga de la API de usuario únicamente los siguientes datos:
  - Nombre completo
  - Edad
  - País
  - Imagen del usuario
- Utilice el país obtenido para consultar la API de países y extraer únicamente:
  - Bandera
  - Capital
  - Región
- Cambie temporalmente su texto a “Cargando...” mientras se obtienen los datos y luego vuelva a su estado original.

**No se deberán realizar nuevas peticiones una vez que los datos hayan sido obtenidos.**

## **2.- Visualización inicial estructurada**

Después de cargar la información, se deberá mostrar:

- Una sección estructurada (tabla o bloque organizado) que contenga exclusivamente los datos solicitados.
- No se permitirá mostrar todos los datos devueltos por la API; únicamente los requeridos.
- Todos los elementos deberán generarse dinámicamente desde el DOM utilizando createElement.
- El contenido previo deberá limpiarse antes de mostrar nueva información.

## **3.- Generación de tarjeta integrada**

La página deberá incluir un segundo botón que:

- Genere una tarjeta visual integrada utilizando únicamente la información previamente almacenada.
- No realice nuevas peticiones a las APIs.
- Integre datos provenientes de ambas fuentes en una sola estructura coherente.
- Genere todos los elementos desde cero mediante el DOM.

La tarjeta deberá incluir:

- Imagen del usuario.
- Nombre completo.
- Edad.
- País.
- Bandera del país.
- Capital.
- Región.

## **Congruencia visual obligatoria**

El estilo visual de la tarjeta deberá ser coherente con el estilo utilizado en la sección inicial de datos. Se evaluará:

- Uso consistente de colores.
- Espaciado adecuado.
- Jerarquía visual clara.
- Uso de clases CSS correctamente aplicadas.
- Cambio dinámico de al menos un estilo desde JavaScript (por ejemplo, color del nombre o fondo de la tarjeta).

## Manejo obligatorio de errores

Se deberán contemplar al menos dos situaciones:

- Error en la obtención de datos desde alguna API.
- Intentar generar la tarjeta sin haber cargado previamente la información.

En ambos casos deberá mostrarse un mensaje claro al usuario sin que la aplicación se detenga.

## Restricciones técnicas

- No se permite mostrar todos los datos devueltos por las APIs.
- No se permite escribir datos directamente en el HTML.
- No se permite utilizar librerías externas.

Ejemplo de la tarjeta:



## RÚBRICA GENERAL DE EVALUACIÓN

**Valor total: 100 puntos**

El examen será evaluado con base en los siguientes criterios generales:

**1. Obtención correcta de información desde ambas APIs (30 puntos).**

Se deberán realizar correctamente las peticiones a las dos APIs indicadas, convertir las respuestas a JSON y almacenar la información en variables para su reutilización.

**2. Generación correcta de la tarjeta integrada (40 puntos).**

La tarjeta deberá integrar coherentemente información proveniente de ambas APIs, generarse dinámicamente y reutilizar los datos previamente almacenados sin realizar nuevas peticiones.

Si la tarjeta no funciona correctamente, no integra ambas fuentes o presenta campos vacíos, el examen no será aprobatorio, independientemente del puntaje acumulado.

**3. Manejo básico de errores (10 puntos).**

Se deberán contemplar al menos dos situaciones de error:

- Fallo en la obtención de datos desde la API.
- Intento de generar la tarjeta sin haber cargado previamente la información.  
En ambos casos se deberá mostrar un mensaje claro sin que la aplicación falle.

**4. Congruencia visual y aplicación de estilos (20 puntos).**

Se evaluará coherencia entre la sección inicial y la tarjeta, uso adecuado de clases CSS y al menos un cambio dinámico de estilo aplicado desde JavaScript.