

TP Vacaciones de invierno

El siguiente trabajo práctico para realizar durante las vacaciones de invierno consiste en desarrollar una página web o pequeña aplicación web y hostearla desde GitHub Pages.

La aplicación o página tiene que estar contenida en tres archivos: un `index.html`, un `style.css` y un archivo de JavaScript. Si hiciera falta se pueden usar imágenes o archivos de audio, pero los archivos de código tienen que ser tres, uno por cada lenguaje.

No se pueden usar frameworks o librerías, la aplicación debe estar hecha en HTML, CSS y JavaScript puros. Sí se puede usar la API de JavaScript para trabajar con el DOM (por ejemplo `document.getElementById()`) o cualquier función que soporten los navegadores modernos. Con frameworks o librerías me refiero a jQuery, lodash, Bootstrap, etc. Que no se trabaje con frameworks CSS no quiere decir que tengan que escribir código de *layout* para cubrir todos los tamaños de pantalla posibles. Se espera que la aplicación se vea bien en un monitor promedio, no que sea *responsive*.

Si la aplicación almacena información no necesita conectarse a una base de datos o algún tipo de almacenamiento persistente. O está *hardcodeada* o bien se pierde al actualizar el navegador (F5). Eso es esperable con las herramientas que tienen.

El trabajo es individual y los temas se asignan a cada alumno en la tabla consignada abajo.

Todas las aplicaciones tienen que tener un pie de página o *footer* donde se lea el nombre y apellido del alumno, la división y un link o botón al repositorio donde está hosteado el código.

Pueden usar como punto de partida y ejemplo el repositorio del ejercicio del parque de diversiones: <https://github.com/santiagotrini/parque-diversiones>.

Entrega

La fecha de entrega límite es el miércoles 7 de agosto hasta las 18:00 hs. Debe enviarse un mail a santiago.trini@bue.edu.ar indicando en el asunto nombre, apellido y división. En el cuerpo del mail debe estar el link a la web (la URL).

El trabajo es individual y se espera que cada uno le aporte su estilo a su aplicación. Trabajos idénticos serán calificados con idéntica baja nota. El trabajo tiene nota numérica y se promedia con las pruebas para el segundo trimestre. También voy a revisar que los trabajos no estén plagiados de otro lado.

El repositorio de GitHub debe ser público. No se aceptarán trabajos enviados luego de la fecha límite.

Cualquier duda o consulta escriban al mail de arriba.

Temas

1. Conversor de monedas.

2. Conversor de temperaturas.
3. Conversor de sistemas de numeración.
4. Lista de tareas.
5. ¿Cuánto falta para la primavera?
6. Trivia.
7. Abecedario de jardín.
8. Traductor de código Morse.
9. Generador de memes.
10. Test.
11. Traductor de Jeringoso.
12. Generador de citas.

Apellido	Curso	Tema
Abella	4º1º	1
Amigo	4º1º	2
Barrague	4º1º	3
Blanco	4º1º	4
Bordon	4º1º	5
Caputo	4º1º	6
Delfino	4º1º	7
Dumont	4º1º	8
Espindola	4º1º	9
Fernandez	4º1º	10
Forneron	4º1º	12
Gantus	4º1º	3
Garavagno	4º1º	9
Garcia	4º1º	12
Granata	4º1º	7
Hernandez	4º1º	6
Lusnich	4º1º	4
Magre	4º1º	1
Martinez Carol	4º1º	2
Miliani	4º1º	10
Pastor	4º1º	6
Perez	4º1º	2

Apellido	Curso	Tema
Perree	4°1°	4
Rodriguez	4°1°	3
Safe	4°1°	12
Sampayo	4°1°	8
Sanchez Slater	4°1°	7
Sanchez Tonarelli	4°1°	9
Sauchella	4°1°	7
Serpa	4°1°	1
Thenon	4°1°	3
Trosero	4°1°	12
Vaca	4°1°	2
Vaccaro	4°1°	2
Abadi	4°2°	6
Alarcon	4°2°	3
Benitez	4°2°	7
Casimiro	4°2°	8
Cingolani	4°2°	9
Constenla	4°2°	1
D'Angelo	4°2°	7
De La Fuente	4°2°	5
Díaz	4°2°	4
Espinosa	4°2°	10
Estrada	4°2°	2
García	4°2°	6
Gerula	4°2°	1
Humano	4°2°	5
Iglesias	4°2°	9
Jimenez	4°2°	4
Kilberg	4°2°	3
Lesta	4°2°	2
Licciardi	4°2°	1
Marconi	4°2°	8
Mendoza	4°2°	4

Apellido	Curso	Tema
Panzitta	4º2º	6
Petean	4º2º	10
Quisbert	4º2º	9
Romero	4º2º	3
Sosa	4º2º	4
Spano	4º2º	10
Staltari	4º2º	7
Toscano	4º2º	8
Valle	4º2º	5
Vázquez	4º2º	7
Velasco	4º2º	5
Williams	4º2º	11
Aguirre	4º3º	1
Alvarez	4º3º	2
Arenas	4º3º	11
Barredo	4º3º	3
Castillo	4º3º	4
Cely Amaya	4º3º	5
Claudio Canaza	4º3º	6
Comesaña	4º3º	8
Díaz	4º3º	7
Doglio	4º3º	2
Faura	4º3º	9
Gayarre	4º3º	10
Grillo	4º3º	12
Inchausti	4º3º	1
Leon	4º3º	3
Mamani Laura	4º3º	2
Mamani Quispe	4º3º	9
Martinez Muller	4º3º	7
Meier	4º3º	6
Montiel	4º3º	4
Moyano	4º3º	10

Apellido	Curso	Tema
Ortiz	4º3º	8
Piergallini	4º3º	3
Pozzi	4º3º	8
Rocha	4º3º	9
Salas	4º3º	10
Tapia	4º3º	6
Tudela	4º3º	4
Valenti	4º3º	1
Yana	4º3º	12

Convertor de monedas

Un convertor de monedas es una serie de entradas de texto y botones para pasar por ejemplo mil dólares a su equivalente en pesos o euros. El convertor tiene que manejar al menos cuatro monedas diferentes. Las cotizaciones estarán *hardcodeadas* en el script de JavaScript. Si se animan pueden hacer una consulta a una API externa para traer la cotización al momento de la consulta. Para realizar eso sin usar librerías tendrían que usar `fetch()` en JavaScript, pero no es obligatorio.

Convertor de temperaturas

Similar al convertor de monedas en la interfaz. Se espera que trabajen con Kelvin, Celsius o Fahrenheit.

Convertor de sistemas de numeración

Similar al convertor de temperaturas, se espera que conviertan entre decimal, binario, octal y hexadecimal.

Lista de tareas

Este es un clásico del desarrollo web, hay cientos de ejemplos con distintos frameworks, arquitecturas y bases de datos para armar el famoso “TODO List”. Como todavía no manejamos bases de datos alcanza con armar una interfaz y presentar la lista de tareas tomada de un *array* o algo similar. Claro que al actualizar la página se pierden todas las tareas, pero es esperable. Las tareas tienen que poder tacharse y deben desaparecer de la lista.

¿Cuánto falta para la primavera?

La idea de este tema es implementar una página web que me reciba con una placa roja de Crónica que me diga “faltan tantos días para la primavera”. Cada vez que entro en la web un script es el encargado de comparar la fecha de hoy con la del 21 de septiembre de este año o del próximo según corresponda para mostrar el número correcto. El estilo de la web tiene que ser obligatoriamente el de una placa de Crónica.

Trivia

Hay que implementar una trivia con preguntas de opción múltiple, la opción más lógica es usar *radio buttons*. Al elegir una respuesta la aplicación me debe mostrar si acerté o si la respuesta es incorrecta. Como mínimo tiene que tener 10 preguntas que pueden limitarse a un tema específico o pueden ser de varios temas.

Abecedario de jardín

En un jardín de infantes suelen estar pegadas en la pared las letras del abecedario en imprenta mayúscula, minúscula y cursiva, con imágenes de palabras que empiecen con esa letra. Si ponemos eso en una web podemos hacer que al clickearlas se lea la letra y produzca el sonido de la cosa o animal que esté en el dibujo. Eso hay que hacer.

Traductor de código Morse

Sencillo, una entrada de texto lee una palabra ingresada por el usuario y con un botón traduce a código Morse. No se espera traducir de Morse a letras. Sólo de letras a Morse. Si se animan o están aburridos intenten traducir frases enteras y no sólo una palabra. No hace falta que salga sonido, solo mostrar los puntos y rayas.

Generador de memes

Esta aplicación tiene un banco de imágenes (entre 5 y 10 por ejemplo). Uno elige una imagen de una lista e ingresa una frase en una entrada de texto y se genera la imagen con el texto.

Test

En muchas revistas aparecen cada tanto “tests”, uno contesta 10 o 20 preguntas de opción múltiple, hace una cuenta y te dicen en qué barrio vas a vivir o si tus hijos van a ser ingenieros. El tema del test es libre, usen su imaginación. La interfaz es muy similar a la de la trivia.

Traductor de Jeringoso

Este es jodido, hay que usar expresiones regulares. A ver, es difícil pensar la expresión regular que funcione con el español, si ya tenés la expresión (copiada de algún lado) es llamar `String.match()` y `String.split()`. Lo demás es fácil.

Generador de citas

Eligen un personaje, por ejemplo Maradona, y hacen una página que te recibe con una cita al azar del Diego y le ponen un poco de onda a la página como para homenajear al pensador que eligieron. Las citas están directamente *hardcodeadas* en el script de JavaScript y eligen una al azar con `Math.random()`.