

# Módulo II

# DESARROLLO WEB II

• • NIVEL INTERMEDIO

¿Qué pasa  
si...?



Orientaciones para los/as mentores/as

¿QUÉ PASA SI...?

Orientaciones para los/as mentores/as

Módulo II

## Autoridades

Jefe de Gobierno

**Horacio Rodríguez Larreta**

Ministra de Educación

**María Soledad Acuña**

Jefe de Gabinete

**Luis Bullrich**

Subsecretario de Carrera Docente  
y Formación Técnica Profesional

**Manuel Vidal**

Subsecretaria de Coordinación  
Pedagógica y Equidad Educativa

**María Lucía Feced Abal**

Subsecretaria Agencia de aprendizaje  
a lo largo de la vida

**Eugenia Cortona**

Subsecretario de Gestión Económico  
Financiera y Administración de Recursos

**Sebastián Tomaghelli**

Subsecretario de Tecnología  
Educativa y Sustentabilidad

**Santiago Andrés**

VERSIÓN PRELIMINAR



=  
¿Qué pasa  
si...?

# Módulo II

# DESARROLLO WEB II

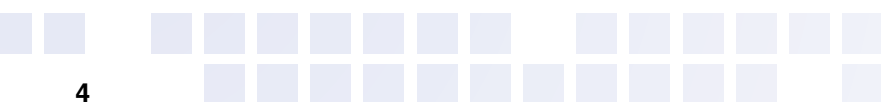
• • NIVEL INTERMEDIO



Orientaciones para los/as mentores/as



PRELIMINAR



# Índice

<b>01.</b>	Sobre el Módulo	7
	Sobre los recursos	7
	Sobre el OPEN LAB	7
	Sobre las actividades dentro de la plataforma	7
<b>02.</b>	Semana 1	8
	Objetivos	8
	Ejes temáticos	8
	Encuentro Sincrónico	8
	Encuentro Asincrónico	10
	Actividad 1. ¿Cuántos días tengo?	10
	Actividad 2. ¿Es par?	10
	Actividad 3. ¿Es divisible?	10
<b>03.</b>	Semana 2	11
	Objetivos	11
	Ejes temáticos	11
	Encuentro Sincrónico	11
	Actividad 1. Camino a elección	12
	Actividad 2. Géneros	12
	Encuentro Asincrónico	12
	Actividad. Código Secreto	13
<b>04.</b>	Semana 3	13
	Objetivos	13
	Ejes temáticos	14
	Encuentro Sincrónico	14
	Actividad. Analizando Expresiones	14
	Encuentro Asincrónico	15
	Actividad 1. Semáforo	15
	Actividad 2. Restando Puntos	15
<b>05.</b>	Semana 4	16
	Objetivos	16
	Ejes temáticos	16
	Encuentro Sincrónico	16
	Descubre al animal	17
	Siempre al final	18
	Encuentro Asincrónico	18
	Actividad. ¿Qué hay en cada expresión?	18

VERSIÓN PRELIMINAR

¿QUÉ PASA SI...?

Orientaciones para los/as mentores/as

Módulo II

<b>06.</b>	<b>Semana 5</b>	<b>19</b>
	Objetivos	19
	Ejes temáticos	19
	Encuentro Sincrónico	19
	Integrador Randomizer	20
	Encuentro Asincrónico	20
	Jugando a “Piedra, papel o tijera”	21
<b>07.</b>	<b>Semana 6</b>	<b>21</b>
	Objetivos	21
	Ejes temáticos	21
	Encuentro Sincrónico	21
	Encuentro Asincrónico	22

VERSIÓN PRELIMINAR



## 01. Sobre el Módulo

La programación es una actividad que hoy en día está presente en muchísimos campos. Aprender a programar es, de cierta forma, aprender a sistematizar el pensamiento para poder ofrecer soluciones a problemas de la vida cotidiana, ya sean de índole laboral, productiva, recreativa o expresión. A su vez permite ejercitar capacidades que son aplicables a diversos aspectos de la vida.

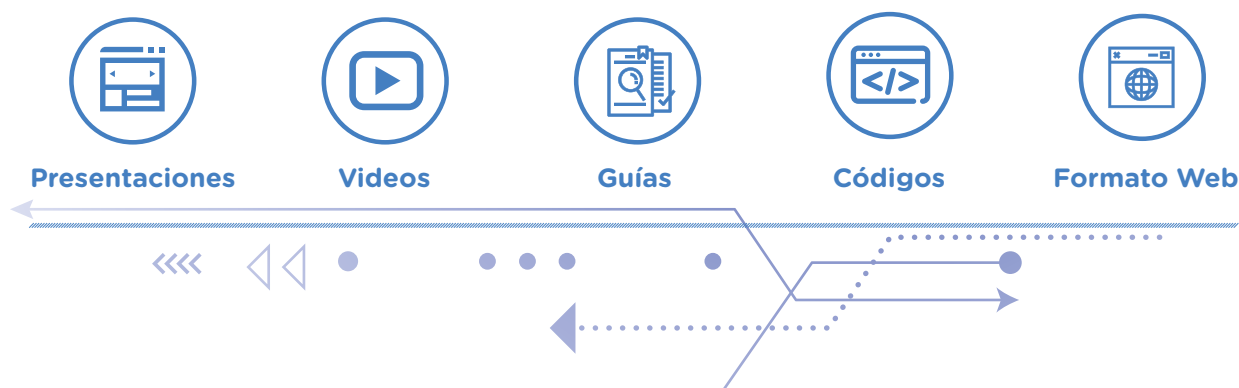
*Javascript* resulta uno de los lenguajes adecuados para adentrarse en el mundo de la programación dado que por sus cualidades de lenguaje no tipado que no requiere compilación, permite que la curva de aprendizaje sea más leve y el aprendizaje se desarrolle de forma “menos estricta” dejando mucho lugar para la creatividad y la implementación rápida y funcional que luego se irá complejizando con los saberes propios del lenguaje.

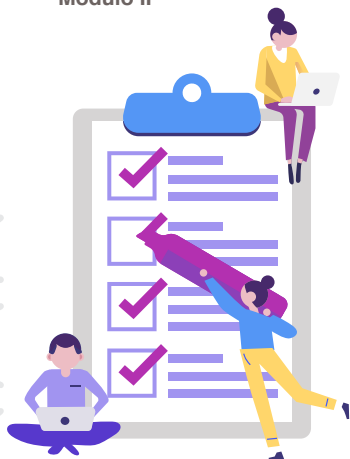
Programar requiere de mucha práctica, prueba y error. Este módulo, así como los siguientes a lo largo del trayecto, propone como entregables múltiples programas como actividades que se irán desarrollando con el correr del tiempo. A medida que las semanas pasen, los programas irán tomando mayor complejidad.

El trabajo colaborativo es algo crucial en el mundo del desarrollo. Será tarea de El/a mentor/a poder transmitir la importancia del trabajo en grupo. Si bien las actividades se plantean como entregables individuales, el desarrollo en conjunto es clave a lo largo de todo trayecto. Proponer espacios de encuentro y debate, la posibilidad de compartir código o realizar *pair programming*, utilizar herramientas/plugins de código colaborativo, etc. Es una dinámica que los/as mentores/as **deben** promover.

## Sobre los recursos

Se propone que haya una variedad de formatos, y sean contenidos articulados que toman más valor al ser explicados por el/la mentor/a. Los formatos esperados y recomendados son:





## Sobre el OPEN LAB

El *Open Lab* es un espacio que cumple principalmente dos funciones: espacio de distensión entre estudiantes así como despeje de dudas y consultas y profundización de conocimientos. Si bien es un espacio que se presenta como optativo, decir esto supone que no se desarrollarán nuevos contenidos sino que se trabajará sobre los conocimientos ya incorporados y dudas particulares. Este espacio propone ser un ámbito de construcción de sinergia entre estudiantes, tutores/as y mentores/as.

## Sobre las actividades dentro de la plataforma

Estas actividades serán a modo pregunta-respuesta (verdadero o falso, pregunta a desarrollar, *múltiple choice*, *drag&drop*) y tienen cómo objetivo (a lo largo de todo el trayecto) motivar a las/os estudiantes a que vean el material, lo repasen y logren avanzar con los contenidos del curso.



## 02. Semana 1

### Objetivos

- Aprender qué es un lenguaje de programación.
- Comprender las diferencias entre un lenguaje de maquetado, estilo y uno de programación.
- Vislumbrar el alcance de *Javascript* tanto del lado del cliente, como del lado del servidor.
- Conocer los fundamentos de este lenguaje de programación, el uso de variables, y la posibilidad de ejecución de lado de cliente con la consola de un navegador.

### Ejes temáticos

- Introducción general a los lenguajes de programación
- Vincular JS
- Tipos de datos
- Variables
- Uso de consola
- *Console.log*, *alert* y *Prompt*
- *Typeof*



## Encuentro Sincrónico

Este encuentro se divide en dos momentos con distintas actividades propuestas.

### Primer momento

En este momento se explicará de manera teórica las diferencias entre un lenguaje de programación vs uno de maquetado/estilo.

Se deberá asegurar que las y los estudiantes logren una correcta vinculación del archivo *javascript* a sus archivos *.html*.

Se conocerá *Javascript* (A partir de ahora *Js*) mediante el uso de la consola de los navegadores. Es bueno mencionar las diferencias entre un lenguaje que requiere compilación previa a diferencia de un lenguaje interpretado.

Se recomienda utilizar por medio de la consola ejercicios de operaciones básicas (suma, resta, división) y a partir de entender estas operaciones ir extendiendo los saberes (concatenar, módulo, *typeof*). Lo mejor para esta etapa inicial de aprendizaje, es poder evaluar qué es lo que pasa en la consola:

### Actividad 1

¿Qué dan las siguientes expresiones?

2 + 2	¿Qué da?
20%2, 21%2, 22%2	¿Qué conclusiones podemos sacar?
"10" + 10	Operador + cómo operador para concatenar
"10" - "10"	Operador -
"Juan" + "Perez"	Concatenar strings.

A partir de las expresiones planteadas, se hablará de la función de los operadores y cómo esta puede cambiar dependiendo el tipo de datos (uso de +, -, entre strings o strings y numbers)

Se explicarán los distintos tipos de datos. En esta primera instancia basta con explicar los datos primitivos de *Javascript*, sin necesidad de adentrarnos en *Symbols*, ni tipos de datos mutables como *arrays* y *objetos*.

Se hablará de qué es una variable y su importancia en la programación. La posibilidad de declarar variables y constantes. Se recomienda explicar el uso de variables mediante el keyword *"let"* y no *"var"*, así como el uso de *"const"* para constante. Más adelante, en encuentros futuros se hablará de las diferencias entre *"let"* y *"var"*.

La introducción del comando `console.log`, *alert* y *prompt*, servirán para la interacción con el usuario y desarrollo de programas en el segundo momento.

### Segundo momento

Se trabajará mediante enunciados y preguntas problemáticas que den lugar al desarrollo, pensamiento y creatividad a la hora de ofrecer soluciones en el desarrollo de un programa:

*¿Cómo podríamos hacer un programa que nos calcule el área de un cuadrado?  
¿Cómo armaríamos un programa que nos salude, luego de ingresar nuestro nombre?*

Es importante que las preguntas den lugar al debate, y al espacio de reflexión para comprender que la lógica de programación responde a la lógica que tenemos en nuestra mente, para luego traducirlo al código.

Con la práctica de los ejercicios en el espacio sincrónico el/la mentor/a se asegurará de resolver dudas y consultas que surgirán por parte de las/os estudiantes.

### Encuentro Asincrónico

Para este encuentro se plantean actividades entregables. El objetivo es que las/os estudiantes sigan trabajando en la creación de programas que resuelvan problemas reales.

A lo largo de todo el trayecto el espacio asincrónico se utilizará para las respuestas de dudas y consultas, así como espacios de distensión, juego y generación de vínculos. Siempre es recomendado plantear las actividades de este encuentro en el encuentro sincrónico, de esta manera las consignas ya están previamente explicadas dando lugar al desarrollo ya sea de forma individual o colectiva.

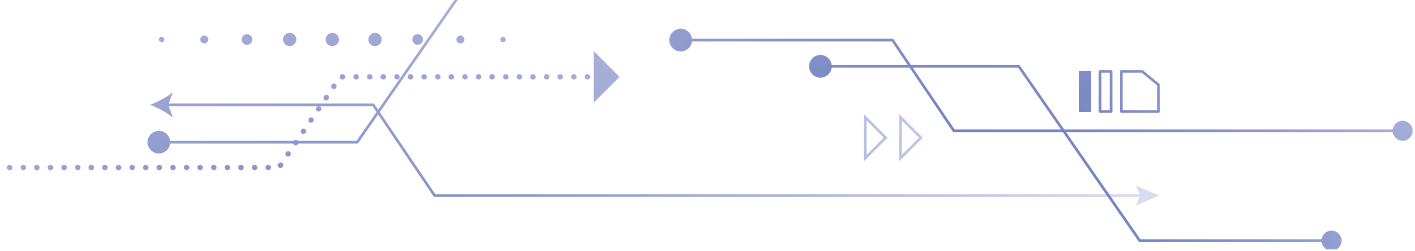
Resulta de suma importancia que los/as mentores/as propongan la posibilidad de desarrollo en conjunto. Si bien los entregables se harán de manera individual, el desarrollo colaborativo es una dirección acertada hacia la cual dirigir el trayecto. Esto aplica tanto para este encuentro como para el resto a lo largo del trayecto.

Se realizarán las actividades *quiz*, planteadas en la plataforma, con la intención de reforzar el conocimiento adquirido.

### Actividad Integradora 1

*¿Cuántos días tengo?*

En esta actividad se desarrolla un programa que calcule los días que tiene un/a estudiante. Se pedirá que ingrese su nombre, su edad, y mediante una cuenta deberá existir un *alert* que salude al usuario y muestre los días que ha vivido.



### Actividad Integradora 2 ¿Es par?

Se le pide al usuario que ingrese un número, y un alert debe informar si el número es par o no es par.

### Actividad Integradora 3 ¿Es divisible?

Se le pide al usuario que ingrese dos números, y un alert debe informar si ambos números son divisibles uno por otro.

Estas primeras actividades son actividades introductorias cuyo objetivo es que comiencen a escribir el código de forma paulatina por ellos/as mismos/as.



## 03. Semana 2

### Objetivos

- Aprender el uso de condicionales.
- Plasmar un problema lógico de la vida real en término de programación.
- Comprender la importancia y uso de distintos operadores lógicos.

### Ejes temáticos

- Uso de *if*
- Operadores de comparación
- Uso de *e/se*
- Resolución de ejercicios lógicos
- Operadores Lógicos (&& ||)
- Operadores de comparación (>, >=, <, <=, !=, !)
- *parseInt* y *Number*

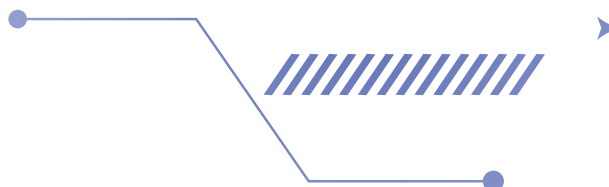
### Encuentro sincrónico

Este encuentro se divide en dos momentos con distintas actividades.

#### Primer momento

En este momento se comenzará debatiendo respecto de las actividades propuestas para el espacio asincrónico pasado. Se hará un repaso por los contenidos vistos en el encuentro anterior y se debatirá sobre las actividades que quedaron pendientes en los espacios anteriores.

Se introducirá el concepto de condicionales: mediante preguntas y respuestas se le pide a los/as estudiantes que hagan enunciados condicionales de la vida real (si me siento mal, no voy a la escuela, si apruebo me felicitan, etc) y se realiza su traducción lógica a la escritura en lenguaje de programación.



## Segundo momento

Se comenzará a profundizar el conocimiento hablando de los operadores de comparación, y el uso de if-else anidados. A su vez se hablará del concepto de parseo y conversión de datos. El uso de Number y parseInt para parsear las respuestas recibidas de un prompt, y poder operar con libertad a la hora de realizar cuentas.

Similar al encuentro anterior, se trabajará mediante enunciados, pero esta vez abordando temáticas más complejas que respondan a lo ya aprendido. Si bien la temática y consigna del problema puede variar dependiendo de la comisión, algunas propuestas son:

### Actividad 1 Camino a elección

Se plantea una situación de juego en donde las/los estudiantes están en un bosque sombrío... se plantea una situación límite en donde tienen que escapar y se encuentran con dos caminos de los cuales solo uno es correcto. Si eligen el correcto, salvan su vida y si no... la pierden.

Posibilidad: Dos caminos

Conocimientos: *Prompt*, variables, *if*, *else*.

### Actividad 2 Géneros

La mayoría de los formularios de inscripción a eventos suelen pedir nombre, apellido, edad, DNI y género. Se plantea la situación de desarrollo de un formulario en donde mediante el uso de *prompt* vamos a preguntar por el género de la persona:

- Si elije "M" el género será masculino
- Si elije "F" el género será femenino
- Si elije "O", el género será otros

Se debate una reflexión acerca del género... **¿Qué otros géneros existen? ¿Es correcto seguir pidiendo el género en formularios? ¿Qué géneros están reconocidos legalmente en Argentina?**

Siempre debemos recordar que nuestro deber como desarrolladores/as conlleva un actuar de manera responsable dado que nuestras acciones pueden impactar a cientos/miles de personas.

Posibilidad: Tres opciones

Conocimientos: *Prompt*, variables, *if*, *else*, *if else*.

Con la práctica de los ejercicios en el espacio sincrónico el/la mentor/a se asegurará de resolver dudas y consultas que surgirán por parte de las/os estudiantes.

Se recomienda el uso de *Code to Graph* (<https://crubier.github.io/code-to-graph/>) para poder *plasmar* de forma visual un programa lógico mediante diagramas de flujo.

### Encuentro asincrónico

El objetivo en este encuentro es que los/as estudiantes sigan trabajando en la creación de programas listados en las actividades que resuelvan problemas reales que a su vez mantengan una dinámica lúdica. Trabajar en un problema integral complejo nunca es la idea, pero sí lo es ir complejizando paso a paso.

Se realizarán las actividades *quiz*, planteadas en la plataforma, con la intención de reforzar el conocimiento adquirido.

### Actividad Integral Código Secreto

En muchos juegos accedemos a beneficios mediante un código. Para este programa se plantea una situación de juego en donde tenemos distintos personajes:

- Hechiceros y hechiceras
- Dragones
- Guerreros y Guerreras

El programa le pedirá al usuario que ingrese qué tipo de personaje es, luego le pedirá el código. Dependiendo del tipo de personaje que sea, y si el código es correcto o no, se le darán beneficios propuestos por los estudiantes a cada personaje.

Todo el programa se desarrollará mediante el uso de *prompts* y *console.log*.

Posibilidad: Múltiple

Conocimientos: Prompt, variables, if, else, if else anidados, operador.



## 04. Semana 3

### Objetivos

- Reforzar conocimientos en el uso de condicionales.
- Conocer el patrón de *fallback* con el *else*.
- Aplicar el uso y conocer el alcance y posibilidades al desarrollar *if,else* anidados.
- Conocer el uso del objeto *Math* y su implicancia en diversas operaciones de programación (acceder a elementos, números aleatorios, redondeos, etc)
- Conocer el uso de *template literals*, así como distintos métodos de incremento (*++*, *+=*) que se usarán a lo largo de todo el desarrollo.

### Ejes temáticos

- Refuerzo de operadores Lógicos
- Evaluación *true/false*
- Uso del *else* como *fallback*
- *If else* anidados
- Patrón de diseño por negación
- Template literals (Propiedades de ES6)
- Objeto *Math* (*Random, Floor, etc*)
- Suma++;
- +=suma;

### Encuentro Sincrónico

Este encuentro se divide en dos momentos con actividades diferenciadas.

#### Primer momento

Este encuentro comenzará retomando lo aprendido en el anterior, y haciendo preguntas sobre los enunciados trabajados en el encuentro anterior.

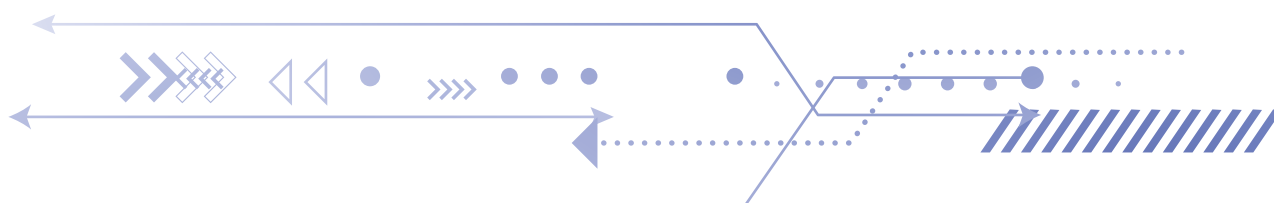
Se enseñará el uso del *else*, pero esta vez como patrón de *fallback*, es decir cómo resultado final si es que no se cumplen las condiciones planteadas en el programa. Si bien este conocimiento lo van incorporando es momento de darle nombre a estos patrones.

Se explicará más en profundidad la evaluación de una condición. Entender que la evaluación parte sobre el verdadero o falso de una condición "*if(true)*". De esta forma podrán entender la evaluación por booleanos, o incluso no caer en el error de evaluación ante un *if(0)* o *if("")*. Este momento es adecuado para comenzar a evaluar expresiones que tal vez resulten "extrañas" pero que están permitida en *Javascript*:

#### Actividad 1 Analizando Expresiones

¿Qué dan los siguientes resultados?

<code>true*true</code>	¿Qué da?
<code>false*false</code>	¿Qué da?
<code>if(0) {}</code>	¿Ejecuta el código?
<code>NaN == NaN</code>	¿Es correcto?
<code>if("") {}</code>	¿Ejecuta el código?



## Segundo momento

Se introducirá el objeto *Math* para generación de números aleatorios y para redondeos tanto hacia arriba como hacia abajo.

Se incorporarán contenidos de sumatoria, ya introduciendo de a poco el uso del `++` y `+=` para utilizar en el próximo encuentro en los bucles, en donde es necesario un incremental.

El uso de *template literals* será de gran ayuda para evitar errores al concatenar *strings* y números. Dicha propiedad, propia de ES6, tiene además su similitud en diversos lenguajes de programación lo cual les da un saber extensible a otros lenguajes.

Similar al encuentro anterior, se trabajará mediante enunciados, pero esta vez más complejos, adaptándose a los ejes temáticos de este encuentro.

## Actividad 2 Números Aleatorios

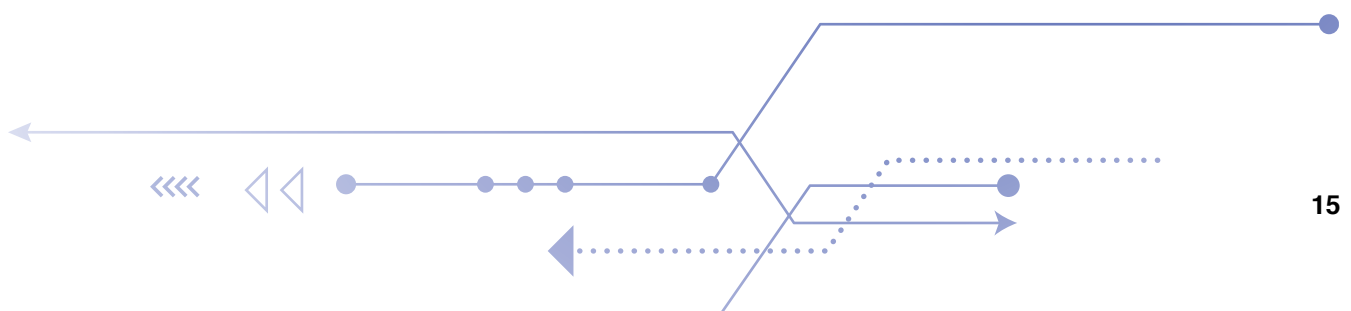
Se creará un programa que cree números aleatorios. En una primera instancia el programa creará números aleatorios (con decimales) de 0 a 1. En una segunda instancia del programa deberá crear números aleatorios (con decimales) de 0 a 20. En una tercera instancia el programa deberá crear números aleatorios (sin decimales) de 0 a 15 y finalmente en la última instancia se deberán crear números aleatorios de 15 a 30.

### Encuentro asincrónico

Para este encuentro se espera que los/as estudiantes entreguen diversas producciones planteadas en las actividades integradoras.

Las actividades a realizar pueden variar dependiendo la comisión y lo requerido por la misma, sin embargo se recomienda partir de las mismas bases que se proponen en la sección de actividades: la forma del programa puede variar, son los aprendizajes los que se deben asegurar para seguir el correcto seguimiento del programa.

Se realizarán las actividades quiz, planteadas en la plataforma, con la intención de reforzar el conocimiento adquirido.



### Actividad Integradora 1 Semáforo

Cuando aprendimos a ver el semáforo nos enseñaron que había tres situaciones:

- Luz roja: No pasar
- Luz amarilla: Precaución
- Luz verde: Permitido pasar

Dado esta situación y forma de aprendizaje, vamos a desarrollar un programa de tipo semáforo. El usuario deberá ingresar mediante un *prompt* el estado del semáforo, y mediante un *alert* diremos si puede cruzar la calle o no.

### Actividad Integradora 2 Restando Puntos

En todos los juegos se pierden y ganan puntos. En este programa se desarrolla un juego de preguntas y respuestas que parten de las situaciones reales y actuales de los/las estudiantes. Se plantean preguntas con opciones y mediante un *prompt* se elige la respuesta. Dependiendo de la respuesta, se quitan puntos o se suman. Los jugadores empiezan con un total de 50 puntos, el puntaje se suma o resta de a 10.

Esta dinámica plantea la división en dos grupos en donde un/a estudiante comparte pantalla y debate con el equipo respecto de las preguntas y respuestas a utilizar. El/a mentor/a se pasa por los grupos y asiste las dudas y consultas. Una vez realizado el juego se unen todos y todas en la sala principal y juegan al juego de los respectivos equipos.

Posibilidad: Dos caminos

Conocimientos: *Prompt, variables, if, else.*



## 05. Semana 4

### Objetivos

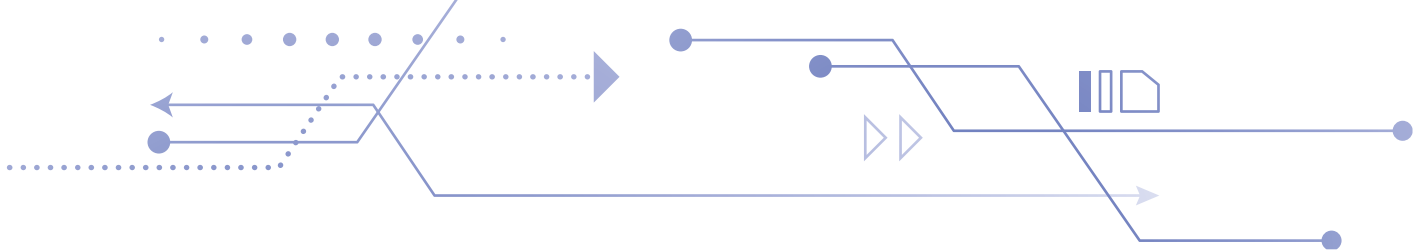
- Conocer los *arrays* y su uso.
- Acceder a distintos elementos dentro de un *array*.
- Visualizar situaciones reales que resuelven problemáticas con el uso necesario de *arrays*.

### Ejes temáticos

- *Arrays*
- *Índice*
- *Length*
- *Matrices*
- *Array* con múltiples valores
- Métodos *push* y *pop*
- Método *indexOf*







## Encuentro Sincrónico

Este encuentro se divide en dos momentos.

### Segundo momento

Tomando como base el saber almacenar datos dentro de variables, se introducirá el concepto de *arrays*. Siempre es recomendable introducir estos conceptos mediante analogías, o mediante apoyo gráfico:

- > Un ropero
- > Una estantería
- > Un conjunto de cajas
- > Etc

Se debe tomar especial cuidado en la diferenciación del comienzo en 0 de los índices de un *array*, en contraposición al *.length* que nos devuelve la cantidad de elementos del *array*.

**En esta instancia, suma valor resolver “micro-ejercicios” planteados en clase, con respuestas y preguntas del tipo:**

*¿Qué me devuelve este array evaluado en cero?*

*¿Cómo accedo al nombre “Juan” en este array de nombres?*

*¿Qué me devuelve el *length* de este array?*  
*Etc.*



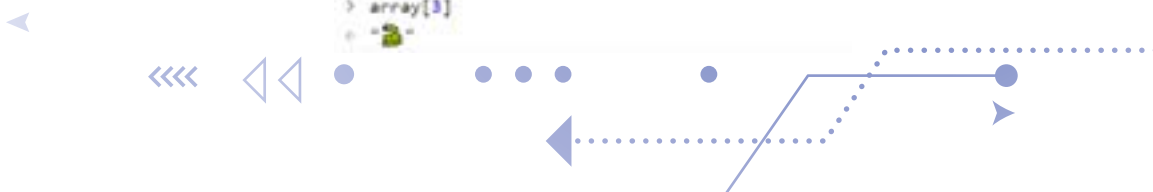
Mencionar qué es lo que pasa cuando evaluamos un array con un valor negativo, o qué es lo que pasa cuando nos excedemos del *length* a la hora de evaluar un *array* (*undefined*) son comentarios esperables a realizar por parte de el/la mentor/a.

Se realizarán actividades para ver el correcto aprendizaje de estos nuevos conceptos. Las actividades, similares a los encuentros anteriores, parten de resolver problemas planteados en distintos enunciados. Algunos de los ejemplos de estas actividades pueden ser:

### Actividad 1 Descubre al animal

En este programa de introducción a *arrays* se plantea hacer una búsqueda de los animales que se encuentran en el *array*. Aprovechando que los emojis son caracteres que se pueden guardar se hará un juego para comprender la idea de índices dentro de listas. El programa consiste en armar un *array* lleno de emojis, y los/las estudiantes deben encontrar dónde están los animales.

```
> let array = ["🍌", "🐱", "🐶", "🐼", "🐸", "🐙"];  
< undefined  
> array[0]  
< "🍌"  
> array[3]  
< "🐼"
```



Conocimientos: *Arrays*, índices

Se introducirán enunciados difíciles de resolver si no conocen los métodos de *arrays*, que dará como pie la posibilidad de introducir nuevos conceptos de métodos. Al ser el primer encuentro con *arrays*, los métodos planteados serán reducidos, no se discutirá en este encuentro sobre todos los métodos disponibles.

## Actividad 2

Siempre al final

Con el fin de entender e incorporar el concepto del *length*, este programa propone armar una lista con todos/as los/as estudiantes del curso e imprimir con un alert a el/la último/a. Luego, mediante un *push* se agregará un/a nuevo/a estudiante (como ejemplo un cambio de comisión) y el programa debe seguir funcionando de la misma manera. Es decir que se busca comprender el uso del *array.length* y no evaluar con números “*hardcodeados*”.

Conocimientos: *Arrays*, índices, *length*

## Encuentro Asincrónico

En este encuentro se preguntarán todas las dudas y consultas que haya respecto a los aprendizajes adquiridos. Es muy importante que El/a mentor/a tengan un seguimiento de cada estudiante particular sobre su progreso así como sus avances y entregas.

Estos encuentros asincrónicos son buenos espacios para que los/as estudiantes realicen las actividades integradoras que ponen en juego todo el conocimiento adquirido previamente.

Se realizarán las actividades *quiz*, planteadas en la plataforma, con la intención de reforzar el conocimiento adquirido.

## Actividad Integradora

¿Qué hay en cada expresión?

Para este ejercicio se plantean múltiples expresiones creadas por el/la menor/a para analizar en conjunto con las/los estudiantes. Cada expresión permitirá analizar propiedades y métodos de los arrays. Algunos ejemplos podrían ser de tipo:

Dado el siguiente *array*

```
let nombres = ["Juan", "Josefina", "Estefanía", "Facundo", "Alejo", "Romina"];
```

¿Qué dan las siguientes expresiones?

```
nombres[0];
```

```
nombres[nombres.length];
```

```
nombres.push("Ale");
```

```
nombres.shift();
```

```
nombres.pop();
```

Conocimientos: *Arrays*, índices, *length*, métodos.



## 06. Semana 5

### Objetivos

- Profundizar el objeto de tipo array, sus métodos y propiedades
- Introducir al bucle de iteración *for* y lograr la iteración sobre los arrays

### Ejes temáticos

- Método *slice*
- Método *splice*
- Método *join*
- Método *split*
- *For loop*
- Variable *i,j,k*
- *Length -1* (en condición de iteración)

### Encuentro Sincrónico

Este encuentro se divide en dos momentos.

#### Primer momento

Este bloque comenzará de manera práctica. Se trabajará sobre el contenido dado en el encuentro anterior y se extenderá la explicación de métodos disponibles en los *arrays*.

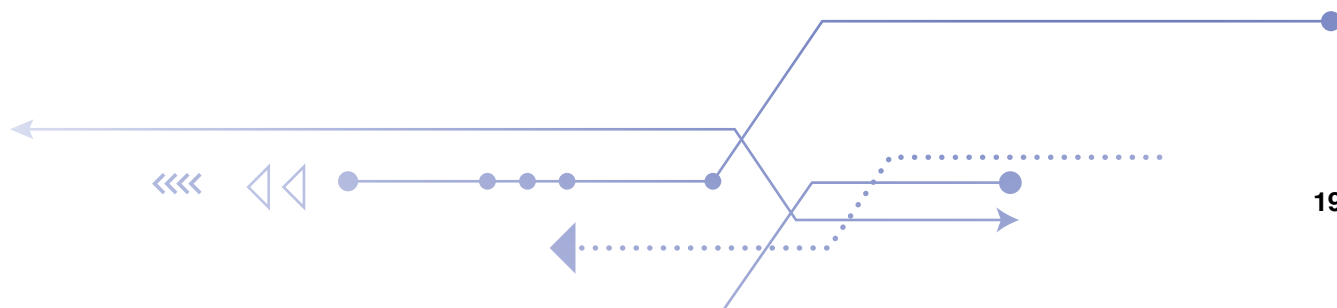
Usar *MDN* como referencia es una gran posibilidad dado que en *MDN* existe mucha documentación oficial de *JS* (Ir a *MDN*)

Con una explicación de algunos de estos métodos, se comenzará la instancia práctica de desarrollar ejercicios en el encuentro. Conforme avancen las/os estudiantes, se irán complejizando las consignas hasta llegar a la necesidad del bucle *for*.

#### Segundo momento

Sin saberlo, en esta instancia las/os estudiantes cuentan con todo el conocimiento necesario para comenzar a abordar las problemáticas más complejas dentro de un lenguaje de programación. Ya conocen condicionales, ya saben de variables, saben del *length* de los *arrays* y saben del uso de incrementales. Es de la mano de todo este conocimiento que se introduce el bucle *for-loop*.

VERSION PRELIMINAR



Bajo el concepto *DRY (Don't repeat yourself)* se introducirá la idea de código repetitivo y performante. Es sumamente recomendado explicar el concepto de bucles mediante gráficos y realizando un paso a paso, reiteradas veces hasta que el concepto quede adquirido. Muchos de estos ejercicios se aprenden mediante la escritura y repetición.

Una vez explicado el contenido teórico-práctico del *for-loop*, se realizarán actividades que tengan que ver con el contenido aprendido. Una de las actividades propuestas es el randomizer:

### Actividad 1

#### Integrador

#### Randomizer

Este programa tiene un doble sentido: por un lado el de seguir desarrollando los conocimientos de programación, y por otro lado generar una herramienta que puede ser útil a lo largo del trayecto.

Este programa consiste en armar una lista con todos los nombres de las/los estudiantes. El *array* se recorrerá y dará el nombre de uno, de manera aleatoria. Este programa se puede usar para hacer preguntas, respuestas, pedir participación en el encuentro sincrónico y otras situaciones en donde utilizar el programa desarrollado por el grupo refleja la utilidad de la programación y como resuelve problemas de la vida cotidiana.

Conocimientos:

*Arrays, índices, length*, condicionales, operadores lógicos

### Encuentro asincrónico

Se realizarán pequeños programas y ejercicios que respondan a enunciados planteados en el encuentro sincrónico.

En este encuentro se plantea el primer programa complejo que se repetirá a lo largo de los distintos módulos venideros, el piedra papel o tijera comenzará a desarrollarse con el uso de condicionales, en los módulos siguientes se comenzará a desarrollar utilizando funciones para finalmente desarrollarlo con una interfaz gráfica. Este modelo de actividad muestra de manera clara un proyecto integral realizado desde cero, pasando por su lógica y terminando con una interfaz.

Se realizarán las actividades *quiz*, planteadas en la plataforma, con la intención de reforzar el conocimiento adquirido.

**Actividad Integradora**  
Jugando a “Piedra, papel o tijera”



**Jugando a “Piedra, papel o tijera”**

Como ejercicio integrador de variables, condicionales y operadores lógicos, sumado al objeto *Math*, se plantea realizar un programa de piedra, papel o tijera.

1. Se le pide al usuario que seleccione una opción: P, Pa, o T
2. Luego de seleccionar, el programa seleccionará uno de forma aleatoria
3. Se le informará al usuario qué eligió el programa, qué eligió el usuario y quién ganó la partida.

Conocimientos:

*Arrays*, índices, *length*, condicionales, operadores lógicos.



**07. Semana 6**

**Objetivos**

- Compartir los avances a partir de la exposición de sus producciones grupales.
- Debatir nuevas soluciones.
- Participar en forma activa y crítica.
- Defender y justificar lo realizado.
- Promover la disposición colaborativa
- Despertar entusiasmo por los desarrollos futuros.

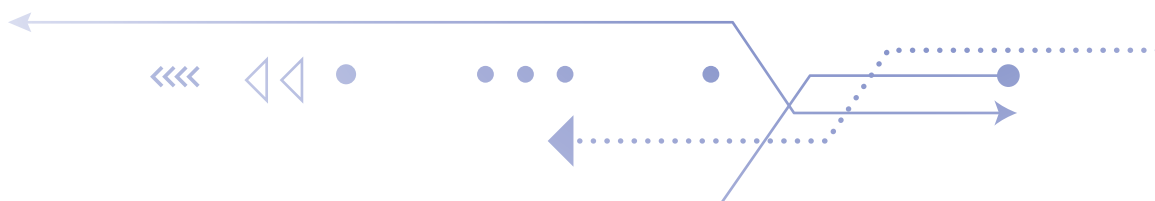
**Ejes temáticos**

- Exposición de proyectos
- Comunicación de saberes y experiencias
- Respuestas y preguntas a dudas que surgieron durante las diversas producciones

**Encuentro Sincrónico**

Se buscará que los/as estudiantes expongan, defiendan sus ideas, cuenten cómo fue el proceso, compartan soluciones que se han planteado y entiendan la diversidad de soluciones a una misma propuesta.

Resultará fundamental potenciar este espacio para que resulte generador de nuevas ideas, y aporte formas de optimización del código de los programas que se van generando.



**Los/as estudiantes compartirán pantalla y presentarán los resultados propuestos.**

Este encuentro es un buen punto de reflexión para que El/a mentor/a pregunte de manera exhaustiva cómo se sienten los/as estudiantes respecto del trayecto. A su vez esta es una buena instancia para asegurarse que las entregas se están realizando de manera correcta y, en caso de no ser así, averiguar por qué no se cumplen e ir pidiendo los entregables conforme los encuentros van avanzando así todos/as los/as estudiantes están al día.

**Encuentro asincrónico**

Como actividad final los/as mentoras/es realizarán una serie de preguntas cuya temática serán todos los saberes adquiridos a lo largo de este módulo. Se dividirán en grupos y mediante un sistema de puntos existirá un grupo ganador dependiendo la cantidad de preguntas que contesten de manera acertada.

Para limitar la competencia se realizará una rotación de integrantes que funcionará para nivelar los saberes dentro de los distintos equipos.

En este juego se recomienda que las preguntas no sean meramente de índole técnico sino que los/as mentoras/es pueda generar pregunta que tengan complicidad con la comisión. Recuperar momentos graciosos, tensos, de catarsis, de conocimiento entre estudiantes, etc. Que sumen al carácter lúdico de la propuesta.