

Módulo III

DESARROLLO WEB II

• • NIVEL INTERMEDIO

Una, otra
y otra vez



Orientaciones para los/as mentores/as

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

=

Una, otra
y otra vez

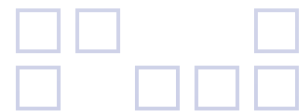
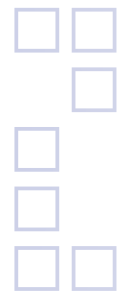
Módulo III

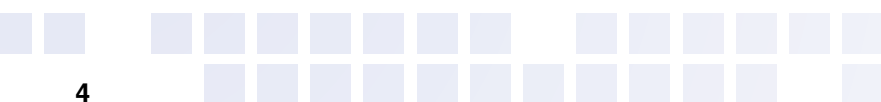
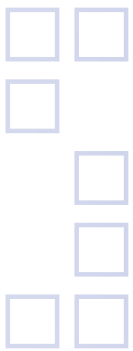
DESARROLLO WEB II

• • NIVEL INTERMEDIO



Orientaciones para los/as mentores/as





Índice

01.	Sobre el Módulo	7
	Sobre los recursos	7
	Sobre el OPEN LAB	7
	Sobre las actividades dentro de la plataforma	7
02.	Semana 1	8
	Objetivos	8
	Ejes temáticos	8
	Encuentro Sincrónico	8
	Actividad 1. Aprendemos a contar, en código	10
	Actividad 2. Nos saludamos	10
	Actividad 3. No dejes nada vacío	10
	Actividad 4. Me olvidé la clave	10
	Encuentro asincrónico	11
	Actividad Integradora 1. Juegos y categorías	11
	Actividad Integradora 2. Piedra, papel o tijera, 2	
03.	Semana 2	11
	Objetivos	11
	Ejes temáticos	11
	Encuentro Sincrónico	11
	Actividad 1. ¿Dónde está Wally	12
	Actividad 2. Contando caracteres	12
	Encuentro Asincrónico	12
	Actividad Integradora. Cuarentena y descuentos	13
04.	Semana 3	13
	Objetivos	13
	Ejes temáticos	14
	Encuentro Sincrónico	14
	Actividad 1. Saludo múltiple	14
	Actividad 2. ¿Qué mes es?	14
	Actividad 3. Saludar a...	14
	Encuentro Asincrónico	15
	Actividad Integradora	15
05.	Semana 4	16
	Objetivos	16
	Ejes temáticos	16
	Encuentro Sincrónico	16
	Actividad 1. Números pares	17
	Actividad 2. Es divisible por	18
	Actividad 3. Hoy me siento...	

	Encuentro Asincrónico	18
	Actividad Integradora. Calculadora	18
06.	Semana 5	19
	Objetivos	19
	Ejes temáticos	19
	Encuentro Sincrónico	19
	Actividad 1. <i>Youtubers</i> - Mi primer <i>objeto</i>	20
	Actividad 2. Mis intereses	20
	Actividad 3. Adivinanzas	20
	Encuentro asincrónico	21
	Actividad Integradora. Comentarios de <i>YouTube</i>	21
07.	Semana 6	21
	Objetivos	21
	Ejes temáticos	21
	Encuentro Sincrónico	21
	Encuentro Asincrónico	22
	Actividad D. Mural digital	22

01. Sobre el Módulo

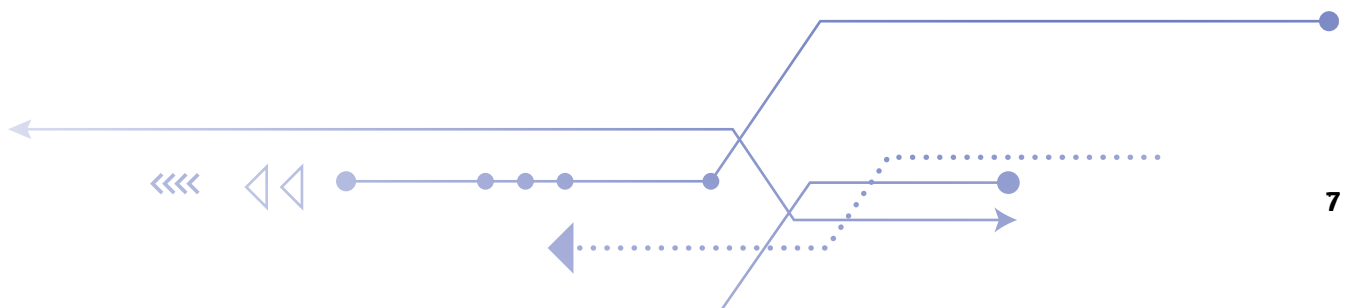
La programación tiene diversas aristas, algunos de los grandes beneficios son la posibilidad de almacenar datos y realizar tareas repetitivas. En el módulo anterior los/as estudiantes ya adquirieron los saberes relacionados con la introducción a la programación, en este módulo esos saberes se amplían y se pone el foco en tres grandes ejes: la posibilidad de realizar tareas repetitivas, el almacenamiento de datos en tipos de datos no primitivos como lo son *objetos* y *arrays*, y finalmente la reutilización de código gracias a las *funciones*.

Mediante el uso de funciones, se espera que los/as estudiantes comiencen a profesionalizar la escritura de código y hacer código eficiente y de calidad. Se comienza a pensar en funciones abstractas que realizan tareas sin necesidad de pertenecer a un programa específico. Se introducen los conceptos de *DRY* (*Don't repeat yourself*) para reducir el código verbose y mantener el foco en la simpleza y eficiencia.

Mediante el uso de *for* y *whiles*, se espera que los/as estudiantes sean capaces de resolver problemas de mediana y alta complejidad gracias a la posibilidad de realizar tareas repetitivas. Así mismo con la introducción de estos nuevos conceptos podrán realizar diversos tipos de validaciones para generar código eficiente.

Del módulo anterior los/as estudiantes ya tienen nociones del uso de *arrays*, sin embargo en este módulo se refuerza y se introduce el tipo de datos de *objetos* que es crucial en el mundo del desarrollo. La posibilidad de acceder a propiedades y métodos de los *objetos* es de suma importancia para comenzar a trabajar pensando en consumir servicios **REST**.

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:



Presentaciones



Videos



Guías



Códigos



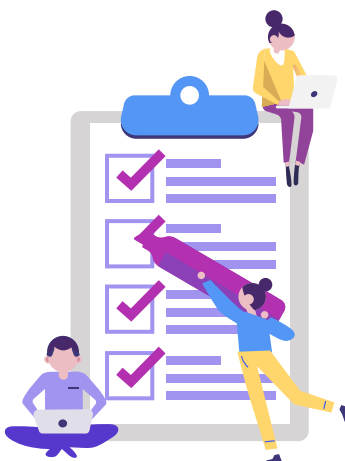
Formato Web

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 como 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.





Semana 1

Objetivos

- Comprender la importancia del uso de bucles para repetir tareas
- Aprender el uso del *bucle while*
- Aprender el uso del *bucle do-while*
- Comprender la importancia de una condición de corte para no entrar en loops infinitos
- Utilizar variables de tipo incremental/acumuladores

Ejes temáticos

- *while*
- *do*
- *bucles*
- *acumulador*
- uso del ++,
- condición de corte
- variable *i*
- evaluación en *i*

Encuentro Sincrónico

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

Primer momento

En este momento se explicará de manera teórica la importancia de los bucles, así como ejemplificación del uso de los mismos. Se mencionará el uso del *while* y el uso del *do,while*.

Resulta muy gráfico comenzar a mostrar las posibilidades de este recurso mediante el uso de consola cómo por ejemplo el armado de un programa que cuente hasta un número específico:

Actividad 1

Aprendemos a contar, en código

Pedirle al usuario que ingrese un número máximo e imprimir por consola todos los números enteros que existen desde el 0 hasta dicho número ingresado.

Mediante el armado de este programa (o programas similares) es muy importante reforzar el concepto de acumulador y las operaciones de suma ++, --, += y -=. A su vez, este tipo de programas iniciales resultan adecuados para explicar **la importancia de la condición de corte** para evitar *loops* infinitos.

Recordando lo aprendido en el módulo anterior relacionado al almacenamiento de datos en *arrays* y su *iteración*, se propondrá una actividad para poder reforzar estos conceptos e iterar sobre un *array* pero esta vez mediante el uso de *while*. Como ejercicio inicial se puede plantear un saludo a todos/as los/as estudiantes del curso.

Actividad 2

Nos saludamos

Se creará un *array* con todos los nombres de los/as estudiantes del curso y mediante un *bucle while* y el uso de concatenación se saludará a todas las personas presentes.

Mediante este programa se hace refuerzo de la importancia de la condición de corte y el uso de la propiedad *length* disponible en los *arrays*.

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 relacionado específicamente al uso de bucles de repetición. En esta instancia también se comenzará a pensar en código validado, articulando los saberes adquiridos en módulos anteriores:
Todas estas actividades se realizarán en conjunto tanto mentores/as como estudiantes.

Actividad 3

No dejes nada vacío

Pedirle a un usuario que ingrese su edad, si no ingresa nada se deberá volver a preguntarle su edad. Si ingresa algo que no sea un número, deberá aparecer un mensaje de error.

Actividad 4

Me olvidé la clave

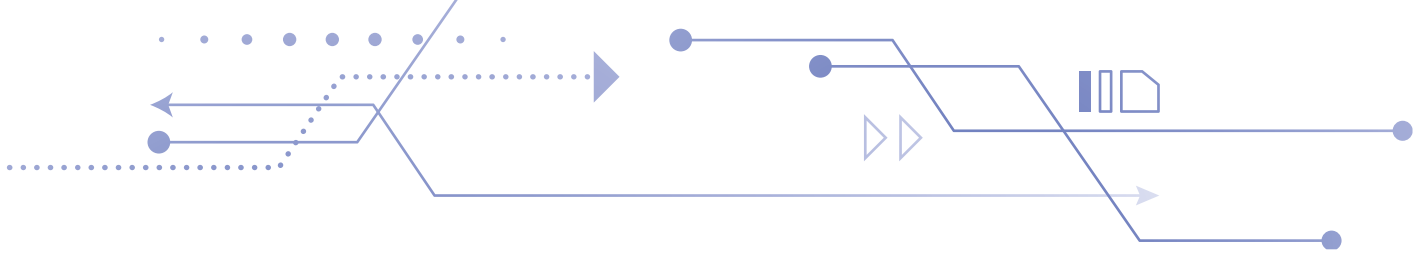
Se generará una constante dentro del programa que almacene una contraseña. Se pedirá al usuario que ingrese la contraseña. Si la clave es ingresada más de 5 veces de manera incorrecta, un cartel diciendo “ha excedido el límite” deberá aparecer. Si es correcta dentro de las cantidades permitidas, un cartel diciendo “Bienvenido/a” deberá aparecer.



Es importante:

Generar un espacio de preguntas que den lugar al debate así como un espacio de reflexión para poder dejar planteado el cómo resolver el problema, y que luego los/as estudiantes lleven a cabo el código y se encuentren con las adversidades (que irán solucionando) de escribir la sintaxis. El proceso debe ser siempre gradual y acompañado.

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



Encuentro asincrónico

Para este encuentro se plantean actividades entregables. De a poco las actividades se irán orientando hacia lógica de juegos. Ahora que los/as estudiantes tienen conocimientos suficientes para comenzar a articular todo lo que fueron aprendiendo hasta el momento, es importante que empiecen a ver el desarrollo como un conjunto del todo.



Como en todos los módulos, 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 I

Juegos y categorías

En un juego tenemos 3 personajes: héroes, hechiceros/as y guerreros/as. Crear un programa que pregunte 5 nombres de personajes y, de manera aleatoria, muestre un mensaje y diga “Bienvenido/a $\${usuario}$, tu personaje es $\${personaje}$ ”

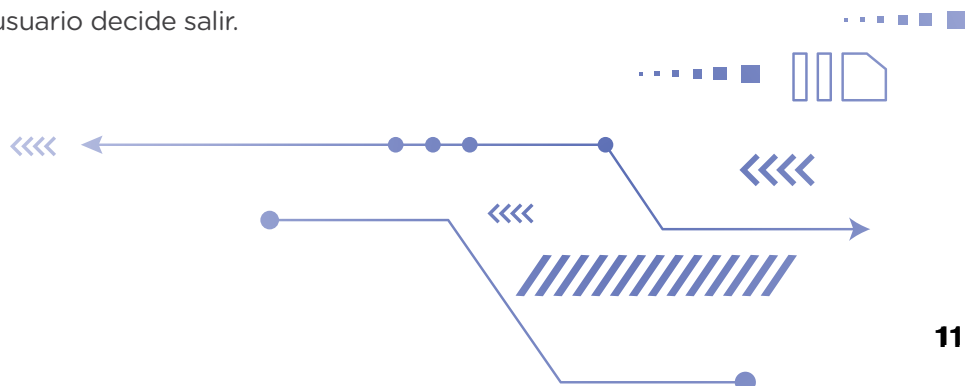
Actividad Integradora II

Piedra, papel o tijera, 2

En un *array* se guardarán las opciones “P”, “Pa”, “T”. Se le pedirá al usuario si quiere jugar, o salir. Siempre que elija jugar, el programa se ejecutará, si elige salir el programa dirá “Usted salió”.

Cuando el usuario juega, se le pide que elija una opción, “P”, “Pa”, “T”. Ese resultado se guardará en una variable y de genera una selección aleatoria. El juego será entre el usuario y la PC.

A diferencia del juego realizado en el módulo anterior, este deberá estar compuesto por *arrays* y tener el bucle de juego constante hasta que el usuario decide salir.





Semana 2

Objetivos

- Reforzar el entendimiento y uso del *for-loop*
- Utilizar el *for-loop* de manera incremental como de manera decreciente
- Asociar las similitudes del uso del *for-loop* con el *while*, y sus diferencias
- Comprender la importancia de una condición de corte para no entrar en *loops* infinitos
- *Bucles* anidados

Ejes temáticos

- *while*
- *do*
- *for*
- diferencias entre *while* y *for* (flexibilidad de uno por sobre el otro)
- acumulador
- uso del ++, --, +=, -=
- condición de corte
- variable *i*
- variable en *j*
- evaluación en *i* y en *j*

Encuentro Sincrónico

Este encuentro se divide en dos momentos con distintas actividades.

Primer momento

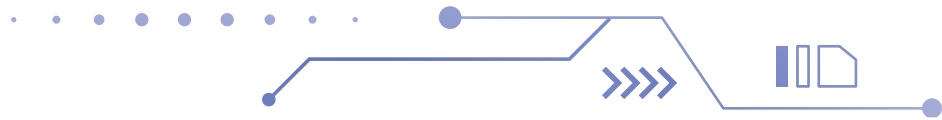
En este momento se comenzará debatiendo respecto de las actividades integradoras en el espacio asincrónico del módulo anterior, se analizarán las soluciones propuestas, los errores y aciertos y de esta manera se hará un repaso por los contenidos vistos en el encuentro anterior.

Se introducirá el concepto del *for anidado* para poder recorrer valores guardados en matrices. A su vez, se reforzará el concepto de bucles y se hará hincapié en el correcto entendimiento de estos saberes dado que son bases para el desarrollo futuro.

Segundo momento

Se trabajará mediante enunciados que harán referencia a los saberes planteados en los ejes temáticos, 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

¿Dónde está Wally

Dada esta lista: `const nombres = ["Juan", "Pedro", "Wally", "Sofía", "Amelia", "Wally"];`

Mostrar los índices dónde está *Wally* utilizando el método `indexOf`.

Actividad 2

Contando caracteres

Cuando trabajamos del lado del *front-end*, lo más seguro que pase es que recibamos el contenido de algún servicio. ¿Qué quiere decir esto? Que lo que hay dentro del `<h1>` o dentro de los `<p>` no lo escribimos nosotros, sino que viene de una base de datos que cargó la gente de contenido.

Resulta que armamos un diseño de *cards* pero tenemos la limitación de que los `<h2>` que hay dentro de las *cards* solamente pueden tener hasta 20 caracteres, sino rompe nuestro diseño.

Debemos armar un programa que contenga la lista de los destinos más populares (los nombres de los lugares). Si alguno de estos lugares tiene más de 20 caracteres, arrojar un error diciendo "Se excedió la cantidad de caracteres para el título de las *cards*".

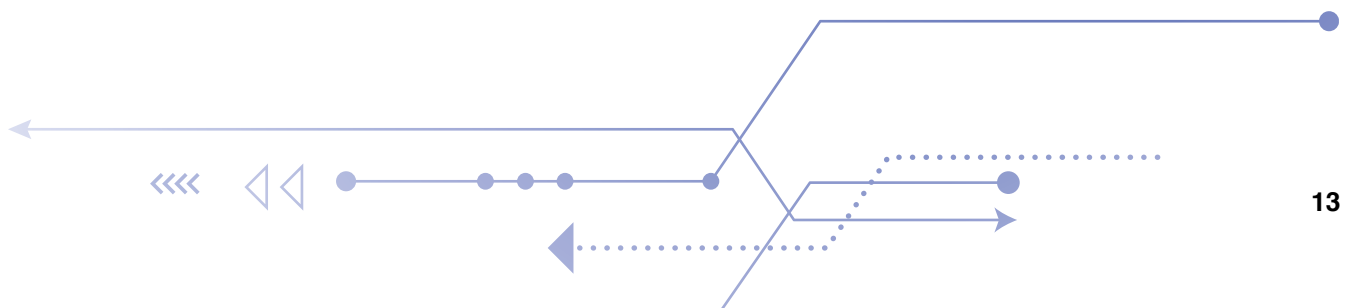
Encuentro asincrónico

El objetivo de este encuentro es que los/as estudiantes puedan ofrecer soluciones a los enunciados listados en las actividades que resuelvan problemas reales y que pueden aparecer y son coherentes con problemas de la vida laboral futura.

Trabajar en un problema integral complejo nunca es la idea, pero sí lo es ir complejizando paso a paso los tipos, es por eso que radica en el/a mentor/a el poder definir la complejidad del ejercicio complejizando el enunciado o manteniendo las partes más simples.

Para este encuentro se trabajarán sobre programas que podrían pasar en situaciones profesionales.

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



Actividad Integradora

Cuarentena y descuentos

Durante la cuarentena los pedidos online incrementaron en gran medida. Todas las tiendas que tienen un *e-commerce* comenzaron a lanzar descuentos para captar más clientes.

Ustedes como futuros desarrolladores/as están a cargo del programa que realiza descuentos. En este programa existen **3 cupones**:



Partiendo de una lista (*array*) de 7 productos en el carrito y otra lista (*array*) con los precios de los 7 productos.

Hacer un programa que:

1. Muestre el nombre de los productos y el precio correspondiente
 - a. Utilizar *console log*
2. Muestre el precio final
 - a. Precio final = x
3. Pida al usuario que ingrese un código de descuento
 - a. Utilizando *prompt*
4. Mostrar el nuevo precio final con el descuento aplicado



Semana 3

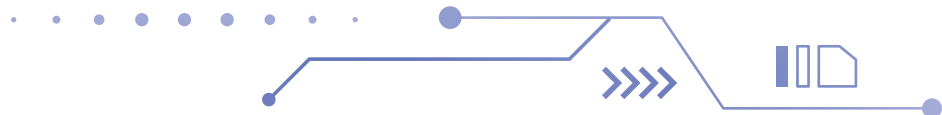
Objetivos

- Conocer el uso de *funciones* y comprender su alcance
- Optimizar el código por medio del uso de *funciones*
- *Función* global como *función* en variable
- Comenzar a comprender el término *scope* de una *función*
- Desarrollar el concepto de función abstracta
- Concepto de *DRY*

Ejes temáticos

- Palabra *function*
- Función global
- Función como variable
- Código optimizado
- Empaquetar líneas de código
- Abstracción de *funciones*
- *Scope*
- *DRY*





Encuentro sincrónico

Este encuentro se divide en dos momentos con actividades diferenciadas.

Primer momento

Este encuentro comenzará retomando lo aprendido, haciendo preguntas sobre los contenidos aprendidos y sobre las actividades integradoras a resolver. Siempre es recomendado abrir al debate, diálogo, preguntas y respuestas para conocer qué problemas encontraron al resolver las actividades y de esta manera realizar un repaso por los contenidos del encuentro anterior.

Una vez realizado el repaso, se explicará qué es una *función*, para qué sirve y se les mostrará que sin darse cuenta lo han utilizado en diversos momentos del trayecto como por ejemplo con los métodos de *arrays*, *console.log*, *alert*, etc.

Se recomienda en este primer momento explicar la teoría de *funciones*, la palabra reservada *function* y cómo se definen las *funciones*. Para esta primera instancia no hace falta asignar funciones a variables dado que tener las *funciones* disponibles en todo momento (gracias al *hoisting*) puede ser beneficioso en esta curva de aprendizaje.

Actividad 1 Saludo múltiple

Se desarrollará un programa cuyo único fin sea saludar, este saludo lo hará mediante el uso del *console.log* y se ejecutará la *función* varias veces para entender cómo la *función* sirve para la reutilización de código.

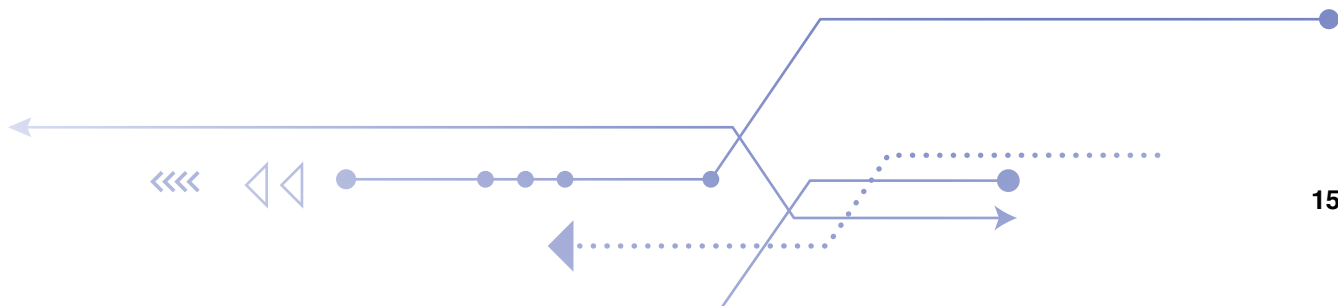
Segundo momento

Se realizarán diversas actividades con el fin de ir comprendiendo paso a paso el uso de *funciones*, su definición y ejecución. En esta instancia se realizarán programas simples para comprender el correcto uso de las mismas, conocer la sintaxis y comenzar a encontrar los primeros errores.

Se explicará el uso de parámetros, qué son, para qué sirven y su similitud con las variables. La explicación se realizará y fundamentará mediante la aplicación práctica de actividades del siguiente estilo:

Actividad 2 ¿Qué mes es?

Se realizará una *función* en donde se reciba como parámetro un número del 1 al 12 (a esta altura del trayecto es esperable que los/as estudiantes puedan realizar distintos tipos de validaciones) y el programa indique qué mes es.



Actividad 3

Saludar a...

Tomando la actividad realizada en el encuentro anterior, esta vez se saludará a la persona cuyo nombre sea recibido como parámetro de la *función*.

Encuentro asincrónico

Para este encuentro se plantearán las dudas y consultas que tengan sobre *funciones* y cualquier temática en relación a los módulos ya cursados. Es esperable que conforme avancen los módulos, los/as estudiantes presenten dudas sobre los contenidos vistos en módulos pasados, y necesiten reforzarlos para poder continuar con lo nuevo. Es tarea de el/la mentor/a el poder entender el estado de aprendizaje de cada estudiante para guiarlos hacia el correcto entendimiento y posibilidad de evolución mediante el trayecto avance.

Teniendo el conocimiento de *funciones y parámetros*, se realizarán los ejercicios propuestos en las actividades para incrementar de manera gradual los distintos saberes relacionados con *funciones*.

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

Actividad Integradora

Se planteará mediante la temática de juegos la creación de un personaje. Con el uso de un *prompt* se le pedirá al usuario que ingrese el nombre del personaje que se guardará en una variable. El personaje comenzará con una vida específica (un valor de tipo *number* por ejemplo 100).

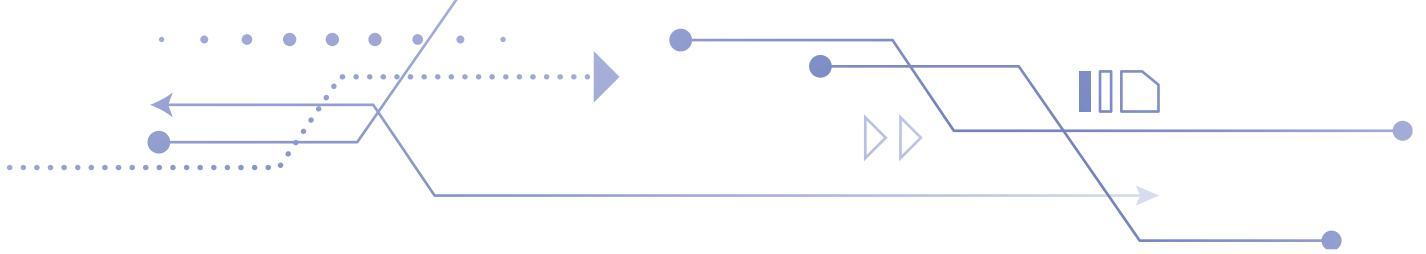
Se realizarán distintas preguntas planteadas por los/as estudiantes y dependiendo las respuestas, se irá perdiendo “vida” o acumulando valor.

La vida no puede pasar de 100 puntos ni ser menor que 0.

Se deberán utilizar varias funciones:

- Sumar Vida
- Restar Vida
- Pregunta Correcta





Semana 4

Objetivos

- Reforzar el conocimiento aprendido sobre *funciones*
- Comprender el uso del *keyword return*
- Almacenar datos de una *función*, en una variable
- Comenzar y utilizar el uso de parámetros y su importancia para generar código abstracto.
- Introducir el concepto de *hoisting*
- Conocer y comprender el método de *arrays* “*forEach*”
- Comprender el uso de *Funciones* anónimas

Ejes temáticos

- Refuerzo palabra *function*
- Refuerzo de abstracción de *funciones*
- Refuerzo de *scope*
- Introducción al *hoisting*
- *Keyword return*
- Patron de *early return*
- *Función* anónima
- *setInterval*, *setTimeout*
- *Función* como *callback* en métodos de *arrays*
- *DRY*

Encuentro Sincrónico

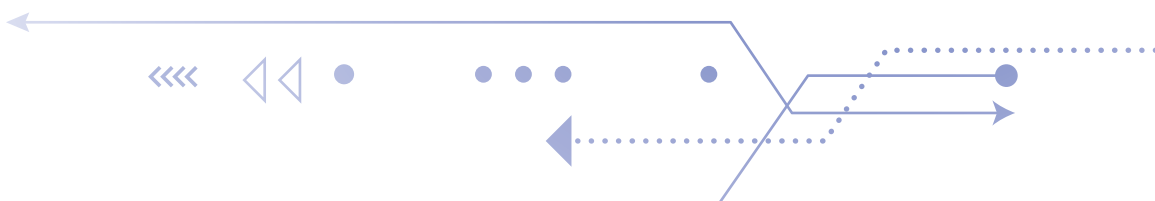
Este encuentro se divide en dos momentos.

Primer momento

Tomando cómo base el saber aprendido de *funciones*, ahora se introducirá la palabra *return* y *funciones* que retornan un valor específico. Se planterarán diversos ejercicios en donde quede explícita la importancia de tener *funciones* que devuelvan valores.

Se recomienda en este momento comenzar a mencionar ciertos patrones de desarrollo como el *early return*. Si bien no hace falta explicarlo cómo tal, sí mostrar los beneficios de poder correr un programa y, mediante la negación, cortar su ejecución si así fuese necesario para lograr programas más eficientes.

Se hablará de las formas de definir una *función*: tanto cómo *función* global así cómo *variable*, se hablará de los beneficios que trae cada modalidad y se introducirá el concepto de *hoisting* para aprenderlo por un lado, y evitar posibles errores en la declaración de variables cómo *funciones*.



Actividad 1

Números pares

Dados dos números, crear una *función* que devuelva true o false si el número es par. Una actividad similar se realizó durante el módulo anterior, sin hacer uso de las *funciones*, es por eso que este programa funcionará como *refactor* de lo creado anteriormente.

Actividad 2

Es divisible por

Dados dos números, crear una *función* que devuelva true o false si el primer número es divisible por el segundo. Si bien el programa es simple, queda en manifiesto el poder del *return* dentro de las *funciones*, y se aprovechará este ejercicio para realizar distintas validaciones utilizando métodos como *parseInt* o *parseFloat*.

En este segundo momento se explicarán algunas *funciones* globales propias del lenguaje de JS. Se enseñarán tanto *setInterval* como *setTimeout* y se comenzará a introducir el concepto de las *funciones* anónimas, sus beneficios y sus contrapartes.

Una vez aprendido el concepto de *funciones* y *funciones* anónimas, se dará lugar a la explicación del método *forEach* disponible en los *arrays*, y cómo una función anónima dentro de este método (*callback*) puede evitarnos la iteración aprendida hasta el momento mediante un uso *del for* o *while*.

Actividad 3

Hoy me siento...

En este programa se crearán distintos estados de ánimos como *strings* dentro de un *array*. Se utilizará la *función setInterval* para ejecutar de manera constante cómo se siente el/la estudiante hoy. Esta opción deberá cambiar cada un segundo, de manera aleatoria. Dependiendo del estado de ánimo que salga, una alerta (con el texto propuesto por la/el estudiante) deberá aparecer. Ejemplos pueden ser “Fuera de acá”, “Vení conmigo”, “No me molestes”, “No quiero estudiar”, “Viva JavaScript”, etc, etc, etc.

Encuentro asincrónico

Para este encuentro se plantearán las dudas y consultas que tengan sobre *funciones*, uso de *return*, *funciones* propias de JavaScript, y cualquier temática en relación a los módulos ya cursados.

Teniendo el conocimiento de *funciones* y *parámetros*, y agregando ahora el conocimiento de *returns*, se realizará una actividad integradora para incrementar de manera gradual los distintos saberes relacionados con *funciones*.

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

Actividad Integradora

Calculadora

Crear un programa que contenga cuatro *funciones* que reciban dos parámetros (los parámetros recibidos deben ser números y deberá estar validado). Cada *función* deberá devolver el valor de una suma, resta, multiplicación o división.

Crear el programa calculadora en donde el usuario ingrese por *prompt* dos valores, y qué operación desea realizar. Dependiendo de dicha operación el programa debe ejecutar una *función* u otra y almacenar el resultado.



Semana 5

Objetivos

- Aprender el uso de *objetos* como alternativa al *array*.
- Comprender la *referencia de objetos* y la importancia y problemas de modificar sobre referencias.
- Entender el concepto de *mutabilidad*.
- Utilizar *objetos* como estructura de datos y su similitud con *JSON*.
- Entender la manipulación y acceso al *objeto* mediante *dot* y *bracket notation*.

Ejes temáticos

- *Objetos*
- *Propiedades en objetos*
- *Strings, Numbers, Booleans, null, undefined en propiedades*
- *Métodos y Arrays en propiedades*
- *Dot notation*
- *Bracket notation*
- *Mutabilidad*
- *JSON*

Primer momento

Encuentro sincrónico

Este encuentro se divide en dos momentos.

En este primer momento se busca que los/as estudiantes adquieran los conocimientos de *objetos*, sus *métodos* y *propiedades*. A su vez se busca que logren tener un entendimiento de la iteración por las propiedades de *objetos* así como poder acceder a las propiedades y métodos mediante *dot* y *bracket notation*.

Junto con *objetos* aparece la aproximación al entendimiento de la *mutabilidad* y la importancia que esta tiene en *Js*.

Es importante en este módulo poder dar un entendimiento del *objeto* como modelo de almacenamiento de datos así pueden asociar este saber con el uso de *JSON* y este como resultado de invocación a una *API* que se verá en el módulo siguiente.



Actividad 1

Youtubers - Mi primer objeto

Las actividades propuestas de introducción a *objetos* tomarán el papel de reforzar el vínculo y conocimiento de los/as estudiantes, poniéndolos cómo protagonistas del programa. A su vez se recomienda hacer mucho hincapié en la sintaxis de los *objetos*, el uso de comas (,) como separadores de propiedades.

Se realizará un *objeto* con el *youtuber* favorito de los/as estudiantes y se les pedirá que se agreguen cómo mínimo cuatro propiedades: Nombre, *Url* al canal de *Youtube*, cantidad de seguidores, temática de su canal. Una vez creado el *objeto*, deberán imprimir por consola todas las propiedades.

Actividad 2

Mis intereses

Este programa consiste en armar un *objeto* que modele quienes son las/os estudiantes. El *objeto* podrá contener propiedades como

- Nombre
- ¿Tenés mascota?
- Apellido
- Si tenés, cual es el nombre?
- Edad
- Comida preferida
- Juego favorito

Luego, se hace un *console.log* por cada propiedad, y las y los estudiantes comparten pantalla para mostrar qué es lo que contiene el *objeto*, y de esta forma seguir conociéndose más.

Segundo momento

En este segundo momento se sigue profundizando sobre las características de los *objetos*. Se propone comenzar a pensar en el *objeto* como tipo de dato para almacenar estructuras complejas. Se proponen actividades a modo juego y se habla de las buenas y malas prácticas a la hora del armado de propiedades:



Solo para que sepan, pero no usen

Actividad 3

Adivinanzas

En este programa se arma un juego con intereses de los/as estudiantes: estos deben almacenarse en *objetos*.

Una vez creado los *objetos* con intereses el/a mentor/a deberá recopilar los trabajos y armar un programa (paso a paso, que los estudiantes vean la aplicación) que muestre mediante un *alert* un interés, y acto seguido un *prompt* preguntando a qué estudiante corresponde ese interés, cualquier estudiante puede responder y así ganar puntos.

Con esta actividad de dinámica juego, se sigue fomentando el juego y conocimiento en este espacio.

Encuentro asincrónico

Para este encuentro se espera que los/as estudiantes manifiesten sus dudas, consultas y comentarios sobre todo lo que están aprendiendo. A su vez se planteará un ejercicio integrador de *objetos* como actividad entregable.

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

Actividad Integradora

Comentarios de youtube

Este programa busca introducir el concepto de *objetos* con *objetos* dentro, abordando la temática de comentarios en youtube (por poner un ejemplo de una plataforma específica, este ejemplo funciona para cualquier plataforma que tenga comentarios).

El programa consiste en armar un *objeto* con una propiedad nombre = "video" y otra propiedad de tipo comentarios. La propiedad de comentarios será un *objeto* que contiene título, fecha y hora, contenido y respuesta. Respuesta contendrá las mismas propiedades que comentario. El *objeto* deberá ser similar a este:

Además de armar el *objeto*, se propondrá que los/as estudiantes puedan acceder a las distintas propiedades del *objeto* haciendo un `console.log` de cada una.

```
const video = {  
  titulo: "Nombre del video",  
  comentarios: {  
    titulo: "Título del comentario",  
    autor: "Autor del comentario",  
    fecha: "Fecha del comentario",  
    respuesta: {  
      titulo: "Título del comentario",  
      autor: "Autor del comentario",  
      fecha: "Fecha del comentario"  
    }  
  }  
}
```



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 de las actividades planteadas y entiendan la diversidad de soluciones a un mismo problema.

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

Entendiendo la primera aproximación a un lenguaje de programación puede resultar frustrante, en este encuentro se armará un mural digital en donde se podrán compartir las experiencias, miedos, inquietudes, cambios y cualquier temática que esté viviendo El/a estudiante en este momento.

Se recomienda en este mural que los/as mentores/as puedan expresar también su punto de vista, contar experiencias pasadas y generar un vínculo de pares con las/os estudiantes.

Este encuentro es un buen punto de reflexión para que el/la 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.

Actividad Mural digital

Se creará mediante un *Padlet* una serie de *posits* disponibles en un mural digital. Se plantean una serie de preguntas y cada estudiante deberá escribir su opinión que quede como registro de lo vivido en este módulo. **Ejemplos de preguntas pueden ser:**

¿Qué es lo que más me gustó?

¿Qué es lo que menos me gustó?

¿Hacia dónde quiero ir ahora que sé JavaScript?

¿Qué cambiaría de todo lo vivido en este trayecto?



Se recomienda que los/as mentores/as formulen preguntas que tengan cierta complicitad con todo lo que ha pasado en la comisión a lo largo de estos módulos. Aprovechando que aún queda un módulo más, parte de esta información podrá ser utilizada para generar cambios en lo que resta del trayecto.

