

## **CLASES**

Los horarios de las clases, si no se estipula ningún cambio, serán:

<b>DÍA</b>	<b>HORARIO</b>
Lunes 22 septiembre	16:30 – 19:30
Miércoles 24 septiembre	16:30 – 19:30
Lunes 29 octubre	16:30 – 19:30
Miércoles 1 octubre	16:30 – 19:30
Lunes 6 octubre	16:30 – 19:30
Miércoles 8 octubre	16:30 – 19:30
Lunes 13 octubre	16:30 – 18:30 EXAMEN ESCRITO
Miércoles 15 octubre	16:30 – 18:30
Lunes 20 octubre	16:30 – 18:30

## **TELEFORMACIÓN**

Los días para la teleformación serán (la teleformación será habilitar una reunión y el que se meta será como una tutoría para dudas de cualquier tipo). **Sólo asistiré si se me preavisa de que irá alguien:**

- 16h de teleformación presencial
- 8h de teleformación revisión de proyectos

<b>DÍA</b>	<b>HORARIO</b>
Viernes 26 septiembre	16:00 – 19:00
Viernes 3 octubre	16:30 – 19:00
Viernes 10 octubre	16:30 – 19:00
Viernes 17 octubre	16:30 – 19:00
Miércoles 22 octubre	17:00 – 20:00
Viernes 24 octubre	16:30 – 19:00

## **SUPERVISIÓN DE PROYECTOS**

Este apartado es para explicar que me podréis escribir y que durante el día os intentaré dar una respuesta. Podéis escribir siempre en general, y os intentaré responder *asap*.

## **CRITERIOS DE EVALUACIÓN GENERALES**

- **En los ejercicios:**

- Cada módulo tiene una serie de ejercicios, y la evaluación contendrá algunos nuevos. Respecto al resto, NO se envían para revisión, pero se pueden preguntar en las tutorías y en clase.
- Las tareas tienen que estar dentro del tiempo asignado, si no, el primer día se quitan 2 puntos, el siguiente otros 2 y el tercer día de retraso automáticamente son un 0 (no apto) y **no se revisarán**.
- Se evaluará de 0 a 10, siendo un 6 apto.
- El seis implica que programa hace lo que tiene que hacer -> apto
- Los siguientes puntos revisarán hasta el 10:
  - Que no esté copiado
  - La belleza del programa
  - Los estándares de programación tipo lint
  - La eficiencia
  - Smell code, criterios de cleaning code
  - Testing en el caso de ser requerido
  - Aportes extras (nuevas ideas...)
- **En el examen escrito:**
  - La claridad y la precisión de las contestaciones.
  - La ortografía.
  - La profundidad de los argumentos
  - La eficiencia, los diseños y el estilo de las soluciones (diagramas, dibujos, borradores, código...)
  - Nomenclaturas de las variables.
- **En el proyecto:**
  - Los criterios se ponen en el apartado final.

## **FIN DE ENTREGAS**

- **TAREAS -> 27 Octubre 2025**
- **PROYECTO -> 8 Noviembre 2025**

## **EVALUACIÓN CORE-JAVASCRIPT**

La siguiente sección servirá para plasmar la evaluación del módulo de core-javascript, teniendo en cuenta que una parte importante del mismo se basará en el proyecto final, cuya evaluación es conjunta con el módulo de HTML.

- **Se recuerda que únicamente se dará el apto si se supera un 6.**
- **En caso de no llegar al 6, se decidirá una solución conjunta con la dirección del curso, teniendo en cuenta la solvencia y la actitud del estudiante y los puntos más vulnerables de su evaluación que le han hecho suspender.**
- **Cualquier sección suspendida implica un suspenso en el módulo, a excepción de los ejercicios entregables y el examen escrito.**
  - **En el caso de los ejercicios entregables, la nota media sí debe dar un apto (encima de 6).**

- Una nota inferior a 4 puntos en el examen escrito implica suspenso.

SECCIÓN	PUNTUACIÓN SOBRE EL TOTAL (%)
Notas del profesor: <ul style="list-style-type: none"> <li>- Calidad de las preguntas y explicaciones, respuestas en el día a día.</li> <li>- Ejercicios durante la teoría.</li> <li>- Actitud en clase con el profesor, con la clase, con los compañer@s</li> </ul>	10
Ejercicio entregable 1	10
Ejercicio entregable 2	15
Ejercicio entregable 3	15
Examen escrito	20
Proyecto Final	30

## **CRITERIOS DE EVALUACIÓN PROYECTO (conjunto con HTML)**

### **1. JAVASCRIPT (CORE) – 25 puntos**

<b>Criterio</b>	<b>Descripción</b>	<b>Puntos</b>
Lógica y funcionamiento del juego	El juego implementa una lógica clara, es funcional y sin errores importantes.	5
Estructura del código	Uso correcto de funciones, control de flujo, eventos, modularidad.	5
Buenas prácticas	Uso correcto de nombres de variables, legibilidad, comentarios cuando son necesarios. Código limpio.	5
Eficiencia y rendimiento	Código optimizado y sin repeticiones innecesarias.	5
Uso de eventos	Eventos bien implementados y correctamente gestionados.	5

---

### **2. GITHUB E INTEGRACIÓN – 15 puntos**


<b>Criterio</b>	<b>Descripción</b>	<b>Puntos</b>
Repositorio organizado	Carpetas y archivos bien estructurados, posible README.	5
Uso de Git	Commits frecuentes, mensajes descriptivos, historial coherente.	4
Integración en el proyecto común	El juego se integra correctamente en la página web compartida.	6

---

### **3. CREATIVIDAD Y ORIGINALIDAD – 10 puntos**

<b>Criterio</b>	<b>Descripción</b>	<b>Puntos</b>
Originalidad del juego	El juego tiene una idea novedosa, mecánica interesante o propuesta única.	5
Diseño visual y experiencia de usuario	Interfaz cuidada, experiencia de uso clara, atractiva y divertida.	5

---

 **TOTAL: 50 PUNTOS**