

# Máster Título Propio

## Programación de Videojuegos

```
modifier_...  
_ob)) # modifier  
d_objects[0]  
ame].select = 1  
select exactly two objects  
ATOR CLASSES
```

```
...Operator):  
...mirror to the selected object""  
...mirror_mirror_x"  
...x"
```



## Máster Título Propio Programación de Videojuegos

- » Modalidad: **online**
- » Duración: **12 meses**
- » Titulación: **TECH Global University**
- » Acreditación: **60 ECTS**
- » Horario: **a tu ritmo**
- » Exámenes: **online**

Acceso web: [www.techtitute.com/videojuegos/master/master-programacion-videojuegos](http://www.techtitute.com/videojuegos/master/master-programacion-videojuegos)

# Índice

01

Presentación del programa

---

*pág. 4*

02

¿Por qué estudiar en TECH?

---

*pág. 8*

03

Plan de estudios

---

*pág. 12*

04

Objetivos docentes

---

*pág. 28*

05

Salidas profesionales

---

*pág. 34*

06

Licencias de software incluidas

---

*pág. 38*

07

Metodología de estudio

---

*pág. 42*

08

Titulación

---

*pág. 52*

# 01

# Presentación del programa

De entre las tareas más importantes y delicadas a la hora de llevar a cabo el proyecto de un Videojuego se encuentra la Programación. En este sentido, un estudio reciente de la Asociación Internacional de Desarrolladores de Juegos revela que más del 75% de los desarrolladores de Videojuegos consideran la Programación como el aspecto más crítico para el éxito de estos. Frente a esta realidad, los programadores tienen la responsabilidad de perfeccionar sus habilidades para crear experiencias de juego más inmersivas y funcionales. Con el objetivo de facilitarles esta labor, TECH presenta un novedoso programa universitario orientado en los conocimientos más avanzados en programación de Videojuegos. Asimismo, se imparte en una modalidad flexible y totalmente online.



“

*Gracias a este Máster Título Propio 100% online, dominarás las herramientas más modernas para el desarrollo de Videojuegos y crearás experiencias interactivas de alta calidad”*

La Programación de Videojuegos ha experimentado notables avances en los últimos años, consolidándose como un pilar esencial dentro del desarrollo de experiencias digitales cada vez más complejas y envolventes. Por ejemplo, la creciente demanda de Videojuegos innovadores ha impulsado a las empresas del sector a acelerar sus procesos de creación sin comprometer la calidad ni la originalidad de los contenidos. En este sentido, los programadores deben contar con una capacitación altamente especializada que les permita dar respuesta a las exigencias técnicas y creativas actuales, garantizando el correcto funcionamiento del código y el éxito de cada proyecto interactivo.

En este contexto, TECH lanza un vanguardista programa en Programación de Videojuegos. Diseñado por referentes en este ámbito, el itinerario académico profundizará en los fundamentos clave de los algoritmos y su estructura lógica. En sintonía con esto, el temario ahondará en los elementos esenciales que componen un programa informático, desde la declaración de variables hasta los bucles de control. Asimismo, los materiales didácticos abordarán de forma práctica el uso del *Visual Scripting* en *Unreal Engine 4*, permitiendo a los profesionales crear dinámicas complejas sin necesidad de escribir código tradicional. De este modo, los egresados desarrollarán competencias avanzadas para diseñar sistemas interactivos de alto rendimiento. Gracias a esto, potenciarán la calidad técnica y creativa de sus proyectos digitales.

Desde otro lado, la titulación universitaria se basa en una práctica modalidad 100% online que permite a los programadores organizar individualmente sus tiempos de estudio. Por ello, lo único que necesitarán es un dispositivo electrónico con conexión a internet para acceder al Campus Virtual. Además, TECH aplica su innovador sistema del *Relearning*, que garantiza que los profesionales consoliden sus conocimientos de manera natural y escalonada. Así pues, los egresados no tendrán que dedicar extensas jornadas al repaso o emplear métodos tradicionales como la memorización. Adicionalmente, contarán con una variada selección de recursos multimedia como videotutoriales, simulaciones interactivas o lecturas especializadas.

Este **Máster Título Propio en Programación de Videojuegos** contiene el programa universitario más completo y actualizado del mercado. Sus características más destacadas son:

- ♦ El desarrollo de casos prácticos presentados por expertos en Videojuegos
- ♦ Los contenidos gráficos, esquemáticos y eminentemente prácticos con los que están concebidos recogen una información científica y práctica sobre aquellas disciplinas indispensables para el ejercicio profesional
- ♦ Los ejercicios prácticos donde realizar el proceso de autoevaluación para mejorar el aprendizaje
- ♦ Su especial hincapié en metodologías innovadoras en la Programación de Videojuegos
- ♦ Las lecciones teóricas, preguntas al experto, foros de discusión de temas controvertidos y trabajos de reflexión individual
- ♦ La disponibilidad de acceso a los contenidos desde cualquier dispositivo fijo o portátil con conexión a internet



*Optimizarás el rendimiento del software para maximizar la eficiencia gráfica y de procesamiento”*

“

*Mediante el disruptivo sistema Relearning de TECH, te capacitarás con menos esfuerzo y más rendimiento, implicándote más en tu especialización como programador de Videojuegos”*

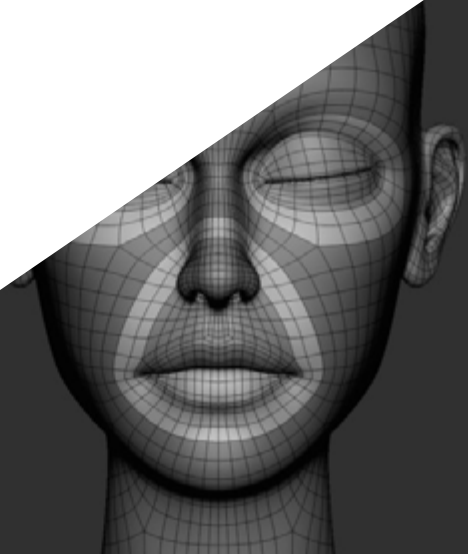
Incluye en su cuadro docente a profesionales pertenecientes al ámbito de Videojuegos, que vierten en este programa la experiencia de su trabajo, además de reconocidos especialistas de sociedades de referencia y universidades de prestigio.

Su contenido multimedia, elaborado con la última tecnología educativa, permitirá al profesional un aprendizaje situado y contextual, es decir, un entorno simulado que proporcionará un estudio inmersivo programado para entrenarse ante situaciones reales.

El diseño de este programa se centra en el Aprendizaje Basado en Problemas, mediante el cual el alumno deberá tratar de resolver las distintas situaciones de práctica profesional que se le planteen a lo largo del curso académico. Para ello, el profesional contará con la ayuda de un novedoso sistema de vídeo interactivo realizado por reconocidos expertos.

*Realizarás pruebas de software para asegurar la calidad técnica de las experiencias digitales.*

*Gestionarás proyectos de Programación aplicando metodologías ágiles como Scrum o Kanban.*



02

# ¿Por qué estudiar en TECH?

TECH es la mayor Universidad digital del mundo. Con un impresionante catálogo de más de 14.000 programas universitarios, disponibles en 11 idiomas, se posiciona como líder en empleabilidad, con una tasa de inserción laboral del 99%. Además, cuenta con un enorme claustro de más de 6.000 profesores de máximo prestigio internacional.





“

*Estudia en la mayor universidad digital del mundo y asegura tu éxito profesional. El futuro empieza en TECH”*

### La mejor universidad online del mundo según FORBES

La prestigiosa revista Forbes, especializada en negocios y finanzas, ha destacado a TECH como «la mejor universidad online del mundo». Así lo han hecho constar recientemente en un artículo de su edición digital en el que se hacen eco del caso de éxito de esta institución, «gracias a la oferta académica que ofrece, la selección de su personal docente, y un método de aprendizaje innovador orientado a formar a los profesionales del futuro».

**Forbes**  
Mejor universidad  
online del mundo

**Plan**  
de estudios  
más completo

### Los planes de estudio más completos del panorama universitario

TECH ofrece los planes de estudio más completos del panorama universitario, con temarios que abarcan conceptos fundamentales y, al mismo tiempo, los principales avances científicos en sus áreas científicas específicas. Asimismo, estos programas son actualizados continuamente para garantizar al alumnado la vanguardia académica y las competencias profesionales más demandadas. De esta forma, los títulos de la universidad proporcionan a sus egresados una significativa ventaja para impulsar sus carreras hacia el éxito.

### El mejor claustro docente top internacional

El claustro docente de TECH está integrado por más de 6.000 profesores de máximo prestigio internacional. Catedráticos, investigadores y altos ejecutivos de multinacionales, entre los cuales se destacan Isaiah Covington, entrenador de rendimiento de los Boston Celtics; Magda Romanska, investigadora principal de MetaLAB de Harvard; Ignacio Wistumba, presidente del departamento de patología molecular traslacional del MD Anderson Cancer Center; o D.W Pine, director creativo de la revista TIME, entre otros.

Profesorado  
**TOP**  
Internacional



La metodología  
más eficaz

### Un método de aprendizaje único

TECH es la primera universidad que emplea el *Relearning* en todas sus titulaciones. Se trata de la mejor metodología de aprendizaje online, acreditada con certificaciones internacionales de calidad docente, dispuestas por agencias educativas de prestigio. Además, este disruptivo modelo académico se complementa con el "Método del Caso", configurando así una estrategia de docencia online única. También en ella se implementan recursos didácticos innovadores entre los que destacan vídeos en detalle, infografías y resúmenes interactivos.

### La mayor universidad digital del mundo

TECH es la mayor universidad digital del mundo. Somos la mayor institución educativa, con el mejor y más amplio catálogo educativo digital, cien por cien online y abarcando la gran mayoría de áreas de conocimiento. Ofrecemos el mayor número de titulaciones propias, titulaciones oficiales de posgrado y de grado universitario del mundo. En total, más de 14.000 títulos universitarios, en once idiomas distintos, que nos convierten en la mayor institución educativa del mundo.

**nº1**  
Mundial  
Mayor universidad  
online del mundo

### La universidad online oficial de la NBA

TECH es la universidad online oficial de la NBA. Gracias a un acuerdo con la mayor liga de baloncesto, ofrece a sus alumnos programas universitarios exclusivos, así como una gran variedad de recursos educativos centrados en el negocio de la liga y otras áreas de la industria del deporte. Cada programa tiene un currículo de diseño único y cuenta con oradores invitados de excepción: profesionales con una distinguida trayectoria deportiva que ofrecerán su experiencia en los temas más relevantes.

### Líderes en empleabilidad

TECH ha conseguido convertirse en la universidad líder en empleabilidad. El 99% de sus alumnos obtienen trabajo en el campo académico que ha estudiado, antes de completar un año luego de finalizar cualquiera de los programas de la universidad. Una cifra similar consigue mejorar su carrera profesional de forma inmediata. Todo ello gracias a una metodología de estudio que basa su eficacia en la adquisición de competencias prácticas, totalmente necesarias para el desarrollo profesional.



### Google Partner Premier

El gigante tecnológico norteamericano ha otorgado a TECH la insignia Google Partner Premier. Este galardón, solo al alcance del 3% de las empresas del mundo, pone en valor la experiencia eficaz, flexible y adaptada que esta universidad proporciona al alumno. El reconocimiento no solo acredita el máximo rigor, rendimiento e inversión en las infraestructuras digitales de TECH, sino que también sitúa a esta universidad como una de las compañías tecnológicas más punteras del mundo.



### La universidad mejor valorada por sus alumnos

Los alumnos han posicionado a TECH como la universidad mejor valorada del mundo en los principales portales de opinión, destacando su calificación más alta de 4,9 sobre 5, obtenida a partir de más de 1.000 reseñas. Estos resultados consolidan a TECH como la institución universitaria de referencia a nivel internacional, reflejando la excelencia y el impacto positivo de su modelo educativo.



# 03

## Plan de estudios

Este Máster Título Propio en Programación de Videojuegos brindará a los programadores las herramientas más innovadoras para dominar el desarrollo de experiencias interactivas. Es así como, el plan de estudios profundizará en los fundamentos esenciales del ciclo de vida de una aplicación informática, así como en la calidad del software, asegurando una experiencia óptima para el usuario. Asimismo, el temario preparará a los egresados para gestionar la integración de las tecnologías de la información y la comunicación en la creación de Videojuegos avanzados, optimizando su rendimiento y jugabilidad.





“

*Integrarás motores de Videojuegos comerciales y desarrollarás motores propios adaptados a diferentes necesidades”*

## Módulo 1. Fundamentos de Programación

- 1.1. Introducción a la Programación
  - 1.1.1. Estructura básica de un ordenador
  - 1.1.2. Software
  - 1.1.3. Lenguajes de Programación
  - 1.1.4. Ciclo de vida de una aplicación informática
- 1.2. Diseño de algoritmos
  - 1.2.1. La resolución de problemas
  - 1.2.2. Técnicas descriptivas
  - 1.2.3. Elementos y estructura de un algoritmo
- 1.3. Elementos de un programa
  - 1.3.1. Origen y características del lenguaje C++
  - 1.3.2. El entorno de desarrollo
  - 1.3.3. Concepto de programa
  - 1.3.4. Tipos de datos fundamentales
  - 1.3.5. Operadores
  - 1.3.6. Expresiones
  - 1.3.7. Sentencias
  - 1.3.8. Entrada y salida de datos
- 1.4. Sentencias de control
  - 1.4.1. Sentencias
  - 1.4.2. Bifurcaciones
  - 1.4.3. Bucles
- 1.5. Abstracción y modularidad: Funciones
  - 1.5.1. Diseño modular
  - 1.5.2. Concepto de función y utilidad
  - 1.5.3. Definición de una función
  - 1.5.4. Flujo de ejecución en la llamada de una función
  - 1.5.5. Prototipo de una función
  - 1.5.6. Devolución de resultados
  - 1.5.7. Llamada a una función: Parámetros
  - 1.5.8. Paso de parámetros por referencia y por valor
  - 1.5.9. Ámbito identificador
- 1.6. Estructuras de datos estáticas
  - 1.6.1. Arrays
  - 1.6.2. Matrices. Poliedros
  - 1.6.3. Búsqueda y ordenación
  - 1.6.4. Cadenas. Funciones de E/S para cadenas
  - 1.6.5. Estructuras. Uniones
  - 1.6.6. Nuevos tipos de datos
- 1.7. Estructuras de datos dinámicas: Punteros
  - 1.7.1. Concepto. Definición de puntero
  - 1.7.2. Operadores y operaciones con punteros
  - 1.7.3. Arrays de punteros
  - 1.7.4. Punteros y arrays
  - 1.7.5. Punteros a cadenas
  - 1.7.6. Punteros a estructuras
  - 1.7.7. Indirección múltiple
  - 1.7.8. Punteros a funciones
  - 1.7.9. Paso de funciones, estructuras y arrays como parámetros de funciones
- 1.8. Ficheros
  - 1.8.1. Conceptos básicos
  - 1.8.2. Operaciones con ficheros
  - 1.8.3. Tipos de ficheros
  - 1.8.4. Organización de los ficheros
  - 1.8.5. Introducción a los ficheros C++
  - 1.8.6. Manejo de ficheros
- 1.9. Recursividad
  - 1.9.1. Definición de recursividad
  - 1.9.2. Tipos de recursión
  - 1.9.3. Ventajas e inconvenientes
  - 1.9.4. Consideraciones
  - 1.9.5. Conversión recursivo-iterativa
  - 1.9.6. La pila de recursión

- 1.10. Prueba y documentación
  - 1.10.1. Pruebas de programas
  - 1.10.2. Prueba de la caja blanca
  - 1.10.3. Prueba de la caja negra
  - 1.10.4. Herramientas para realizar las pruebas
  - 1.10.5. Documentación de programas

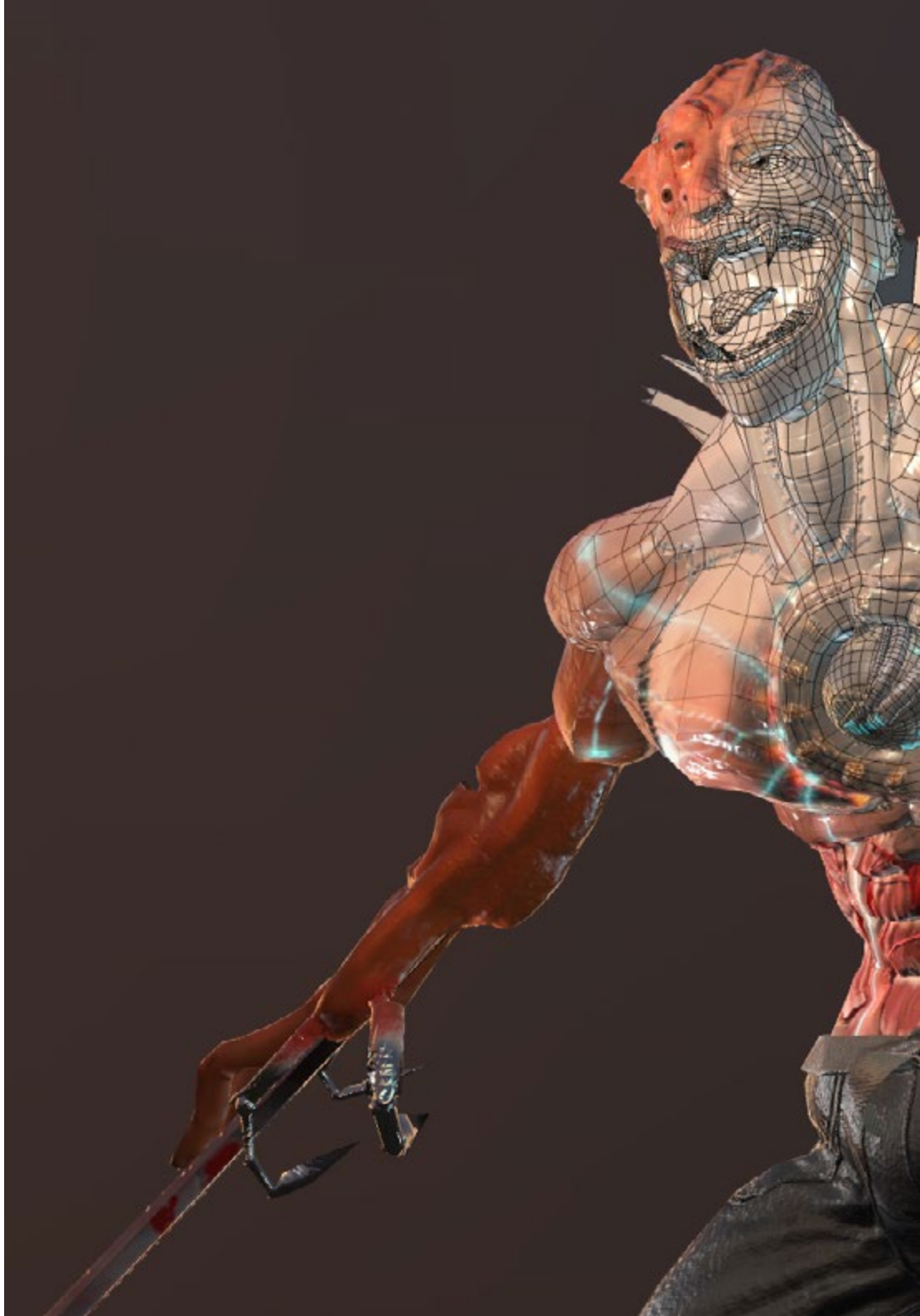
## Módulo 2. Estructura de Datos y Algoritmos

- 2.1. Introducción a las estrategias de diseño de algoritmos
  - 2.1.1. Recursividad
  - 2.1.2. Divide y conquista
  - 2.1.3. Otras estrategias
- 2.2. Eficiencia y análisis de los algoritmos
  - 2.2.1. Medidas de eficiencia
  - 2.2.2. Medir el tamaño de la entrada
  - 2.2.3. Medir el tiempo de ejecución
  - 2.2.4. Caso peor, mejor y medio
  - 2.2.5. Notación asintótica
  - 2.2.6. Criterios de análisis matemático de algoritmos no recursivos
  - 2.2.7. Análisis matemático de algoritmos recursivos
  - 2.2.8. Análisis empírico de algoritmos
- 2.3. Algoritmos de ordenación
  - 2.3.1. Concepto de ordenación
  - 2.3.2. Ordenación de la burbuja
  - 2.3.3. Ordenación por selección
  - 2.3.4. Ordenación por inserción
  - 2.3.5. Ordenación por mezcla (Merge\_Sort)
  - 2.3.6. Ordenación rápida (Quick\_Sort)
- 2.4. Algoritmos con árboles
  - 2.4.1. Concepto de árbol
  - 2.4.2. Árboles binarios
  - 2.4.3. Recorridos de árbol
  - 2.4.4. Representar expresiones
  - 2.4.5. Árboles binarios ordenados
  - 2.4.6. Árboles binarios balanceados
- 2.5. Algoritmos con Heaps
  - 2.5.1. Los Heaps
  - 2.5.2. El algoritmo Heapsort
  - 2.5.3. Las colas de prioridad
- 2.6. Algoritmos con Grafos
  - 2.6.1. Representación
  - 2.6.2. Recorrido en anchura
  - 2.6.3. Recorrido en profundidad
  - 2.6.4. Ordenación topológica
- 2.7. Algoritmos Greedy
  - 2.7.1. La estrategia Greedy
  - 2.7.2. Elementos de la estrategia Greedy
  - 2.7.3. Cambio de monedas
  - 2.7.4. Problema del viajante
  - 2.7.5. Problema de la mochila
- 2.8. Búsqueda de caminos mínimos
  - 2.8.1. El problema del camino mínimo
  - 2.8.2. Arcos negativos y ciclos
  - 2.8.3. Algoritmo de Dijkstra

- 2.9. Algoritmos Greedy sobre grafos
  - 2.9.1. El árbol de recubrimiento mínimo
  - 2.9.2. El algoritmo de Prim
  - 2.9.3. El algoritmo de Kruskal
  - 2.9.4. Análisis de complejidad
- 2.10. Backtracking
  - 2.10.1. El backtracking
  - 2.10.2. Técnicas alternativas

### Módulo 3. Programación Orientada a Objetos

- 3.1. Introducción a la Programación orientada a objetos
  - 3.1.1. Introducción a la Programación orientada a objetos
  - 3.1.2. Diseño de clases
  - 3.1.3. Introducción a UML para el modelado de los problemas
- 3.2. Relaciones entre clases
  - 3.2.1. Abstracción y herencia
  - 3.2.2. Conceptos avanzados de herencia
  - 3.2.3. Polimorfismo
  - 3.2.4. Composición y agregación
- 3.3. Introducción a los patrones de diseño para problemas orientados a objetos
  - 3.3.1. ¿Qué son los patrones de diseño?
  - 3.3.2. Patrón *factory*
  - 3.3.3. Patrón *singleton*
  - 3.3.4. Patrón *observer*
  - 3.3.5. Patrón *composite*
- 3.4. Excepciones
  - 3.4.1. ¿Qué son las excepciones?
  - 3.4.2. Captura y gestión de excepciones
  - 3.4.3. Lanzamiento de excepciones
  - 3.4.4. Creación de excepciones







- 3.5. Interfaces de usuario
  - 3.5.1. Introducción a Qt
  - 3.5.2. Posicionamiento
  - 3.5.3. ¿Qué son los eventos?
  - 3.5.4. Eventos: Definición y captura
  - 3.5.5. Desarrollo de interfaces de usuario
- 3.6. Introducción a la Programación concurrente
  - 3.6.1. Introducción a la Programación concurrente
  - 3.6.2. El concepto de proceso e hilo
  - 3.6.3. Interacción entre procesos o hilos
  - 3.6.4. Los hilos en C++
  - 3.6.5. Ventajas e Inconvenientes de la Programación concurrente
- 3.7. Gestión de hilos y sincronización
  - 3.7.1. Ciclo de vida de un hilo
  - 3.7.2. La clase *thread*
  - 3.7.3. Planificación de hilos
  - 3.7.4. Grupos de hilos
  - 3.7.5. Hilos de tipo demonio
  - 3.7.6. Sincronización
  - 3.7.7. Mecanismos de bloqueo
  - 3.7.8. Mecanismos de comunicación
  - 3.7.9. Monitores
- 3.8. Problemas comunes dentro de la Programación concurrente
  - 3.8.1. El problema de los productores y consumidores
  - 3.8.2. El problema de los lectores y escritores
  - 3.8.3. El problema de la cena de los filósofos
- 3.9. Documentación y pruebas de software
  - 3.9.1. ¿Por qué es importante documentar el software?
  - 3.9.2. Documentación de diseño
  - 3.9.3. Uso de Herramientas para la documentación

- 3.10. Pruebas de software
  - 3.10.1. Introducción a las pruebas del software
  - 3.10.2. Tipos de pruebas
  - 3.10.3. Prueba de unidad
  - 3.10.4. Prueba de integración
  - 3.10.5. Prueba de validación
  - 3.10.6. Prueba del sistema

## Módulo 4. Consolas y Dispositivos para Videojuegos

- 4.1. Historia de la Programación en Videojuegos
  - 4.1.1. Período Atari (1977 - 1985)
  - 4.1.2. Período NES y SNES (1985 - 1995)
  - 4.1.3. Período PlayStation / PlayStation 2 (1995 - 2005)
  - 4.1.4. Período Xbox 360, PS3 y Wii (2005 - 2013)
  - 4.1.5. Período Xbox One, PS4 y Wii U – Switch (2013 - Actualidad)
  - 4.1.6. El futuro
- 4.2. Historia de la jugabilidad en Videojuegos
  - 4.2.1. Introducción
  - 4.2.2. Contexto social
  - 4.2.3. Diagrama estructural
  - 4.2.4. Futuro
- 4.3. Adaptación a los tiempos modernos
  - 4.3.1. Juegos basados en movimiento
  - 4.3.2. Realidad virtual
  - 4.3.3. Realidad aumentada
  - 4.3.4. Realidad mixta
- 4.4. Unity: *Scripting* I y ejemplos
  - 4.4.1. ¿Qué es un *script*?
  - 4.4.2. Nuestro primer *script*
  - 4.4.3. Añadiendo un *script*
  - 4.4.4. Abriendo un *script*
  - 4.4.5. *MonoBehaviour*
  - 4.4.6. *Debugging*
- 4.5. Unity: *Scripting* II y ejemplos
  - 4.5.1. Entrada de teclado y ratón
  - 4.5.2. *Raycast*
  - 4.5.3. Instanciación
  - 4.5.4. Variables
  - 4.5.5. Variables públicas y serializadas
- 4.6. Unity: *Scripting* III y ejemplos
  - 4.6.1. Obteniendo componentes
  - 4.6.2. Modificando componentes
  - 4.6.3. Testeo
  - 4.6.4. Múltiples objetos
  - 4.6.5. *Colliders* y *triggers*
  - 4.6.6. *Cuaterniones*
- 4.7. Periféricos
  - 4.7.1. Evolución y clasificación
  - 4.7.2. Periféricos e interfaces
  - 4.7.3. Periféricos actuales
  - 4.7.4. Futuro próximo
- 4.8. Videojuegos: Perspectivas futuras
  - 4.8.1. Juego basado en la nube
  - 4.8.2. Ausencia de controladores
  - 4.8.3. Realidad inmersiva
  - 4.8.4. Otras alternativas
- 4.9. Arquitectura
  - 4.9.1. Necesidades especiales de los Videojuegos
  - 4.9.2. Evolución de la arquitectura
  - 4.9.3. Arquitectura actual
  - 4.9.4. Diferencias entre arquitecturas
- 4.10. *Kits* de desarrollo y su evolución
  - 4.10.1. Introducción
  - 4.10.2. Tercera generación de *kits* de desarrollo
  - 4.10.3. Cuarta generación de *kits* de desarrollo
  - 4.10.4. Quinta generación de *kits* de desarrollo
  - 4.10.5. Sexta generación de *kits* de desarrollo

## Módulo 5. Ingeniería de Software

- 5.1. Introducción a la ingeniería del software y al modelado
  - 5.1.1. La naturaleza del software
  - 5.1.2. La naturaleza única de las *webApps*
  - 5.1.3. Ingeniería del software
  - 5.1.4. El proceso del software
  - 5.1.5. La práctica de la ingeniería del software
  - 5.1.6. Mitos del software
  - 5.1.7. Cómo comienza todo
  - 5.1.8. Conceptos orientados a objetos
  - 5.1.9. Introducción a UML
- 5.2. El proceso del software
  - 5.2.1. Un modelo general de proceso
  - 5.2.2. Modelos de proceso prescriptivos
  - 5.2.3. Modelos de proceso Especializado
  - 5.2.4. El proceso unificado
  - 5.2.5. Modelos del proceso personal y del equipo
  - 5.2.6. ¿Qué es la agilidad?
  - 5.2.7. ¿Qué es un proceso ágil?
  - 5.2.8. *Scrum*
  - 5.2.9. Conjunto de herramientas para el proceso ágil
- 5.3. Principios que guían la práctica de la ingeniería del software
  - 5.3.1. Principios que guían el proceso
  - 5.3.2. Principios que guían la práctica
  - 5.3.3. Principios de comunicación
  - 5.3.4. Principios de planificación
  - 5.3.5. Principios de modelado
  - 5.3.6. Principios de construcción
  - 5.3.7. Principios de despliegue
- 5.4. Comprensión de los requisitos
  - 5.4.1. Ingeniería de requisitos
  - 5.4.2. Establecer las bases
  - 5.4.3. Indagación de los requisitos
  - 5.4.4. Desarrollo de casos de uso
  - 5.4.5. Elaboración del modelo de los requisitos
  - 5.4.6. Negociación de los requisitos
  - 5.4.7. Validación de los requisitos
- 5.5. Modelado de los requisitos: Escenarios, información y clases de análisis
  - 5.5.1. Análisis de los requisitos
  - 5.5.2. Modelado basado en escenarios
  - 5.5.3. Modelos UML que proporcionan el caso de uso
  - 5.5.4. Conceptos de modelado de datos
  - 5.5.5. Modelado basado en clases
  - 5.5.6. Diagramas de clases
- 5.6. Modelado de los requisitos: Flujo, comportamiento y patrones
  - 5.6.1. Requisitos que modelan las estrategias
  - 5.6.2. Modelado orientado al flujo
  - 5.6.3. Diagramas de estado
  - 5.6.4. Creación de un modelo de comportamiento
  - 5.6.5. Diagramas de secuencia
  - 5.6.6. Diagramas de comunicación
  - 5.6.7. Patrones para el modelado de requisitos
- 5.7. Conceptos de diseño
  - 5.7.1. Diseño en el contexto de la ingeniería del software
  - 5.7.2. El proceso de diseño
  - 5.7.3. Conceptos de diseño
  - 5.7.4. Conceptos de diseño orientado a objetos
  - 5.7.5. El modelo del diseño

- 5.8. Diseño de la arquitectura
  - 5.8.1. Arquitectura del software
  - 5.8.2. Géneros arquitectónicos
  - 5.8.3. Estilos arquitectónicos
  - 5.8.4. Diseño arquitectónico
  - 5.8.5. Evolución de los diseños alternativos para la arquitectura
  - 5.8.6. Mapeo de la arquitectura con el uso del flujo de datos
- 5.9. Diseño en el nivel de componentes y basado en patrones
  - 5.9.1. ¿Qué es un componente?
  - 5.9.2. Diseño de componentes basados en clase
  - 5.9.3. Realización del diseño en el nivel de componentes
  - 5.9.4. Diseño de componentes tradicionales
  - 5.9.5. Desarrollo basado en componentes
  - 5.9.6. Patrones de diseño
  - 5.9.7. Diseño de software basado en patrones
  - 5.9.8. Patrones arquitectónicos
  - 5.9.9. Patrones de diseño en el nivel de componentes
  - 5.9.10. Patrones de diseño de la interfaz de usuario
- 5.10. Calidad del software y administración de proyectos
  - 5.10.1. Calidad
  - 5.10.2. Calidad del software
  - 5.10.3. El dilema de la calidad del software
  - 5.10.4. Lograr la calidad del software
  - 5.10.5. Aseguramiento de la calidad del software
  - 5.10.6. El espectro administrativo
  - 5.10.7. El personal
  - 5.10.8. El producto
  - 5.10.9. El proceso
  - 5.10.10. El proyecto
  - 5.10.11. Principios y prácticas



## Módulo 6. Motores de Videojuegos

- 6.1. Los Videojuegos y las TIC
  - 6.1.1. Introducción
  - 6.1.2. Oportunidades
  - 6.1.3. Desafíos
  - 6.1.4. Conclusiones
- 6.2. Historia de los motores de Videojuegos
  - 6.2.1. Introducción
  - 6.2.2. Época Atari
  - 6.2.3. Época de los 80
  - 6.2.4. Primeros motores. Época de los 90
  - 6.2.5. Motores actuales
- 6.3. Motores de Videojuegos
  - 6.3.1. Tipos de motores
  - 6.3.2. Partes de un motor de Videojuegos
  - 6.3.3. Motores actuales
  - 6.3.4. Selección de un motor para nuestro proyecto
- 6.4. Motor *GameMaker*
  - 6.4.1. Introducción
  - 6.4.2. Diseño de escenarios
  - 6.4.3. *Sprites* y animaciones
  - 6.4.4. Colisiones
  - 6.4.5. *Scripting* en GML (*gameMaker language*)
- 6.5. Motor *Unreal Engine 4*: Introducción
  - 6.5.1. ¿Qué es *Unreal Engine 4*? ¿Cuál es su filosofía?
  - 6.5.2. Materiales
  - 6.5.3. UI (interfaz de usuario)
  - 6.5.4. Animaciones
  - 6.5.5. Sistema de partículas
  - 6.5.6. Inteligencia artificial
  - 6.5.7. FPS (*first - person shooter*)

- 6.6. Motor *Unreal Engine 4: visual scripting*
  - 6.6.1. Filosofía de los *blueprints* y el *visual scripting*
  - 6.6.2. *Debugging*
  - 6.6.3. Tipos de variables
  - 6.6.4. Control de flujo básico
- 6.7. Motor *Unity 5*
  - 6.7.1. Programación en C# y Visual Studio
  - 6.7.2. Creación de *prefabs*
  - 6.7.3. Uso de Gizmos para el control del videojuego
  - 6.7.4. Motor adaptativo: 2D y 3D
- 6.8. Motor *Godot*
  - 6.8.1. Filosofía de diseño de *Godot*
  - 6.8.2. Diseño orientado a objetos y composición
  - 6.8.3. Todo Incluido en un paquete
  - 6.8.4. Software libre y dirigido por la comunidad
- 6.9. Motor *RPG Maker*
  - 6.9.1. Filosofía de *RPG Maker*
  - 6.9.2. Tomando como referencia
  - 6.9.3. Crear un juego con personalidad
  - 6.9.4. Juegos comerciales de éxito
- 6.10. Motor *Source 2*
  - 6.10.1. Filosofía de *Source 2*
  - 6.10.2. *Source* y *Source 2*: Evolución
  - 6.10.3. Uso de la comunidad: contenido audiovisual y Videojuegos
  - 6.10.4. Futuro del Motor *Source 2*
  - 6.10.5. *Mods* y juegos de éxito

## Módulo 7. Sistemas Inteligentes

- 7.1. Teoría de agentes
  - 7.1.1. Historia del concepto
  - 7.1.2. Definición de agente
  - 7.1.3. Agentes en inteligencia artificial
  - 7.1.4. Agentes en ingeniería de *software*
- 7.2. Arquitecturas de agentes
  - 7.2.1. El proceso de razonamiento de un agente
  - 7.2.2. Agentes reactivos
  - 7.2.3. Agentes deductivos
  - 7.2.4. Agentes híbridos
  - 7.2.5. Comparativa
- 7.3. Información y conocimiento
  - 7.3.1. Distinción entre datos, información y conocimiento
  - 7.3.2. Evaluación de la calidad de los datos
  - 7.3.3. Métodos de captura de datos
  - 7.3.4. Métodos de adquisición de información
  - 7.3.5. Métodos de adquisición de conocimiento
- 7.4. Representación del conocimiento
  - 7.4.1. La importancia de la representación del conocimiento
  - 7.4.2. Definición de representación del conocimiento a través de sus roles
  - 7.4.3. Características de una representación del conocimiento
- 7.5. Ontologías
  - 7.5.1. Introducción a los metadatos
  - 7.5.2. Concepto filosófico de ontología
  - 7.5.3. Concepto informático de ontología
  - 7.5.4. Ontologías de dominio y ontologías de nivel superior
  - 7.5.5. ¿Cómo construir una ontología?

- 7.6. Lenguajes para ontologías y *software* para la creación de ontologías
  - 7.6.1. Tripletas RDF, Turtle y N3
  - 7.6.2. RDF *Schema*
  - 7.6.3. OW
  - 7.6.4. SPARQL
  - 7.6.5. Introducción a las diferentes herramientas para la creación de ontologías
  - 7.6.6. Instalación y uso de Protégé
- 7.7. La web semántica
  - 7.7.1. El estado actual y futuro de la web semántica
  - 7.7.2. Aplicaciones de la web semántica
- 7.8. Otros modelos de representación del conocimiento
  - 7.8.1. Vocabularios
  - 7.8.2. Visión global
  - 7.8.3. Taxonomías
  - 7.8.4. Tesoros
  - 7.8.5. Folksonomías
  - 7.8.6. Comparativa
  - 7.8.7. Mapas mentales
- 7.9. Evaluación e integración de representaciones del conocimiento
  - 7.9.1. Lógica de orden cero
  - 7.9.2. Lógica de primer orden
  - 7.9.3. Lógica descriptiva
  - 7.9.4. Relación entre diferentes tipos de lógica
  - 7.9.5. *Prolog*: Programación basada en lógica de primer orden
- 7.10. Razonadores semánticos, sistemas basados en conocimiento y sistemas expertos
  - 7.10.1. Concepto de razonador
  - 7.10.2. Aplicaciones de un eazonador
  - 7.10.3. Sistemas basados en el conocimiento
  - 7.10.4. *MYCIN*, historia de los sistemas expertos
  - 7.10.5. Elementos y arquitectura de sistemas expertos
  - 7.10.6. Creación de sistemas expertos

## Módulo 8. Programación en Tiempo Real

- 8.1. Conceptos básicos de la Programación concurrente
  - 8.1.1. Conceptos fundamentales
  - 8.1.2. Concurrencia
  - 8.1.3. Beneficios de la concurrencia
  - 8.1.4. Concurrencia y *hardware*
- 8.2. Estructuras básicas de soporte a la concurrencia en Java
  - 8.2.1. Concurrencia en Java
  - 8.2.2. Creación de *threads*
  - 8.2.3. Métodos
  - 8.2.4. Sincronización
- 8.3. *Threads*, ciclo de vida, prioridades, interrupciones, estados, ejecutores
  - 8.3.1. *Threads*
  - 8.3.2. Ciclo de vida
  - 8.3.3. Prioridades
  - 8.3.4. Interrupciones
  - 8.3.5. Estados
  - 8.3.6. Ejecutores
- 8.4. Exclusión mutua
  - 8.4.1. ¿Qué es la exclusión mutua?
  - 8.4.2. Algoritmo de Dekker
  - 8.4.3. Algoritmo de Peterson
  - 8.4.4. Exclusión mutua en Java
- 8.5. Dependencias de estados
  - 8.5.1. Inyección de dependencias
  - 8.5.2. Implementación del patrón en Java
  - 8.5.3. Formas de inyectar las dependencias
  - 8.5.4. Ejemplo

- 8.6. Patrones de diseño
  - 8.6.1. Introducción
  - 8.6.2. Patrones de creación
  - 8.6.3. Patrones de estructura
  - 8.6.4. Patrones de comportamiento
- 8.7. Uso de bibliotecas Java
  - 8.7.1. ¿Qué son las bibliotecas en Java?
  - 8.7.2. *Mockito - All, Mockito - Core*
  - 8.7.3. *Guava*
  - 8.7.4. *Commons - IO*
  - 8.7.5. *Commons - Lang, Commons - Lang3*
- 8.8. Programación de *shaders*
  - 8.8.1. *Pipeline 3D y rasterizado*
  - 8.8.2. *Vertex Shading*
  - 8.8.3. *Pixel Shading: Iluminación I*
  - 8.8.4. *Pixel Shading: Iluminación II*
  - 8.8.5. *Post - efectos*
- 8.9. Programación de tiempo real
  - 8.9.1. Introducción
  - 8.9.2. Procesamiento de interrupciones
  - 8.9.3. Sincronización y comunicación entre procesos
  - 8.9.4. Los sistemas de planificación en tiempo real
- 8.10. Planificación de tiempo real
  - 8.10.1. Conceptos
  - 8.10.2. Modelo de referencia de los sistemas de tiempo real
  - 8.10.3. Políticas de planificación
  - 8.10.4. Planificadores cíclicos
  - 8.10.5. Planificadores con propiedades estáticas
  - 8.10.6. Planificadores con propiedades dinámicas

## Módulo 9. Diseño y Desarrollo de Juegos Web

- 9.1. Orígenes y estándares de la web
  - 9.1.1. Orígenes de internet
  - 9.1.2. Creación de *world wide web*
  - 9.1.3. Aparición de los estándares web
  - 9.1.4. El auge de los estándares web
- 9.2. HTTP y estructura cliente - servidor
  - 9.2.1. Rol cliente - servidor
  - 9.2.2. Comunicación cliente - servidor
  - 9.2.3. Historia reciente
  - 9.2.4. Computación centralizada
- 9.3. Programación web: Introducción
  - 9.3.1. Conceptos básicos
  - 9.3.2. Preparando un servidor web
  - 9.3.3. Conceptos básicos de HTML5
  - 9.3.4. Formas HTML
- 9.4. Introducción a HTML y ejemplos
  - 9.4.1. Historia de HTML5
  - 9.4.2. Elementos de HTML5
  - 9.4.3. APIs
  - 9.4.4. CSS3
- 9.5. Modelo de objeto de documento
  - 9.5.1. ¿Qué es el modelo de objetos del documento?
  - 9.5.2. Uso de DOCTYPE
  - 9.5.3. La importancia de validar el HTML
  - 9.5.4. Accediendo a los elementos
  - 9.5.5. Creando elementos y textos
  - 9.5.6. Usando innerHTML
  - 9.5.7. Eliminando un elemento o nodo de texto
  - 9.5.8. Lectura y escritura de los atributos de un elemento
  - 9.5.9. Manipulando los estilos de los elementos
  - 9.5.10. Adjuntar múltiples ficheros a la vez



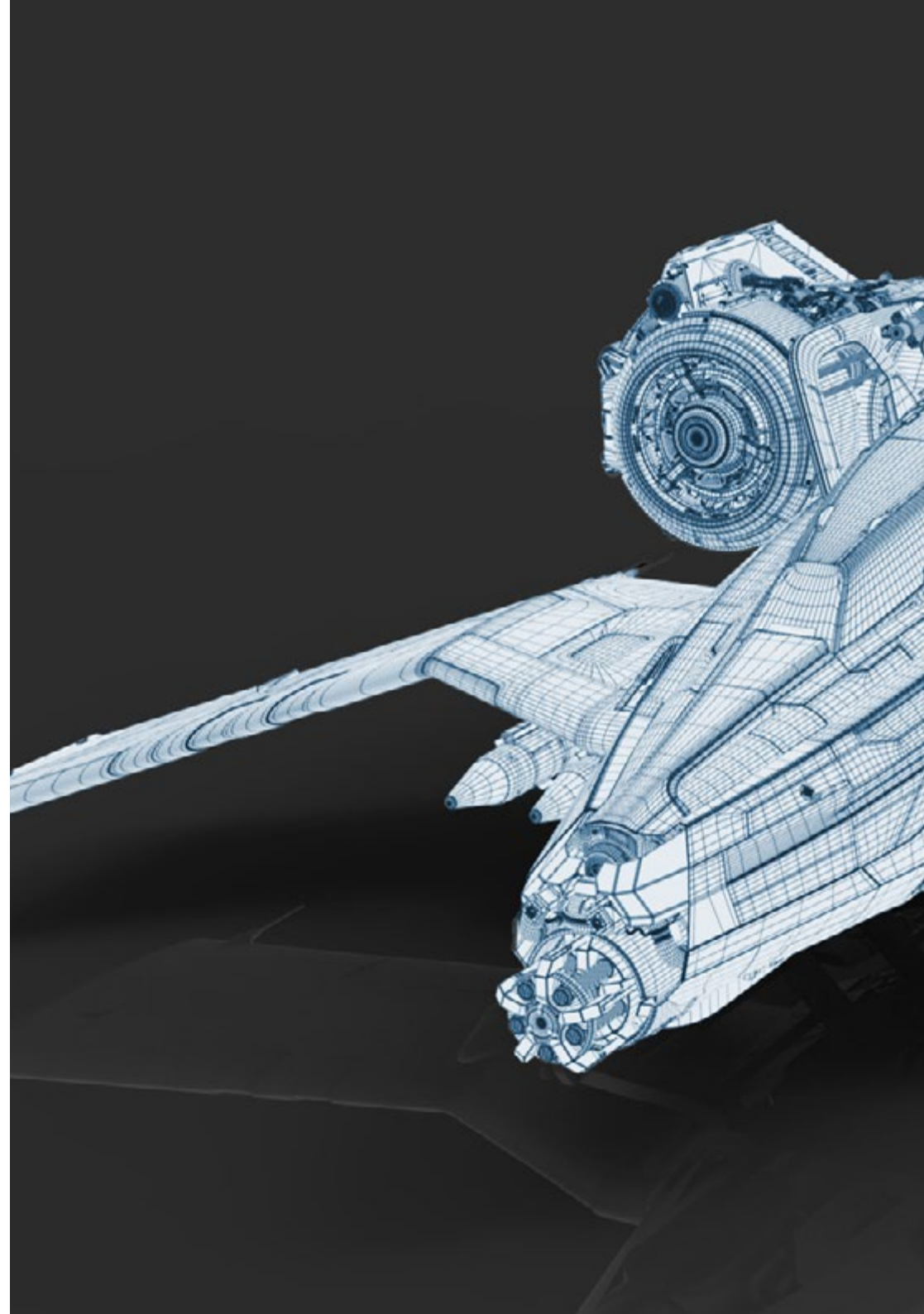
- 9.6. Introducción a CSS y Ejemplos
  - 9.6.1. Sintaxis CSS3
  - 9.6.2. Hojas de estilo
  - 9.6.3. Etiquetas
  - 9.6.4. Selectores
  - 9.6.5. Diseño web con CSS
- 9.7. Introducción a JavaScript y ejemplos
  - 9.7.1. ¿Qué es JavaScript?
  - 9.7.2. Breve historia del lenguaje
  - 9.7.3. Versiones de JavaScript
  - 9.7.4. Mostrar un cuadro de diálogo
  - 9.7.5. Sintaxis de JavaScript
  - 9.7.6. Comprensión de *scripts*
  - 9.7.7. Espacios
  - 9.7.8. Comentarios
  - 9.7.9. Funciones
  - 9.7.10. JavaScript en la página y externo
- 9.8. Funciones en JavaScript
  - 9.8.1. Declaraciones de función
  - 9.8.2. Expresiones de función
  - 9.8.3. Llamar a funciones
  - 9.8.4. Recursividad
  - 9.8.5. Funciones anidadas y cierres
  - 9.8.6. Preservación de variables
  - 9.8.7. Funciones multi - Anidadas
  - 9.8.8. Conflictos de nombres
  - 9.8.9. Clausuras o cierres
  - 9.8.10. Parámetros de una función

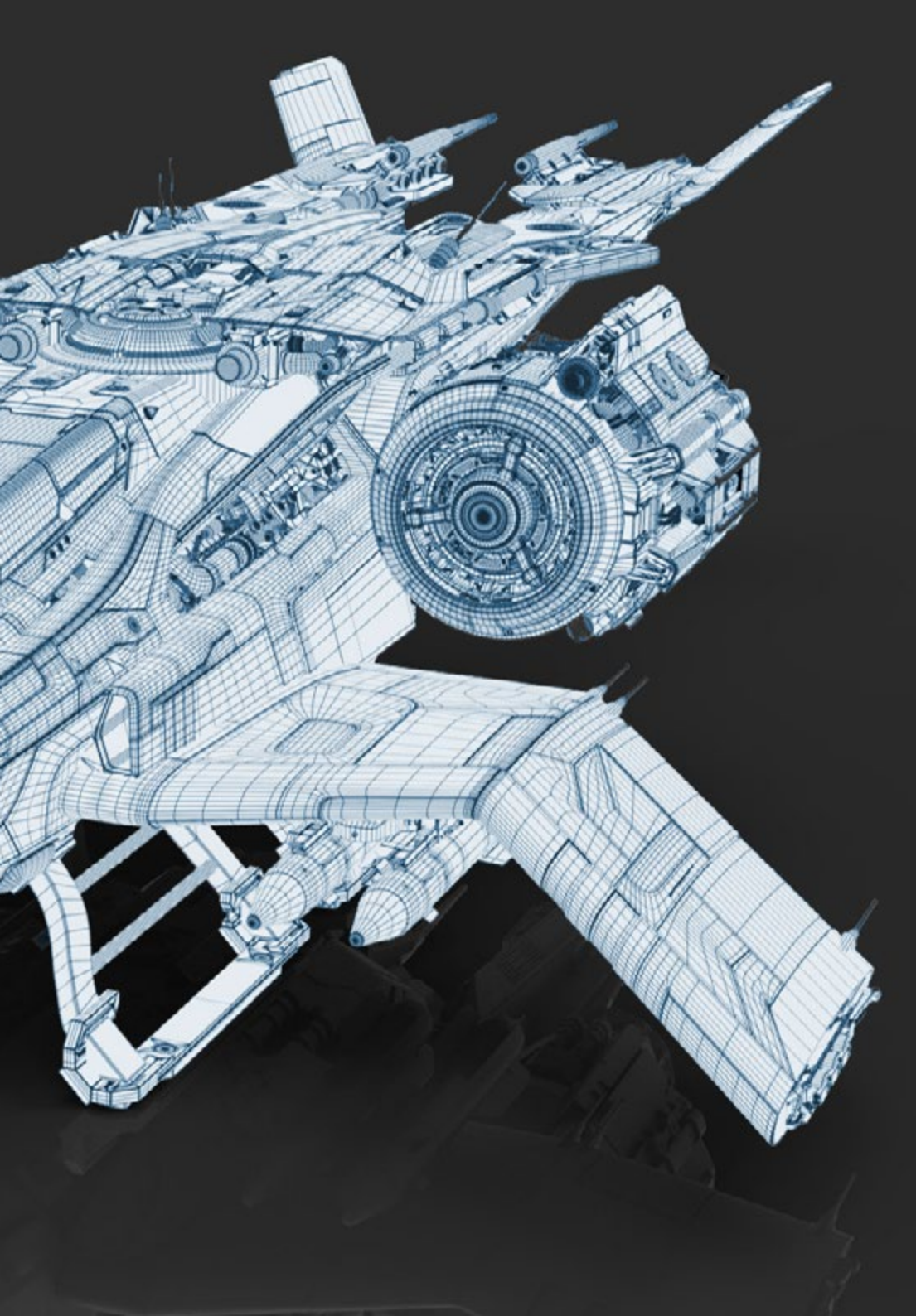
- 9.9. PlayCanvas para desarrollar Juegos Web
  - 9.9.1. ¿Qué es PlayCanvas?
  - 9.9.2. Configuración del proyecto
  - 9.9.3. Creando un objeto
  - 9.9.4. Agregando físicas
  - 9.9.5. Añadiendo un modelo
  - 9.9.6. Cambiando los ajustes de gravedad y escena
  - 9.9.7. Ejecutando *scripts*
  - 9.9.8. Controles de cámara
- 9.10. *Phaser* para desarrollar juegos web
  - 9.10.1. ¿Qué es *phaser*?
  - 9.10.2. Cargando recursos
  - 9.10.3. Construyendo el mundo
  - 9.10.4. Las plataformas
  - 9.10.5. El jugador
  - 9.10.6. Añadir físicas
  - 9.10.7. Usar el teclado
  - 9.10.8. Recoger *pickups*
  - 9.10.9. Puntos y puntuación
  - 9.10.10. Bombas de rebote

## Módulo 10. Redes y Sistemas Multijugador

- 10.1. Historia y evolución de Videojuegos multijugador
  - 10.1.1. Década 1970: Primeros juegos multijugador
  - 10.1.2. Años 90: *Duke nukem, doom, quake*
  - 10.1.3. Auge de Videojuegos multijugador
  - 10.1.4. Multijugador local y online
  - 10.1.5. Juegos de fiesta

- 10.2. Modelos de negocio multijugador
  - 10.2.1. Origen y funcionamiento de los modelos de negocio emergentes
  - 10.2.2. Servicios de venta en línea
  - 10.2.3. Libre para Jugar
  - 10.2.4. Micropagos
  - 10.2.5. Publicidad
  - 10.2.6. Suscripción con pagos mensuales
  - 10.2.7. Pagar por juego
  - 10.2.8. Prueba antes de comprar
- 10.3. Juegos locales y juegos en red
  - 10.3.1. Juegos locales: Inicios
  - 10.3.2. Juegos de fiesta: Nintendo y la unión de la familia
  - 10.3.3. Juegos en red: Inicios
  - 10.3.4. Evolución de los juegos en red
- 10.4. Modelo OSI: Capas I
  - 10.4.1. Modelo OSI: Introducción
  - 10.4.2. Capa física
  - 10.4.3. Capa de enlace de datos
  - 10.4.4. Capa de red
- 10.5. Modelo OSI: Capas II
  - 10.5.1. Capa de transporte
  - 10.5.2. Capa de sesión
  - 10.5.3. Capa de presentación
  - 10.5.4. Capa de aplicación
- 10.6. Redes de computadoras e internet
  - 10.6.1. ¿Qué es una red de computadoras?
  - 10.6.2. Software
  - 10.6.3. Hardware
  - 10.6.4. Servidores
  - 10.6.5. Almacenamiento en red
  - 10.6.6. Protocolos de red





- 10.7. Redes móviles e inalámbricas
  - 10.7.1. Red móvil
  - 10.7.2. Red inalámbrica
  - 10.7.3. Funcionamiento de las redes móviles
  - 10.7.4. Tecnología digital
- 10.8. Seguridad
  - 10.8.1. Seguridad personal
  - 10.8.2. *Hacks* y *cheats* en Videojuegos
  - 10.8.3. Seguridad anti-trampas
  - 10.8.4. Análisis de sistemas de seguridad anti-trampas
- 10.9. Sistemas multijugador: Servidores
  - 10.9.1. Alojamiento de servidores
  - 10.9.2. Videojuegos MMO
  - 10.9.3. Servidores de Videojuegos dedicados
  - 10.9.4. *LAN parties*
- 10.10. Diseño de Videojuegos multijugador y Programación
  - 10.10.1. Fundamentos de diseño de Videojuegos multijugador en Unreal
  - 10.10.2. Fundamentos de diseño de Videojuegos multijugador en Unity
  - 10.10.3. Cómo hacer que un juego multijugador sea divertido
  - 10.10.4. Más allá de un mando: innovación en controles multijugador

“ Colaborarás en equipos multidisciplinares, entendiendo las necesidades del diseño artístico, narrativo y experiencia de usuario”

# 04

## Objetivos docentes

Este Máster Título Propio de TECH, está diseñado para proporcionar a los programadores las herramientas más avanzadas para el desarrollo de juegos interactivos de calidad. Por lo tanto, los egresados dominarán operaciones con ficheros, lo que les permitirá gestionar eficientemente los datos y recursos dentro de los proyectos de Videojuegos. Además, estarán especializados en arquitecturas de agentes, creando personajes autónomos con inteligencia artificial para una interacción realista dentro de los juegos. Igualmente, estarán capacitados en el uso de algoritmos con grafos, esenciales para la resolución de problemas complejos de navegación y toma de decisiones en entornos dinámicos.



“

*Diseñarás experiencias interactivas optimizadas para consolas como Xbox One, aprovechando las características técnicas y capacidades de cada plataforma”*



## Objetivos generales

---

- ♦ Conocer los diferentes lenguajes y métodos de Programación aplicados al videojuego
- ♦ Profundizar en el proceso de producción de un videojuego y en la integración de la Programación en estas etapas
- ♦ Comprender los fundamentos del diseño de Videojuegos y aquellos conocimientos teóricos que un diseñador debe conocer
- ♦ Dominar los lenguajes de Programación básicos empleados en Videojuegos
- ♦ Aplicar conocimientos de la ingeniería de *software* y programación especializada a los Videojuegos
- ♦ Entender el papel de la Programación en el desarrollo de un videojuego
- ♦ Conocer las distintas consolas y plataformas existentes
- ♦ Desarrollar Videojuegos web y multijugador



*Optimizarás los códigos y recursos para garantizar un rendimiento eficiente en diversas plataformas como consolas y dispositivos móviles”*





## Objetivos específicos

---

### Módulo 1. Fundamentos de Programación

- ♦ Comprender la estructura básica de un ordenador, el software y de los lenguajes de Programación de propósito general
- ♦ Interpretar algoritmos, que son la base necesaria para poder desarrollar programas informáticos

### Módulo 2. Estructura de Datos y Algoritmos

- ♦ Comprender las principales estrategias de diseño de algoritmos, así como los distintos métodos y medidas para de cálculo de los mismos
- ♦ Distinguir el funcionamiento de los algoritmos, su estrategia y ejemplos de su uso en los principales problemas conocidos
- ♦ Entender la técnica de *backtracking* y sus principales usos
- ♦ Desarrollar habilidades para implementar algoritmos de búsqueda de caminos mínimos

### Módulo 3. Programación Orientada a Objetos

- ♦ Conocer los distintos patrones de diseño para problemas orientados a objetos
- ♦ Entender la importancia de la documentación y las pruebas en el desarrollo del software
- ♦ Gestionar el uso de los hilos y la sincronización, así como la resolución de los problemas comunes dentro de la Programación concurrente
- ♦ Adquirir los conocimientos y habilidades necesarios para diseñar e implementar interfaces de usuario interactivas

### Módulo 4. Consolas y Dispositivos para Videojuegos

- ♦ Conocer el funcionamiento básico de los principales periféricos de entrada y salida
- ♦ Entender las principales implicaciones de diseño de las diferentes plataformas
- ♦ Examinar la estructura, organización, funcionamiento e interconexión de los dispositivos y sistemas
- ♦ Comprender la función del sistema operativo y los kits de desarrollo para dispositivos móviles y plataformas de Videojuegos

### Módulo 5. Ingeniería de Software

- ♦ Distinguir las bases de la ingeniería del software, así como el proceso del software en los distintos modelos para su desarrollo incluyendo tecnologías ágiles
- ♦ Desarrollar la capacidad de estructurar la arquitectura del *software* de un videojuego de forma escalable, modular y eficiente

### Módulo 6. Motores de Videojuegos

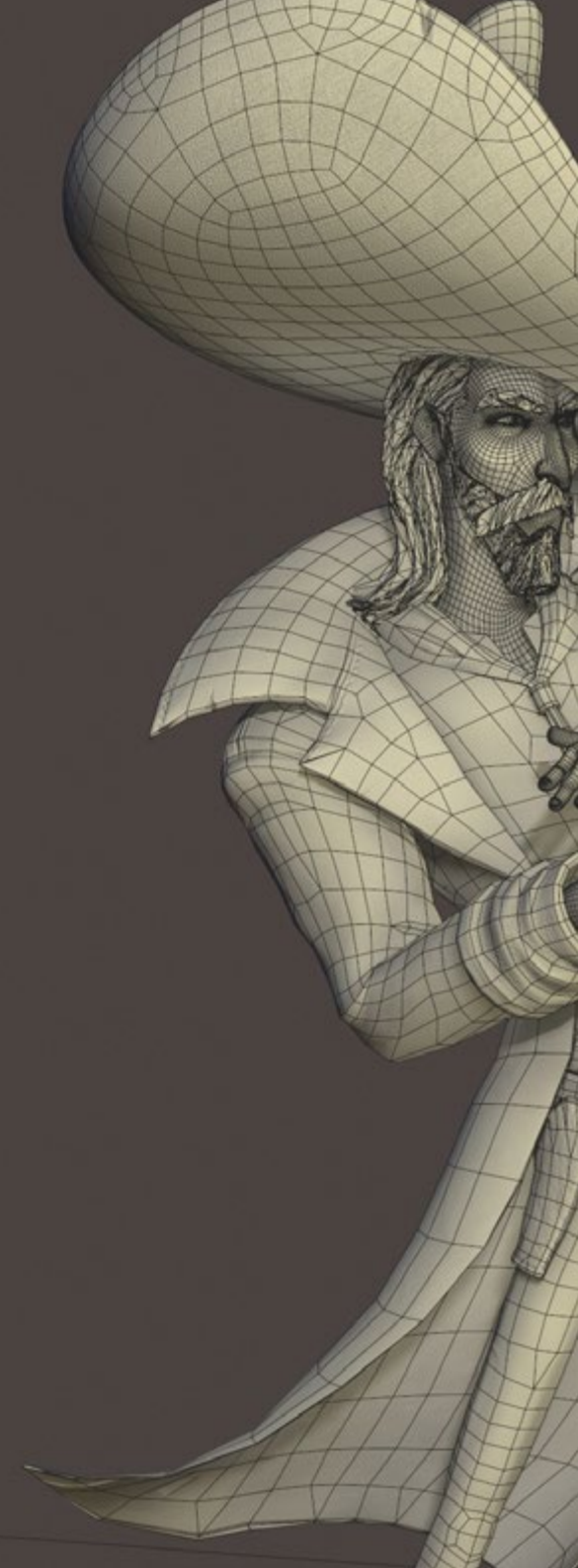
- ♦ Descubrir el funcionamiento y la arquitectura de un motor de Videojuegos
- ♦ Comprender las características básicas de los motores de juegos existentes
- ♦ Ejecutar la programación de aplicaciones de manera correcta y eficiente aplicadas a motores de Videojuegos
- ♦ Elegir el paradigma y los lenguajes de Programación más apropiados para programar aplicaciones aplicadas a motores de Videojuegos

### Módulo 7. Sistemas Inteligentes

- ♦ Establecer los conceptos relacionados con la teoría de agentes y la arquitectura de agentes y su proceso de razonamiento
- ♦ Asimilar la teoría y la práctica detrás de los conceptos de información y conocimiento, así como las distintas maneras de representar el conocimiento
- ♦ Comprender el funcionamiento de los razonadores semánticos, los sistemas basados en conocimiento y los sistemas expertos
- ♦ Integrar los principios de la web semántica en el desarrollo de Videojuegos que incluyan funcionalidades en línea

### Módulo 8. Programación en Tiempo Real

- ♦ Analizar las características clave de un lenguaje de Programación en tiempo real que lo diferencian del lenguaje de Programación tradicional
- ♦ Comprender los conceptos básicos de los sistemas informáticos
- ♦ Adquirir la capacidad de aplicar las principales bases y técnicas de Programación en tiempo real
- ♦ Dominar el manejo de hilos y sus componentes asociados, como el ciclo de vida, prioridades, interrupciones, estados y ejecutores







### **Módulo 9. Diseño y Desarrollo de Juegos Web**

- ♦ Capacitar para diseñar juegos y aplicaciones web interactivas con la documentación correspondiente
- ♦ Evaluar las características principales de los juegos y las aplicaciones web interactivas para comunicarse de manera profesional y correcta
- ♦ Adquirir las competencias necesarias para desarrollar y gestionar la comunicación entre cliente y servidor en Videojuegos, utilizando el protocolo HTTP y principios de arquitectura cliente-servidor
- ♦ Desarrollar la capacidad de utilizar funciones en JavaScript de manera eficiente para estructurar, modularizar y optimizar el código de Videojuegos interactivos basados en la web

### **Módulo 10. Redes y Sistemas Multijugador**

- ♦ Describir la arquitectura del protocolo de control de transmisión/protocolo de Internet (TCP / IP) y el funcionamiento básico de las redes Inalámbricas
- ♦ Analizar la seguridad aplicada a Videojuegos
- ♦ Adquirir la capacidad para desarrollar juegos en línea para múltiples jugadores
- ♦ Desarrollar Videojuegos en modo local y en red, implementando mecánicas multijugador que aseguren la correcta sincronización entre jugadores

# 05

## Salidas profesionales

Este programa universitario de TECH representa una gran oportunidad para los programadores de Videojuegos que desean perfeccionar sus competencias en el desarrollo de sistemas interactivos avanzados. Es así como, tras concluir el plan de estudios, los egresados dominarán la implementación de aplicaciones de razonamiento para crear decisiones inteligentes dentro de sus juegos. Asimismo, estarán capacitados para emplear técnicas de *backtracking* con el fin de resolver problemas complejos en entornos dinámicos, optimizando la jugabilidad y la experiencia del usuario. Además, tendrán la habilidad de trabajar con threads para optimizar la ejecución de sus programas y mejorar el rendimiento de las aplicaciones.



“

*¿Buscas ejercitarte como Desarrollador de Motores de Videojuegos? Lógralo mediante esta titulación universitaria en tan solo 12 meses”*

### Perfil del egresado

El egresado de Máster Título Propio en Programación de Videojuegos de TECH, dominará el desarrollo de interfaces de usuario, diseñando experiencias interactivas intuitivas y atractivas para los jugadores. Además, aplicará sistemas de planificación en tiempo real para gestionar de manera eficiente las tareas dentro del videojuego, asegurando un rendimiento fluido. A su vez, manejará hilos en C++ para optimizar la ejecución de procesos concurrentes, mejorando la fluidez y la respuesta del juego en tiempo real. En adición, estará capacitado para desarrollar Videojuegos altamente funcionales, interactivos y con un rendimiento excepcional.

*Crearás herramientas internas para facilitar los procesos de producción de los equipos de arte, diseño y sonido.*

- ♦ **Adaptación Tecnológica en Entornos Interactivos:** integrar nuevas tecnologías en el diseño y desarrollo de Videojuegos, optimizando la experiencia del usuario y la eficiencia del proceso creativo
- ♦ **Resolución de Problemas Computacionales:** aplicar pensamiento lógico y crítico en la depuración de código, garantizando el rendimiento fluido de las mecánicas y sistemas de juego
- ♦ **Implementación de Algoritmos Complejos:** utilizar estructuras avanzadas como Grafos, *backtracking* o arquitecturas de agentes, mejorando la toma de decisiones dentro del entorno virtual
- ♦ **Ética Profesional y Experiencia del Jugador:** desarrollar Videojuegos responsables, centrados en la calidad, la inclusión y la satisfacción del usuario final



Después de realizar el programa universitario, podrás desempeñar tus conocimientos y habilidades en los siguientes cargos:

1. **Gameplay Programmer:** encargado de implementar las mecánicas jugables, interacciones y reglas del juego, asegurando que la experiencia del jugador sea fluida, divertida y funcional.
2. **AI Programmer:** responsable de diseñar y desarrollar el comportamiento de los personajes no jugables, utilizando algoritmos y técnicas como árboles de decisión.
3. **Graphics Programmer:** dedicado al renderizado y optimización gráfica. Trabaja con *shaders*, iluminación, efectos visuales y APIs gráficas para mejorar el aspecto visual del juego.
4. **Tools Programmer:** encargado de crear herramientas personalizadas para facilitar el trabajo del equipo de desarrollo, como editores de niveles, *scripts* de automatización o interfaces de usuario internas.
5. **Network Programmer:** responsable de la arquitectura multijugador y la sincronización en red. Desarrolla los sistemas que permiten la comunicación entre jugadores y servidores de forma estable y segura.
6. **Engine Programmer:** encargado trabajar directamente en el motor del juego, modificando o creando componentes fundamentales del sistema como físicas, audio, animaciones o compilación de escenas.

“ Analizarás las tendencias tecnológicas y aplicarás innovaciones que mejoren la calidad de los productos ”

# 06

## Licencias de software incluidas

TECH es referencia en el mundo universitario por combinar la última tecnología con las metodologías docentes para potencial el proceso de enseñanza-aprendizaje. Para ello, ha establecido una red de alianzas que le permite tener acceso a las herramientas de software más avanzadas del mundo profesional.



“

*Al matricularte recibirás, de forma completamente gratuita, las credenciales de uso académico de las siguientes aplicaciones de software profesional”*

TECH ha establecido una red de alianzas profesionales en la que se encuentran los principales proveedores de software aplicado a las diferentes áreas profesionales. Estas alianzas permiten a TECH tener acceso al uso de centenares de aplicaciones informáticas y licencias de software para acercarlas a sus estudiantes.

Las licencias de software para uno académico permitirán a los estudiantes utilizar las aplicaciones informáticas más avanzadas en su área profesional, de modo que podrán conocerlas y aprender su dominio sin tener que incurrir en costes. TECH se hará cargo del procedimiento de contratación para que los alumnos puedan utilizarlas de modo ilimitado durante el tiempo que estén estudiando el programa de Máster Título Propio en Programación de Videojuegos, y además lo podrán hacer de forma completamente gratuita.

TECH te dará acceso gratuito al uso de las siguientes aplicaciones de software:



### Unity

**Unity** es una plataforma líder en desarrollo de software para la creación de experiencias interactivas y multimedia en 2D y 3D. Durante el transcurso de este programa universitario, los alumnos tendrán **acceso gratuito** a esta plataforma, cuyo valor aproximado es de **2040 dólares**, permitiendo así trabajar con una solución profesional **sin coste adicional**.

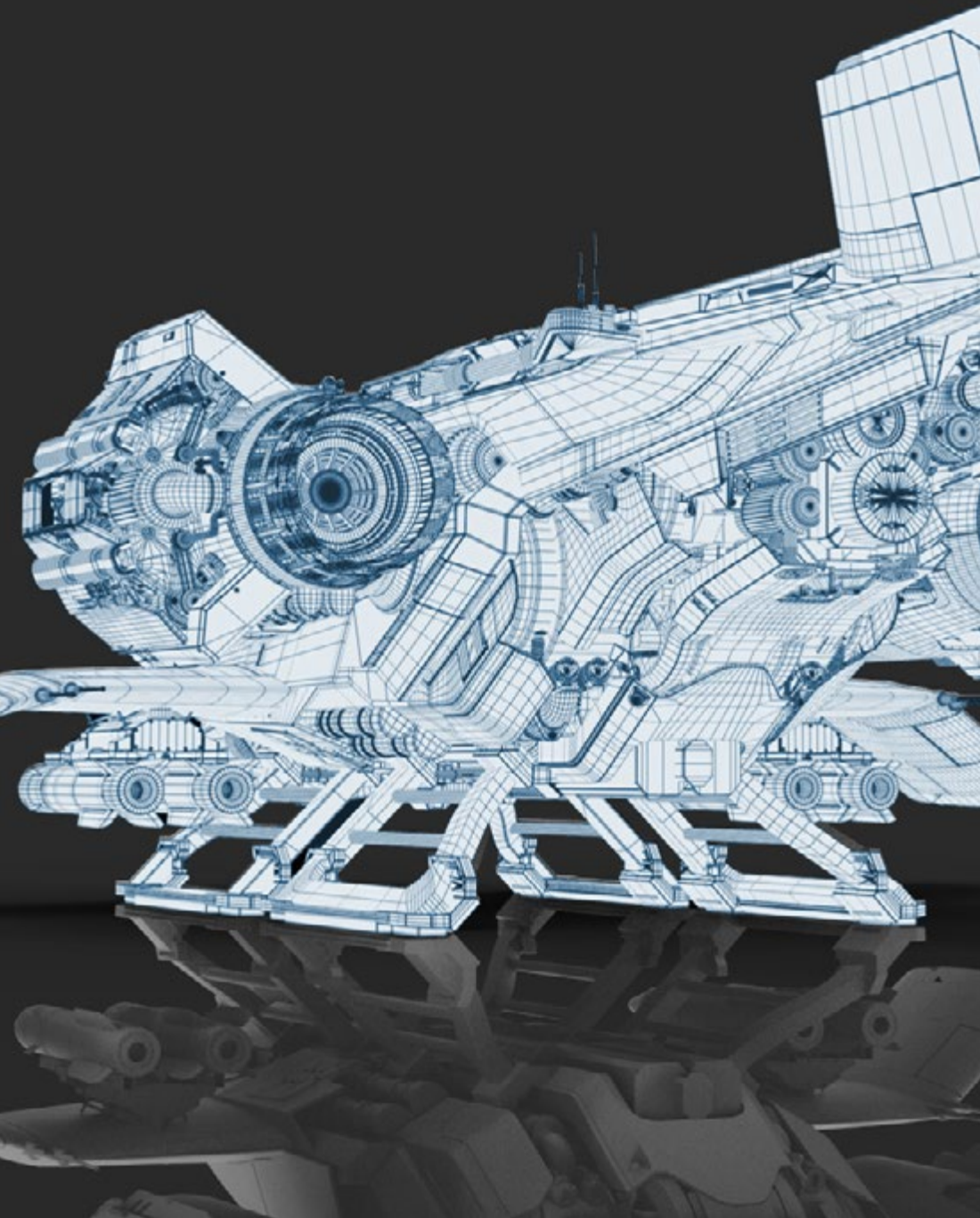
Esta herramienta ofrece un motor en tiempo real que soporta gráficos con renderizado de alta calidad, además de potentes herramientas de scripting que facilitan la personalización avanzada. **Unity** incluye un sistema de física integrado para simulaciones realistas, soporte multiplataforma para dispositivos móviles, consolas y tecnologías VR/AR y una amplia Asset Store con recursos que aceleran el desarrollo.

### Google Career Launchpad

**Google Career Launchpad** es una solución para desarrollar habilidades digitales en tecnología y análisis de datos. Con un valor estimado de **5.000 dólares**, se incluye de forma **gratuita** en el programa universitario de TECH, brindando acceso a laboratorios interactivos y certificaciones reconocidas en el sector.

Esta plataforma combina capacitación técnica con casos prácticos, usando tecnologías como BigQuery y Google AI. Ofrece entornos simulados para experimentar con datos reales, junto a una red de expertos para orientación personalizada.





### 3ds Max

Durante la realización del programa, TECH pone a disposición de los egresados la licencia oficial de **3ds Max**, valorada en **2.300 euros, sin coste** adicional. Esta herramienta destaca en áreas como diseño arquitectónico, animación digital y simulación visual, y permite trabajar con tecnología líder del sector profesional.

Esta plataforma ofrece un entorno robusto para modelar, animar y renderizar proyectos complejos con precisión y eficiencia. Gracias a su arquitectura flexible, los usuarios pueden desarrollar desde visualizaciones estáticas hasta escenas completas de animación, utilizando funciones avanzadas en un espacio optimizado para resultados de alto nivel.

### Maya

Durante este programa universitario, los egresados tendrán acceso **gratis** a **Maya**, una potente herramienta profesional valorada en **2.250 euros**. Se emplea ampliamente en la industria audiovisual para desarrollar animaciones 3D, modelado, simulación y renderizado, siendo esencial en cine, televisión y videojuegos de alto nivel.

Esta plataforma permite construir entornos y personajes con alto nivel de detalle, gestionar efectos visuales complejos y ejecutar procesos avanzados de animación. Su presencia en este programa refuerza las capacidades técnicas en contextos reales, favoreciendo la inserción laboral con recursos utilizados por estudios internacionales líderes en contenido digital

07

# Metodología de estudio

TECH es la primera universidad en el mundo que combina la metodología de los **case studies** con el **Relearning**, un sistema de aprendizaje 100% online basado en la reiteración dirigida.

Esta disruptiva estrategia pedagógica ha sido concebida para ofrecer a los profesionales la oportunidad de actualizar conocimientos y desarrollar competencias de un modo intensivo y riguroso. Un modelo de aprendizaje que coloca al estudiante en el centro del proceso académico y le otorga todo el protagonismo, adaptándose a sus necesidades y dejando de lado las metodologías más convencionales.



“

*TECH te prepara para afrontar nuevos retos en entornos inciertos y lograr el éxito en tu carrera”*

## El alumno: la prioridad de todos los programas de TECH

En la metodología de estudios de TECH el alumno es el protagonista absoluto. Las herramientas pedagógicas de cada programa han sido seleccionadas teniendo en cuenta las demandas de tiempo, disponibilidad y rigor académico que, a día de hoy, no solo exigen los estudiantes sino los puestos más competitivos del mercado.

Con el modelo educativo asincrónico de TECH, es el alumno quien elige el tiempo que destina al estudio, cómo decide establecer sus rutinas y todo ello desde la comodidad del dispositivo electrónico de su preferencia. El alumno no tendrá que asistir a clases en vivo, a las que muchas veces no podrá acudir. Las actividades de aprendizaje las realizará cuando le venga bien. Siempre podrá decidir cuándo y desde dónde estudiar.

“

*En TECH NO tendrás clases en directo  
(a las que luego nunca puedes asistir)”*



### Los planes de estudios más exhaustivos a nivel internacional

TECH se caracteriza por ofrecer los itinerarios académicos más completos del entorno universitario. Esta exhaustividad se logra a través de la creación de temarios que no solo abarcan los conocimientos esenciales, sino también las innovaciones más recientes en cada área.

Al estar en constante actualización, estos programas permiten que los estudiantes se mantengan al día con los cambios del mercado y adquieran las habilidades más valoradas por los empleadores. De esta manera, quienes finalizan sus estudios en TECH reciben una preparación integral que les proporciona una ventaja competitiva notable para avanzar en sus carreras.

Y además, podrán hacerlo desde cualquier dispositivo, pc, tableta o smartphone.

“

*El modelo de TECH es asincrónico, de modo que te permite estudiar con tu pc, tableta o tu smartphone donde quieras, cuando quieras y durante el tiempo que quieras”*

## Case studies o Método del caso

El método del caso ha sido el sistema de aprendizaje más utilizado por las mejores escuelas de negocios del mundo. Desarrollado en 1912 para que los estudiantes de Derecho no solo aprendiesen las leyes a base de contenidos teóricos, su función era también presentarles situaciones complejas reales. Así, podían tomar decisiones y emitir juicios de valor fundamentados sobre cómo resolverlas. En 1924 se estableció como método estándar de enseñanza en Harvard.

Con este modelo de enseñanza es el propio alumno quien va construyendo su competencia profesional a través de estrategias como el *Learning by doing* o el *Design Thinking*, utilizadas por otras instituciones de renombre como Yale o Stanford.

Este método, orientado a la acción, será aplicado a lo largo de todo el itinerario académico que el alumno emprenda junto a TECH. De ese modo se enfrentará a múltiples situaciones reales y deberá integrar conocimientos, investigar, argumentar y defender sus ideas y decisiones. Todo ello con la premisa de responder al cuestionamiento de cómo actuaría al posicionarse frente a eventos específicos de complejidad en su labor cotidiana.



## Método Relearning

En TECH los *case studies* son potenciados con el mejor método de enseñanza 100% online: el *Relearning*.

Este método rompe con las técnicas tradicionales de enseñanza para poner al alumno en el centro de la ecuación, proveyéndole del mejor contenido en diferentes formatos. De esta forma, consigue repasar y reiterar los conceptos clave de cada materia y aprender a aplicarlos en un entorno real.

En esta misma línea, y de acuerdo a múltiples investigaciones científicas, la reiteración es la mejor manera de aprender. Por eso, TECH ofrece entre 8 y 16 repeticiones de cada concepto clave dentro de una misma lección, presentada de una manera diferente, con el objetivo de asegurar que el conocimiento sea completamente afianzado durante el proceso de estudio.

*El Relearning te permitirá aprender con menos esfuerzo y más rendimiento, implicándote más en tu especialización, desarrollando el espíritu crítico, la defensa de argumentos y el contraste de opiniones: una ecuación directa al éxito.*



## Un Campus Virtual 100% online con los mejores recursos didácticos

Para aplicar su metodología de forma eficaz, TECH se centra en proveer a los egresados de materiales didácticos en diferentes formatos: textos, vídeos interactivos, ilustraciones y mapas de conocimiento, entre otros. Todos ellos, diseñados por profesores cualificados que centran el trabajo en combinar casos reales con la resolución de situaciones complejas mediante simulación, el estudio de contextos aplicados a cada carrera profesional y el aprendizaje basado en la reiteración, a través de audios, presentaciones, animaciones, imágenes, etc.

Y es que las últimas evidencias científicas en el ámbito de las Neurociencias apuntan a la importancia de tener en cuenta el lugar y el contexto donde se accede a los contenidos antes de iniciar un nuevo aprendizaje. Poder ajustar esas variables de una manera personalizada favorece que las personas puedan recordar y almacenar en el hipocampo los conocimientos para retenerlos a largo plazo. Se trata de un modelo denominado *Neurocognitive context-dependent e-learning* que es aplicado de manera consciente en esta titulación universitaria.

Por otro lado, también en aras de favorecer al máximo el contacto mentor-alumno, se proporciona un amplio abanico de posibilidades de comunicación, tanto en tiempo real como en diferido (mensajería interna, foros de discusión, servicio de atención telefónica, email de contacto con secretaría técnica, chat y videoconferencia).

Asimismo, este completísimo Campus Virtual permitirá que el alumnado de TECH organice sus horarios de estudio de acuerdo con su disponibilidad personal o sus obligaciones laborales. De esa manera tendrá un control global de los contenidos académicos y sus herramientas didácticas, puestas en función de su acelerada actualización profesional.



*La modalidad de estudios online de este programa te permitirá organizar tu tiempo y tu ritmo de aprendizaje, adaptándolo a tus horarios”*

### La eficacia del método se justifica con cuatro logros fundamentales:

1. Los alumnos que siguen este método no solo consiguen la asimilación de conceptos, sino un desarrollo de su capacidad mental, mediante ejercicios de evaluación de situaciones reales y aplicación de conocimientos.
2. El aprendizaje se concreta de una manera sólida en capacidades prácticas que permiten al alumno una mejor integración en el mundo real.
3. Se consigue una asimilación más sencilla y eficiente de las ideas y conceptos, gracias al planteamiento de situaciones que han surgido de la realidad.
4. La sensación de eficiencia del esfuerzo invertido se convierte en un estímulo muy importante para el alumnado, que se traduce en un interés mayor en los aprendizajes y un incremento del tiempo dedicado a trabajar en el curso.



## La metodología universitaria mejor valorada por sus alumnos

Los resultados de este innovador modelo académico son constatables en los niveles de satisfacción global de los egresados de TECH.

La valoración de los estudiantes sobre la calidad docente, calidad de los materiales, estructura del curso y sus objetivos es excelente. No en valde, la institución se convirtió en la universidad mejor valorada por sus alumnos según el índice global score, obteniendo un 4,9 de 5.

*Accede a los contenidos de estudio desde cualquier dispositivo con conexión a Internet (ordenador, tablet, smartphone) gracias a que TECH está al día de la vanguardia tecnológica y pedagógica.*

*Podrás aprender con las ventajas del acceso a entornos simulados de aprendizaje y el planteamiento de aprendizaje por observación, esto es, Learning from an expert.*



Así, en este programa estarán disponibles los mejores materiales educativos, preparados a conciencia:



#### Material de estudio

Todos los contenidos didácticos son creados por los especialistas que van a impartir el curso, específicamente para él, de manera que el desarrollo didáctico sea realmente específico y concreto.

Estos contenidos son aplicados después al formato audiovisual que creará nuestra manera de trabajo online, con las técnicas más novedosas que nos permiten ofrecerte una gran calidad, en cada una de las piezas que pondremos a tu servicio.



#### Prácticas de habilidades y competencias

Realizarás actividades de desarrollo de competencias y habilidades específicas en cada área temática. Prácticas y dinámicas para adquirir y desarrollar las destrezas y habilidades que un especialista precisa desarrollar en el marco de la globalización que vivimos.



#### Resúmenes interactivos

Presentamos los contenidos de manera atractiva y dinámica en píldoras multimedia que incluyen audio, vídeos, imágenes, esquemas y mapas conceptuales con el fin de afianzar el conocimiento.

Este sistema exclusivo educativo para la presentación de contenidos multimedia fue premiado por Microsoft como "Caso de éxito en Europa".



#### Lecturas complementarias

Artículos recientes, documentos de consenso, guías internacionales... En nuestra biblioteca virtual tendrás acceso a todo lo que necesitas para completar tu capacitación.





#### Case Studies

Completarás una selección de los mejores *case studies* de la materia. Casos presentados, analizados y tutorizados por los mejores especialistas del panorama internacional.



#### Testing & Retesting

Evaluamos y reevaluamos periódicamente tu conocimiento a lo largo del programa. Lo hacemos sobre 3 de los 4 niveles de la Pirámide de Miller.



#### Clases magistrales

Existe evidencia científica sobre la utilidad de la observación de terceros expertos. El denominado *Learning from an expert* afianza el conocimiento y el recuerdo, y genera seguridad en nuestras futuras decisiones difíciles.



#### Guías rápidas de actuación

TECH ofrece los contenidos más relevantes del curso en forma de fichas o guías rápidas de actuación. Una manera sintética, práctica y eficaz de ayudar al estudiante a progresar en su aprendizaje.



08

# Titulación

El Máster Título Propio en Programación de Videojuegos garantiza, además de la capacitación más rigurosa y actualizada, el acceso a un título de Máster Propio expedido por TECH Global University.



“

*Supera con éxito este programa y recibe tu titulación universitaria sin desplazamientos ni farragosos trámites”*

Este programa te permitirá obtener el título propio de **Máster en Programación de Videojuegos** avalado por **TECH Global University**, la mayor Universidad digital del mundo.

**TECH Global University**, es una Universidad Oficial Europea reconocida públicamente por el Gobierno de Andorra (*boletín oficial*). Andorra forma parte del Espacio Europeo de Educación Superior (EEES) desde 2003. El EEES es una iniciativa promovida por la Unión Europea que tiene como objetivo organizar el marco formativo internacional y armonizar los sistemas de educación superior de los países miembros de este espacio. El proyecto promueve unos valores comunes, la implementación de herramientas conjuntas y fortaleciendo sus mecanismos de garantía de calidad para potenciar la colaboración y movilidad entre estudiantes, investigadores y académicos.

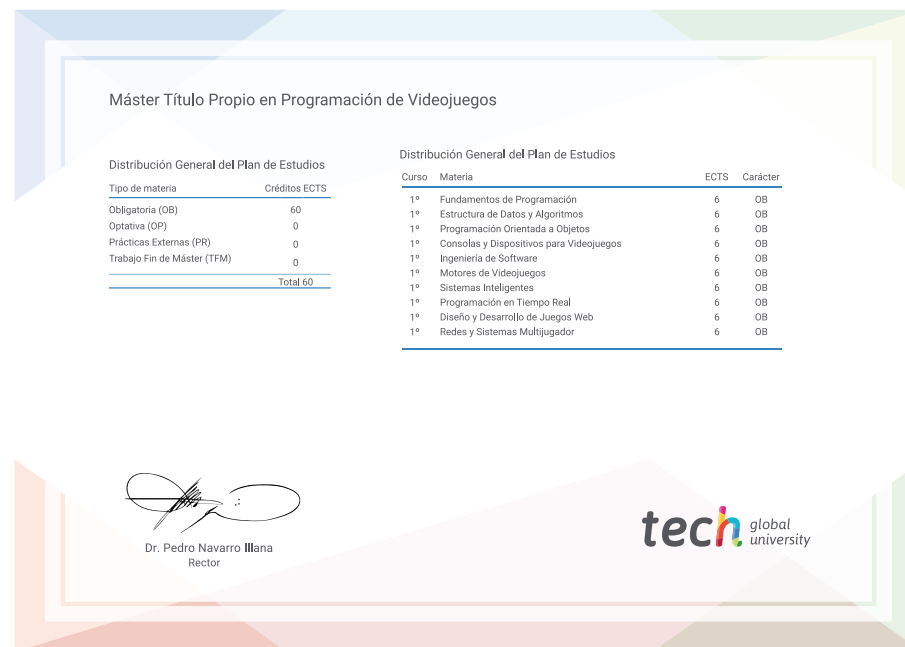
Este título propio de **TECH Global University**, es un programa europeo de formación continua y actualización profesional que garantiza la adquisición de las competencias en su área de conocimiento, confiriendo un alto valor curricular al estudiante que supere el programa.

Título: **Máster Título Propio en Programación de Videojuegos**

Modalidad: **online**

Duración: **12 meses**

Acreditación: **60 ECTS**



\*Apostilla de La Haya. En caso de que el alumno solicite que su título en papel recabe la Apostilla de La Haya, TECH Global University realizará las gestiones oportunas para su obtención, con un coste adicional.



## Máster Título Propio Programación de Videojuegos

- » Modalidad: online
- » Duración: 12 meses
- » Titulación: TECH Global University
- » Acreditación: 60 ECTS
- » Horario: a tu ritmo
- » Exámenes: online

# Máster Título Propio

## Programación de Videojuegos