

# Máster Semipresencial

## Programación de Videojuegos





**tech** corporación universitaria  
UNIMETA

## Máster Semipresencial Programación de Videojuegos

Modalidad: Semipresencial (Online + Prácticas)

Duración: 12 meses

Titulación: TECH Corporación Universitaria UNIMETA

60 + 5 créditos ECTS

Horas lectivas: 1.620 h.

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

# Índice

01

Presentación

---

*pág. 4*

02

¿Por qué cursar este  
Máster Semipresencial?

---

*pág. 8*

03

Objetivos

---

*pág. 12*

04

Competencias

---

*pág. 16*

05

Estructura y contenido

---

*pág. 20*

06

Prácticas

---

*pág. 34*

07

¿Dónde puedo hacer  
las prácticas?

---

*pág. 40*

08

Metodología

---

*pág. 44*

09

Titulación

---

*pág. 52*

# 01

# Presentación

El sector de creación de videojuegos está más que consolidado y atrae cada vez más en el público joven. Uno de los pilares que sostienen un gran título en la industria de los videojuegos es la programación. El procesado que crea las instrucciones básicas y dicta su funcionamiento general son esenciales para que la historia y el desarrollo del juego funcione a la perfección. Esta titulación semipresencial ofrece al profesional del juego una especialización que le permitirá avanzar en su carrera en el sector *Gaming*. Una titulación que cursará 100% online, adaptada a sus necesidades y que cuenta con prácticas en un estudio de creación y desarrollo de videojuegos. Todo ello, le permitirá avanzar en su carrera profesional de la mano de los mejores expertos.





“

*Conviértete en uno de los mejores programadores de la industria de los videojuegos con este Máster Semipresencial”*

La industria del videojuego tiene un gran potencial. El incremento de la demanda y las exigencias de los propios *Gamers* ha llevado a este sector a una carrera por la perfección en sus títulos. El alto nivel de calidad en cada una de las creaciones lleva detrás a un equipo de profesionales de la programación que poseen una excelente cualificación.

Este Máster Semipresencial en Programación de Videojuegos da respuesta a las necesidades actuales del mercado, que reclama a profesionales cada vez más especializados y con un alto grado implicación en las creaciones. La creatividad juega una baza importante, pero sin unos conocimientos sólidos no sería posible obtener unos videojuegos de gran nivel.

Por esta razón, esta titulación ofrece al alumnado un conocimiento exhaustivo de los fundamentos de la programación e ingeniería software, profundiza en la estructura de datos y algoritmo, además de enseñar la programación orientada a objetos y especificaciones de los motore. Asimismo, este programa aborda la programación en tiempo real para ofrecer al profesional de los videojuegos una Máster Semipresencial completo.

Para alcanzar el objetivo de progreso en la carrera profesional de la programación de videojuegos, el alumnado contará con un cuadro docente experto en esta área, que lo guiará y lo tutorizará en todo momento. Además, el contenido interactivo con vídeo resúmenes, casos prácticos y lecturas adicionales complementarán el amplio temario puesto a disposición por TECH en esta titulación 100% online y con prácticas en empresa.

Así, el alumnado podrá completar este recorrido académico con una estancia presencial e intensiva, de 3 semanas de duración, en un estudio puntero en la programación de videojuegos. Un entorno profesional ideal, donde podrá comprobar In Situ, los métodos de trabajo, programas y técnicas más sofisticadas para la creación de títulos de alta calidad. Una oportunidad única, que tan solo te ofrece TECH, la universidad digital más grande del mundo.

Este **Máster Semipresencial en Programación de videojuegos** contiene el programa científico más completo y actualizado del mercado. Sus características más destacadas son:

- ♦ Desarrollo de más de 100 casos de Programación de Videojuegos presentados por profesionales de la programación y profesores universitarios de amplia experiencia en la industria de los videojuegos
- ♦ Sus contenidos gráficos, esquemáticos y eminentemente prácticos con los que están concebidos, recogen una información científica y asistencial sobre aquellas disciplinas médicas indispensables para el ejercicio profesional
- ♦ El desarrollo de casos prácticos presentados por expertos en programación y desarrollo de videojuegos
- ♦ Los contenidos gráficos, esquemáticos y eminentemente prácticos con los que está concebido 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
- ♦ 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
- ♦ Todo esto se complementará con lecciones teóricas, preguntas al experto, foros de discusión de temas controvertidos y trabajos de reflexión individual
- ♦ Disponibilidad de los contenidos desde cualquier dispositivo fijo o portátil con conexión a internet
- ♦ Además, podrás realizar una estancia de prácticas clínicas en uno de los mejores centros hospitalarios

“

*Este programa 100% online te brinda la oportunidad de realizar prácticas en un estudio y ponerte a prueba con los mejores programadores”*

En esta propuesta de Máster, de carácter profesionalizante y modalidad semipresencial, el programa está dirigido a la actualización de profesionales de los videojuegos que desarrollan sus funciones en grandes estudios creativos, y que requieren un alto nivel de cualificación. Los contenidos están basados en la última evidencia científica, y orientados de manera didáctica para integrar el saber teórico en la práctica de la industria *gaming*, y los elementos teórico-prácticos facilitarán la actualización del conocimiento y permitirán la toma de decisiones en la Programación de Videojuegos.

Gracias a su contenido multimedia elaborado con la última tecnología educativa, permitirán al profesional de los videojuegos un aprendizaje situado y contextual, es decir, un entorno simulado que proporcionará un aprendizaje inmersivo programado para entrenarse ante situaciones reales. El diseño de este programa se centra en el Aprendizaje Basado en Problemas, mediante el cual deberá tratar de resolver las distintas situaciones de práctica profesional que se le planteen a lo largo del mismo. Para ello, contará con la ayuda de un novedoso sistema de vídeo interactivo realizado por reconocidos expertos.

*Este Máster Semipresencial te permitirá ser un referente en el área de la programación de videojuegos. Inscríbete ya.*

*Desarrolla aplicaciones de manera eficiente aplicadas a motores de videojuegos gracias a este Máster Semipresencial.*



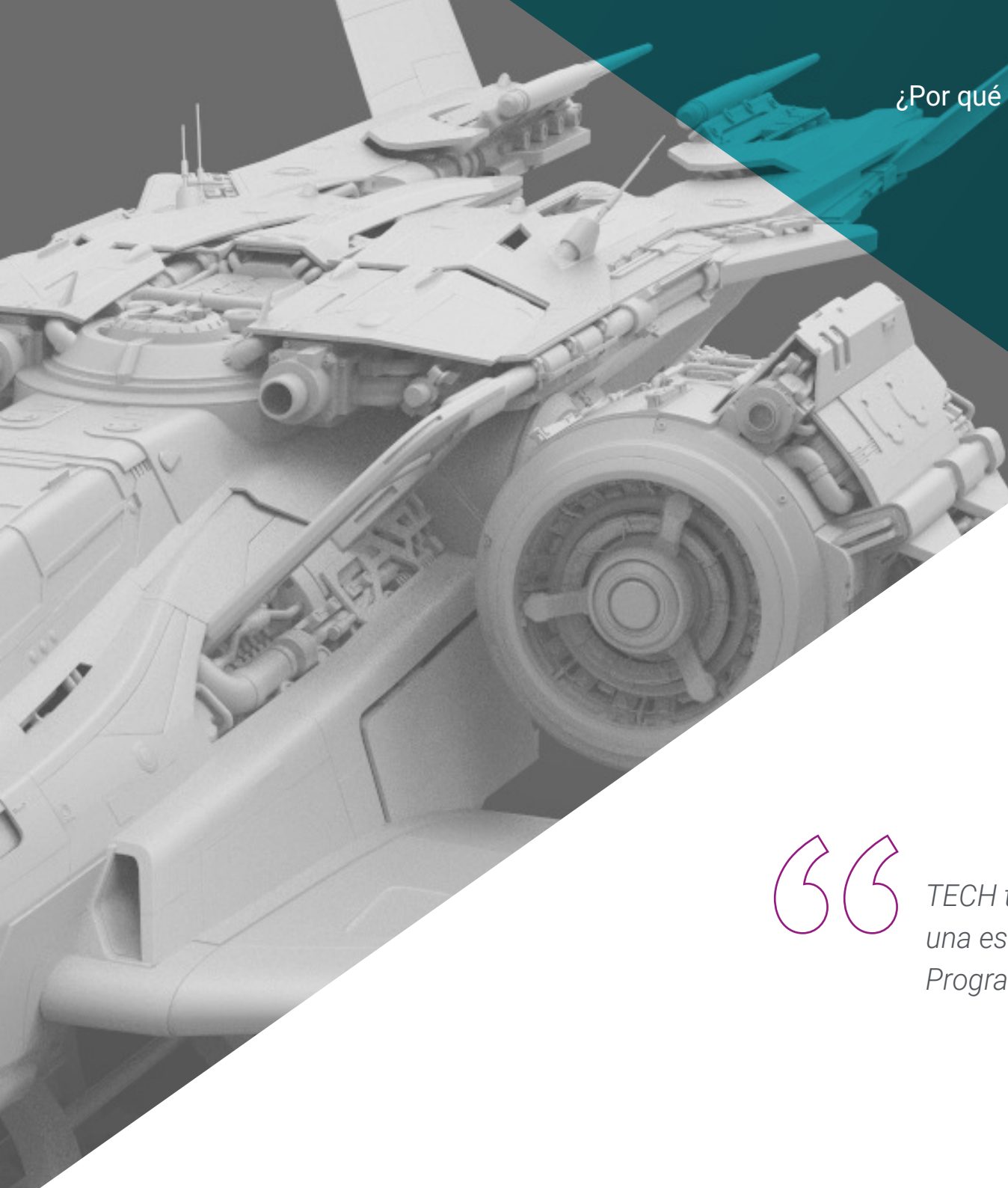
# 02

## ¿Por qué cursar este Máster Semipresencial?

La programación requiere de unos conocimientos teóricos sólidos, pero sin duda, la práctica diaria es la que va perfeccionando la maestría del profesional en el sector de los videojuegos. Es por ello, por lo que TECH ha creado esta titulación, que combina a la perfección el temario online más avanzado en áreas como el diseño de algoritmos, los sistemas inteligentes y la programación en tiempo real con una estancia práctica en un estudio de programación puntero. De esta manera, el alumnado obtendrá una visión mucho más completa de la Programación de Videojuegos, de los programas empleados y de las técnicas más recientes. Todo ello, siempre guiado en todo momento por los mejores expertos en esta materia.







“

*TECH te da la oportunidad de poder cursar una estancia práctica eficaz e intensiva en Programación de videojuegos”*

### 1. Actualizarse a partir de la última tecnología disponible

La tecnología ha revolucionado en los últimos años el área de Programación de Videojuegos, favoreciendo la creación de títulos de mayor calidad y realismo. Por ello, y con el fin de poder acercar al alumnado a esta tecnología, TECH ha creado este Máster Semipresencial, donde el profesional se adentrará en los motores de videojuegos, los desafíos tecnológicos actuales y los softwares para la creación de ontologías. De esta manera logrará una puesta al día en la última tecnología disponible.

### 2. Profundizar a partir de la experiencia de los mejores especialistas

En este recorrido académico, el alumnado estará siempre guiado por los mejores especialistas en Programación de Videojuegos. Así, el egresado contará con un excelente equipo docente con amplia experiencia en el sector, que lo acompañará durante la fase teórica, mientras que, en la estancia presencial, estará junto a auténticos expertos, que forman parte del equipo del estudio donde efectuará la fase práctica.

### 3. Adentrarse en entornos clínicos de primera

TECH lleva a cabo un riguroso proceso de selección de todos los estudios y empresas donde se realizan las estancias prácticas. De esta manera, el alumnado tiene garantizado el acceso a un entorno profesional de primer nivel en Programación de Videojuegos. De esta manera, podrá comprobar de primera mano, cómo es el día a día de un programador especializado, las técnicas y métodos empleados para lograr títulos de calidad.



#### 4. Combinar la mejor teoría con la práctica más avanzada

TECH con esta titulación se adapta al quehacer diario de los programadores y es por ello, por lo que adopta el enfoque teórico-práctico, alejado, además, de las largas horas de estudio, para centrarse en los conceptos claves. Esta institución académica ofrece así un modelo de aprendizaje innovador que permite al alumnado adquirir los conocimientos necesarios para ponerse al frente de la programación de videojuegos de alta calidad.

#### 5. Expandir las fronteras del conocimiento

TECH ofrece las posibilidades de realizar esta Capacitación Práctica no solo en centros de envergadura nacional, sino también internacional. De esta forma, el especialista podrá expandir sus fronteras y ponerse al día con los mejores profesionales, que ejercen en hospitales de primera categoría y en diferentes continentes. Una oportunidad única que solo TECH, la universidad digital más grande del mundo, podría ofrecer.



*Tendrás una inmersión práctica total en el centro que tú mismo elijas”*

# 03

## Objetivos

El diseño del programa de este Máster Semipresencial permitirá al alumnado adquirir las competencias necesarias avanzar en una programación altamente competitiva y que requiere de un personal cualificado. Por tanto, el objetivo principal de esta enseñanza es lograr que el alumnado se convierte en un excelente desarrollador de videojuegos a partir de todas las herramientas que esta enseñanza le aporta. Para lograrlo, esta titulación facilita un contenido multimedia actualizado con lecturas adicionales y un sistema de aprendizaje *Relearning*, basado en la reiteración de contenido, que facilitará la cimentación de conceptos.





“

*¿Quieres poner en marcha un juego multijugador?  
Aprende a transformar tu idea en lenguaje de  
programación real para este tipo de videojuegos”*



## Objetivo general

---

- Lograr que el alumnado conozca los diferentes lenguajes y métodos de programación aplicados al videojuego. Para ello se profundizará en el proceso de producción de un título y su integración en las diferentes etapas. Asimismo, el profesional de los videojuegos aprenderá los fundamentos del diseño de videojuego y los principales conocimientos teóricos. Además, al finalizar esta enseñanza será capaz de entender el papel de la programación y desarrollar videojuegos web y multijugador



*Este programa te permitirá avanzar profesionalmente. Serás capaz de dominar cualquier estructura de datos y algoritmos con este Máster Semipresencial"*



## 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
- Analizar los elementos esenciales de un programa informático, como son los distintos tipos de datos, operadores, expresiones, sentencias, E/S y sentencias de control
- Interpretar algoritmos, que son la base necesaria para poder desarrollar programas informáticos

### Módulo 2. Estructura de datos y algoritmos

- Aprender 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

### 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

#### **Módulo 4. Consolas y dispositivos para videojuegos**

- ♦ Saber el funcionamiento básico de los principales periféricos de entrada y salida
- ♦ Entender las principales implicaciones de diseño de las diferentes plataformas
- ♦ Estudiar 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 y los distintos modelos para su desarrollo incluyendo tecnologías ágiles
- ♦ Reconocer la ingeniería de requisitos, su desarrollo, elaboración, negociación y validación a fin de entender las principales normas relativas a la calidad del software y a la administración de proyectos

#### **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
- ♦ Programar 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

#### **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

#### **Módulo 9. Diseño y desarrollo de juegos web**

- ♦ Ser capaz de 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

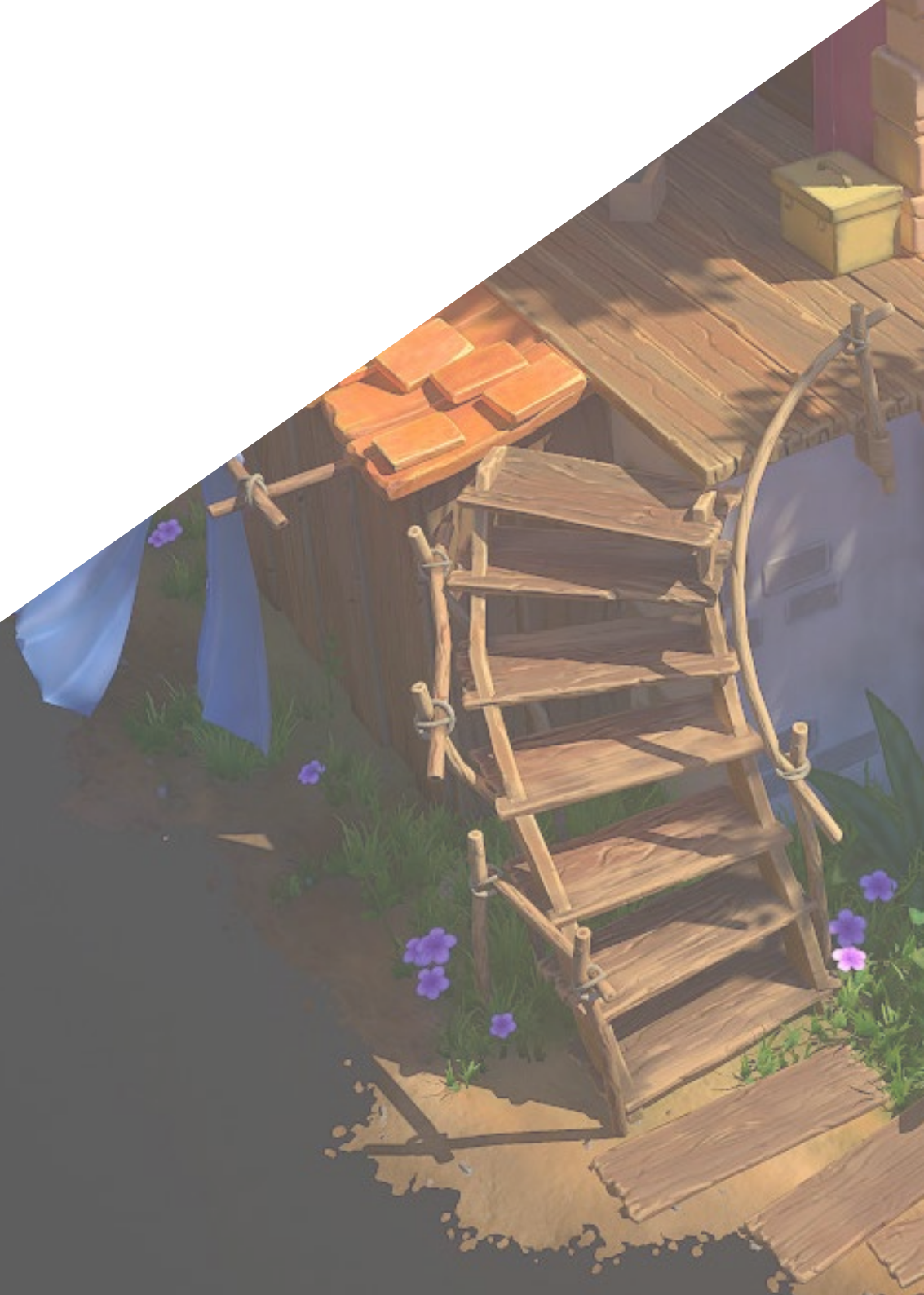
#### **Módulo 10. Redes y sistemas multijugador**

- ♦ Describir la arquitectura del protocolo de control de transmisión/protocolo de Internet (TCP/IP) y el funcionamiento básico de las redes inalámbricas
- ♦ Analizando la seguridad aplicada a videojuegos
- ♦ Adquirir la capacidad para desarrollar juegos en línea para múltiples jugadores

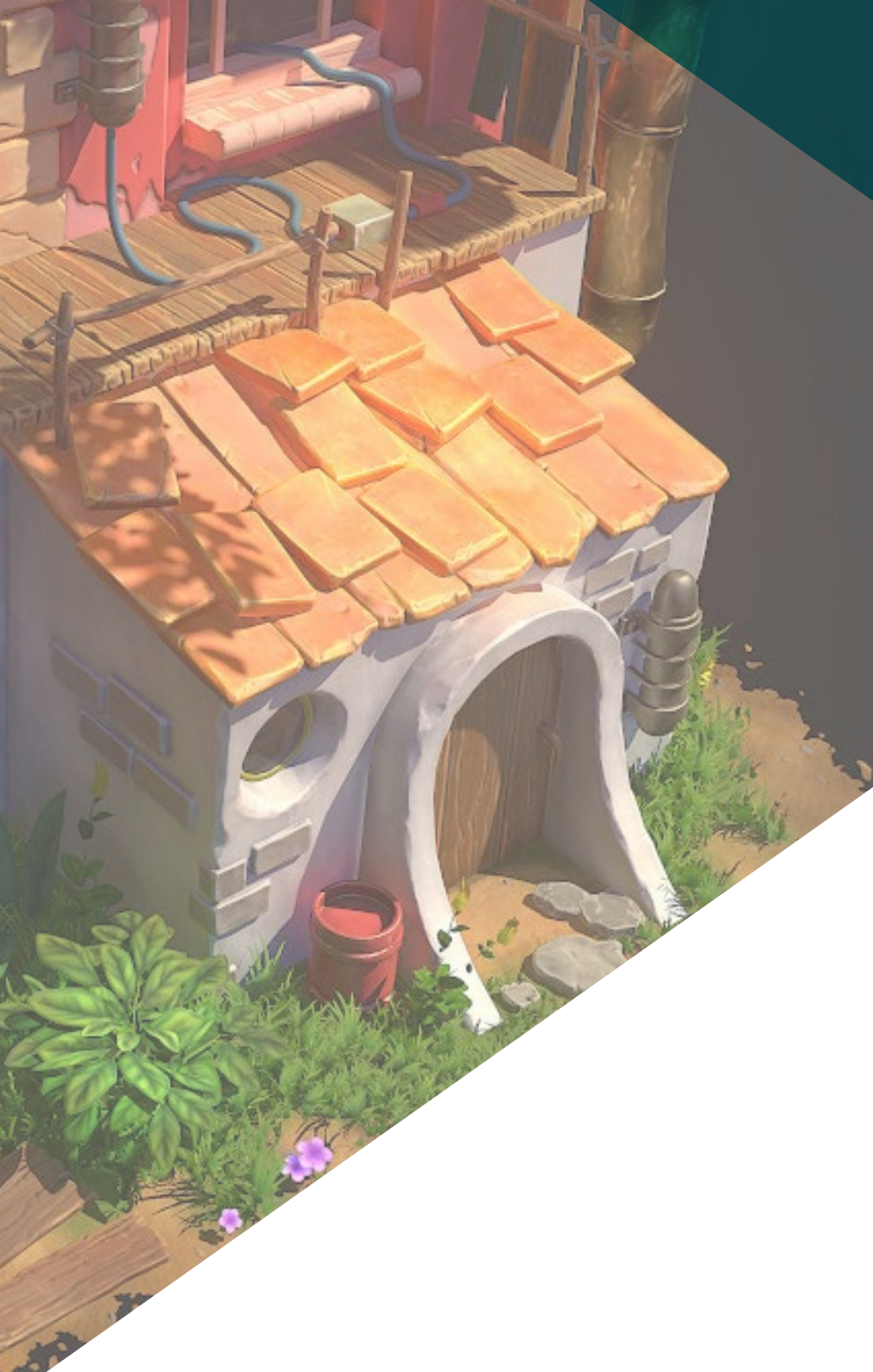
# 04

# Competencias

El profesional de los videojuegos que curse este Máster Semipresencial adquirirá una serie de habilidades que le permitirá incorporarse a cualquier tipo de estudio desarrollador de videojuegos. Así, el alumnado al finalizar esta enseñanza tendrá las competencias necesarias para programar en diferentes lenguajes empleados en la industria *gaming* y adquirirá las capacitaciones esenciales para ejecutar un proyecto de principio a fin en distintas plataformas y motores de videojuegos.







“

*Adquiere las habilidades necesarias para programar videojuegos para Xbox One, PS4 y Wii U-Switch”*



## Competencias generales

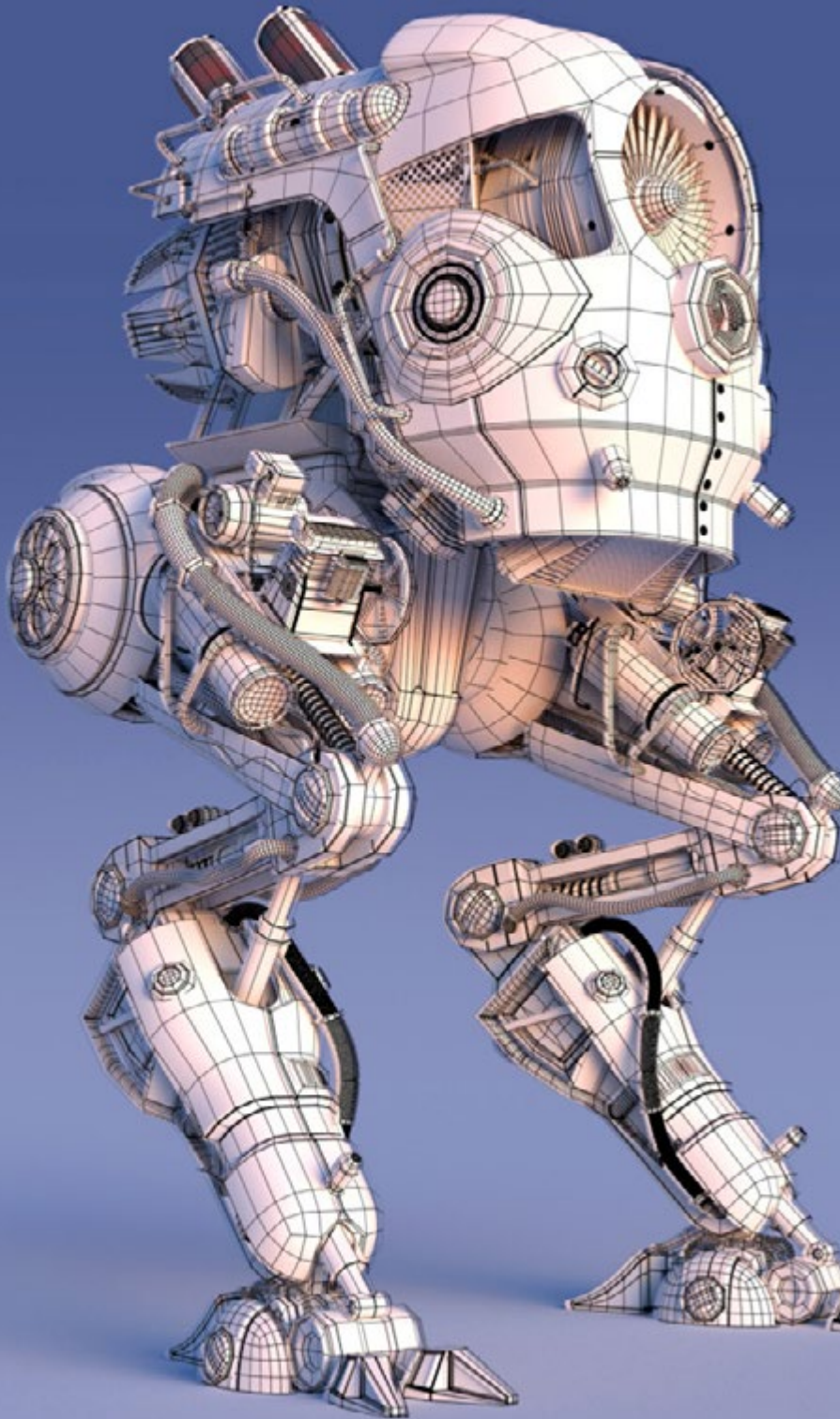
---

- ♦ Diseñar todas las fases de un videojuego, desde la idea inicial hasta el lanzamiento final
- ♦ Especializarse como programador de videojuegos
- ♦ Profundizar en todas las partes del desarrollo, desde la arquitectura inicial, la programación del personaje jugador y de todos los elementos que intervienen en el proceso de juego
- ♦ Obtener una visión de conjunto del proyecto, pudiendo aportar soluciones a las diferentes problemáticas y retos que surjan en el diseño de un videojuego

“

*Antes de programar conoce la experiencia del gamer y analiza la jugabilidad del videojuego. Con esta Máster Semipresencial aprenderás a alcanzar el éxito”*





## Competencias específicas

---

- Conocer el software necesario para ser un profesional del desarrollo de videojuegos
- Comprender la experiencia del jugador y saber analizar la jugabilidad del videojuego
- Entender todo el procedimiento teórico y práctico del proceso de programación de un videojuego
- Dominar los lenguajes de programación más útiles para el mundo del videojuego
- Integrar la programación aprendida a diferentes tipos de consolas y plataformas
- Programar videojuegos web y multijugador
- Asimilar el concepto de motor de videojuegos para poder programar de forma correcta
- Aplicar conocimientos de ingeniería de software a la programación de videojuegos

# 05

## Estructura y contenido

Este Máster Semipresencial en Programación de Videojuegos ofrece al alumnado un completo y amplio contenido en desarrollo de videojuegos estructurado en 10 módulos especializados. El cuerpo docente que integra esta titulación podrá profundizar en los fundamentos de la programación y adquirir a la vez un conocimiento actualizado y orientado a la práctica. De esta forma, el alumnado recibe una enseñanza de calidad, con contenido novedoso en formato multimedia y descargable para ser consultado en cualquier momento. La capacitación teórica 100% online le concede libertad para distribuir la carga lectiva en función de sus necesidades.





“

*Un plan de estudio diseñado para ofrecerte el contenido más avanzado y actual sobre la Programación de videojuegos”*

## 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 usuarios
  - 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 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 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. Periodo Atari (1977-1985)
  - 4.1.2. Periodo NES y SNES (1985-1995)
  - 4.1.3. Periodo PlayStation/PlayStation 2 (1995-2005)
  - 4.1.4. Periodo Xbox 360, PS3 y Wii (2005-2013)
  - 4.1.5. Periodo 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 TICs
  - 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 Game Maker*
  - 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
- 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
  - 6.5.4. Animaciones
  - 6.5.5. Sistema de partículas
  - 6.5.6. Inteligencia artificial
  - 6.5.7. FPS



- 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. OWL
  - 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 razonador
  - 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-effectos
- 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. CCS3
- 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 computadores 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



*Gracias al método Relearning, basado en la reiteración de conceptos claves, podrás disminuir las largas horas de estudio”*

# 06 Prácticas

Una vez el profesional de los videojuegos concluya la enseñanza teórica online, esta titulación contempla la realización de prácticas en un estudio creativo y desarrollador de videojuego. De esta forma, el alumnado podrá entrar en contacto directamente con equipos desarrolladores del sector y podrá demostrar todo lo aprendido. Durante esta etapa de aprendizaje el estudiante contará con un tutor que le acompañará hasta concluir esta enseñanza.



“

*Siéntate junto a grandes desarrolladores y trabaja en equipo durante la fase práctica de esta enseñanza”*

El periodo de Capacitación Práctica de este programa de Programación de Videojuegos tiene una duración de 3 semanas. El alumnado acudirá al estudio creativo y desarrollador de videojuegos de lunes a viernes con una jornada de 8 horas consecutivas de enseñanza práctica de la mano de un especialista.

Esta fase práctica le permitirá al alumnado desplegar todos sus conocimientos sobre programación adquiridos en este Máster Semipresencial. Junto a otros profesionales de esta empresa será capaz de coordinarse con el resto del equipo y aprender las principales herramientas y software de programación. De esta forma, obtiene un aprendizaje pegado a la realidad de la industria de los videojuegos.

Durante la estancia en este estudio, el profesional no sólo empleará el habitual lenguaje de programación, sino que podrá diseñar videojuegos multijugador. Una categoría de videojuegos que va en aumento y que fascina al *Gamers*.

La parte práctica se realizará con la participación activa del estudiante desempeñando las actividades y procedimientos de cada área de competencia (aprender a aprender y aprender a hacer), con el acompañamiento y guía de los profesores y demás compañeros de entrenamiento que faciliten el trabajo en equipo y la integración multidisciplinar como competencias transversales para la praxis de programación de videojuegos (aprender a ser y aprender a relacionarse).

“

*Capacítate en un estudio que te brindará la oportunidad de demostrar tu potencial como programador de videojuegos”*



Los procedimientos descritos a continuación serán la base de la parte práctica de la capacitación, y su realización estará sujeta a la disponibilidad propia del centro y su volumen de trabajo, siendo las actividades propuestas las siguientes:

Módulo	Actividad Práctica	Cantidad
Creación de estructura de datos y algoritmos	Llevar a cabo la técnica de <i>Backtracking</i> , para la creación de datos y algoritmos	1
	Participar en el análisis de los algoritmos para obtener mayor eficiencia	
	Realizar tareas de medición del tamaño de la entrada y de tiempo de ejecución	
	Crear algoritmos de ordenación con árboles, con <i>Heaps</i> , con grafos y con <i>Greedy</i>	
Programación orientada a objetos	Emplear el <i>Patrón Factory</i> , <i>Patrón Singleton</i> , <i>Patrón Observer</i> y <i>Patrón Composite</i> en la creación de objetos	1
	Crear, capturar y gestionar excepciones en la creación de objetos	
	Realizar programación concurrente	
	Emplear mecanismos de bloqueo y de comunicación	
	Crear documentación y pruebas de software	
Programación en tiempo real	Crear y sincronizar <i>Threads</i>	1
	Programar <i>Shaders</i>	
	Implementar el patrón en Java y usar bibliotecas Java.	
	Crear post-efectos	
	Procesar interrupciones, sincronización y comunicación entre procesos	
Diseño y desarrollo de juegos web	Programar la web con formas HTML	1
	Usar DOCTYPE e innerHTML para desarrollar juegos web	
	Hacer uso de PlayCanvas para desarrollar juegos web	
	Configurar el proyecto de diseño y desarrollo de juego web	

## Seguro de responsabilidad civil

La máxima preocupación de esta institución es garantizar la seguridad tanto de los profesionales en prácticas como de los demás agentes colaboradores necesarios en los procesos de capacitación práctica en la empresa. Dentro de las medidas dedicadas a lograrlo, se encuentra la respuesta ante cualquier incidente que pudiera ocurrir durante todo el proceso de enseñanza-aprendizaje.

Para ello, esta entidad educativa se compromete a contratar un seguro de responsabilidad civil que cubra cualquier eventualidad que pudiera surgir durante el desarrollo de la estancia en el centro de prácticas.

Esta póliza de responsabilidad civil de los profesionales en prácticas tendrá coberturas amplias y quedará suscrita de forma previa al inicio del periodo de la capacitación práctica. De esta forma el profesional no tendrá que preocuparse en caso de tener que afrontar una situación inesperada y estará cubierto hasta que termine el programa práctico en el centro.



## Condiciones Generales de la Capacitación Práctica

Las condiciones generales del acuerdo de prácticas para el programa serán las siguientes:

**1. TUTORÍA:** durante el Máster Semipresencial el alumno tendrá asignados dos tutores que le acompañarán durante todo el proceso, resolviendo las dudas y cuestiones que pudieran surgir. Por un lado, habrá un tutor profesional perteneciente al centro de prácticas que tendrá como fin orientar y apoyar al alumno en todo momento. Por otro lado, también tendrá asignado un tutor académico cuya misión será la de coordinar y ayudar al alumno durante todo el proceso resolviendo dudas y facilitando todo aquello que pudiera necesitar. De este modo, el profesional estará acompañado en todo momento y podrá consultar las dudas que le surjan, tanto de índole práctica como académica.

**2. DURACIÓN:** el programa de prácticas tendrá una duración de tres semanas continuadas de formación práctica, distribuidas en jornadas de 8 horas y cinco días a la semana. Los días de asistencia y el horario serán responsabilidad del centro, informando al profesional debidamente y de forma previa, con suficiente tiempo de antelación para favorecer su organización.

**3. INASISTENCIA:** en caso de no presentarse el día del inicio del Máster Semipresencial, el alumno perderá el derecho a la misma sin posibilidad de reembolso o cambio de fechas. La ausencia durante más de dos días a las prácticas sin causa justificada/médica, supondrá la renuncia las prácticas y, por tanto, su finalización automática. Cualquier problema que aparezca durante el transcurso de la estancia se tendrá que informar debidamente y de forma urgente al tutor académico.

**4. CERTIFICACIÓN:** el alumno que supere el Máster Semipresencial recibirá un certificado que le acreditará la estancia en el centro en cuestión.

**5. RELACIÓN LABORAL:** el Máster Semipresencial no constituirá una relación laboral de ningún tipo.

**6. ESTUDIOS PREVIOS:** algunos centros podrán requerir certificado de estudios previos para la realización del Máster Semipresencial. En estos casos, será necesario presentarlo al departamento de prácticas de TECH para que se pueda confirmar la asignación del centro elegido.

**7. NO INCLUYE:** el Máster Semipresencial no incluirá ningún elemento no descrito en las presentes condiciones. Por tanto, no incluye alojamiento, transporte hasta la ciudad donde se realicen las prácticas, visados o cualquier otra prestación no descrita.

No obstante, el alumno podrá consultar con su tutor académico cualquier duda o recomendación al respecto. Este le brindará toda la información que fuera necesaria para facilitarle los trámites.

07

# ¿Dónde puedo hacer las Prácticas?

Esta titulación de Máster Semipresencial contempla en una estancia práctica en un estudio referente donde el alumnado pondrá en práctica todo lo aprendido sobre programación de videojuegos. De esta forma, TECH brinda la oportunidad al alumnado de acercarse a profesionales del videojuego que deseen adquirir una enseñanza de calidad y acorde a las demandas actuales del sector.





“

*Alcanza tus metas con aplicando todos tus conocimientos en un estudio referente en el sector de los videojuegos”*



El alumno podrá cursar la parte práctica de este Máster Semipresencial en los siguientes centros:



**Videojuegos**

**Startreming Games**

País	Ciudad
Argentina	Mendoza

Dirección: Ruta 160 esquina Buenos Aires 88, San Rafael, Mendoza

Estudio de desarrollo de videojuegos independiente y con trabajo remoto

---

**Capacitaciones prácticas relacionadas:**

- Programación de Videojuegos
- Modelado 3D Hard Surface

**Programación de Videojuegos**





“

*Aprovecha la oportunidad para conocer los últimos avances en esta materia para aplicarla a tu práctica diaria”*

08

# Metodología

Este programa de capacitación ofrece una forma diferente de aprender. Nuestra metodología se desarrolla a través de un modo de aprendizaje de forma cíclica: **el Relearning**.

Este sistema de enseñanza es utilizado, por ejemplo, en las facultades de medicina más prestigiosas del mundo y se ha considerado uno de los más eficaces por publicaciones de gran relevancia como el **New England Journal of Medicine**.





“

*Descubre el Relearning, un sistema que abandona el aprendizaje lineal convencional para llevarte a través de sistemas cíclicos de enseñanza: una forma de aprender que ha demostrado su enorme eficacia, especialmente en las materias que requieren memorización”*

## Estudio de Caso para contextualizar todo el contenido

Nuestro programa ofrece un método revolucionario de desarrollo de habilidades y conocimientos. Nuestro objetivo es afianzar competencias en un contexto cambiante, competitivo y de alta exigencia.

“

*Con TECH podrás experimentar una forma de aprender que está moviendo los cimientos de las universidades tradicionales de todo el mundo”*



*Accederás a un sistema de aprendizaje basado en la reiteración, con una enseñanza natural y progresiva a lo largo de todo el temario.*



*El alumno aprenderá, mediante actividades colaborativas y casos reales, la resolución de situaciones complejas en entornos empresariales reales.*

## Un método de aprendizaje innovador y diferente

El presente programa de TECH es una enseñanza intensiva, creada desde 0, que propone los retos y decisiones más exigentes en este campo, ya sea en el ámbito nacional o internacional. Gracias a esta metodología se impulsa el crecimiento personal y profesional, dando un paso decisivo para conseguir el éxito. El método del caso, técnica que sienta las bases de este contenido, garantiza que se sigue la realidad económica, social y profesional más vigente.

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

El método del caso ha sido el sistema de aprendizaje más utilizado por las mejores escuelas de negocios del mundo desde que éstas existen. Desarrollado en 1912 para que los estudiantes de Derecho no solo aprendiesen las leyes a base de contenidos teóricos, el método del caso consistió en presentarles situaciones complejas reales para que tomaran decisiones y emitieran juicios de valor fundamentados sobre cómo resolverlas.

En 1924 se estableció como método estándar de enseñanza en Harvard.

Ante una determinada situación, ¿qué debería hacer un profesional? Esta es la pregunta a la que te enfrentamos en el método del caso, un método de aprendizaje orientado a la acción. A lo largo de 4 años, te enfrentarás a múltiples casos reales. Deberás integrar todos tus conocimientos, investigar, argumentar y defender tus ideas y decisiones.

## Relearning Methodology

TECH aúna de forma eficaz la metodología del Estudio de Caso con un sistema de aprendizaje 100% online basado en la reiteración, que combina 8 elementos didácticos diferentes en cada lección.

Potenciamos el Estudio de Caso con el mejor método de enseñanza 100% online: el Relearning.

*En 2019, obtuvimos los mejores resultados de aprendizaje de todas las universidades online en español en el mundo.*

En TECH aprenderás con una metodología vanguardista concebida para capacitar a los directivos del futuro. Este método, a la vanguardia pedagógica mundial, se denomina Relearning.

Nuestra universidad es la única en habla hispana licenciada para emplear este exitoso método. En 2019, conseguimos mejorar los niveles de satisfacción global de nuestros alumnos (calidad docente, calidad de los materiales, estructura del curso, objetivos...) con respecto a los indicadores de la mejor universidad online en español.





En nuestro programa, el aprendizaje no es un proceso lineal, sino que sucede en espiral (aprender, desaprender, olvidar y reaprender). Por eso, se combinan cada uno de estos elementos de forma concéntrica. Con esta metodología se han capacitado más de 650.000 graduados universitarios con un éxito sin precedentes en ámbitos tan distintos como la bioquímica, la genética, la cirugía, el derecho internacional, las habilidades directivas, las ciencias del deporte, la filosofía, el derecho, la ingeniería, el periodismo, la historia o los mercados e instrumentos financieros. Todo ello en un entorno de alta exigencia, con un alumnado universitario de un perfil socioeconómico alto y una media de edad de 43,5 años.

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

A partir de la última evidencia científica en el ámbito de la neurociencia, no solo sabemos organizar la información, las ideas, las imágenes y los recuerdos, sino que sabemos que el lugar y el contexto donde hemos aprendido algo es fundamental para que seamos capaces de recordarlo y almacenarlo en el hipocampo, para retenerlo en nuestra memoria a largo plazo.

De esta manera, y en lo que se denomina Neurocognitive context-dependent e-learning, los diferentes elementos de nuestro programa están conectados con el contexto donde el participante desarrolla su práctica profesional.

Este programa ofrece los mejores materiales educativos, preparados a conciencia para los profesionales:



#### 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, para crear el método de trabajo online de TECH. Todo ello, con las técnicas más novedosas que ofrecen piezas de gran calidad en todos y cada uno los materiales que se ponen a disposición del alumno.



#### 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 las futuras decisiones difíciles.



#### Prácticas de habilidades y competencias

Realizarán 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.



#### Lecturas complementarias

Artículos recientes, documentos de consenso y guías internacionales, entre otros. En la biblioteca virtual de TECH el estudiante tendrá acceso a todo lo que necesita para completar su capacitación.





#### Case studies

Completarán una selección de los mejores casos de estudio elegidos expresamente para esta titulación. Casos presentados, analizados y tutorizados por los mejores especialistas del panorama internacional.



#### Resúmenes interactivos

El equipo de TECH presenta los contenidos de manera atractiva y dinámica en píldoras multimedia que incluyen audios, vídeos, imágenes, esquemas y mapas conceptuales con el fin de afianzar el conocimiento.

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



#### Testing & Retesting

Se evalúan y reevalúan periódicamente los conocimientos del alumno a lo largo del programa, mediante actividades y ejercicios evaluativos y autoevaluativos para que, de esta manera, el estudiante compruebe cómo va consiguiendo sus metas.



09

# Titulación

El Máster Semipresencial en Programación de Videojuegos garantiza, además de la capacitación más rigurosa y actualizada, el acceso a dos diplomas de Máster Semipresencial, uno expedido por TECH Global University y otro expedido por la Corporación Universitaria del Meta.



“

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

El programa del **Máster Semipresencial en Programación de Videojuegos** es el más completo del panorama académico actual. A su egreso, el estudiante recibirá un diploma universitario emitido por TECH Global University, y otro por la Corporación Universitaria del Meta.

Estos títulos de formación permanente y actualización profesional de TECH Global University y Corporación Universitaria del Meta garantizan la adquisición de competencias en el área de conocimiento, otorgando un alto valor curricular al estudiante que supere las evaluaciones y acredite el programa tras cursarlo en su totalidad.

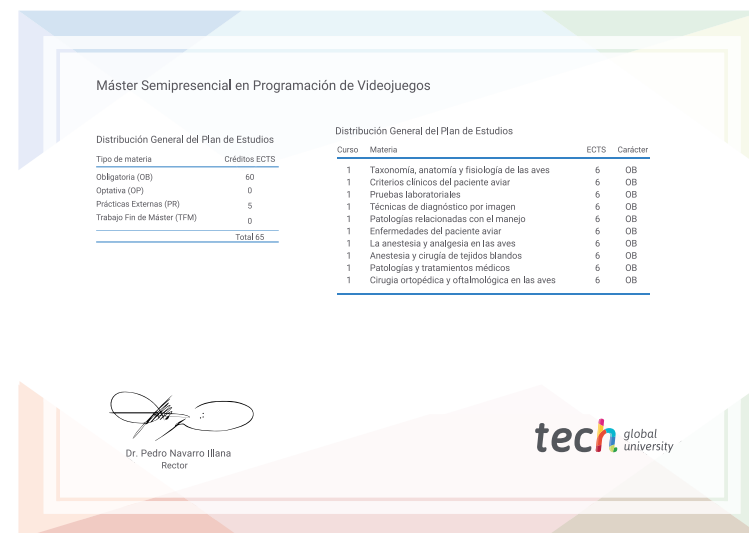
Este doble reconocimiento, de dos destacadas instituciones universitarias, suponen una doble recompensa a una formación integral y de calidad, asegurando que el estudiante obtenga una certificación reconocida tanto a nivel nacional como internacional. Este mérito académico le posicionará como un profesional altamente capacitado y preparado para enfrentar los retos y demandas en su área profesional.

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

Modalidad: **Semipresencial (Online + Prácticas)**

Duración: **12 meses**

Acreditación: **60 + 5 créditos ECTS**



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



## Máster Semipresencial Programación de Videojuegos

Modalidad: Semipresencial (Online + Prácticas)

Duración: 12 meses

Titulación: TECH Corporación Universitaria UNIMETA

60 + 5 créditos ECTS

Horas lectivas: 1.620 h.

# Máster Semipresencial

## Programación de Videojuegos

