Trayecto Programador júnior

# Misión 3: ¿Cómo administrar el flujo y los datos de la Escena?



## Descripción del trayecto

Diseñado para cualquier persona interesada en aprender a programar u obtener un rol de nivel inicial en Unity, este curso incluye un conocimiento básico de Unity y no tiene requisitos previos de matemáticas. Programador júnior te prepara para obtener una certificación de Unity para que puedas demostrar tu capacidad de trabajo a los empleadores.

**Detalles clave**

* Esta misión te tomará aproximadamente 14 horas para completarla. Hazla a tu propio ritmo. Recibirás experiencia a cada paso durante tu camino.
* Conecta con la comunidad de Unity mientras aprendes y consultas el calendario Learn Live para seguir sesiones prácticas con creadores consolidados de Unity.
* Cuando hayas finalizado la misión, obtendrás una insignia para lucir en tu perfil y tu portafolio.

**Habilidades abordadas en este curso**

Comprensión del código básico

* ¿Cómo interpretar código simple?
* Mejora el código simple utilizando las características de un IDE

Creación de *scripts* de aplicación básica

* Utiliza estructuras lógicas comunes para controlar la ejecución del código.
* Escribe código que utilice las distintas API de Unity.
* Implementa tipos de datos adecuados
* Escribe código que se integre con un sistema existente
* Implementa un estilo de código que sea eficiente y fácil de leer
* Nuevos conceptos de prototipos:

Depuración básica

* Diagnostica y corrige código que compila, pero no logra rendir según lo esperado.
* Diagnostica y corrige errores comunes de compilación.
* Diagnostica y corrige errores de compilación relacionados con la API de creación de *scripts* de Unity
* Diagnostica y corrige la causa de una excepción

Creación de *scripts* de aplicación para principiantes

* ¿Cómo crear el flujo de la Escena en el estado de una aplicación?
* ¿Cómo implementar la continuación de datos entre escenas y sesiones de usuario?
* Nivel 1: Control de versiones
* ¿Cómo mantener un proyecto al implementar correctamente el control de versiones?
* ¿Cómo implementar las prácticas recomendadas del control de versiones con Unity Collaborate?

Optimización de código básico

* Maximiza la eficiencia del código al ejecutar de manera correcta las prácticas recomendadas de la creación de código
* Depura problemas de rendimiento

Teoría de la programación para principiantes

* Analiza los pilares fundamentales de la programación orientada a objetos
* Simplifica el código y hazlo reutilizable al implementar correctamente los principios de herencia y polimorfismo
* Aumenta la seguridad del código y su usabilidad al implementar de forma correcta los principios de abstracción y encapsulación, incluido el uso de interfaces
* Escribe código eficiente, organizado y comprensible al implementar de forma correcta los principios de la programación orientada a objetos

| **¿Cómo usar el trayecto?** | | |
| --- | --- | --- |
| El trayecto Aspectos básicos de Unity se divide en 3 «misiones», cada una con muchos tutoriales y evaluaciones. Las misiones siguientes forman todo el trayecto: | | |
|  | **Programador júnior: ¿Cómo crear con código 1?** | 13 horas 45 minutos |
|  | **Programador júnior: ¿Cómo crear con código 2?** | 24 horas 15 minutos |
|  | **Programador júnior: ¿Cómo administrar el flujo y los datos de la Escena?** | 2 horas |
|  | **Programador júnior: ¿Cómo aplicar los principios orientados a objetos?** | 1 hora 45 minutos |
| Instamos a los estudiantes a completar todas las misiones en el orden correcto para garantizar una mejor experiencia de aprendizaje. | | |

# 

# 

# 

# 

# Índice

[Misión 3: ¿Cómo administrar el flujo y los datos de la Escena?1](#_heading=h.x82pqsn8rube)

[Descripción del trayecto](#_heading=h.1fob9te) [1](#_heading=h.1fob9te)

[Índice](#_heading=h.qsj6fbui9phm) [2](#_heading=h.qsj6fbui9phm)

[Misión 3: ¿Cómo administrar el flujo y los datos de la Escena?](#_heading=h.lpoonn66x6js) [3](#_heading=h.lpoonn66x6js)

[Introducción al diseño de experiencias 3D en tiempo real](#_heading=h.ep4x6w3z1v2s) [3](#_heading=h.ep4x6w3z1v2s)

[¿Cómo configurar el control de versiones?](#_heading=h.7xys0m5mjzn6) [4](#_heading=h.7xys0m5mjzn6)

[Principios de la programación orientada a objetos](#_heading=h.3apovlmf2jl) [5](#_heading=h.3apovlmf2jl)

[¿Cómo crear un flujo de Escena?](#_heading=h.gjjzk3430xc) [6](#_heading=h.gjjzk3430xc)

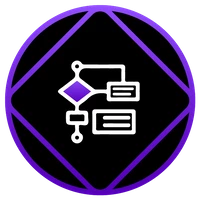
[¿Cómo implementar la continuación de datos entre escenas?](#_heading=h.7pq98ehwc97n) [7](#_heading=h.7pq98ehwc97n)

[¿Cómo implementar la continuación de datos entre sesiones?](#_heading=h.ayxtwbo1ms9m) [8](#_heading=h.ayxtwbo1ms9m)

[Evaluación de la misión 38](#_heading=h.8iy5888cyafv)

## Misión 3: ¿Cómo administrar el flujo y los datos de la Escena?

Parte del [trayecto Programador júnior](https://learn.unity.com/mission/programming-basics?pathwayId=5f7e17e1edbc2a5ec21a20af)



**Descripción general de la misión**

En esta misión de Programador júnior, aprenderás sobre la continuación de datos o cómo administrar la carga y descarga de datos entre escenas y sesiones. En el proceso, conocerás los pilares fundamentales de la programación orientada a objetos, la cual te ayudará a estructurar tu código durante toda la misión. Solo habrá un ejemplo práctico durante la misión, el cual se desarrollará a partir de las nuevas funcionalidades que crees con cada tutorial. Al final de esta misión, tomarás todo lo que aprendiste y lo aplicarás a un nuevo proyecto al desarrollar un sistema de puntaje persistente para un pequeño juego.

**Detalles clave**

* ¡Llevas ya la mitad del trayecto de aprendizaje de Programador júnior!
* Esta misión te tomará aproximadamente 5 horas para completarla. Hazla a tu propio ritmo. Recibirás experiencia a cada paso durante tu camino.
* Recuerda, no estás solo: conéctate con la Comunidad de Unity durante la misión y revisa el calendario de Learn Live para asistir a sesiones prácticas tutoriales con creadores consolidados de Unity.
* Cuando hayas finalizado la misión, obtendrás una insignia para lucir en tu perfil y tu portafolio.

**Habilidades**

Control de versiones básico

* ¿Cómo mantener un proyecto al implementar correctamente el control de versiones?
* ¿Cómo implementar las prácticas recomendadas del control de versiones con Unity Collaborate?

Creación de *scripts* de aplicación para principiantes

* ¿Cómo crear el flujo de la Escena en el estado de una aplicación?
* ¿Cómo implementar la continuación de datos entre escenas y sesiones de usuario?

## Introducción al diseño de experiencias 3D en tiempo real

| **Enlace de la lección** | [Introducción al diseño de experiencias 3D en tiempo real](https://learn.unity.com/tutorial/introduction-to-real-time-3d-experience-design?pathwayId=5f7e17e1edbc2a5ec21a20af&missionId=5f751af7edbc2a0022cdbbb6#) | |
| --- | --- | --- |
| **Duración** | **8 horas** | |
| **Resumen**  En este tutorial, explorarás los conceptos básicos del diseño de experiencias 3D en tiempo real. Podrás realizar lo siguiente:   * Resumen de las diferentes fases de producción. * Revisar el proceso de diseño básico para crear una experiencia digital. * Considerar distintos métodos para la creación de experiencias 3D en tiempo real. * Identificar tu propio proceso de diseño personalizado para un escenario específico. | |  |
| **Habilidades**  Proceso de diseño básico   * ¿Cómo implementar un proceso de diseño iterativo? * ¿Cómo practicar el principio de que los usuarios son lo más importante? | | |
| **Pasos**   1. Descripción general 2. Diseño de la experiencia y el ciclo de producción de tiempo real 3. Revisión del proceso de diseño: idea e investigación 4. Revisión del proceso de diseño: tareas más claras y prototipos iniciales 5. Revisión del proceso de diseño: diseño y desarrollo iterativos 6. Métodos comunes de diseño y desarrollo de la experiencia en tiempo real 7. Ejercicio: planifica un proceso de diseño personalizado 8. Próximos pasos | | |

## ¿Cómo configurar el control de versiones?

| **Enlace de la lección** | [¿Cómo configurar el control de versiones?](https://learn.unity.com/tutorial/set-up-version-control?pathwayId=5f7e17e1edbc2a5ec21a20af&missionId=5f751af7edbc2a0022cdbbb6) | |
| --- | --- | --- |
| **Duración** | **5 minutos** | |
| **Resumen**  En este tutorial, aprenderás lo básico del control de versiones y las razones para implementarlo en tus propios proyectos, incluso si desarrollas aplicaciones por tu cuenta. También aprenderás sobre las distintas opciones de control de versiones disponibles para usar con Unity y pondrás tus nuevas habilidades en práctica al descargar el proyecto que utilizarás en los próximos tutoriales.  **Resultado**  Al finalizar este tutorial, podrás realizar lo siguiente:   * Identificar las distintas soluciones de control de versiones que pueden utilizarse para mantener un proyecto. * Describir cómo el control de versiones se utiliza para mantener un proyecto. | |  |
| **Habilidades**  Control de versiones básico   * ¿Cómo mantener un proyecto al implementar correctamente el control de versiones? | | |
| **Materiales**  [Programador-júnior-archivos-de-inicio](https://connect-prd-cdn.unity.com/20210602/a2bc463f-fd07-4ba1-8894-650f4ef32950/Junior-Programmer-Starter-Files.zip?_ga=2.236486088.837465463.1630261839-1660102106.1600431951&_gl=1*24zqc2*_ga*MTY2MDEwMjEwNi4xNjAwNDMxOTUx*_ga_1S78EFL1W5*MTYzMDMyMjQ1Ny44My4xLjE2MzAzMjQ2MjguNTk.) (.zip) | | |
| **Pasos**   1. Descripción general 2. ¿Qué es el control de versiones? 3. ¿Quién utiliza el control de versiones? 4. ¿Cómo seleccionar tu herramienta de control de versiones? 5. ¿Cómo comenzar a usar GitHub Desktop? 6. ¿Cómo crear un nuevo archivo en tu disco duro? 7. ¿Cómo descargar, abrir y comenzar a monitorizar tu proyecto? 8. ¿Cómo realizar una entrega inicial y publicar tu archivo? 9. ¿Cómo funciona el control de versiones? 10. Resumen | | |

## Principios de la programación orientada a objetos

| **Enlace de la lección** | [Principios de la programación orientada a objetos](https://learn.unity.com/tutorial/principles-of-object-oriented-programming?uv=2020.3&pathwayId=5f7e17e1edbc2a5ec21a20af&missionId=5f751af7edbc2a0022cdbbb6) | |
| --- | --- | --- |
| **Duración** | **10 minutos** | |
| **Resumen**  En este tutorial, aprenderás sobre los conceptos básicos del paradigma de programación orientada a objetos y sus cuatro principios asociados.  **Resultado**  Al finalizar este tutorial, podrás realizar lo siguiente:   * Definir la encapsulación. * Definir la herencia. * Definir el polimorfismo. * Definir la abstracción. * Explicar cómo los pilares de la OOP trabajan en conjunto para crear código eficiente y organizado. | |  |
| **Pasos**   1. Descripción general 2. ¿Qué es la programación orientada a objetos? 3. Los cuatro pilares 4. Programación orientada a objetos en las misiones 5. Próximos pasos | | |
| **Pasos**   1. Descripción general 2. Antes de comenzar 3. Te damos la bienvenida al proyecto 4. Explora la simulación 5. Revisa el resumen del proyecto 6. Tu objetivo en esta misión 7. Próximos pasos | | |

## ¿Cómo crear un flujo de Escena?

| **Enlace de la lección** | [¿Cómo crear un flujo de Escena?](https://learn.unity.com/tutorial/create-a-scene-flow?uv=2020.3&pathwayId=5f7e17e1edbc2a5ec21a20af&missionId=5f751af7edbc2a0022cdbbb6) | |
| --- | --- | --- |
| **Duración** | **20 minutos** | |
| **Resumen**  En este tutorial, configurarás un flujo de Escena entre las Escenas menú y principal, y el flujo de salida de la aplicación en la aplicación.  **Resultado**  Al finalizar este tutorial, podrás realizar lo siguiente:   * Llamar a los métodos de inicio adecuados en la secuencia de inicialización de la aplicación. * Cargar la siguiente Escena cuando se desencadene por un evento. | |  |
| **Pasos**   1. Descripción general 2. ¿Cómo revisar el flujo de Escena requerido? 3. ¿Cómo revisar el menú en la interfaz de usuario? 4. ¿Cómo escribir un método para cargar la Escena principal? 5. ¿Cómo definir el índice de la Escena menú? 6. ¿Cómo configurar el botón Start? 7. ¿Cómo escribir un método para salir de la aplicación? 8. ¿Cómo revisar tu método con compilación condicional? 9. ¿Cómo agregar un nuevo espacio de nombres? 10. Desafío: Configura la transición de regreso a la Escena menú 11. Próximos pasos | | |

## ¿Cómo implementar la continuación de datos entre escenas?

| **Enlace de la lección** | [¿Cómo implementar la continuación de datos entre escenas?](https://learn.unity.com/tutorial/implement-data-persistence-between-scenes?uv=2020.3&pathwayId=5f7e17e1edbc2a5ec21a20af&missionId=5f751af7edbc2a0022cdbbb6) | |
| --- | --- | --- |
| **Duración** | **20 minutos** | |
| En este tutorial, aprenderás a usar la continuación de datos para conservar información en diferentes escenas al tomar un color que el usuario seleccione en la Escena menú y aplicarlo a las unidades de transporte en la Escena principal.  **Resultado**  Al finalizar este tutorial, podrás realizar lo siguiente:   * Garantizar que los datos se conserven durante toda una sesión de la aplicación mediante el método DontDestroyOnLoad de Unity. * Reconocer cuándo utilizar clases estáticas, *singletons* y variables estáticas para implementar la continuación de datos. * Modificar un GameObject con un *script* que contiene las variables que se guardarán entre escenas. | |  |
| **Pasos**   1. Descripción general 2. ¿Qué es continuación de datos? 3. ¿Cómo revisar tu tarea? 4. ¿Cómo crear un nuevo *script*? 5. ¿Cómo revisar el código? 6. ¿Cómo hacer pruebas en el Editor de Unity? 7. ¿Cómo modificar el método Awake? 8. ¿Cómo almacenar y transmitir el color seleccionado? 9. Próximos pasos | | |

## ¿Cómo implementar la continuación de datos entre sesiones?

| **Enlace de la lección** | [¿Cómo implementar la continuación de datos entre sesiones?](https://learn.unity.com/tutorial/implement-data-persistence-between-sessions?uv=2020.3&pathwayId=5f7e17e1edbc2a5ec21a20af&missionId=5f751af7edbc2a0022cdbbb6) | |
| --- | --- | --- |
| **Duración** | **20 minutos** | |
| **Resumen**  En este tutorial, escribirás código para guardar y cargar el color que el usuario seleccione para que continúe entre las sesiones de la aplicación.  **Resultado**  Al finalizar este tutorial, podrás realizar lo siguiente:   * Llamar a los métodos de inicio adecuados en la secuencia de inicialización de la aplicación. * Almacenar y organizar datos al aplicar estructuras de datos, tales como listas y diccionarios. * Guardar los datos del usuario en la secuencia de cierre de la aplicación. | |  |
| **Pasos**   1. Descripción general 2. ¿Cómo evaluar tu resumen? 3. ¿Cómo puede haber continuación de datos entre sesiones? 4. ¿Qué es JSON? 5. ¿Por qué JSON es una buena opción para tu tarea? 6. ¿Cómo agregar una clase SaveData? 7. ¿Cómo agregar un método SaveColor? 8. ¿Cómo agregar un método LoadColor? 9. ¿Cómo cargar y guardar el color en la aplicación? 10. ¿Cómo agregar una funcionalidad de prueba? 11. Próximos pasos | | |

## Evaluación de la misión 3

| **Prueba:** [¿Cómo administrar el flujo y los datos de la Escena?](https://learn.unity.com/quiz/quiz-manage-scene-flow-and-data?labelRequired=true&pathwayId=5f7e17e1edbc2a5ec21a20af&missionId=5f751af7edbc2a0022cdbbb6) | | |
| --- | --- | --- |
| **Entrega:** [Continuación de datos en un nuevo archivo](https://learn.unity.com/tutorial/submission-data-persistence-in-a-new-repo?labelRequired=true&pathwayId=5f7e17e1edbc2a5ec21a20af&missionId=5f751af7edbc2a0022cdbbb6) | | |
| **Una entrega exitosa incluirá lo siguiente:**   * Un enlace al archivo de GitHub del proyecto que muestre varias entregas con mensajes. * Un poco de continuación de datos entre escenas * Un poco de continuación de datos entre sesiones | | |