



Componente formativo

Programación de eventos de personajes y entornos

Breve descripción:

El componente formativo se enfoca en el diseño de interfaces de usuario y eventos para facilitar la interacción de las diferentes acciones que ocurrirán en los niveles de juego.

Área ocupacional:

Ciencias Naturales.

Mayo 2023

Tabla de contenido

1. Introducción	3
1. Programación de comportamientos de personajes controlados por el usuario.....	4
1.1. Configuración de animaciones de los personajes.....	7
1.2. Máquina de estado finito.....	12
2. Programación de eventos de interfaz HUD.....	15
2.1. Componentes Canvas	16
2.2. Interfaces de usuario responsivas	17
2.3. Eventos.....	18
3. Programación de eventos disparadores (interacción con el entorno)	20
4. Programación de comportamientos relativos a la mecánica de videojuego.....	23
5. Programación de inteligencia artificial.....	24
Síntesis	29
Material complementario	30
Glosario.....	31
Referencias bibliográficas	32
Créditos	33

1. Introducción

Los contenidos desarrollados en este componente formativo están enfocados en el diseño de interfaces de usuario y la programación de eventos y personajes para facilitar la interacción de las diferentes acciones que se dan en un videojuego.

Se invita a ver el video a continuación para conocer un poco más sobre los contenidos a desarrollar en este componente:

Video 1. Programación de eventos de personajes y entornos



[Programación de eventos de personajes y entornos](#)

Síntesis del video: Programación de eventos de personajes y entornos

Durante el ciclo de vida del desarrollo de un videojuego, el proceso de diseño de los comportamientos de los personajes e interfaces de usuario e inteligencia artificial juegan un papel fundamental para el desarrollo de las mecánicas del videojuego haciéndolo llamativo en cuanto a su diseño y dificultad.

Teniendo en cuenta lo anterior, es fundamental aprender acerca de las diferentes técnicas usadas en los videojuegos para el diseño de interfaces e inteligencia artificial aplicada a cada uno de los enemigos a los cuales se tendrán que enfrentar los jugadores de acuerdo a las mecánicas establecidas en el diseño del videojuego.

En este componente formativo se podrán desarrollar habilidades necesarias para entender el funcionamiento de cada uno de los componentes de juego del motor de desarrollo Unity3D, permitiéndole definir campos visuales de los enemigos que permitan activar eventos de ataque, persecución y patrullaje cuando el jugador se encuentre en un rango determinado.

1. Programación de comportamientos de personajes controlados por el usuario

La programación de los comportamientos de los personajes permitirá definir las diferentes acciones que serán ejecutadas por el personaje cuando el jugador interactúe con los diferentes dispositivos de juego (joystick, teclado, mouse).

Input manager

El input manager es una ventana especializada en el motor de desarrollo Unity3D que permite definir los diferentes ejes (axes) que se encuentren asociados a las diferentes acciones del juego y/o personaje.

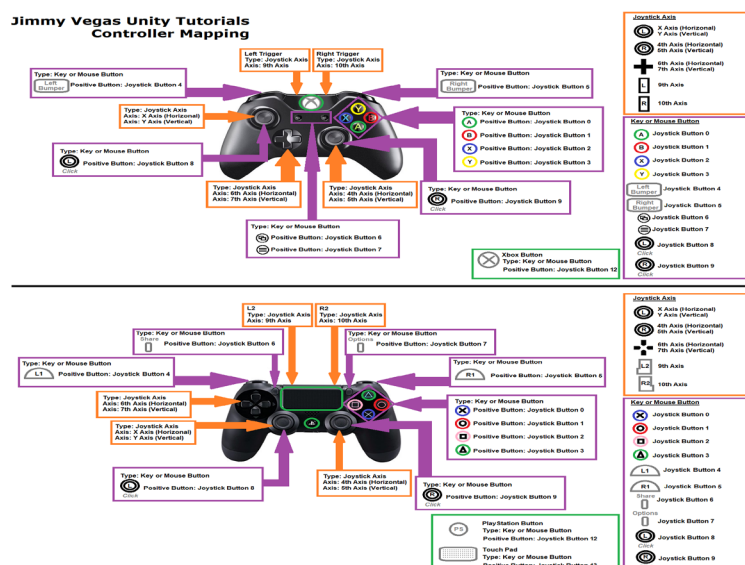
El input manager en Unity3D utiliza los siguientes tipos de controles:

Tabla 1. Controles Unity 3D

Controles	Controles Unity 3D
Key	Hace referencia a cualquier tecla (key) que se encuentre en teclado físico, como por ejemplo X, Ctrl, Shift.
Button	Hace referencia a cualquier tipo de botón que se encuentre en un control físico como un gamepad, joystick.
Virtual Axis	Permite realizar el mapeo de los ejes de un control físico. Por ejemplo, cuando el jugador mueve las palancas de un gamepad.

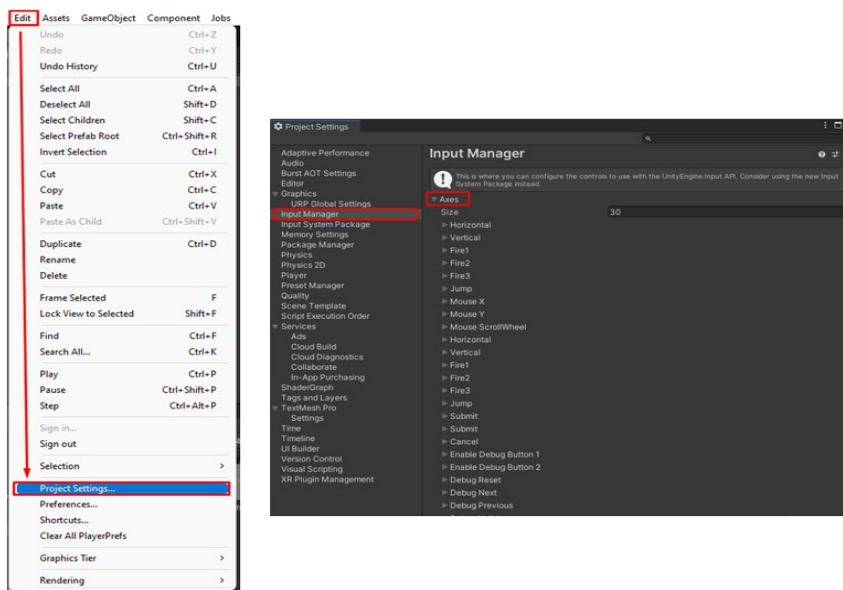
En la siguiente figura se puede observar el mapeo de controles de Unity:

Figura 1. Mapeados controles de juego



Para acceder a la ventana Input Manager dar clic en el menú Edit > Project Settings como se muestra en la siguiente imagen:

Figura 2. Input Manager Unity3D



En el siguiente video tutorial se observa la programación de personajes controlados por el usuario y la creación de un nuevo proyecto.

Video 2. Input Manager



Input Manager

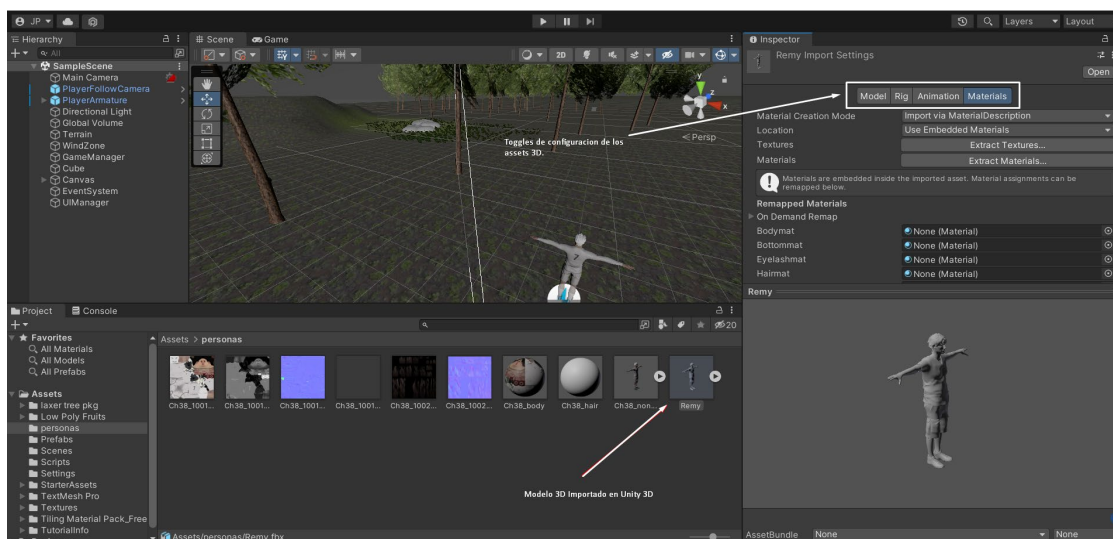
Síntesis del video: Input Manager

El video tutorial muestra paso a paso de la programación de personajes controlados por el usuario y la creación de un nuevo proyecto.

1.1. Configuración de animaciones de los personajes

El motor de desarrollo Unity 3D posee diferentes herramientas en su editor que permiten manipular las animaciones de los personajes importados. Las animaciones de los personajes pueden encontrarse embebidas en el personaje o externas al personaje.

Figura 3. Editor Unity 3D



En la web existen un gran número de recursos que facilitan los procesos de animación de personajes si no se tienen las habilidades o no se tienen las animaciones para llevar a

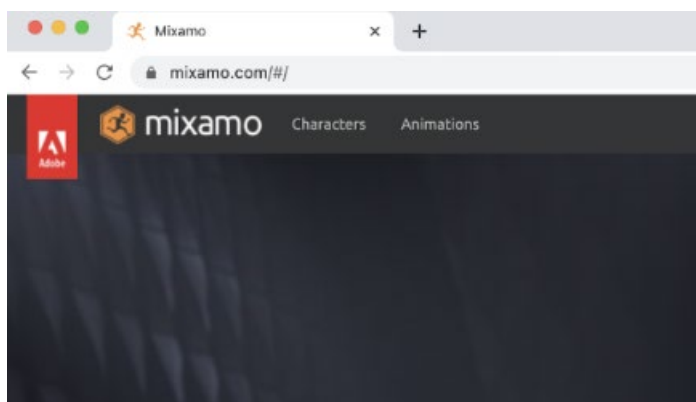
cabo este proceso. Para este componente se va a trabajar con la página Mixamo (www.mixamo.com), que permite realizar las animaciones de personajes de un juego.

Para acceder a la página de Mixamo y comenzar a hacer las animaciones se deben seguir estos pasos:

Ingreso Mixamo

Para lograr trabajar con este sitio web se debe hacer el registro en la página Web de Mixamo: www.mixamo.com. Iniciar sesión se debe hacer clic en el botón Log In.

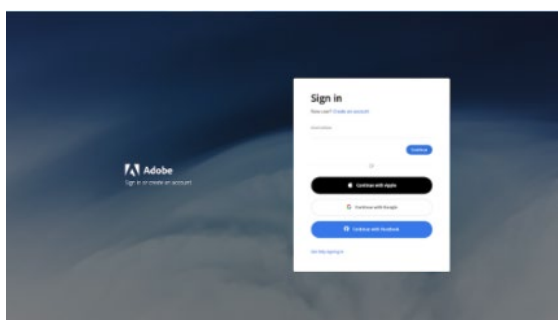
Figura 4. Ingreso Mixamo



Inicio de sesión

Una de las opciones de inicio de sesión que da la página web es ingresar con una cuenta de Google, Facebook, Apple o la de Adobe Cloud.

Figura 5. Inicio de sesión

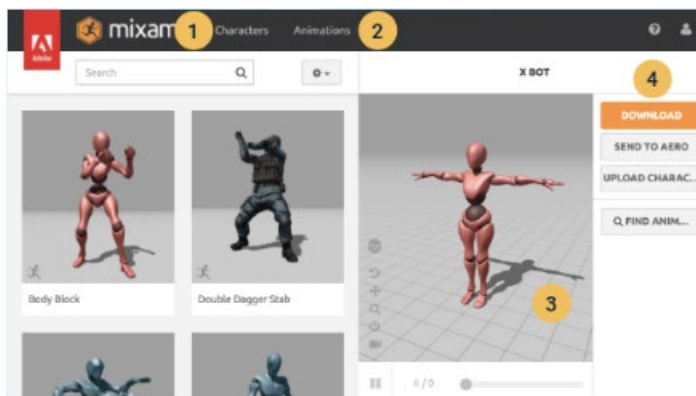


Panel principal

El panel principal cuenta con diferentes acciones que permiten realizar cambios de personajes y así tener el flujo de animaciones correcta para lograr programar los comportamientos de los personajes.

1. Explorador de personajes: permite cambiar el personaje en el proceso de animación.
2. Explorador de animaciones: permite ingresar a la galería de animaciones.
3. Previsualizador de personaje.
4. Botones de acción: permite realizar descarga de los modelos personalizados.

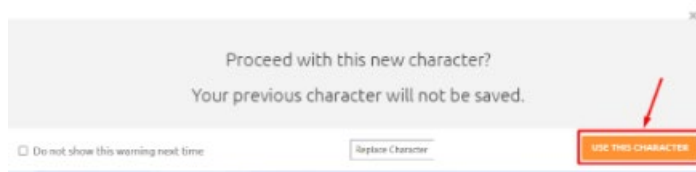
Figura 6. Panel principal



Animando el personaje

Seleccione el personaje a su gusto. Seleccione la opción Character. En el explorador de personajes seleccione el personaje que se ajuste a su criterio. Cuando seleccione el personaje se visualizará el siguiente mensaje.

Figura 7. Animando el personaje



Asignación de animaciones

El proceso de asignación de las animaciones consiste en seleccionar las animaciones adecuadas teniendo en cuenta los comportamientos que se deseen generar.

Seleccione la opción **Animations** para asignar las animaciones al personaje.

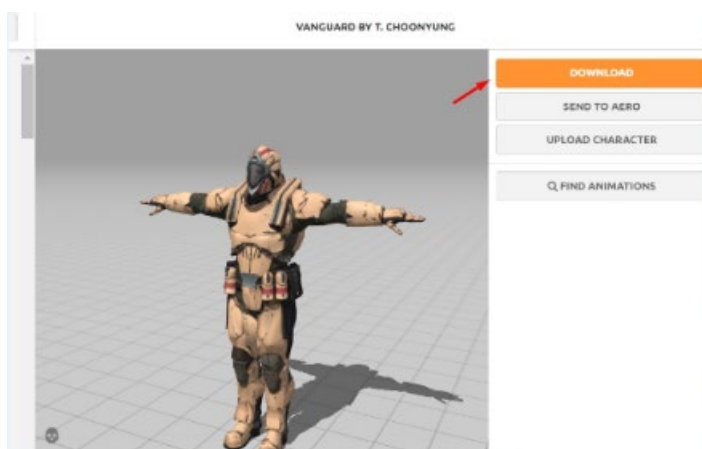
Figura 8. Asignación de animaciones

Tipos de animaciones	
<i>Idle</i>	Animación que representa el estado de reposo del personaje.
<i>Walk</i>	Animación que representa el estado de caminar.
<i>Run</i>	Animación que representa el estado de correr.
<i>Jump</i>	Animación que representa el estado de salto.

Descargue el modelo

Una vez asignadas las animaciones al personaje, descargue el modelo en Download.

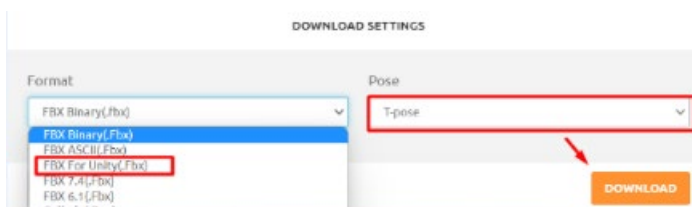
Figura 9. Descargue el modelo



Configuración de descarga

En la opción de formato, Format, seleccione FBX For Unity y en la opción Pose seleccione la opción T-Pose.

Figura 10. Configuración de descarga

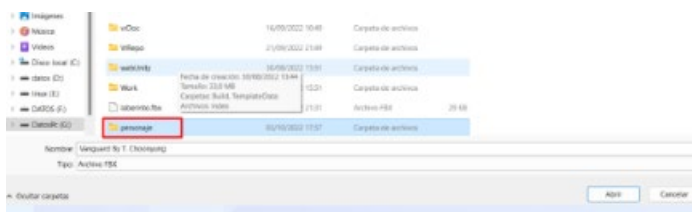


Guardar animaciones

Se recomienda que el personaje y las animaciones se encuentren en la misma carpeta de almacenamiento. Cree una carpeta y llámela personaje para efectos de la guía instruccional.

Renombrar el archivo a Soldado.

Figura 11. Guardar animaciones



En los siguientes videos tutoriales se explica cómo se pueden generar personajes y darles diferentes animaciones:

[Obtener modelo 3D en Mixamo](#)

[Importar personaje](#)

[Asignar animaciones](#)

1.2. Máquina de estado finito (Animator Controller)

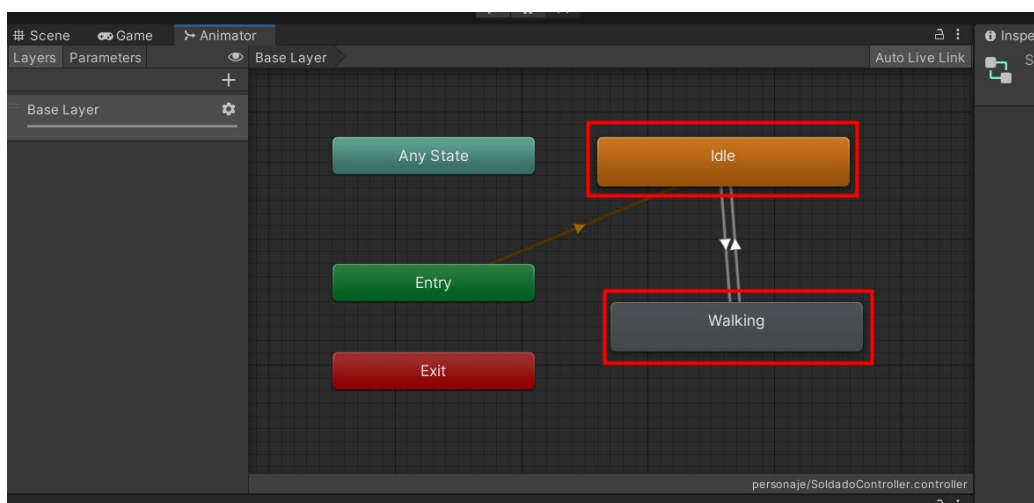
El Animator Controller es un panel del editor de Unity3D que permite organizar y mantener un conjunto de clips de animación y asociar estas animaciones por medio de transiciones. En la gran mayoría de los casos, el Animator Controller puede contener varias animaciones y cambiar entre cada una de ellas dependiendo de las condiciones que ocurran en el juego, como se puede observar en el siguiente video.

[Animator Controller](#)

Estados de animación

Los estados en la máquina de estados finito (Animator Controller) son la representación de los comportamientos que el personaje va a tener (reposo, caminar, correr y saltar).

Figura 12. Estados de la FMS

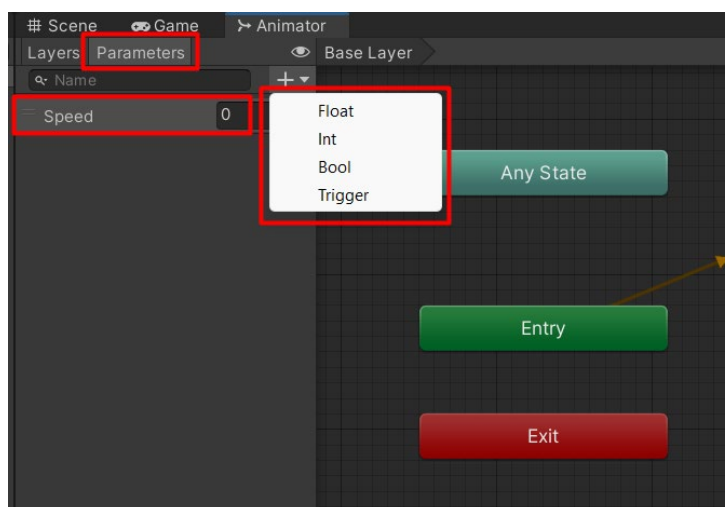


Parámetros

Los parámetros en la máquina de estados finito representan una variable que permite controlar el cambio de un estado a otro haciendo uso de una transición. Los parámetros cuentan con cuatro tipos:

- a) Float.
- b) Int.
- c) Bool.
- d) Trigger.

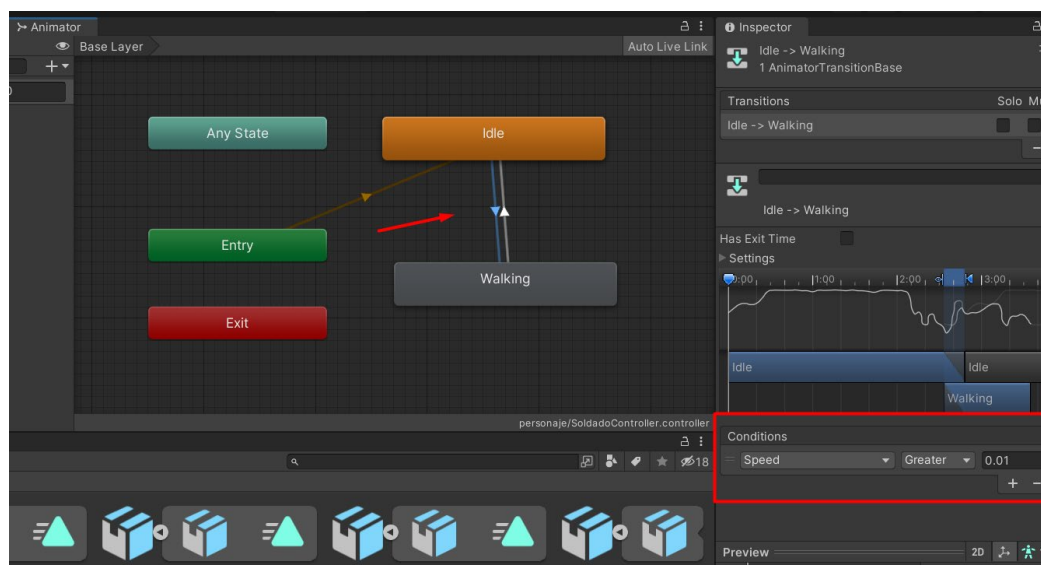
Figura 13.Parámetros FMS



Transiciones

Las State Machine Transitions existen para simplificar el proceso de transición entre los diferentes estados del Animator Controller. Las transiciones permiten tener un mayor nivel de complejidad sobre la lógica de la máquina de estados; y las transiciones poseen condiciones que permiten controlar el proceso de cambio de un estado a otro.

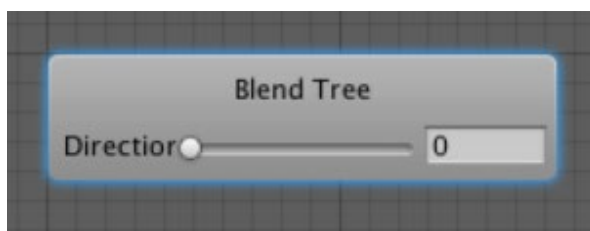
Figura 14.Transiciones FMS



Árboles de mezclado (Blend Tree)

El Blend Tree permite realizar combinaciones entre los diferentes clips de animaciones y así generar comportamientos complejos.

Figura 15. Blend Tree Unity



En el video que sigue a continuación se observa cómo se trabaja con el árbol de mezclado o Blend Tree.

[Blend tree](#)

2. Programación de eventos de interfaz HUD

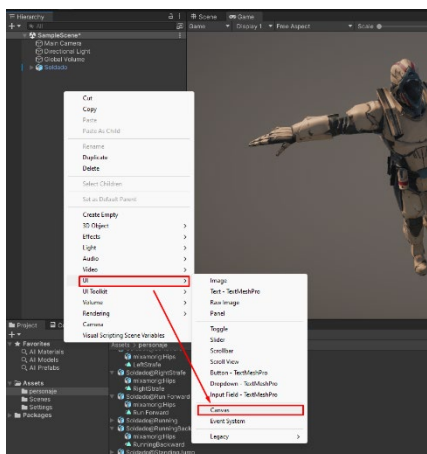
La interfaz de usuario es uno de los principales elementos de los niveles de juego ya que, por medio de estos, los usuarios pueden monitorear el comportamiento del nivel de juegos. Por ejemplo, pueden conocer cuánto tiempo les queda de juego, cuánto nivel de salud les queda, etc.

Canvas

Unity3D en su conjunto de GameObject posee uno llamado Canvas que es un lienzo que permite organizar de forma detallada la interfaz de usuario. El Canvas trabaja con un componente adicional llamado el EventSystem el cual permite detectar los eventos que se ejecutan cuando el usuario interactúa con los elementos de la UI.

Para agregar el Canvas a la escena hacer clic derecho sobre el panel de jerarquía y seleccionar la opción UI>Canvas.

Figura 16.Canvas



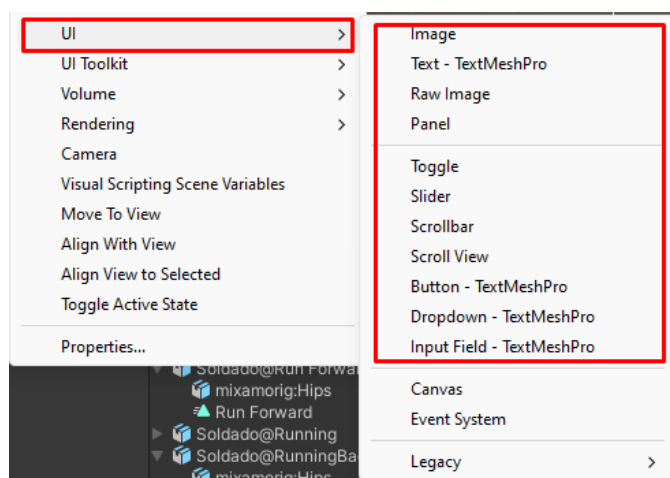
Para manipular y crear interfaces de usuario con Canvas, se invita a ver este video:

[Iniciando Canvas](#)

2.1. Componentes Canvas

Los componentes del Canvas son los elementos usados para realizar el diseño de la interfaz de usuario. Los principales componentes de la clase UI son los siguientes:

Figura 17. Componentes Canvas



En el siguiente video se analizan cada uno de los componentes del Canvas.

[Componentes Canvas](#)

2.2. Interfaces de usuario responsivas

Las interfaces responsivas son aquellas que se ajustan de forma automática a cualquier resolución de pantalla. Unity3D tiene una serie de elementos que permiten realizar este tipo de configuraciones en las interfaces de usuario, a saber:

En este video se observa cómo funciona el componente Rect transformer para manipular cada uno de los elementos del Canvas.

Elementos Rect Transform:

Unity 3D tiene una serie de elementos que permiten realizar este tipo de configuraciones en las interfaces de usuario:

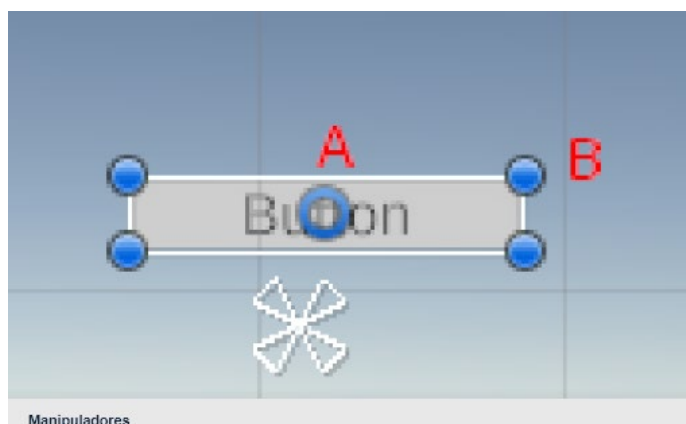
Elementos Rect Transform

A. Pivot: controla el punto de origen el cual se usa como referencia para rotar, escalar y mover el control.

B. Manipuladores: redimensionan el control seleccionado.

C. Anchor: elemento usado para anclar el punto de creación de los controles de la UI.

Figura 18. Elementos Rect Transform



Elementos Rect Transform

Los manipuladores del punto de anclaje permiten cambiar de posición de forma automática el Anchor de cada uno de los controles de la UI.

Figura 19. Elementos Rect Transform



En este video se observa cómo funciona el componente Rect Transform para manipular cada uno de los elementos del Canvas.

[Rec transformer en Canvas](#)

2.3. Eventos

Los eventos en la programación son acciones que se ejecutan cuando los usuarios interactúan con los diferentes elementos de la interfaz de usuario. En la siguiente tabla se describe cada uno de esos elementos.

Evento	Elemento UI	Descripción
OnClic	Button	Permite ejecutar una acción cuando el usuario interactúa con algún botón de la UI.
OnValueChanged	DropDown	Ejecuta una acción cuando el jugador selecciona un elemento de la lista desplegable.
OnValueChanged	InputField	Ejecuta una acción cuando cambia el valor de la caja de texto.
OnEndEdit		Ejecuta una acción cuando se finaliza el proceso de edición del texto del elemento.
OnSelect		Ejecuta una acción cuando se selecciona la caja de texto.
OnDeselect		Ejecuta una acción cuando se deselecciona la caja de texto.
OnValueChanged	Slider	Ejecuta una acción cuando el valor del slider ha cambiado.
OnChangeValue	Toggle	Ejecuta una acción cuando el valor del elemento ha cambiado. Retorna un valor booleano (true o false)

En el siguiente video se explica cada uno de los elementos de la interfaz de usuario, que componen la UI.

[Eventos Interfaz de Usuario](#)

Para ampliar el tema sobre la programación de eventos de interfaz HUD, se sugiere consultar en el material complementario la página [Unity Learn](#) el tema de IU Components.

3. Programación de eventos disparadores (interacción con el entorno)

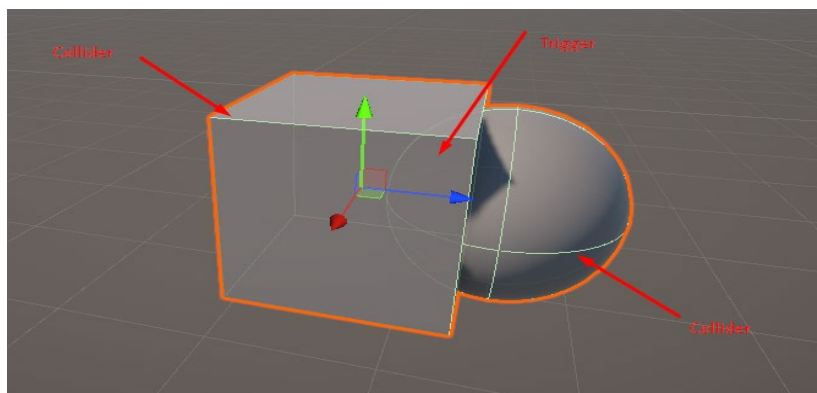
En el diseño de las mecánicas del videojuego es necesario crear interacciones con el entorno, es en este punto donde los eventos ocurridos por condiciones físicas permiten a los jugadores recoger ítems de los niveles de juego, sufrir daño por pasar sobre zonas no permitidas, etc.

Los motores de videojuegos permiten simular los comportamientos físicos del entorno real haciendo uso de diferentes componentes como rigidbodies, materiales físicos, colisionadores etc., permitiendo que las mecánicas sean dinámicas y parecidas a la realidad.

Triggers

Los triggers son eventos que ocurren cuando los objetos de videojuego se entrelazan entre sí como se observa en la figura.

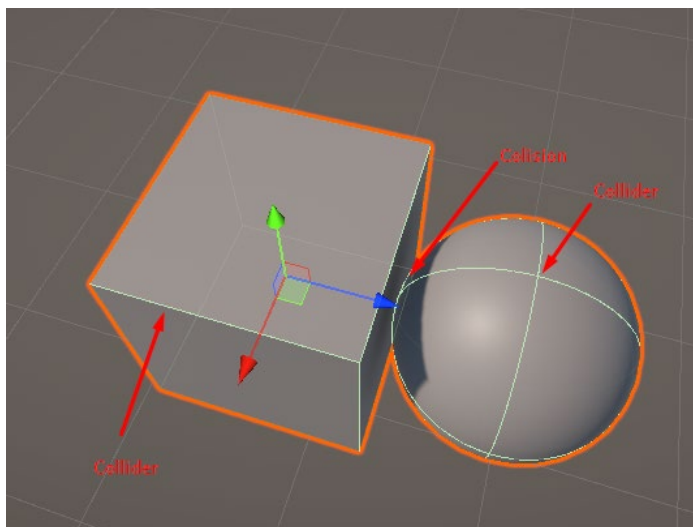
Figura 20.Trigger entre gameobjects



Collision

La colisión es un evento que ocurre cuando dos gameobjects se tocan entre sí, este evento es muy utilizado para provocar daño a los enemigos en un videojuego.

Figura 21. Collision



Los tipos de colliders son:

1. **Sphere Collider:** este tipo de collider es muy usado en formas redondas.
2. **Box Collider:** es empleado en formas cubicas, edificios, carretera, etc.
3. **Capsule Collider:** utilizado en los personajes para crear zonas de daño; es muy utilizado en los troncos de los árboles.

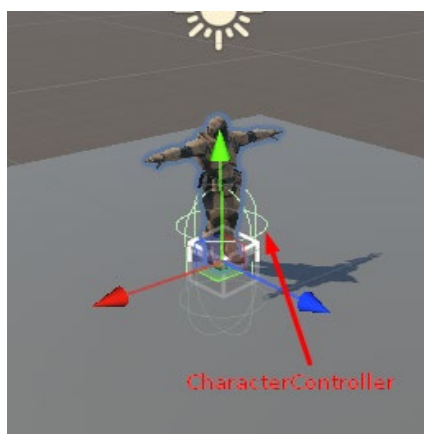
En el siguiente video se estudian las físicas y colisiones en el motor de desarrollo.

[Físicas y colisiones](#)

OnControllerHit

Este tipo de colisión es característico en los personajes de los videojuegos, ya que permite detectar cuando el jugador recoge elementos del nivel de juego.

Figura 22. Character controller



A través de los siguientes videos se explicará cómo controlar un personaje y aplicar animaciones:

[Character controller para controlar el movimiento de un personaje.](#)

Colisión con otros objetos de la escena.

[Collider Hit](#)

Generación de cámara de seguimiento al personaje.

[Cinemachine](#)

Para ampliar el tema sobre la programación de eventos disparadores e interacción con el entorno, consultar en el material complementario la página Unity Learn el tema de [Catapult Physics; Forces, and Energy](#).

4. Programación de comportamientos relativos a la mecánica de videojuego

La interacción del jugador con los objetos del videojuego desempeña un papel muy importante en las mecánicas del videojuego, ya que así puede saber el estado actual del nivel, los elementos que ha recolectado para su inventario, qué nivel de vida le queda, entre otros muchos aspectos más.

Interactividad UI Recolección de ítems de juego

Una de las principales actividades que se deben cumplir en la gran mayoría de mecánicas es la recolección de objetos en el escenario; esto hace que el jugador pueda recoger pociones mágicas, pociones de salud e incluso municiones para sus armas.

Continuando con las colisiones del Character Controller, en el siguiente video se explica la creación de un sistema de recolección y conteo que se visualiza en la interfaz de usuario (UI).

En este video se explicará paso a paso:

[Recolector](#)

5. Programación de inteligencia artificial

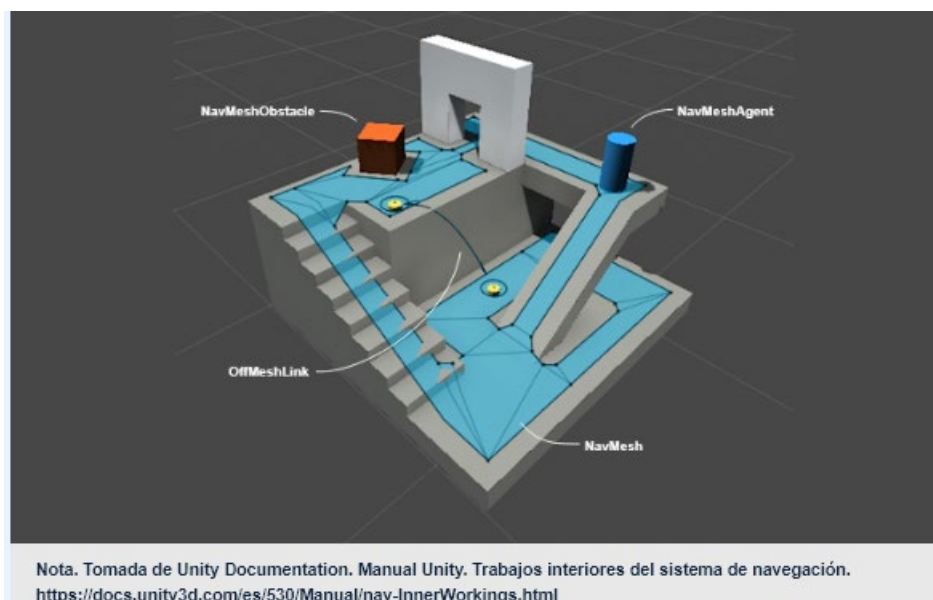
Los desarrolladores de videojuegos viven preocupados por ofrecer juegos que contengan las mejores experiencias gráficas y una mayor inmersión en cada mecánica desarrollada. La inteligencia artificial en los videojuegos también juega un papel fundamental ya que permite tener grandes comportamientos de los jugadores NPC, haciendo que cada nivel de juego sea más atractivo.

Unity 3D, al igual que muchos motores de desarrollo de videojuegos, posee una estructura bastante robusta que permite integrar de forma sencilla la inteligencia artificial a nuestros juegos. Para tener una visión general de la navegación en Unity, son necesarios los siguientes elementos:

Agente de Navegación (NavMeshAgent)

Los agentes de navegación permitirán crear personajes que tengan comportamientos autónomos mientras que se dirigen a su objetivo. Estos agentes de navegación utilizan el componente NavMesh para prevenir que los NPC no choquen entre ellos mientras que caminan en el nivel de juego.

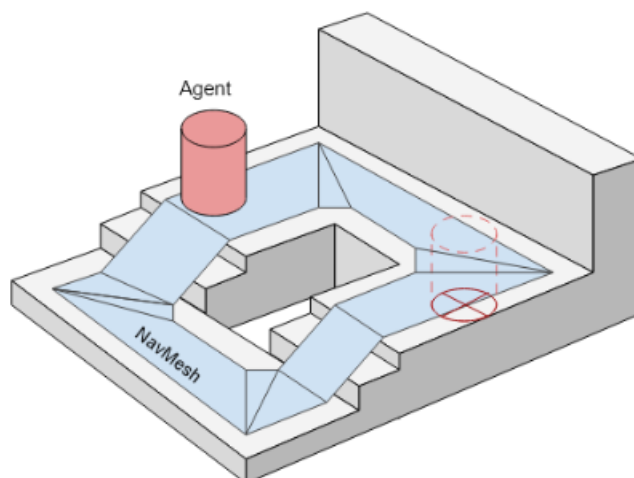
Figura 23. Agente de Navegación



Áreas de camino

El sistema de navegación debe contener información que le permita representar las diferentes áreas por donde los personajes puedan caminar en la escena de juego.

Figura 24. Áreas de camino

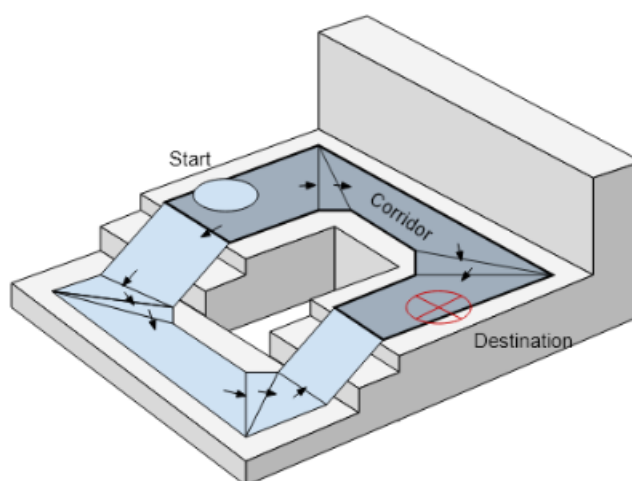


Nota. Tomada de Unity Documentation. Manual de Unity. Sistema de navegación.
<https://docs.unity3d.com/es/530/Manual/nav-NavigationSystem.html>

Encontrar caminos

Para encontrar el camino entre dos puntos en el nivel de juego, primero se debe mapear las posiciones de inicio y fin; posteriormente, el algoritmo inicia el recorrido visitando cada uno de los puntos vecinos hasta lograr alcanzar el punto final.

Figura 25. Encontrar caminos



Nota. Tomada de Unity Documentation. Manual Unity. Encontrando caminos.
<https://docs.unity3d.com/es/530/Manual/nav-InnerWorkings.html>

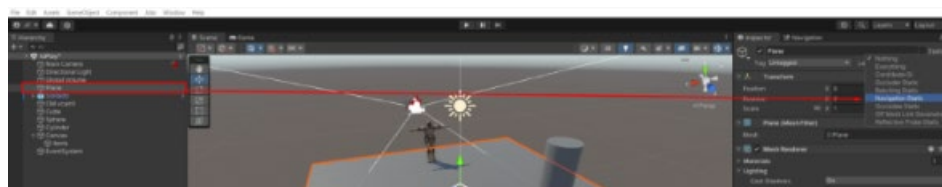
Agregar agente de navegación

Estos son los pasos para agregar el agente de navegación:

Paso 1

Seleccione el GameObject que se va a utilizar como superficie de navegación y marque la opción Static en el panel Inspector.

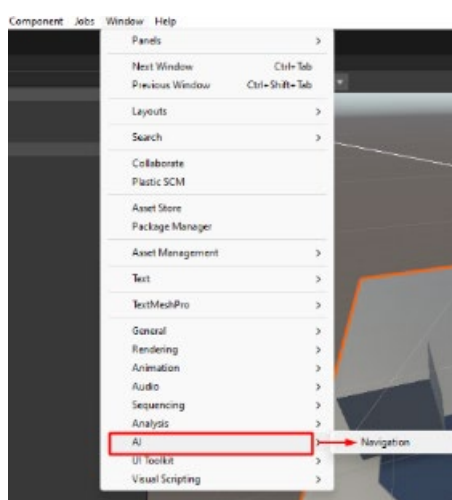
Figura 26. Paso 1



Paso 2

Ingresa al menú Windows>AI>Navigation.

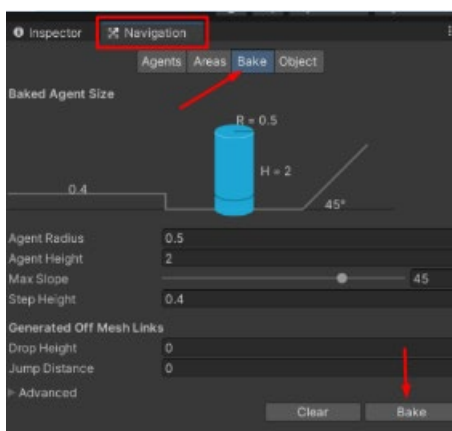
Figura 27. Paso 2



Paso 3

Seleccione la pestaña Navigation y haga clic sobre la opción Bake y para finalizar haga clic sobre el botón Bake.

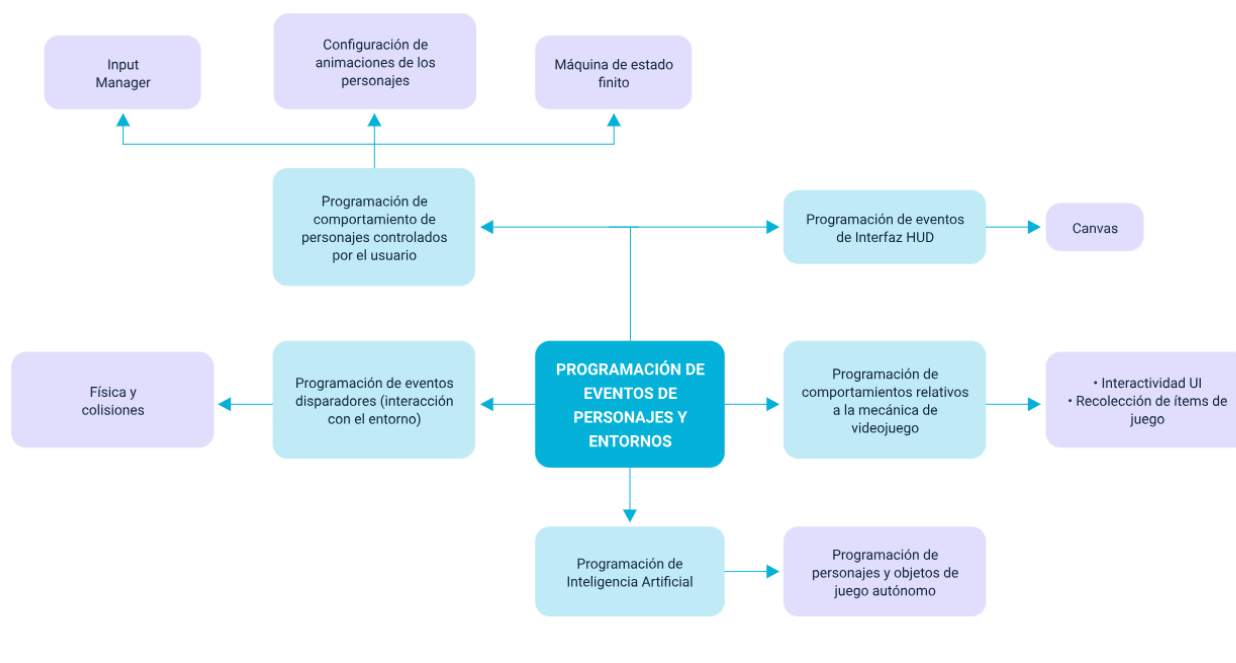
Figura 28. Paso 3



Para ampliar el tema sobre Programación de Inteligencia Artificial, consultar en el material complementario la página [Unity Learn, el tema de Navigation Meshes.](#)

Síntesis

En el siguiente esquema se presenta una recopilación de los temas trabajados en este componente formativo “Programación de eventos de personajes y entornos”:



Material complementario

Tema	Referencia APA del Material	Tipo de material	Enlace del Recurso o Archivo del documento material
Programación de eventos de Interfaz HUD	Unity Technologies. (2022). <i>UI Components</i> . Unity Learn. https://learn.unity.com/tutorial/ui-components	Website	https://learn.unity.com/tutorial/ui-components
Programación de eventos disparadores (interacción con el entorno)	Unity Technologies. (2022) <i>Catapult Physics; Forces, and Energy</i> . Unity Learn. https://learn.unity.com/project/catapult-physics-forces-and-energy	Website	https://learn.unity.com/project/catapult-physics-forces-and-energy
Programación de Inteligencia Artificial	De Byl, P. (2022). <i>Navigation Meshes</i> . Unity Learn. https://learn.unity.com/project/navigation-meshes	Website	https://learn.unity.com/project/navigation-meshes

Glosario

FMS: acrónimo usado en Unity para referenciar la máquina de estado finito.

Input: clase de Unity usado para detectar cuando el jugador interactúa con un dispositivo de entrada (teclado, mouse, joystick).

Parámetro: variable definida en una máquina de estado para controlar las transiciones entre los estados.

Public: modificador de acceso que permite dar accesibilidad a las variables y métodos en un programa.

Transición: relación que existe entre cada estado en una máquina de estados finitos.

Using palabra reservada en .net que permite importar referencias a clases.

Referencias bibliográficas

Acerenza, N., Coopes, A., Mesa, G., Viera, A., Fernández, E., Lorenzo, T., y Vallespir, D. (2009). Una metodología para desarrollo de videojuegos. *38º JAIIO - Simposio Argentino de Ingeniería de Software*, pp. 171-176.
https://www.fing.edu.uy/sites/default/files/biblio/22811/asse_2009_16.pdf

GameDevTraum. (2021). *Scripts en programación*.
<https://gamedevtraum.com/es/programacion-informatica/introduccion-a-la-programacion/que-es-script-programacion/>

Lenguajes de programador.net. (s. f). *¿Qué es una variable en programación?*.
<https://lenguajesdeprogramacion.net/diccionario/que-es-una-variable-en-programacion/>

Lenguajes de programador.net. (s. f). *¿Qué es una función (o un método) en programación?*.
<https://lenguajesdeprogramacion.net/diccionario/que-es-una-funcion-o-metodo-en-programacion/>

Unity Technologies. (2022) *Unity Learn*. <https://learn.unity.com/>

Unity Technologies. (2016). *Unity Documentation*. Manual Unity.
<https://docs.unity3d.com/es/530/Manual/UnityManual.html>

Créditos

Nombre	Cargo	Regional y Centro de Formación
Claudia Patricia Aristizábal Gutiérrez	Responsable del equipo	Dirección General
Liliana Victoria Morales Gualdrón	Responsable de línea de producción	Centro de Gestión De Mercados, Logística y Tecnologías de la Información - Regional Distrito Capital
Johlver José Pardo García	Experto temático	Centro Industrial de Mantenimiento Integral - Regional Santander
Luz Aída Quintero Velásquez	Diseñadora instruccional	Centro de Gestión Industrial - Regional Distrito Capital
Rafael Neftalí Lizcano Reyes	Responsable equipo de desarrollo curricular	Centro industrial del diseño y la manufactura - Regional Santander
Silvia Milena Sequeda Cárdenas	Asesora metodológica y pedagógica	Centro de Diseño y Metrología - Regional Distrito Capital
José Gabriel Ortiz Abella	Corrector de estilo	Centro de Diseño y Metrología - Regional Distrito Capital
Gloria Amparo López Escudero	Adecuadora instruccional - 2023	Centro de Gestión de Mercados, Logística y Tecnologías de la Información - Regional Distrito Capital
Andrés Felipe Velandia Espitia	Metodólogo para la formación virtual - 2023	Centro de Gestión de Mercados, Logística y Tecnologías de la Información - Regional Distrito Capital
Francisco José Lizcano Reyes	Responsable del equipo	Centro Industrial del Diseño y la Manufactura - Regional Santander
Yuly Andrea Rey Quiñonez	Diseñador de contenidos digitales - 2023	Centro de Gestión de Mercados, Logística y Tecnologías de la Información - Regional Distrito Capital
Laura Gisselle Murcia Pardo	Animador y productor multimedia - 2023	Centro de Gestión de Mercados, Logística y Tecnologías de la Información - Regional Distrito Capital
Edison Eduardo Mantilla Cuadros	Diseño web	Centro Industrial del Diseño y la Manufactura - Regional Santander

Jorge Enrique Haylock Calderín	Desarrollo front-end	Centro Industrial del Diseño y la Manufactura - Regional Santander
Jhon Jairo Urueta Alvarez	Desarrollador full-stack - 2023	Centro de Gestión de Mercados, Logística y Tecnologías de la Información - Regional Distrito Capital
Carolina Coca Salazar	Evaluadora de contenidos inclusivos y accesibles	Centro de Gestión De Mercados, Logística y Tecnologías de la Información - Regional Distrito Capital
Lina Marcela Pérez Manchego	Validadora de recursos educativos	Centro de Gestión De Mercados, Logística y Tecnologías de la Información - Regional Distrito Capital
Leyson Fabian Castaño Pérez	Validadora de recursos educativos	Centro de Gestión De Mercados, Logística y Tecnologías de la Información - Regional Distrito Capital