

PROYECTO V4T CURSO E-LEARNING

MÓDULO 3: PROGRAMACIÓN DE APLICACIONES PARA DISPOSITIVOS **MÓVILES**

Este módulo empieza con una visión general de las tecnologías y herramientas disponibles para la programación de aplicaciones móviles y proporciona ejemplos que se podrían usar con estudiantes de escuelas primarias. Le sigue una presentación detallada del entorno App Inventor. Se describe el proceso de desarrollo de dos juegos populares con App Inventor. Finalmente, el módulo presenta la plataforma de realidad aumentada "Metaverse" que permite a estudiantes y profesores crear sus propias aplicaciones de realidad aumentada.

Capítulo 1 - Introducción a al programación de APPs para dispositivos móviles

1.1 - Tecnologías disponibles

Antes de que pueda empezar a escribir su primera app debe decidir sobre qué sistema de software funcionará su app y qué lenguaje de programación utilizará para ello. Los sistemas de software más populares son Android proporcionado por Google o iOS por Apple.

Programar para Android requiere conocimientos de programación en Java y Android SDK. iOS sólo funciona con dispositivos Apple, como también ocurre con la herramienta de programación XCode, por lo que para ello necesitará un computador Apple con el Sistema Operativo Mac OSX. Para desarrollar una app para iOS necesitará conocer el lenguaje Objective-C o bien Swift

Si quiere empezar a programar dispone de varias posibilidades. En la siguiente unidad nos centraremos en las tecnologías disponibles

Aplicaciones Nativas





Las aplicaciones nativas están escritas en los lenguajes citados para los sistemas de software anteriores (Objective-C, Cocoa)

Web Apps (Apps HTML5)

Una tecnología con la que empezar si no está seguro de qué código usar o por dónde empezar. Usted empezará a crear aplicaciones web, que pueden ser fácilmente convertidas en versiones para dispositivos móviles, utilizando emuladores. Un emulador es un sistema que clona un programa de una arquitectura en otra estructura o sistema software. Con un emulador puede convertir una aplicación web en otra usable para otro sistema (como por ejemplo Android).

Recursos en la red

Descubra lo básico sobre aplicaciones nativas https://searchsoftwarequality.techtarget.com/definition/native-application-native-app

1.2 - Entornos de desarrollo y lenguajes de programación.

Si quiere empezar a escribir una app para Android puede empezar con "Android Studio" que proporciona un entorno integrado de desarrollo.

Android Studio ofrece:

- Sistema de compilación basado en Gradle
- Emulador con múltiples características
- Entorno que permite el desarrollo para varios dispositivos Android
- Ejecución instantánea a la aplicación en ejecución para materializar los cambios sin regenerar un fichero APK nuevo
- Plantillas de código e integración con GitHub para implementar funciones comunes e importar código de ejemplo
- Herramientas de prueba y "frameworks"
- Herramientas Lint para perfilar problemas de rendimiento, usabilidad, compatibilidad y otros
- Soporte a C++ y NDK •

La guía de usuario está escrita en un lenguaje simple que motiva al principiante a crear su primera app.





Utilizar la tecnología de bloques es también posible para empezar con su primera app. Esta opción es mucho más fácil que aprender a utilizar un lenguaje de programación; sin embargo, también tiene sus limitaciones. GoodBarber es un proveedor de herramientas de desarrollo. Al registrarse se dispone de un periodo de prueba gratuito de 30 días. La herramienta le permite crear app fácilmente con funciones avanzadas para los usuarios de la app (autenticación de usuarios, notificaciones "push", monetización y demás). GoodBarber ofrece tarifas variadas que empiezan por 32 € al mes.

En el capítulo 2 nos centramos en App Inventor, que es un entorno de desarrollo para Android suministrador por Google

Recursos en la red

Android Studio - https://developer.android.com/studio Descargue Android Studio y dele un vistazo a la guía de usuario para empezar a programar.

GoodBarber - https://www.goodbarber.com/ Herramienta para construir apps utilizando blogues.

1.3 - Apps y tecnologías emergentes usadas en Educación Primaria

¿Cómo preparar a nuestros jóvenes para la era digital que ya nos rodea a todos? Hemos de comunicar contenidos complejos en formas fáciles de ser usados. Entender programas de ordenador y algoritmos son competencias clave en el mundo digital.

Hay varias formas mejores de enseñar a los niños competencias digitales que simplemente dejarles experimentar en entornos digitales

Aprendiendo con Apps, Juguetes y Plataformas

Existen ya varias App, plataformas online y Juguetes que pueden ser usados para realizar los primeros pasos en la programación de apps (vea los recursos en la red). Estas Apps, plataformas y juguetes podrían ser usados en las escuelas o en casa para tener las primeras experiencias digitales

App - Swift Playgrounds

En esta app para iPad proporcionada por Apple, los programadores principantes solucionan puzzles y controlan personajes utilizando sentencias en el lenguaje de programación Swift. En el libro 'Introducción al Desarrollo de Apps con Swift' aprenderá cómo construir la primera app de principio al final,





aprendiendo los elementos básicos Xcode. Además, Apple proporciona una ayuda a profesores "Cualquiera puede programar curriculum" para impartir programación desde la educación primaria hasta la universidad.

Lenguajes de programación para niños

Existen diversos lenguajes de programación para niños que están mayormente basados en códigos visuales.

Scratch

Scratch es un lenguaje visual de programación diseñado en los laboratorios MIT Media especialmente para jóvenes de 8 a 16 años, para programar historias interactivas, juegos y animaciones, que pueden ser compartidas con otros. Además, estudiantes desde primaria hasta universitarios pueden aprender otras disciplinas como matemáticas, programación, lenguaje, artes y ciencias sociales. En el archivo ScratchEd Online Community Archive los educadores pueden encontrar recursos, discusiones e historias de otros educadores

NEPO - Open Roberta Lab

La iniciativa Open Roberta (del instituto Fraunhofer para el análisis inteligente de los sistemas de información IAIS) introduce la programación de robots. Los niños aprenden el lenguaje de programación visual NEPO y experimentan sus habilidades con un robot virtual. La iniciativa Roberta proporciona materiales, entrenamiento, currículums y manuales de construcción para profesores.

Workshops

Los workshops son una gran posibilidad para que los jóvenes trabajen juntos y aprendan en grupo.

Maker Spaces

Con Maker Spaces, los niños aprenden a crear algo nuevo con materiales y herramientas. Maker Spaces utiliza herramientas como Arduino, Raspberry Pi, Calliope Mini, BBC micro:bit y Makey Makey y otras herramientas para construir y programar robots. Los teléfonos inteligentes, Apps, tabletas e impresoras 3d pueden utilizarse para probar. No se trata tan sólo de habilidades para programar, sino un lugar donde los niños pueden ser creativos.

Recursos en la red





Video 'Lo que no enseñan muchas escuelas' - https://youtu.be/nKlu9yen5nc: Video motivador para construir computadores y programarlos

Scoyo - Aprendiendo programación para niños: Mit Spaß fit für die Zukunft - https://wwwde.scoyo.com/eltern/kinder-und-medien/programmieren-lernen-kinder-fit-fuer-die-zukunft Online-Magazin für Eltern rund um Lernen, Schule, Familienleben & Medienkompetenz

Code.org - <u>https://code.org/:</u> Proporcioan tutoriales de una hora para todas las edades en 45 idiomas. CodeAcademy - https://www.codecademy.com/: Sitio web de referencia para aprender lenguajes como Python, PHP, jQuery, JavaScript,...

Lightbot - http://lightbot.com/index.html: App para que los niños aprendan a programar Dash - https://uk.makewonder.com/dash/: Dash es un robot que puede ser controlado por apps. Válido para diferentes edades

CoderDojo - https://zen.coderdojo.com/find: Una comunidad global de clubs de programación compuesta de voluntarios con edades entre 7 y 17 años.

Making -Diseño digital creativo y experimental con niños. -

https://de.slideshare.net/sandra slideshare/making-kreatives-digitales-gestalten-und-experimentierenmit-kindern-einfhrung-und-ausgewhlte-werkzeuge

Arduino - https://www.arduino.cc/

Rasperry Pi - https://www.raspberrypi.org/

Makey Makey - <u>https://makeymakey.com/</u>

Calliope Mini - https://calliope.cc/en

BBC micro:bit - https://microbit.org/

Capítulo 2: Construir Apps con App Inventor

2.1 – Empezando con App Inventor

App Inventor es una aplicación web que le permite desarrollar aplicaciones para dispositivos Android (smartphones y tablets) utilizando un lenguaje de programación visual. Es gratuito y software libre. Para trabajar con App Inventor necesita

Una cuenta Google





- Un computador con acceso a internent.,
- Un navegador web compatible (Mozilla Firefox, Apple Safari, Google Chrome Microsoft Internet Explorer no está soportado).

Si tiene problemas con la conexión a internet o accediendo a su cuenta Google puede utilizar App Inventor 2 Ultimate.

App Inventor guarda el proyecto en un almacenamiento en la nube con el nombre de su cuenta Google. No necesita guardar proyectos en su disco duro local.

Una Aplicación App Inventor se obtiene con el Diseñador de Componentes (Component Designer) y el editor de bloques (Block Editor). Con el Diseñador, puede crear los formularios y pantallas de de la aplicación, disponiendo los elementos o componentes de la aplicación. Con el Editor de Bloques puede establecer un comportamiento arrastrando y ubicando bloques de programa o código. Los bloques representan los comandos que realizan alguna función a hacer.

Hay dos tipos de componentes: visibles y no visibles. Los componentes visibles (como botones, etiquetas de texto, recuadros, etc) son aquellos que serán visibles cuando la aplicación se lance y formarán parte del interfaz de usuario. Los componentes no visibles (como el acelerómetro, el sonido o el sensor de orientación) no son visibles pero proporcionan acceso a la funcionalidad del dispositivo. No forman parte de la interfaz de usuarios

Puede crear más de un formulario o pantalla en la aplicación pero debe ser cuidadoso porque cada pantalla consume recursos del dispositivo. En general, no debería tener más de 10 pantallas para una app.

Si no dispone de un dispositivo android que conectar al computador, todavía puede desarrollar y probar apps con App Inventor. El cual proporciona un emulador de Android que instala un dispositivo virtual en su computador y funciona tal cual lo hace un dispositivo Android real.

Recursos en la red

Un tutorial para principantes: http://appinventor.mit.edu/explore/ai2/beginner-videos.html Empezando don App Inventor 2 de MIT: https://appinventor.mit.edu/explore/get-started.html Un vídeo de App Inventor hecho por google: https://www.youtube.com/watch?v=sGiaXOKqeKg Un resumen del Diseñador y Editor de bloques https://appinventor.mit.edu/explore/designer-blocks.html

2.2 - El entorno de diseños





El Diseñador (Designer) le ayuda a establecer el aspecto de su aplicación. Consiste en cinco columnas o áreas:

Paleta

En la paleta ("Palette") puede seleccionar los componentes visuales y no visuales y arrastralos al área de visualización. La columna de la paleta está organizada en grupos de componentes llamados cajoneras ("drawers") compuesta de User Interface ("Interfaz de usuario"), Layout ("diseño"), Media, Drawing and Animation ("Dibujos y animaciones"), Maps ("Mapas"), Sensors, Social, Storage ("Almacenamiento"), Connectivity ("Conexiones"), Experimental, Extensions etc.

User Interface ("Interfaz de usuario"). Estos componentes son: Botones, checkboxes, selectores de fechas, imágenes, etiquetas, listas, notificaciones, cajas de texto para contraseñas, deslizadores, rodillos, selectores de hora, y visualizadores de elementos web.

Layout ("diseño y disposición") Permite ubicar y dar formato. Utilizando estos componentes, puede definir una disposición y aspecto en la pantalla (Horizontal, Horizontal con desplazamiento, Con filas y columnas, vertical, vertical con desplazamiento)

Media . Esta cajonera consiste en elementos como:

- Camcorder (Un componente visible que permite grabar vídeo utilizando la cámara del dispositivo)
- Camera (Un componente no visible para tomar fotografías usando la cámara del dispositivo)
- Player (Un componente que reproduce audio y controla la vibración)

Dibujo y animacion Esta cajonera le permite crear dibujos y animaciones. Consiste en elementos como Ball, Canvas y sprites de imágenes.

- ٠ El Canvas es un panel rectangular sensible al tacto en su aplicación donde se puede dibujar y ubicar sprites.
- Puede programar animaciones situando el elemento "Ball" y sprites dentro del canvas. Estos componentes reaccionan a toques y arrastres, interactuando con otros sprites en el borde del Canvas, moviéndose y transformándose de acuerdo a ciertos valores en sus atributos

Sensor Este componente proporciona acceso a los sensores del dispositivo (acelerómetros, sensores de posicionamiento global, de orientación, lectores de códigos de barras, sensores de proximidad, etc).





Storage Es un componente no visible para escribir y leer archivos de una carpeta privada con datos asociados a su aplicación.

Connectivity Esta cajonera consisteen componentes de tipo ActivityStarter, Bluetooth y Web

Viewer

En el área Viewer puede ver el contenido de cada pantalla de su aplicación

Components

En la columna **Components** puede ver una estructura en forma de árbol de todos los componentes (visibles o no) de un vistazo. Puede renombrar o eliminar un componente

Media

De la columna Media puede organizar los archivos de medios (sonidos, imágenes y vídeos)

Propiedades

En la columna Propiedades ("Properties") puede determinar aspectos de la apariencia y otras características de cada componente.

Recursos en la red

Una referencia rápida sobre el Designer https://appinventor.mit.edu/explore/sites/all/files/Teach/media/MITAppInventorQuickReference.pdf

Tutorial de uso de los componentes de la interfaz de usuario http://www.appinventor.org/content/howDoYou/UIModules

Una lista de los componentes básicos de App Inventor

https://appinventor.mit.edu/explore/content/basic.html

2.3 - Programando con bloques

El Editor de Bloques manipula bloques de código que programan el comportamiento de su aplicación. Los bloques elementos con forma de piezas de puzzle que se utilizan para crear el programa.

Las aplicaciones creadas por App Inventor son guiadas por eventos. Las instrucciones no se ejecutan más que como respuesta a eventos. Un evento es una acción como hacer clic en un botón o hacer un gesto en la pantalla. Mover o girar el teléfono también es un evento. El programador describe cómo debe el





dispositivo responder ante los eventos utilizando bloques que responden ante eventos (event handler blocks)

Built-in blocks (bloques incorporados)

Los Built-in blocks o bloques incorporados (Control, Lógica, Matemáticas, Texto, Listas, Colores, Variables y bloques de procedimientos) están disponibles siempre e independientemente de los componentes del proyecto.

Los bloques de control Control blocks se usan para tomar decisiones

Logic blocks. Son bloques que realizan operaciones lógicas.

Math blocks son bloques que realizan operaciones matemáticas

Text son bloques que gestionan texto.

Con los bloques Lists o de "listas" puede manipular colecciones de items bajo una lista.

Con los bloques de Variable puede manipular variables locales y globales. Existen cinco tipos de bloque s de variable (de inicialización y declaración del nombre globales, lectura ("get"), escritura ("set"), inicialización y declaración para "in-do", e inicialización de nombre local para "in-return").

Bloques de Componentes Específicos.

Muchos de los componentes de su app pueden tener bloques de métodos ("method call") en su cajonera específica. Un bloque método es un conjunto predefinido de instrucciones que le permiten usar la funcionalidad de un componente.

Bloques de procedimiento

Con los bloques de Procedimiento puede gestionar procedimientos. Un procedimiento es una secuencia de instrucciones que se agrupan bajo un nombre y llevan a cabo una tarea específica. En vez de tener un programa complejo, puede crear procedimientos e invocarlos cuando desee para aprovecharlos.

Los bloques de procedimiento y muchos otros, incluyendo el 'if-else' y 'make list', proporcionan un botón "blue mutator". Haciendo clic en este botón puede transformar el bloque, por ejemplo, añadiendo argumentos a los procedimientos o ramificaciones 'else' para un 'if'.

Recursos en la red

Aquí se explica cómo funcionan los bloques básicamente. https://appinventor.mit.edu/explore/understanding-blocks.html





Un resumen de la Arquitectura de una App http://www.appinventor.org/Architecture2

En este tutorial se enseña cómo crear procedimientos en App Inventor. http://appinventor.mit.edu/explore/ai2/support/concepts/procedures.html

Este tutorial trata las llamadas a procedimientos en App Inventor http://ai2.appinventor.mit.edu/reference/blocks/procedures.html#do

Un resumen breve de cómo utilizar mutators en App Inventor https://appinventor.mit.edu/explore/ai2/support/concepts/mutators.html

2.4 - Conexión a smartphones o tablets.

Configurando App Inventor

Dispone de dos maneras de ejecutar la aplicación en su dispositivo Android: gracias a una conexión Wi-Fi (Al Companion) o mediante un cable USB. En el primer caso necesitará tanto que su computador como el dispositivo Android estén conectados a la misma red inalámbrica. Si no dispone de un smartphone o table Android, todavía puede crear aplicaciones con App Inventor, ya que proporciona un emulador de android que funciona como un dispositivo Android pero todo aparecerá en la pantalla de su ordenador. Puede probar las aplicaciones en el emulador y distribuirlas a otros incluso mediante Play Store. En este caso sí que debe tener la app App Inventor Companion instalada en su dispositivo

Instalar y ejecutar el emulador en App Inventor 2

Los pasos para instalar y ejecutar App Inventor son los siguientes: Paso 1. Instalar App Inventor a partir del instalable Paso 2. Lanzar a ejecución aiStarter (sólo para Windows y GNU/Linux) Paso 3. Abrir un proyecto en App Inventor y conectarlo al emulador

Conectar al teléfono o tableta mediante un cable USB

Los pasos para utilizar App Inventor con el cable USB son los siguientes: Paso 1: Instalar App Inventor a partir del instalable. Paso2: Descargar e instalar el Companion MIT App Inventor 2 en su teléfono Paso 3: Lanzar a ejecución aiStarter (Sólo Windows y GNU/Linux) Paso 4: Configurar su dispositivo para USB (Apague el modo depuración de USB) Paso 5: Conecte el computador y el teléfono y autentíquese si hace falta





Paso 6: Compruebe la conexión.

Conecte su teléfono o Tableta mediante WiFi

Los siguientes pasos completarán este proceso: Paso 1: Descargar e instalar la App Companion de MIT App Inventor 2 en su teléfono Paso 2: Conectar su teléfono y su computadora a la misma red inalámbrica Paso 3: Abrir un proyecto App Inventor y conectarlo a su dispositivo

Recursos en la red

Un resumen de la configuración de App Inventor 2 http://appinventor.mit.edu/explore/ai2/setup.html

Una guía rápida de la instalación y ejecución del Emulador http://appinventor.mit.edu/explore/ai2/setup-emulator.html

Una guía rápida de la conexión de su dispositivo Android mediante cable USB http://appinventor.mit.edu/explore/ai2/setup-device-usb.html

Un resumen del proceso de conexión de la tableta o teléfono a una red WiFi http://appinventor.mit.edu/explore/ai2/setup-device-wifi.html





Capítulo 3: Crear juegos con App Inventor

3.1 - Cómo crear un juego simple

A continuación tiene una lista de pasos orientativos que pueden seguirse para crear un juego simple usando App Inventor.

Diseñar la interfaz gráfica de usuario del juego

En el Designer de App Inventor:

- Seleccione componentes de la paleta ('Palette') (botones, etiquetas, etc.).
- Arrastre los componentes seleccionados en el área de visualización ('Viewer') y sitúelos adecuadamente para lograr el aspecto que usted desee.
- Del diseño de la paleta ('Layout palette'), puede utilizar disposiciones horizontales de elementos (de izquierda a derecha) mediante HorizontalArrangement, disposiciones verticales (de arriba a abajo) con VerticalArrangement, o disposiciones como una tabla mediante TableArrangement.
- Los componentes seleccionados son ubicados en una área de componentes y nombrados por App Inventor. Puede eliminar o cambiar el nombre de los componentes allí si lo desea
- Añada ficheros de medios (imágenes, sonidos) a su juego subiéndolos desde su computador. Los ficheros de medios están situados en un área ('Media area') dentro de la columna de los Component.
- Haciendo clic en un componente desde el área de visualización, puede cambiar sus detalles (título, fondo, color, etc.). Los detalles posibles se listan en el área de Propiedades ('Properties')
- De la paleta de Dibujos y Animaciones, seleccione y configure el componente Canvas. En el panel rectangular y bidimensional, que es sensible a los gestos usted puede situar imágenes o sprites
- Si lo necesita, añada un componente Reloj ('Clock'). El Reloj actúa como un componente no visible que ofrece cálculos y manipulación del tiempo basándose en el reloj interno del dispositivo

Definir el comportamiento de los objetos. El editor de bloques

- Cree e inicialice variables globales y locales
- Seleccione un componente integrado de la cajonera (Lógica, control, bloque matemático, de texto, etc.) o bien un componente específico que usted haya creado previamente con el designer. Cada componente específico tiene bloques para gestión de eventos, llamadas a métodos, establecimiento de propiedades.
- Elija los bloques adecuados y arrástrelos al área de visualización





- Configure estos bloques y conéctelos apropiadamente para establecer el comportamiento
- Puede definir procedimientos para crear nuevos blogues de instrucciones que se repiten
- Utilice procedimientos y llamadas a métodos

3.2 – Crear un juego como Pong con App Inventor

Pong es un juego que se usa para demostrar los pasos básicos para crear un juego tal como describe la sección 3.1. Los siguientes dos documentos describen cómo puede hacerse Diseñar el interfaz gráfico del juego Pong Definir la funcionalidad de los componentes del juego Pong. El editor de bloques

Recursos adicionales

https://appinventor.mit.edu/explore/sites/.../Pong%205a 13.pdf http://www.appinventor.org/content/ai2apps/intermediateApps/pong

3.3 – Crear un juego como MoleMash con App Inventor

MoleMash es otro juego usado como demostración de los pasos que se toman para crear un juego como describe la sección 3.1. Los siguientes dos documentos describen cómo puede hacerse. Diseñar el interfaz gráfico de usuario del juego MoleMash Definir la funcionalidad de los componentes del juego MoleMash game - El editor de bloques

Recursos adicionales

Mole Mash - App Inventor Tutorial Parte 1 https://youtu.be/Ya1ejdRwKvw Mole Mash - App Inventor Tutorial Parte 1 https://youtu.be/IPSG7bYXN2M





Capítulo 4 - Metaverse. Realidad Aumentada

4.1 – ¿Qué es Metaverse?

Metaverse es una plataforma web gratuita de fácil aprendizaje que permite a los profesores y estudiantes crear sus propias experiencias de realidad aumentada (RA) sin necesidad de escribir ningún programa. Las experiencias de RA. se desarrollan con Metaverse Studio y pueden ser reproducidas con la App Metaverse Mobile (iOS y Android).

¿Qué pueden los profesores y estudiantes crear con Metaverse? Las posibilidades son interminables, pero hay algunos ejemplos orientativos:

- Búsqueda de tesoros con RA,
- Juegos RA,
- Historias interactivas con RA,
- Puzzles y concursos con rA,
- Viajes virtuales con RA,
- Muros mediáticos •

¿Qué beneficios proporciona a los estudiantes? Crear experiencias interactivas de RA mejora la alfabetización digital, las habilidades creativas y pensamiento lógico. Adicionalmente, los estudiantes están más motivados en el propio proceso de aprendizaje, aplican su conocimiento y actúan como investigadores dependiendo de la materia en la que está basado su proyecto de RA. Los profesores pueden utilizar Metaverse con cualquier asignatura y estudiantes de cualquier nivel, incitando a aprender mediante proyectos y mediante el trabajo colaborativo.

Recursos en la Red

Lo que los educadores tienen que contar sobre Metaverse: (en Inglés) En este vídeo, educadores usan Metaverse con sus estudiantes y comparten sus propias experiencias y la de sus estudiantes Metaverse Preview: Una vista previa rápida de Metaverse Studio y la App Metaverse Mobile. Portales de RA: Este vídeo muestra portales de RA "teleport" de varios usuarios de todo el mundo. Los portales han sido creados con cámaras de 360° y Metaverse Joe Merrill's Lake Park Elementary Class: Un juego de RA sobre vocabulario orientado a estudiantes de

primaria creado con Metaverse





4.2 - Empiece a usar Metaverse Studio

Para utilizar Metaverse Studio primero debe crear una cuenta en la plataforma y familiarizarse con el entorno de la app. El siguiente documento proporciona los pasos que se necesitan dar para iniciarse en el uso de Metaverse. Incluye una presentación del entorno Metaverse Studio

Utilice este documento para Familiarizarse con Metaverse Studio

Recursos en la red

Metaverse Studio Overview: Demostración del procedimiento básico de uso de Metarverse Studio para crear Experiencias de RA.

Metaverse Dashboard Overview: Un resumen rápido del aspecto del panel principal de Metaverse.

4.3 - Creando una experiencia de RA.

Metaverse Studio ofrece muchas herramientas para crear experiencias RA simples o complejas fácilmente. El siguiente documento presenta los pasos básicos que debe seguir para crear experiencias en RA.

Use este documento para Crear una experiencia AR con Metaverse

Recursos en la red

Metaverse Storyboard Overview: Un resumen rápido de Metaverse Experience Storyboard. Character Scenes: Demostración de cómo se pueden usar las escenas con personajes. Text Input Scenes: Demostración de cómo se pueden usar las escenas de introducción de texto. Publishing Experiences: Una guía rápida sobre cómo publicar sus experiencias Metaverse. Add a GPS Location: Tutorial sobre cómo añadir ubicaciones GPS a las experiencias Metaverse. Build a Digital Breakout: Aprenda cómo construir una actividad "Digital Breakout" para su clase j En menos de 15 minutos! Dese un paseo por una experiencia aquí: https://studio.gometa.io/discover/me/...

Build a Simple Trivia Game: Aprenda cómo hacer un juego de trivia de 3 cuestiones que usa escenas de Alarma para las preguntas y contabiliza los puntos usando Property Blocks.

