	Curso 2017- 2018 SaladilloFitVR	2º C.F.G.S. DESARROLLO DE APLICACIONES MULTIPLATAFORMA
		DESARROLLO DE INTERFACES

Nuestro afán emprendedor no tiene fin y, de los creadores de *SaladilloWare* y *SaladilloFit*, ahora llega **SaladilloFitVR**, la nueva aplicación para ir al gimnasio de un modo virtual para ponernos a punto y estar a tono en la playa este verano que ya nos acecha.

Se trata de una aplicación de realidad virtual para Android utilizando el SDK Google VR para Cardboard.

Para la persistencia de datos nos apoyaremos en la funcionalidad que incluye Unity 3D *PlayerPrefs*, también haremos uso de servicios REST a través de la Web API SaladilloFitVR (proporcionada).

La aplicación


1. Escena Main: Mostrará una habitación correspondiente a un cubo cerrado.

- Lo primero que veremos será un mensaje de bienvenida, en un panel **Welcome** situado en la pared de enfrente, la que se ve nada más iniciar la aplicación.



- En la pared izquierda mostrará un panel **Configuration** similar al de la siguiente imagen:

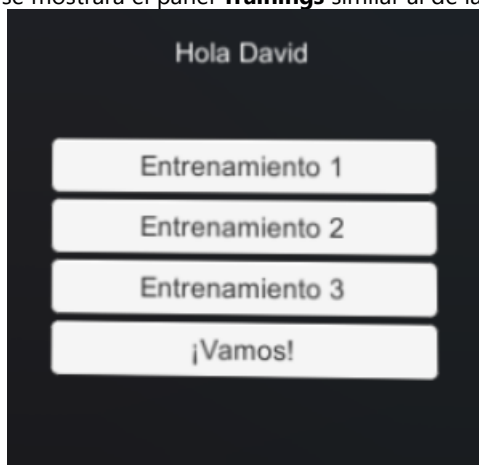


	Curso 2017- 2018 SaladilloFitVR	2º C.F.G.S. DESARROLLO DE APLICACIONES MULTIPLATAFORMA
		DESARROLLO DE INTERFACES

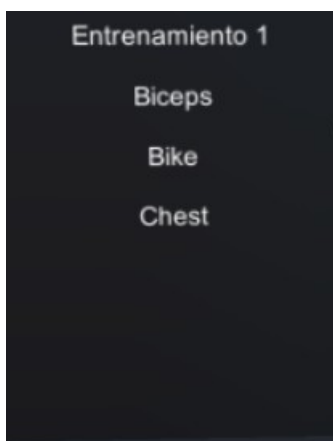
- En la pared derecha mostrará un panel **Client** similar al de la siguiente imagen:




- Bajo el panel **Welcome** se mostrará el panel **Trainings** similar al de la siguiente imagen:

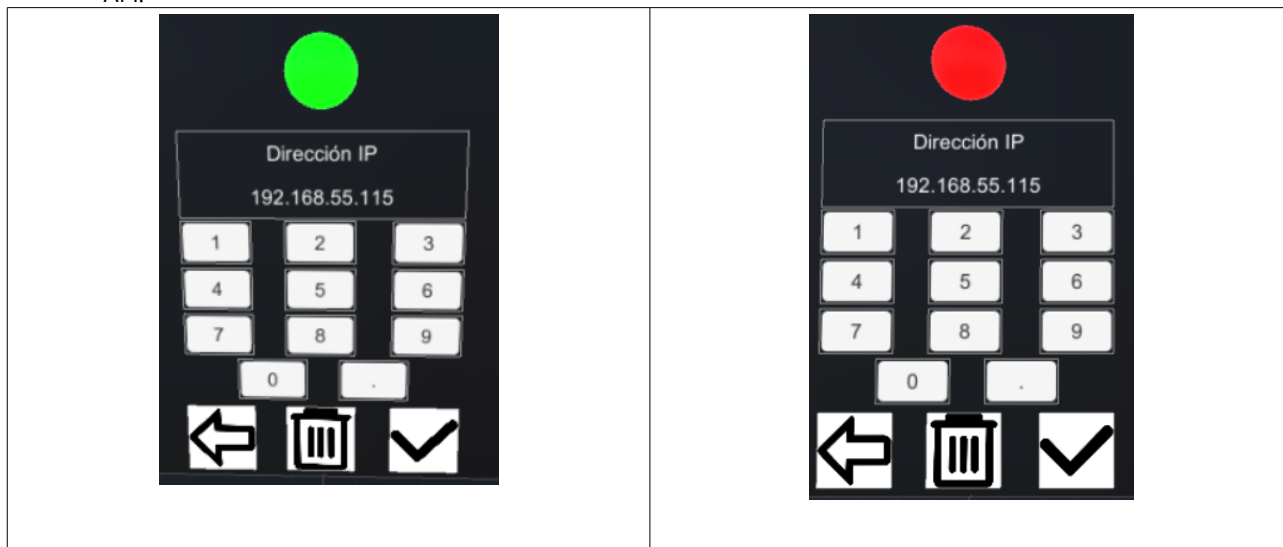


- A la derecha de los paneles **Welcome** y **Trainings** se mostrará un panel **Detail** similar al de la siguiente imagen:



	Curso 2017- 2018 SaladilloFitVR	2º C.F.G.S. DESARROLLO DE APLICACIONES MULTIPLATAFORMA
		DESARROLLO DE INTERFACES

1.1 Panel Configuration: El panel es igual al creado en clase y el funcionamiento es el mismo, mostrando una esfera verde si se ha podido establecer conexión o una esfera roja en caso de no poder conectar con la Web API.



La única diferencia es que los botones Back, Clear y Save deberán tener la propiedad Text del componente Text del objeto Text de cada Button vacía (ver Imagen 1) y en la propiedad Source Image del componente Image de cada Button se deberán asignar las imágenes back_button, clear_button y confirm_button respectivamente.

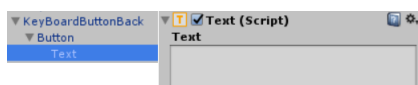



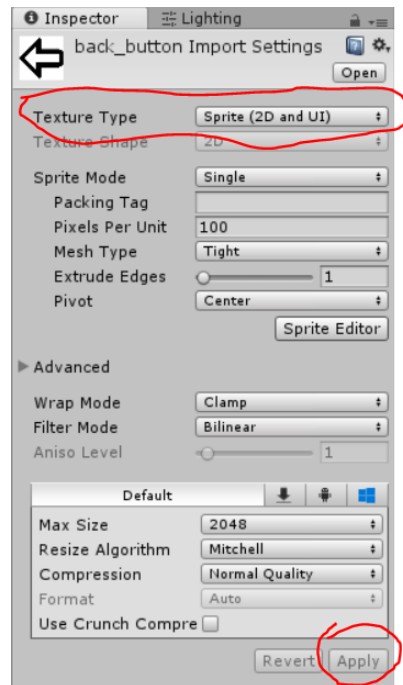
Imagen 1



Imagen 2

Estas imágenes deberán ser arrastradas previamente a una carpeta **Images** como hemos hecho en otras ocasiones y, recordad que, una vez que hayan sido arrastradas, antes de ser asignadas a los botones, hay que cambiarles el **Texture Type** a **Sprite 2D and UI** y darle al botón **Apply** para que el cambio tenga efecto.

	Curso 2017- 2018 SaladilloFitVR	2º C.F.G.S. DESARROLLO DE APLICACIONES MULTIPLATAFORMA
		DESARROLLO DE INTERFACES



Cada vez que se pulse el botón de **Confirmación**:


- se guardará la dirección IP en los PlayerPrefs
- se guardará la dirección IP en una variable en GameManager
- se intentará establecer conexión con la Web API, mostrándose la esfera correspondiente que indica el estado de la conexión. El método de la Web API para comprobar la conexión responde en la dirección: <http://{0}/SaladilloFitVR/api/SaladilloFitVR/CheckConnectivity>, con los siguientes parámetros:
 - {0} → deberá ser reemplazado por la dirección IP de la Web API

Si se establece conexión, el panel **Client** estará activo, pero si no se consigue establecer conexión el panel **Client** deberá desactivarse.

1.2 Panel Client: Panel similar a **Configuration**, pero sin las esferas que indican el estado de conexión.

Cada vez que se pulse el botón de **Confirmación**:

- se guardará el DNI en una variable en GameManager
- accederá a la Web API para obtener el Nombre de la persona correspondiente. El método de la Web API que nos devuelve el nombre de un cliente responde en la dirección: <http://{0}/SaladilloFitVR/api/SaladilloFitVR/GetClientName?dni={1}>, con los siguientes parámetros:
 - {0} → deberá ser reemplazado por la dirección IP de la Web API
 - {1} → deberá ser reemplazado por el DNI del cliente cuyo nombre se quiere obtener
- Si se recupera un nombre:
 - Se guarda el nombre en una variable del GameManager
 - Se reemplaza el valor del DNI por el del nombre recuperado:

	Curso 2017- 2018 SaladilloFitVR	2º C.F.G.S. DESARROLLO DE APLICACIONES MULTIPLATAFORMA
		DESARROLLO DE INTERFACES




- Se cambia el mensaje de bienvenida de "Bienvenid@ a Saladillo FIT VR" a "**Hola <nombre>**" y activa el panel **Training**.
- Si no se recupera ningún nombre (cadena vacía) el mensaje de bienvenida debe mostrar "Bienvenid@ a Saladillo FIT VR" y el panel **Training** se debe ocultar en caso de que se estuviera mostrando. Se seguirá mostrando el mismo DNI esperando que el cliente lo corrija.

NOTA 1: Por confidencialidad, no se guardará el DNI de los clientes en las PlayerPrefs al pulsar el botón **OK**, de modo que cada vez que se inicie la aplicación, su valor estará vacío.


NOTA 2: Los valores válidos de DNI y los nombres correspondientes devueltos por la Web API son los siguientes:

DNI	NOMBRE
21341568	Laura
31342568	Alejandro
41343568	Sergio
61344568	Fran
71345568	Antonio
81346568	Diego
91347568	Carlos
11348568	Ramón
51349568	David

1.3 Panel Trainings: muestra los tres botones con los diferentes tipos de entrenamiento que ofrece nuestro centro deportivo y un botón **¡Vamos!** para comenzar a entrenar. Estos botones de "Entrenamiento" deberán tener asociados un script **TrainingButtonScript** tal y como se indica en el siguiente apartado.

1.3.1 TrainingButtonScript: Recibe los siguientes parámetros:

- Un valor entero **Training** indicando el número de entrenamiento y que será asignado por cada botón (1, 2, 3, respectivamente).
- Un GameObject llamado **Detail**, que se corresponde con el panel **Detail**.
- Un GameObject llamado **TrainingItem**, que corresponderá con un prefab **TrainingItem** que será instanciado.

	Curso 2017- 2018 SaladilloFitVR	2º C.F.G.S. DESARROLLO DE APLICACIONES MULTIPLATAFORMA
		DESARROLLO DE INTERFACES

Accederá a la Web API para traer una lista de nombres correspondientes a las máquinas del gimnasio que habrá que utilizar en cada entrenamiento. Para ello, llamará al servicio de la Web API:

<http://{0}/SaladilloFitVR/api/SaladilloFitVR/GetTraining?training={1}>, con los siguientes parámetros:

- {0} → deberá ser reemplazado por la dirección IP de la Web API
- {1} → deberá ser reemplazado por el identificador de entrenamiento correspondiente

Activa el panel **Detail**, que por defecto estará desactivado.

Una vez que se ha recuperado la lista del entrenamiento correspondiente:

- Se destruirán todos los objetos **TrainingItem** hijos de **Detail** en caso de que existieran por un acceso anterior
- Se irá añadiendo un objeto **TrainingItem** a **Detail** por cada uno de los elementos recuperados.

El botón “¡Vamos!” tendrá asociado un script llamado **StartTrainingScript** tal y como se indica en el siguiente apartado.

1.3.1 StartTrainingScript: Hará un log del entrenamiento que se va a realizar y cargará la escena con las máquinas para entrenar. Para ello, llamará al servicio de la Web API:

<http://{0}/SaladilloFitVR/api/SaladilloFitVR/LogTraining>, con los siguientes parámetros:

- {0} → deberá ser reemplazado por la dirección IP de la Web API


Esta llamada, igual que vimos en clase la llamada a LogClient, debe ser de tipo Post en vez de tipo Get, ya que espera información para almacenarla en la base de datos. Los parámetros que deben ir en el Form son:

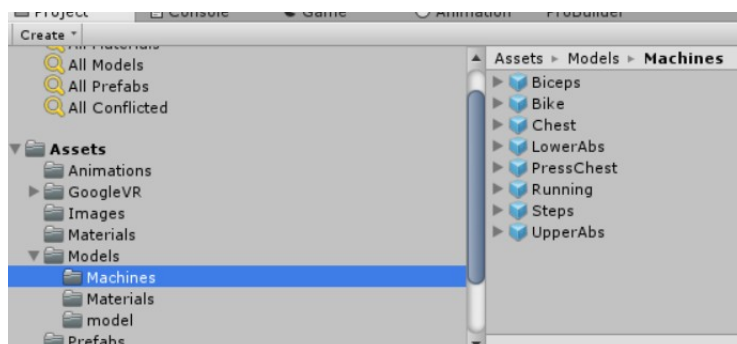
```
// Construye la información que se envía a la Web API
WWWForm form = new WWWForm();
form.AddField("dni", GameManager.dni);
form.AddField("training", GameManager.training);
```

1.4 Panel Detail: Será un **Empty** situado a la derecha del panel **Training** situado en una posición que permita mostrar una lista de hasta tres **TrainingItem**.

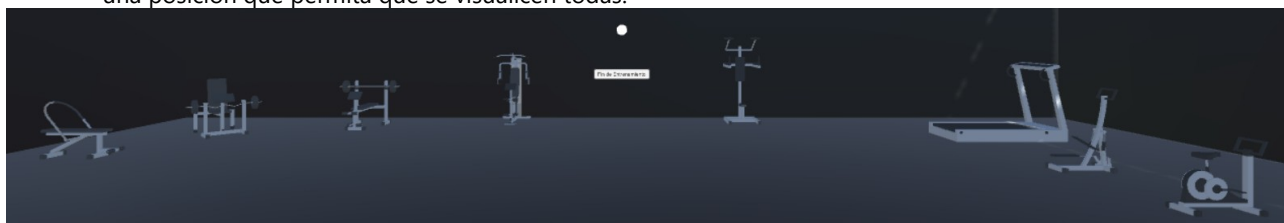
1.4.1 TrainginItem: Será un prefab parecido al **ClientButton** que hemos hecho en clase, pero en vez de tener como hijo un Button, tendrá como hijo un **Text**, ya que la lista será meramente informativa y no se va a interactuar con ella.

2. Escena Machines: Mostrará una habitación correspondiente a un cubo cerrado con máquinas de gimnasio. Para añadir las máquinas de gimnasio será necesario importar el paquete **SaladilloFitVRMachines**. Nos creará una carpeta **Models** en el proyecto, dentro de la que habrá otra llamada **Machines** que contiene los prefabs con las máquinas del gimnasio tal y como se muestra a continuación:

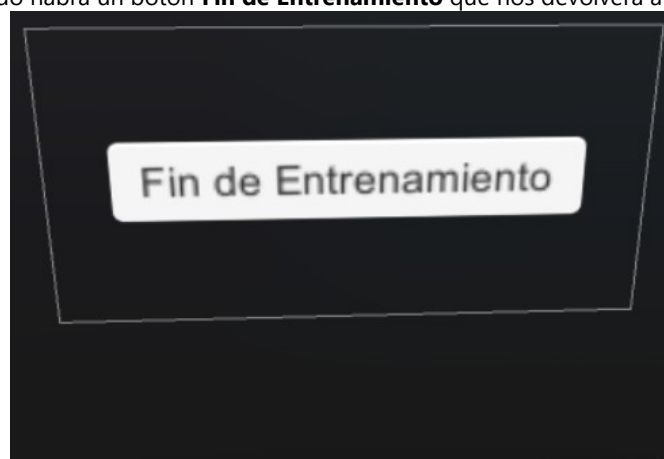
	<p>Curso 2017- 2018 SaladilloFitVR</p>	<p>2º C.F.G.S. DESARROLLO DE APLICACIONES MULTIPLATAFORMA</p>
		<p>DESARROLLO DE INTERFACES</p>




Bastará con crear un empty llamado **Machines**, arrastrar dentro los prefabs con las máquinas y colocarlas en una posición que permita que se visualicen todas.



En la pared del fondo habrá un botón **Fin de Entrenamiento** que nos devolverá a la escena **Main**.



	Curso 2017- 2018 SaladilloFitVR	2º C.F.G.S. DESARROLLO DE APLICACIONES MULTIPLATAFORMA
		DESARROLLO DE INTERFACES

Para la realización de esta práctica podéis utilizar una copia del proyecto de Unity creado en clase, así podéis aprovechar la configuración inicial, el Player, el CustomButton para crear otros botones a partir de él y tener implementada la parte del temporizador, etc., eliminando lo que no utilicéis. Solo deberá contener dos escenas: Main y Machines.

La Web API creada para que la utilicéis en vuestro entorno local está disponible en https://drive.google.com/open?id=1tVVj8lxHtKt1bE-HtBp_D7X4acKG19Uj

Os recuerdo la importancia de la realización de esta práctica, ya que el examen consistirá en hacerle modificaciones (si no hay práctica, es imposible hacer el examen).