Desarrollo de videojuego para enseñar medidas de control del Dengue.

Documento de Arquitectura de Software

Versión <4.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 02/09/14 | 1.0 | Se crea un borrador de la plantilla, sin agregar gran parte de diagramas. | José Omar Colorado Pérez |
| 04/09/14 | 2.0 | Se agregar los diagramas correspondientes, y las referencias a documentos externos para el entendimiento de este. | José Omar Colorado Pérez |
| 09/09/14 | 3.0 | Se cambia el diagrama de clases, por ende se revisan las plantillas que se ven afectadas por este cambio. | José Omar Colorado Pérez |
| 10/10/14 | 4.0 | Reemplazo de la vista de implementación generada inicialmente, por una más detallada y con las correcciones discutidas. | José Omar Colorado Pérez |

Table of Contents

1. Introduction 3

1.1 Purpose 3

1.2 Scope 3

1.3 Definitions, Acronyms, and Abbreviations 3

1.4 References 3

1.5 Overview 3

2. Architectural Representation 3

3. Architectural Goals and Constraints 3

4. Use-Case View 3

5. Logical View 3

5.1 Overview 3

5.2 Architecturally Significant Design Packages 3

5.3 Use-Case Realizations 3

6. Process View 3

7. Deployment View 3

8. Implementation View 3

8.1 Overview 3

8.2 Layers 3

9. Data View (optional) 3

10. Size and Performance 3

11. Quality 3

Software Architecture Document

# Introducción

El grupo GYMOL de la Universidad del Quindío, encargado de vigilar el Dengue en el departamento del Quindío, percibió que las tasas de presencia de esta enfermedad en eran muy altas, debido a esto busco la forma de ayudar a mejorar la prevención y control del virus. Para esto con ayuda del grupo SINFOCI, específicamente el semillero creado para este propósito, se planteó la idea de creación de un juego serio con el propósito de inculcar correctos métodos de control del dengue y enfermedades transmitidas por vectores.

## Propósito

Este documento busca exponer las decisiones arquitectónicas elegidas, y mostrarlas de forma que todos los interesados puedan comprenderlas, empleando diferentes vistas dependiendo el interesado, en general busca que el modelo de arquitectura sea comprensible para los usuarios, los interesados en el desarrollo de este, los encargados del desarrollo y sus responsables.

## Alcance

Aspectos de calidad, desarrollo e implementación se ven descritos en este documento de forma detallada para la socialización con el usuario, desarrolladores y ponentes de la idea con el fin de determinar la claridad de lo que se va hacer.

## Definiciones, Acrónimos y Abreviaciones

Videojuego: Juego electrónico que se visualiza promedio de una pantalla.

Usabilidad: Facilidad con la que un usuario interactúa con una herramienta o aplicación.

Plataforma: se refiere al sistema operativo en el que se va a ejecutar la aplicación.

Dengue: Es una enfermedad infecciosa, que es trasmitida a través de un zancudo.

Unity: Es un motor de desarrollo de videojuegos.

Vector: un vector es un agente generalmente orgánico que sirve como medio de transmisión de un organismo a otro. Los vectores biológicos se estudian por ser causas de enfermedades, pero también como posibles curas.

GYMOL: Grupo de investigación de inmunología molecular, para el cual se está desarrollando el videojuego.

SINFOCI: Sistemas de Información y Control Industrial, grupo encargado del desarrollo de software e investigación afín al tema.

Nicho: El nicho está relacionado con la adaptabilidad de las especies y designa su ubicación en el ambiente global y su sistema de vida

Pupa: Fase de desarrollo de un insecto posterior al estado de larva y anterior al de adulto.

Motor de desarrollo: Herramienta informática especializada en el desarrollo, prueba e implementación de videojuegos, diseños 3D, entre otros.

## Referencias

[http://unity3d.com/es/unity/performance.]

## Descripción General

A continuación se presentara la vista 4+1, con una descripción de las ilustraciones que allí se emplean, buscando dejar claro las bases de desarrollo sobre las cuales se pretenden trabajar, así como la integración al modelo que trae inmerso el motor de desarrollo Unity, con los cambios necesarios según las necesidades identificadas.

# Representación Arquitectonica

En esta representación arquitectónica se usara UML para modelar la forma en que se pretende desarrollar los diferentes escenarios, la forma en que se comportaran normalmente, teniendo en cuenta los posibles escenarios que se presentan bajo un funcionamiento normal. De igual modo se creó un diagrama de clases (***Ilustración 3)*** basados en las necesidades del videojuego, teniendo en cuenta la estructura fundamental que presenta el Unity, esto con el fin de lograr un equilibrio entro lo que la plataforma nos ofrece y lo que se desea construir. Basados en este, se realizan los diagramas correspondientes que soportaran los casos de uso identificados, como lo son los diagramas de realización de caso de uso, diagramas de secuencia. Gran parte de estos diagramas están contenidos en las plantillas anexas a este documento, buscando eliminar la duplicación de información.

# Metas Arquitectonicas y Restricciones

3.1. Escenarios de calidad

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Atributo de Calidad** | Facilidad de entendimiento | | | |
| **Interés del atributo** | Medio | | | |
| **Descripción** | El sistema facilitara la interacción con una comunidad en riegos de contagio d dengue. | | | |
| **ID Escenario** | 001 | | | |
| **Escenario** | El usuario hace uso de la página web o la aplicación móvil para el pago de sus comparendos, tratando de evitar congestiones en las oficinas del SMT. Dado esto se hace de vital importancia asegurar la facilidad en el procesamiento de estas tareas, para que los usuarios aprendan el funcionamiento de manera sencilla. | | | |
|  |  |  |  |  |
|  |  | **Fuente del Estimulo** | Usuario |  |
|  |  | **Estímulo** | Interacción con el menú principal |  |
|  |  | **Artefacto** | Menu principal app |  |
|  |  | **Ambiente** | Bajo condiciones normales |  |
|  |  | **Respuesta** | Esta operación se realizara en un tiempo inferior a 30 sg. |  |
|  |  | **Medida de respuesta** | Tiempo de entendimiento funcionamiento menú principal |  |
|  |  |  |  |  |
| **Importancia** | Alta | | | |

# Vista de caso de usos

## Modelo de caso de usos

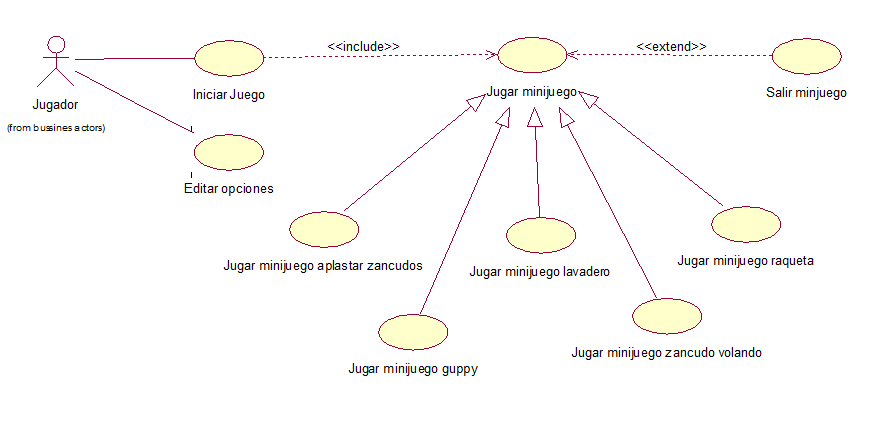


Ilustración 1

En la ilustracion 1

## Prioridad de Casos de Uso

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Caso de Uso** | **Prioridad para el Negocio** | **Prioridad Técnica** |
|  | Jugar minijuego aplastar zancudos | Alta | Alta |
|  | Jugar minijuego guppy | Alta | Alta |
|  | Jugar minijuego lavadero | Alta | Alta |
|  | Jugar minijuego zancudo volando | Alta | Alta |
|  | Jugar minijuego raqueta | Alta | Alta |
|  | Jugar minijuego | Alta | Alta |
|  | Salir minijuego | Media | Media |
|  | Iniciar juego | Media | Baja |
|  | Editar opciones | Media | Baja |

## Descripción de los Casos de Uso

### Jugar minijuego aplastar zancudos

El primer nivel con el que el usuario se encontrara luego de iniciar el juego, en este el entrara en contexto del enemigo al cual combatirá, el mosquito y descubrirá que le será imposible combatirlo dadas las circunstancias y las herramientas que emplea.

### Jugar minijuego guppy

En este nivel el usuario conocerá una herramienta, por así llamarla el pez “Guppy” poco usual para combatir la proliferación de vectores. Controlando este pez tratara de eliminar huevos, larvas y pupas, durante este juego.

### Jugar minijuego lavadero

Un nicho muy común dentro del hogar de los mosquitos, son los lavaderos debido a la falta de aseo de estos y al tiempo que el agua mantiene reposada en estos lugares. El jugador combatirá los mosquitos en sus tres fases iniciales y se buscara que entienda la importancia de la limpieza de los lavaderos, pocetas o similares.

### Jugar minijuego zancudo volando

El jugador entrara literalmente en los zapatos del zancudo, y buscara evadir a uno de sus enemigos naturales, los sapos.

### Jugar minijuego raqueta

En este nivel el usuario hacer uso de una raqueta eléctrica tratando de evitar, que el zancudo hembra deposite sus huevos en unos recipientes con agua. En tres diferentes niveles de dificultad, en los cuales se presentaran tres oleadas con diferentes dificultades y contara con tres vidas para superar estas.

### Jugar minijuego

Generalización de la forma en como el sistema cargara todo los niveles que se encuentran en el juego, esto se hace para evitar repetir información en la descripción de los casos de uso, y para facilitar la implementación de los diferentes niveles.

### Iniciar juego

Luego de que se hayan calculados los valores de la energía de contratos es necesario realizar el cálculo de la energía comprada en el Mercado Ocasional.

### Editar opciones

En el menú inicial el usuario tendrá algunas opciones como sonido, información de los desarrolladores, funcionamiento de los niveles.

Para mayor información de los casos de uso y los correspondientes diagramas de secuencia, de comunicación de cada uno de estos, diríjase al documento **RUC,** anexo a este documento.

# Vista Logica

Teniendo en cuenta la estructura definida en el motor Unity 3D, el cual fue el seleccionado para el desarrollo del videojuego se busca la creación de un juego compuesto por 5 minijuegos. En el diagrama de clases del motor Unity podemos observar que todos los objetos presentes en los juegos creados con este motor son de tipo Object, ya que presentan similitudes, las cuales pueden ser usadas por los desarrolladores, tales como la textura, el material, los scripts que pueden llegar a tener, entre otras particularidades que podemos observar en el diagrama. Empleando esta estrategia de desarrollo inmersa en el motor de desarrollo, se construirán los casos de uso definidos anteriormente siguiendo la descripción propuesta en la especificación de estos, cualquier cambio deberá ser diligenciado en estas plantillas, sin dejar de lado la estructura del Unity 3D.

Se propone un único paquete contenedor de los 5 minijuegos, partiendo del modelo que se propone en la ilustración 3.

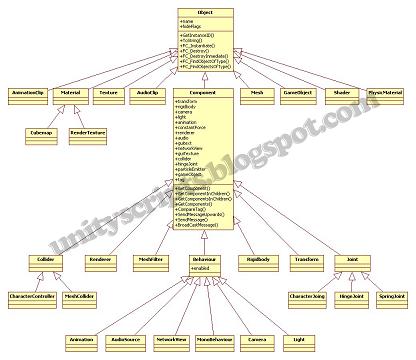


Ilustración 2

El siguiente diagrama nace luego de identificar los elementos necesarios para la realización de los diversos escenarios, en donde se definen clases como enemigo que representan los diversos ciclos de vida del zancudo, brindando de este modo la posibilidad de emplear cualquiera de estos en los 5 escenarios. Las herramientas empleadas para combatir los mosquitos están definidas en la identificación de los casos de uso, dado esto se propone esta clase “herramienta”. Las anteriores clases son de tipo gameObject de unity, debido a que son objetos que interactúan con el usuario y pueden llegar a sufrir cambios más rápidamente que los escenarios. La clase componente presente en la Ilustración 2, se modela en el diagrama de clases dado que son los objetos necesarios para ambientar los escenarios, estos difieren de los gameObject debido a que pueden sufrir cambios, pero no interactúan directamente con el usuario.

Las clases sprite, script, sonidos, animaciones son elementos que puede contener un objeto y puede alterar su comportamiento o la forma en que el usuario vea los objetos, como había mencionado Componente y GameObject son un tipo de objeto especifico cada uno dentro de las escenas. Las clases restantes se hacen necesarias para llevar el control del usuario, su puntaje, el manejo de las escenas en que se encuentre, su puntaje, entre otros factores necesarios para el correcto desarrollo del juego.

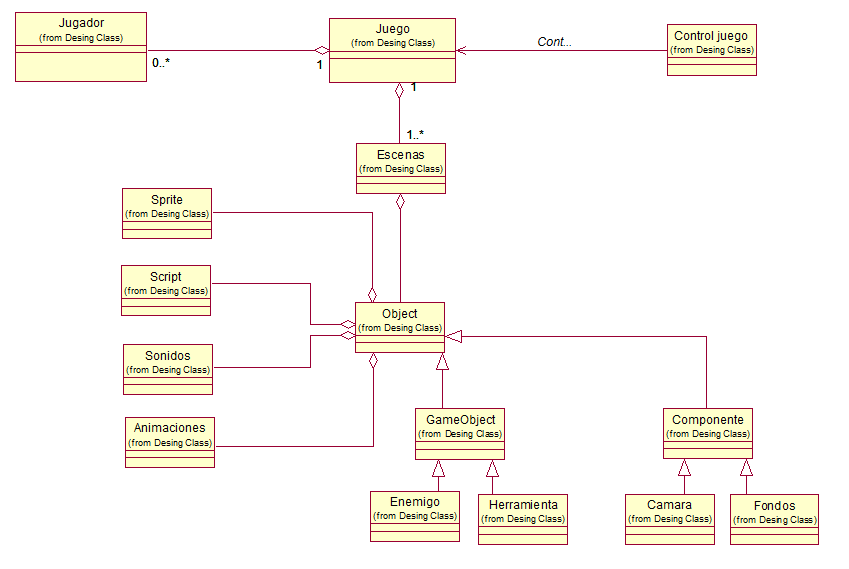


Ilustración 3

## Paquetes de Diseño

Se propone un único paquete que como se había mencionado anteriormente contendrá los minijuegos, es decir el juego estará compuesto por 5 mini juegos, y estos serán manipulados por el jugador que intervendrá en todos los procesos de activación de estos.

El juego contara con 5 minijuegos que a su vez tendrán unos niveles de dificultad, los cuales se verán ya sea por tiempo, cantidad de enemigos o un minimo de acciones necesarias para completar los miniJuegos. Los puntajes obtenidos en los minijuegos seran acumulados como la puntuación final del juego.

## Realizacion casos de uso

Para mayor información acerca de cómo se implementaran los casos de uso en el software a desarrollar, remítase al documento **UCRS** correspondiente a cada caso de uso**.** En este encontrara la forma elegida para el desarrollo de los elementos de cada paquete, la forma en que interactúan los elementos, así como la descripción del proceso.

# Vista de procesos

Está en la estructura básica de los 5 minijuegos, por tal motivo solo se muestra un diagrama de actividades de estos 5, siendo actividades básicas pero importantes para el desarrollo y entendimiento de lo que se pretende hacer, se planteó ese diagrama siguiente, en el cual se identifica un solo tipo de usuario, el Jugador, quien interactuara con el sistema y desarrolla las actividades que podemos observar sin percibir la secuencialidad de sus acciones.

Para evitar ambigüedades o interpretaciones diversas de como se comportaran los minijuegos, se especifica en los casos de uso los pasos necesarios para satisfacer las necesidades del cliente y cumplir con las metas pactadas, siguiendo un comportamiento que se adapta al diagrama de actividades, que en cada nivel es modelado siguiendo esta estructura.

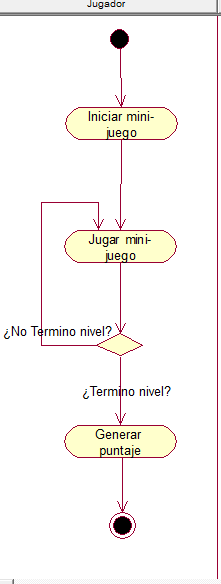


Ilustración 4

# Vista de Despliegue

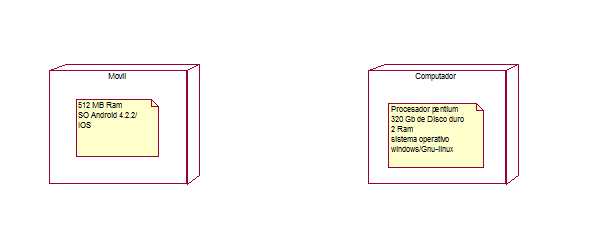


Ilustración 5

La configuración básica para dos entornos de despliegue, en dispositivos móviles y computadoras, se buscara que la aplicación corra con las características básicas en los diferentes ambientes a fin de mejorar el alcance del juego. A futuro se podrá implementar el juego en plataforma online, con el fin de lograr una mayor penetración de este.

# Vista de Implementación

A continuaion se presenta una vista de implementacion general del juego, en sus cinco componentes esenciales, y los menus que intereacturan con estos. Ademas para un mayor entendimiento se hacen los diagramas de implementación correspondientes para los cinco minijuegos, de los cuales explicare las diferencias fundamentales entre los escenarios.

En general hay componentes de scripts y animaciones, dado que los primeros hacen las modificacion de las animaciones del objeto contenedor de estas. Las animaciones, sonidos y cambios visuales estan contenidos en este modulo, y son instanciados desde los que usan sus servicios.

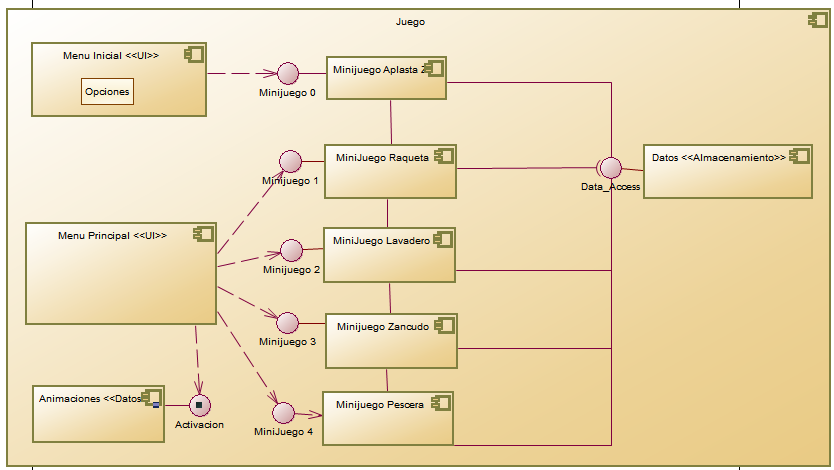


Ilustración 6

**Minijuego Aplastar Zancudo**

Este contiene una única escena, por lo tanto no se hace necesario insertar un componente para el manejo de escenas, en esta escena todos los elementos se ven influenciados por scripts que modifican la forma en que son visualizados, o en que actúan.

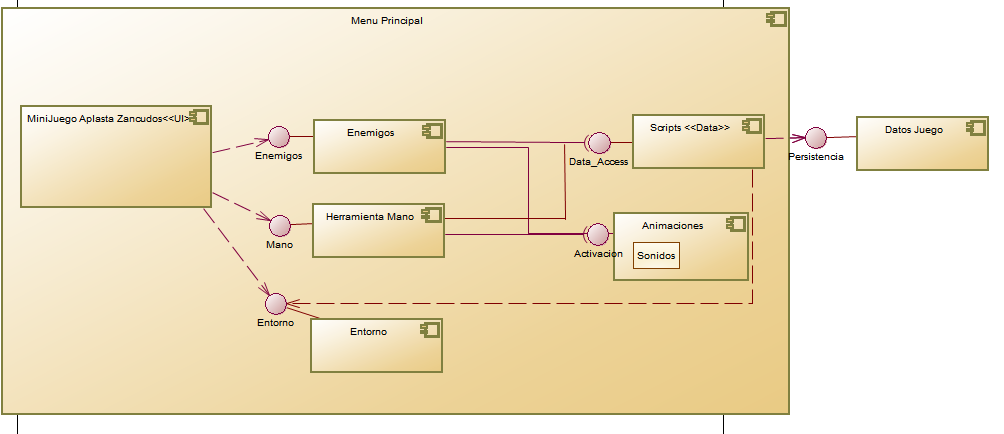


Ilustración 7

**Minijuego Raqueta**

Este minijuego está compuesto por 4 elementos primordiales escenas, enemigos, herramienta y el entorno en el cual tienen lugar los hechos del juego. Esto integrado en la interfaz del minijuego y es manipulado, activado por los scripts. Los puntajes obtenidos por el usuario son persistidos en el dispositivo en el cual el juego está corriendo.

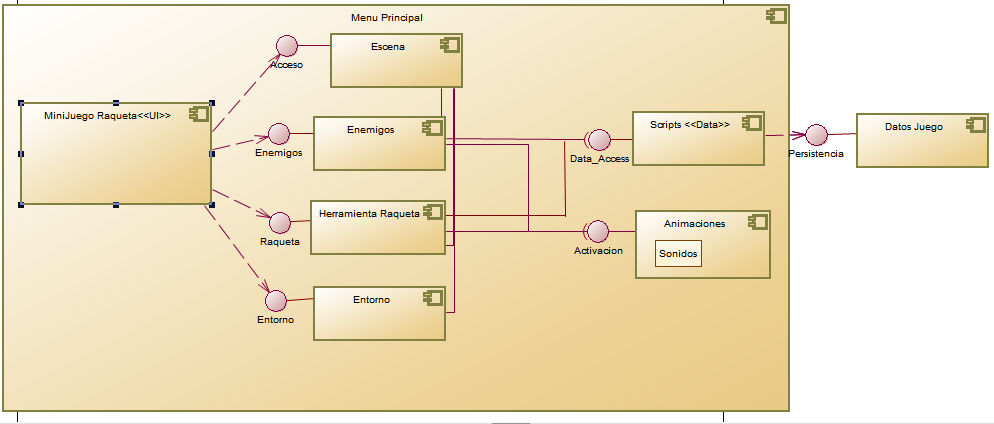


Ilustración 8

**Minijuego Lavadero**

El presente escenario es similar al anterior, no obstante el componente entorno emplea los servicios ofrecidos por los scripts, dado que estos son modificados para representar los cambios en las escenas.

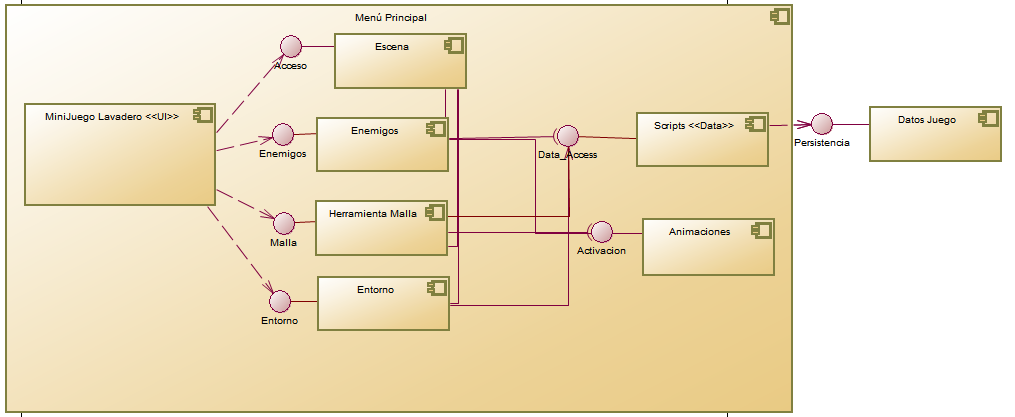


Ilustración 9

**MiniJuego Guppy**

El componente script modifica la escena a fin de definir detalles específicos de las escenas, los elementos restante se comportan muy similar a los anteriores, teniendo en cuenta las dependencias creadas entre la herramienta, los enemigos con respecto a los scripts.

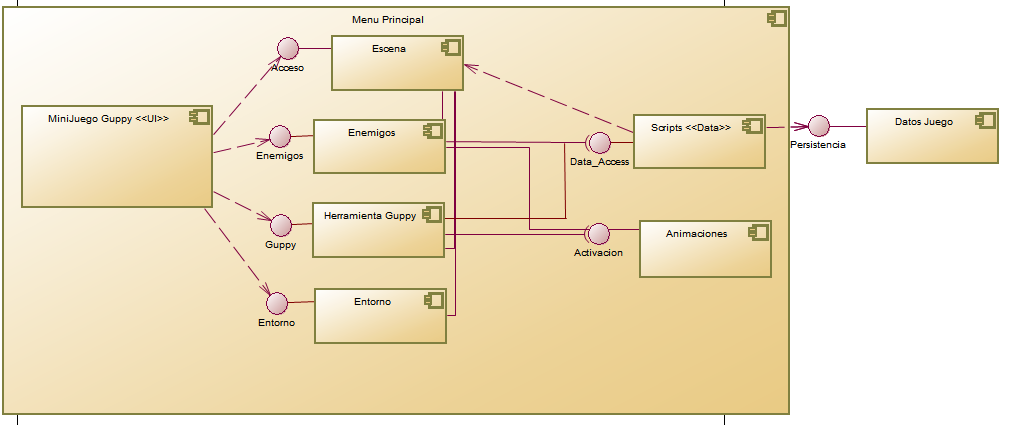


Ilustración 10

**Minijuego Zancudo Volando**

Aquí no hay un enemigo definido, el entorno será el obstáculo que se debe enfrentar por lo tanto se plantea la creación de los modulos que contengan estos. De igual manera se hace la manipulación del entorno por parte de los scripts, y se salvan los datos mediante el componente Datos juego.

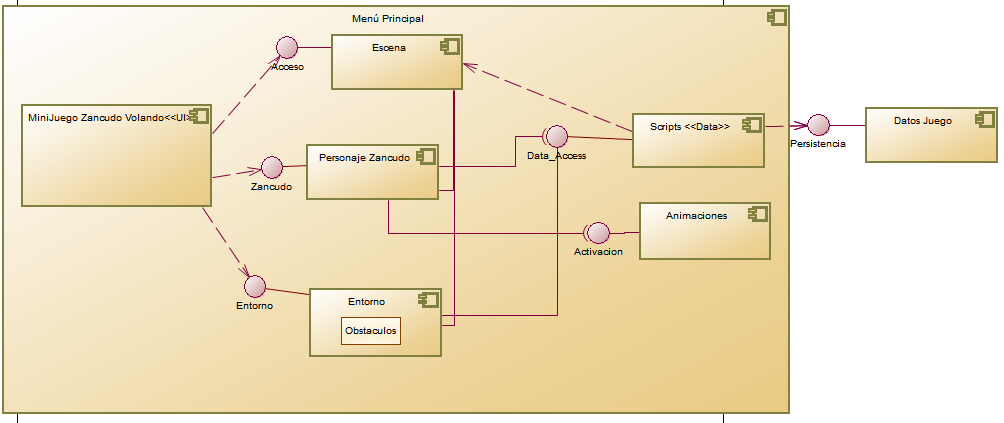


Ilustración 11

## Capas

Se divide el juego en 5 grandes componentes, siendo estas los casos de uso de los minijuegos identificados con el usuario, con lo que se posibilita la expansión del juego al agregar minijuegos como niveles de este. La instalación del juego corresponderá a la instalación completa de los minijuegos que lo componen, para este caso en específico son 5 componentes de instalación.

# Rendimiento

Al implementar esta plataforma de desarrollo según las métricas establecidas por los desarrolladores de Unity, los cuales muestran medidas de rendimiento muy eficientes al emplear una estructura simple y empleando solo los elementos que el usuario está viendo, haciendo referencia a los que podría utilizar o ya utilizo. Además de contar con herramientas para medir la asignación de memoria en el sistema, y de este modo evitar cuellos de botella. Cuenta además con unas herramientas para conectarse a los dispositivos para monitorear el rendimiento durante el juego.

Con esto se pretende obtener el mayor rendimiento con la aplicación, empleando estas herramientas y pruebas de usabilidad que garanticen una fluidez para los usuarios, sin importar la comunidad en la cual se implemente.

# Calidad

Unity contribuye a la portabilidad, ya que es un motor de desarrollo que tiene una gran gama de plataformas de exportación como lo son Web, plataformas móviles, consolas de videojuegos. Esto contribuye a una mejor calidad en uno de los aspectos fundamentales del proyecto, que es la inserción de este en gran medida de la población juvenil, de niños y estudiantes de colegios de zonas en las cuales se identifiquen posible brotes de dengue.