Documento de Arquitectura

de Software (SAD)

**V 1.0.0**

|  |  |
| --- | --- |
|  | **Equipo de desarrollo de software**  **Yensen Limón Priego** |
|  |
|
|
|
|
|
|
|

**Never without you**

Contenido

[**1. Estructura del Documento 5**](#_Toc481868644)

[**1.1 Propósito y alcance del SAD 6**](#_Toc481868645)

[**1.2 Presentación de los Interesados 6**](#_Toc481868646)

[***1.3* Definición de las Vistas y Relación con los *stakeholders* 6**](#_Toc481868647)

[**1.4 Definición de Puntos de Vista.** 7](#_Toc481868648)

[**1.5 Vistas Documentadas 7**](#_Toc481868649)

[**1.6 Documentación de las Vistas 7**](#_Toc481868650)

[**2. Antecedentes de la Arquitectura 8**](#_Toc481868651)

[**2.1 Antecedentes del Problema 8**](#_Toc481868652)

[**2.1.1 Descripción Especifica del Sistema 8**](#_Toc481868653)

[**2.1.2 Objetivos y Contexto 8**](#_Toc481868654)

[**2.1.3 Requisitos significativos 9**](#_Toc481868655)

[**2.2 Antecedentes de la Solución 9**](#_Toc481868656)

[**2.2.1 Enfoque Arquitectónico 9**](#_Toc481868657)

[**2.2.2 Requisitos de Cobertura 9**](#_Toc481868658)

[**2.3 Consideraciones de Reutilización 10**](#_Toc481868659)

[**3. Vistas 10**](#_Toc481868660)

[**3.1 Vista de Descomposición 10**](#_Toc481868661)

[**3.1.1 Descripción de la Vista 10**](#_Toc481868662)

[**3.1.2 Diagrama de la Vista de descomposición 11**](#_Toc481868663)

[**3.1.3 Antecedentes de la Vista de Descomposición 11**](#_Toc481868664)

[**3.1.4 Vista de los módulos de la vista de Descomposición. 11**](#_Toc481868665)

[**3.2 Vistas de Capas de Diseño 11**](#_Toc481868666)

[**3.2.1 Descripción de la vista 12**](#_Toc481868667)

[**3.2.2 Diagrama de la vista de Capas de Diseño 13**](#_Toc481868668)

[**3.2.3 Antecedentes de la Vista de Capas de Diseño 13**](#_Toc481868669)

[**3.2.4 Vista de Capas de Diseño 13**](#_Toc481868670)

[**3.3 Vista de Despliegue 15**](#_Toc481868671)

[**3.3.1 Descripción de la vista de Despliegue 15**](#_Toc481868672)

[**3.3.2 Diagrama de la vista de despliegue 15**](#_Toc481868673)

[**3.3.3 Antecedentes de la Vista de Despliegue 15**](#_Toc481868674)

[**4. Relación entre las Vistas 16**](#_Toc481868675)

[**4.1 Relaciones Generales entre las Vistas 16**](#_Toc481868676)

[**4.2 Relaciones vista a vista 16**](#_Toc481868677)

Lista de Ilustraciones

[**Figura 1. Diagrama de la vista de descomposición. 11**](#_Toc481868491)

[**Figura 2.Diagrama de la vista de capas de diseño 13**](#_Toc481868492)

[**Figura 3. Diagrama de la vista de la capa controlador 14**](#_Toc481868493)

[**Figura 4 Diagrama de la vista de la capa Modelo 14**](#_Toc481868494)

[**Figura 5 Diagrama de la vista de despliegue. 15**](#_Toc481868495)

Listado de Tablas

[**Tabla 1. Interesados en el videojuego “never without you”. 7**](#_Toc481868505)

[**Tabla 2. Prioridades de los interesados. 7**](#_Toc481868506)

# Estructura del Documento

El presente documento describe la arquitectura del videojuego “never without you”. En esta sección describe la organización del presente documento.

En la Sección 1 se introduce al lector el concepto de arquitectura de software, y en las sub-secciones incluyen:

* Sección 1.1 (Propósito y Alcance del SAD) explica el propósito y el alcance de este SAD, así como también indica cuál información ha sido incluida o excluida.
* Sección 1.2 (Presentación de los interesados) explica a los interesados a quienes, en particular, va dirigido el SAD.
* Sección 1.3 (Definición de las vistas) explica las vistas usadas en este SAD. Para cada una, definida en la sección 1.3, existe una vista correspondiente definida en la sección 3 (Vistas).
* Sección 1.4 (Vistas documentadas) explica la organización estándar usada para documentar vistas arquitectónicas en este SAD.

La Sección 2 describe las restricciones que provee la influencia significativa sobre la arquitectura; describe la función general y propósito para el sistema o subsistema cuya arquitectura se describe en este SAD; describe también las metas y factores importantes para la arquitectura de software, los roles de la arquitectura, la relación hacia los resultados y componentes de ingeniería de sistemas y demás factores relevantes; describe requisitos de comportamiento y atributos de calidad que moldean la arquitectura de software; provee una descripción de las decisiones importantes de diseño, profundizando en los requisitos propuestos en la arquitectura de software.

En la Sección 3 del actual documento se especifican detalladamente las vistas correspondientes al videojuego “Never without you”.

Finalmente, la Sección 4 señala la importancia de la relación entre las vistas empleadas en este documento con la perspectiva del sistema.

## Propósito y alcance del SAD

La arquitectura del videojuego “never without you”, se basa en cuatro vistas arquitectónicas, las cuales muestran los componentes, módulos, funcionalidades e interfaces propuestas, mostrando una guía de diseño de la arquitectura del software antes de iniciar con la construcción del videojuego.

Este documento tiene como propósito mostrar la relación que existe entre los diversos componentes que forman parte del videojuego, a través de vistas y elementos que contiene cada una de ellas.

Las vistas identificadas para el videojuego que se documentarán en las siguientes secciones son:

* Descomposición
* Capas de Análisis
* Capas de Diseño
* De Despliegue

## Presentación de los Interesados

En este apartado se muestra los principales *stakeholders* identificados, que interactuarán con el sistema, los cuales son:

Usuario administrativo, gerente, analista QA, arquitecto de software, desarrollador de software, y líder del proyecto del videojuego. En la Sección 1.3 se detallan los intereses de cada uno de ellos.

## Definición de las Vistas y Relación con los *stakeholders*

En la Tabla 1, se muestran los usuarios involucrados con respecto al Never without you.

|  |  |
| --- | --- |
| Stakeholder | Inquietud |
| Usuario administrativo | Personas interesadas de que el sistema sea fiable y disponible cuando se necesite. |
| Gerente, Director | Preocupados (además de los costos y tiempos) de que la arquitectura permita a los equipos trabajar de forma independiente en gran medida, interactuando de manera disciplinada y controlada |
| Analista QA | Interesados en que las estrategias para alcanzar todos esos objetivos sean adecuadas. |
| Arquitecto del software | Responsables de conocer y administrar los recursos de hardware y software, el aseguramiento de la información, para el buen funcionamiento del videojuego*.* |
| Desarrollador | Responsables de la reutilización de los componentes de hardware y software, y el desarrollo de los nuevos componentes |
| Líder de Proyecto | Encargado de administrar y controlar los recursos de manera que permita alcanzar el objetivo planteado. |
| Diseñador | Visualizan y conceptualizan el videojuego, según la visión del productor y sus creadores. Documentan cada paso del proceso, al que dan seguimiento con descripciones y narraciones detalladas. No los crean, solo los describen. |
| Área de sonido | Musicaliza y agrega cada sonido de cada elemento del juego. |
| Área de ventas y mercadotecnia | Se encarga de difundir y promocionar el videojuego. |

Tabla 1. Interesados en el videojuego “never without you”.

## Definición de Puntos de Vista.

En la Tabla 2 se muestra la relación de la prioridad entre los interesados con cada una de las vistas.

Entiéndase por nivel 5 al mayor grado de prioridad y relación con la vista, así sucesivamente el nivel decrementa hasta el nivel 1, que representa el menor grado de prioridad y relación con la vista, en nivel 1 en este caso no aplica para la tabla 2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Interesado  Vista | Vista de Componentes | Vista de Despliegue | Vista de Descomposición | Vista de Capas y Usos |
| Desarrollador de SW | 5 | 5 | 4 | 5 |
| Líder de Proyecto | 2 | 4 | 5 | 5 |
| Arquitecto de SW | 5 | 5 | 3 | 5 |
| Analista de QA | 3 | 2 | 4 | 2 |
| Usuario Administrativo | 3 | 2 | 5 | 3 |

Tabla 2. Prioridades de los interesados.

## Vistas Documentadas

Para el videojuego, en este documento se han documentado las siguientes Vistas Arquitectónicas:

* Descomposición
* Capas de Análisis
* Capas de Diseño
* De Despliegue

## Documentación de las Vistas

De las vistas que se seleccionaron para la especificación de la arquitectura del videojuego se documentaran para cada una de ellas su descripción, los antecedentes de la arquitectura, el diagrama grafico de la representación de la vista y la relación con sus componentes, así como la descripción de cada uno de los elementos de la vista.

# Antecedentes de la Arquitectura

En esta sección se especifica cuáles fueron las limitantes, objetivos, y atributos de calidad del videojuego “Never without you” que permitieron llegar a una especificación de la arquitectura, incluyendo la solución para las limitantes en mención y justificando las elecciones arquitectónicas.

## Antecedentes del Problema

Las limitaciones para desarrollar el videojuego y que contribuyeron para llegar a la actual especificación de la arquitectura fueron las siguientes:

* Utilizar una herramienta que permita el desarrollo de videojuegos para plataformas 2D.
* El soporte que cuenta la herramienta unity3d en la web, así como la comunidad de la misma.
* Existe un gran mercado llamado assetstore la cual vende módulos para funcionalidad específica dentro del videojuego.
* Utilización de un motor unity3d para el desarrollo de videojuegos 2D que cuenta con lenguaje de programación orientada a objetos como C#.
* Utilización de la tecnología existente de los videojuegos de plataformas en 2D.
* Integración de componentes de audio y visuales dentro de la herramienta unity3d

### **Descripción Especifica del Sistema**

Elvideojuego *never without you* es un juego 2D del genero de plataformas y puzles. Su es propósito resolver puzles que se presenten dentro de cada escena y que te ayude a entre, estos estarán definidos como objetivos. Una vez recogidos todos estos ítems se podrá pasar al siguiente nivel. Al usuario le aparecerá algún obstáculo como un árbol, piedra, etc. El usuario tendrá que resolver objetivos que entre cada vez con mayor dificultad.

### **Objetivos y Contexto**

De acuerdo a las limitantes y funcionalidades descritas previamente para el videojuego, el presente documento de especificación de la arquitectura del software tiene como objetivo proporcionar el marco de referencia para guiar la construcción del software, para establecer las vistas, sus elementos y la relación que exista entre ellos, así como establecer los fundamentos para que el equipo de desarrollo del videojuego trabaje en una línea común, con la finalidad de lograr los objetivos del sistema.

### **Requisitos significativos**

Dentro de los Atributos de calidad sugeridos para la utilización del motor Unity3D se encuentran*:*

* **Facilidad de modificación.** Que la modificación y la adaptación de los módulos sea fácil.
* **Modularidad.** Que la independencia funcional de los componentes sea alta.
* **Tolerancia al error**. Que en cierta medida el sistema pueda controlar errores, para que estos no causen daños mayores.
* **Independencia del hardware.** Que el grado del videojuego sea independiente al del hardware sobre el que se opera, sea alto.
* **Simplicidad.** Que el grado en el que el videojuego sea entendido, sea alto.
* **Uso.** Que el videojuego sea fácil de entender sin necesidad de ayuda o asesoría.

## Antecedentes de la Solución

En esta sección se describe la arquitectura seleccionada que pretende cubrir los objetivos del videojuego *“never without you”,* así como los atributos de calidad descritos anteriormente.

### **Enfoque Arquitectónico**

Para poder garantizar los requerimientos arquitectónicos del videojuego “*never without you*”, de acuerdo a los atributos de calidad contemplados, se plantean los siguientes enfoques:

* Lectura previa de las diferentes vistas del sistema.
* Análisis de los requerimientos.
* Identificación de los stakeholders que apoyaran en la fase de desarrollo, para poder relacionarlo a una vista.
* Diseño de la Matriz de relaciones de vistas y stakeholders.
* Lluvia de ideas para obtener escenarios y validar los atributos de calidad.

### **Requisitos de Cobertura**

Dentro de la especificación de la arquitectura, los requisitos esenciales a definir son los siguientes:

* Diseño basado en componentes de propósito claro y con alto grado de cohesión.
* Desacoplamiento entre componentes, que permita el fácil remplazo de los mismos.

## Consideraciones de Reutilización

Las funcionalidades relacionadas a los módulos candidatos a reutilizarse son las siguientes:

* Funcionalidad de la pantalla inicio.
* Funcionalidad de la pantalla menú.
* Funcionalidad del personaje (animaciones del personaje y su controlador).
* Funcionalidad de los enemigos.
* Funcionalidad del núcleo del juego (objetivos del nivel, control de animaciones y sonidos).

# Vistas

Esta sección contiene las vistas de la arquitectura del videojuego, las cuales muestran un conjunto de elementos arquitectónicos de software que están presentes en un sistema, sus propiedades y las relaciones entre ellos.

En la siguiente tabla se describen las vistas que se documentan en el videojuego:

|  |  |
| --- | --- |
| Vista | Descripción |
| Descomposición | Muestra las responsabilidades del videojuegoparticionadas a través de módulos y sub-módulos. |
| Análisis | Muestra de manera conceptual las dependencias entre las funcionalidades del nivel específico con los módulos de nivel general del videojuego*.* |
| Clases | Muestra las clases del videojuego, así como sus respectivas asociaciones entre clases. |
| Diseño | Muestra los elementos del videojuego, dividido en 3 capas: Presentación, Control y Datos. |
| Despliegue | Muestra los componentes y conectores identificados del videojuego que viven en el hardware en el cual se ejecuta el software. |
| Esquema de Datos | Muestra las entidades de datos del videojuego, así como sus respectivas asociaciones entre entidades. |

Tabla 3.Descripción de las vistas a desarrollar

## Vista de Descomposición

En esta sección se describe la vista de descomposición, su diagrama y componentes que se han identificado para el videojuego “never without you”*.*

### **Descripción de la Vista**

Esta vista se caracteriza por mostrar cómo las responsabilidades del sistema son particionadas a través de los módulos, los cuales son descompuestos en sub-módulos. La Figura 1 muestra la vista de descomposición.

### **Diagrama de la Vista de descomposición**

Figura 1. Diagrama de la vista de descomposición.

### **Antecedentes de la Vista de Descomposición**

Los arquitectos tienden a atacar un problema con la técnica de *divide y vencerás*, y una vista producida en este estilo nos recuerda a esas cruzadas. Tomando los elementos y las propiedades del tipo de vista por módulos y enfocándose en la relación *parte de*, obtenemos el estilo de vista de descomposición.

### **Vista de los módulos de la vista de Descomposición.**

Los módulos generales que se muestran en la vista de descomposición son:

* Modulo intro presenta el slash del videojuego donde el usuario presiona la pantalla y permite mostrar una presentación del videojuego.
* Modulo menú, este módulo contiene sub-módulos que permiten iniciar la partida, cargar nivel y salir.
* Modulo partida, este módulo contiene sub-módulos que permiten contralar el del videojuego.
* Modulo niveles presenta la lista de niveles desbloqueados del videojuego.
* Módulo lógica del juego, este módulo contiene los controladores del videojuego, de los niveles, animaciones de los objetos, eventos táctiles de cada nivel, así como plugins necesarios para su funcionamiento.

**.**

## Vistas de Capas de Diseño

En esta sección se documenta la descripción, elementos y su interacción en las tres capas de la vista de capas de diseño para el videojuego “Never without you”.

### **Descripción de la vista**

La vista de capas ayuda a estructurar las aplicaciones que pueden ser descompuestas en distintos grupos de acuerdo a las tareas de cada grupo o de acuerdo a un particular nivel de abstracción.

En el [diseño](http://es.wikipedia.org/wiki/Diseño) de [sistemas informáticos](http://es.wikipedia.org/wiki/Sistema_informático) actual se suele usar las [arquitecturas multinivel](http://es.wikipedia.org/wiki/Arquitectura_software) o programación por capas. En dichas arquitecturas a cada nivel se le confía una misión simple, lo que permite el diseño de arquitecturas escalables (que pueden ampliarse con facilidad en caso de que las necesidades aumenten).

En el caso de la siguiente vista los elementos del sistema se agrupan en tres capas, ya que son las que se han considerado pertinentes de acuerdo a la tarea que realiza cada elemento del sistema.

1.- Capa de presentación: es la que ve el jugador, presenta el sistema al jugador, le comunica la información como sonidos e imágenes y captura la información del jugador dando un mínimo de proceso. Esta capa se comunica únicamente con la capa de control.

2.- Capa de Control: es donde residen los [programas](http://es.wikipedia.org/wiki/Programa) que se ejecutan, recibiendo las peticiones del jugador y enviando las respuestas tras el proceso. En esta capa se incluyen las operaciones del videojuego pues es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de [base de datos](http://es.wikipedia.org/wiki/Base_de_datos) para almacenar o recuperar datos de él.

3.- Capa de datos: es donde residen los datos y se reciben solicitudes de almacenamiento o recuperación de información desde la capa de Control. En esta capa se guarda la información e los niveles desbloqueados del jugador.

### **Diagrama de la vista de Capas de Diseño**



Figura 2. Diagrama de la vista de capas de diseño

### **Antecedentes de la Vista de Capas de Diseño**

Se decidió mostrar la arquitectura del videojuego, desde esta vista, ya que se considera que es una vista que permite mostrar los distintos módulos del videojuego de acuerdo a las tareas que realizan.

### **Vista de Capas de Diseño**

En esta sección se describen las diferentes capas que componen la vista de capas y usos.

#### **Capa de Presentación**

Esta capa contiene elementos que son los que directamente tienen interacción con el jugador, en este caso el dispositivo móvil, ya que por este medio el usuario interactúa con el videojuego never without you.

#### **Capa de Control**

En la capa de control se agrupan las utilizadas dentro del videojuego, en este caso para son utilizadas para responder a eventos dentro del videojuego, como lo es el controlador del personaje, la detección de la pantalla táctil del dispositivo, colisiones, disparadores, corrutinas, comportamiento de la cámara, la lógica de los ítems, así como del juego, en la Figura 3 se puede observar la capa.



Figura 3. Diagrama de la vista de la capa controlador

CamaraController: se encargada de adaptar la resolución del videojuego, así como de los movimientos que esta pueda presentar.

InteractivoController: se encarga de controlar los eventos del control virtual del dispositivo, esto con el fin de detectar cuando el jugador toca el botón en la pantalla.

LogicaItems: se encarga de obtener los eventos dentro de los ítems en el juego como disparadores, colisiones, sonidos y animaciones.

LogicaJuego: es encargada de la lógica de los escenarios como eventos animaciones en el escenario y en detectar eventos táctiles.

PlayerController: es encargada de la lógica del personaje principal, así como de reproducir animaciones y físicas que tiene el personaje.

#### **Capa de Datos**

En esta capa los datos del sistema son almacenados en una base de datos única, para la cual se utiliza el motor de base de datos SQL lite. En la Figura 4 se puede observar la capa del modelo



Figura 4. Diagrama de la vista de la capa Modelo

**Escenario**: representa los niveles que han pasado el usuario y cuales tienes disponibles para que puedan visualizarse dentro del videojuego.

#### **Catálogo de Elementos**

* **Capas:** Las capas están definidas de acuerdo a la tarea que cubre cada módulo o elemento dentro de ella, por ejemplo, en la de control se encuentran los que son necesarios para la operación del videojuego.
* **Relaciones:** Las relaciones que se manejan, son de dependencia y uso.
* **Los Nombres de las Capas:** Estos están dados por la tarea que cumplen los elementos de cada capa.
* **Coherencia entre capas:** En este caso las capas tienen un nivel medio de cohesión, ya que aunque dependen unas de otras, en algunos casos los elementos tienen acoplamiento débil.

## Vista de Despliegue

En esta sección se documentan cómo interactúan los componentes, hardware y software identificados para la vista de despliegue del videojuego.

### **Descripción de la vista de Despliegue**

En la vista de despliegue, se presentan los elementos para la ejecución del videojuego. Además se documenta cómo interactúan los componentes tanto de hardware como de software.

### **Diagrama de la vista de despliegue**



Figura 5. Diagrama de la vista de despliegue.

El diagrama de despliegue mostrado en la Figura 5 define la arquitectura en la que el videojuego se ejecuta. *Never without you* será desarrollado con el motor de videojuegos Unity3d que se compila en su propia plataforma y exportado a un dispositivo móvil. El usuario final empleará un dispositivo móvil.

### **Antecedentes de la Vista de Despliegue**

La elección de la arquitectura está fuertemente ligada de acuerdo a las necesidades del motor unity3d, debido a que ya cuentan con un área de desarrollo, soporte, una amplia comunidad, servicios como la assetstore que cuenta con código y módulos ya probados e implementados por terceros, modelos 3d, así como arte sonora.

#### **Catálogo de Elementos**

* Dispositivo móvil tableta android.

##### **Elementos**

* El dispositivo móvil debe de contar con un sistema operativo android superior a la versión 4.3

# Relación entre las Vistas

En esta sección se describen las principales relaciones que existen entre las vistas abordadas en la Sección 3 (vista de descomposición, vista de capas de diseño y vista de despliegue).

## Relaciones Generales entre las Vistas

Cada una de las vistas especificadas en la sección 3 provee una perspectiva diferente y manejo de diseño en un sistema, así también, cada una es válida y útil en su manera. Aunque las vistas dan diferentes perspectivas, no son independientes.

La vista de descomposición permite mostrar las responsabilidades del sistema particionadas a través de módulos y sub-módulos, de igual manera la vista de capa de análisis pretende identificar las principales funcionalidades agrupándolas en módulos, identificando componentes serán los que se podrán reutilizar a futuro. La vista de capa de diseño permite identificar los elementos del software de acuerdo a su funcionalidad, permitiendo generar así un flujo de comunicación entre dichos elementos. La vista de despliegue muestra a su vez la relación de los componentes que pudieron quizás identificarse en las otras vistas con el hardware y software a ocupar.

## Relaciones vista a vista

Vista de descomposición

* Capa de análisis, son muy parecidas salvo que la vista de descomposición muestra de manera más concreta las funcionalidades agrupadas de acuerdo a los paquetes de los casos de uso generados en la fase de requerimientos. Sin embargo, en la vista de capa de análisis se agrupan las funcionalidades del sistema de acuerdo a su operación, se pueden relacionar dichas funcionalidades en módulos, de esta manera se podrá reutilizar el código para diferentes funcionalidades sin importar si estas cambian o no, el modulo no se verá afectado.

Vista de capa de análisis

* Capa de diseño, con la documentación de la vista de capa de análisis permite identificar los componentes que participarán en la vista de diseño. La capa de análisis brinda un panorama claro de los componentes que van a requerir para el desarrollo de las funcionalidades listadas.
* Despliegue, de igual manera, la capa de análisis brinda un panorama de que elementos de hardware y software serán los que van a requerir para el desarrollo de las funcionalidades listadas.

Vista de capa de diseño

* Despliegue, se encontró una relación muy fuerte con los componentes que se describen de manera detallada en la vista de capa de diseño, evidenciando que la vista de despliegue es más abstracta, y donde lo relevante es conocer los equipos físicos o lógicos y el software para el funcionamiento del videojuego.