**DOCUMENTO DE ARQUITECTURA**

**DIEGO ALEJANDRO GONZALEZ GUALTEROS**

**CRISTIAN ANDRES CASTELLANOS FINO**

**EDUARDO OSPINA MEJIA**

**ADVENTURE MAP**

# Introducción

Dentro del desarrollo de Adventure Map, se llegaron a plantear varios requisititos que se necesitarían para el desarrollo del proyecto. Al ser este un proyecto trabajo por un grupo, se requieren plantear correctamente todas las decisiones de arquitectura para permitir el desarrollo funcional de la aplicación.

## Propósito

*Este documento proporciona un resumen general sobre la arquitectura de la aplicación pensada como proyecto final de la materia Arquitectura de Software. La idea del proyecto es poder crear un mapa con múltiples jugadores y monstruos donde estos puedan interactuar para peleas. Se tiene planteado la implementación con real time y batallas de manera concurrente. Se desea tener una implementación a partir de los servicios de Azure, y se debe tener en cuenta los atributos de calidad descritos mas adelante para la arquitectura en su totalidad.*

## Audiencia

*Diego Alejandro González Gualteros (Arquitecto-Estudiante)*

*Cristian Andrés Castellanos Fino (Arquitecto-Estudiante)*

*Eduardo Ospina Mejía (Arquitecto-Estudiante)*

*Diego Andrés Triviño Gonzalez (responsable de Aseguramiento de Calidad)*

***1.3. Alcance***

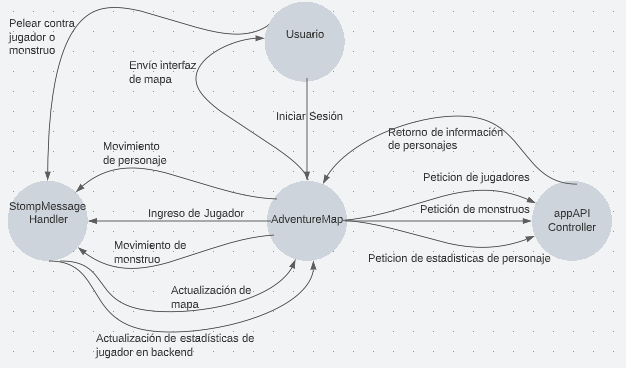
*El Documento de Arquitectura abarca la definición del proyecto a través de las vistas de casos de uso, lógica (análisis y diseño) e implementación, atributos de calidad, requerimientos, restricciones y descripciones detalladas. De igual manera, se va a revisar el ambiente de despliegue al cual va a ser sometido el proyecto. Esto con la finalidad de que se pueda asegurar los atributos de calidad pedidos por el responsable de Aseguramiento de Calidad y se permita entender en su totalidad la arquitectura a desarrollar.*

# Arquitectura del Producto/Sistema

## Vista Funcional

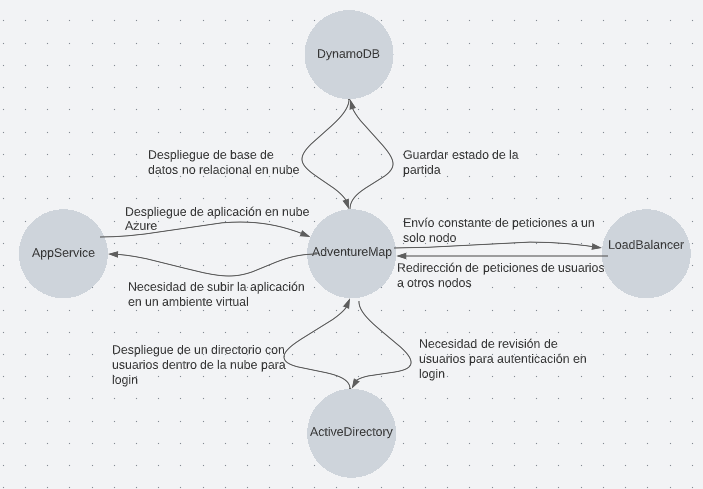


### Modelo de Contexto lógico

**

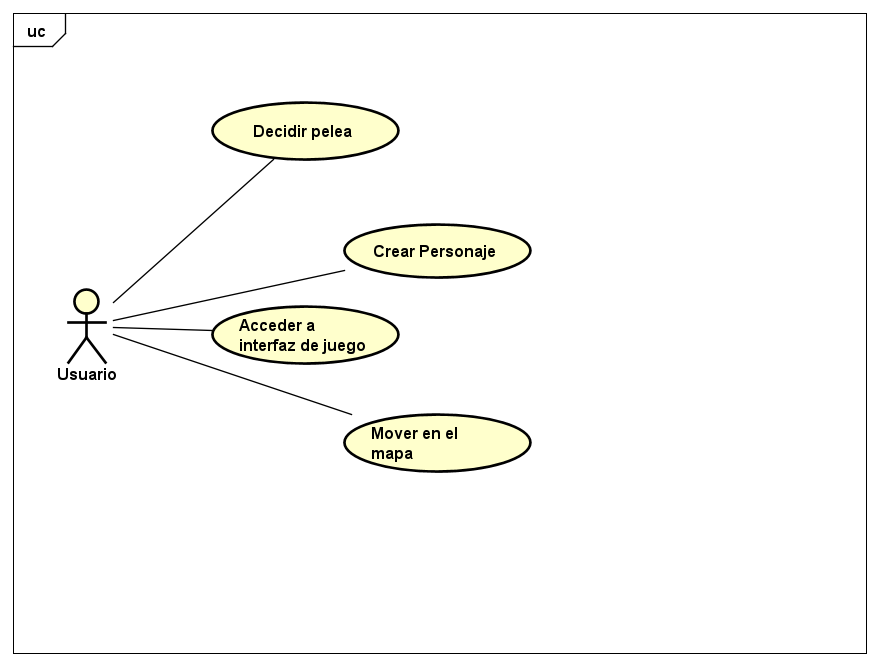
*Diagrama de Contexto lógico*

* + 1. **Modelo de contexto implementación Cloud (Azure)**

**

*Diagrama de Contexto Servicios Azure*

* + 1. **Modelo de Casos de Uso**

**

*Diagrama Casos de Uso*

***C*ontexto de casos de Uso**

* COMO usuario QUIERO decidir el punto de acción PARA poder revisar si se pelea o se huye de algún monstruo o usuario
* COMO usuario QUIERO crear personaje PARA poder entrar a la interfaz de juego
* COMO usuario QUIERO acceder a la interfaz de juego PARA poder interactuar con los demás usuarios y monstruos
* COMO usuario QUIERO acceder a la interfaz de juego PARA poder interactuar con los demás usuarios y monstruos
  + 1. Modelos de Secuencia

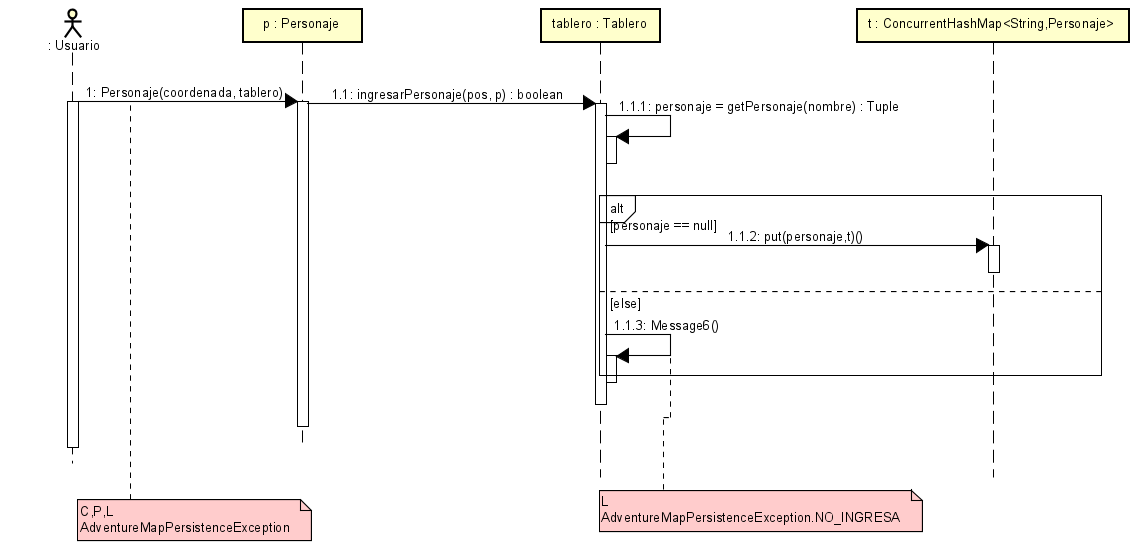


Diagrama de Secuencia Creación de Jugador

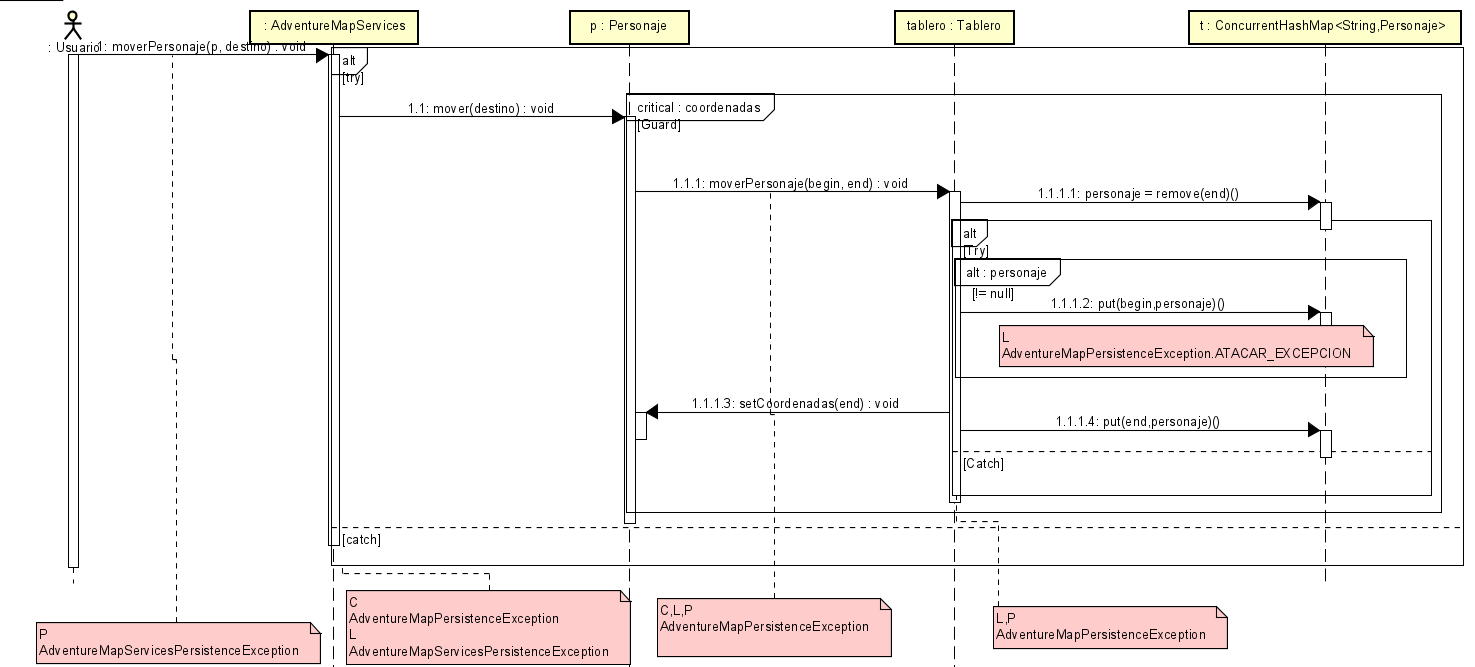
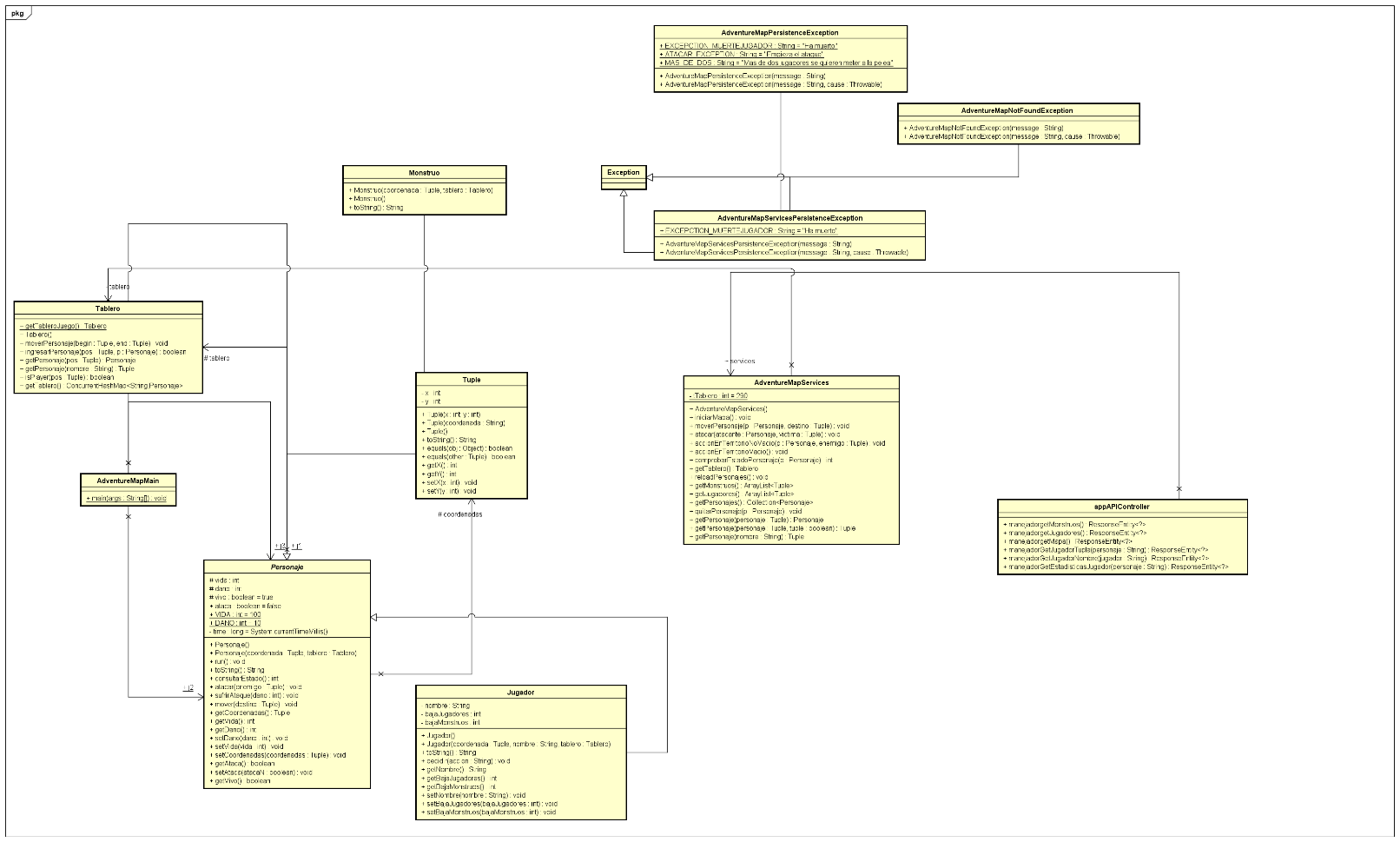


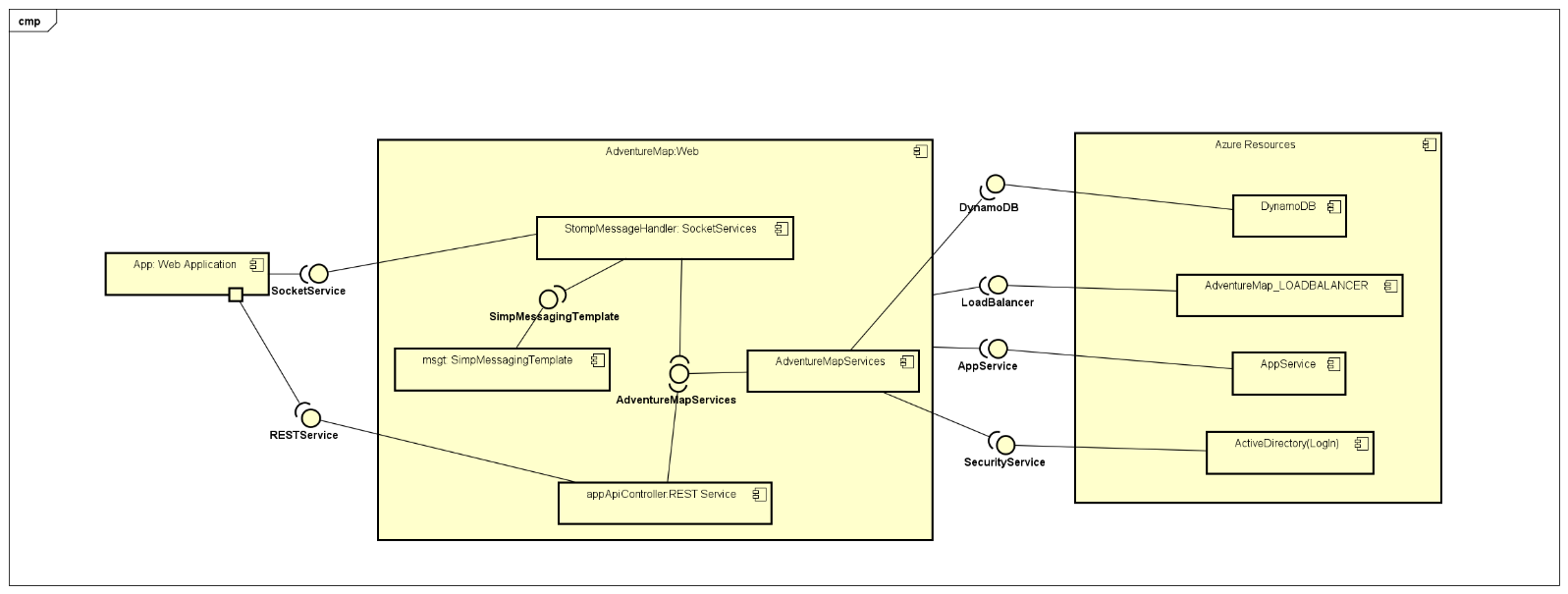
Diagrama de Secuencia Movimiento de Jugador

## Vista Lógica

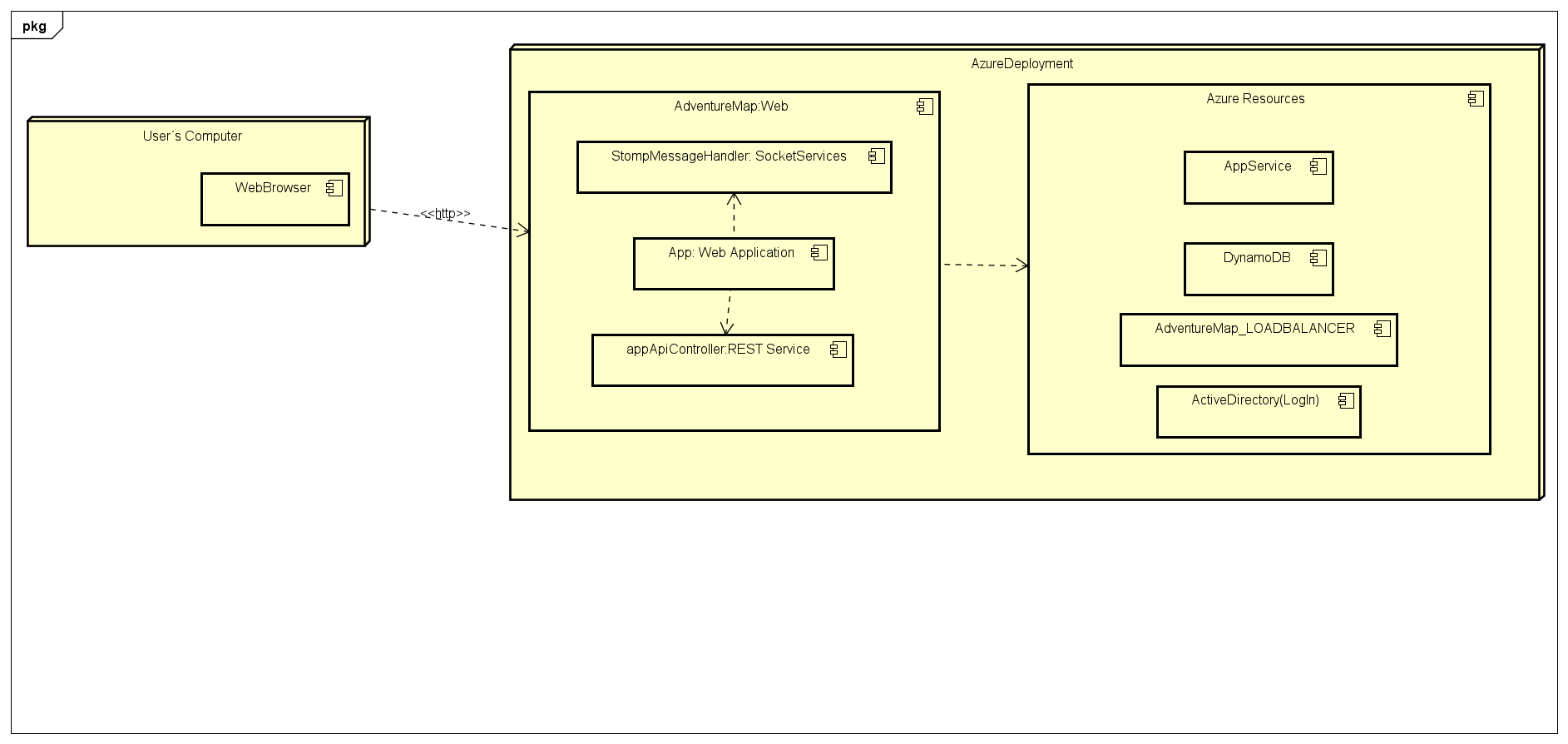
### Paquetes de Diseño Arquitectónicamente Significativos



## Vista de Implementación - Componentes

**

## Vista de Despliegue

******

**Diagrama de Despliegue**

## Requisitos de Software/Hardware

*Equipamiento y Configuración:*

*Requisitos de dispositivo de usuario:*

* *Sistema Operativo* Windows, Linux o MAC

*Requisitos de componentes*

* *Modalidad de alojamiento de funcionalidad: Cloud (Azure)*

# Atributos de Calidad



Para este proyecto se van a manejar 3 atributos de calidad específicos:

Seguridad:

• Se debe especificar un escenario de calidad asociado al atributo de Seguridad, el equipo  
es libre de elegir cual escenario implementar.  
• El escenario de calidad debe ser aprobado por el profesor.  
• Se debe actualizar la arquitectura para validar que cumple con el escenario anteriormente  
descrito.  
• Se debe realizar la implementación que solucione el escenario anteriormente descrito.

Disponibilidad:

• Se debe plantear una arquitectura que permita escalar horizontalmente la aplicación y  
hacer una implementación de concepto de dicha arquitectura, no deberían estar todos  
los elementos en el mismo servidor, para esta funcionalidad debe especificar un escenario  
de calidad que refleje el atributo de Disponibilidad – Escalabilidad del sistema.  
• Se debe agregar como restricción tecnológica el uso de la plataforma Azure para la  
ejecución del proyecto, se pude apoyar en máquinas virtuales.  
• Hacer una implementación de concepto de dicha arquitectura, la prueba de concepto  
debe ser, físicamente, lo más distribuida posible. Es decir, montar todos los componentes  
de la arquitectura en un mismo servidor perjudicará la evaluación.

Mantenibilidad:

• Se debe especificar un escenario de calidad asociado al atributo de Seguridad, el equipo  
es libre de elegir cual escenario implementar.  
• El escenario de calidad debe ser aprobado por el profesor.  
• Se debe actualizar la arquitectura para validar que cumple con el escenario anteriormente  
descrito.  
• Se debe realizar la implementación que solucione el escenario anteriormente descrito.

• Se debe especificar un escenario de calidad asociado al atributo de Mantenibilidad, en el  
cual se especifiqué que la aplicación va a ser sometida a una revisión de código estático,  
esto se debe incluir dentro del ciclo de integración continúa, este proceso en DevOps es  
conocido como Inspección continúa, en el siguiente enlace puede encontrar información  
adicional al respecto:  
o https://www.devopsschool.com/blog/what-is-continuous-inspection/  
• Se debe implementar dentro del ciclo de Integración Continúa un paso para ejecutar el  
escaneo del código fuente por medio de SonarQube, para esto es necesario que  
investigue la forma en que su herramienta de IC realiza la integración con SonarQube.  
• El profesor suministrará la guía para la implementación del SonarQube para la correspondiente  
inspección, por favor solicítela cuando la vaya a implementar.  
• Bono: Si logra que las todas las métricas de SonarQube se encuentren en estado “A” (Verde)  
tendrá una bonificación de 5 décimas sobre la nota del proyecto.

1. **Requisitos:**

Para el desarrollo de esta arquitectura requiere varios requerimientos que permiten a la aplicación funcionar como se desea. Para empezar este el requerimiento del real time en donde los jugadores puedan ver el movimiento de los monstruos y otros jugadores. Esto para permitir una experiencia buena de jugabilidad dentro del uso de la aplicación. A parte del real time, se debe implementar concurrencia con todos sus elementos para que no se generen condiciones de carrera ni deadlocks dentro del juego al momento de las batallas contra los monstruos y otros jugadores que podrían afectar de manera negativa la forma en como se juegan las batallas.

Dentro de los requerimientos de funcionalidades que debe tener el juego tenemos en cuenta las principales que son: los movimientos de los monstruos automáticamente dentro del mapa, la capacidad de cada jugador de moverse de un cuadro a un cuadro en el mapa dependiendo de la dirección a la que quieran ir, que el jugador pueda iniciar batallas en contra de monstruos otros jugadores, pero el monstruo no debe tener la capacidad de iniciar batallas contra otros monstruos o jugadores. Debe estar implementado el sistema de batalla mixto, contando con la habilidad del jugador de entrar en batalla con otros jugadores y monstruos. El jugador debe poder atacar al enemigo y los datos deben actualizar en tiempo real, el jugador puede recibir daño y perder la batalla. Los monstruos atacarán cada 2 ataques que el jugador le realice y le infringirán daño al jugador continuamente. Se debe modificar los datos en el backend para llevar a tener los datos relacionados con cada juego que se inicie.

1. **Restricciones:**

Las restricciones que aplican para este proyecto son:

* La mayoría de componentes deben ser alojados en la nube.
* Se debe manejar persistencia únicamente al momento que el juego esta activo y un apartida se encuentra en movimiento.
* Debe permitir mas de 1 jugador entrar al tablero por cada juego.
* Deben generase únicamente 5 monstruos al inicio de la partida.
* Debe funcionar correctamente el realtime y la concurrencia.
* Se debe utilizar tópicos para el manejo de datos entre jugadores.

1. **Supuestos:**

* Para poder hacer login con el active directory, el usuario necesita tener un acuenta activa en el directorio activo generado.
* Solo se puede acceder con un jugador por pestaña de navegador.
* f

1. **Conclusiones**

Se logre completar la arquitectura que se desea utilizar para la creación de Adventure map, indicando su funcionamiento, como planeamos hacer uso de los recursos disponibles y las reglas que debemos seguir para su creación. Se busco generar un documento especializado que contenga toda la información del proyecto y su desarrollo. aaaa