**ENTREGA FINAL PROYECTO DE AULA**

**Institución Universitaria Politécnico Grancolombiano**

**Integrantes:**

Christian Gonzalez

Melisa Gómez

Richard Hermes

Andrés Guevara

**Mayo 26 de 2017**

**Bogotá - Colombia**

Contenido

[Introducción 4](#_Toc483581321)

[Modelado de Negocio CANVAS 4](#_Toc483581322)

[Descripción CANVAS 5](#_Toc483581323)

[Análisis del entorno empresarial y contexto empresarial 6](#_Toc483581324)

[Objetivos 6](#_Toc483581325)

[Perspectiva del producto 6](#_Toc483581326)

[Documento de levantamiento de requerimientos 7](#_Toc483581327)

[Análisis de riesgos 7](#_Toc483581328)

[Identificación de riesgos 7](#_Toc483581329)

[Evaluación de los riegos 8](#_Toc483581330)

[Grafica de evaluación de riesgos 9](#_Toc483581331)

[Planes y acciones a tomar 9](#_Toc483581332)

[Diagrama de casos de uso 11](#_Toc483581333)

[Diagrama de clases 12](#_Toc483581334)

[Diagrama de secuencia 13](#_Toc483581335)

[User Story Map y Release Planning 13](#_Toc483581336)

[Patrones de diseño 13](#_Toc483581337)

[Abstract factory 14](#_Toc483581338)

[Constructor 15](#_Toc483581339)

[Singleton 16](#_Toc483581340)

[Decorador 17](#_Toc483581341)

[Observador 18](#_Toc483581342)

[Herramientas y frameworks: 19](#_Toc483581343)

[Librería 19](#_Toc483581344)

[Framework 19](#_Toc483581345)

[Pruebas y especificaciones 20](#_Toc483581346)

[Planeación de pruebas 20](#_Toc483581347)

[Pruebas de carga 20](#_Toc483581348)

[Especificación del servidor 20](#_Toc483581349)

[Especificación de Ancho de Banda 20](#_Toc483581350)

[Conclusiones 21](#_Toc483581351)

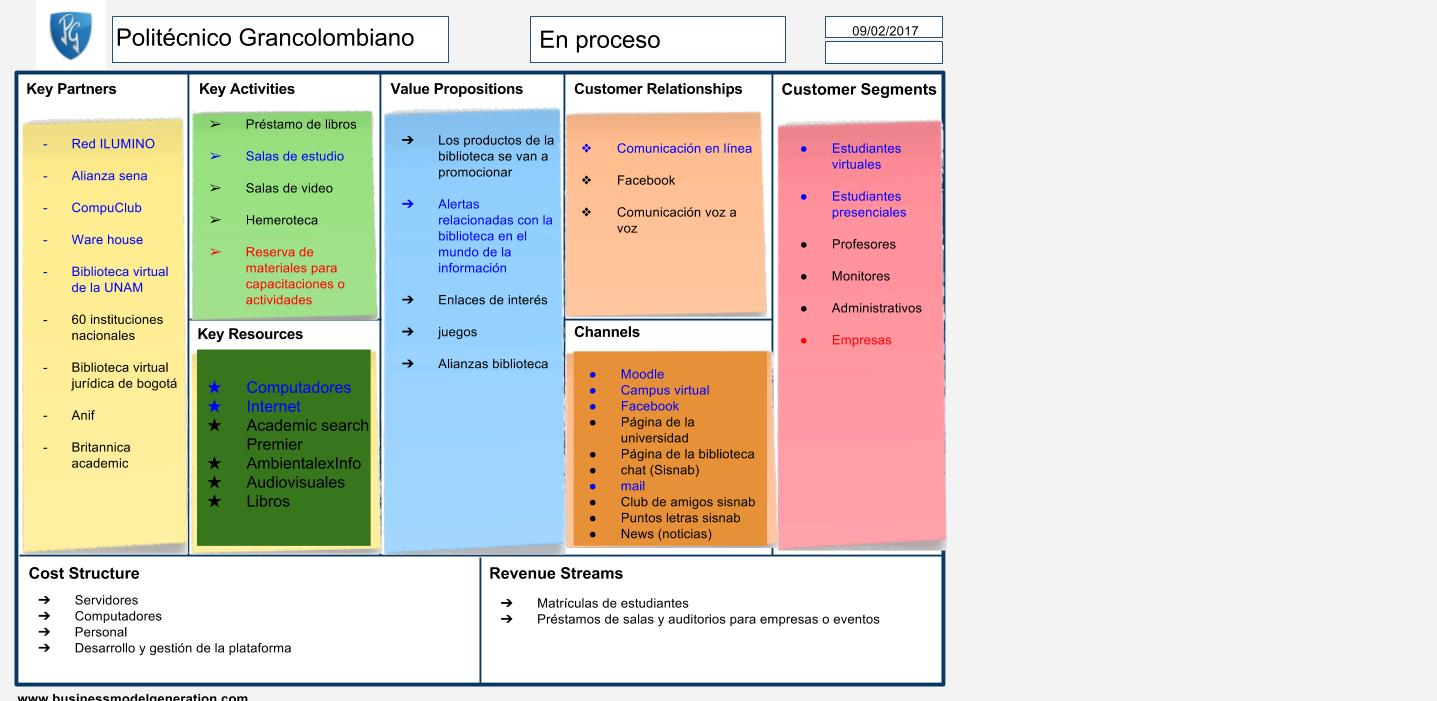
[Referencias Bibliográficas 21](#_Toc483581352)

# Introducción

En el presente documento se evidencia el entregable final del proyecto de aula para la materia Ingeniería del software II construido a lo largo del semestre, el seguimiento en la construcción del producto Bibliogame el cual es una solución al problema generado en la biblioteca de la universidad Politécnico Grancolombiano. El tema es la desinformación de los servicios que se ofrecen dentro de la misma y la falta de interacción con la misma. Se dará a conocer la solución y planteamiento de esta solución basados en los conceptos obtenidos dentro de la clase, arquitectura y desarrollo del producto como tal.

# Modelado de Negocio CANVAS

El modelado del negocio mediante el canvas es el cual describe un modelo de negocio o más bien la lógica de cómo una organización crea, entrega, y captura valor.   
Para el proyecto de aula se hizo el modelo de negocio mediante el canvas de la universidad enfocado hacia la biblioteca.



**Imagen 1: Modelado de negocio CANVAS**

## Descripción CANVAS

**Customer Segments:** El objetivo del politécnico grancolombiano es aportar a la transformación de la educación superior, brindando mayor acceso a la formación universitaria de alta calidad, disminuyendo la brecha digital que puede existir entre las diferentes personas que usan los diferentes servicios de la biblioteca.

**Value Propositions:** Los productos de la biblioteca se van a promocionar, alertas relacionadas con la biblioteca en el mundo de la información, enlaces de interés, juegos, alianzas biblioteca.

**Channels:** Los estudiantes pueden aprender de manera virtual o presencial para esto la universidad cuenta con diferentes canales de comunicación constantes y portales web para consulta de libros video o revistas.

**Customer Relationships:** La universidad atiende a todos los estudiantes. El asesoramiento de personas es una actividad a largo plazo basada en las relaciones con comunicación continua y personalizada.

**Revenue Streams:** El ofrecer un servicio dinámico contribuirá al crecimiento académico de los estudiantes permitiendo ganar más usuarios que buscaran este tipo de soluciones.

**Key Resources:** La biblioteca no solo ofrece préstamos de libros físicos, también virtuales, por esta razón tiene alianzas con sitios de búsqueda para que los estudiantes tengan más material de donde pueden obtener más información sobre los proyectos que están o pueden estar desarrollando.

También, se busca un bienestar de la comunidad, por esta razón en sus instalaciones tiene cubículos para el préstamo de películas o documentales. Por otro lado, brinda un espacio amigable para que los estudiantes puedan estudiar en un ambiente tranquilo e idóneo para esta actividad.

**Key Activities:** Las actividades principales para tener un desempeño óptimo de lo procesos o servicios que se prestan dentro de la biblioteca es la de difundir estos servicios, mediante los canales que permitan un rápido desarrollo de las actividades y más aún, quedarse con un gran aprendizaje. Dentro de esta etapa es fundamental saber qué es lo más importante a realizar, para que los recursos de la biblioteca puedan llegar a ser conocidos por cada uno de los miembros de la institución y se animen a participar activamente de cada uno de los talleres que se ofrezcan.

**Key Partners:** Mediante los aliados estratégicos podemos desarrollar, intercambiar y compartir recursos que van ligados en función de los estudiantes, estos aliados y sus diferentes recursos permiten la interacción de los estudiantes para enriquecer y abarcar mecanismos de conocimientos muchos más amplios.

**Cost Structure:** Se cuenta con el costo de las instalaciones, computadores, libros revistas, suscripciones a revistas, televisores, materiales para capacitaciones como video beams, tablets para que los estudiantes puedan consultar una gran variedad de libros, películas todo esto que hace que el estudiante pueda desarrollar mejor sus habilidades también está el costo del personal, como los bibliotecarios, el personal del aseo, los desarrolladores de los portales web de las bibliotecas virtuales y profesores.

## Análisis del entorno empresarial y contexto empresarial

La empresa se desenvuelve en un ámbito académico, en el cual quiere que sus estudiantes tengan un nivel competitivo con estudiantes de otras universidades del país y del mundo. Por esta razón busca ofrecer materiales y sitios en donde los estudiantes puedan obtener información que pueda ser usada en el desarrollo de la carrera y a lo largo de la vida. El objetivo de la institución universitaria es la prestación de servicios en todos los campos de acción de la educación superior.

# Objetivos

* Concientizar a toda la comunidad de la universidad sobre los servicios que se ofrecen en la biblioteca y de los grandes beneficios que se adquieren utilizando estos servicios.
* El producto irá dirigido a los estudiantes de la comunidad politécnico grancolombiano.
* Diseñar e implementar un juego para el politécnico grancolombiano para que los estudiantes conozcan los servicios de la biblioteca
* Diseñar e implementar un juego que sirva para promocionar e invite a los estudiantes a utilizar los servicios de la biblioteca.

# Perspectiva del producto

Se realizará un juego para dar a conocer los servicios de la biblioteca de acuerdo a las necesidades expuestas por el personal de la biblioteca. Con este juego se busca poder darle a conocer a toda la comunidad del politécnico grancolombiano de que la biblioteca no es solo un sitio en el cual solo se va a consultar libros sino mostrar los otros servicios que hacen de la biblioteca un lugar interesante para visitar y pasar un rato agradable.

BiblioGame será un video juego dirigido explícitamente a la comunidad de la universidad Politécnico grancolombiano para dar a conocer los diferentes servicios que ofrece la biblioteca para todos sus estudiantes como al personal administrativo. La intención del juego es mostrar de manera didáctica los diferentes servicios que ofrece la biblioteca y como se pueden hacer uso de ellos mediante diferentes situaciones que llevan al jugador a realizar ciertos desafíos los cuales permiten dar a conocer cómo opera cierto servicio en la biblioteca.

# Documento de levantamiento de requerimientos

La especificación de requisitos de software (ERS) es una descripción completa del comportamiento del sistema que se va a desarrollar. Incluye un conjunto de casos de uso que describe todas las interacciones que tendrán los usuarios con el software y se especifica cada una de las funcionalidades que tendrá el software y sus respectivos responsables. El documento de especificación de requerimientos del proyecto se encuentra adjunto a la entrega del mismo.

# Análisis de riesgos

## Identificación de riesgos

**Abandono del proyecto:**

* Algún compañero del grupo abandone el proyecto antes de finalizarlo.
* Algún compañero del grupo tenga un inconveniente familiar.
* Algún compañero del grupo se enferme.
* Desinterés de algún compañero del grupo por el proyecto.

**Cambio de especificaciones:**

* Los requerimientos de la aplicación cambien durante el desarrollo del proyecto.
* Después de realizar la documentación y luego de desarrollar el juego se cambie las especificaciones.

**Demoras en los tiempos:**

* Que algunos de los requerimientos tomen más tiempo de la planeado.
* Que no se cumplan los tiempos en la especificación o en el desarrollo como se especificó al principio y no se cumpla con el objetivo final.

**Herramientas:**

* Todos los miembros del equipo alcancen las habilidades necesarias para el desarrollo del proyecto en la herramienta seleccionada para completarlo.
* Cambio de la herramienta en la cual se va a desarrollar porque no cumplió con todas las especificaciones del proyecto.
* Daño de alguna de las máquinas con las cuales se estén trabajando.

**Aceptación:**

* Al personal de la biblioteca no le agrade el producto final.

## Evaluación de los riegos

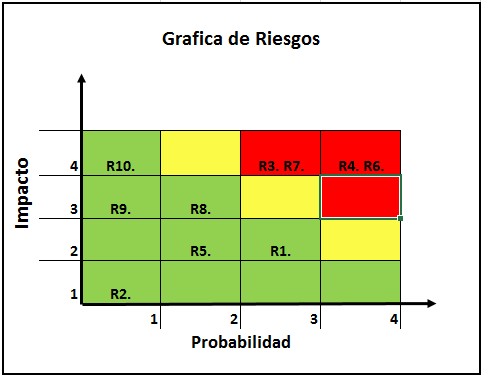
**Convenciones de Impacto**

Insignificante = 1 Baja = 2 Mediana = 3 Alta = 4

|  |  |  |  |
| --- | --- | --- | --- |
| **RIESGO** | **PROBABILIDAD** | **IMPACTO** | **PROB. x IMPACT.** |
| **R1.** Algún compañero del grupo abandone el proyecto | 2 | 3 | 6 |
| **R2.** Algún compañero tenga intermitencia en la asistencia | 1 | 1 | 1 |
| **R3.** Los requerimientos cambien durante el desarrollo del proyecto | 3 | 4 | 12 |
| **R4.** Después de realizar documentación y desarrollo se cambien las especificaciones | 4 | 4 | 16 |
| **R5.** Algún requerimiento tome más tiempo del estimado | 2 | 2 | 4 |
| **R6.** Desarrollo de la aplicaciones sobre una herramienta desconocida | 4 | 4 | 16 |
| **R7.** Que la herramienta no cumpla con los requisitos mínimos para desarrollar el proyecto | 3 | 4 | 12 |
| **R8.** Daño de alguna de las máquinas con la cual se está trabajando | 2 | 3 | 6 |
| **R9.** Hurto de alguna de las máquinas o herramientas con las que se trabaja | 1 | 3 | 3 |
| **R10.** Al personal de la biblioteca no le agrade el producto final. | 1 | 4 | 4 |

**Tabla 1: Tabla de evaluación de riesgos**

## Grafica de evaluación de riesgos

****

**Imagen 2: Grafica Resultado evaluación de riesgos**

## Planes y acciones a tomar

**R3. Los requerimientos cambien durante el desarrollo del proyecto:**

**Impacto:** Tendría que cambiarse lo que ya se desarrolló y cambiar el cronograma para ajustar tiempos podría no alcanzar el tiempo.

**Mitigación:** Con una comunicación más fluida y constante con el cliente en el levantamiento de los requerimientos y la especificación clara del juego, podríamos disminuir el riesgo de cambios en lo solicitado.

**Contingencia:** Desde el inicio dejar un espacio de tiempo para realizar el cambio en la documentación si es necesario y más tiempo en desarrollo para realizar los cambios necesarios los controles de cambio si llegaran a presentarse.

**Reducción:** Mejorar la fluidez con el cliente para mejorar los requerimientos y ajustar el tiempo para un posible control de cambio.

Exposición: Realizando estas prevenciones podríamos mejorar el tiempo en el desarrollo y una aplicación más apetecible por el cliente.

**R4. Después de realizar documentación y desarrollo se cambien las especificaciones**

**Impacto:** Construir un producto o sistema excelente que no quiere nadie en realidad (riesgo de mercado).

**Mitigación:** cuidadosa evaluación del plan del proyecto, del entorno técnico en el que se desarrolla el proyecto y conseguir fuentes de información viables.

**Contingencia:** los riesgos se pueden disminuir haciendo una cuidadosa evaluación entre las especificaciones y las historias de usuario disminuyendo la falta de comunicación con el cliente.

**Reducción:** Disminuir la falta de comunicación con el cliente y hacer evaluaciones periódicas de las especificaciones del proyecto con el cliente.

**Exposición:** Entablar un diálogo más fuerte con el cliente para dejar claro las especificaciones del producto

**R6. Desarrollo de la aplicación sobre una herramienta desconocida:**

**Impacto:** Si se presentaran problemas de compatibilidad o compilación existiría menos documentación o soporte por parte de la herramienta utilizada lo que generaría buscar nuevas soluciones que se verían representadas en tiempo e inversión.  
**Mitigación:** Al tener los requerimientos de se debe realizar un análisis de equipo sobre los aplicativos o programas más utilizados para este tipo de soluciones.

Contingencia: De acuerdo a la evaluación de riesgos sobre la herramienta escogida se debe tomar por lo menos dos alternativas que sean las más acertadas con el desarrollo del proyecto por su capacidad y funcionalidad para las tareas requeridas.

**Reducción:** Es importante lograr realizar el mayor número de pruebas y casos de uso al momento de escoger la herramienta, esto permitirá contemplar una opción dos en caso que la herramienta principal no cumpla con lo requerido. Es muy importante tener muy específicos los casos de uso y requerimientos.

**Exposición:** Se debe contemplar en todo momento las matrices de riesgo con un constante monitoreo por parte del director de proyecto con el fin de asegurar los cambios realizados sobre la documentación y los resultados de controles efectuados.

**R7.  Que la herramienta no cumpla con los requisitos mínimos para desarrollar el proyecto:**

**Impacto:** Al ya haber comenzado el proyecto y llevar un buen tiempo desarrollando la aplicación y ver que la herramienta utilizada no posee los suficientes componentes para seguir según la planeación del proyecto forzará  a que hayan retrasos en las entregas estipuladas.   
**Mitigación:** Se debe hacer una investigación exhaustiva antes de empezar la parte de desarrollo sobre la herramienta que se está a punto de utilizar, comparar contra las historias de usuario y dar certeza de que cada una de ellas se puedan de una u otra forma lograr desarrollarlas con la herramienta.

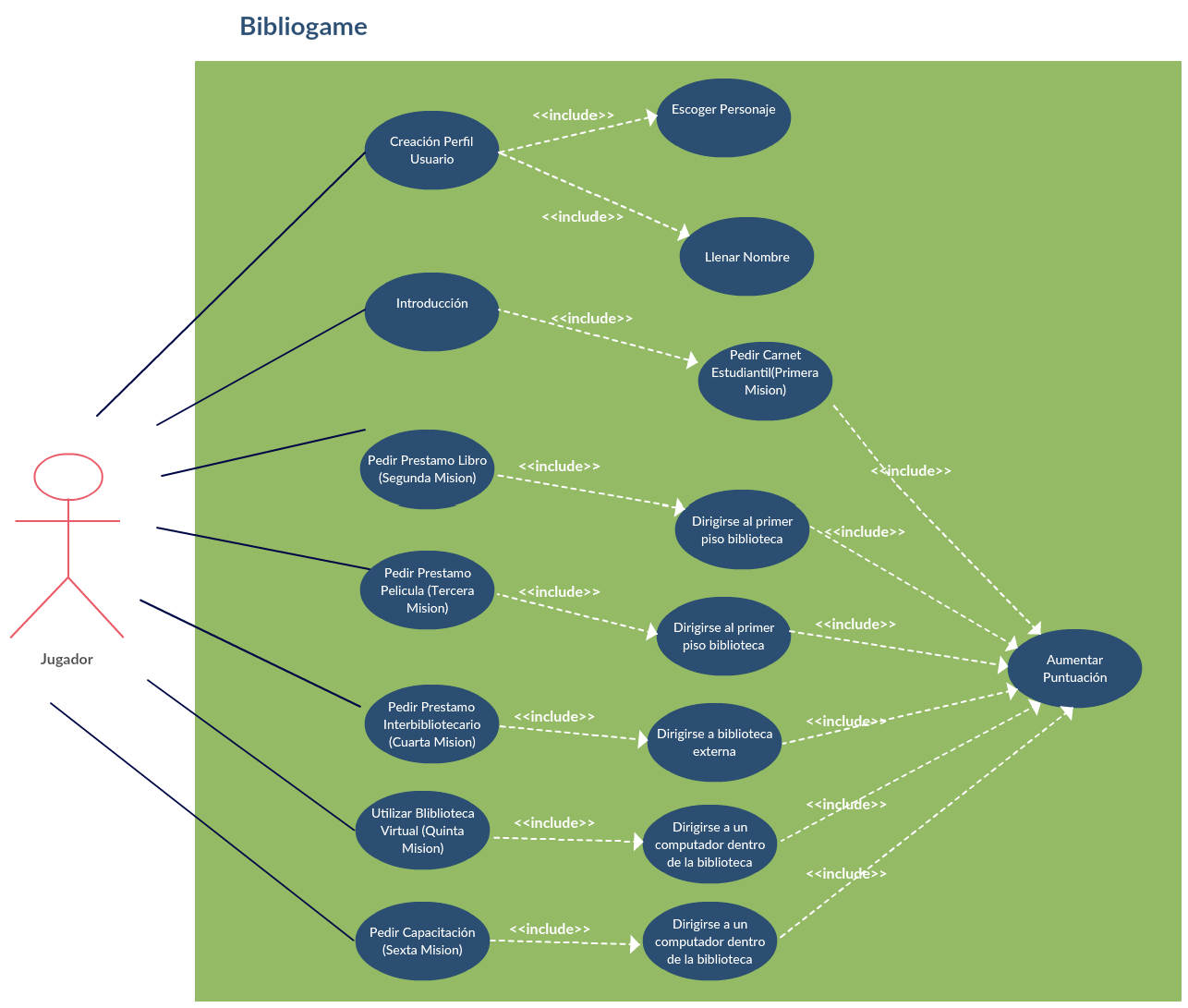
**Contingencia:** Se debe tener como alternativa una segunda herramienta que cumpla con los objetivos que requiere el proyecto.

**Reducción:** Revisar detalladamente las historias de usuario y casos de uso, y cada uno de los cambios que estas requieran y comparar constantemente con la investigación previamente hecha acerca de la herramienta a utilizar

**Exposición:** Hacer constante monitoreo a la herramienta utilizada y pruebas a la aplicación en sus diferentes fases.

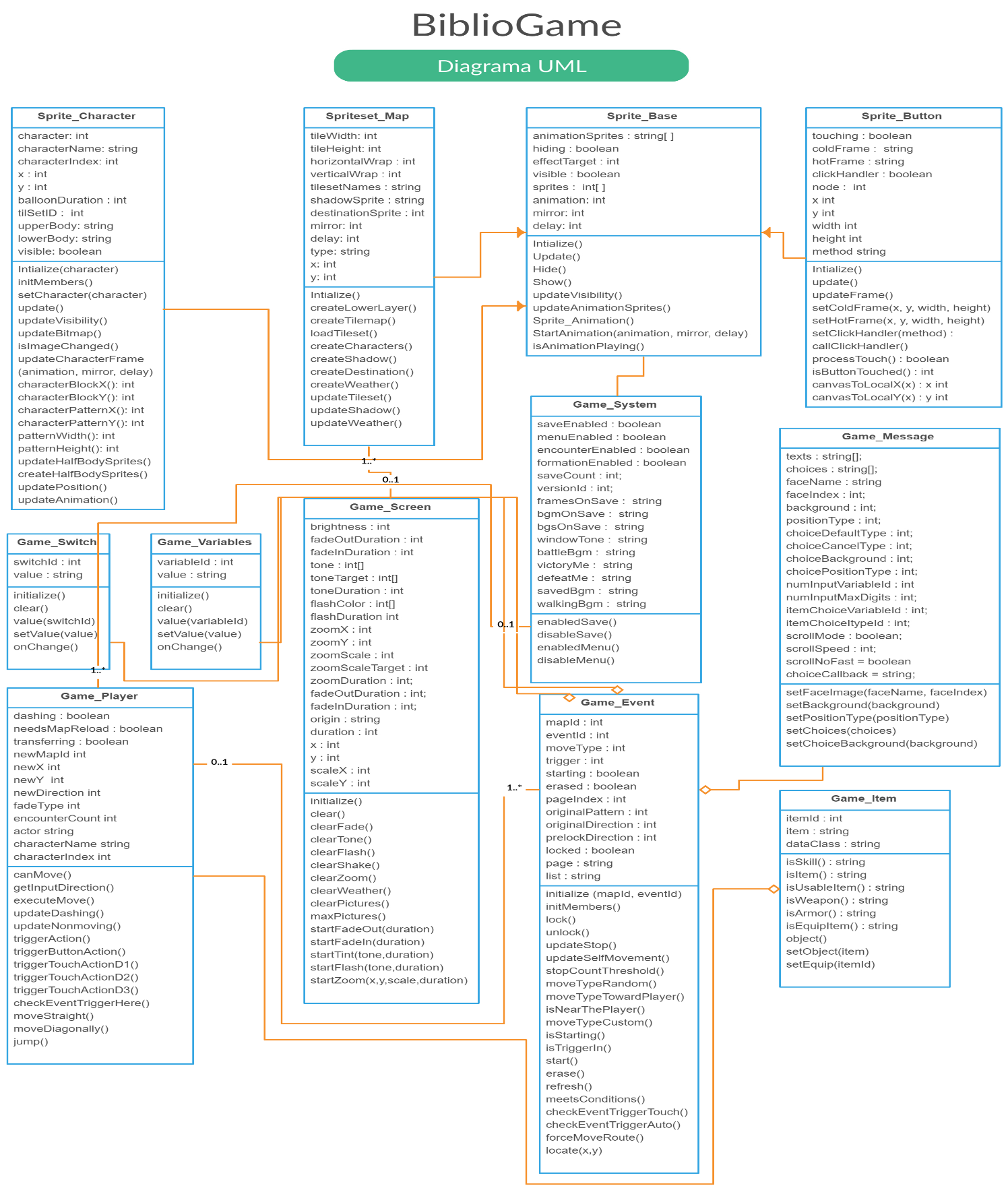
# Diagrama de casos de uso

En el Lenguaje de Modelado Unificado, un diagrama de casos de uso es una forma de diagrama de comportamiento UML mejorado. El Lenguaje de Modelado Unificado (UML), define una notación gráfica para representar casos de uso llamada modelo de casos de uso.

  
**Imagen 3: Diagrama de casos de uso**

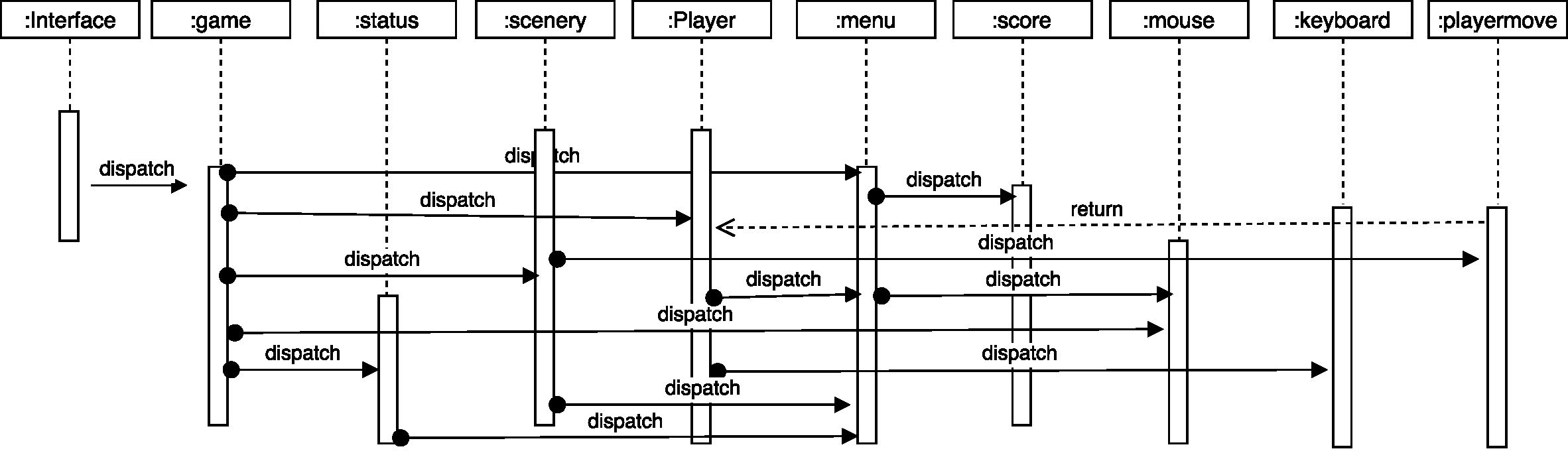
# Diagrama de clases

Un diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia, de uso y de contenimiento.

  
**Imagen 4: Diagrama de clases**

# Diagrama de secuencia

El diagrama de secuencia es un tipo de diagrama usado para modelar interacción entre objetos en un sistema, para determinar cada una de los objetos que interactúan y cuál sería el flujo en tiempo de ejecución según UML.

  
**Imagen 5: Diagrama de secuencia**

# User Story Map y Release Planning

Es una técnica descrita por primera vez por Jeff Patton y consiste en representar el product backlog en dos dimensiones en vez de en una. Al construir el User story map al principio del proyecto, sirve para tomar decisiones y provocar muchas conversaciones que posteriormente serán útiles durante la construcción. Durante el resto del proyecto es muy útil para no perder el horizonte sobre lo que se pretende construir. El User story map se encuentra adjunto a la entrega del mismo con sus respectivas especificaciones.

# Patrones de diseño

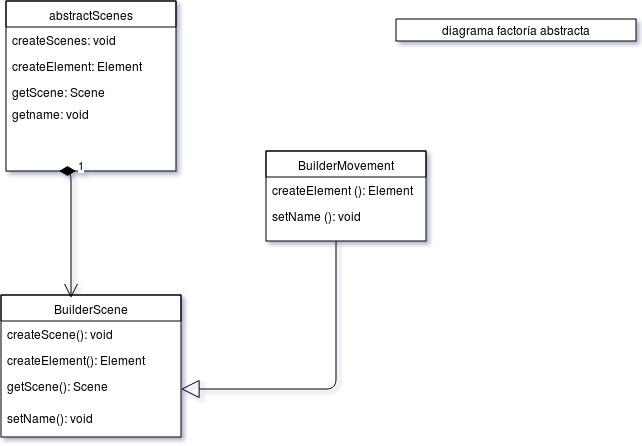
Los patrones de diseño y la inteligencia artificial se mezclaron para poder utilizar una estructura reutilizable. En esta estructura adiciona nuevos objetos sin afectar al modelo como una forma de inteligencia artificial. Esta relación permite crear un modelo altamente escalable y cambiante.

Los patrones utilizados se describen los patrones utilizados, ventajas y desventajas que podrían presentar en el juego, y como están implementados de acuerdo a sus funcionalidades:

## Abstract factory

Este patrón permite poder crear familias de objetos con algunas características en común. Cada familia define esas características de forma autónoma, precisa de forma precisa. Además se  puede añadir familias de objetos.

El cliente tiene que hacer una “Abstracción”, puesto que debe crear familias que pueden ser concretas, ósea existir, se trabaja con una clase abstracta y no con una familia concreta (que existe).

  
**Imagen 6: Diagrama de clases patrón abstract factory**

**Escenas del proyecto:** Las componentes de este patrón son los siguientes:

* Abstract factory: Define una interfaz para la creación de objetos abstractos. Se representan por los siguientes elementos:
* AbstractScenes: Es una interfaz que representa la creación de escenas abstractas del juego. Además de lo anterior define los métodos en los que debe implementar las fábricas de las escenas concretas.
* BuilderScene: Esta es una clase abstracta que implementa la interfaz.
* AbsFactScenes que define el método “createScene()” ya que se quiere que la responsabilidad  la debe tener una escena abstracta y no las escenas concretas.
* Factoría Concreta: es la que Implementa las operaciones que crean objetos concretos. Las factorías concretas quedan definidas por las siguientes clases:

BuilderMovement: Define los métodos de “AbstractScenes”,  para crear las escenas de estrategia.

**Producto Abstracto:** Declara una interfaz con objetos de un tipo de producto. En el proyecto, que corresponde con las siguientes interfaces: Element: Establece los métodos necesarios para definir los distintos tipos de Element.

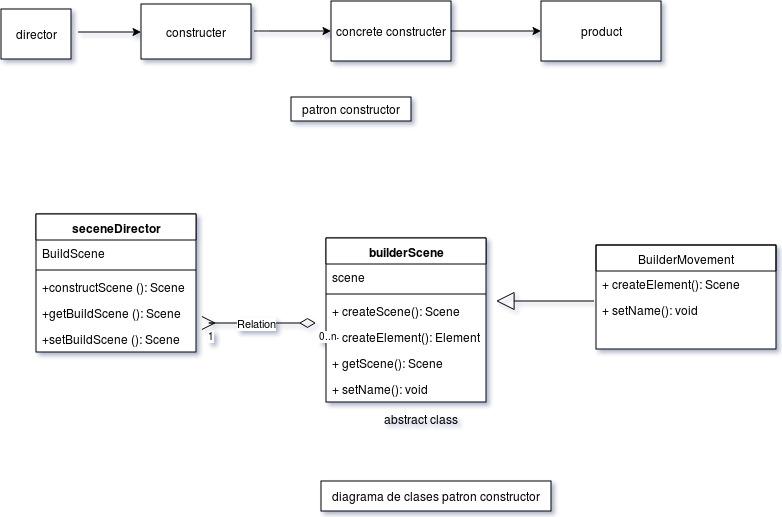
**Producto Concreto:** Implementa la interfaz del producto abstracto correspondiente. Especifica un producto concreto que la debida fábrica.

Se establecen las siguientes:

* clases: Element: Producto concreto representativo de las escenas de estrategia que corresponden al juego de estrategia.
* Cliente: Utiliza la fábrica para crear distintas familias. En el proyecto queda representado por la siguiente clase: Scene: Cliente que utiliza la clase abstracta “BuilderScene” para crear las distintas escenas. Se utiliza esta clase abstracta, en vez de la interfaz “abstractScenes”

Constructor

Este patrón crea un objeto “completo” de otros más “simples”. Realiza el proceso de creación en un punto, con el objetivo de que dicho objeto se cree de varias instancias, correspondiendo a esas pequeñas partes (objetos simples) que hacen ya parte de él.

  
**Imagen 7: Diagrama de clases patrón constructor**

**El patrón constructor posibilita:**

* adicionar partes simples que contengan un objeto completo.
* Aumentar las funciones de un objeto.
* cambiar procesos de creación de una forma mucho más simple.

**Componentes:**

* componente Abstracto: Es capaz de declarar una interfaz de objetos de un tipo.
* componente Concreto: obtiene el objeto complejo que se construye.
* Constructor: Declara todas las operaciones, que permiten la creación de partes del Producto y es capaz de implementar el constructor acoplando el objeto completo.
* Director: ensambla  un objeto con referencia a la interfaz Constructor.

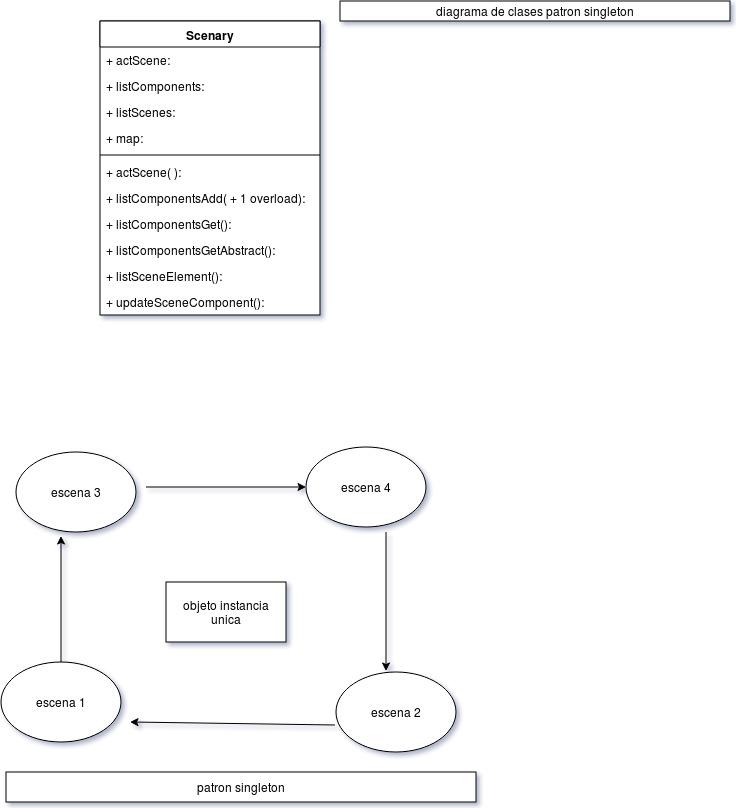
## Singleton

Solamente crea objetos que tienen una clase. Es por ello que sus clases solamente tienen instancia a un solo objeto.

Este patrón se requiere porque se necesita guardar el estado del juego como otros componentes importantes.

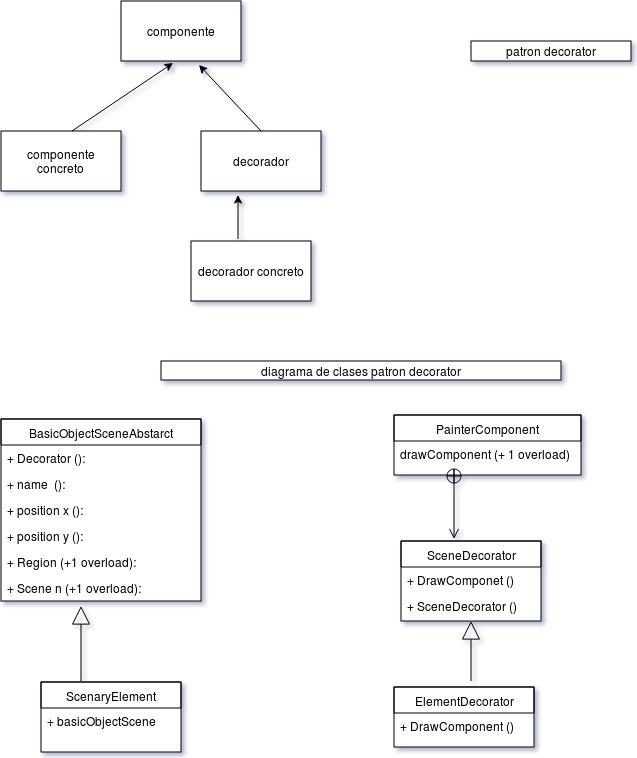
**Este patrón posibilita:**

* Única creación de un objeto completo (de varios elementos).
* Instancia accesible desde los demás objetos de la aplicación.

  
**Imagen 8: Diagrama de clases patrón singleton**

## Decorador

Permite añadir responsabilidades a objetos concretos de forma dinámica. Se utiliza para la visualización de todos objetos que componen los distintos escenarios. Posibilita que la creación cambie de acuerdo al tipo de escena que se pueda dibujar.

  
**Imagen 9: Diagrama de clases patrón decorador**

**Posibilita:** Adicionar responsabilidades dinámicas y continuas y Crea un tipo de herencia funcional.

**Las componentes del patrón son los siguientes:** Componente: Define la interfaz que adiciona responsabilidades dinámicas a objetos que las pueden recibir. Se representa por la siguiente clase: BasicObjectSceneAbstract: interfaz que representa la creación de objetos abstractos del juego define los métodos que deben implementar los componentes concretos.

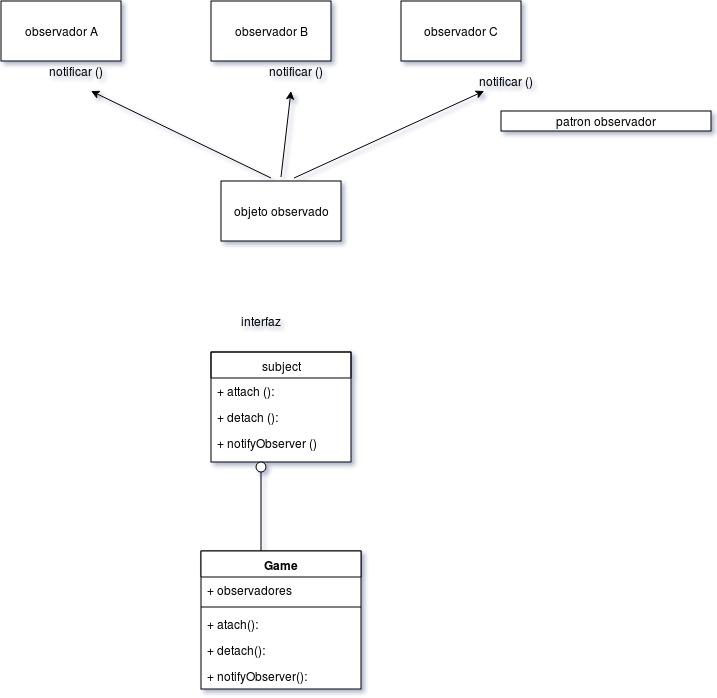
**ScenaryElement:** Define el objeto al que se le puede agregar una responsabilidad. En el proyecto, se corresponde con las siguientes componentes:

* PainterComponent: Es la interfaz que contiene los decoradores concretos. Define métodos que pueden implementar los decoradores de las escenas concretas.
* SceneDecorator: Es una clase abstracta que implementa la interfaz PainterComponent la cual  tiene la referencia al Objeto Componente.
* Decorador Concreto: Añade la responsabilidad al Componente.

Si se añade un nuevo decorador, el desarrollador debería crear una clase que extienda de la clase abstracta “SceneDecorator”. La nueva clase que se crea extiende la responsabilidad de componentes que existían antes de crear los nuevos.

## Observador

Este patrón de diseño hace que ciertas clases u observadores, puedan reaccionar ante eventos determinados. Define una dependencia del tipo uno-a-muchos entre objetos. Cuando uno de los objetos observados cambia su estado, notifica este cambio a todos los objetos dependientes de esta información.

  
**Imagen 10: Diagrama de clases patrón observador**

Permite un bajo acoplamiento entre clases. Un objeto puede tener muchos observadores, los cuales reciben notificaciones al cambiar el primer objeto. Un objeto no sabe que existen muchos observadores, solo sabe que puede ser observado.

**Las componentes del patrón son los siguientes:**

**Sujeto:** Ofrece la posibilidad de añadir y quitar observadores. Está representado por la siguiente clase: Subject: Tienen los métodos como attach () y  detach () que sirven para adicionar o quitar observadores.

**Sujeto Concreto:** se encarga de notificar a observadores si su estado cambió. Corresponde a las siguientes clases. Game: extiende de la interfaz Sujeto, realiza la notificación cuando se producen cambios de su estado a los observadores.

**Observador:** Define la interfaz con la cual se notifican los observadores de los cambios producidos por el Sujeto.

# Herramientas y frameworks:

## Librería

Es un módulo de software orientado a suministrar una funcionalidad concreta, como puede ser descodificar vídeo H.264, importar modelos en 3D de cierto programa.

## Framework

Hace referencia a una arquitectura del software en su totalidad con varios puntos de extensibilidad para poder incluir funcionalidades personalizadas. Es una estructura base que se encarga de administrar la carga, descarga, y elementos del juego que permiten importar algunos recursos del juego como son modelos, texturas y sonidos Está compuesto por varios elementos más pequeños que se pertenecen a una serie de funcionalidades (en módulos y/o bibliotecas). Se integran de un subsistema a un sistema en el juego.

El framework entonces  es el motor de juego gráfico. Un motor gráfico se encarga de toda la parte relacionada con los gráficos: gestión del bucle de render, de los recursos gráficos, de reservar contextos gráficos según el hardware sobre el que se ejecuta. Es así como el framework en RPGmaker interconecta subsistemas al motor de juego que es el que se encarga de añadir la lógica de cómo gestionar los posibles niveles del juego, las reglas del juego, etc.

  
**Imagen 11: Programa utilizado para creación juego**

# Pruebas y especificaciones

## Planeación de pruebas

Es la etapa en donde se ejecutan las primeras actividades correspondientes al proceso de pruebas y tiene como resultado un entregable denominado plan de pruebas el cual debe estar conformado en cuando menos por aspectos tales como Alcance de la prueba, Tipos de Prueba, Estrategia de Pruebas, Criterios de Salida. El plan de pruebas del proyecto bibliogame se encuentra adjunto a la entrega del mismo con sus respectivas especificaciones.

## Pruebas de carga

El objetivo de este documento es recoger las pruebas que verifican la capacidad máxima soportada por el aplicativo que se encuentra alojado en un servidor de pruebas. El detallado de la pruebas de carga se especifican en el adjunto a la entrega del mismo.

## Especificación del servidor

Según el resultado de las pruebas de carga el servidor necesario para cubrir la demanda de 40.000 usuarios se describe a continuación:

* Tipo de servidor: HP Proliant DL160 G3
* Sistema Operativo: Windows Server 2012 R2 con IIS 8.5
* Procesador: Intel Xeon CPU E5-2609 1.70 GHZ Dual Core
* Memoria RAM: 16 GB
* Capacidad Disco Duro: 1 TB
* Arquitectura: 64 GB
* Fuentes de Poder: 2 AC 110V
* Puertos Ethernet: 2 Puertos Gigabit Ethernet
* Phisical NIC: 1 de 4 puertos Gigabit Ethernet
* Uplinks: 2

## Especificación de Ancho de Banda

Para brindar un buen ancho de banda a nivel de LAN los dos puertos Gigabit Ethernet se deben de configurar bajo el protocolo LACP para manejar un throughput a nivel de enlace de 2 GB.

**Topología de conexión**

  
**Imagen 12: Topología de conexión de ancho de banda**

# Conclusiones

# Referencias Bibliográficas

AMERICAN PSYCHOLOGICAL ASSOCIATION (1991), Publication Manual of The American

Psychological Association: (3ª. Edición. 10ª. Impresión). Washintong DC. Autor.