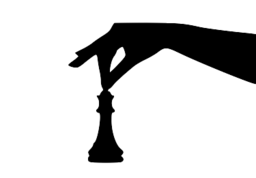
**CHESS KING**



**Juego de Rol y Estrategia para Pc’s**

Pc, Google Chrome, Mozilla Firefox, Opera

**Local:** 1 Jugador

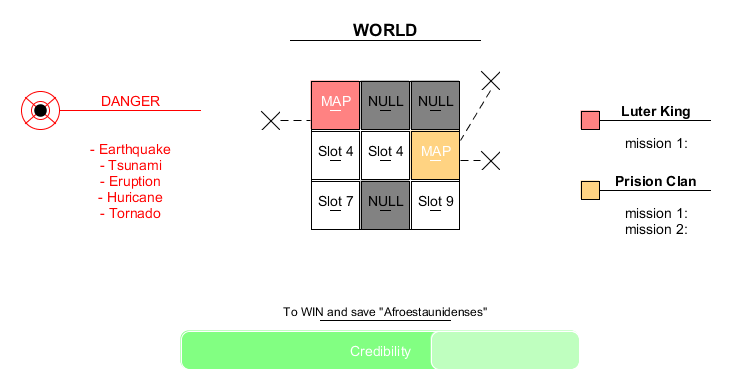
**En-línea:** Multijugador

**Descripción:**

En 1955 Martin Luther King se posiciona frente al movimiento por los derechos civiles de los afroestaunidenses y desarrolla una guerra contra la segregación racial en el sur de Estados Unidos, el juego se gana al lograr evitar el magnicidio del siglo XX (la muerte de Martin Luther King) en Memphis, Tennessee el 4 de abril de 1968.

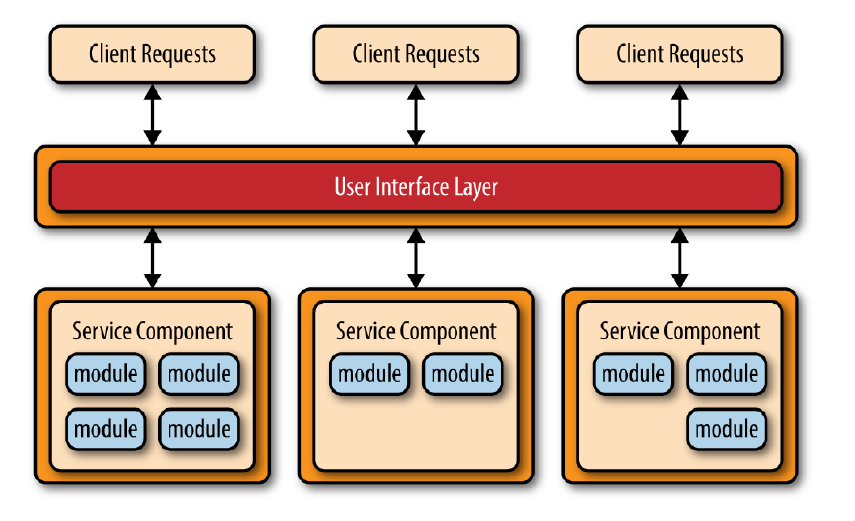
**Reglamento:**

|  |
| --- |
| 1. Los Personajes podrán elegir entre varios estilos de juego: (Chess, Checkers, Sudoku, Domino, Damas, Mahjong, backgammon, Tic tac Toe,..) 2. Los personajes se limitaran a las características y el comportamiento según el estilo que elijan (únicamente para las batallas). 3. Los escenarios de batalla serán tableros adaptados al estilo. 4. Solo hay un mundo de tamaño personalizado: B (big) – M (med) – S (small). 5. Las partidas tienen límite de tiempo (parametrizable). 6. Las partidas tienen un máximo de clanes (parametrizable). 7. Puede aliarse con otros clanes por un motivo en común. 8. Debe tomar decisiones, respondiendo a preguntas específicas a lo largo de la exploración para ganar aliados. 9. La exploración permite el desarrollo de habilidades encontrando aliados NPC’s. 10. Explorando encontraras tips de ajedrez para preparar tu estrategia. 11. Los personajes de cada bando tomaran las características y el comportamiento de las piezas del ajedrez (únicamente las posiciones) para darle versatilidad de estrategias y lograr cautivar la atención del usuario a lo largo del desarrollo de sus planes de ataque y partidas de guerra entre clanes. 12. El juego se desarrolla en base a la exploración para recaudar recursos y fortalecer el clan para ganar las batallas de estrategia de mesa contra los enemigos. 13. La batalla se desarrolla como una partida de ajedrez limitando las posiciones de ataque en función de las posiciones que permite dicho reglamento, sin embargo el resultado puede ser variado respecto del poder de ataque y vida residual de los personajes en conflicto. 14. Los personajes del clan pueden obtener armas y pociones para aumentar su poder de ataque y su defensa vital. 15. Los desastres naturales afectan el estado de la partida. 16. Los grupos se dividen en jugador, clan, club, comunidad. 17. El jugador puede evolucionar su poder de ataque aprendiendo nuevas técnicas para potencializar su estrategia y más poder consiguiendo armas o herramientas. |



**Arquitectura General (Patrones)**

1. Cliente / Servidor
2. Arquitectura de Micro servicios

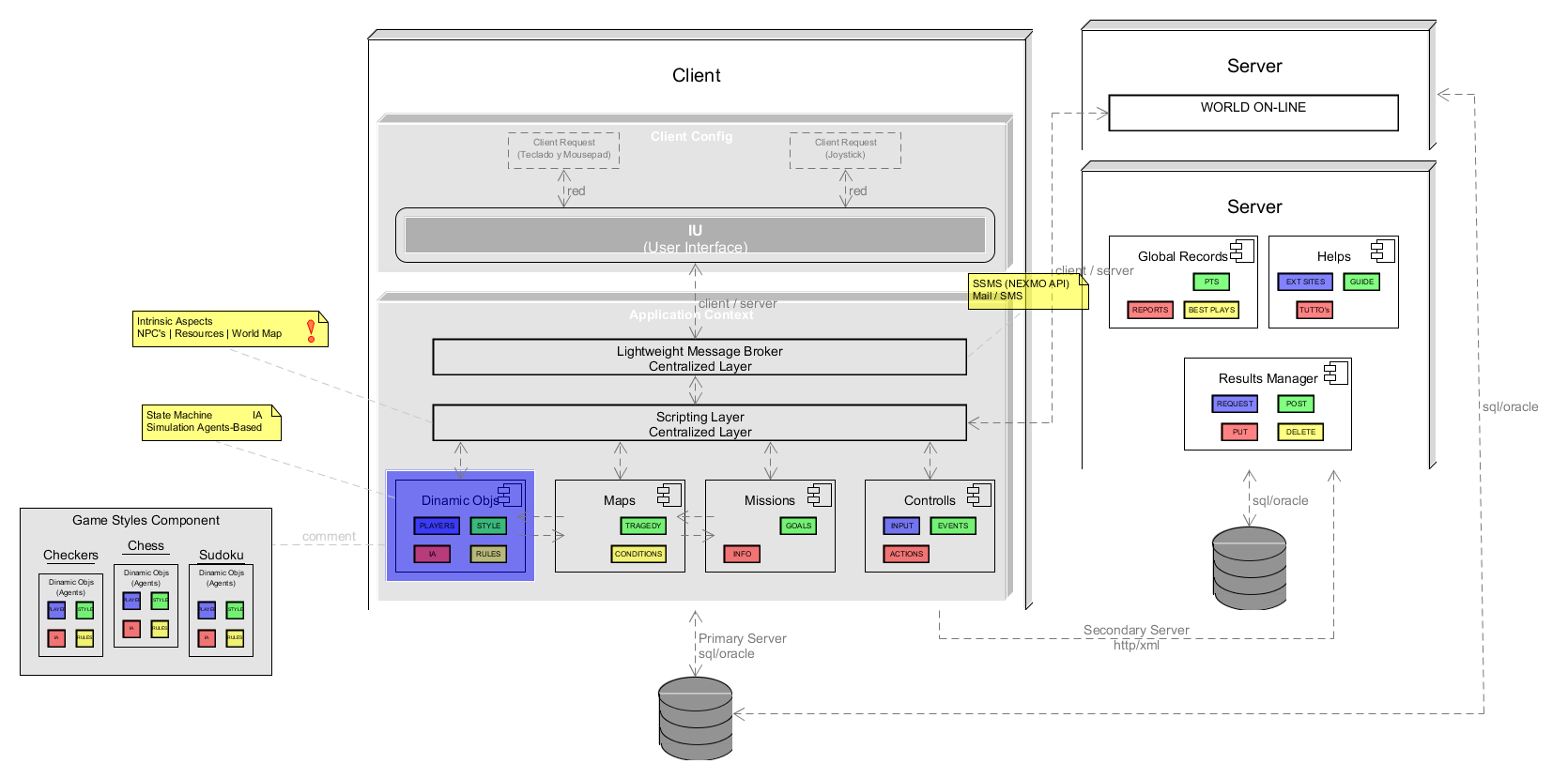


Sistema de alto nivel

Sistema de Scripting (Configuración del comportamiento base)

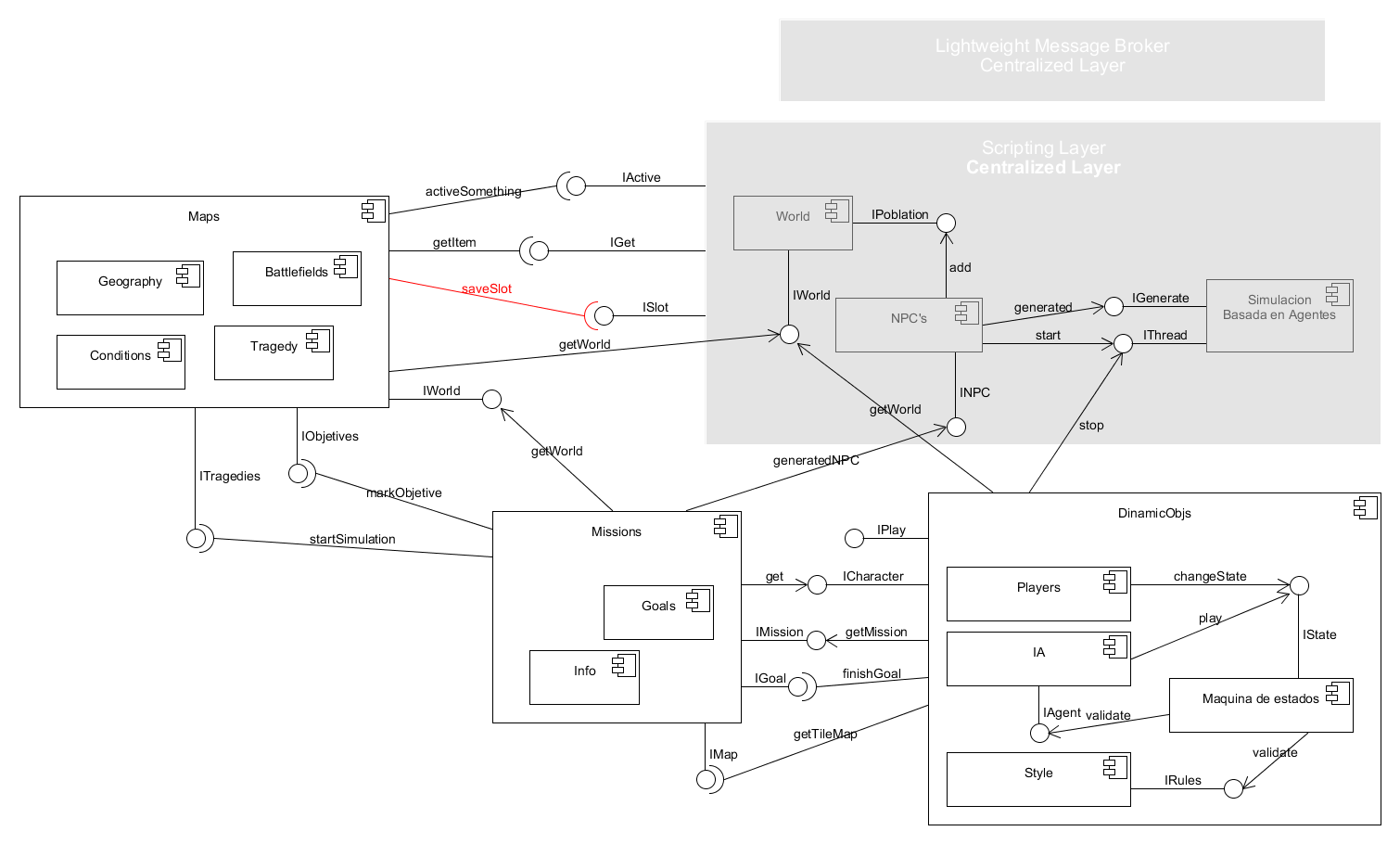
El sistema de scripting permite modelar fácilmente la lógica del juego, por ejemplo el comportamiento “estático” de los NPC’s en la exploración y el mundo principal, sin necesidad de volver a compilar, modularizando jerarjicamente aspectos o detalles del juego en cuestión, en algunos casos los motores del juego permiten que siga en funcionamiento al mismo tiempo que se carga un nuevo script, esto permite centralizar funciones estáticas del juego.

La funcionalidad IA de los NPC y los enemigos en las batallas estratégicas entre clanes se agregara en componentes externos según la versión del juego, permitiendo la escalabilidad y flexibilidad del juego.

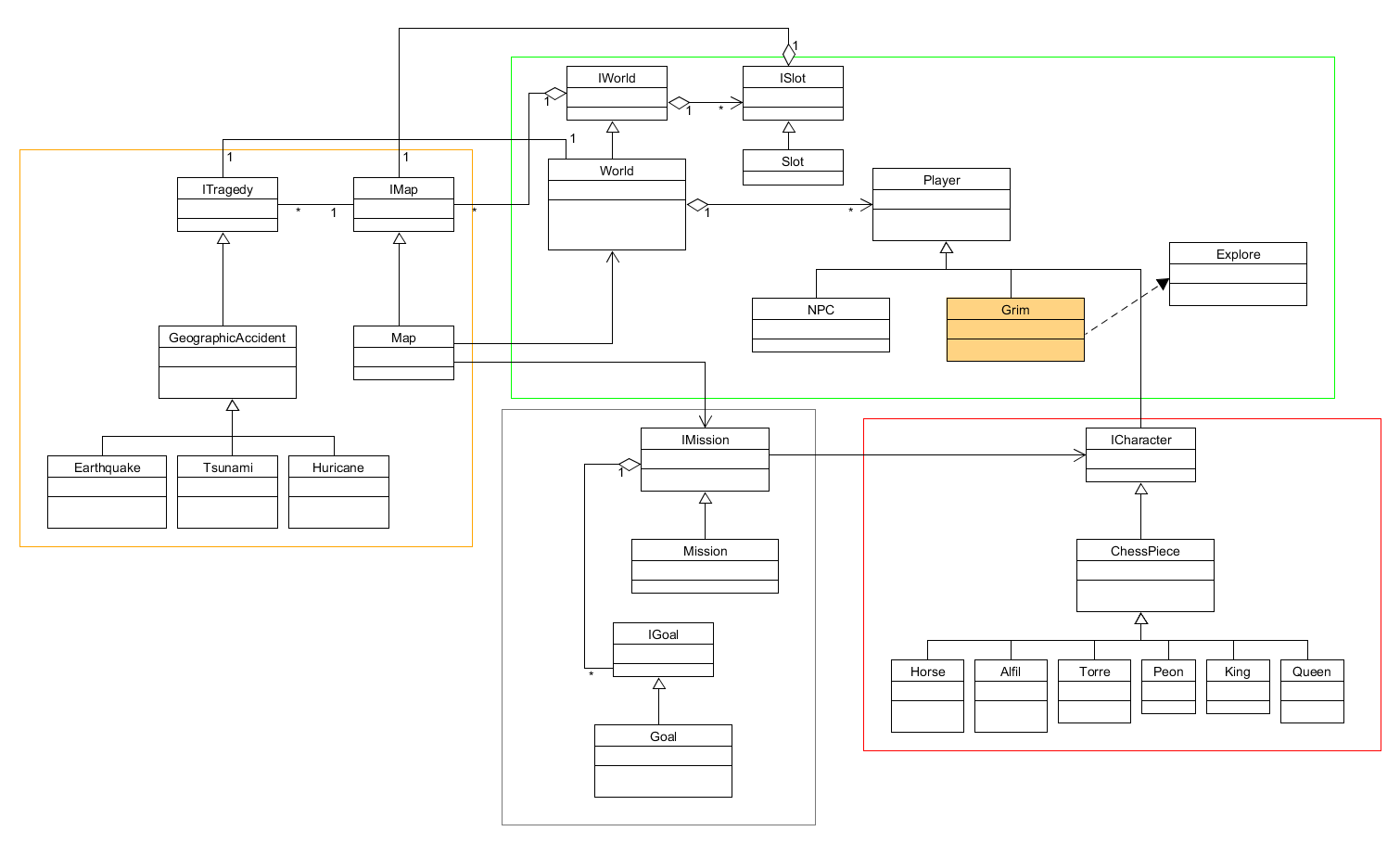


**Diagrama de Componentes**

En este diagrama se especifica un poco más la interacción entre los diferentes subsistemas del software.



**Diagrama de Clases**



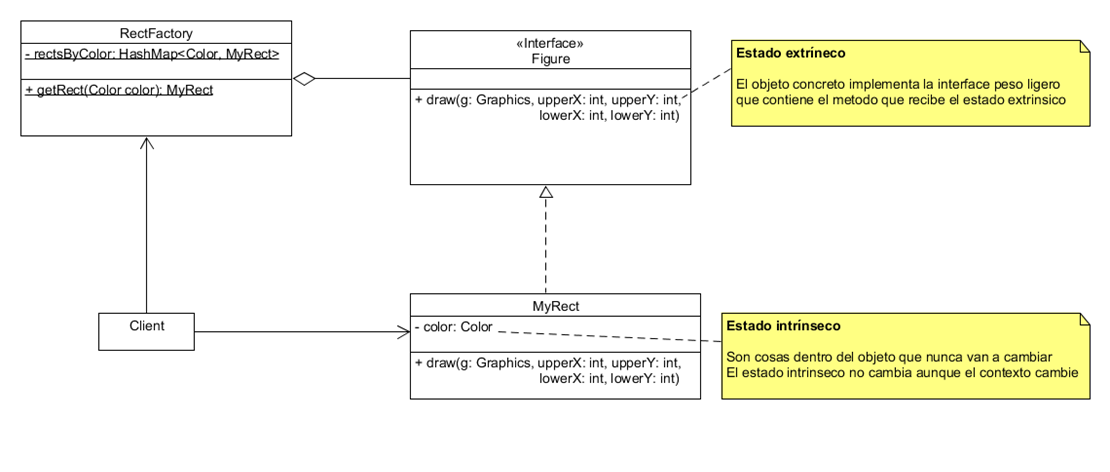
**Diseño e implementación (Patrones)**

1. FlyWeight (Escenario, Objetos).
2. Singleton (Jugador Explorador).
3. Factory (Items).
4. State (Características de los personajes, Simulación basada en Agentes)
5. Adapter (Controles, Mods)

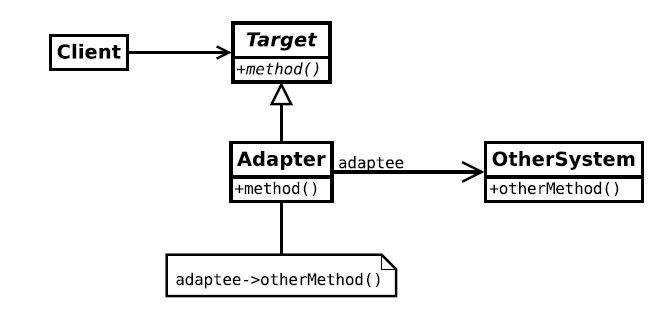
**Patrón FlyWeight**

El patrón flyweigth viene a dar solución al problema de crear grandes cantidades de objetos, para almacenar estos objetos sin un gran coste de memoria, se apoya en objetos que almacenan los estados compartidos (intrínseco) que pueden ser usados por varios objetos simultáneamente, esto reduce en gran cantidad el peso de los datos en un servidor. Su propósito es eliminar la redundancia que se produce al tener una gran cantidad de objetos que contienen información idéntica, además de lograr un equilibrio entre flexibilidad y rendimiento, (se busca tratar de ser más eficiente utilizando el mínimo de recursos necesarios para disminuir el uso de memoria RAM).

A la hora de realizar búsquedas consume un poco más de tiempo, sin embargo, el ahorro de almacenamiento depende de cuantas instancias se puedan reducir, esto quiere decir que no aplicar el patrón, aumentaría los costos de almacenamiento y al aplicarlo aumentaría el costo en función del tiempo de ejecución.



**Patron Adapter**

Usando el patrón Adapter es posible crear una nueva interfaz de acceso a un determinado objeto, por lo que proporciona un mecanismo de adaptación entre las demandas del objeto cliente y el objeto servidor que proporciona la funcionalidad.

**Referencias**

agentes.wordpress.com. (n.d.). agentes-inteligentes-y-videojuegos.

David Vallejo Fernandez, C. M. (n.d.). Desarrollo de videojuegos - 2da Edicion : Arquitectura del Motor de Videojuegos.

Erich Gamma, R. H. (n.d.). Patrones de Diseño: Elementos de Software Orientado a Objetos Reutilizable.

Media, O. (n.d.). Software Architecture Patterns.