



THE KERNEL KREW
DISTRIBUI ET CONQUERE

REQUERIMIENTOS DE SOFTWARE

INTEGRANTES:
EMILIO ALDEAN
JUAN DIEGO MATHEUS
JAIRÓ PÉREZ



Diseño de Arquitectura de Software: Juego sobre principios de la informática

Integrantes: Juan Diego Mateus, Emilio Aldean y Jairo Perez



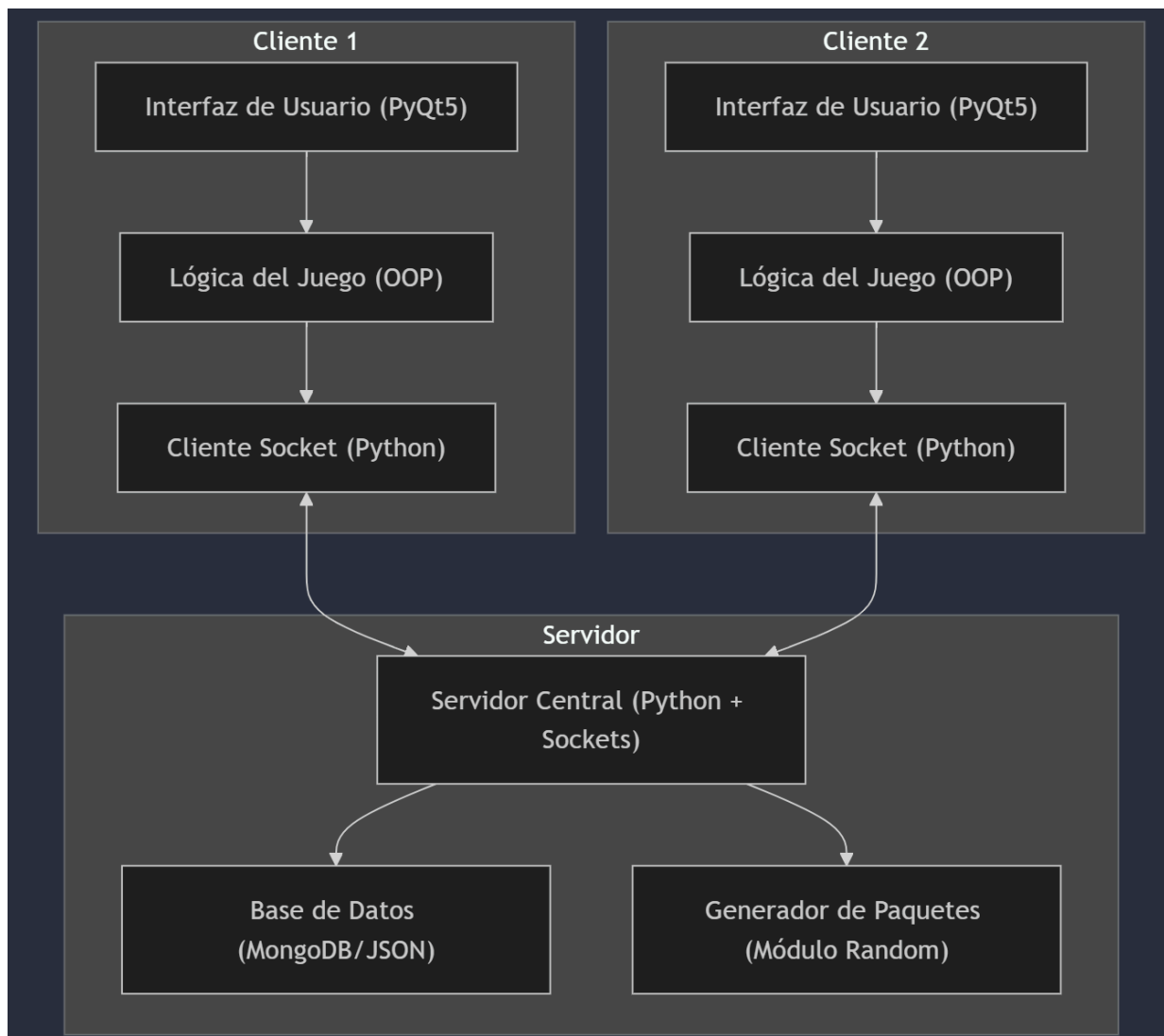
Historial de Versionamiento

Versión	Fecha	Participante	Tipo de Cambio / Descripción
1.0	2024-11-07	Juan Diego Mateus	Creación inicial del documento con una descripción general del proyecto y objetivos.
1.1	2024-11-14	Emilio Aldean	Agregado de la sección de arquitectura macro y micro del sistema, incluyendo detalles sobre clientes, servidores y bases de datos.
1.2	2024-11-14	Jairo Pérez	Inclusión de especificaciones de los minijuegos con detalles sobre funcionalidad y tecnologías utilizadas.
1.3	2024-11-15	Juan Diego Mateus	Incorporación de la sección sobre roles de usuario y matriz de permisos para definir accesos en el sistema.
1.4	2024-11-18	Emilio Aldean	Añadidos los modelos de entidad-relación para la base de datos MongoDB, especificando las relaciones entre jugadores, partidas y puntuaciones.
1.5	2024-11-19	Jairo Pérez	Revisión general para corregir errores de formato y ortografía, y consolidación de todas las secciones.
2.0	2024-11-20	Todos	Versión final validada, incluyendo actualizaciones menores en las tecnologías utilizadas y presentación final del documento.

Índice

1. Infraestructura.....	4
1.1 Clientes.....	5
1.2 Servidor.....	5
1.3 Conexión Cliente-Servidor.....	5
2. Arquitectura Macro.....	5
2.1 Jugadores (Clientes).....	6
2.2 Servidor Central.....	6
2.3 Base de Datos.....	6
2.4 Conexión a Internet.....	6
3. Arquitectura Micro.....	7
3.1 Clientes (Cliente 1 y Cliente 2).....	8
3.2 Servidor.....	8
3. Bases de Datos.....	9
4. Modelo Entidad Relación (Bases de Datos) MongoDB.....	9
4.1 Player.....	10
4.2 Game.....	10
4.3 Score.....	10
5. Roles de Usuario y Matriz de Permisos.....	10
5.1 Roles de Usuarios.....	10
5.2 Matriz de Permisos.....	11

Infraestructura



Clientes:

- Cada cliente tiene una interfaz gráfica desarrollada en PyQt5 que interactúa con la lógica del juego.
- La lógica del juego se conecta al servidor a través de sockets.

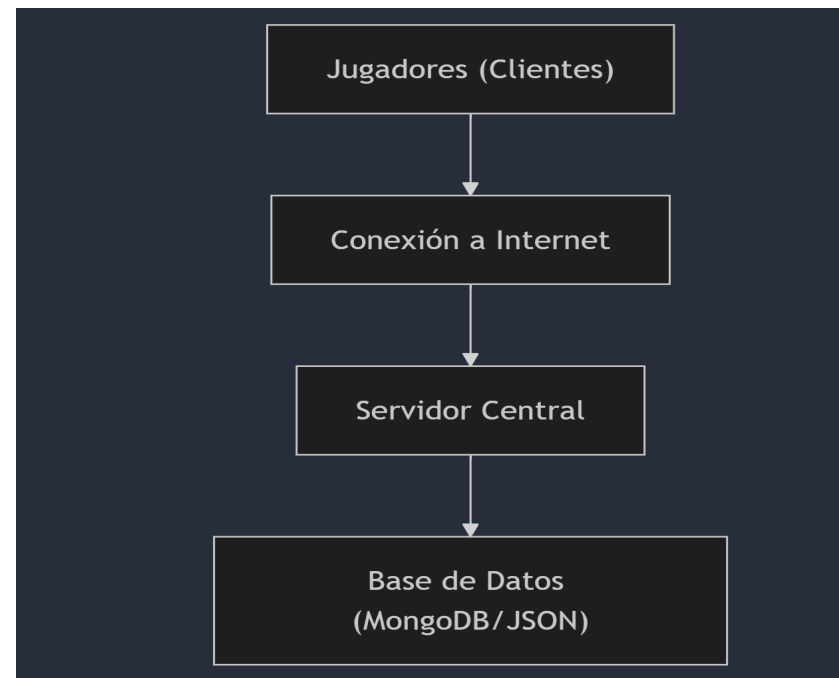
Servidor:

- El servidor central maneja toda la lógica del juego compartida, como sincronización de paquetes, puntuaciones y condiciones de fin de juego.
- Utiliza MongoDB o JSON para almacenar puntuaciones y tablas de clasificación.
- Genera paquetes de red sincronizados para ambos jugadores usando el módulo Random.

Conexión Cliente-Servidor:

- Los clientes se conectan al servidor mediante sockets, asegurando actualizaciones en tiempo real entre los jugadores.

Arquitectura Macro





Jugadores (Clientes):

- Representan los dispositivos de los jugadores que interactúan con el juego.
- Se conectan a través de la interfaz gráfica (UI) y el cliente socket al servidor.

Servidor Central:

- Coordina la lógica del juego y sincroniza las acciones entre los jugadores.
- Maneja la lógica principal del juego multijugador y la sincronización de datos.

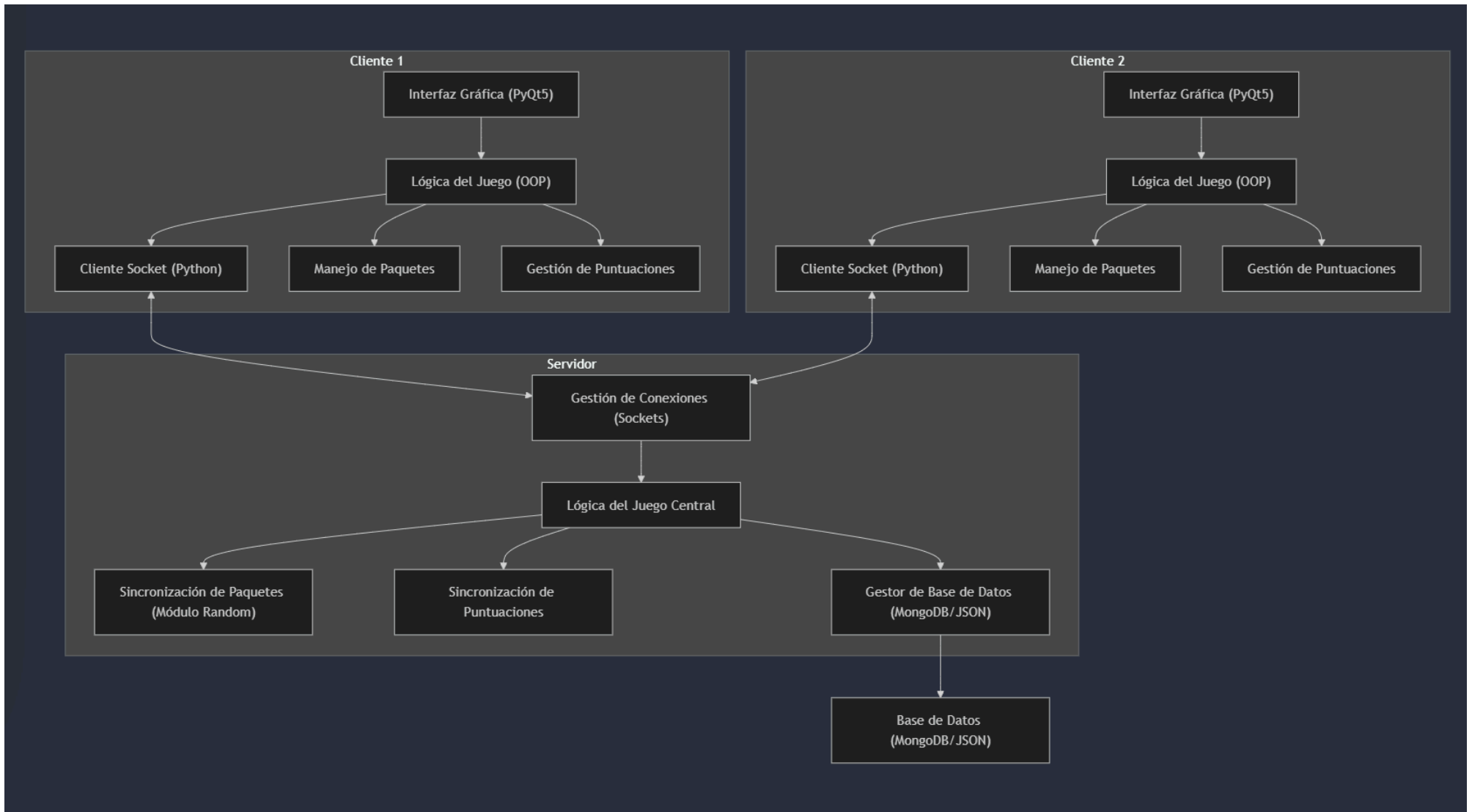
Base de Datos:

- El servidor central se conecta a una base de datos para almacenar puntuaciones y gestionar las tablas de clasificación.

Conexión a Internet:

- Es el medio que conecta a los jugadores con el servidor central.

Arquitectura Micro



Clientes (Cliente 1 y Cliente 2):

1. Interfaz Gráfica (UI):

- Permite a los jugadores interactuar con el juego.

2. Lógica del Juego (OOP):

- Maneja eventos de juego, actualizaciones en la interfaz y procesamiento de datos.

3. Cliente Socket:

- Gestiona la comunicación con el servidor.

4. Manejo de Paquetes:

- Analiza y clasifica los paquetes recibidos.

5. Gestión de Puntuaciones:

- Administra la puntuación local del jugador y la sincroniza con el servidor.

Servidor:

1. Gestión de Conexiones:

- Controla las conexiones de los jugadores mediante sockets.

2. Lógica del Juego Central:

- Coordina las acciones de los jugadores y resuelve conflictos.

3. Sincronización de Paquetes:

- Genera y distribuye paquetes sincronizados entre los jugadores.

4. Sincronización de Puntuaciones:

- Asegura que las puntuaciones estén actualizadas en tiempo real.

5. Gestor de Base de Datos:

The Kernel Krew - Emilio Aldean, Juan Diego Matheus y Jairo Pérez

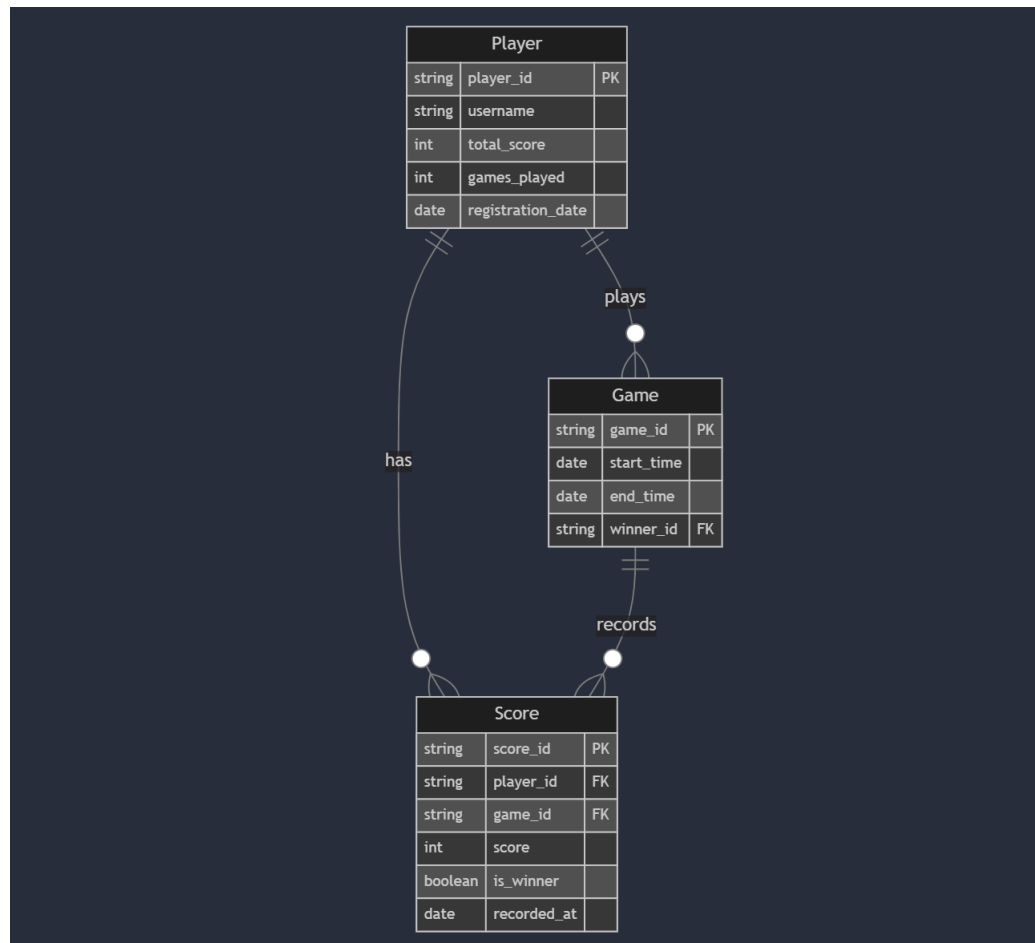
- Guarda las puntuaciones y estadísticas en MongoDB o JSON.

Base de Datos:

- **MongoDB/JSON:**

- Almacena la tabla de clasificación y los historiales de puntuación

Modelo Entidad Relación (Base de Datos) MongoDB



1. Player:

- Tabla o colección que almacena información básica sobre los jugadores.
- Relación de uno a muchos con **Game** y **Score**.

2. Game:

- Almacena información sobre cada partida, incluyendo el identificador del ganador.
- Relación de uno a muchos con **Score**.

3. Score:

- Registra las puntuaciones de cada jugador en una partida específica.
- Contiene relaciones con **Player** y **Game**.

Controles de Acceso y Roles de Usuario

Roles de Usuarios

1. Administrador:

- Acceso total al sistema.
- Permisos para gestionar usuarios (crear, editar, eliminar).
- Configurar y personalizar los minijuegos (niveles de dificultad, parámetros de juego, etc.).
- Visualizar y exportar estadísticas globales de uso y rendimiento.

2. Usuario Estándar (Estudiante):

- Acceso a los minijuegos.
- Participación en sesiones individuales o multijugador.
- Visualización de sus propias estadísticas y progreso.
- Personalización limitada del perfil de usuario.

Matriz de Permisos

Acción	Administrador	Usuario Estándar
Crear usuarios	✓	✗
Editar/eliminar usuarios	✓	✗
Configurar parámetros de juego	✓	✗
Acceder a todos los minijuegos	✓	✓
Participar en multijugador	✓	✓
Visualizar estadísticas globales	✓	✗
Visualizar estadísticas personales	✓	✓
Personalizar perfil	✓	✓
Acceso a modo de prueba	✓	✓