



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico II: The Curry Game release v7.1.2

Ingeniería de Software II
Primer Cuatrimestre de 2016

Integrante	LU	Correo electrónico
Leandro Matayoshi	79/11	leandro.matayoshi@gmail.com
Matías Pizzagalli	257/12	matipizza@gmail.com
Gastón Requeni	400/11	grequeni@hotmail.com
Martín Santos	413/11	martin.n.santos@gmail.com



1. Primer Entrega

No realizamos ninguna corrección sobre la primer entrega, por lo que no la volvemos a presentar en esta entrega.

2. Atributos de calidad

En esta sección presentamos los escenarios de calidad que obtuvimos tras analizar los resultados del QAW.

2.1. Atributos de disponibillidad

Atributo de disponibilidad

'El sistema debe estar andando todo el tiempo'

- **Fuente:** Externa
- **Estímulo:** Solicita acceso al sistema
- **Entorno:** Normal
- **Artefacto:** Sistema
- **Respuesta:** El sistema responde normalmente
- **Medición de respuesta:** Disponibilidad del 99,99 % (se puede caer aprox 1h en todo el año)

Atributo de disponibilidad

'Si falla un enlace regional, se redirige el tráfico a regiones cercanas de manera uniforme.'

- **Fuente:** Servidor regional
- **Estímulo:** No responde
- **Entorno:** Normal
- **Artefacto:** Sistema
- **Respuesta:** El sistema detecta la falla en el servidor regional y redirige el tráfico a las regiones más cercanas. La región entra en modo degradado. Se loguea la falla y se envía una notificación al técnico en redes
- **Medición de respuesta:** El servidor / enlace son reparados en menos de 24hs.

Atributo de disponibilidad

'Se produce una falla en una base de datos de un servidor'

- **Fuente:** Interna
- **Estímulo:** Falla en la base de datos de un servidor
- **Entorno:** Normal
- **Artefacto:** Subsistema de almacenamiento y manejo de datos persistentes
- **Respuesta:** El subsistema detecta y loguea la falla. El servidor original continúa operando normalmente a través del uso de votación. La base de datos cambia a silencioso en el caso de ser la primera falla, y es reemplazada por una nueva instancia en caso de ser la segunda. Se envía una notificación al Data Base Manager.
- **Medición de respuesta:** Se garantiza disponibilidad a pesar de una falla en la base de datos el 99.99 % de las veces

Atributo de disponibilidad

'Se pierden datos de una base de datos de servidor'

- **Fuente:** Interna
- **Estímulo:** Pérdida de datos en una base de datos de servidor
- **Entorno:** Normal
- **Artefacto:** Subsistema de almacenamiento y manejo de datos persistentes
- **Respuesta:** El subsistema detecta y loguea la falla. El servidor recupera los datos ya que cuenta con una base redundante. La base de datos cambia a silencioso en el caso de ser la primera falla, y es reemplazada por una nueva instancia en caso de ser la segunda. Se envía una notificación al Data Base Manager.
- **Medición de respuesta:** Se mantienen los datos a pesar de una pérdida en una de las bases de datos el 99.99 % de las veces

Atributo de disponibilidad

'Se pierden los datos de todas las bases de datos de servidor'

- **Fuente:** Interna o externa
- **Estímulo:** Pérdida de datos de todas las bases de datos de servidor
- **Entorno:** Normal
- **Artefacto:** Subsistema de almacenamiento y manejo de datos persistentes
- **Respuesta:** El subsistema detecta y loguea la falla. El servidor vuelve al último estado consistente de las últimas 4 horas, ya que realiza un backup cada esa cantidad de tiempo. Se restauran todas las bases. Se envía una notificación al Data Base Manager.
- **Medición de respuesta:** La probabilidad del escenario anterior es $< 0.00001\%$. Las bases se restauran en un tiempo < 4 horas

Atributo de disponibilidad

'En cada región habrá varios servidores con una capacidad máxima de usuarios que puede atender, debido a limitaciones de hardware / conexión. Si nuevos usuarios se agregan y superan el 90 % de la capacidad, habrá que agregar un nuevo servidor y balancear la carga'

- **Fuente:** Externa
- **Estímulo:** Solicita acceso al sistema
- **Entorno:** A 1 pedido de alcanzar el límite de usuarios
- **Artefacto:** Servidor regional
- **Respuesta:** El servidor responde normalmente. Se incorpora un nuevo servidor a la red, y se aplica el balanceo de carga correspondiente
- **Medición de respuesta:** Se realiza la subdivisión de la región agregando un nuevo nodo en menos de 6 horas

Atributo de disponibilidad

'Enlaces congestionados durante streaming de partido real'

- **Fuente:** Externa
- **Estímulo:** Disminución de la capacidad de enlace durante streaming de partido real

- **Entorno:** Normal
- **Artefacto:** Servidor regional
- **Respuesta:** Se detecta el cambio de bitrate. Se realiza un downgrade de la calidad de video. El usuario continúa observando el partido de forma fluida, pero con menor calidad.
- **Medición de respuesta:** El streaming del video mantiene un rate constante de cuadros por segundo el 99.99 % de los casos

Atributo de disponibilidad

'Enlaces congestionados durante streaming de video de simulación'

- **Fuente:** Externa
- **Estímulo:** Disminución de la capacidad de enlace durante streaming de simulación
- **Entorno:** Normal
- **Artefacto:** Servidor regional
- **Respuesta:** Se detecta el cambio de ancho de banda del enlace. Se realiza un downgrade en el bitrate
- **Medición de respuesta:** El streaming de la simulación mantiene un rate constante de cuadros por segundo el 99.99 % de los casos

Atributo de disponibilidad

'Se caen enlaces de región durante transmisión de torneo continental o mundial'

- **Fuente:** Externa
- **Estímulo:** Caída de enlaces de región durante transmisión de torneo continental o mundial
- **Entorno:** Normal
- **Artefacto:** Servidor regional
- **Respuesta:** Se detecta la caída de los enlaces en la región. Se utiliza la topología de la conexión de regiones para triangular los paquetes y que lleguen a los usuarios.
- **Medición de respuesta:** El 99.99 % de las veces el usuario continúa viendo la transmisión del evento sin cortes abruptos, experimentando a lo sumo un cambio en la calidad del video

2.2. Atributos de performance

Atributo de performance

'Quiere que todo lo respectivo al manejo de dinero (depósitos y retiros de los participantes via tarjeta de crédito o caja de ahorro) sea super seguro (no quiere papelones y que los datos de las millones de tarjetas de los participantes aparezcan publicados en Reddit), transparente y rápido. Que los datos queden resguardados y sólo haya que actualizarlos esporádicamente.'

- **Fuente:** usuario
- **Estímulo:** depósito / retiro de dinero
- **Entorno:** operación normal
- **Artefacto:** subsistema de pagos
- **Respuesta:** el sistema realiza la operación satisfactoriamente

- **Medición de respuesta:** el sistema realiza la operación en menos de 15 segundos

Atributo de performance

'Propone un sistema de bitrate variable automático/manual de los streams de video para que se pueda bajar la calidad de los videos en base al bandwidth detectado disponible del usuario.'

- **Fuente:** usuario
- **Estímulo:** observa transmisión de partido
- **Entorno:** sistema degradado
- **Artefacto:** sistema
- **Respuesta:** se modifica calidad del video
- **Medición de respuesta:** se modifica la calidad del video en menos de 10 segundos

Atributo de performance

'Quiere que mientras sea posible se use el engine 3d de mayor calidad al 2d.'

- **Fuente:** usuario
- **Estímulo:** observa simulación de partido
- **Entorno:** sistema degradado
- **Artefacto:** dispositivo móvil antiguo
- **Respuesta:** se reemplaza engine 3d por 2d
- **Medición de respuesta:** antes de comenzar la reproducción de la simulación se reemplaza el engine 3d por 2d

2.3. Atributos de seguridad

Atributo de seguridad

'Atacante intenta robar datos de tarjetas de crédito o cuentas corrientes bancarias, pero el sistema lo impide'

- **Fuente:** Atacante
- **Estímulo:** Intenta robar datos de tarjetas de crédito o cuentas corrientes almacenados en servidores
- **Entorno:** Normal
- **Artefacto:** Datos del sistema
- **Respuesta:** Se detecta y se impide el ataque.
- **Medición de respuesta:** Se detecta y se impide el 99 % de los ataques.

Atributo de seguridad

'Atacante roba datos de tarjetas de crédito o cuentas corrientes bancarias, pero no puede descifrarlos'

- **Fuente:** Atacante
- **Estímulo:** Roba datos de tarjetas de crédito o cuentas corrientes almacenados en servidores
- **Entorno:** Normal
- **Artefacto:** Datos del sistema

- **Respuesta:** Se guardan los datos en un formato imposible de leer.
- **Medición de respuesta:** Toma más de 1000 años descifrar los datos.

Atributo de seguridad

'Atacante intercepta comunicación del sistema con el usuario.'

- **Fuente:** Atacante
- **Estímulo:** Interviene pasivamente una comunicación entre el usuario y el sistema
- **Entorno:** Normal
- **Artefacto:** Comunicación del sistema
- **Respuesta:** La comunicación está protegida por SSL, con lo cual el contenido de los paquetes es imposible de leer
- **Medición de respuesta:** Toma más de 1000 años descifrar los datos interceptados

Atributo de seguridad

'Atacante se hace pasar por el sistema para robarle datos al usuario.'

- **Fuente:** Externa
- **Estímulo:** Interviene activamente una comunicación entre el usuario y el sistema, tomando el rol del sistema
- **Entorno:** Normal
- **Artefacto:** Comunicación del sistema
- **Respuesta:** El sistema utiliza un mecanismo de autenticación del servidor mediante certificados y clave asimétrica. El browser alerta al usuario de que se han vulnerado los certificados SSL. Los paquetes obtenidos por el atacante están encriptados, por lo cual su contenido no puede determinarse
- **Medición de respuesta:** Los usuarios advertidos acerca del posible riesgo comienzan una nueva sesión segura el 99.99 % de las veces. En el caso de que el usuario envíe datos sin darse cuenta, toma más de 1000 años descifrar los datos interceptados

Atributo de seguridad

'Atacante modifica mensajes enviados entre el sistema y el usuario para forzar al sistema a realizar acciones no solicitadas por el usuario.'

- **Fuente:** Externa
- **Estímulo:** Interviene activamente una comunicación entre el usuario y el sistema, modificando mensajes capturados en el canal de comunicación
- **Entorno:** Normal
- **Artefacto:** Comunicación del sistema
- **Respuesta:** El sistema utiliza un mecanismo de verificación de integridad de los mensajes recibidos, tanto del lado del cliente como del servidor. El mecanismo de integridad viaja encriptado para evitar que sea modificado.
- **Medición de respuesta:** Toma más de 1000 años encontrar un mensaje que estando modificado tenga sentido y verifique la integridad.

Atributo de seguridad

'Un usuario logueado logra vulnerar el subsistema de pagos y cobros.'

- **Fuente:** Usuario identificado
- **Estímulo:** Vulnera el subsistema de pagos y cobros y genera movimientos de dinero a su favor
- **Entorno:** Normal
- **Artefacto:** Subsistema de pagos y cobros
- **Respuesta:** El sistema tiene un audit trail con el registro de todas las acciones realizadas por todos los usuarios logueados y revierte las operaciones realizadas por el usuario.
- **Medición de respuesta:** El 99.99 % de las veces el log tiene todos los datos necesarios para revertir las operaciones del usuario.

Atributo de seguridad

'Usuario no autorizado desea hacer uso de los datos recolectados por minería'

- **Fuente:** Usuario sin privilegios de administrador
- **Estímulo:** Intento de acceso a datos recolectados por minería
- **Entorno:** Normal
- **Artefacto:** Sistema
- **Respuesta:** El sistema loguea el intento de acceso. El sistema valida los permisos del usuario en el sistema y posteriormente niega el acceso
- **Medición de respuesta:** Los usuarios no autorizados no logran acceder a los datos el 99.99999 % de los casos

Atributo de seguridad

'Auditor verifica que el código de la simulación y cálculo de resultados de desafíos no se haya modificado'

- **Fuente:** Auditor
- **Estímulo:** Solicitud de hashes de auditoría para módulos de simulación y cálculo de resultados de desafíos
- **Entorno:** Normal
- **Artefacto:** Sistema
- **Respuesta:** Se otorgan los hashes correspondientes a ambos módulos
- **Medición de respuesta:** La coincidencia de los hashes obtenidos con los conservados con el auditor garantizan que el código no ha cambiado el 99.9999 % de los casos (muy baja probabilidad de colisiones en la función de hash)

Atributo de seguridad

'Usuario culpa al sistema de que no se le ha asignado el premio de un desafío en el que ha participado y ganado, pero no figura su lista de desafíos'

- **Fuente:** Usuario
- **Estímulo:** Acusación de premio no otorgado

- **Entorno:** Normal
- **Artefacto:** Sistema
- **Respuesta:** Se le muestra en base al audit trail del sistema el listado de todas las inscripciones a desafíos que realizó, quedando en evidencia que el usuario no se ha inscripto en dicho desafío
- **Medición de respuesta:** El audit trail mantiene una relación 1 a 1 con las operaciones del usuario en un 99.99 % respecto a las acciones del usuario del sistema. Es decir, no hay acciones que no estén logueadas y en el log aparecen únicamente acciones realizadas por dicho usuario

Atributo de seguridad

'Resolvidor de desafío de liga de fantasía obtiene un dato erróneo de una jugada provisto por el sistema externo que brinda resultados en real-time'

- **Fuente:** Externa
- **Estímulo:** Dato erróneo del sistema proveedor
- **Entorno:** Normal
- **Artefacto:** Sistema
- **Respuesta:** La resolución de una jugada en el minuto a minuto de un desafío de liga de fantasía es obtenida a partir de una votación, por lo que el resultado correcto es calculado
- **Medición de respuesta:** El proceso de votación obtiene el resultado correcto el 99.99 % de las veces

2.4. Atributos de modificabilidad

Atributo de modificabilidad

'Quiere ver estadísticas acerca del comportamiento de los participantes de todas las temporadas (los más ganadores/perdedores en desafíos/dinero, los mejores/peores equipos formado por participantes de diferentes caps, valores de caps de participantes, rankings de regiones más ganadoras en desafíos/dinero, el modo de desafío más utilizado por los participantes, etc...). Lo que está en paréntesis son sólo ejemplos. Quiere que la mayor cantidad de datos que el sitio maneja pueda ser fácilmente minada por datos. Y que esos datos pueda estar a cargo de administradores expertos para luego crear desafíos acordes o otorgar créditos a participantes que califiquen.'

- **Fuente:** administrador
- **Estímulo:** agregar estadísticas acerca del comportamiento de los participantes
- **Entorno:** tiempo de ejecución
- **Artefacto:** sistema
- **Respuesta:** se agrega las nuevas estadísticas
- **Medición de respuesta:** se agregan las nuevas estadísticas en menos de una hora sin reiniciar el sistema

Atributo de portabilidad

'Debe de poder correr en la mayor cantidad de plataformas posibles, incluyendo móviles.'

- **Fuente:** desarrollador
- **Estímulo:** adaptar interfaz a una nueva plataforma

- **Entorno:** tiempo de diseño
- **Artefacto:** interfaz de usuario
- **Respuesta:** se adapta la interfaz a la nueva plataforma
- **Medición de respuesta:** se adaptan los cambios en menos de 50hs

Atributo de modificabilidad

'Quiere una interfaz similar al representante de empresas con derechos de televisión (Maxi) para controlar publicidades en las simulaciones y el sitio en general.'

- **Fuente:** administrador
- **Estímulo:** modificar publicidad
- **Entorno:** tiempo de ejecución
- **Artefacto:** interfaz para el manejo de publicidades
- **Respuesta:** se hace modificación de las publicidades
- **Medición de respuesta:** se muestran las nuevas publicidades en menos de 20 segundos

Atributo de modificabilidad

'Plantea necesidad de regionalizar la plataforma, debido a lo limitado y la mala calidad del hardware/-servidores disponibles para la plataforma en las regiones iniciales, sobre todo teniendo en cuenta que los streams de video pasan por "dentro" del sistema. No se está hablando de tercerizar el servicio a sitios como Vimeo o Youtube, sino que el tráfico pase de alguna manera por los servidores del sitio. Los enlaces físicos/hardware subyacente implica una cantidad de usuarios limitada / máxima por servidor'

- **Fuente:** Interna
- **Estímulo:** Incorporación una nueva región al sistema
- **Entorno:** normal
- **Artefacto:** sistema
- **Respuesta:** se incorpora una nueva región
- **Medición de respuesta:** el setup de la configuración se hace en menos de 5 horas

Atributo de modificabilidad

'También se quiere aumentar el caudal de redes sociales que se utilizan para "afectar" las estadísticas (no solo Twitter, sino incorporar Facebook, Google+, etc.)'

- **Fuente:** desarrollador
- **Estímulo:** incorporar una nueva red social al sistema
- **Entorno:** tiempo de diseño
- **Artefacto:** sistema
- **Respuesta:** se incorpora la nueva red social
- **Medición de respuesta:** se emplean menos de 40 hs

Atributo de modificabilidad

'Quiere que el énfasis se de en mejorar el módulo de simulación, para que sea lo más real posible. Tiene contacto con asociaciones de jugadores e incluso jugadores, técnicos y periodistas deportivos de diferentes deportes, para ayudar a mejorar el motor de reglas/simulación. También tiene contacto con empresas de redes sociales/sentiment analysis, para ayudar a interfacear con las mismas, y mejorar el módulo de menciones/popularidad de los jugadores. El espíritu es que toda la simulación pueda irse mejorando poco a poco hasta que represente lo mejor posible la realidad sin que sea un dolor de cabeza introducir cambios.'

- **Fuente:** sponsor, stakeholder
- **Estímulo:** agregar nuevas reglas/acciones al motor de simulación
- **Entorno:** tiempo de diseño
- **Artefacto:** motor de simulación
- **Respuesta:** reglas/acciones nuevas agregadas sin efectos secundarios
- **Medición de respuesta:** se invierten menos de 10hs hombre

2.5. Atributos de usabilidad

Atributo de usabilidad

'Quiere poder que él y sus administradores de confianza puedan ver un dashboard en tiempo real del estado de cuenta del sitio de cada una de las regiones y niveles (incluye locales, continentales, global, etc...) y de cualquier grupo de participantes.'

- **Fuente:** administrador
- **Estímulo:** ver estado de cuenta del sitio
- **Entorno:** tiempo de ejecución
- **Artefacto:** sistema
- **Respuesta:** el sistema provee un dashboard con el estado de cuenta del sitio de cada una de las regiones
- **Medición de respuesta:** el administrador es capaz de administrar / comprender la información suministrada en menos de 5 minutos

Atributo de usabilidad

'Aunque no es su responsabilidad, le interesa que la interfaz gráfica de usuarios tenga la calidad de un "juego", sobre toda al momento de ver a los jugadores, las jugadas de los técnicos, colocar el nombre y logo del equipo del participante, etc... con animaciones y efectos especiales con aceleración gráfica (blurs, iluminación dinámica, depth of field, etc).'

- **Fuente:** usuario
- **Estímulo:** el usuario desea administrar su equipo
- **Entorno:** operación normal
- **Artefacto:** Interfaz web / Móvil
- **Respuesta:** se muestra una interfaz gráfica llena de animaciones en donde el usuario puede administrar su equipo
- **Medición de respuesta:** el test de usabilidad supera el 95 % de satisfacción

Atributo de usabilidad

'Usabilidad del subsistema controlador de publicidades en las simulaciones y en el sitio'

- **Fuente:** Usuario administrador de publicidades
- **Estímulo:** Desea minimizar el impacto de sus errores al configurar publicidades
- **Entorno:** Runtime
- **Artefacto:** Sistema
- **Respuesta:** Se provee un botón de cancelación para volver a la configuración anterior
- **Medición de respuesta:** Los cambios realizados se vuelven atrás y las publicidades no cambian.

Atributo de usabilidad

'Usabilidad del subsistema controlador de publicidades en las simulaciones y en el sitio'

- **Fuente:** Usuario administrador de publicidades
- **Estímulo:** Desea estar seguro de dónde se muestra en el sistema la publicidad que está modificando
- **Entorno:** Runtime
- **Artefacto:** Sistema
- **Respuesta:** Se provee una previsualización de cada publicidad, ubicada en el entorno gráfico que corresponde, acompañada de un texto descriptivo.
- **Medición de respuesta:** Durante 1 hora se le enseña a dos personas a usar la interfaz de ABM de publicidades y luego se les solicita hacer cambios en todas las publicidades. Modificarán correctamente al menos el 80 %.

Atributo de usabilidad

'Usabilidad del subsistema controlador de publicidades en las simulaciones y en el sitio'

- **Fuente:** Usuario administrador de publicidades
- **Estímulo:** Desea insertar la misma publicidad en muchos lugares del sistema de forma eficiente
- **Entorno:** Runtime
- **Artefacto:** Sistema
- **Respuesta:** Se provee una funcionalidad especial para cargar una publicidad eligiendo múltiples sitios.
- **Medición de respuesta:** Toma a lo sumo 3 clicks adicionales cargar una publicidad en muchos sitios que cargarla en un único sitio (además de todos los clicks necesarios para seleccionar los distintos sitios) (suponiendo que no se cometen errores en la selección de sitios).

Atributo de usabilidad

'Usabilidad del subsistema de pagos y cobros'

- **Fuente:** Usuario
- **Estímulo:** Desea estar tranquilo de que ingresar en el sistema su número de tarjeta de crédito o cuenta corriente es seguro.
- **Entorno:** Runtime

- **Artefacto:** Sistema
- **Respuesta:** Se muestran los nombres de las autoridades que auditaron la seguridad del sistema y los documentos que lo prueban.
- **Medición de respuesta:** En menos de 5 minutos el usuario se anima a ingresar sus datos.

3. Arquitectura

En esta sección presentamos la arquitectura que proponemos para cumplir con los atributos de calidad ya planteados.

3.1. Diagrama de nivel 0

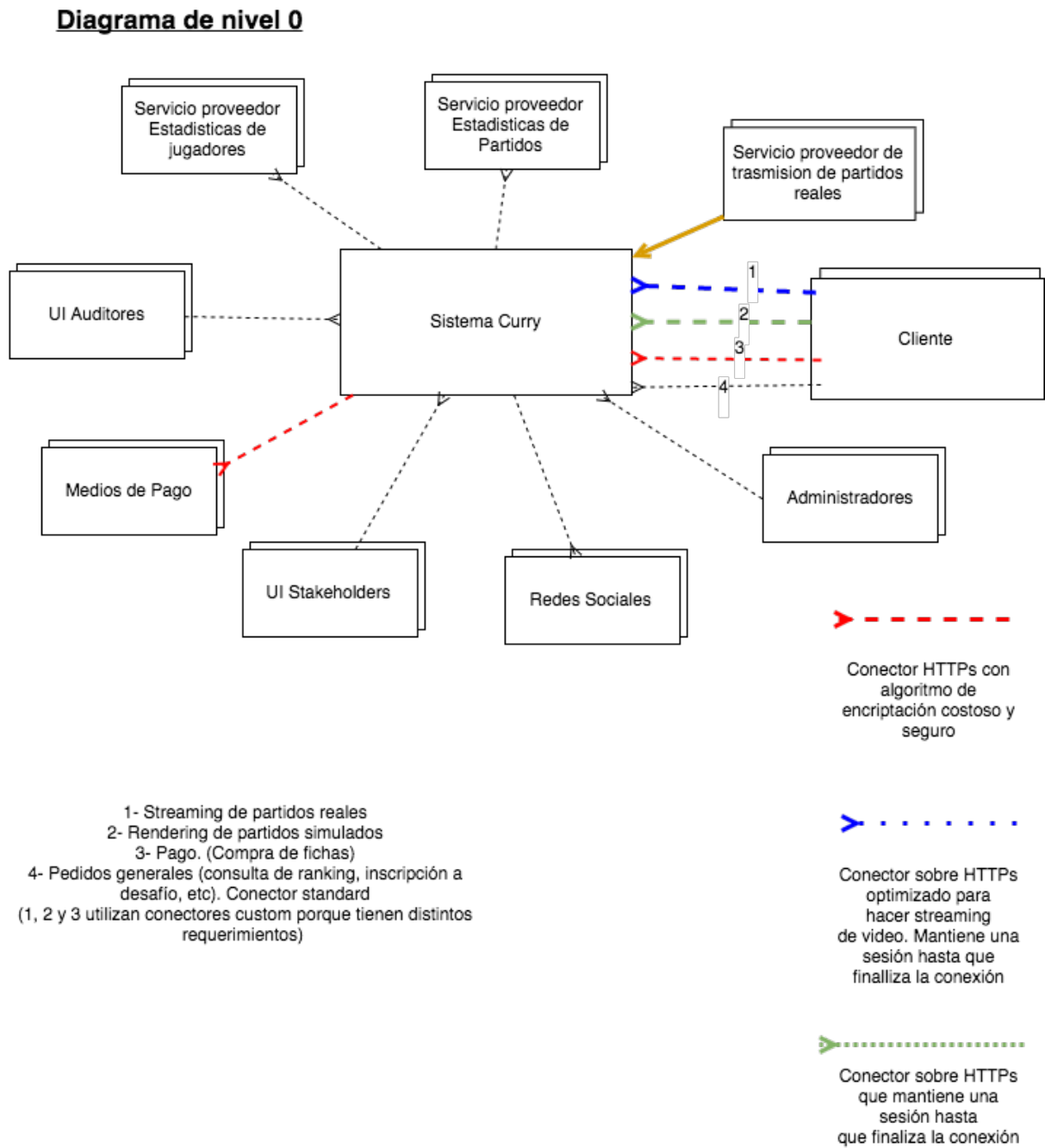


Figura 1: Diagrama de Arquitectura de nivel 0.

3.2. Diagrama de nivel 1

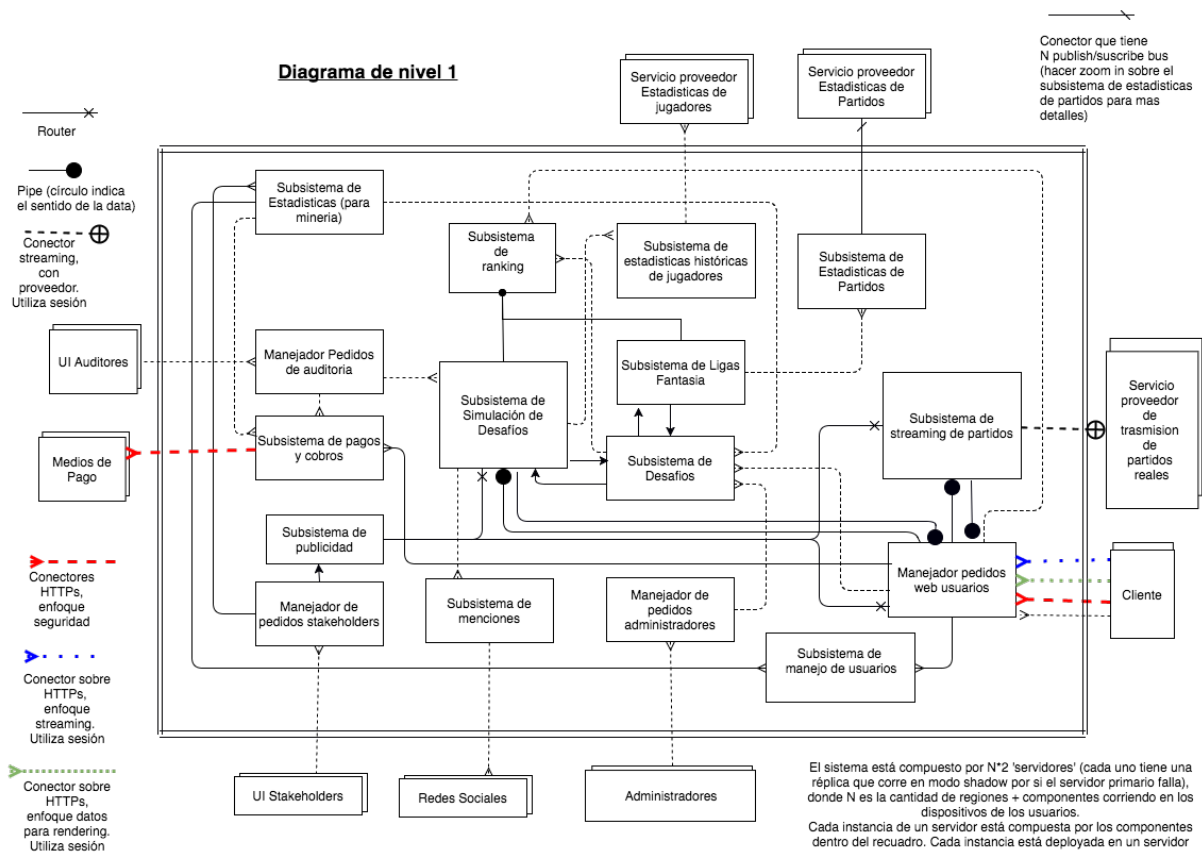


Figura 2: Diagrama de Arquitectura de nivel 1.

El sistema se compone de N de estas instancias. Cada una representa el software que corre en un servidor. N representa la cantidad de regiones (mínima unidad geográfica) en donde corre el sistema. Cada instancia está desplegada en un servidor independiente, ubicado en dicha región. Cada uno de estos servidores tiene una réplica que corre en modo shadow por si el servidor primario falla. El sistema se compone también del conjunto de componentes que corre en los dispositivos de los usuarios.

Cada servidor almacena datos locales a su región. Las estadísticas que almacena, la información de desafíos, la información de sus usuarios son en su gran mayoría regionales, con la excepción de que dicho servidor esté siendo utilizado para algún desafío como nodo intermedio en el árbol de jerarquía de desafíos extra-regionales. (Ver documento jerarquía global). En ese caso, almacena y propaga los datos del desafío que está simulando.

La comunicación entre el cliente y el servidor se realiza de 4 formas diferentes: Streaming de partido real, obtención de datos para rendering de simulación, realizar una compra de fichas para apostar o extracción de dinero y finalmente pedidos de consulta web comunes como ser: registración, login, consulta de ranking, creación de equipo, participación de desafío, etc. Llamamos 'pedido general' a esta última categoría. Cada una de estas 4 formas de comunicación utiliza un conector distinto ya que necesitan satisfacer diferentes requerimientos. Estos conectores tienen un estilo call-return para poner énfasis en que la comunicación es sincrónica: Un usuario hace un pedido de streaming. A partir de ese momento se inicia una sesión entre el servidor y el usuario para mandar un flujo de datos a través del canal de comunicación que dura hasta que finaliza la conexión.

El subsistema de desafíos maneja tanto las simulaciones como las ligas de fantasía.

'Currification' tiene un acuerdo con las empresas televisivas proveedoras de transmisiones, mediante la cual permite crear enlaces para satisfacer los pedidos de los usuarios para ver partidos televisados. El manejador de pedidos recibe los datos de cada transmisión a través de un pipe, y resuelve a través de un

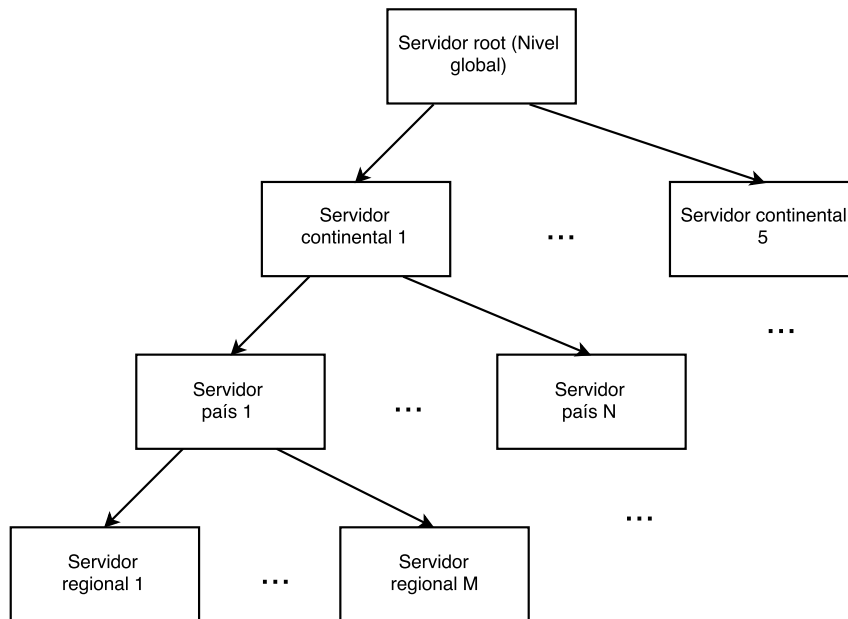
mapping sesiones de usuarios vs transmisiones para saber qué transmisiones debe enviarle a cada usuario. Una situación análoga sucede con las simulaciones.

Los stakeholders pueden agregar publicidades a través de una interfaz gráfica intuitiva y simple. Esta les permite decidir en qué lugar mostrarán cada publicidad: UI de la web, rendering de simulaciones o streaming de partidos. Por este motivo, el subsistema de publicidad rutea las nuevas publicidades a los subsistemas correspondientes.

Los controles de auditoría se hacen sobre los subsistemas de pagos y cobros (para corroborar que no haya movimientos irregulares de dinero), y sobre el subsistema de simulación de desafíos, verificando que el hash del código de simulación no haya sido modificado.

La autenticación y restricción al sitio por parte de los usuarios se realiza a partir de reglas: Algunas direcciones de IP están restringidas. Correspondientes a aquellas zonas en donde están prohibidas las apuestas web. Por otro lado, cuando un usuario asocia una tarjeta a su cuenta el sistema verifica que la tarjeta sea local a la región a la cual se encuentra. Finalmente, cuando un usuario quiere realizar una compra utilizando una tarjeta, esta última debe coincidir con aquella que ha sido registrada previamente.

3.3. Jerarquía Virtual



Cuando desea simularse un desafío en donde intervienen usuarios de más de una región, los distintos servidores intercambian mensajes para organizarse en una jerarquía virtual (lógica) cuyo propósito es evitar que el servidor en donde se ejecuta la simulación del desafío deba satisfacer la totalidad de los pedidos sobre dicho desafío. En el caso de los desafíos globales esta demanda puede llegar a ser inmensa.

El mecanismo funciona de la siguiente manera:

Se selecciona un servidor entre la totalidad para funcionar como raíz. Dicho servidor va a ser el encargado de realizar la simulación del desafío. Dicho servidor selecciona un servidor por continente. Estos últimos constituirán el nivel 1 del árbol.

A medida que va generando los datos que serán procesados por los renderizadores, su responsabilidad es streamear los datos hacia los servidores del nivel 1. En particular: Al "receptor de datos de simulación" del subsistema de simulación.

Se presenta una situación análoga para el caso de los servidores continentales. Cada uno actuará como raíz dentro de su propio continente. Cada uno de estos seleccionará N servidores, uno por cada país del continente en donde esté corriendo la aplicación, que constituirán el nivel 2 del árbol.

Finalmente, los servidores de cada país serán los encargados de propagar los datos de la simulación del evento global/continental/etc. a cada servidor regional que se encuentre del mismo país.

Independientemente del nivel del árbol al que pertenezcan para un determinado desafío, los servidores continúan siendo servidores regionales. Por lo que en todo momento deben seguir cumpliendo sus funciones como tales. Por ejemplo, si se determina que el desafío de la final mundial será simulado en Buenos Aires, el servidor de Buenos Aires no sólo deberá propagar los datos de dicha simulación a los servidores continentales, sino que también deberá satisfacer cualquier pedido que venga de un usuario radicado en Buenos Aires.

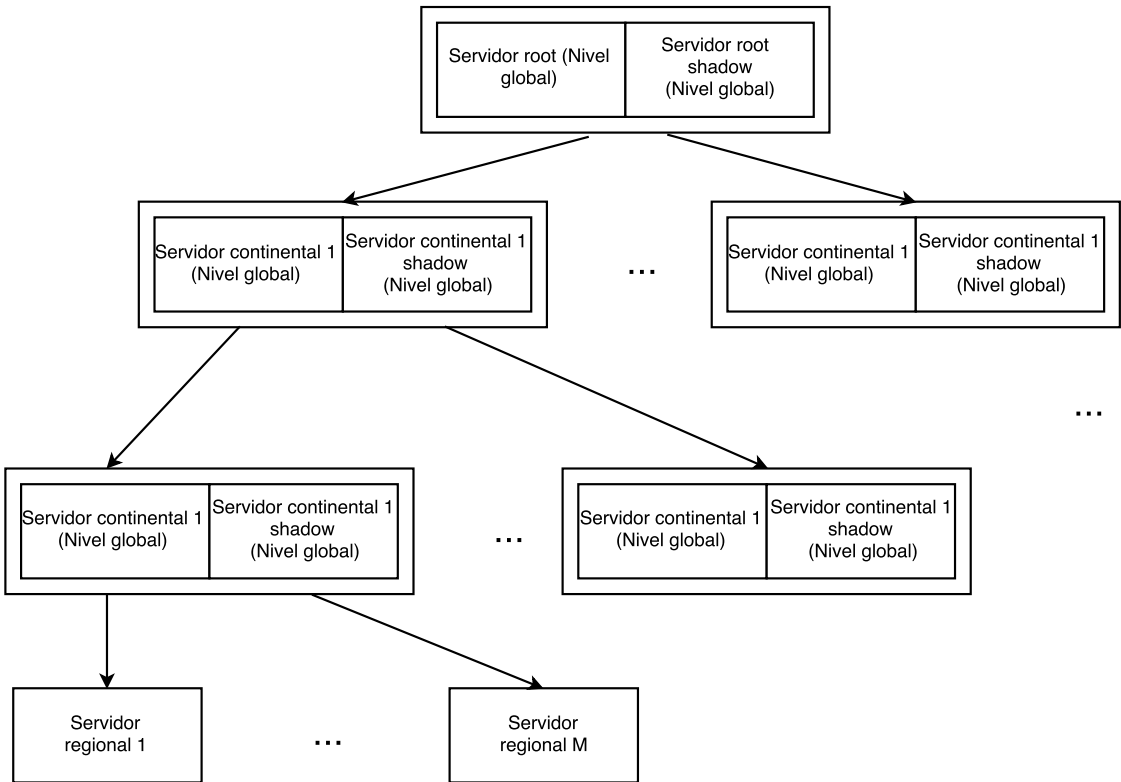
Notar que esta jerarquía no permite que un usuario Americano pueda ver un desafío continental europeo, por ejemplo. En estos casos especiales, el usuario del servidor regional deberá suscribirse explícitamente al servidor que está simulando el desafío. (Ver diagrama de c&c, subsistema de simulación)

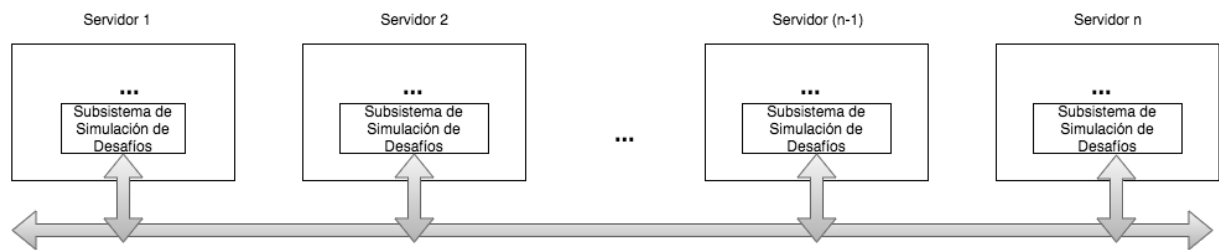
Garantizando disponibilidad

La solución anteriormente presentada tiene un problema subyacente. Qué pasa si uno de los servidores que no son hojas falla? Por ejemplo, el servidor que actúa como raíz. En ese caso, se detendría la simulación por lo cual el desafío quedaría suspendido. La solución propuesta consiste en tener un servidor que actúe en modo shadow por cada nodo intermedio.

La idea es que ambos servidores se monitoreen entre sí. Si alguno detecta que el otro falla deberá seleccionar un nuevo servidor para reemplazarlo. Es decir, siempre deben haber 2 servidores funcionales por cada nodo intermedio.

Cada dupla deberá mantenerse sincronizada.





3.4. Subsistema de Desafíos

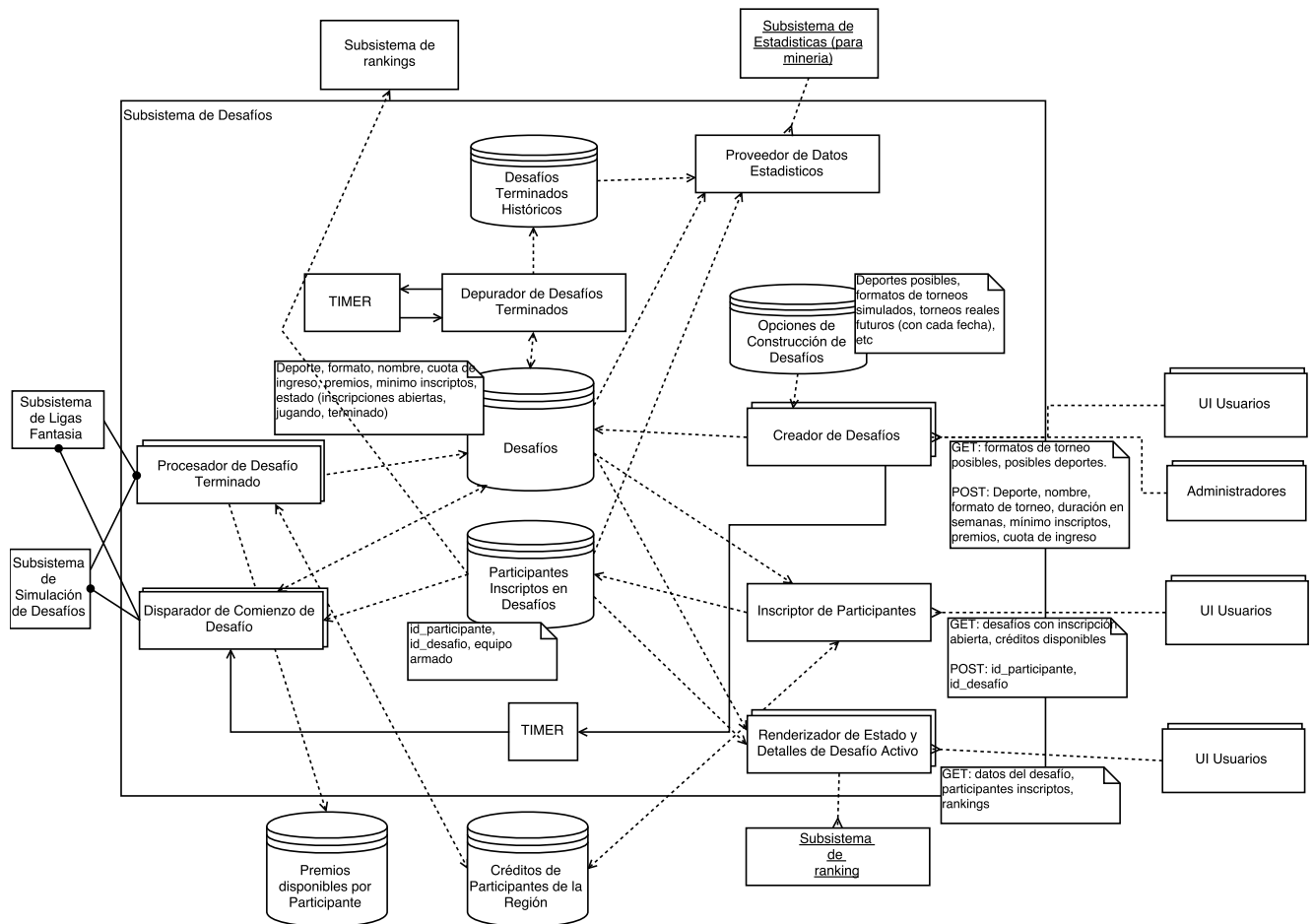


Figura 3: Subsistema de Desafíos.

El subsistema de desafíos se encarga de crear desafíos, inscribir participantes, iniciar desafíos en el momento indicado, proveer detalles y estado de cada desafío a los usuarios. Cuando un desafío comienza se encarga de informar al subsistema de simulación o liga de fantasía según corresponda. Además se encarga de cobrar créditos de inscripción y de pagar premios (tanto en créditos como premios especiales, que luego los usuarios podrán cambiar por el premio real en dinero o lo que corresponda). También provee una interfaz de consulta de estadísticas de desafíos.

- **Desafíos Terminados:** Los desafíos finalizados durante la última semana van a tener más solicitudes de consulta de estado. Es por esto que esos desafíos se mantienen en el repositorio principal de desafíos con redundancia activa para mejorar la disponibilidad y la performance. Pasados los 7 días de finalizado, un proceso que ejecuta una vez por día se encarga de depurar el repositorio principal para mejorar la performance de las búsquedas, asumiendo que esos desafíos serán consultados con menos frecuencia.
- **Creador de Desafíos:** Devuelve todas las opciones posibles para construir un desafío (tanto para modo simulado como para modo liga de fantasía). Luego recibe la configuración elegida y crea el desafío. Se almacena en un repositorio local que luego se propaga a todas las réplicas regionales.
- **Inscripción de participantes:** Muestra una lista con todos los desafíos por cada deporte (tanto simulados como liga de fantasía) que aún estén con la inscripción abierta (aún no comenzaron a jugarse).

Por cada uno muestra el valor en créditos para ingresar. Luego se inscribe un participante (siempre y cuando los créditos le alcancen). Para mejorar la performance y disponibilidad (muchos podrán intentar inscribirse a la vez), luego de hacer el chequeo se guardan en una caché todos los pedidos de inscripción que luego se van persistiendo por procesos dedicados.

- **Renderizador de estado y detalles:** Devuelve el estado de un desafío específico (si está abierto a inscripciones, si se está jugando o si está terminado) así como también la cuenta regresiva para que empiece el desafío (y se cierren inscripciones) y todos los detalles asociados (premios, cuota de inscripción, tipo de desafío, formato de torneo o fechas que se juegan, etc.), y el ranking, si corresponde. Internamente se utiliza una caché. Cada pedido que llega se guarda en un repositorio y se solicitan los datos a un manager de cache (que se crea en el momento que se necesita) y a un proceso que busca los datos en los repositorios persistentes (también se crea cuando se necesita). Si el dato llega de la caché, se mata al proceso que fue a disco (y también se mata al proceso de la caché). En cambio si no estaba en la caché, el dato llega del disco (y ahí también se guarda en la caché para soportar eficientemente un período corto de muchos pedidos de detalles del mismo desafío. Un proceso se ejecuta cada 1 minuto y borra los datos de caché que tengan más de un minuto de antigüedad. Dado que los datos no deben mostrarse en tiempo real, este mecanismo mejora la disponibilidad y la performance de respuesta a muchas solicitudes de detalles de un desafío popular.

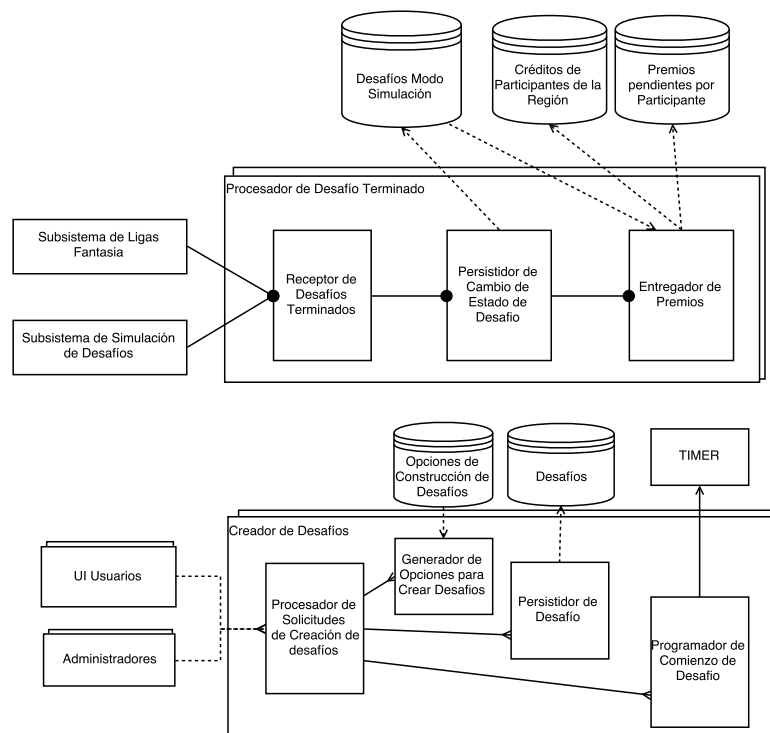


Figura 4: Zoom en Procesador Desafío Terminado y Creador de Desafíos

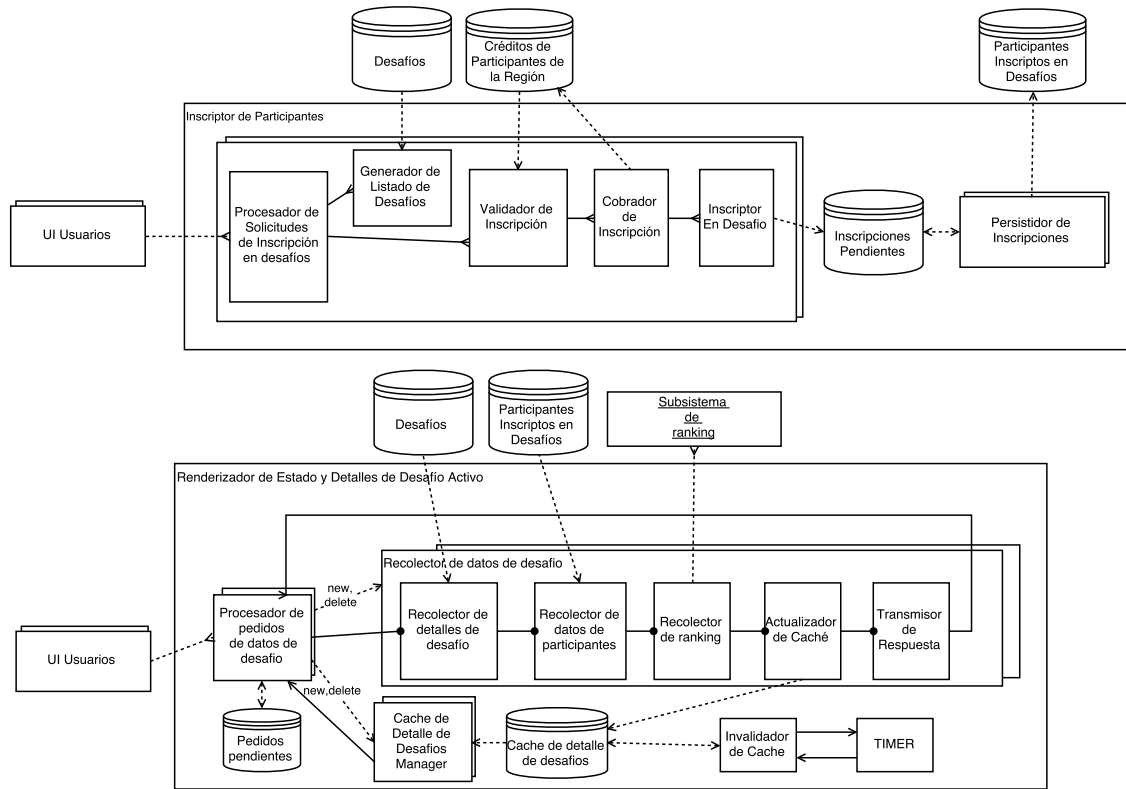


Figura 5: Zoom en Inscriptor de Participantes y Renderizador de Estados y Detalles Desafío Activo

3.5. Subsistema de Simulacion

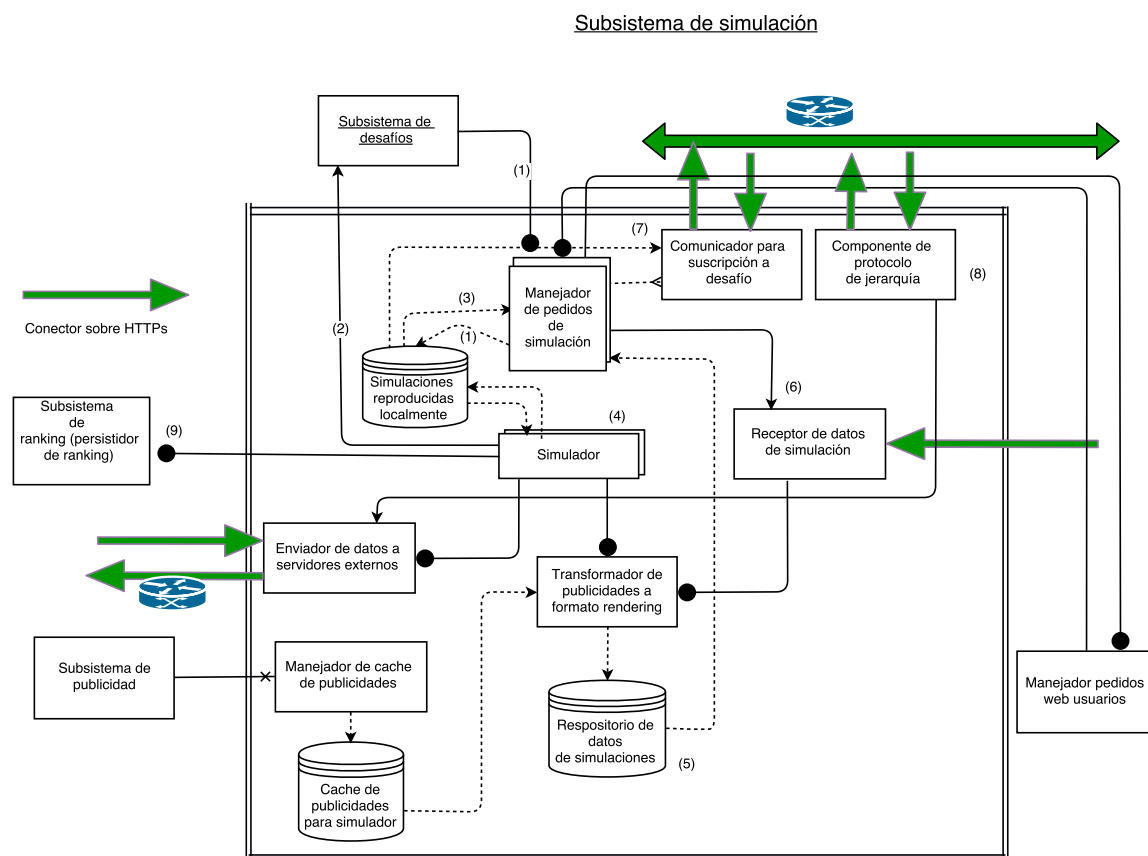


Figura 6: Subsistema de Simulacion.

Idea general:

- Relación subsistema de simulación/subsistema de desafíos:
- (1) El subs. de desafíos **inicia una simulación** de forma asincrónica con el subs. de simulación. Agrega dicha simulación al repositorio de **'Simulaciones reproducidas localmente'** para que pueda comenzar a ser ejecutada por los **simuladores**.
- (2) El subs. de simulación comunica **fin de simulación** y resultado de forma asincrónica con el subs. de desafíos
- (3) El manejador lee del repositorio saber si el desafío correspondiente al pedido está siendo simulado por el simulador o requiere de un servicio de simulación externo (Procesado en otro servidor del sistema)
- (4) Los **simuladores** están replicados para mejorar la performance
- (5) Se utiliza un repositorio para guardar los datos ya procesados para el rendering de los usuarios
- (6) En el caso de que el manejador no encuentre datos para el desafío pedido ni en el **'Repositorio de simulaciones reproducidas localmente'**, ni en el **'Repositorio de datos de simulaciones'**, el manejador se suscribe al **'Enviador de datos a servidores externos'** del servidor que está ejecutando la simulación.
- (7) Para llevar a cabo el punto (6) y saber qué servidor es el que está corriendo utiliza el **'Comunicador para suscripción a desafío'** de los servidores receptores revisan su repositorio local para determinar si son ellos quienes están ejecutando dicha simulación. Esto permite que por ejemplo, un usuario de América pueda ver la simulación de un desafío continental europeo. Notar que según el **documento de jerarquía global**, los servidores americanos no están incluidos en la rama del continente europeo. Por lo tanto la suscripción debe realizarse de forma explícita.
- (8) Se utiliza un **'Componente de protocolo de jerarquía'** para comunicarse con los demás servidores y determinar qué rol juega el servidor en el desafío extra-regional. Si por ejemplo le correspondiera el papel de root, debe suscribir en el **'Enviador de datos a servidores externos'** a los servidores continentales correspondientes. (Ver **documento de jerarquía global**)
- (9) Cuando un jugador mete un punto en una simulación, debe informarse este hecho al subsistema de ranking para aumentar los puntos de dicho participante

Enviador de datos a servidores externos

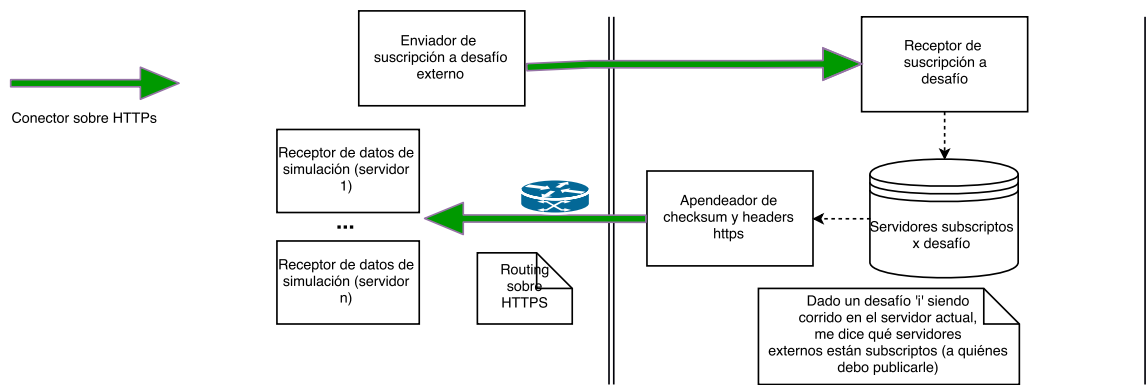


Figura 7: Enviador Datos a servidores externos.

Receptor de datos de simulación

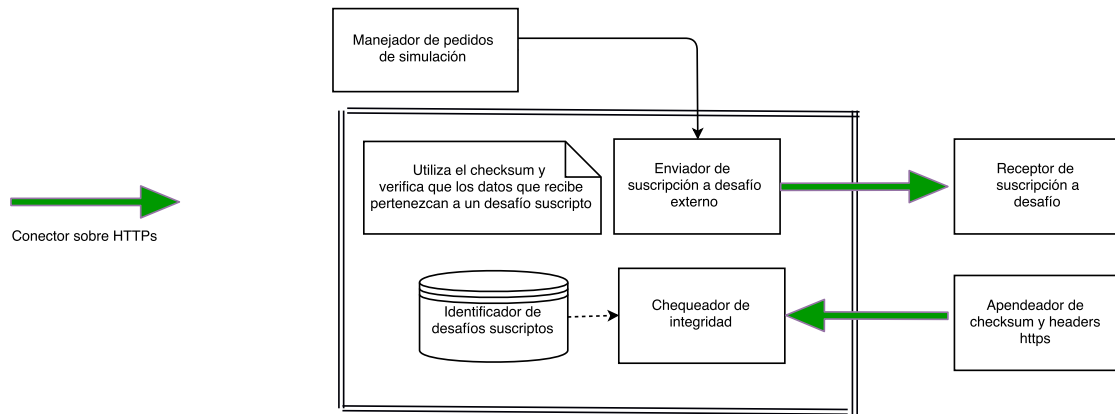


Figura 8: Receptor de datos de Simulación.

3.6. Subsistema de Ligas de Fantasia

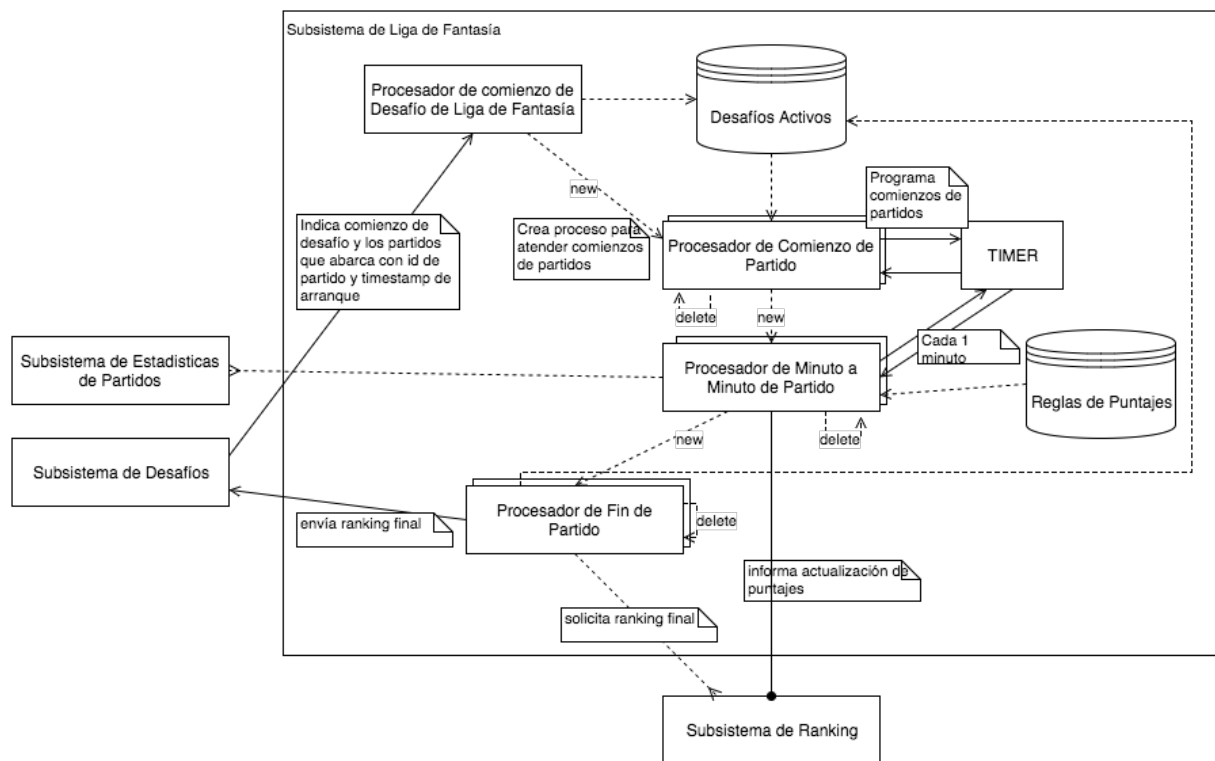


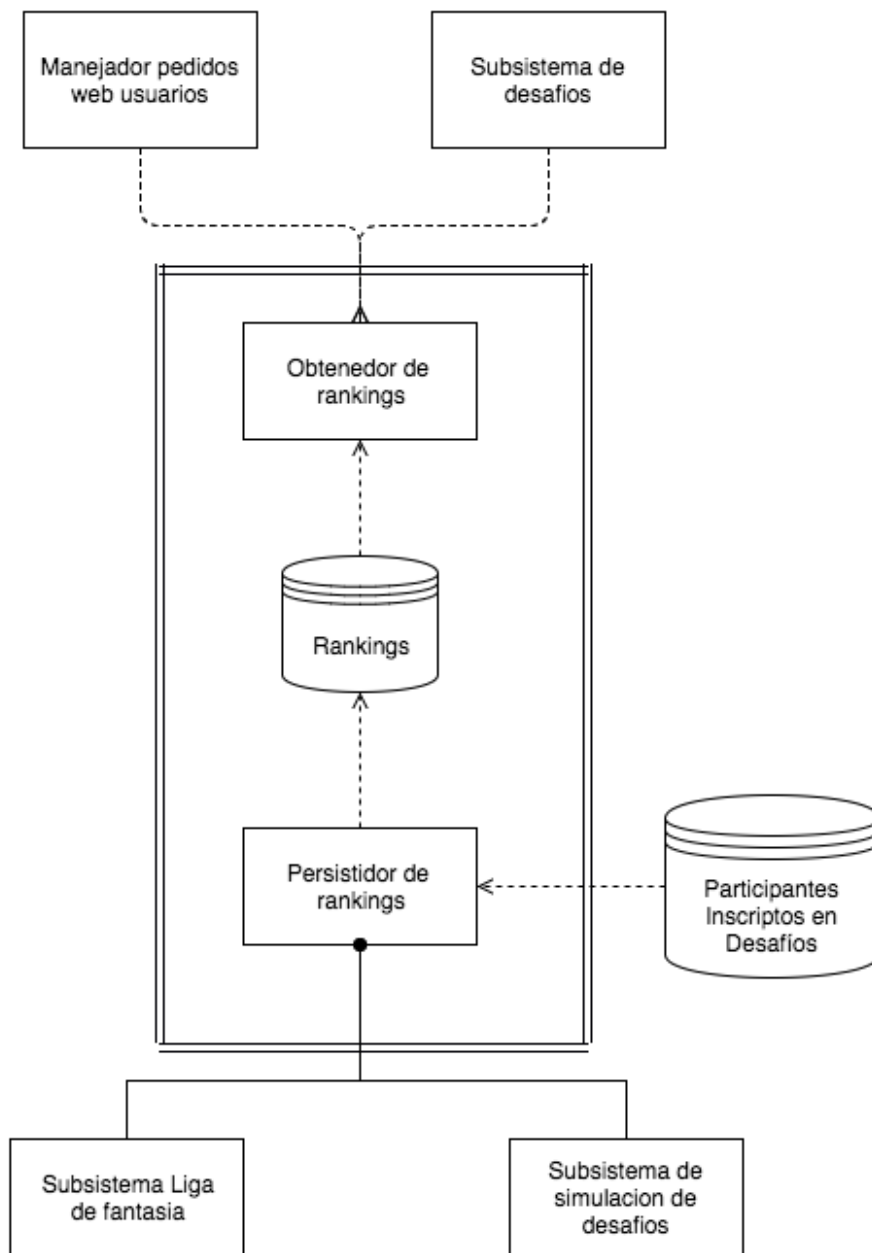
Figura 9: Subsistema de Ligas de Fantasia.

El procesador de comienzo de desafío almacena el desafío nuevo en el repositorio de desafíos activos, junto con todos los partidos a tener en cuenta. Luego crea un procesador de comienzo de partido, que se encarga de revisar si el desafío indicado tiene algún partido que comenzar, y en caso que no sea así, programa un timer para que le avise cuándo comienza el próximo partido. En este esquema, se crea un procesador de comienzo de partido para cada desafío nuevo que llega. Cada vez que hay un partido, el procesador de comienzo de partido crea un procesador de minuto a minuto de partido. Este último, cada un minuto, solicita al Subsistema de Estadísticas de Partidos las últimas actualizaciones. Con ellas evalúa las reglas de puntajes y actualiza los rankings (por ejemplo, una regla podría ser "Si mete gol, suma 3 puntos", y si una actualización indica que Agüero metió un gol, entonces se envía al ranking A todos los participantes cuyo equipo tenga a Agüero suma 3 puntos). Una vez actualizado los rankings, vuelve a programar el timer por un minuto. De esta manera habrá un procesador minuto a minuto por cada partido activo, que en cada minuto actualizará los puntajes del ranking.

El procesador de comienzo de partido, luego de crear el de minuto a minuto, se programa para el próximo partido. Si no hay más partidos, simplemente se destruye a sí mismo.

Cuando cada partido termina, el procesador de minuto a minuto se destruye a sí mismo, pero antes avisa del fin de partido al procesador de fin de partido, que verifica si fue el último del desafío y en tal caso informa de la finalización del partido al subsistema de desafíos.

3.7. Subsistema de Ranking



El subsistema de rankings se encarga de mantener un repositorio con el ranking de cada participante de la región. Funcionamiento:

La idea es que el subsistema de liga fantasía y el subsistema de simulación de desafíos envíen minuto a minuto mensajes del estilo "sumale X puntos a todos los equipos que tengan Y jugador". Frente a esto, el persistidor de rankings consulta el repositorio Participantes Inscriptos en Desafíos (del subsistema de desafíos) y obtiene todos aquellos participantes que cumplen con el mensaje previamente mencionado. A estos participantes les suma X puntos para luego persistirlo en el repositorio de rankings.

Figura 10: Subsistema de Rankings.

3.8. App Cliente

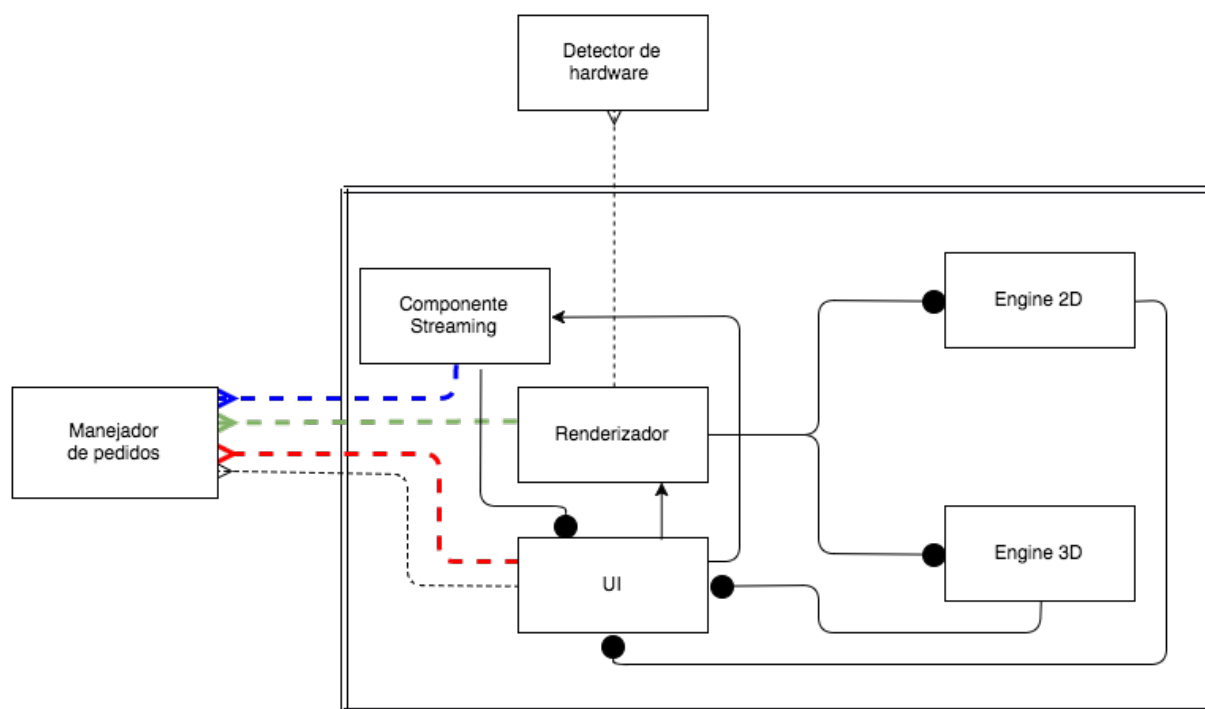


Figura 11: Aplicacion del lado del Cliente.

3.9. Manejador pedidos Cliente

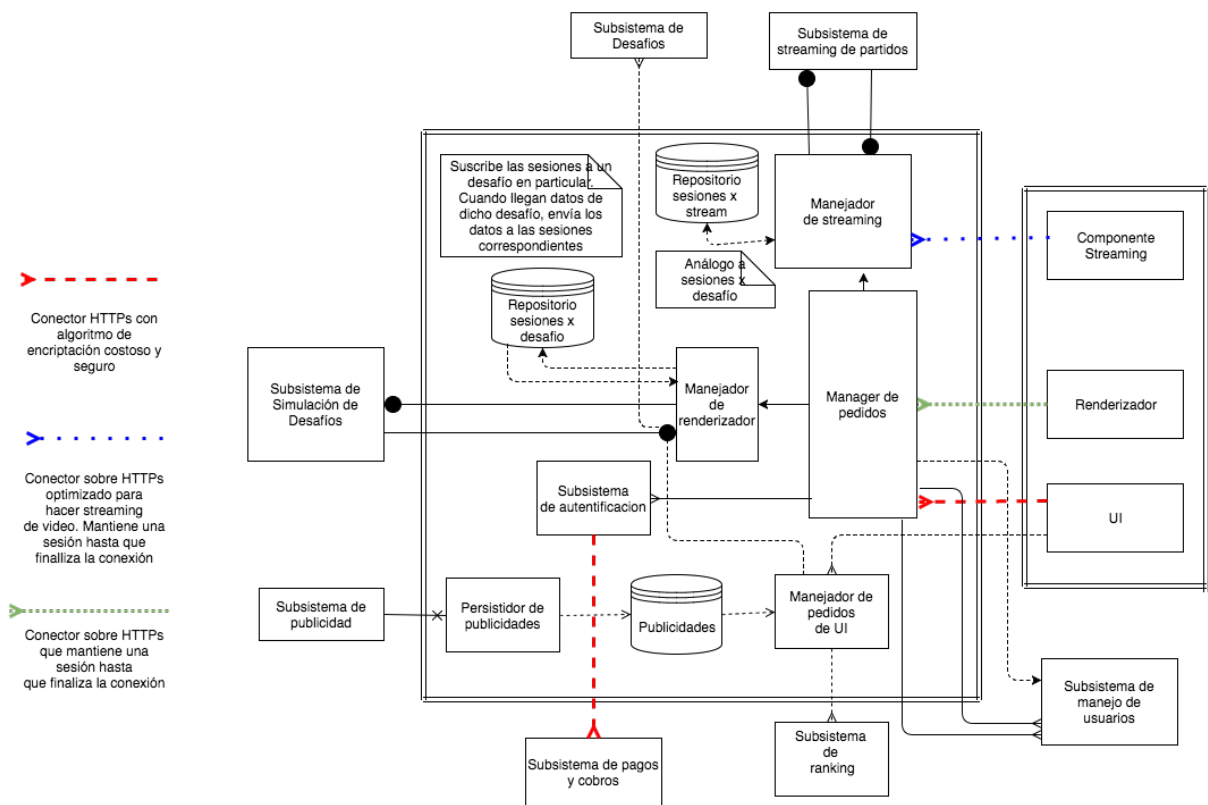


Figura 12: Manejador de pedidos realizados por el cliente.

3.10. Subsistema de Usuarios

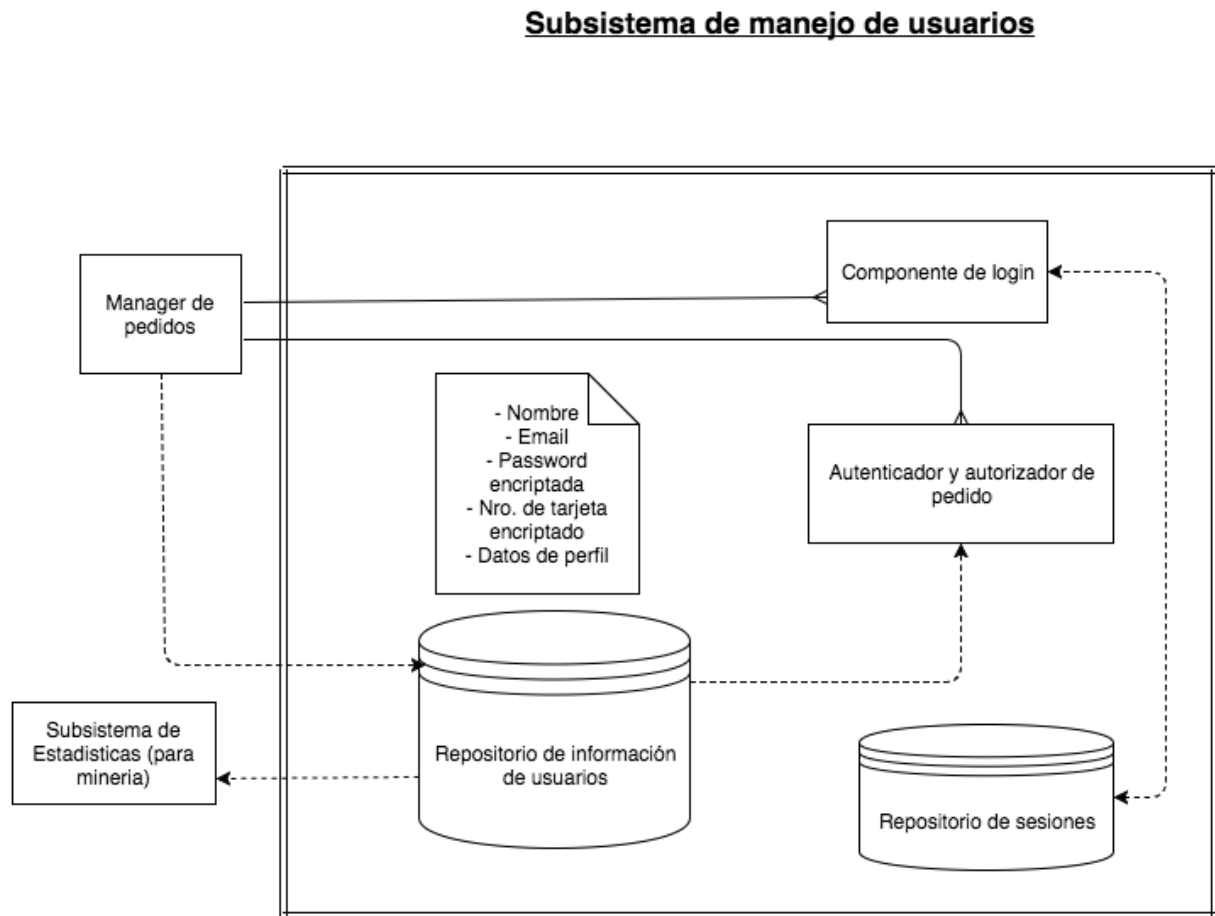


Figura 13: Subsistema de manejo de usuarios.

Cuando un usuario se registra, el manejador ingresa la información en la base de datos. Para interactuar con el sistema, el usuario inicia una sesión. Ante cada pedido, el sistema autoriza que el usuario que generó el pedido tenga suficientes permisos para hacerlo

3.11. Subsistema de Cobro y Pagos

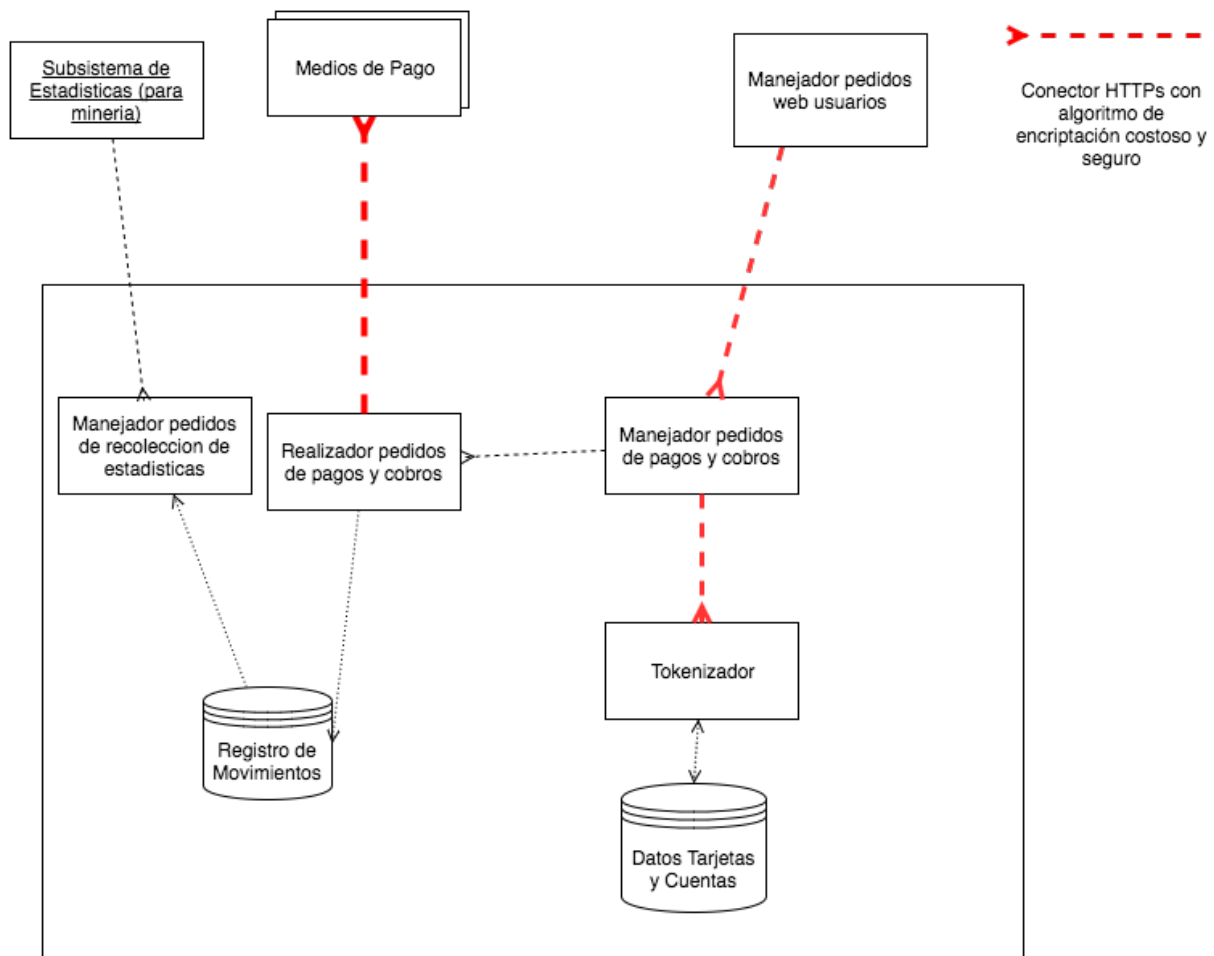


Figura 14: Subsistema de cobros y pagos.

Los datos de tarjetas y cuentas y el Tokenizador corren en una maquina distinta a todo el resto. Esto se hace para aislarla lo maximo posible y evitar cualquier vulnerabilidad que pueda llegar a tener el resto del sistema. Ademas estas maquinas deberan tener seguridad fisica, para evitar posibles robos y/o violaciones fisicas al sistema.

Funcionamiento:

- Caso medio de pago nuevo: Al manejador le llegan pedidos de pagos y/o cobros, autenticados y con informacion de un medio de pago nuevo. Entonces le pasamos esta informacion a el Tokenizador, que le genera un token, persiste en base de datos la informacion y el token correspondiente y le devuelve el token al Manejador de pedidos.
 - Caso medio de pago existente: Al manejador le llegan pedidos de pagos o cobros, autenticados y con un id o referencia minima (elegida en por el Cliente) de que medio de pago se usara. Entonces con ese id o referencia, le pedimos al Tokenizador y obtenemos el token correspondiente.
- Luego con el token correspondiente, se lo brindamos al Realizador de pedidos, que entiende de tokens y con el se comunica con el medio de pago correspondiente y realiza la accion pertinente brindandole el token al medio de pago.
- Luego con el resultado de la accion, la logeamos en el registro de movimiento y le devolvemos el resultado de la accion al Realizador y este al Manejador

Ademas la información mas sensible (Numero de Tarjeta o Cuenta completa, codigo de seguridad de la tarjeta, etc) seran almacenada encriptada, mediante un algoritmo de clave asimetrica, donde la clave para descryptar la tengan unicamente los dueños del sistema.

De esta forma, cumpliriamos con un estandar de seguridad llamado PCI, el cual es necesario para para poder realizar cobros y pagos con tarjetas de credito y cuentas bancarias. Este estandar sera todo el tiempo testado para verificar que estemos siempre cumpliendo y en norma, ya que en caso de no estarlo estariamos abierto a posibles ataques y/o robos de datos.

3.11.1. Biografia consultada

- Norma PCI.

- Como cuidan sus datos compañías del estilo Amazon.

3.12. Subsistema de Streaming

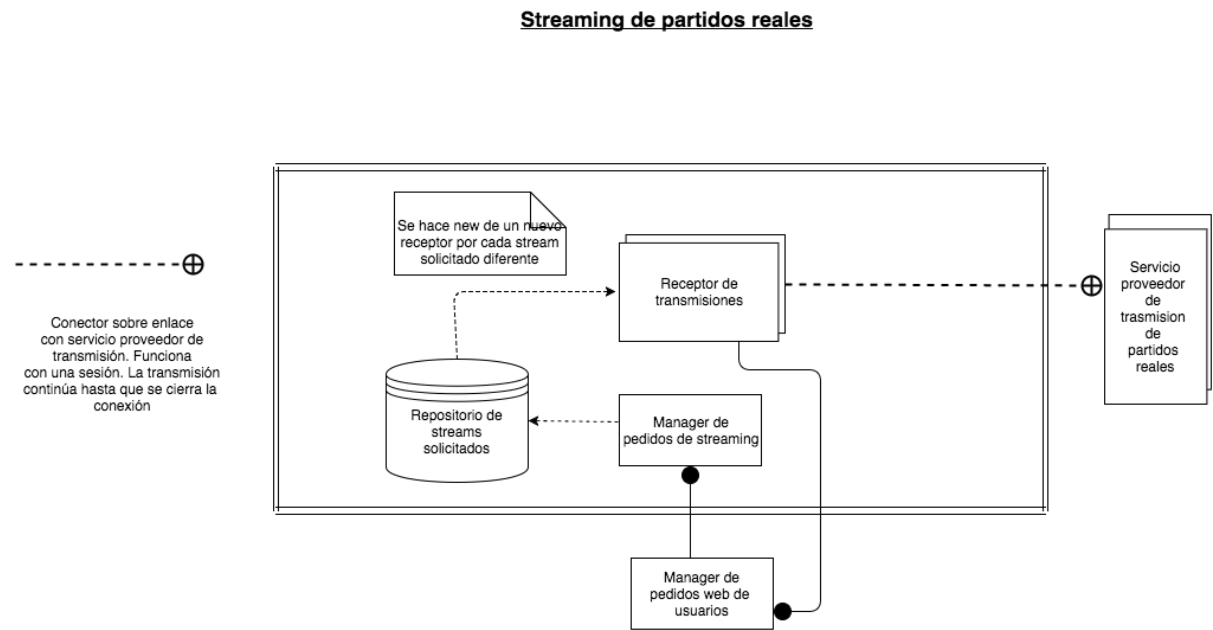


Figura 15: Subsistema de streaming de partidos reales.

3.13. Subsistema de Publicidad

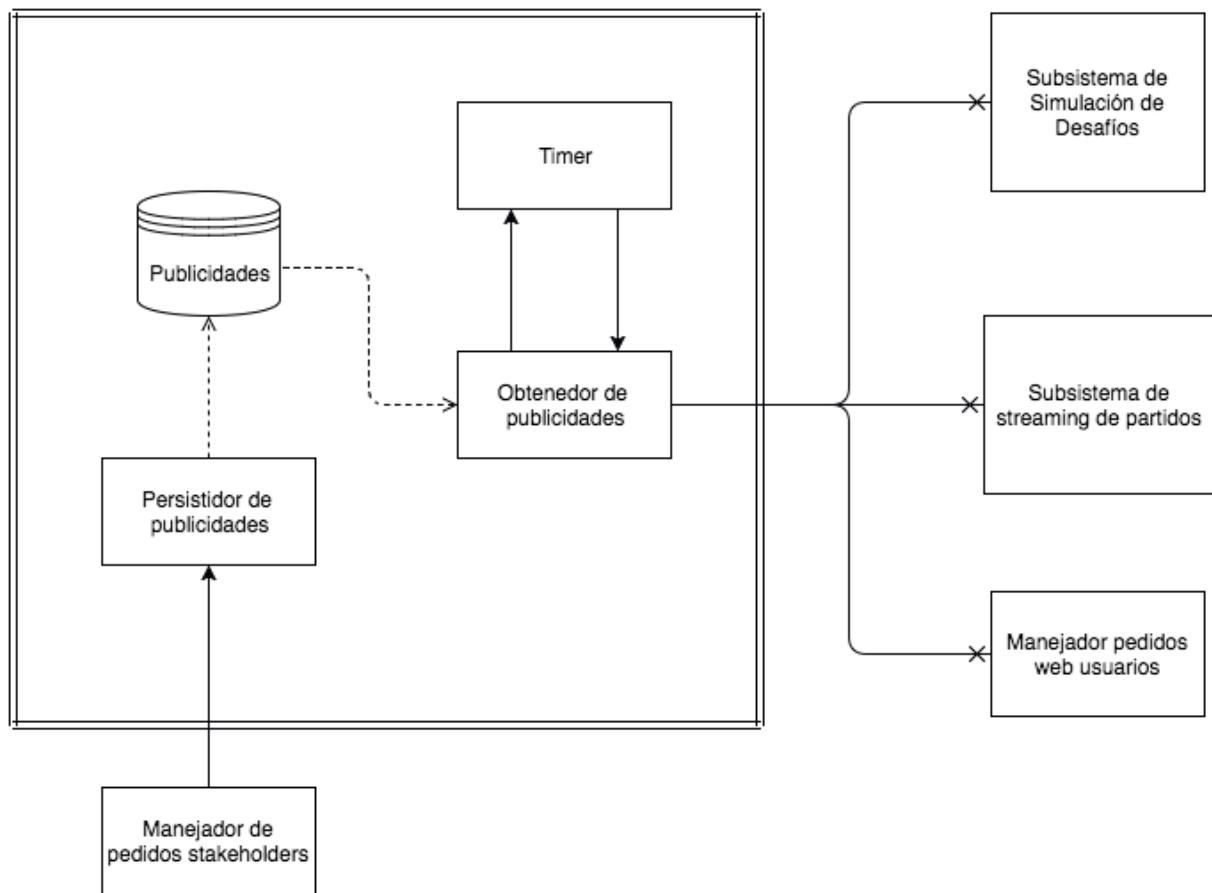


Figura 16: Subsistema de Publicidades.

La idea es que el manejador de pedidos stakeholders le indique al persistidor de publicidades qué publicidad desea guardar, indicándole en dónde deber mostrarse (transmisión de un partido, simulación o página principal del sitio) y en qué momento (hora). Luego, el obtenedor de publicidades consulta periódicamente (cada 3 segundos) el repositorio solicitándole publicidades que deban mostrarse en el momento en que consulta (+/- 3 segundos) y las envía a través de un router a cada componente que hará uso del mismo (subsistema de simulación, subsistema de streaming y manejador de pedidos de usuario)

3.14. Subsistema de Estadísticas de Partidos

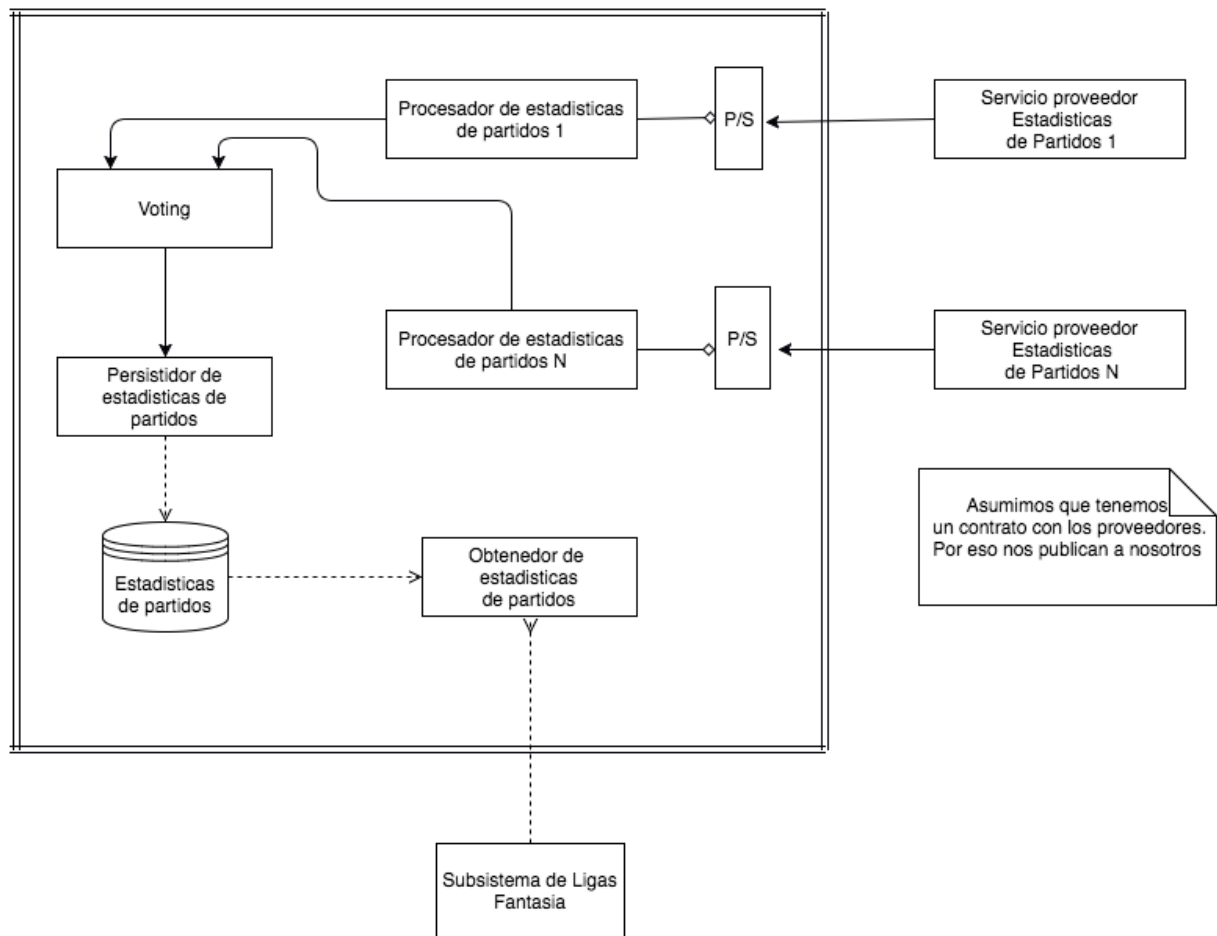


Figura 17: Subsistema de Estadísticas de Partidos.

Los proveedores estadísticos que tenemos contratados publican actualizaciones periódicas de los partidos que se están disputando. Como los datos publicados son propensos a errores decidimos utilizar un voter central que recibe las salidas de los múltiples procesadores y decide el resultado correcto en función de los votos. Estos resultados son persistidos en un repositorio el cual puede ser accedido mediante el obtenedor de estadísticas de partidos.

4. Arquitectura TP1

A continuación presentamos lo que sería un especie de Diagrama de Nivel 1 sobre la Arquitectura de la solución del TP1.

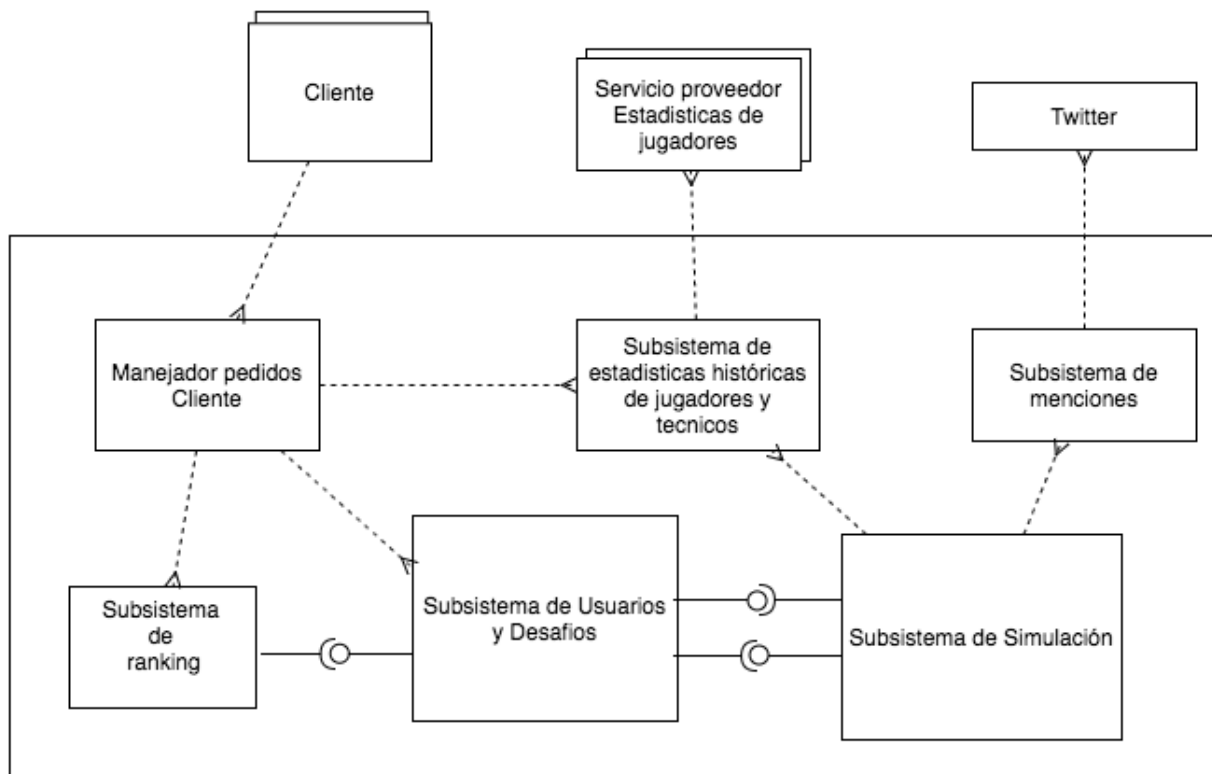


Figura 18: Arquitectura del TP1.

La idea básica es muy similar a la arquitectura propuesta en este trabajo práctico, tenemos varios clientes que nos realizan pedidos, en este caso serían consultas a los Rankings, consultas a las estadísticas de jugadores y técnicos o acciones respecto al sistema de desafíos (creación de cuentas, creación de equipos, creación de desafíos, manejo de fichas, etc). Todos estos pedidos son recibidos por el Manejador de Pedidos, el cual redirige cada pedido al componente correspondiente.

Por otro lado en el componente Subsistema de Usuarios y Desafíos, como ya dije estará toda la lógica sobre el tema de la creación de equipos, desafíos, etc. Este componente luego le brindará la información necesaria del desafío (Equipos, Jugadores, Técnicos) al Subsistema de Simulación para que este pueda simular el desafío. Luego una vez resuelta la simulación, el subsistema de simulación devuelve el log del partido para que el Subsistema de Usuarios y Desafíos haga lo correspondiente, por un lado devolverle el resultado y el log al Manejador de pedidos para que le llegue al cliente, y por otro lado registrar el resultado, actualizar rankings, actualizar cap y pagar apuestas (en caso de ser necesario), etc.

El Subsistema de Simulación para poder simular el desafío consume por un lado las estadísticas históricas de los jugadores y por otro lado las menciones en Twitter, para así de esta forma tener los valores correspondientes a la hora de calcular los umbrales de éxito.

Como podemos apreciar hay conceptos compartidos entre las dos Arquitecturas propuestas, pero la del Trabajo Práctico 1 es mucho más simple y chica, ya que la aplicación en sí era más chica y menos compleja. No tenemos nada sobre ligas de fantasía, dinero real, tarjetas de crédito, streaming de partidos reales, engines 2d y 3d para ver la simulación, etc. Básicamente no teníamos ningún atributo de calidad por cumplir a raja tabla, por lo que es una Arquitectura pensada mucho más en la funcionalidad que en otras cosas.

5. Comparaciones y Conclusiones

5.1. UP vs Scrum

Desde el punto de vista de la planificación, UP y Scrum se parecen en que ambas hacen un desglose en tareas de cada uno de los módulos/user stories a realizar en la siguiente etapa/sprint. Sin embargo, UP realiza asignación de horas y tareas a cada recurso de antemano, mientras que en Scrum cada desarrollador elige la user storie que va a realizar cuando desea, siempre y cuando se llegue con los tiempos de entrega (finalización del sprint). En ambas metodologías se priorizan actividades: módulos o casos de uso en UP mediante el análisis de riesgos y user stories en Scrum mediante la estimación de esfuerzo y valor de negocio.

Desde el punto de vista de la ejecución de las actividades, UP las organiza en etapas mientras que Scrum en sprints. Estos difieren principalmente en que UP podría tener distinta duración de las iteraciones de distintas etapas (pero las iteraciones de una misma etapa duran lo mismo), mientras que en Scrum los sprints son todos de la misma duración. Además en UP se planifican los módulos que se tratarán en cada etapa desde el principio, mientras que en Scrum se elige el sprint backlog justo antes de cada sprint (de hecho podrían agregarse stories nuevas al backlog).

Scrum garantiza entregas en poco tiempo (idealmente al final de cada sprint), mientras que al final de las iteraciones de UP no necesariamente se pueden realizar entregas. Por otro lado, UP permite realizar una planificación a largo plazo de gran parte del proyecto, mientras que Scrum se va adaptando sobre la marcha y se actualiza el futuro del proyecto antes de cada sprint.

5.2. Programming in the Small vs Programming in the Large

Programming in the Small consiste en el modelado de sistemas pequeños, donde todos los detalles son conocidos de antemano y se puede modelar con un gran nivel de detalle. Los requerimientos no funcionales son poco exigentes: pocos usuarios, no hay grandes problemas de performance o disponibilidad; la modificabilidad se logra principalmente desde el diseño orientado a objetos; seguridad y usabilidad son los únicos atributos de calidad que podrían considerarse más en detalle.

Por otro lado, Programming in the Large consiste en el modelado de sistemas muy grandes, que no pueden diseñarse por completo usando clases y objetos, sino que se debe hacer un diseño de alto nivel, partir el problema en módulos (WBS) y atacar los módulos por separado y con una planificación detallada. Los atributos de calidad juegan un rol fundamental ya que la performance y disponibilidad del sistema suelen ser atributos muy importantes y difíciles de satisfacer. Además el dinero juega un papel importante ya que en la mayoría de las veces marcará la diferencia en la mejoría de la calidad del sistema.

PitL requiere una planificación mucho más detallada que PitS, ya que al tratarse de un proyecto de mayor duración temporal y de mayor consumo de recursos es importante que el uso de estos recursos se haga de manera eficiente. Los cambios de requerimientos en un proyecto grande van a generar un mayor impacto dependiendo de en qué fase del proyecto se realicen, es por esto que es fundamental la elicitación exhaustiva en las primeras etapas para evitar cambios de último momento que atrasen la finalización. En cambio las modificaciones en requerimientos de un proyecto pequeño se pueden tratar en todo momento dado que un impacto grande siempre estará limitado por el tamaño del proyecto que sigue siendo pequeño.

5.3. Conclusiones

En general para proyectos de PitS conviene utilizar metodologías ágiles como Scrum, ya que permiten generar entregables rápidamente. En un proyecto PitL las entregas a corto plazo pueden no ser factibles al principio porque quizás haya que construir muchos módulos antes de poder ver algún resultado interesante. Además Scrum permite el cambio de requerimientos antes de cada sprint, mientras que en UP se espera que los cambios de requerimientos sean sólo al principio del proyecto. Pero esto último no sirve para PitS dado que los cambios en requerimientos podrían surgir en cualquier momento, luego de que el cliente vea un entregable y cambie de opinión con respecto a cosas que había pedido previamente.

Por otro lado, los proyectos de PitL se acoplan mejor con UP ya que este provee un marco de planificación detallada muy necesario para proyectos grandes, que permite llevar el control global del proyecto. El esfuerzo y los recursos consumidos en la planificación son muy necesarios para PitL para

evitar problemas futuros: el análisis y mitigación de riesgos es la clave para evitar problemas. En cambio Scrum prioriza actividades según valor de negocio, que en PitL podrían ser muchas y luego de varios meses de desarrollo podrían surgir problemas en módulos que no eran prioritarios para el negocio pero sí riesgosos para el funcionamiento correcto (por ejemplo, cuestiones de seguridad). Además en PitL no se esperan grandes cambios en los requerimientos (al menos no a nivel global, siempre podría haber pequeños cambios en los módulos), con lo cual Scrum sería demasiado flexible en este aspecto.

Por ende en general veremos Scrum aplicado a proyectos pequeños con requerimientos cambiantes y priorización según valor de negocio, y UP a proyectos grandes que requieren planificación a largo plazo y priorización según riesgos para evitar problemas.