

Ingeniería de Software II

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Trabajo Práctico II

Tutor: Javier Martínez Viademonte

Integrante	LU	Correo electrónico
Laporte, Matías	686/09	matiaslaporte@gmail.com
Salegas, Matías	750/01	matias.salegas@gmail.com
Vallejo, Nicolás	500/10	nico_pr08@hotmail.com
Zanitti, Gastón	58/10	gzanitti@gmail.com

Índice

1. Introducción	4
2. Casos de Uso	5
2.1. Desafíos	5
2.1.1. Administración de desafíos	5
2.1.2. Participación en desafíos	5
2.2. Monetización	5
2.2.1. Dinero real	5
2.2.2. Publicidad y marketing	6
2.3. Registro, Autenticación y Datos de usuario	6
2.3.1. Cuentas de usuarios	6
2.3.2. Ranking y puntajes	6
2.4. Social	7
2.5. Simulaciones	7
2.5.1. Ejecución	7
2.5.2. Visualización	7
2.6. Regionalización	7
3. Análisis de riesgos	8
4. Plan de Proyecto	12
4.1. Iteraciones	12
4.1.1. I01 - Primera Iteración (Elaboración)	12
4.1.2. I02 - Segunda iteración (Elaboración)	13
4.1.3. I03 - Tercera Iteración (Elaboración)	13
4.1.4. I04 - Cuarta Iteración (Elaboración)	13
4.1.5. I05 - Quinta Iteración (Construcción)	13
4.1.6. I06 - Sexta Iteración (Construcción)	14
4.2. Primera iteración (I01) en detalle	14
4.3. Diagrama de Gantt	17

5. Atributos de calidad	19
5.1. Disponibilidad	19
5.2. Seguridad	20
5.3. Modificabilidad	22
5.4. Performance	23
5.5. Usabilidad	24
6. Arquitectura	26
6.1. Paneo General	27
6.2. Controlador de Alta de Usuarios	28
6.3. Gestor de Crédito	29
6.4. Gestor de Chats	31
6.5. Controlador de Desafíos	32
6.6. Simulador de Jugadas	34
6.7. Generador Gráfico	36
6.8. Streamer	38
7. Arquitectura TP1	40
7.1. Paneo General	40
7.2. Actualizador de jugadores	41
7.3. Creador y confirmador de desafíos	41
7.4. Creador de equipos	42
7.5. Controlador de registros	42
7.6. Simulador	43
8. Conclusiones	44
8.1. Metodologías utilizadas en la materia	44
8.2. Programming in the large <i>vs</i> Programming in the small	44

1. Introducción

Debido al éxito rotundo de nuestra implementación particular del Curry Game, los inversores originales, más otros nuevos interesados, decidieron extender considerablemente la plataforma en cuanto a funcionalidad (con el objetivo de tener mayor alcance), incluyendo entre otras cosas la posibilidad de abarcar otros juegos aparte del básquet, basarse no sólo en simulaciones sino también en datos de partidos en tiempo real -como una liga de fantasía tradicional-, poder realizar simulaciones gráficas a través de motores 2D/3D, apuestas con dinero real, exhibir publicidad, y un largo y redituable etcétera.

El objetivo de este trabajo práctico consistió en la planificación de las etapas de *Elaboración*, *Construcción* y *Transición* siguiendo la metodología UP, asumiendo la etapa de *Incepción* como concluida, y en la construcción de una arquitectura que resolviera el problema propuesto teniendo en cuenta los diversos detalles extraídos del enunciado y del QAW.

Basados tanto en el enunciado presentado como en el QAW provisto por la cátedra, se definió una lista de casos de uso (**Sección 2 – Casos de Uso**). Ortogonalmente a eso, se realizó un análisis de riesgos (**Sección 3 – Análisis de riesgos**, y en combinación con los casos de uso obtenidos se construyó el Plan de Proyecto (**Sección 4 – Plan de Proyecto**), indicando en el mismo las distintas iteraciones que consideramos necesarias para la concreción del desarrollo solicitado, al igual que los casos de uso que lo componen.

Además, por pedido expreso del enunciado, se hizo énfasis en la descripción de la primera iteración del Plan (**Sección 4.2 – Primera iteración (I01) en detalle**), correspondiente a la etapa de *Elaboración*. Se indican los casos de uso que forman parte de dicha iteración, cómo se descompone cada uno en tareas, la dependencia de las mismas, y la estimación en horas hombre de su concreción.

A continuación, se realizó un diagrama de Gantt que detalla la planificación de la primera iteración, indicando en él, para cada tarea, la asignación de los recursos disponibles (los cuatro integrantes del presente trabajo).

Posteriormente, se realizaron numerosos diagramas que conforman la arquitectura que proponemos para el sistema en cuestión. Además, se comparó esta arquitectura con la diagramada para el primer trabajo práctico de la materia.

2. Casos de Uso

A continuación se presenta una lista de los casos de uso que el grupo identificó a partir del enunciado y del QAW provisto. Intentan ser lo suficientemente exhaustivos para cubrir la funcionalidad requerida del sistema a desarrollar.

En conjunto con el análisis de riesgos presentado en una sección subsiguiente, la lista presentada se utilizó para definir el alcance de las iteraciones del plan de proyecto, así como para determinar con mayor nivel de detalle la primer iteración del plan.

Los casos de uso están agrupados aproximadamente según funcionalidad; dado que algunos a veces abarcan más de una de las funcionalidades con las que se los decidió jerarquizarlos a gran escala, realizar una categorización estricta resultaba complejo.

El número asociado a cada caso de uso se corresponde con la numeración interna que manejó el grupo a la hora de encarar el desarrollo del trabajo. Por cuestiones de tiempo no se los re-enumeró. A fin de cuentas sigue siendo posible seguir los casos de uso de esta manera.

2.1. Desafíos

2.1.1. Administración de desafíos

- (CU3) - **Definiendo reglas de desafío:** un administrador define las reglas de los partidos de cada liga en base al desempeño de los jugadores reales (cantidad de puntos en el desafío, según acción en la realidad).
- (CU4) - **Creando desafío:** un administrador o un participante crea un desafío en el que pueden anotarse una cantidad libre de jugadores.
- (CU13) - **Definiendo premios:** el creador de un desafío (un administrador o un participante) elige cómo distribuir los premios o porcentajes del dinero apostado, según las posiciones finales del mismo.
- (CU26) - **Definiendo desafíos interzonales:** un administrador del sistema define desafíos para que participen los mejores jugadores de cada una de las regiones definidas.
- (CU31) - **Configurando visibilidad de los desafíos:** el administrador configura la visibilidad de los desafíos con el fin de que participantes no calificados para participar puedan visualizarlos.

2.1.2. Participación en desafíos

- (CU1) - **Eligiendo liga para competir:** un participante elige en qué deporte y liga quiere inscribir a un equipo.
- (CU5) - **Aceptando desafío:** un participante se anota para participar en uno de los desafíos ya creados.
- (CU14) - **Consultando estado (cuenta regresiva, participantes, posiciones) del desafío:** un participante puede consultar datos del desafío donde decidió involucrarse: cuánto falta para que empiece, datos de apuestas, quiénes son los otros participantes, sus posiciones, etc.

2.2. Monetización

2.2.1. Dinero real

- (CU8) - **Apostando:** un participante apuesta cierta cantidad de dinero en un desafío.

- (CU10) **Actualizando datos de medios de pago:** el participante asocia algún medio de pago para poder apostar en los desafíos.
- (CU33) - **Consultando estado de cuenta y movimientos de usuario:** un usuario consulta su estado de cuenta y movimiento. Un administrador puede consultar los datos de cualquier participante del sistema.
- (CU34) - **Auditando movimientos de dinero:** una entidad gubernamental de control puede auditar los movimientos de todos los usuarios para analizar el movimiento de dinero en el sistema.

2.2.2. Publicidad y marketing

- (CU16) - **Configurando publicidad en el sitio y simulaciones:** un representante de empresa sponsor del proyecto puede acceder a una interfaz desde donde se configura la publicidad que se muestra en el sitio y las simulaciones a los participantes.
- (CU38) - **Configurando publicidad en transmisiones:** un representante de empresa dueña de los derechos de televisión puede modificar las publicidades que se muestran en las transmisiones de los partidos a los participantes.
- (CU17) - **Acceder a datos de preferencia/comportamiento de usuarios:** un administrador o un representante de empresa accede a estadísticas que permiten obtener insights de negocio, en base al comportamiento y las opciones más utilizadas por los usuarios.

2.3. Registro, Autenticación y Datos de usuario

2.3.1. Cuentas de usuarios

- (CU9) - **Creando cuenta de usuario:** un participante se registra en el sistema para poder participar de los desafíos
- (CU12)- **Consultando cuenta de usuario:** un participante o administrador consulta los datos con los que se dio de alta el usuario en el sistema.
- (CU41) - **Desactivando cuenta:** un participante o representante de una organización de protección al consumidor puede desactivar una cuenta temporalmente para ayudar a adictos al juego a no tener recaídas.
- (CU42) - **Reactivando cuenta:** un participante puede reactivar su cuenta luego del tiempo establecido en la desactivación.
- (CU28) - **Consultando dashboard regional o global:** un administrador o representante de empresa sponsor del proyecto puede acceder a un dashboard con el estado de cuenta en tiempo real del sitio para cada una de las regiones y niveles y de cualquier grupo de participantes.

2.3.2. Ranking y puntajes

- (CU29) - **Consultando ranking de jugadores:** un participante puede consultar el ranking de jugadores en cualquier momento y ver su posición en el mismo.
- (CU32) - **Configurando reinicio del ranking de jugadores:** un administrador configura cuándo se realiza el reinicio automático del ranking del sitio (p.ej., cuando comienza una nueva temporada o año).

2.4. Social

- (CU6) - **Participando del chat general**: un participante envía un mensaje al chat general para que lo vean todos los demás participantes del desafío.
- (CU7) - **Participando de chat privado**: un participante envía un mensaje privado a otro participante y solo éste puede verlo.
- (CU23)- **Recolectando opiniones de redes sociales y chats**: se recolectan opiniones de las redes sociales y los chats generales y privados del sistema para afectar los resultados de las simulaciones.
- (CU24)- **Definiendo impacto de opiniones**: un administrador define de qué forma los comentarios obtenidos en las redes sociales impactan en el resultado de los desafíos o en la performance de los jugadores.

2.5. Simulaciones

2.5.1. Ejecución

- (CU18) - **Definiendo reglas de simulación**: un administrador modifica las reglas de las simulaciones de los deportes basándose en los comentarios recibidos por el comité de expertos de cada uno de ellos.
- (CU2) - **Obteniendo datos en tiempo real**: se obtienen en tiempo real, mediante empresas proveedoras, datos de los jugadores y del desarrollo de los partidos de las APIs provistas para mantenerlos actualizados.

2.5.2. Visualización

- (CU19) - **Mostrando detalle minuto a minuto de la simulación**: los participantes pueden ver un detalle minuto a minuto de las simulaciones.
- (CU21) - **Mostrando simulación gráfica (motor 2d, motor 3d)**: se genera una simulación gráfica que se muestra al usuario para que pueda disfrutar del desafío de otra manera.
- (CU22) - **Observando la transmisión de un partido de liga**: los participantes pueden ver la transmisión de los partidos en los desafíos en modo *liga de fantasía tradicional*.
- (CU35) - **Auditando simulaciones**: una entidad gubernamental de control puede auditar las simulaciones para corroborar que se correspondan con los resultados obtenidos y los desafíos pagados.

2.6. Regionalización

- (CU25) - **Definiendo regiones de la plataforma**: un administrador de la plataforma define regiones y niveles para los mismos, con el fin de regionalizar la plataforma y facilitar la comunicación entre los participantes, la integración con los bancos, el cumplimiento de legislación vigente en los distintos países, etc.
- (CU36) - **Definiendo restricciones por zona**: un administrador define qué deportes están disponibles para cada una de las zonas y si es posible apostar con dinero real, etc. Involucra también definir si el sitio es accesible en cada país.

3. Análisis de riesgos

Riesgo 1

Descripción: El experto en redes de la empresa proveedora de infraestructura exige de manera prioritaria para el próximo mes el diagrama híbrido c&c/allocation;deployment que detalla el manejo del streaming de partidos/simulaciones a todas las regiones en el caso de un evento global o continental. Dado que los miembros del equipo carecen del conocimiento necesario en el tema, es posible que no se llegue a cumplir el plazo.

Probabilidad: Alta

Impacto: Alto

Exposición: Alta

Mitigación: Priorizar el desarrollo del mecanismo de streaming, introduciéndolo en la primera iteración.

Plan de Contingencia: Contratar los servicios de una empresa especializada en el tema que se encargue del desarrollo.

Riesgo 2

Descripción: Es importante que el sistema esté disponible la mayor cantidad del tiempo posible. Hay que tener especial cuidado ya que se rumorea que los servidores de la empresa proveedora de servicios se caen sin previo aviso, y la caída del sistema implicaría pérdida de dinero.

Probabilidad: Media

Impacto: Alto

Exposición: Alta

Mitigación: Definir un mecanismo de redundancia de servidores. Cualquier acción como la carga de datos por parte del usuario podría hacerse de forma offline y sincronizarse con el sistema una vez resuelto el problema de conexión.

Plan de Contingencia: Contratar otra empresa proveedora de servicios, ya sea como respaldo o reemplazando a la actual como proveedora principal.

Riesgo 3

Descripción: La transmisión de la partidos debe hacerse con la mejor calidad posible y sin cortes ya que no hacerlo impediría cumplir con el estándar de calidad que exigen las ligas para mantener los derechos de transmisión.

Probabilidad: Media

Impacto: Alto

Exposición: Alta

Mitigación: Contratar un producto que permita ajustar el bitrate del streaming de video, depen-

diendo de las características de la conexión de los usuarios.

Plan de Contingencia: Tercerizar el streaming a través de una plataforma con harta experiencia en el tema, como YouTube o TwitchTV.

Riesgo 4

Descripción: Es importante proveer el acceso correspondiente tanto a la plataforma en su totalidad como a sus diferentes secciones de acuerdo a la legislación vigente de cada país. No hacerlo podría incurrir en la comisión de delitos.

Probabilidad: Media

Impacto: Alto

Exposición: Alta

Mitigación: Diseñar e implementar un mecanismo lo suficientemente elástico para configurar el acceso a las distintas modalidades de la plataforma. Mantener un equipo que periódicamente se informa acerca de las distintas legislaciones de cada país.

Plan de Contingencia: En caso de poseer dudas con respecto a la legalidad del funcionamiento de la plataforma en un país, impedir el ingreso al sitio hasta tanto se haya validado y consultado sobre las normativas, evitando así potenciales actos ilegales.

Riesgo 5

Descripción: Es menester mantener la transparencia de los módulos de simulaciones, a modo de evitar la interferencia de agentes externos en su ejecución. Un ataque al sistema podría involucrar la alteración del módulo de simulación, provocando la existencia de resultados adulterados.

Probabilidad: Media

Impacto: Alto

Exposición: Alto

Mitigación: Desarrollar mecanismos de testeo de la simulación en servidores testigo, utilizando como input un conjunto de datos para los cuales ya se conozca el resultado y contrastar. Diseñar un mecanismo que ante inconsistencias realice un deploy de una versión válida del módulo simulador.

Plan de Contingencia: Cuando se detecte alguna inconsistencia en el módulo simulación de algún servidor, inhabilitarlo y redirigir las peticiones a otro, hasta tanto se verifique el correcto funcionamiento del módulo correspondiente.

Riesgo 6

Descripción: Los stakeholders quieren tener confianza en que vamos a ser capaces de guardar los datos de tarjetas de crédito, como hacen otros sitios exitosos. Si no nos mostramos seguros, pueden darle (parte de) el proyecto a otro equipo de desarrollo.

Probabilidad: Baja.

Impacto: Alto.

Exposición: Media.

Mitigación: Mostrar ejemplos de desarrollos anteriores donde no existieron esos problemas.

Plan de contingencia: Realizar una certificación de normas de seguridad internacionales con el fin de asegurar que el sistema es seguro.

Riesgo 9

Descripción: Los distintos sponsors tienen intereses distintos, algunos de ellos contrapuestos. Por ejemplo, las empresas televisivas están en contra de las simulaciones. Puede ser que parte del proyecto cambie o se cancele.

Probabilidad: Media.

Impacto: Alta.

Exposición: Alta.

Mitigación: Mantener reuniones periódicas con los sponsors para convencerlos de que ambas funcionalidades son buenas para el proyecto.

Plan de contingencia: Diseñar una arquitectura flexible, que se adapte a los cambios.

Riesgo 10

Descripción: Hay otros aspectos todavía no definidos y que no dependen de nosotros. Por ejemplo, en qué países se va a usar cada parte del juego. Puede pasar que cambien los requerimientos después de que empecemos.

Probabilidad: Media.

Impacto: Alto.

Exposición: Alto.

Mitigación: Mantener comunicación fluida con los stakeholders para estar al tanto de sus decisiones y lograr que las comuniquen lo antes posible.

Plan de contingencia: Diseñar una arquitectura flexible, que se adapte a los cambios.

Riesgo 11

Descripción: El equipo nunca trabajó con esta metodología. Puede ocurrir que la adaptación lleve más tiempo de lo planeado.

Probabilidad: Alta.

Impacto: Medio.

Exposición: Alta.

Mitigación: Consultas periódicas con un experto.

Plan de contingencia: Aumentar las consultas al experto.

Riesgo 12

Descripción: El equipo tuvo problemas de comunicación antes. Si se repiten, pueden afectar todo el trabajo. Trabajamos mucho de manera remota y tenemos distintos horarios.

Probabilidad: Alta.

Impacto: Alto.

Exposición: Alta.

Mitigación: Reuniones periódicas (semanales) e intercambio fluido de mails.

Plan de contingencia: Reuniones de emergencia (en caso de que algún miembro se encuentre imposibilitado para seguir con sus tareas asignadas) y buscar vías de comunicación instantáneas, como chats y teléfono.

Riesgo 13

Descripción: Es probable que la planificación no sea exacta, tanto en las tareas como en los tiempos pensados para cada una.

Probabilidad: Alta.

Impacto: Medio.

Exposición: Alta.

Mitigación: Sobreestimar los tiempos y las dificultades en un 25 % de horas hombre extra.

Plan de contingencia: Evaluar y adaptar la planificación luego de la primera iteración.

4. Plan de Proyecto

4.1. Iteraciones

Las iteraciones tienen una duración aproximada de 3 semanas (15 días hábiles). Teniendo en cuenta que los recursos asignados son 4 arquitectos de software/programadores trabajando de modo part-time (6 horas), se estiman unas 360hrs. aproximadas por iteración.

Nuestra decisión sobre la conformación de las iteraciones -en cuanto a los casos de uso- se rigió por varios factores:

- Se tuvieron en cuenta las necesidades extraídas del QAW con los stakeholders
- Se trató de agrupar a los casos de uso por su temática, prefiriendo en caso de ser posible juntar dentro de una misma iteración los casos de uso que se relacionan con funcionalidades similares.
- Otro factor que se tuvo en cuenta a la hora de definir el orden de las iteraciones, fueron los riesgos detectados y analizados en **Sección 3 – Análisis de riesgos**.
- También se consideró la prioridad de la funcionalidades referidas en los casos de uso, intentando en lo posible desarrollar antes las más importantes para el negocio.

La primera iteración quedó expresamente definida por el QAW y el análisis de riesgo, factores a los que se les dio mayor prioridad. Del análisis de riesgo, y dado el factor prioritario que tiene la disponibilidad del sistema, se extrajo que la parte más conflictiva ocurre a la hora de transmitir los desafíos (especialmente para los eventos globales), así como al mostrar correctamente las simulaciones.

El siguiente punto en cuanto a nivel de riesgo, fue lo concerniente a la seguridad, integridad y transparencia de la transmisión de los datos de pago y movimientos de dinero, y las transmisiones de datos usadas para definir los resultados de los partidos de la realidad y minuto a minuto de las simulaciones. Decidimos encarar esto en la segunda iteración del proyecto.

Otro foco de preocupación y de vital importancia para el proyecto, es su alcance y monetización. Por esa razón, se decidió trabajar en lo que respecta a las publicidades del sitio, simulaciones y transmisiones (y el uso de los datos de comportamiento de usuarios), además de la regionalización del sistema (con todas las reglas y restricciones que eso conlleva) en la tercer iteración.

El manejo administrativo de las cuentas de los usuarios, y el aspecto *social* (chat, utilización de las opiniones vertidas en redes sociales, etc.), consideradas cuestiones de menor complejidad, se decidieron enfrentar en la cuarta iteración.

Decidimos dedicarnos en la quinta iteración a todo lo referido a los desafíos, su creación, configuración, definición de reglas y de los premios. En la sexta iteración, nos encargaremos de lo que concierne a los rankings de los jugadores globales, regionales, etc. Como los casos de uso de estas dos iteraciones tienen un menor riesgo y complejidad que los anteriores, consideramos que estas dos fases son de construcción, ya que se trabajará bastante en lo implementativo y poco en el análisis/diseño.

4.1.1. I01 - Primera Iteración (Elaboración)

- (CU19) Mostrando detalle minuto a minuto de la simulación
- (CU2) Obteniendo datos en tiempo real
- (CU21) Mostrando simulación gráfica
- (CU22) Observando la transmisión de un partido de liga

4.1.2. I02 - Segunda iteración (Elaboración)

- (CU9) Creando cuenta de usuario
- (CU10) Actualizando datos de medios de pago
- (CU34) Auditando movimientos de dinero
- (CU33) Consultando estado de cuenta y movimientos de usuario
- (CU35) Auditando simulaciones
- (CU8) Apostando

4.1.3. I03 - Tercera Iteración (Elaboración)

- (CU36) Definiendo restricciones por zona
- (CU25) Definiendo regiones de la plataforma
- (CU18) Definiendo reglas de simulación
- (CU38) Configurando publicidad y ads en transmisiones
- (CU16) Configurando publicidad en el sitio y simulaciones
- (CU17) Acceder a datos de preferencia/comportamiento de usuarios

4.1.4. I04 - Cuarta Iteración (Elaboración)

- (CU6) Participando del chat general
- (CU7) Participando de chat privado
- (CU12) Consultando cuenta de usuario
- (CU41) Desactivando cuenta
- (CU42) Reactivando cuenta
- (CU23) Recolectando opiniones de redes sociales y chats
- (CU24) Definiendo impacto de opiniones

4.1.5. I05 - Quinta Iteración (Construcción)

- (CU1) Eligiendo liga para competir
- (CU3) Definiendo reglas de desafío
- (CU4) Creando desafío
- (CU5) Aceptando desafío
- (CU14) Consultando estado (cuenta regresiva, participantes, posiciones) del desafío
- (CU13) Definiendo premios

4.1.6. I06 - Sexta Iteración (Construcción)

- (CU28) Consultando dashboard regional o global
- (CU29) Consultando ranking de jugadores
- (CU32) Reiniciando el ranking de jugadores
- (CU26) Definiendo desafíos interzonales
- (CU31) Configurando visibilidad de los desafíos

4.2. Primera iteración (I01) en detalle

En esta sección se realiza un detalle de las tareas que se consideran necesarias realizar en la primer iteración, junto a las horas hombres esperadas. Dado que ya se realizó un reconocimiento, priorización, y estimación de tiempo de los casos de uso, además del análisis de riesgo, no serán tareas prioritarias ni de mucha intensidad en esta etapa, sino que se utilizarán para poder corregir detalles de las futuras iteraciones.

Algunas de las tareas (por ejemplo I01T01, I01T02, I01T03) que no se corresponden directamente con un caso de uso, tienen como objetivo la mitigación de los riesgos analizados (**Sección 3 – Análisis de riesgos**).

- I01T01 - Reunión semanal breve con stakeholders para asegurarse que los requerimientos estén actualizados: **10hs.**
 - I01T01ST01 - Reunión la primer semana: 4hs.
 - I01T01ST02 - Reunión la segunda semana: 3hs.
 - I01T01ST03 - Reunión la tercer semana: 3hs.
- I01T02 - Reunión cada 3 días hábiles del equipo para evitar problemas de comunicación como en el pasado: **10hs.**
 - I01T02ST01 - Primera reunión: 2hs.
 - I01T02ST02 - Segunda reunión: 2hs.
 - I01T02ST03 - Tercer reunión: 2hs.
 - I01T02ST04 - Cuarta reunión: 2hs.
 - I01T02ST05 - Quinta reunión: 2hs.
- I01T03 - Reunión semanal de consulta con experto en metodología UP: **8hs.**
 - I01T03ST01 - Reunión la primer semana: 3hs.
 - I01T03ST02 - Reunión la segunda semana: 2hs.
 - I01T03ST03 - Reunión la tercer semana: 2hs.
- I01T04 - Identificación y descripción de los atributos de calidad del sistema: **40hs.**
 - I01T04ST01 - Cotejamiento de los atributos definidos por stakeholders en QAW y relación con los casos de uso definidos: 16hrs.
 - I01T04ST02 - Descripción de escenarios de atributos de calidad: 20hs.
 - I01T04ST03 - Verificación de la documentación escrita: 4hs.

- I01T05 - Diseño de la arquitectura del sistema: **60hs.**
 - I01T05ST01 - Analizar los escenarios descritos en I01T04 e identificar drivers de arquitectura: 8hs.
 - I01T05ST02 - Estudiar y elegir patrones arquitectónicos que satisfagan los drivers identificados: 12hs.
 - I01T05ST03 - Verificar y refinar los casos de usos y los escenarios: 10hs.
 - I01T05ST04 - Iterar: 30hs.

- I01T06 - Realización de las tareas del (CU2) Obteniendo datos en tiempo real: **40hs.**
 - I01T06ST01 - Reunirse con empresas que provean datos en tiempo real de la evolución de los partidos y contratar alguna: 12hs.
 - I01T06ST02 - Reunirse con la empresa contratada y obtener documentación técnica sobre la API que proveen: 2hs.
 - I01T06ST03 - Investigar la API y analizar qué datos podemos usar como input en nuestro sistema: 6hs.
 - I01T06ST04 - Adaptar nuestro sistema para que los desafíos en modo *Liga de fantasía* utilicen los datos obtenidos a través de la api: 10hs.
 - I01T06ST05 - Realizar algunos desafíos y verificar que los resultados de los puntajes se condigan con lo que ocurrió en los partidos reales: 6hs.
 - I01T06ST06 - Corregir potenciales errores: 4hs.

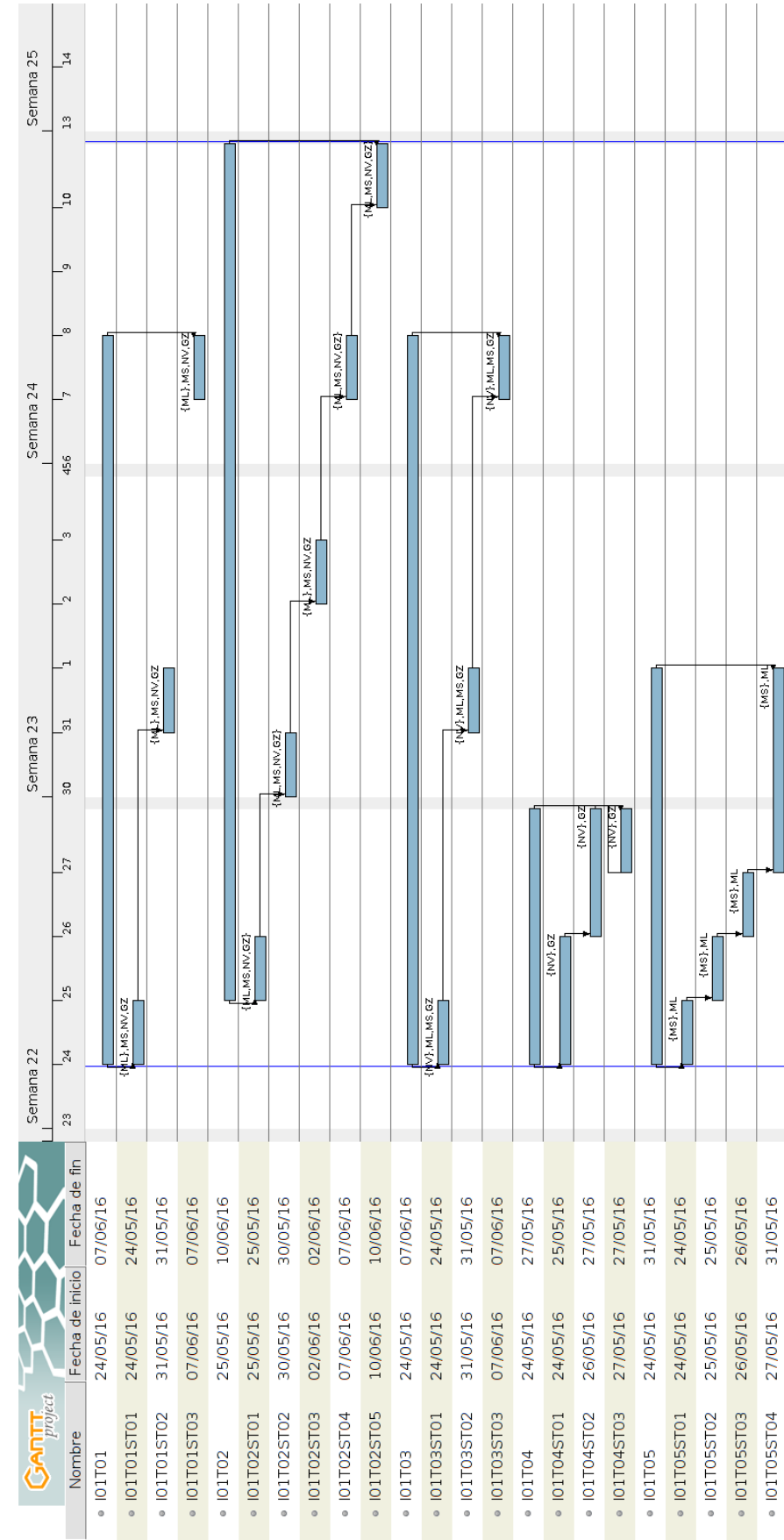
- I01T07 - Realización de las tareas del (CU19) Mostrando detalle minuto a minuto de la simulación: **40hs.**
 - I01T07ST01 - Investigar el log del desarrollo anterior, cómo se genera y la calidad de la salida: 4hs.
 - I01T07ST02 - Reunirse con las empresas desarrolladoras de los dos motores gráficos y averiguar qué tipo de entrada necesitan: 6hs.
 - I01T07ST03 - Comparar el log actual con el que se necesita y definir qué es lo que falta agregar: 4hs.
 - I01T07ST04 - Agregar el detalle necesario al log de salida para cumplir con lo requerido por las empresas: 10hs.
 - I01T07ST05 - Ajustar la velocidad de salida del log para que no sea instantánea, sino minuto a minuto, y respete la nueva duración (similar a la de un partido real) de las simulaciones: 5hs.
 - I01T07ST06 - Realizar algunas ejecuciones de prueba y obtener logs de salida de ejemplo: 4hs.
 - I01T07ST07 - Corroborar con empresas proveedoras que el detalle del log obtenido sea el correcto: 2hs.
 - I01T07ST08 - Corregir potenciales errores: 5hs.

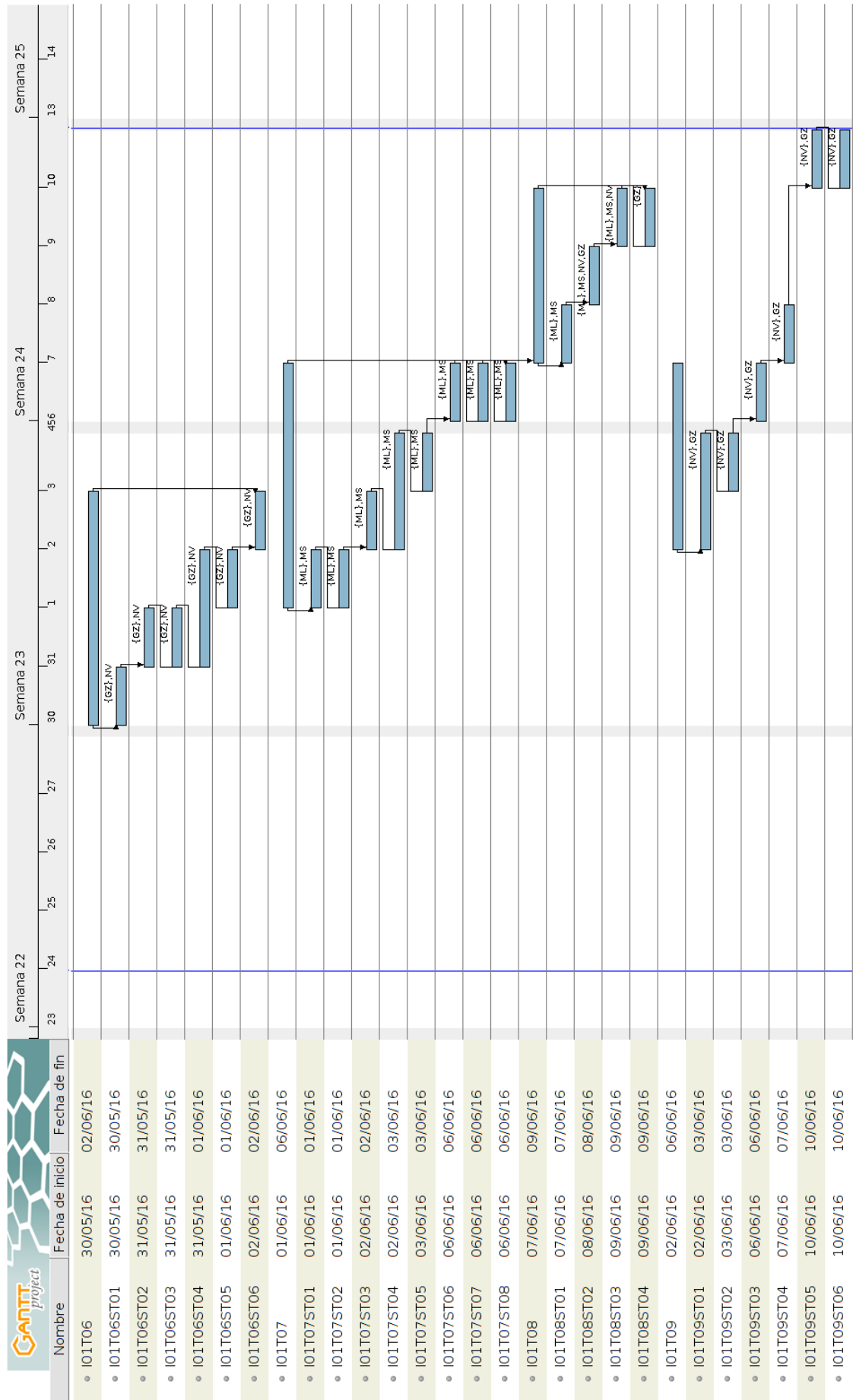
- I01T08 - Realización de las tareas del (CU21) Mostrando simulación gráfica: **50hs.**
 - I01T08ST01 - Reunirse con empresa de motor 3D y discutir sobre los requerimientos técnicos para la transmisión de video en dispositivos no soportados por el motor gráfico: 10hs.
 - I01T08ST02 - Obtener varios logs minuto a minuto de prueba y utilizarlos como entrada para los distintos motores gráficos. Comparar que el resultado gráfico (según limitaciones de cada motor) de los partidos sea el mismo y no haya diferencias: 20hs.

- I01T08ST03 - Probar los motores en distintos dispositivos y obtener los requerimientos mínimos y recomendados de hardware para poder informar a los participantes: 15hs.
 - I01T08ST04 - Escribir documentación sobre los distintos motores, obtener screenshots para poder publicar en el home del producto como ejemplo de jugabilidad: 5hs.
- I01T09 - Realización de las tareas del (CU22) Observando la transmisión de un partido de liga: **80hs.**
- I01T09ST01 - Reunirse con la empresa dueña de los derechos de televisión y la empresa proveedora de infraestructura para definir sus requerimientos, necesidades, y llegar a acuerdos comunes en los puntos álgidos: 15hs.
 - I01T09ST02 - Reunirse con la empresa proveedora de infraestructura de redes para definir la arquitectura de hardware a utilizar para el sistema: 10hs.
 - I01T09ST03 - Implementar la solución de hardware convenida con la empresa proveedora de infraestructura de redes: 20hs.
 - I01T09ST04 - Realizar pruebas internas de transmisión en vivo de eventos con diferentes cargas en los servidores y desde distintas regiones: 15hs.
 - I01T09ST05 - Dar feedback a la empresa proveedora de infraestructura y ver, de ser necesario, cómo mejorar el rendimiento: 10hs.
 - I01T09ST06 - Mostrarle a la empresa dueña de los derechos de televisión el funcionamiento del sistema y verificar que cumpla con sus requerimientos: 10hs.

La suma de las tareas para la primer iteración da 338hs. Habiendo calculado una iteración ideal de 360hs. nos da cierto margen como para poder afrontar tareas urgentes o no previstas.

4.3. Diagrama de Gantt





5. Atributos de calidad

En esta sección se presentan los atributos de calidad identificados en el TP (ordenados según la prioridad definida por los *stakeholders* en el *Quality Attribute Workshop*), junto a sus respectivos escenarios.

5.1. Disponibilidad

Motivación

- El sistema tiene que estar andando todo el tiempo.
- Cantidad de usuarios limitada - máxima por servidor
- Múltiples nodos distribuidos de pocos usuarios cada uno.
- Los partidos streameados deben darse en excelente calidad sin cortes.
- Cuando ninguno de los motores gráficos se encuentra disponible, debe usarse el streaming del engine 3D.
- Debe esforzarse por respetar a los países en los que la legislación vigente no permite directamente que se ingrese al sitio.
- Debe ser fácil desactivar una cuenta por un tiempo, sin poder reactivarla, tanto para usuarios adictos en recuperación como para usuarios de países en los que no está disponible el sitio.

Escenarios

Descripción	En caso de saturación de un nodo, deben direccionarse nuevas conexiones a otro nodo.
Fuente	Usuario
Estímulo	Se realiza un request al sistema.
Entorno	Nodo saturado
Artefacto	Nodo del sistema
Respuesta	Se selecciona un nuevo nodo regional y se continúa el servicio mediante una conexión con él.
Medición	El 99.99% de las nuevas conexiones se realizan sin inconveniente alguno y de forma transparente al usuario.

Descripción	Los streamings de partidos se realizan con la menor cantidad de cortes posibles.
Fuente	Usuario
Estímulo	Solicita ver el streaming de un partido
Entorno	Normal
Artefacto	Generador gráfico
Respuesta	Se realizan los cálculos pertinentes para determinar el ancho de banda del usuario y se comienza el streaming con la calidad determinada.
Medición	Se produce a lo sumo un corte con duración menor a 5 segundos por transmisión.

Descripción	Si los motores gráficos para una simulación no están disponibles, debe realizarse un streaming del engine 3D para la simulación.
Fuente	Usuario
Estímulo	Intenta observar una simulación
Entorno	Motores gráficos del dispositivo no disponibles.
Artefacto	Generador gráfico
Respuesta	Ante la detección de la imposibilidad de utilizar los motores gráficos en el dispositivo, se activa el streaming del engine 3D de la simulación.
Medición	Se transmite la simulación 3D sin inconvenientes en el 99 % de los casos.

Descripción	Si se detecta que los motores gráficos vuelven a estar disponibles, se los vuelve a utilizar en lugar del streaming de la simulación
Fuente	Identificador de dispositivo
Estímulo	Detecta motores gráficos disponibles
Entorno	Streaming de una simulación
Artefacto	Generador gráfico
Respuesta	Se utiliza alguno de los motores gráficos del dispositivo
Medición	Se corta el streaming de video y en menos de 10 segundos se continúa la simulación con el motor gráfico.

Descripción	Si un usuario identificado como adicto en recuperación intenta conectarse, se le niega el acceso.
Fuente	Usuario identificado como adicto
Estímulo	Intento de conexión
Entorno	Normal
Artefacto	Controlador de Alta de Usuarios
Respuesta	Se le niega el acceso al sistema.
Medición	En el 99.9999 % de los casos de intento de conexión proveniente de un usuario identificado como adicto en recuperación, se le niega el acceso al sistema con un cartel informativo.

Descripción	Se bloquea el acceso desde países que prohíben el uso del sistema.
Fuente	Usuario de país que prohíbe el uso del sistema
Estímulo	Intento de conexión
Entorno	Normal
Artefacto	Controlador de Alta de Usuarios
Respuesta	Se le niega el acceso al sistema.
Medición	En el 99.9999 % de los casos de intento de conexión proveniente de un país que no permite el acceso al sitio, se le niega el acceso y se le muestra un cartel informativo.

5.2. Seguridad

Motivación

- Todo lo relativo al manejo del dinero (depósito y retiro de los participantes vía tarjeta de crédito o caja de ahorro) deber ser seguro y transparente.
- Se teme por la seguridad de los datos de los usuarios, tanto por el robo de la información de las tarjetas de crédito o cajas de ahorro como de la explotación de los datos de los usuarios por terceros para fines no autorizados.

- Preocupa el resultado de los desafíos, por el pago/cobro a los participantes, y la coherencia con los datos provenientes de las empresas que relevan los resultados de los partidos.
- Es importante proveer transparencia acerca del funcionamiento de las simulaciones.
- Los módulos de las simulaciones deben poder ser inspeccionados fácilmente por entidades de control.
- Se debe loggear todo movimiento de dinero para evitar cualquier tipo de evasión impositiva.

Escenarios

Descripción	La información de los medios de pago de los usuarios está protegida contra el robo de datos
Fuente	Atacante externo
Estímulo	Intenta vulnerar la información de medios de pago de los usuarios mediante el descifrado de los datos.
Entorno	Normal
Artefacto	Datos del Almacén de información de crédito
Respuesta	Los datos en texto plano no son accesibles al atacante en tiempos razonables.
Medición	En el 99.9999 % de los casos, los ataques no son exitosos.

Descripción	Los movimientos de dinero se registran en el sistema
Fuente	Usuario o Gestor de crédito
Estímulo	Se apuesta dinero o se recibe un pago.
Entorno	Normal
Artefacto	Gestor de crédito
Respuesta	Se realiza correctamente el movimiento y se loggea un registro con los datos de la transacción
Medición	El registro se guarda correctamente en el 99.999 % de los casos.

Descripción	Un usuario no puede consultar los datos de pago que no le corresponden.
Fuente	Usuario
Estímulo	Intenta obtener los datos de pago de otro usuario.
Entorno	Normal
Artefacto	Gestor de crédito
Respuesta	Se rechaza el intento y se loggea un registro de la solicitud.
Medición	La solicitud se rechaza un 99.99 % de las veces.

Descripción	Los resultados de los desafíos en modo Liga de Fantasía, se corresponden con los datos provistos por las empresas relevadoras de datos de los partidos
Fuente	Empresa relevadora de datos de partidos
Estímulo	Se envían resultados minuto a minuto de un partido
Entorno	Normal
Artefacto	Controlador de desafíos
Respuesta	Se autentica la identidad de la empresa relevadora de datos y se persisten los resultados en el sistema.
Medición	En más del 99 % de los casos los resultados de los desafíos se contrastan correctamente con los datos provistos por las empresas.

Descripción	Se rechazan los datos de los partidos de fuentes no autenticadas correctamente
Fuente	Atacante externo
Estímulo	Se hace pasar por una de las empresas contratadas para proveer los datos y de esta forma alterar los resultados de los desafíos
Entorno	Normal
Artefacto	Controlador de desafíos
Respuesta	Se detecta la intrusión al no poder autenticar la identidad de la fuente de datos, se rechazan los datos y se registra el potencial ataque para futuros análisis
Medición	En más del 99.99 % de los casos el ataque es correctamente detectado y repelido.

Descripción	Un usuario puede acceder a sus datos de medios de pago
Fuente	Usuario
Estímulo	Intenta consultar o modificar sus datos de medios de pago
Entorno	Normal
Artefacto	Controlador de Alta de Medios de pago
Respuesta	Se autoriza el acceso.
Medición	La solicitud se autoriza 99.9999 % de las veces.

Descripción	El funcionamiento de los módulos de simulación es el esperado
Fuente	Auditor
Estímulo	Prueba los módulos en servidores testigo
Entorno	Entorno de prueba
Artefacto	Calculador de jugadas
Respuesta	Se ejecutan los casos de prueba de los sistemas testigos.
Medición	Se superan todos los casos de prueba de los sistemas testigos.

Descripción	El software instalado en los servidores del sistema se corresponde con el realizado por el equipo de desarrollo.
Fuente	Auditor y equipo de desarrollo
Estímulo	Se ejecutan mecanismos de hashing para comprar los componentes.
Entorno	Entorno de prueba
Artefacto	Sistema
Respuesta	Se obtiene el hash del software instalado en los servidores.
Medición	En todos los casos el hash obtenido es igual al hash esperado por el equipo de desarrollo.

5.3. Modificabilidad

Motivación

- Se deben poder agregar fácilmente nuevos deportes
- Debe resultar fácil modificar las simulaciones con el fin de mejorarlas.
- Los datos del sitio deben ser fácilmente minados, y además debe poder crearse fácilmente reportes a partir de ellos.
- Se deben poder modificar fácilmente las publicidades, tanto de las simulaciones como del sistema.

Escenarios

Descripción	Es fácil agregar un nuevo deporte al sistema.
Fuente	Desarrollador
Estímulo	Introduce un nuevo deporte al sistema.
Entorno	Tiempo de desarrollo y diseño
Artefacto	Sistema
Respuesta	Los cambios se agregan sin ninguna complicación y se pueden realizar simulaciones del nuevo deporte.
Medición	En menos de 40 horas hombre debe poder realizarse la integración del nuevo deporte al sistema sin afectar negativamente a las funcionalidades previamente provistas. El grueso del costo temporal se dedica al modelado y relevado del deporte en cuestión y no a la integración en el sistema.

Descripción	Se pueden modificar fácilmente las simulaciones para mejorarlas.
Fuente	Desarrollador
Estímulo	Introduce cambios en el módulo de simulación.
Entorno	Tiempo de desarrollo y diseño
Artefacto	Sistema
Respuesta	Los cambios se agregan sin ninguna complicación, y la nueva versión del simulador inmediatamente se pone en marcha.
Medición	En menos de 50 horas hombre se pueden agregar cambios a los módulos de simulación.

Descripción	Se pueden agregar nuevos reportes de manera fácil y rápida.
Fuente	Desarrollador
Estímulo	Introduce un nuevo reporte.
Entorno	Tiempo de desarrollo y diseño
Artefacto	Sistema
Respuesta	Se realiza el reporte y se lo agrega al sistema sin ningún inconveniente.
Medición	En menos de 30 horas hombre se puede generar un nuevo reporte con los datos persistidos en el sistema.

5.4. Performance

Motivación

- Se requiere que las transmisiones de los partidos se realicen con excelente calidad y sin cortes.
- Las simulaciones deben realizarse en tiempo real minuto a minuto.
- Los eventos globales deben funcionar correctamente.

Escenarios

Descripción	En lo posible las simulaciones se grafican en los dispositivos del usuario y no en el sistema.
Fuente	Usuario
Estímulo	Visualiza una simulación.
Entorno	Normal
Artefacto	Graficador de simulaciones
Respuesta	Se envían solamente los datos necesarios para que el engine opere en el dispositivo del cliente.
Medición	En 9 de cada 10 dispositivos se grafica la simulación en alguno de los dos motores gráficos en vez de streamear el engine 3D.

Descripción	Se puede consumir la transmisión de un partido de manera rápida y con buena calidad.
Fuente	Usuario
Estímulo	Visualiza un partido y se detecta transmisión no óptima.
Entorno	Ancho de banda del usuario limitado
Artefacto	Streameador de videos
Respuesta	Se ejecuta los calculos pertinentes para acomodar el bitrate del video de manera tal que se ajuste acordemente al ancho de banda disponible para el usuario
Medición	En el 99 % de los casos, el usuario puede visualizar un partido de manera fluida esperando. El tiempo de espera para el comienzo del streaming de video es menor a 20 segundos.

Descripción	Los eventos globales se visualizan correctamente
Fuente	Usuario
Estímulo	Visualización de un evento global
Entorno	Sistema con altos índices de demanda para la visualización del evento.
Artefacto	Streameador de videos
Respuesta	De acuerdo a los anchos de banda de los usuarios se transmite el evento de la mejor manera posible.
Medición	Más del 99 % de los usuarios pueden visualizar el evento correctamente sin cortes y con tiempos de espera menores al minuto para comenzar la visualización.

5.5. Usabilidad

Motivación

- Debe resultar simple y rápido crear equipos y participar de desafíos
- Los administradores deben poder acceder a un dashboard en tiempo real de estado de cuenta regional del sitio y de los grupos de usuarios y analizarlo fácilmente

Escenarios

Descripción	Es fácil y rápido crear un equipo.
Fuente	Usuario
Estímulo	Intenta crear un nuevo equipo.
Entorno	Normal
Artefacto	Interfaz de Usuario
Respuesta	Se crea correctamente un equipo.
Medición	En un tiempo promedio de 20 minutos el usuario puede crear un equipo a su parecer.

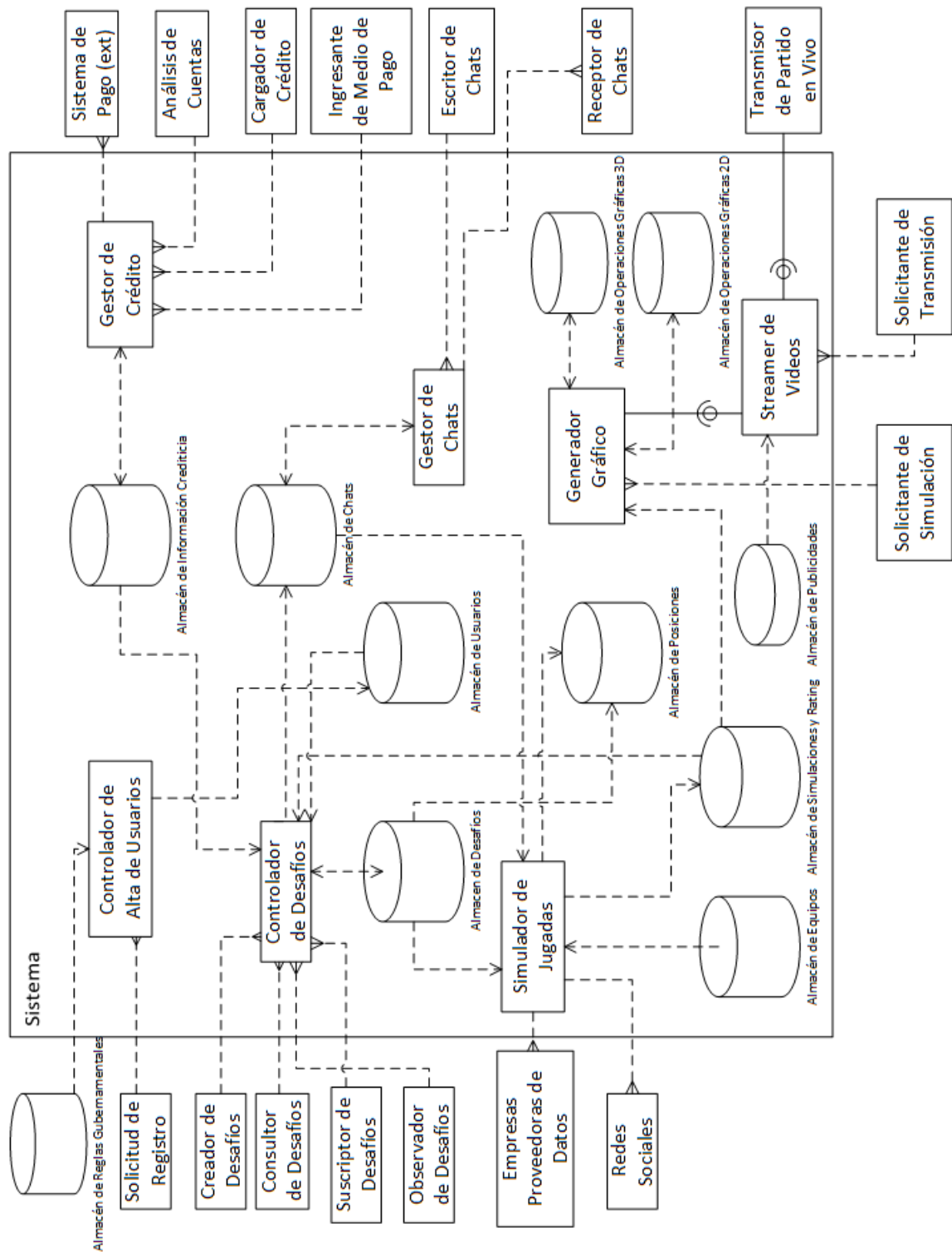
Descripción	Un administrador puede acceder al dashboard de estado de cuenta y de los grupos de participantes de manera rápida y simple.
Fuente	Administrador
Estímulo	Accede al dashboard
Entorno	Normal
Artefacto	Interfaz de Administrador
Respuesta	Se muestran el dashboard y sus opciones
Medición	En menos de de un minuto el administrador puede consultar la información que requiere del dashboard.

6. Arquitectura

En esta sección mostraremos los diagramas (**vista C&C**) desarrollados para la arquitectura del sistema. Por razones de claridad y espacio, se omitió en el diagrama el detalle de los puertos y roles en los componentes y conectores.

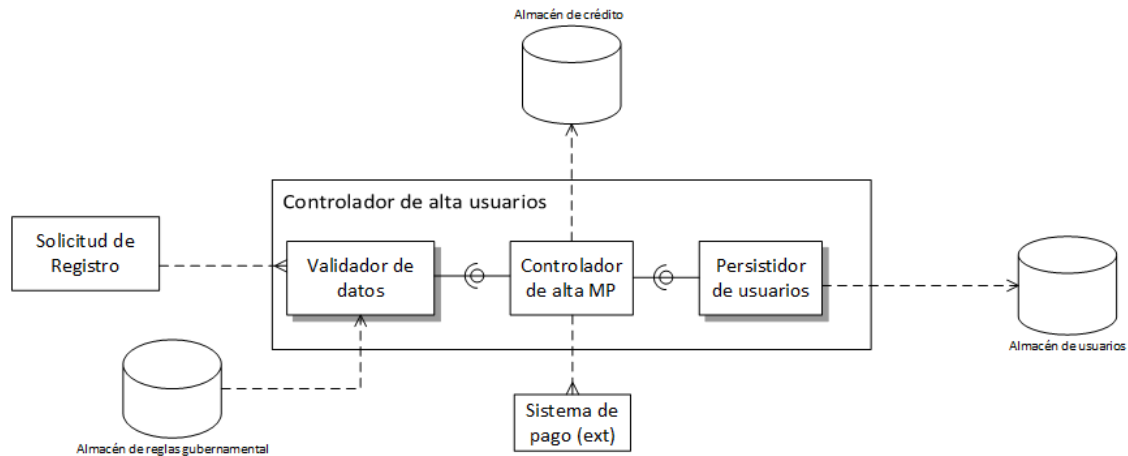
Consideramos nuestra arquitectura con un estilo predominantemente Cliente-Servidor, donde el Servidor es nuestro sistema, distribuido en distintas máquinas (*nodos*) a través del mundo, mientras que el Cliente es el navegador web que se encuentra en el dispositivo que utiliza el usuario.

6.1. Paneo General



En la figura 6.1 se ven los principales componentes del sistema, sobre los cuales se hace zoom en las secciones posteriores, además de los repositorios utilizados para nuestra solución.

6.2. Controlador de Alta de Usuarios



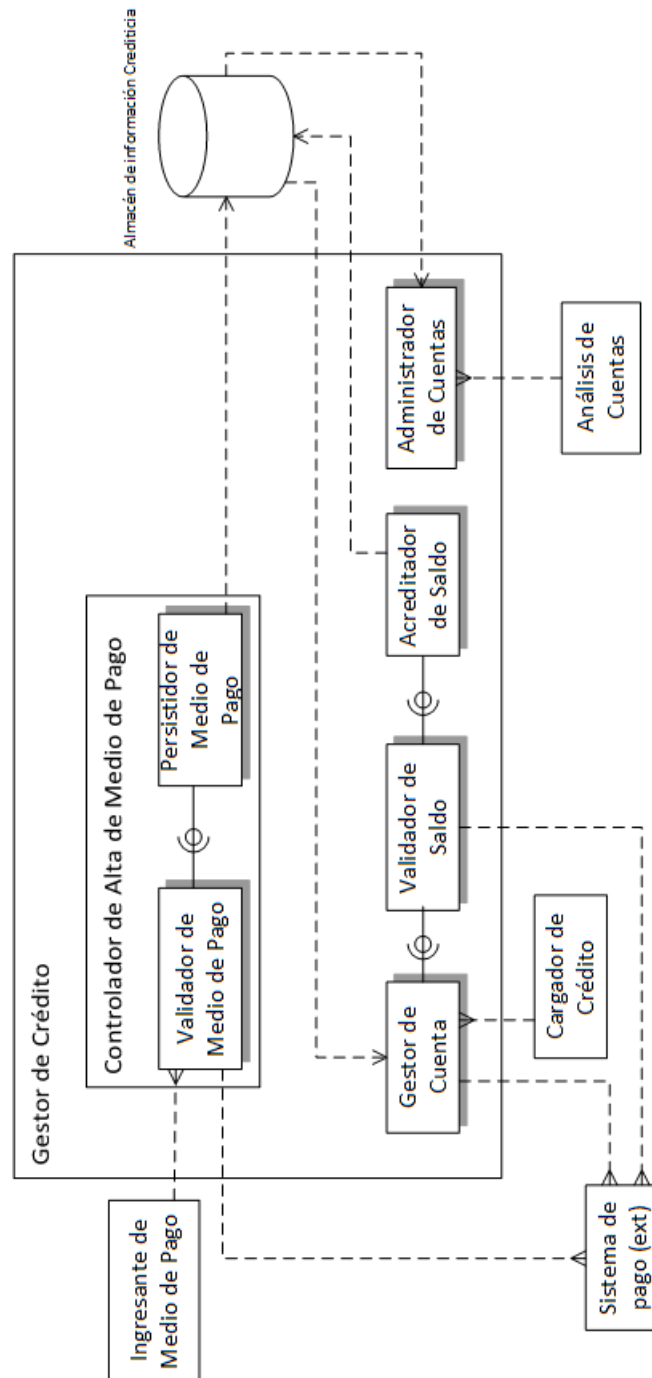
Toda solicitud de registro al sistema es recibida por el componente *Controlador de Alta de Usuario*, que consta de tres etapas.

El *Validador de Datos* es la primera. Tal componente se encarga de validar las distintas posibilidades del usuario con respecto a las reglas gubernamentales establecidas en su país (que se consultan en el repositorio correspondiente). Es, básicamente, un filtro que le permite o no al usuario registrarse de acuerdo a la legislación vigente del territorio desde donde esté entrando al sistema.

Una vez validados los datos, si el usuario incluyó en su registro un método de pago, el *Controlador de Alta de Medios de Pago* validará la información relativa al mismo con el servicio externo correspondiente, y la almacenará en el repositorio adecuado en caso de que sea correcta. El zoom de este componente se verá en la figura 6.3.

Por último, la tercer etapa, el *Persistidor de Usuarios*, es el encargado, ya con todos los datos anteriores validados, de almacenarlos en el repositorio *Almacén de Usuarios*, efectivizando así el registro del usuario.

6.3. Gestor de Crédito



Si un usuario desea pasar del servicio de modalidad gratuita (p.ej., en caso de no haber ingresado un medio de pago a la hora de registrarse) de nuestro sistema al servicio pago, puede realizarlo en cualquier momento (*Ingresante de Medio de Pago*). El componente *Controlador de Alta de Medio de Pago* es el encargado de recibir tales solicitudes, que siempre se encuentran encriptadas.

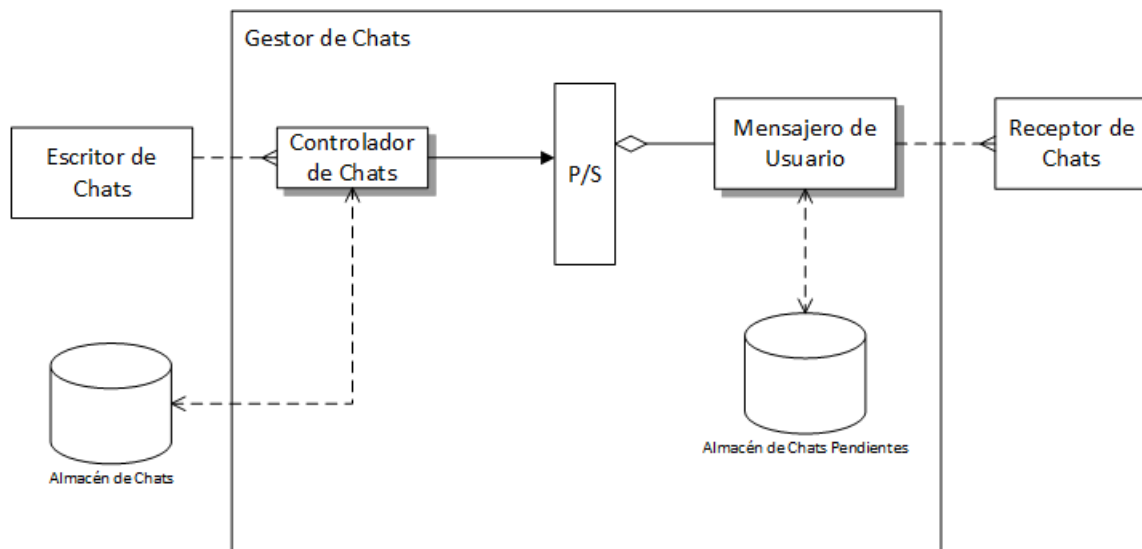
Este componente cuenta de dos pasos. El primero, *Validador de Medio de Pago*, se encarga de validar la información ingresada comunicándose con el correspondiente servicio de medio de pago externo. El segundo, *Persistidor de Medio de Pago*, es quien -una vez todas las validaciones hayan sido corroboradas y aceptadas- persiste esta información en el *Almacén de Información Crediticia*.

Un usuario puede cargar crédito a su cuenta en cualquier momento (*Cargador de Crédito*). Esta infor-

mación llega al componente de manera encriptada, y deberá pasar por un pipeline de validaciones, que comienza en el *Gestor de Cuenta*. Este primer paso consta de obtener todos los datos correspondientes a la cuenta, y validar que todavía esté activa. Una vez esta información haya sido confirmada por el *Sistema de Pago (externo)*, el validador de saldo será el encargado de corroborar que la cifra a cargar en cuestión, tenga un respaldo en saldo en el edio de pago del usuario (consultando, nuevamente, al *Sistema de Pago (externo)*). Si se confirma esto, el *Acreditador de Saldo* será el que se comunique con el repositorio *Almacén de Información Crediticia* e ingrese el nuevo saldo del usuario.

Cuando un usuario administrador desee consultar los datos de alguna cuenta (*Análisis de Cuentas*), lo hará a través de un componente con privilegios especiales, el *Administrador de Cuenta*, que tiene la capacidad de mostrar y analizar la información crediticia de todos los usuarios. Las comunicaciones con este componente se encuentran también cifradas.

6.4. Gestor de Chats

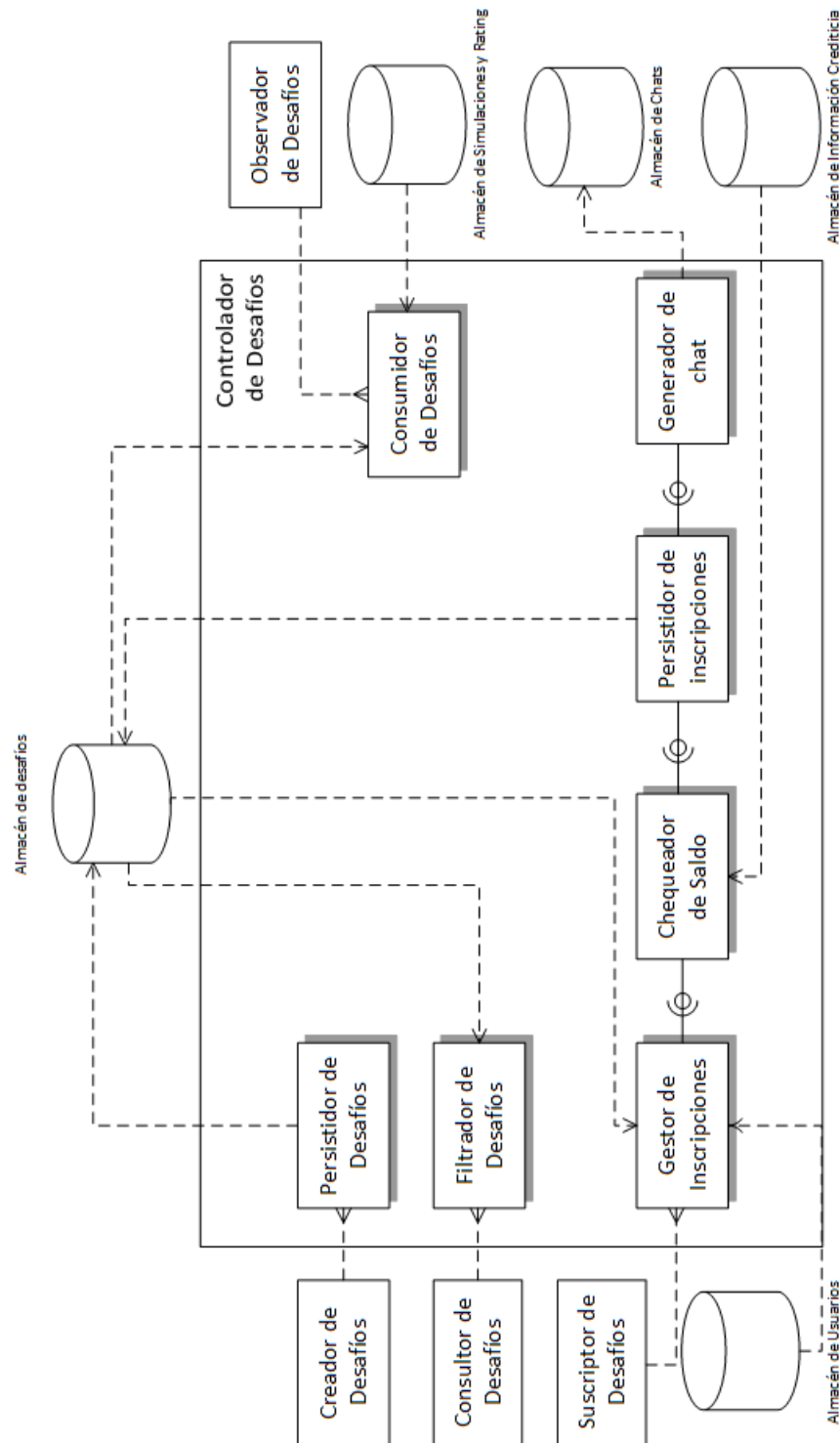


Los usuarios pueden escribir chats (*Escritor de Chats*) o bien ser quien los recibe (*Receptor de Chats*). Esto ocurre en los distintos tipos de chats que ofrece el sistema, tanto el global del desafío como el individual.

Quien desee escribirle a otro usuario, se comunicará con el *Controlador de Chats*, quien validará que la persona está disponible para comunicarse en ese chat global en el caso de que sea un chat de desafío, o creará/buscará un chat con el nuevo participante en el caso de los chats individuales. Una vez validada esta información, el componente publicará mediante uno conector del tipo **Publish/Subscribe** la nueva información de chat, y será alguno de los *Mensajero de Usuario* quien se encargará de ponerlo a disponibilidad del receptor.

Si el usuario receptor no está conectado para recibir el mensaje, el *Mensajero de usuario* cuenta con su propio repositorio (*Almacén de Mensajes Pendientes*, lo que le permite mantener control sobre los mensajes que aún le restan enviar.

6.5. Controlador de Desafíos



El *Controlador de Desafíos* es ya un componente más complejo que el anterior, y ofrece diversas interacciones.

Por un lado, el *Creador de Desafíos* se comunica con el *Persistidor de Desafíos* y le comunica sus intenciones de crear un nuevo desafío; el *Persistidor* almacena el nuevo desafío en el *Almacén de Desafíos*, desde donde puede ser consultado por otros de los componentes.

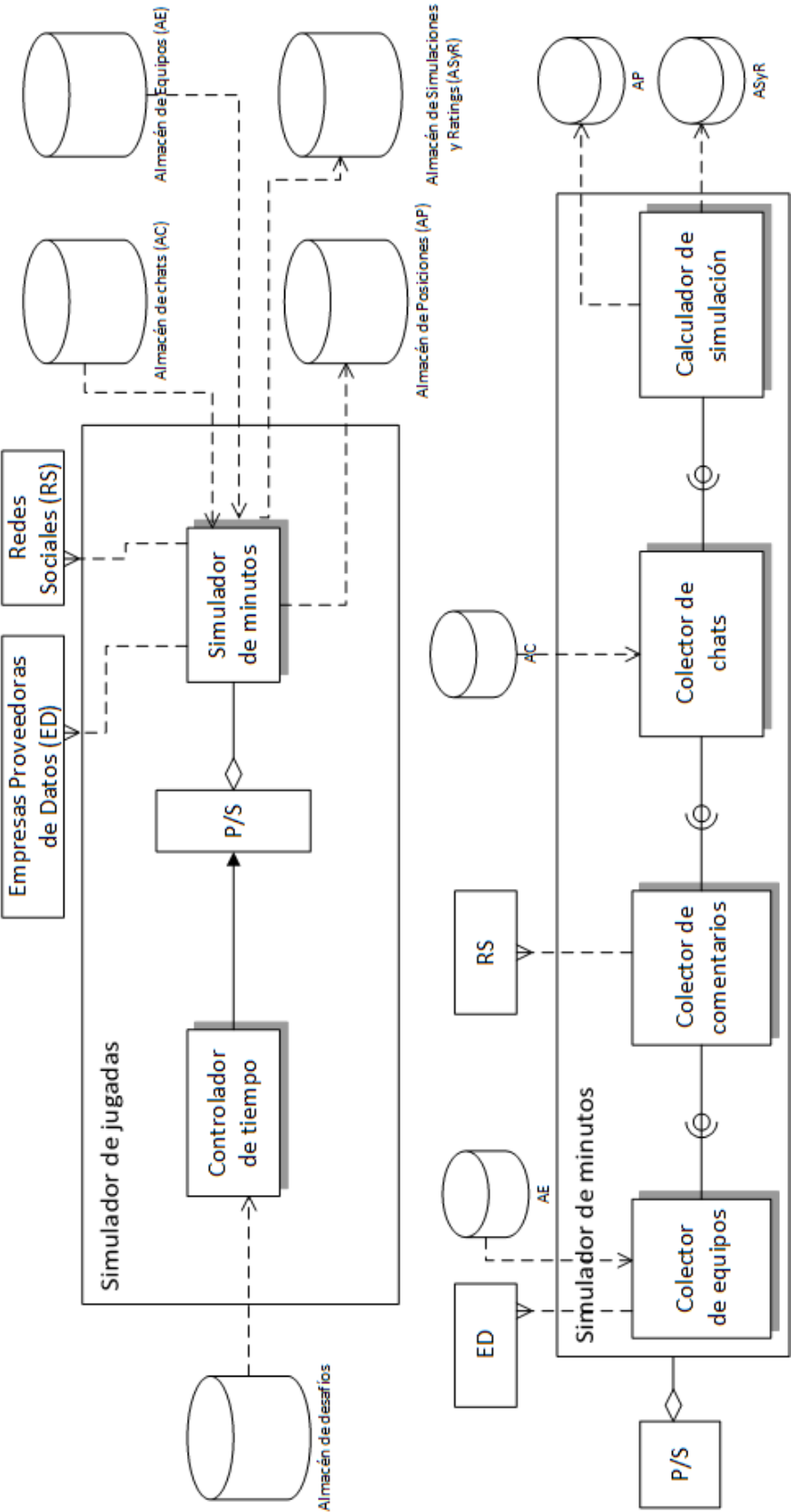
Cuando un usuario desea consultar los desafíos a los que puede inscribirse (*Consultor de Desafíos*), el

Filtro de Desafíos se encarga de interactuar con los datos del usuario y el *Almacén de Desafíos*, para presentar la información solicitada de manera correcta.

Cuando un usuario desea inscribirse en un desafío (*Suscriptor de Desafíos*), el *Gestor de Inscripciones* es quien recibe la petición con todos los datos del usuario y el desafío en cuestión, y analiza si es posible o no cumplir con el pedido. Si lo es, se le transmite la información al *Chequeador de Saldo*, quien validará -en caso de ser necesario- que el usuario cuente con el saldo suficiente para participar del desafío. Estos datos, luego, pasan por un pipe al componente siguiente, quien es el encargado de persistir la información del registro del usuario en el desafío elegido, dejando constancia de la inscripción. Finalmente, el *Generador de Chat* es quien inscribe al nuevo usuario en el chat global del desafío.

Por otro lado, las noticias y comunicaciones relativas a los desafíos, estén dirigidas a usuarios inscriptos o no, son leídas por el *Consumidor de Desafíos*, quien es el encargado de, a partir de los datos de los desafíos generados y las simulaciones, ofrecer la información correspondiente a quien sigue un desafío (*Seguidor de Desafíos*).

6.6. Simulador de Jugadas

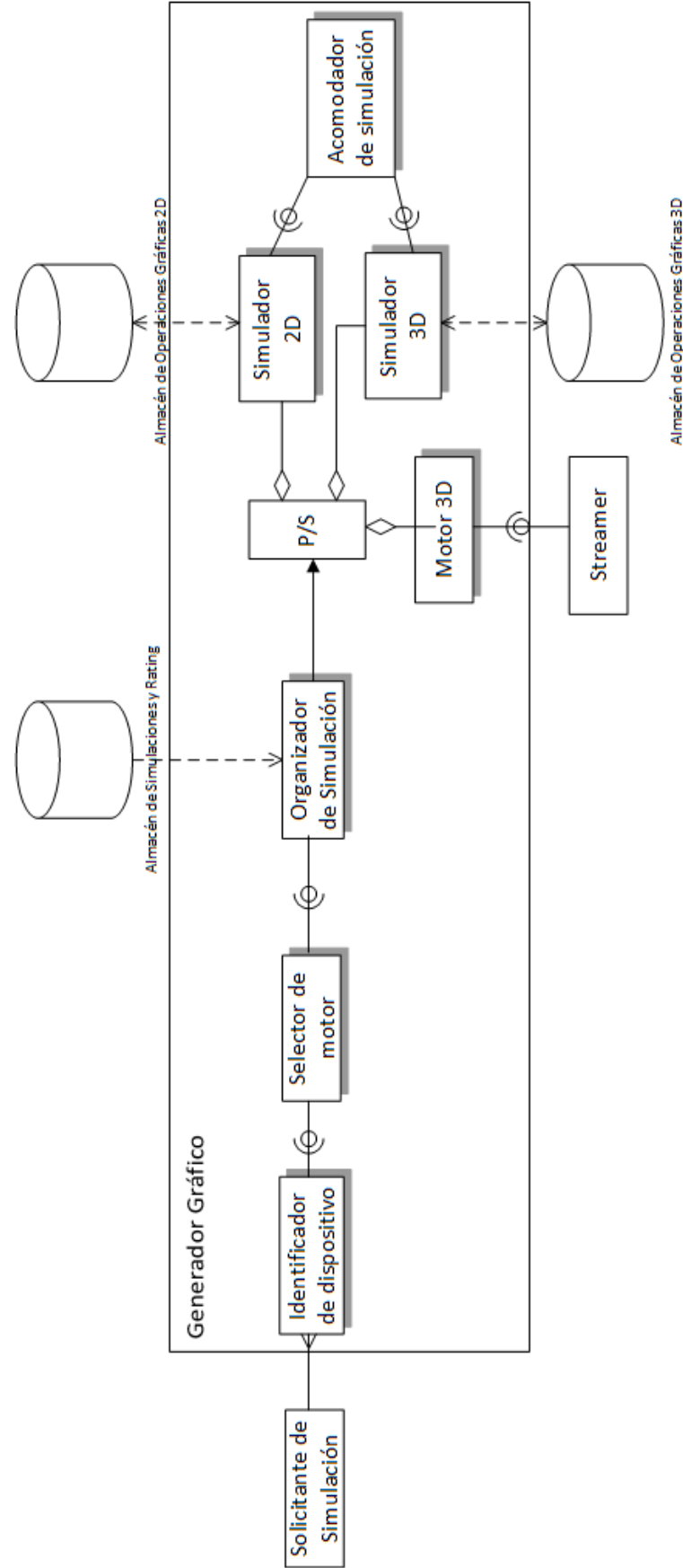


Cada vez que se le dé inicio a un desafío, es el *Controlador de tiempo* del *Simulador de Jugadas* el encargado de publicar la información en un conector de tipo **Publish/Subscribe**. El *Simulador de Minutos* será quien realice todos los cálculos de la simulación, y los dejará disponibles en el repositorio correspondiente (*Almacén de Simulaciones y Ratings*).

Realizando *zoom* sobre el *Simulador de Minutos*, vemos que se descompone de la siguiente manera:

- *Colector de Equipos*: este componente es el encargado de conectarse al *Repositorio de Equipos* y traer la información necesaria de cada uno de los equipos que serán utilizados en la simulación del desafío. La información obtenida es entregada al siguiente componente, el
- *Colector de Comentarios*: este componente recorre las distintas redes sociales de forma paralela con el objetivo de procesar la mayor cantidad de comentarios sobre los jugadores de las últimas horas para influir sobre el resultado de la simulación. Obtenida esta información (**Nota:** asumimos que todas las redes sociales cuentan con algún servicio que pueda responder con baja latencia a búsquedas del tipo *¿Lionel Messi?*), se enviará por medio de un pipe al siguiente componente en el camino, el
- *Colector de Chats*: este componente/conjunto de componentes en paralelo se conecta con el repositorio *Almacén de Chats* para buscar información y comentarios sobre los jugadores, técnicos, etc. de cada uno de los equipos del desafío. Recabada esta información, pasa al siguiente componente,
- *Calculador de Simulación*: cuando llegue la información a este componente, la simulación será calculada y persistida en el *Almacén de Simulaciones y Ratings*, junto al desempeño de cada uno de los jugadores participantes en el desafío y, en caso de estar calculando la última jugada del partido, las nuevas posiciones de los participantes en cuestión (persistidas en el *Almacén de Posiciones*).

6.7. Generador Gráfico



Cuando un usuario solicite ver una simulación, será el *Identificador de Dispositivo* el primero en recibir este pedido, y analizar las capacidades gráficas con las que cuenta el aparato en cuestión.

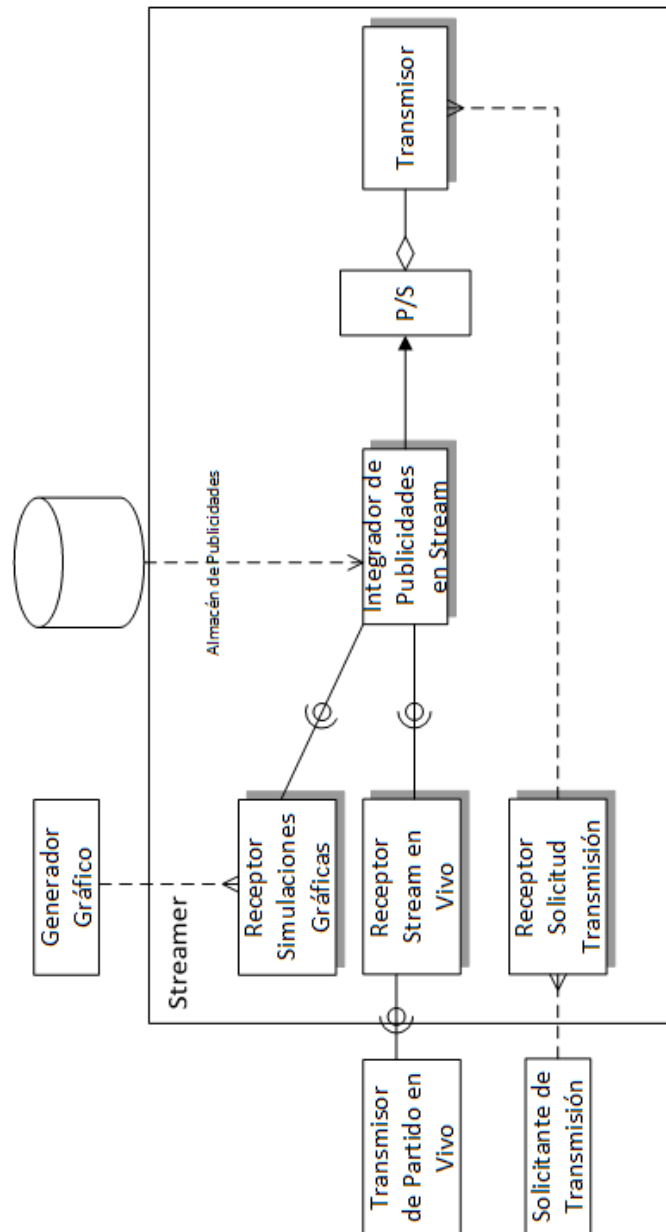
Esta información es enviada al *Selector de Motor*, quien toma la decisión sobre qué motor es el más adecuado (si es que hay uno) para el dispositivo que realizó la solicitud, y envía el resultado mediante un pipe al *Organizador de Simulación*.

Este componente es el encargado de buscar todos los datos necesarios en el repositorio *Almacén de Simulaciones y Ratings*, y publicarlos con un conector Publish/Subscribe junto a la información del motor necesario.

Según el tipo de motor elegido (2D, 3D), el componente adecuado (*Simulador {2D,3D}*) lee esa información del Publish/Subscribe, y es el encargado de calcular a partir de los datos recopilados, las instrucciones que reciben los motores gráficos de los dispositivos. Este componente consulta a su propio repositorio (*Almacén de Operaciones Gráficas {2D,3D}*), con el fin de tener un caché de aquellos cálculos que se realizan de forma más frecuente. En el caso de que no sea posible utilizar el motor gráfico del dispositivo, un *Motor 3D* en nuestros servidores recibe directamente las instrucciones para la simulación, la genera y la envía al *Streamer*.

Una vez que las instrucciones para los motores (2D, 3D) de la simulación gráfica fueron generadas, este stream de datos que se calcula en tiempo real, será enviado a un último componente, el *Acomodador de Simulación*. La función del mismo es enviar el stream como una serie de paquetes ordenados al dispositivo (de forma parecida al protocolo de sliding window en las conexiones TCP de Internet).

6.8. Streamer



El componente *Streamer* es el encargado de realizar la transmisión de video hacia los dispositivos de los usuarios en dos casos, cuando los motores gráficos no soportan al dispositivo en cuestión, o cuando el usuario decidió ver la transmisión de un partido en vivo.

El usuario (*Solicitante de Transmisión*) se comunica con el sistema cuando desea ver un partido en vivo antes de que empiece, y su pedido es recibido por el *Receptor Solicitud de Transmisión*, quien lo reenvía al *Transmisor*. Éste se suscribe al conector *Publish/Subscribe* con el tipo de evento solicitado por el usuario, y queda a la espera de que sea transmitido.

Cuando alguna empresa con los derechos de transmisión de algún partido (*Transmisor de Partido en Vivo*) desea ofrecerlo en nuestra plataforma, se comunica con nuestro sistema y el pedido es recibido por el *Receptor de Stream en Vivo*. El receptor encola este stream en el *Integrador de Publicidades en Stream*.

Este componente agrega a las transmisiones (ya sean en vivo o simulaciones) las publicidades que se encuentren activas en el repositorio *Almacén de Publicidades*. Luego, se conecta mediante el conector

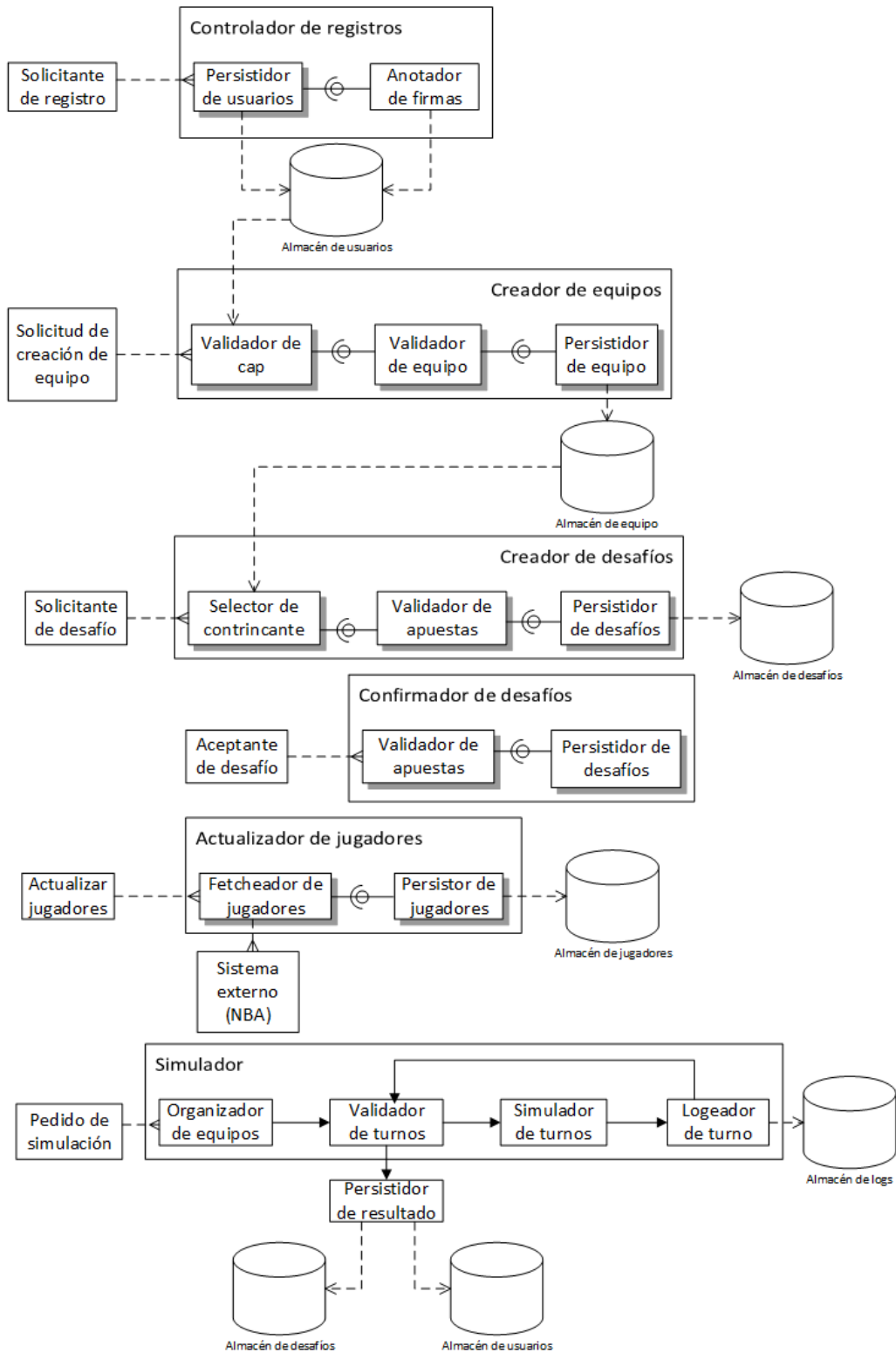
Publish/Subscribe al *Transmisor*, al que le deja disponible el partido con publicidades integradas y listo para transmitir al *Solicitante de Transmisión*.

En caso de las simulaciones que no pueden ser generadas en un dispositivo, llega el pedido desde el *Generador Gráfico* al *Receptor de Simulaciones Gráficas*, que encola también el video listo para transmitir en el *Integrador de Publicidades en Stream*.

7. Arquitectura TP1

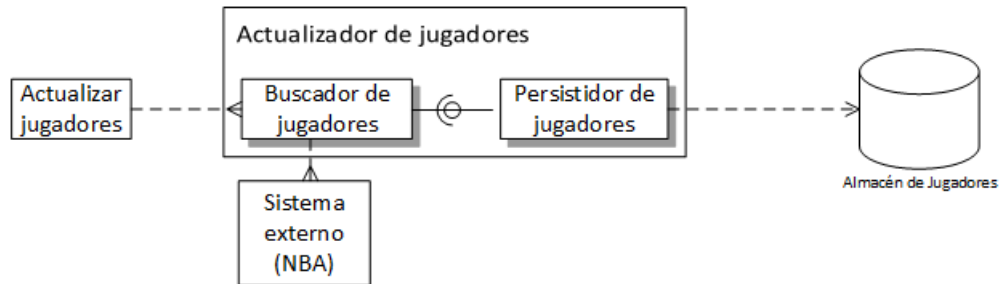
A continuacion detallaremos la arquitectura diseñada en base al enunciado del TP1.

7.1. Paneo General



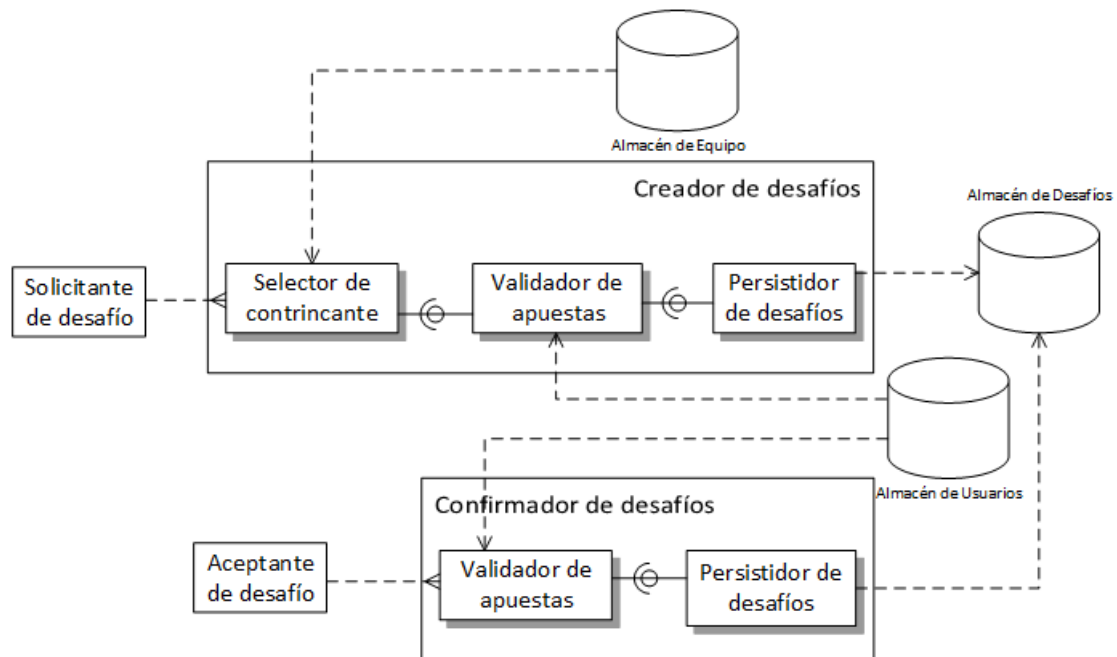
En la figura 7.1 se ven los principales componentes del sistema, sobre los cuales se hace zoom en las secciones posteriores, además de los repositorios utilizados para nuestra solución.

7.2. Actualizador de jugadores



Dado que existe la posibilidad de actualizar las estadísticas de los jugadores mediante un sistema externo, un administrador podrá realizar esta solicitud que será enviada directamente al *Buscador de jugadores* quien se conectará directamente con el sistema externo para traer la última versión de los datos correspondientes. Esta será enviada directamente al *Persistidor de jugadores* quien actualizará la información de los mismos o agregará nuevos, según sea necesario.

7.3. Creador y confirmador de desafíos

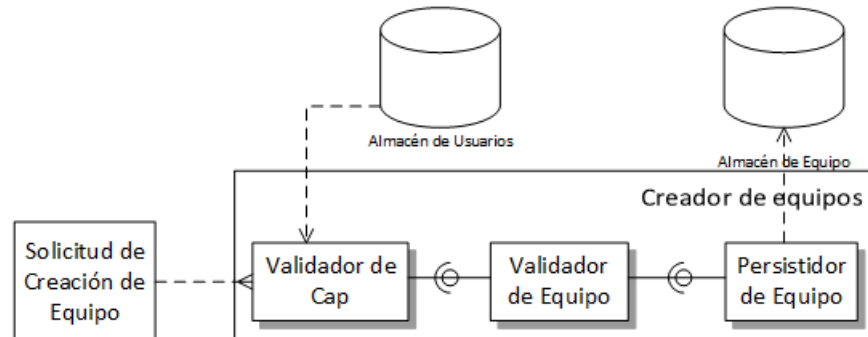


Cuando un usuario solicite desafiar a otro, se enviara la información correspondiente a un componente *Selector de contrincantes* que se encargara de buscar toda la información correspondiente a ambos equipos y la enviara al *Validador de apuestas*. Este último será quien deba confirmar que ambos participantes cuentan con las fichas necesarias (por lo menos en primera instancia), para poder participar del desafío y descontara las fichas necesarias del usuario que solicito el desafío.

Cuando el participante desafiado acepte alguno de los desafíos que se le envían, la comunicación se enviara directamente hacia el *Validador de apuestas*. Este componente validara una vez más que el equipo desafiado cuente con las fichas necesarias (para que no se dé el caso en que, luego de haber

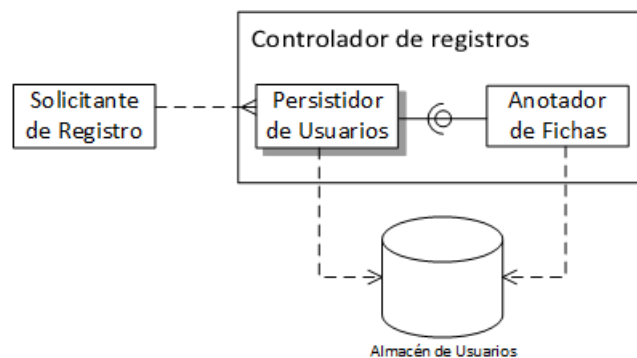
sido desafiado, haya gastado sus fichas en otros desafíos) y luego se comunicara con el *Persistidor de desafíos* para que el desafío sea finalmente confirmado, ya por ambas partes.

7.4. Creador de equipos



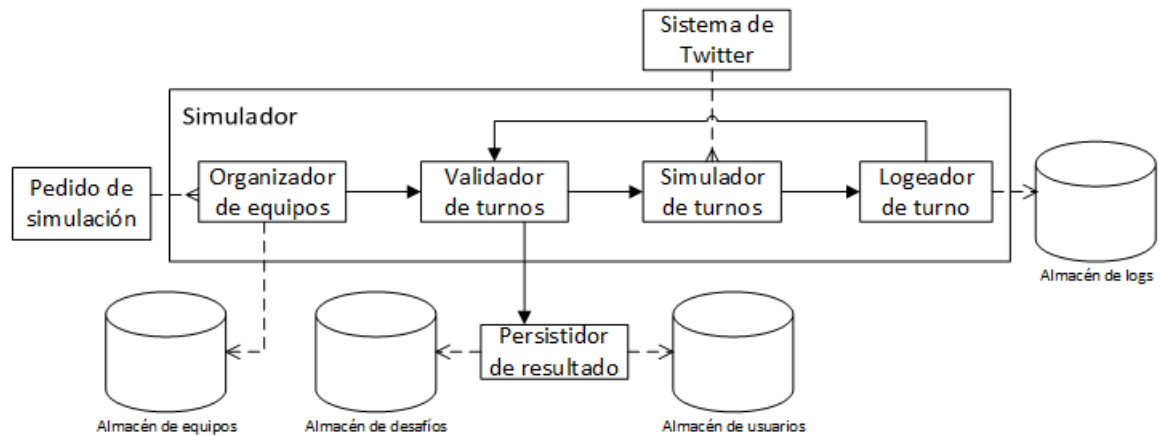
Toda comunicación que se realice con el fin de crear un nuevo equipo, dará inicio a través de un componente *Validador de CAP* que se encargara, mediante la verificación de los datos almacenados en el repositorio de usuarios, de confirmar que el usuario cuenta con el CAP necesario para poder formar dicho equipo. Un segundo componente (*Validador de equipo*), recibirá la confirmación de este y tendrá como función la de validar la información concreta del equipo (están todas las posiciones ocupadas, jugador estrella elegido, etc.) para luego enviarla al *Persistidor de equipos* que se encargara de almacenarla en el repositorio correspondiente (*Almacén de equipos*).

7.5. Controlador de registros



Toda solicitud de registro de un nuevo usuario es recibida por el *Persistidor de usuarios* encargado de almacenar los datos del mismo en el repositorio *Almacén de usuarios* y envía un mensaje hacia el *Anotador de fichas* que persiste la cantidad de fichas otorgadas al nuevo usuario. El objetivo de mantener estos dos componentes separados es el de ofrecer flexibilidad en el manejo de las fichas y ofrecer la posibilidad de cambiar este módulo con bajo impacto en el sistema.

7.6. Simulador



Cuando un nuevo pedido de simulación llegue al sistema, será recibida por un *Organizador de equipos*. Este componente traerá la información de los dos equipos comprometidos en el desafío y se la enviara al *Validador de turnos*. Este componente, que es parte de un ciclo de ejecución, validara que la cantidad de turnos total configurada en el sistema todavía no se haya cumplido y en caso de que todavía haya pendientes, notificara al *Simulador de turnos*. A través de una comunicación directa con el API de Twitter, este componente calculara los resultados de las simulaciones del turno correspondiente. Estas se irán enviando una tras otra hacia el *Logeador de turnos* que persistirá los datos para que luego estén disponibles para los usuarios. Cuando este componente termine de cumplir su función, notifica una vez más al *Validador de turnos*, para que este revise si todavía quedan pasos por simular. En caso de que esto no sea así, se comunicara con el *Persistidor de resultados*, encargado de almacenar el resultado del desafío y la nueva posición (o el cambio de puntos) de los usuarios participantes.

8. Conclusiones

8.1. Metodologías utilizadas en la materia

Tanto **Unified Process** como **Scrum** son metodologías para el desarrollo de software, que comparten la característica de seguir un modelo iterativo incremental, pero difieren en muchas otras cosas. Aún siguiendo modelos iterativos, por ejemplo, en **Scrum** cada iteración se planifica cuando termina la anterior, mientras que en **UP** se planifican todas al comienzo.

Scrum se vende como una metodología ágil con muy poca planificación, lo que la hace por lo tanto muy flexible y adaptable a los cambios y las diferentes circunstancias que vayan surgiendo a medida que se avanza en el desarrollo. Uno de sus pilares, *individuos e interacciones sobre procesos y herramientas*, favorece la comunicación continua entre los programadores y el Project Owner (en forma de Stand Up Meetings), de manera de poder ir resolviendo rápidamente cualquier problema o inquietud que haya surgido.

Otra máxima de la metodología **Scrum** es *Software andando sobre documentación exhaustiva*, la cual pone en un lugar de privilegio al cliente, ya que éste valora más tener un software andando a documentación de desarrollo que nunca use y tal vez ni siquiera vea (hay que tener en cuenta, también, que muchas veces el cliente no sabe lo que quiere, por lo que empezar con el producto desde temprano permite que se definan mejor sus requerimientos). Es por esto también que a medida que se planea cada iteración de **Scrum** (donde están en juego todas las fases de una metodología de desarrollo clásica: requerimiento, análisis, diseño, desarrollo, testing, etc.), se privilegian las *user stories* que le dan más *business value* al cliente. Cada iteración de **Scrum** (inmodificable; las modificaciones que surjan deberán llevarse a cabo sí o sí en futuras iteraciones) concluye con la presentación de un *entregable*, software ya andando y que debería poder ser puesto en producción, para poder obtener feedback.

Unified Process, en cambio, sigue un proceso mucho más estricto guiado por la arquitectura y los casos de uso del software a desarrollar. Consta de 4 fases bien definidas (Inception, Elaboration, Construction y Transition), que se diferencian entre sí según la actividad en la que se haga hincapié (análisis del problema, desarrollo de la solución, etc., por poner dos ejemplos contrapuestos). Este enfoque más estructurado y con foco en la documentación (*conocimiento*) permite ir desarrollando un modelo mental del software a construir e ir entrando en contacto con el problema y las maneras de solucionarlo.

A diferencia de **Scrum**, en **UP** el objetivo es en cada iteración ir evitando (o, de ser imposible, reduciendo) los riesgos del sistema a desarrollar, por lo que resulta clave tener un buen entendimiento del dominio del problema -e idealmente, experiencia en él- para ir previendo lo que pueda surgir. Si bien pueden haber entregables en cada etapa de **UP**, no necesariamente son *código ejecutable*: pueden ser documentos de requerimiento, de análisis de dominio, u otros.

Al tener mucha planificación de entrada, resulta más fácil en **UP** tener una idea estimada de cuándo finalizará el proyecto, algo que no sucede en **Scrum** (siempre y cuando, claro, que en UP se hayan estimado bien las cosas, algo que no siempre sucede).

Por lo descrito, **Scrum** se presenta como una solución ventajosa para proyectos poco críticos y de poca duración, con equipos chicos y sin fechas de finalización estrictas, donde la adaptabilidad y flexibilidad sean importantes para la concreción del desarrollo. Por el enfoque más formal, **UP** se presenta como una solución ideal para proyectos grandes, complejos, con equipos numerosos, y donde las fechas sean estrictas y jueguen un rol importante.

8.2. Programming in the large vs Programming in the small