

The Ionian Bookshelf

Documento de Requisitos

Índice

Esquema del documento	2
Descripción del sistema	2
Glosario de términos	3
Entidades involucradas	3
Usuarios	3
Entidades del sistema	4
Historias de usuario	4
Administrador	4
Registro de revisores	4
Gestión de campeones	5
Eliminación de campeones	5
Gestión de ítems	6
Gestión de runas	6
Eliminación de mensajes en hilos	6
Invocador	7
Gestión de páginas de runas	7
Gestión de builds	8
Modificar privacidad en las builds	9
Crear propuestas de campeón	12
Revisores	13
Eliminación de mensajes en hilos	13
Baneo de Invocadores	13
Gestión de propuestas	14
Historias de usuario que afectan a todos los tipos de usuario	14
Gestión de hilos	14
Mostrar hilos	14
	1

Reglas de negocio	16
Hilos de ligas	16
Hilos de campeones	16
Nivel de invocador	16
Límite de ítems en builds	16
Reporte automático de hilos mal votados	17
Propuesta sobredimensionada de campeón	17
Planificación del proyecto	17

Esquema del documento

Documento de descripción del proyecto que incluye:

- Una descripción a grandes rasgos de la aplicación web que se implementará.
- Entidades involucradas en la aplicación web
- Historias de usuario (distintas de la gestión de usuarios y login).
 - Al menos un escenario positivo y un escenario negativo definidos por cada historia de usuario
- Planificación de las historias de usuario para los sprints siguientes y una asignación de historias por parejas.

Descripción del sistema

La aplicación web que vamos a desarrollar es un sistema de información basado en el videojuego League of Legends. El sistema almacenará la información de los Campeones, Ítems y Ligas del juego. Dentro del sistema podemos encontrar los roles de Administrador, Revisor e Invocador.

Contaremos con un foro, formado por hilos. Habrá un hilo por cada Liga existente en el juego, donde solo podrán añadir mensajes Invocadores que se encuentren en esa determinada Liga. Los Administradores podrán escribir mensajes en cualquier hilo independientemente de la liga en la que se encuentren.

Los hilos podrán tratar sobre builds formadas por Ítems de determinados campeones y Página de Runas, estas builds serán creadas por los Invocadores, Revisores o

Administradores. En los mensajes de dichos hilos, un segundo Invocador podrá sugerir un cambio de items a la build inicial.

Además los Invocadores podrán sugerir cambios a las características de los Campeones e Ítems, estos cambios podrán ser aceptados/denegados por los Revisores y Administradores del sistema.

Glosario de términos

- **Build:** Conjunto de ítems y runas asociada a un campeón.
- **Campeón:** Personaje jugable en una partida de League of Legends.
- **Habilidad:** Acción específica que puede realizar un campeón. Cada campeón tiene un conjunto de habilidades.
- **Habilidad pasiva:** Tipo de habilidad que posee un campeón y que siempre está activa o se activa periódicamente de forma automática (Cada Campeón posee habilidades pasivas diferentes).
- **Invocador:** Jugador de League of Legends.
- **Item:** Objeto comprado en una partida de League of Legends.
- **Liga:** Clasificación de un invocador competitivamente. En orden ascendente: Bronce, Plata, Oro y Platino.
- **Main:** Campeón más jugado por un Invocador.
- **Runa:** Conjunto de ítems que pueden ser seleccionados y combinados en cada partida, independientemente del campeón. Se agrupa en páginas compuestas por 1 rama principal, compuesta de 1 runa clave y 3 menores y 1 rama secundaria, compuesta de 2 runas menores. Hay 3 nodos de runas menores en cada rama, no pueden seleccionarse más de 1 runa menor de cada nodo.

Entidades involucradas

Usuarios

Administrador: Aquel usuario encargado de administrar el sistema.

Invocador: Aquel usuario para el que el sistema está más enfocado. Estará relacionado con la mayor parte del sistema.

Revisor: Aquel usuario encargado de moderar ciertos aspectos del sistema.

Entidades del sistema

Campeón: Entidad principal de la que dependen en gran parte de las demás. Hace referencia a cada personaje que se puede usar en una partida de League of Legends.

Item: Objetos que pueden ser comprados durante una partida de League of Legends.

Runas: Habilidades que los invocadores seleccionan para sus partidas. Se organizan en ramas de runas y en nodos.

Página de runas: Conjunto de runas.

Hilo: Conjunto de mensajes relativos a un tema. Cada build tendrá un hilo asociado donde podrá comentarse ésta o sugerir cambios.

Mensaje: Comentario o respuesta que se da a un hilo.

Builds: Conjuntos de objetos y runas seleccionados en una partida para un campeón.

Voto: Valoración positiva o negativa que un invocador haga sobre un campeón, una build, un hilo o mensaje.

Reporte: Informe donde se notifica una conducta negativa.

Historias de usuario

Administrador

Registro de revisores

Como Administrador,

Para apoyar la labor de revisión de determinadas funciones del sistema como la revisión de mensajes o las distintas propuestas de cambio.

Quiero poder registrar usuarios del tipo revisor.

Caso positivo	Caso negativo
Añadir al revisor "Luis-Rev" al Repositorio de Revisores. Llamar al método ListaRevisores(). Comprobar que "Luis-Rev" está en la lista devuelta.	Asegurar que "Luis-Rev" aparece al llamar al método ListaRevisores(). Añadir de nuevo a "Luis-Rev" al Repositorio de Revisores. Comprobar que salta la excepción

	apropiada, debido a que el revisor ya existía.
--	--

Gestión de campeones

Como Administrador,

Para mostrar su nombre, vida, maná/energía, velocidad de ataque, daño de ataque, velocidad de movimiento base

Quiero poder crear y modificar.

Caso positivo	Caso negativo
Añadir al campeón "Kiko" al Repositorio de Campeones. Llamar al método ListaCampeones(). Comprobar que "Kiko" está en la lista devuelta.	Asegurar que "Kiko" aparece al llamar al método ListaCampeones(). Añadir de nuevo a "Kiko" al Repositorio de Campeones. Comprobar que salta la excepción apropiada, debido a que el campeón ya existía.

Eliminación de campeones

Como Administrador,

Para no mostrar campeones que ya no se encuentran disponibles.

Quiero poder eliminar campeones.

Caso positivo	Caso negativo
Asegurar que el campeón "Rodri" aparece a llamar al método ListaCampeones(). Eliminar a "Rodri" del Repositorio de Campeones. Llamar de nuevo al método ListaCampeones() y comprobar que efectivamente el campeón "Rodri" ya no se encuentra en el repositorio.	Asegurar que el campeón "Rodri" no aparece a llamar al método ListaCampeones(). Eliminar a "Rodri" del Repositorio de Campeones. Comprobar que salta la excepción apropiada, debido a que el campeón no existía.

Gestión de ítems

Como Administrador,

Para mostrar su nombre, descripción y sus estadísticas.

Quiero poder crear y eliminar ítems

Caso positivo	Caso negativo
Crear un ítem "Item1" con nombre, descripción y estadísticas Listar todos los ítems Comprobar que "Item1" está en la lista y es igual que como lo introducimos	Crear un ítem "Item2" con estadísticas y descripción pero sin nombre Comprobar que no se crea "Item2" y se muestra el error "Nombre no puede estar vacío"

Gestión de runas

Como Administrador,

Para mostrar su nombre, descripción, rama, nodo y sus estadísticas.

Quiero poder crear, actualizar y eliminar runas

Caso positivo	Caso negativo
Crear una runa "Runa1" con nombre, descripción, rama, nodo y estadísticas Listar todas las runas Comprobar que "Runa1" está en la lista y es igual que como lo introducimos	Crear runa con nombre vacío Comprobar que se devuelve la excepción apropiada

Eliminación de mensajes en hilos

Como Administrador,

Para eliminar los comentarios de spam o que incumpla las reglas de la comunidad.

Quiero poder eliminar automáticamente mensajes marcados como negativos.

Caso positivo	Caso negativo
Eliminar mensaje "Mensaje1" Listar todos los mensajes Comprobar que "Mensaje1" no está en la lista	Llamar al método eliminarMensaje(null) Comprobar que se devuelve la excepción apropiada

Invocador

Gestión de páginas de runas

Como invocador,

Para poder gestionar mis páginas de runas.

Quiero poder crear y actualizar mis páginas de runas con 6 runas, 4 de una rama principal, siendo una de ellas una runa clave; 2 de una rama secundaria, sin ninguna clave; y todas de distintos nodos en su rama.

Caso positivo	Caso negativo
Añadir una página de runas "PRunas1" con una rama principal, una rama secundaria seleccionadas, una runa clave de la rama principal, 3 runas de distintos nodos de la rama principal y 2 runas no claves de distintos nodos de la rama secundaria. Llamar al método ListaPaginaRunas() del invocador que ha creado la build. Asegurar que en esta lista de páginas de runas se encuentra "PRunas1".	Añadir una página de runas sin ninguna runa clave. Comprobar que se devuelve la excepción apropiada
	Añadir una página de runas con rama principal seleccionada "Precisión" y rama secundaria seleccionada "Precisión" Comprobar que se devuelve la excepción apropiada
	Crear runas clave "Conquistador" y "Compás Letal" Añadir una página de runas con runa clave seleccionada "Conquistador" y con runa secundaria seleccionada "Compás Letal"

	Comprobar que se devuelve la excepción apropiada
	<p>Crear runas de nodo 3 de la rama "Dominación" "Cazador Voraz" y "Cazador Definitivo"</p> <p>Añadir una página de runas con "Cazador Voraz" y "Cazador Definitivo" entre las runas de la rama principal</p> <p>Comprobar que se devuelve la excepción apropiada</p>
	<p>Crear runa de rama "Precisión" "Tenacidad" y runa de rama "Valor" "Sobrecrecimiento"</p> <p>Añadir una página de runas con rama principal "Precisión" y rama secundaria "Valor", con runas principales "Tenacidad" y "Sobrecrecimiento"</p> <p>Comprobar que se devuelve la excepción apropiada</p>

Gestión de builds

Como invocador,

Para poder gestionar mis builds.

Quiero poder crear y actualizar mis builds.

Caso positivo	Caso negativo
<p>Añadir una build "Build3" asociada a un Campeón y un invocador y que incluya entre 1 y 6 items y 6 runas.</p> <p>Llamar al método ListaBuilds() del invocador que ha creado la build.</p> <p>Asegurar que en esta lista de builds se encuentra "Build3".</p>	<p>Añadir una build "BuildX" asociada a un Campeón y un invocador y que incluya entre 1 y 6 runas u objetos, de los cuales 1 de esos items está prohibido para el Campeón asociado.</p> <p>Comprobar que salta la excepción apropiada, debido a que había un item prohibido para el Campeón asociado.</p>

Modificar privacidad en las builds

Como invocador,

Para poder gestionar quién puede ver mis builds.

Quiero poder cambiar la privacidad de mi build entre privada o pública.

Caso positivo	Caso negativo
Crear una build "Build1" Comprobar que "Build1" es privada por defecto	Crear una build "Build4" privada Comprobar que "Build4" no es accesible por un usuario cualquiera y muestra el mensaje de error apropiado
Crear una build "Build2" y marcarla como pública Comprobar que "Build2" es accesible por un usuario cualquiera	
Crear una build "Build3" privada Compartir "Build3" con un invocador "Invocador1" Comprobar que "Invocador1" puede acceder a la build	

Puntuar hilo

Como invocador,

Para saber cómo de bueno es un hilo,

Quiero poder puntuar un hilo de forma positiva (up vote) o negativa (down vote).

Caso positivo	Caso negativo
<ul style="list-style-type: none">· Crear invocador "Invocador1"· Crear hilo "Hilo1"· Invocar el método puntuarHilo("positivo", Invocador1, Hilo1)· Se ejecuta el método Hilo1.getPuntuacion()· Retorna la puntuación que tenía antes de la prueba más uno	<ul style="list-style-type: none">· Crear invocador "Invocador1"· Se invoca el método puntuarHilo("positivo", Invocador1, null)· Comprobar que se muestra la excepción apropiada, ya que debe haber un hilo al que puntuar

	<ul style="list-style-type: none"> · Crear hilo "Hilo1" · Se invoca el método <code>puntuarHilo("positivo", null, Hilo1)</code> · Comprobar que se muestra la excepción adecuada, ya que debe haber un invocador que puntúe
	<ul style="list-style-type: none"> · Crear invocador "Invocador1" · Crear hilo "Hilo1" · Se invoca el método <code>puntuarHilo("positivo", Invocador1, Hilo1)</code> · Se invoca de nuevo el método <code>puntuarHilo("positivo", Invocador1, Hilo1)</code> · Comprobar que se muestra la excepción apropiada, ya que un invocador no puede puntuar positivamente más de una vez el mismo hilo

Puntuar mensajes

Como invocador,

Para saber cómo de bueno es un mensaje,

Quiero poder puntuar un mensaje de forma positiva (up vote) o negativa (down vote).

Caso positivo	Caso negativo
<ul style="list-style-type: none"> · Crear invocador "Invocador1" · Crear mensaje "Mensaje1" · Invocar el método <code>puntuarMensaje("positivo", Invocador1, Mensaje1)</code> · Se ejecuta el método <code>Mensaje1.getPuntuacion()</code> · Retorna la puntuación que tenía antes de la prueba más uno 	<ul style="list-style-type: none"> · Crear invocador "Invocador1" · Se invoca el método <code>puntuarMensaje("positivo", Invocador1, null)</code> · Comprobar que se muestra la excepción apropiada, ya que debe haber un mensaje al que puntuar
	<ul style="list-style-type: none"> · Crear mensaje "Mensaje1" · Se invoca el método <code>puntuarHilo("positivo", null, Mensaje1)</code> · Comprobar que se muestra la excepción adecuada, ya que debe haber un invocador que puntúe
	<ul style="list-style-type: none"> · Crear invocador "Invocador1" · Crear mensaje "Mensaje1" · Se invoca el método <code>puntuarMensaje("positivo", Invocador1, Mensaje1)</code> · Se invoca de nuevo el método <code>puntuarMensaje("positivo", Invocador1, Mensaje1)</code>

	<ul style="list-style-type: none"> · Comprobar que se muestra la excepción apropiada, ya que un invocador no puede puntuar positivamente más de una vez el mismo mensaje
--	---

Puntuar build

Como invocador,

Para saber cómo de buena es una build,

Quiero poder puntuar una build de forma positiva (up vote) o negativa (down vote).

Caso positivo	Caso negativo
<ul style="list-style-type: none"> · Un invocador vota negativo una build · Se ejecuta el método <code>getPuntuacion()</code> de la build · Retorna la puntuación que tenía antes de la prueba menos uno 	<ul style="list-style-type: none"> · Crear invocador "Invocador1" · Se invoca el método <code>puntuarBuild("positivo", Invocador1, null)</code> · Comprobar que se muestra la excepción apropiada, ya que debe haber una build a la que puntuar
	<ul style="list-style-type: none"> · Crear build "Build1" · Se invoca el método <code>puntuarBuild("positivo", null, Build1)</code> · Comprobar que se muestra la excepción adecuada, ya que debe haber un invocador que puntúe
	<ul style="list-style-type: none"> · Crear invocador "Invocador1" · Crear build "Build1" · Se invoca el método <code>puntuarBuild("positivo", Invocador1, Build1)</code> · Se invoca de nuevo el método <code>puntuarBuild("positivo", Invocador1, Build1)</code> · Comprobar que se muestra la excepción apropiada, ya que un invocador no puede puntuar positivamente más de una vez la misma build

Actualizar datos de invocador

Como invocador,

Para poder seleccionar mis mains actuales (campeones)

Quiero poder actualizar mi perfil de invocador.

Caso positivo	Caso negativo
<ul style="list-style-type: none"> · Un invocador cambia un main suyo de Lux a Nasus · Se ejecuta el método getMains() de ese invocador · Retorna una lista de mains donde no está Lux pero si Nasus 	<ul style="list-style-type: none"> · Se invoca el método updateInvocador(null) · Comprobar que se devuelve la excepción apropiada

Crear propuestas de campeón

Como invocador,

Para poder proponer mejoras sobre campeones.

Quiero poder crear propuestas donde sugiero mejoras sobre campeones.

Caso positivo	Caso negativo
Un invocador crea una propuesta "Propuesta1" al campeón "Garen". Selecciona un cambio a la vida, un aumento de 30. Se listan todas las propuestas. Se comprueba que "Propuesta1" está en la lista y que es igual que como lo introdujimos	Un invocador crea una propuesta "Propuesta2" sin campeón asociado. Comprobar que se devuelve la excepción apropiada
Un invocador crea una propuesta "Propuesta6" con campeón asociado "Garen", cuya vida actual es 450. Selecciona un cambio a la vida, se propone un cambio a 899. Listar todas las propuestas. Comprobar que "Propuesta6" aparece en el listado y es igual que como fue introducida	Un invocador crea una propuesta "Propuesta3" con campeón asociado "Garen". Selecciona un cambio a la vida, pero no especifica cómo querría cambiarla. Comprobar que se devuelve la excepción apropiada
Un invocador crea una propuesta "Propuesta7" con campeón asociado "Garen", cuya vida actual es 450. Selecciona un cambio a la vida, se propone un cambio a 226 Listar todas las propuestas. Comprobar que "Propuesta7" aparece en el listado y es igual que como fue introducida	Un invocador crea una propuesta "Propuesta4" con campeón asociado "Garen", cuya vida actual es 450. Selecciona un cambio a la vida, se propone un cambio a 900 Comprobar que se devuelve la excepción apropiada, puesto que no puede aumentarse una característica más del doble de su original
	Un invocador crea una propuesta "Propuesta5" con campeón asociado

	"Garen", cuya vida actual es 450. Selecciona un cambio a la vida, se propone un cambio a 225 Comprobar que se devuelve la excepción apropiada, puesto que no puede reducirse una característica por debajo de la mitad de su original
--	---

Revisores

Eliminación de mensajes en hilos

Como Revisor,

Para eliminar los comentarios de spam o que incumpla las reglas de la comunidad.

Quiero poder eliminar automáticamente mensajes marcados como negativos.

Caso positivo	Caso negativo
Eliminar el mensaje de id "83" en el hilo de id "2". Llamar al método getMensajes() del hilo de id "2". Comprobar que el mensaje de id "83" no se halla en la lista.	Intentar eliminar el mensaje de id "183" en el hilo de id "2", comprobando anteriormente que este mensaje no existe, llamando al método getMensajes(). Comprobar que se devuelve la excepción apropiada.

Baneo de Invocadores

Como Revisor,

Para castigar a los invocadores que reinciden en tener conductas inapropiadas.

Quiero poder limitar su actividad a la hora de crear hilos, mensajes y builds hasta una fecha determinada.

Caso positivo	Caso negativo
Se crea un invocador "Invocador1" Se banea a "Invocador1" hasta <code>LocalDate.now().plusDays(70)</code> Comprobar que se devuelven las excepciones apropiadas cuando se trata de crear un hilo, un mensaje o una build como "Invocador1"	Se invoca al método <code>banInvocador(null, 7)</code> Comprobar que se devuelve la excepción apropiada

Gestión de propuestas

Como Revisor,

Para aceptar o denegar cambios en los campeones o ítems propuestos por invocadores,

Quiero poder visualizar las propuestas de los invocadores sobre los campeones o ítems y aceptarlas/denegarlas y que dichos cambios surtan efecto sobre el ítem o campeón indicado.

Caso positivo	Caso negativo
Aceptar la propuesta de id "23", cuyo estado era pendiente.. Llamar al método getEstado() de la propuesta y comprobar que ha cambiado a "aceptado".	Aceptar la propuesta de id "27", cuyo estado era rechazada. Asegurar que salta el error adecuado debido a que se ha intentado modificar el estado de una propuesta ya resuelta.

Historias de usuario que afectan a todos los tipos de usuario

Gestión de hilos

Como usuario del sistema,

Para poder comunicarme con gente que juegue a LoL,

Quiero poder crear y eliminar hilos.

Caso positivo	Caso negativo
Se crea un hilo "Hilo1" Se listan todos los hilos Comprobar que "Hilo1" está en la lista	Ejecutar el método createHilo(null) Comprobar que se muestra la excepción apropiada

Mostrar hilos

Como usuario del sistema,

Para poder mostrar hilos en función de puntuación, antigüedad, builds relacionadas con mis mains o liga,

Quiero poder elegir en función de qué muestro los hilos.

Caso positivo	Caso negativo
Se crean dos hilos, "Hilo1" e "Hilo2", "Hilo2" con más puntuación que "Hilo1" Se ejecuta mostrarHilosPorPuntuacion() Comprobar que "Hilo2" está antes que "Hilo1" en la lista	Ejecutar el método mostrarHilosDeBuildCampeon(null) Comprobar que se devuelve la excepción apropiada
Se crean dos hilos, "Hilo1" e "Hilo2", "Hilo2" más antiguo que "Hilo1" Se ejecuta mostrarHilosPorFecha() Comprobar que "Hilo2" está antes que "Hilo1" en la lista	
Se crean dos builds, "Build1" y "Build2", "Build1" asociada al campeón "Garen", "Build2" asociada al campeón "Lux" Se crean dos hilos, "Hilo1" e "Hilo2", asociados a "Build1" y a "Build2", respectivamente Se ejecuta mostrarHilosDeBuildCampeon("Garen") Comprobar que "Hilo1" está en la lista y que "Hilo2" no está en ella	

Crear mensaje en hilo

Como invocador,

Para comentar en un hilo,

Quiero poder crear mensajes en un hilo.

Caso positivo	Caso negativo
Se crea el mensaje "Mensaje1" en el hilo "Hilo1" Se listan los mensajes de "Hilo1" Se comprueba que "Mensaje1" aparece en la lista	Se crea un mensaje sin hilo asociado Comprobar que se devuelve la excepción apropiada, ya que un mensaje no puede existir sin estar asociado a un hilo
	Se invoca al método de creación de mensaje con un hilo de la liga "Platino", donde el invocador que escribe el mensaje pertenece a la liga "Bronce" Comprobar que se devuelve la excepción apropiada
	Un invocador con main "Lux" intenta escribir en un hilo sobre el campeón "Garen"

	Comprobar que se devuelve la excepción apropiada
--	--

Reglas de negocio

Hilos de ligas

Un invocador no puede escribir en un hilo de una liga a la que no pertenece.

Hilos de campeones

Un invocador no puede escribir en un hilo de un campeón que no es su main.

Nivel de invocador

Un invocador que no ha alcanzado el nivel 30 no puede tener liga ni escribir en un hilo de liga.

Límite de ítems en builds

Una build debe tener como máximo 6 ítems.

Runa clave no secundaria

Una runa clave no puede ser usada como una runa secundaria.

Límite en runas menores

Sólo puede escogerse 1 runa menor de cada nodo

Límite de selección rama secundaria de runas

Solo pueden escogerse 2 runas menores y la rama ha de ser distinta de la principal

Items prohibidos para campeones

En una build no puede haber objetos que el campeón asociado tenga prohibidos.

Todas las build deben de contener unas botas

En cada build debe de incluir un ítem que sean botas.

Reporte automático de hilos mal votados

Se generará un reporte automático de hilos que tengan el doble de votos negativos que positivos, con un mínimo de 10 votos.

Propuesta sobredimensionada de campeón

Una propuesta de cambio de campeón no puede duplicar su valor original o reducir por debajo de la mitad su valor original.

Planificación del proyecto

La Planificación de las historias de usuario para los sprints siguientes y la asignación de historias por parejas se encuentra en el siguiente documento:

[Planificación de las Historias de Usuario](#)