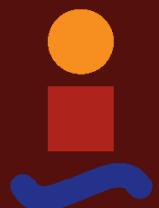


Trabajo Fin de Grado
Grado en Ingeniería de las Tecnologías de
Telecomunicación

Integración de herramientas de gamificación
multijugador en plataformas de Enseñanza Virtual

Autor: Álvaro Ángel Santiago Cuenca
Tutor: Francisco José Fernández Jiménez

Dpto. de Ingeniería Telemática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla
Sevilla, 2024



Trabajo Fin de Grado
Grado en Ingeniería de las Tecnologías de Telecomunicación

Integración de herramientas de gamificación multijugador en plataformas de Enseñanza Virtual

Autor:

Álvaro Ángel Santiago Cuenca

Tutor:

Francisco José Fernández Jiménez

Profesor Colaborador

Dpto. de Ingeniería Telemática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla
Sevilla, 2024

Trabajo Fin de Grado: Integración de herramientas de gamificación multijugador en plataformas de Enseñanza Virtual

Autor: Santiago Cuenca, Álvaro Ángel

Tutor: Fernández Jiménez, Francisco José

El tribunal nombrado para juzgar el Trabajo arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2024

El Secretario del Tribunal

*A mis padres, a mi mujer y a mis
hijos*

Agradecimientos

Me gustaría agradecer este trabajo a todos los profesores de la escuela que han hecho posible que llegue hasta aquí.

También me gustaría mencionar de manera especial a mis padres y a mi hoy mujer, por el apoyo a lo largo de estos años de carrera, sin los cuales no podría estar ahora finalizándola.

Álvaro Ángel Santiago Cuenca

Sevilla, 2024

Resumen

En trabajos anteriores se ha ido desarrollando una plataforma de gamificación que permite complementar las herramientas de aprendizaje a distancia con actividades lúdicas que favorecen la implicación de los alumnos en el proceso de aprendizaje.

Este trabajo amplía el Proyecto Gamifica del departamento de Ingeniería Telemática de la ETSI de Sevilla. Anteriormente se habían realizado en otros proyectos: la web de administración (Ana Lobón), el desarrollo de partidas (Alberto Jiménez) y la comunicación LTI (Pedro García Frutos).

El primer paso en el proyecto ha sido actualizar la interfaz de la web de administración a tecnología JSF, cuyos aspectos más destacados son la construcción de páginas web usando un lenguaje de plantillas, la comunicación con Ajax desde el cliente al servidor y la validación de datos en el cliente, y se ha desarrollado el acceso a esta web mediante LTI por parte de los profesores.

Como segundo requisito se solicitaba la migración y unificación de las bases de datos del proyecto, para ello se ha realizado la migración de la base de datos existente en SQLite a la actual en PostgreSQL mejorando con ello la gestión y accesibilidad a la misma.

Otro de los requisitos del proyecto es la inclusión de las partidas distribuidas, en las que cada jugador visualiza el juego en su terminal de manera que pueden visualizar la ejecución de la partida desde su propio navegador.

Por ultimo se añade la funcionalidad de un chat en tiempo real entre los participantes en la partida, de manera que tanto el profesor como los alumnos puedan comunicarse entre sí, ya sea entre los miembros de un equipo, entre dos alumnos o desde el profesor a los alumnos o sus equipos.

Para verificar el correcto funcionamiento de todas las funcionalidades desarrolladas, en un entorno real, se ha utilizado la plataforma Moodle del IES Joaquín Turina de Sevilla, facilitado por Jesús Muñoz Calle.

Abstract

In previous works, a gamification platform has been developed to complement distance learning tools with recreational activities that encourage student involvement in the learning process.

This work extends the Gamifica Project of the Department of Telematics Engineering of the ETSI of Seville. Previously, the following projects had been carried out: the administration website (Ana Lobón), game development (Alberto Jiménez) and LTI communication (Pedro García Frutos).

The first step in the project was to update the interface of the administration website to JSF technology, the most important aspects of which are the construction of web pages using a template language, communication with Ajax from the client to the server and data validation on the client, and access to this website via LTI by teachers has been developed.

The second requirement was the migration and unification of the project's databases, for which the migration of the existing database in SQLite to the current one in PostgreSQL has been carried out, thus improving its management and accessibility.

Another of the requirements of the project is the inclusion of distributed games, in which each player visualises the game on their terminal so that they can view the execution of the game from their own browser.

Finally, the functionality of a real-time chat between the participants in the game is added, so that both the teacher and the students can communicate with each other, either between the members of a team, between two students or from the teacher to the students or their teams.

To verify the correct functioning of all the developed functionalities, in a real environment, the Moodle platform of the IES Joaquín Turina of Seville, provided by Jesús Muñoz Calle, has been used.

Índice

Agradecimientos.....	ix
Resumen.....	xi
Abstract.....	xiii
Índice.....	xv
Índice de Tablas.....	xvii
Índice de Figuras.....	xix
1 Introducción.....	1
1.1. Motivación.....	1
1.2. Gamificación.....	1
1.1.1 Ventajas de usar la gamificación en el aula.....	2
1.3. Objetivos.....	2
1.4. Tecnología WebSocket.....	3
1.5. Tecnologías Web.....	4
1.6. Alcance.....	6
2 Estado del arte.....	9
2.1 Herramientas LMS/LTI.....	9
2.1.1 Introducción a las herramientas LMS.....	9
2.1.2 Herramientas LTI.....	10
2.1.3 Integración de herramientas LTI.....	10
2.1.4 Arquitectura LTI.....	11
2.1.5 Bibliotecas y Ejemplos LTI.....	14
2.2 Jakarta EE.....	14
2.3 Ajax.....	15
2.4 WebSocket.....	15
2.5 ADJA.....	15
2.5.1 Juegos.....	15
2.5.2 Equipos y grupos.....	15
2.5.3 Configuración de partida.....	16
2.5.4 Partida.....	16
3 Diseño.....	17
3.1 Introducción.....	17
3.2 Análisis de requisitos y especificaciones.....	17
3.2.1 Requisitos generales.....	18
3.2.2 Requisitos funcionales.....	22
3.2.3 Requisitos no funcionales.....	36
3.3 Arquitectura y funcionamiento básico de la aplicación.....	38
3.3.1 Arquitectura de capas de la aplicación.....	38
3.3.2 Arquitectura de componentes.....	40
3.3.3 Funcionamiento básico de la aplicación.....	41
3.3.4 Base de Datos.....	46

4	Implementación.....	55
4.1	Código.....	55
4.1.1	Bibliotecas utilizadas.....	55
4.1.2	Clases.....	58
4.1.3	Ficheros JSF.....	62
4.1.4	Ficheros localización – I10n.....	90
4.1.5	Hojas de estilo (CSS).....	90
4.1.6	Javascript (JS).....	90
4.1.7	Mensajes WebSocket y LTI.....	91
4.1.8	Envío calificaciones al LMS.....	123
4.1.9	Configuración.....	123
4.1.10	config.properties.....	123
4.1.11	Descriptores.....	123
4.2	Otras funcionalidades.....	124
4.2.1	Autenticación LTI.....	124
5	Pruebas de ejecución.....	127
5.1	Entorno LMS utilizado.....	127
5.2	Plataforma Gamificación.....	127
5.3	Pruebas de funcionamiento realizadas y otros datos de interés.....	127
5.4	Pruebas de Rendimiento.....	129
5.5	Pruebas de Seguridad.....	129
6	Conclusiones y Líneas de Futuro.....	131
6.1	Conclusiones.....	131
6.2	Líneas de Futuro.....	132
7	Anexos.....	133
7.1	Anexo A Manual de instalación.....	133
7.1.1	Preparación del entorno.....	133
7.1.2	Fichero de configuración “config.properties”.....	133
7.1.3	Configuración Eclipse y WildFly.....	133
7.1.4	Creación de base de datos.....	133
7.2	Anexo B Manual de usuario.....	134
	Referencias.....	139

ÍNDICE DE TABLAS

Tabla 1. Caracterización de los requisitos del Sistema.....	17
Tabla 2. Requisito general 01 “Sistema de gestión de la plataforma de gamificación”	18
Tabla 3. Requisito general 02 “Acceso al módulo de administración mediante Login”.....	18
Tabla 4. Requisito general 03 “Acceso LTI al módulo de administración ”	18
Tabla 5. Requisito general 04 “Acceso LTI a la partidas”.....	19
Tabla 6. Requisito general 05 “Entorno Gestión JSF”	19
Tabla 7. Requisito general 06 “Implementación Partidas Multijugador”.....	19
Tabla 8. Requisito general 07 “Implementación Partidas distribuidas”.....	20
Tabla 9. Requisito general 08 “Implementación espectadores en las partidas”	20
Tabla 10. Requisito general 09 “Versión LTI”.....	20
Tabla 11. Requisito general 10 “Migración Base de datos”.....	21
<i>Tabla 12. Requisito general 11 “Configuración de la aplicación en base de datos”</i>	21
Tabla 13: Requisito general 12 “Control de acceso mediante roles”	21
Tabla 14. Caso de uso 01 “Iniciar sesión”.....	22
Tabla 15: Caso de uso 02 “Consulta de la configuración de la aplicación”	22
Tabla 16: Caso de uso 03 “Modificación de la configuración de la aplicación”	23
Tabla 17: Caso de uso 04 “Consulta de consumidores LTI”.....	23
Tabla 18: Caso de uso 05 “Registro de un consumidor LTI”.....	23
Tabla 19: Caso de uso 06 “Borrado de un consumidor LTI”.....	24
Tabla 20: Caso de uso 07 “Modificación de un consumidor LTI”.....	24
Tabla 21: Caso de uso 08 “Consulta de usuarios”.....	24
Tabla 22: Caso de uso 09 “Registro de un usuario no LTI”.....	25
Tabla 23: Caso de uso 10 “Borrado de un usuario no LTI”.....	25
Tabla 24: Caso de uso 11 “Modificación de un usuario no LTI”	25
Tabla 25: Caso de uso 12 “Consulta de juegos”	26
Tabla 26: Caso de uso 13 “Registro de un juego”.....	26
Tabla 27: Caso de uso 14 “Borrado de un juego”.....	26
Tabla 28: Caso de uso 15 “Modificación de un juego”	27
Tabla 29: Caso de uso 16 “Consulta de ficheros de preguntas”	27
Tabla 30: Caso de uso 17 “Registro de un fichero de preguntas”	27
Tabla 31: Caso de uso 18 “Borrado de un fichero de preguntas”	28
Tabla 32: Caso de uso 19 “Modificación de un fichero de preguntas”	28
Tabla 33: Caso de uso 20 “Consulta de configuraciones de partida”	28

Tabla 34: Caso de uso 21 “Registro de una configuración de partida”.....	29
Tabla 35: Caso de uso 22 “Borrado de una configuración de partida”	29
Tabla 36: Caso de uso 23 “Modificación de una configuración de partida”	29
Tabla 37: Caso de uso 24 “Consulta de partidas”	30
Tabla 38: Caso de uso 25 “Registro de una partida”.....	30
Tabla 39: Caso de uso 26 “Borrado de una partida”.....	30
Tabla 40: Caso de uso 27 “Modificación de una partida”.....	31
Tabla 41: Caso de uso 28 “Acceso del iniciador de partida multijugador”	31
Tabla 42: Caso de uso 29 “Jugar en una partida multijugador”.....	31
Tabla 43: Caso de uso 30 “Seleccionar equipo y grupo”	32
Tabla 44: Caso de uso 31 “Cambio de equipo y grupo”	32
Tabla 45: Caso de uso 32 “Espectador en una partida multijugador”	32
Tabla 46: Caso de uso 33 “Jugar en una partida distribuida multijugador”	33
Tabla 47: Caso de uso 34 “Espectador en una partida distribuida multijugador”	33
<i>Tabla 48: Caso de uso 35 “Enviar mensaje”</i>	33
Tabla 49: Caso de uso 36 “Recibir mensajes”	34
Tabla 50: Regla de negocio 01 “Administración con acceso por LTI”.....	34
Tabla 51: Regla de negocio 02 “Envío calificaciones LTI”.....	35
Tabla 52: Requisitos de conducta 01 “Reconexión de un jugador”	35
Tabla 53: Requisitos de conducta 02 “Reconexión del iniciador”	35
Tabla 54: Requisito de información 01: Sistema de gestión de la plataforma de gamificación.....	36
Tabla 55: Requisito de usabilidad 01: Entrega de mensajes del chat.....	36
Tabla 56: Requisito de portabilidad 01: Aplicación multiplataforma.....	36
Tabla 57: Requisito de seguridad 01: Control del acceso a los recursos.....	37

ÍNDICE DE FIGURAS

Figura 1: Establecimiento y cierre de conexión WebSocket.....	3
Figura 2: Comparativa HTTP - WebSocket.....	4
Figura 3: Cuota de Mercado Herramientas LMS en Europa.....	10
Figura 4: Ejemplo conexión LTI Usuario LMS.....	12
Figura 5: Diagrama de comunicación entre las capas de la aplicación.....	39
Figura 6: Diagrama de componentes.....	40
Figura 7: Estructura del menú de la aplicación.....	42
Figura 8: Añadir consumidor.....	42
Figura 9: Añadir juego.....	43
Figura 10: Añadir fichero de preguntas.....	43
Figura 11: Añadir configuración de partida.....	44
Figura 12: Añadir partida.....	45
Figura 13: Estructura de la base de datos.....	46
Figura 14: Inicio de Sesión.....	63
Figura 15: Inicio de Sesión Correcto.....	63
Figura 16: Error en inicio de Sesión.....	64
Figura 17: Preferencias.....	64
Figura 18: Estructura del menú de administración.....	65
Figura 19: Lista de propiedades.....	65
Figura 20: Editar Propiedades.....	66
Figura 21: Ver Propiedad.....	66
Figura 22: Menú desplegable Juegos.....	67
Figura 23: Lista de ficheros de preguntas.....	67
Figura 24: Añadir Fichero de preguntas.....	68
Figura 25: Editar Fichero de Preguntas.....	68
Figura 26: Ver Fichero de Preguntas.....	69
Figura 27: Lista de Juegos.....	69
Figura 28: Añadir Juego.....	70
Figura 29: Editar Juego.....	71
Figura 30: Ver Juego.....	72
Figura 31: Lista Configuraciones de Partidas.....	73
Figura 32: Añadir Configuración de Partida.....	73
Figura 33: Editar Configuración de Partida.....	74

Figura 34: Ver Configuración de Partida.....	75
Figura 35: Lista de Partidas.....	75
Figura 36: Añadir Partida.....	76
Figura 37: Editar Partida.....	77
Figura 38: Ver Partida.....	78
Figura 39: Lista de Consumidores.....	79
Figura 40: Añadir Consumidor.....	79
Figura 41: Editar Consumidor.....	80
Figura 42: Ver Consumidor.....	81
Figura 43: Lista de usuarios.....	82
Figura 44: Añadir Usuario.....	82
Figura 45: Editar Usuario.....	83
Figura 46: Ver Usuario.....	83
Figura 47: Pantalla inicial Iniciador Partida Multijugador.....	84
Figura 48: Pantalla Iniciador Partida Multijugador.....	84
Figura 49: Pantalla Inicial Alumno/Jugador Partida Multijugador.....	85
Figura 50: Pantalla Panel de respuestas Alumno/Jugador Partida Multijugador.....	85
Figura 51: Pantalla resultado Pregunta Alumno/Jugador en Partida Multijugador.....	86
Figura 52: Pantalla Inicial Espectador Partida Multijugador.....	86
Figura 53: Pantalla Seguimiento Espectador Partida Multijugador I.....	87
Figura 54: Pantalla Seguimiento Espectador Partida Multijugador II.....	87
Figura 55: Pantallas Jugadores Partida Distribuida I.....	87
Figura 56: Pantallas Jugadores Partida Distribuida II.....	88
Figura 57: Pantalla Espectador Partida Distribuida.....	88
Figura 58: Ejemplo chat común.....	89
Figura 59: Ejemplo mensaje Alumno-Profesor.....	89
Figura 60: Ejemplo mensaje Profesor-Alumno.....	89
Figura 61: Ejemplo mensaje Equipos.....	90
Figura 62: Registro partida.....	92
Figura 63: Conexión jugador alva-02.....	93
Figura 64: Conexión jugador alva-01.....	94
Figura 65: Conexión jugador alva-03.....	95
Figura 66: Conexión espectador alva-06.....	96
Figura 67: Conexión jugador alva-05.....	97
Figura 68: Conexión jugador alva-04.....	98
Figura 69: Cambio grupo y equipo.....	99
Figura 70: Inicio de la partida.....	100
Figura 71: Pregunta grupo A.....	102
Figura 72: Respuesta grupo A (I).....	103

Figura 73: Respuesta grupo A (II).....	104
Figura 74: Pregunta grupo B.....	105
Figura 75: Respuesta grupo B.....	106
Figura 76: Final de la partida.....	107
Figura 77: Registro y conexión jugador alva-02.....	108
Figura 78: Conexión espectador alva-03.....	109
Figura 79: Registro y conexión jugador alva-04.....	110
Figura 80: Comenzar partida - jugador alva-02.....	112
Figura 81: Comenzar partida - espectador alva-03.....	113
Figura 82: Comenzar partida - jugador alva-04.....	114
Figura 83: Reconexión.....	115
Figura 84: Primera pregunta.....	116
Figura 85: Siguientes preguntas - jugador alva-04.....	117
Figura 86: Siguientes preguntas - jugador alva-02.....	118
Figura 87: Final partida - jugador alva-04.....	119
Figura 88: Final partida – espectador alva-03.....	120
Figura 89: Final partida - jugador alva-02.....	121
Figura 90: Envío mensaje de chat.....	122
Figura 91: Envío calificaciones al LMS.....	123
Figura 92: Contenido del curso de Moodle para realizar las pruebas.....	128
Figura 93: Moodle - Activar Edición.....	134
Figura 94: Moodle - Añadir una actividad o un recurso.....	134
Figura 95: Moodle - Añadir Herramienta Externa.....	135
Figura 96: Moodle - Configuración Herramienta Externa.....	135
Figura 97: Moodle - Configuración Herramienta Externa para acceso al módulo de Administración.....	136
Figura 98: Moodle - Configuración Herramienta Externa para acceso a la partida.....	137
Figura 99: Moodle - Configuración Herramienta Externa para acceso a la partida como espectador.....	137

1 INTRODUCCIÓN

1.1. Motivación

La motivación de este proyecto es fomentar el interés de los alumnos en el aprendizaje, haciéndolo más ameno y accesible. De manera que puedan realizar las evaluaciones de manera divertida en modo juego, ya sea de manera individual o como equipo, fomentando también así el compañerismo entre ellos y la cooperación para ayudarse unos a otros en el proceso de aprendizaje de la materia.

En una época además en la que los estudiantes están tan familiarizados con los videojuegos, adaptar la enseñanza y acercarla a una de sus formas de ocio facilita que se motiven en ello y puedan aprender de forma más cercana [1].

La Gamificación ha sido objeto de estudio y desarrollo por parte del departamento de Ingeniería Telemática durante varios años con el Proyecto Gamifica. Por ello, este proyecto continua el trabajo realizado en TFGs anteriores en los que se desarrollaron la web de administración (Ana Lobón) [2], el desarrollo de partidas (Alberto Jiménez) [3] y la comunicación LTI (Pedro García Frutos) [4].

Partiendo de ellos, se ha realizado la actualización a JSF de la web de administración. Como veremos más adelante JSF nos permite la construcción de las páginas basada en plantillas, lo cual facilita que otros desarrolladores puedan continuar este proyecto en el futuro.

Para poder aprovechar todo el potencial que nos permite tanto LTI como la propia Gamificación, se ha implementado la posibilidad de realizar partidas distribuidas, las cuales nos permiten que cada usuario tenga en su propio navegador todo lo necesario para poder jugar con otro de manera remota.

A raíz de lo anterior, ya que cada usuario puede estar jugando desde una ubicación distinta, se implementa la funcionalidad de Chat entre usuarios, de forma que puedan compartir la información que necesiten para el desarrollo de la partida.

Los juegos que se usan en este TFG son propiedad de Jesús Muñoz Calle, y están accesibles en la web del Proyecto ADJA [5].

1.2. Gamificación

La Gamificación como técnica permite trasladar la capacidad de jugabilidad de los juegos al ámbito educativo de forma que permite a los alumnos mejorar sus resultados de forma atractiva. Para ello, es imprescindible que los alumnos previamente tengan conocimiento de la dinámica y normas de los juegos con los que se va a realizar la Gamificación en el aula, para poder conseguir una mayor implicación y por lo tanto una mejora de los resultados.

Mediante la Gamificación buscamos que por medio de la experimentación y los juegos, al ser una experiencia más divertida para el alumno, se absorban mejor los conocimientos, potenciando las habilidades de los alumnos.

Los principales objetivos de este método de aprendizaje son:

- Crear un vínculo lúdico entre el alumno y el contenido de que se imparte.
- Ser una herramienta motivadora y efectiva que llame la atención del alumno, de forma que luche contra el aburrimiento y les impulse en el aprendizaje.
- Premiar al alumno, sobre todo en aquellos casos donde no existe más incentivo que el propio aprendizaje.

- Mejorar los resultados de la enseñanza.

El factor diferencial que se le aporta en este proyecto con nuestra herramienta a la Gamificación, respecto a otras herramientas existentes, es la capacidad de integración de un gran numero de juegos en múltiples de plataformas de enseñanza virtual, de manera que toda la información y configuración esta centralizada e integrada en un solo lugar [1].

1.1.1 Ventajas de usar la gamificación en el aula

La Gamificación presenta el proceso de enseñanza de forma más atractiva a la tradicional. Es capaz de presentar como lúdico un proceso que habitualmente no se percibe como tal, como por ejemplo el aprendizaje de las matemáticas. Se puede decir que siempre se aprenderá mejor si se aprende jugando.

Una de las grandes ventajas es que se puede adaptar a niveles educativos y materias muy diferentes. Puede utilizarse desde etapa infantil hasta una etapa Post Obligatoria.

Con esta técnica se permite emplear diversos recursos y herramientas en el aula que ayudarán a los docentes a motivar a los alumnos. Pudiendo personalizar actividades y contenidos en función de las necesidades de cada estudiante, favoreciendo así la adquisición de conocimientos y la mejora de la atención [1].

1.3. Objetivos

El objetivo principal es permitir desde plataformas LMS, el uso de Juegos de Gamificación basados en LTI permitiendo partidas multijugador y distribuidas.

Para ello se hace uso de una aplicación web desde la que se gestiona todo el proceso de creación de partidas. Esta aplicación permite un acceso limitado cuando se identifica el rol de profesor a través de LTI, accediendo desde su plataforma LMS.

Esta aplicación web ha sido diseñada para permitir al profesor de una forma fácil e intuitiva poder crear partidas, seleccionando los juegos así como los ficheros de preguntas que quieran aplicar en cada Juego.

Desde la plataforma además se ofrece un menú adaptado según la plataforma LMS que se este empleando, esto permite que pueda ser utilizado por multitud de profesores permitiendo la interoperabilidad de la aplicación.

Los objetivos fundamentales que se persiguen con la implantación de esta aplicación son los siguientes:

1. Migración de las páginas de administración LTI de tecnología JSP a JSF (*Facelets*) [6].
2. Migración de la base de datos en SQLite a PostgreSQL.
3. Implementación de partidas distribuidas en las que cada jugador puede interactuar con el juego en su propio navegador, encargándose la aplicación de la coordinación entre los distintos jugadores.
4. Implementación de un chat entre los participantes de una partida, de forma que puedan enviarse mensajes entre ellos en tiempo real, con la posibilidad de escoger a quién va dirigido cada mensaje.
5. Permitir al usuario elegir el idioma en el que desea visualizar las páginas.

La aplicación desarrollada logra cumplir con todos estos objetivos, dando soporte a los idiomas español, inglés y francés.

1.4. Tecnología WebSocket

Una aplicación web básica hace uso de al menos dos elementos, un **navegador** (cliente) y un **Servidor** que procesa, almacena y distribuye la información en respuesta a las peticiones del navegador.

La tecnología WebSocket permite a las aplicaciones que la usan que el servidor inicie la comunicación con el navegador sin necesidad de una petición por parte del cliente.

Los protocolos que se usan para estas comunicaciones Navegador-Servidor son HTTP o HTTPS, ambos basados en TCP/IP.

Para las comunicaciones bidireccionales y *full-duplex* se emplea el protocolo WebSocket, descrito en la especificación RFC 6455 [7], el cual permite intercambiar datos entre el navegador y el servidor por medio de una conexión persistente. Los datos pueden ser pasados en ambas direcciones como paquetes “*packets*”, sin cortar la conexión y sin peticiones adicionales de HTTP “*HTTP-requests*”.

WebSocket es especialmente bueno para servicios que requieren intercambio de información continua, por ejemplo juegos en línea, sistemas de negocios en tiempo real, entre otros [8].

Con la conexión HTTP tradicional, el cliente realiza una solicitud y, después de que el servidor envía su respuesta, la conexión se cierra. Si el cliente necesita más datos, debe abrir una nueva conexión.

Con una conexión WebSocket, el cliente puede abrir y utilizar una única conexión para todas sus comunicaciones WebSocket con el servidor. Esta conexión persistente permite mensajes bidireccionales de baja latencia, es decir, las peticiones pueden tener origen en el servidor, no solo en el cliente.

Esta conexión WebSocket también puede ser *con estado*. Una conexión HTTP no tiene estado: esto significa que cada solicitud se maneja de forma aislada, sin retención de información sobre las solicitudes anteriores. WebSocket, por otro lado, tiene estado gracias a su conexión persistente.

Que una aplicación aproveche o no la capacidad con estado depende totalmente del desarrollador y de cómo utiliza su conexión WebSocket [9].

Establecimiento y cierre conexión WebSocket

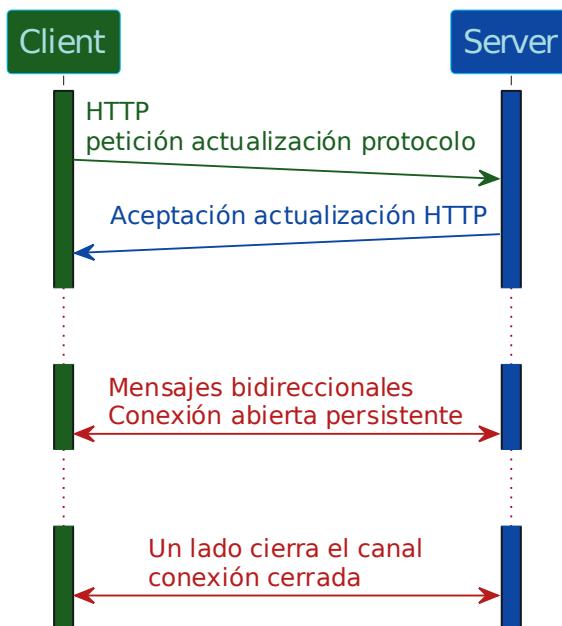


Figura 1: Establecimiento y cierre de conexión WebSocket

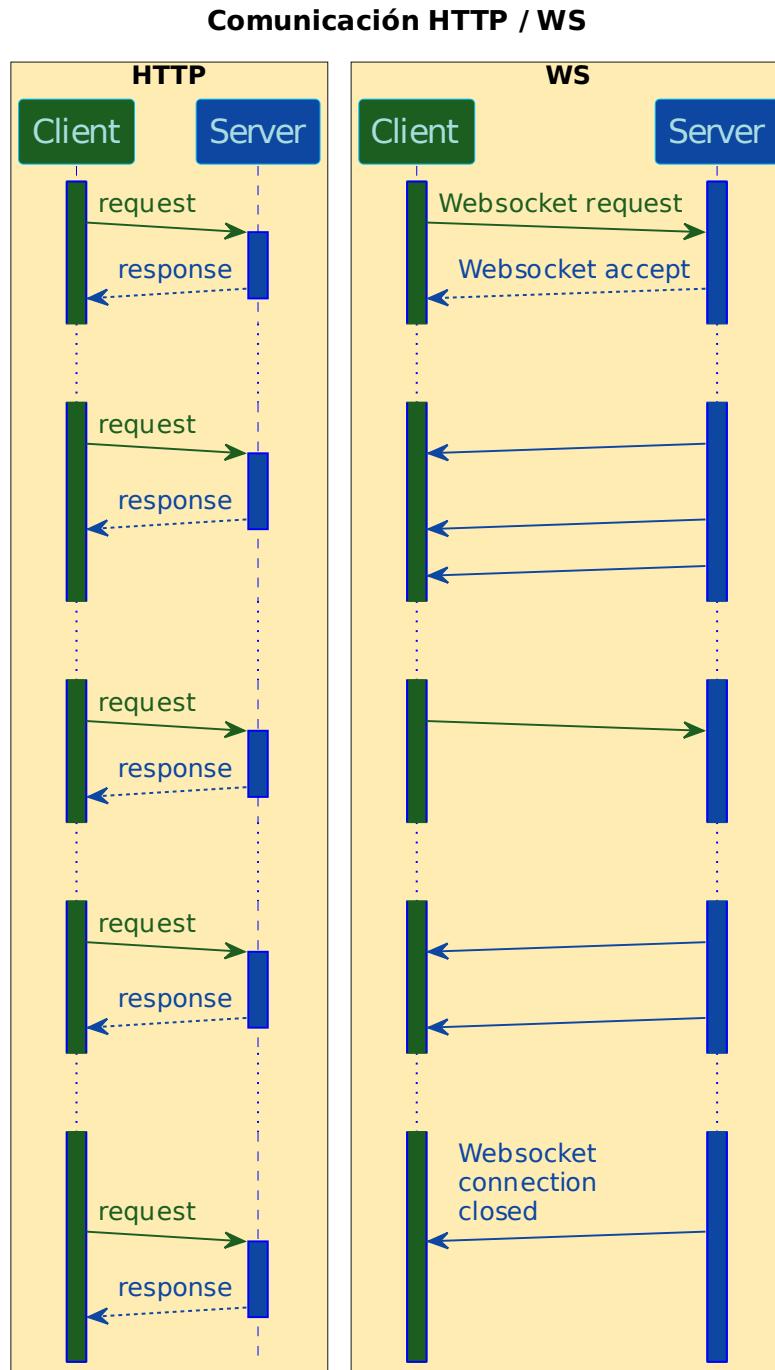


Figura 2: Comparativa HTTP - WebSocket

1.5. Tecnologías Web

Las **Tecnologías frontend o del lado del cliente** son las que se encargan de la creación de la interfaz de usuario y de establecer las comunicaciones con el servidor. Las más relevantes para nuestra aplicación son las siguientes:

- **JavaScript** (abreviado comúnmente JS) es un lenguaje de programación interpretado, dialecto del estándar ECMAScript [10]. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.
Se utiliza principalmente del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas y JavaScript del lado del servidor (*Server-side JavaScript* o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente *widgets*) es también significativo.
- **HTML**, lenguaje de marcado utilizado en la creación de páginas web. Este estándar [11], derivado de SGML [12], está definido por el W3C (World Wide Web Consortium) [13].
- **CSS** (*Cascading Style Sheets* [14], en español Hojas de estilo en cascada), es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Es muy usado para establecer el diseño visual de los documentos web, e interfaces de usuario escritas en HTML.
- **JSF (Facelets)** JavaServer Faces [6] es una tecnología y *framework* para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Jakarta EE (anteriormente llamado Java EE). Puesto que en el trabajo se ha utilizado Jakarta EE 8 [15], la versión de JSF utilizada es la 2.3.
- PrimeFaces [16] es una biblioteca de componentes para Java Server Faces (JSF) de código abierto que cuenta con un conjunto de componentes enriquecidos que facilitan la creación de las aplicaciones web.

Todas ellas utilizadas en el proyecto.

Tecnologías *back-end* o del lado del servidor, implementan la funcionalidad de la web en el servidor. Entre ellas, podemos distinguir las siguientes [17]:

- **PHP** [18] es un lenguaje de programación interpretado del lado del servidor y de uso general que se adapta especialmente al desarrollo web.
- **Python** [19] es un lenguaje de alto nivel de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código. Se trata de un lenguaje de programación multiparadigma, ya que soporta parcialmente la orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, dinámico y multiplataforma.
- **Java EE (Java Enterprise Edition), renombrado como Jakarta EE** [15], es una plataforma de programación para desarrollar y ejecutar *software* de aplicaciones en el lenguaje de programación Java. Permite utilizar arquitecturas de N capas distribuidas y se apoya ampliamente en componentes de *software* modulares ejecutándose sobre un servidor de aplicaciones.
- **C** [20], es un lenguaje de programación de propósito general originalmente desarrollado por Dennis Ritchie entre 1969 y 1972 en los Laboratorios Bell, como evolución del anterior lenguaje B, a su vez basado en BCPL.

Al igual que B, es un lenguaje orientado a la implementación de sistemas operativos: concretamente Unix. C es apreciado por la eficiencia del código que produce.

En los trabajos anteriores sobre los que se basa este se disponía de una base de datos en PostgreSQL [21] en la que se gestionaban los datos correspondientes a los juegos y partidas, y por otro lado una base de datos en SQLite [22] en la que se gestionaban los datos correspondientes a las conexiones LTI.

Respecto a la base de datos las principales alternativas son MySQL, SQLite, Oracle, H2, DB2, PostgreSQL... Se ha elegido PostgreSQL ya que uno de los requisitos es la migración a PostgreSQL de la base datos para la administración de LTI. Para ello se ha migrado a PostgreSQL la base de datos anterior en SQLite y se ha unificado con la existente en PostgreSQL. Esta unificación ha consistido en crear nuevas tablas en PostgreSQL correspondientes a las existentes en SQLite sin correspondencia en PostgreSQL, y en adaptar la estructura de las tablas en las que había correspondencia para incorporar los datos que no estaban contemplados en la base de datos PostgreSQL.

También se ha utilizado esta base de datos para el almacenamiento de la información relativa a la funcionalidad de un chat entre los participantes en una partida.

En lo relativo a las **Tecnologías frontend o del lado del cliente** se ha escogido JSF puesto que uno de los objetivos es la migración de las páginas de administración de las conexiones LTI a tecnología JSF. Además se utilizaba en algunos de los trabajos anteriores en los que se basa éste. Adicionalmente se utilizan CSS para el formato visual de las página y JavaScript para permitir interacciones más complejas desde el cliente.

Puesto que los trabajos anteriores en los que se basa este trabajo están desarrollados en Java y se va a usar JSF como tecnología para la construcción de las páginas web, la elección de la tecnología del lado del servidor ha sido usar el lenguaje de programación Java, utilizando la plataforma Jakarta EE en su versión 8. Además con Java tenemos un lenguaje de programación no excesivamente complejo, que dispone de gran cantidad de *software* libre y es fácilmente editable y portable.

1.6. Alcance

- Se explicarán las “**Herramientas LMS**” denominadas en la aplicación como consumidores, cómo se usan, para qué sirven y su estado actual.

Estas herramientas permiten el aprendizaje a distancia y asíncrono, ya que los profesores pueden definir en la misma los contenidos de su asignatura y las actividades a desarrollar, y los alumnos pueden acceder en cualquier momento dentro del período definido por el profesor.

El estándar LTI permite la interconexión automatizada de estas herramientas LMS con proveedores de herramientas destinadas al aprendizaje que podrán incorporarse en el proceso de aprendizaje.

En el apartado Herramientas LMS/LTI del capítulo Estado del arte se describe el funcionamiento de estas herramientas.

- Se desarrollará una aplicación que ampliará los trabajos anteriores en los siguientes aspectos:
 - Los profesores de las instituciones que usan la plataforma podrán acceder a las páginas de administración mediante LTI, lo que simplifica la administración de esta aplicación, puesto que no hay que crear manualmente cuentas para todos los profesores de la institución, sino simplemente definir un consumidor de forma que se acceda por LTI desde el LMS de la institución.
 - Permitirá no sólo que los jugadores contesten en su navegador a las respuestas de un juego que se está ejecutando en un único navegador, sino que podrán disputarse partidas distribuidas en las que cada jugador vea en su navegador el desarrollo de la partida, actualizándose automáticamente según intervengan otros jugadores.

Esto permite que no sólo puedan disputarse partidas en un aula en la que todos los jugadores pueden ver el juego en una misma pantalla, sino que podrán disputarse partidas no presenciales en las que los jugadores no estén en la misma ubicación.

- Permitirá la participación de espectadores en la partida. Los espectadores pueden ver el desarrollo de la partida, pero no responder a las preguntas, aunque ven las respuestas de los jugadores.
- Se incluye un chat entre los participantes en una partida, de forma que pueden comunicarse entre sí en tiempo real.
- Una vez finalizada la partida se enviará al LMS, en caso de jugadores conectados por LTI, la calificación obtenida mediante mensajes LTI al LMS correspondiente a cada uno de los jugadores.
- El diseño de la aplicación se detalla en el capítulo Diseño, y la descripción de la implementación de la misma está detallada en el capítulo Implementación.
- Las pruebas realizadas para comprobar el funcionamiento de la aplicación están relacionadas en el capítulo Pruebas de ejecución.
- Las conclusiones extraídas en la elaboración de este trabajo y las posibles líneas futuras de la aplicación aparecen en el apartado Conclusiones y Líneas de Futuro.
- Finalmente en los Anexos se dispone de instrucciones para:
 - La instalación de la aplicación, en el Anexo A Manual de instalación.
 - Uso de la aplicación, en el Anexo B Manual de usuario.
- El punto de partida desde el que se ha desarrollado esta aplicación son básicamente dos trabajos anteriores:
 - Un trabajo que implementa la comunicación LTI, permitiendo autenticar las conexiones, pero que usa un consumidor distinto para cada partida y no permite el acceso por LTI a las páginas de administración.

Esta aplicación no cuenta con la posibilidad de disputar partidas entre varios jugadores.

- En otro trabajo se implementan partidas multijugador en las que el servidor coordina el desarrollo de la partida entre el profesor que controla el transcurso de la partida y la aplicación de los jugadores con un panel de respuestas que se activa cuando les corresponde el turno.

2 ESTADO DEL ARTE

En este apartado se explicará inicialmente la información necesaria acerca de las herramientas LTI, para poder entender el alcance del proyecto, los usos y ventajas del mismo, continuaremos con ejemplo práctico incluyendo una explicación más detallada del mismo.

2.1 Herramientas LMS/LTI

2.1.1 Introducción a las herramientas LMS

Los LMS son herramientas muy versátiles que se adaptan a multitud de necesidades diferentes. Los principales casos de uso de los LMS son:

- Educación: Se emplean las plataformas LMS para impartir cursos en línea, ya que permite a los profesores administrar el material docente y realizar un seguimiento de los estudiantes.
- Formación corporativa: Permite a las empresas formar a sus empleados en cualquier materia que se requiera, desde habilidades específicas para su puesto de trabajo como las políticas internas de la empresa.
- Desarrollo profesional: Desde este tipo de plataformas se ofrecen cursos que permiten que los usuarios se actualicen o continúen su formación y puedan obtener certificaciones.
- Evaluación de competencias. Las plataformas LMS disponen de herramientas de evaluación lo que permite el seguimiento de lo aprendido.

Algunos de los LMS más destacados son **Blackboard Learn, Canvas y Moodle [23]**. Este ultimo es el que se ha empleado para realizar las pruebas.

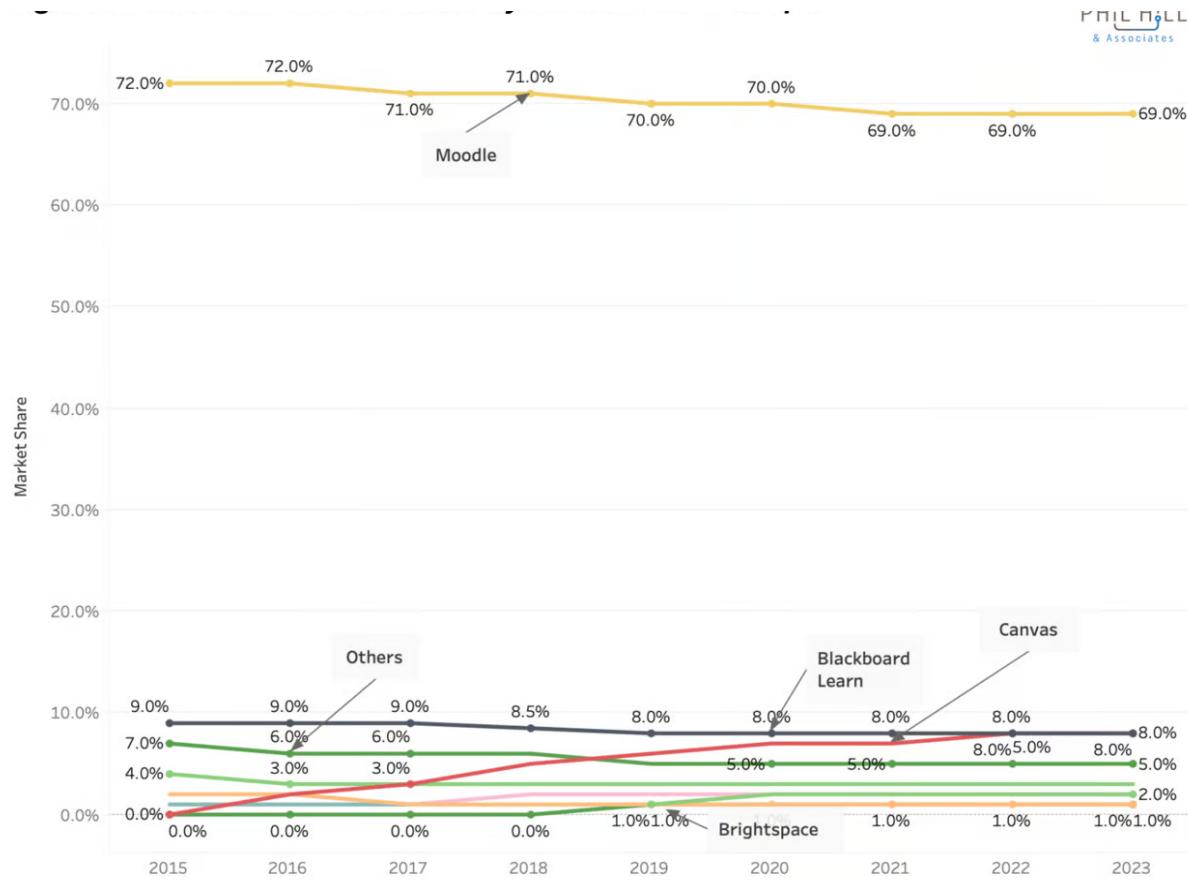


Figura 3: Cuota de Mercado Herramientas LMS en Europa

2.1.2 Herramientas LTI

LTI “*Learning Tools Interoperability*”, o traducido al castellano “*Interoperabilidad de las herramientas de aprendizaje*”, es un estándar que permite el intercambio de información entre un LMS y una herramienta de aprendizaje externa, como la desarrollada en este proyecto.

Otra de las características de LTI es que elimina la necesidad de que los usuarios necesiten credenciales de acceso de inicio de sesión para diferentes sitios web, ya que mediante LTI la herramienta externa puede verificar la identidad del usuario en la plataforma LMS.

2.1.3 Integración de herramientas LTI

Describiremos ahora el proceso de integración de herramientas LTI en plataformas LMS, para ello definiremos los conceptos de *Tool provider* (TP – proveedor de herramientas) y *Tool Consumer* (TC – consumidor de herramientas).

Tool Provider, son las herramientas externas que se integran en las plataformas LMS a través de LTI, como la aplicación desarrollada en este trabajo.

Tool Consumer, serán las plataformas LMS, el hecho de emplear un estándar como LTI nos permite integrar nuestro TP en multitud de TC.

El estándar LTI [24] es el que permite que se realice la comunicación entre el TP y el TC. Este estándar ha sido desarrollado por 1EdTech [25], anteriormente IMS Global Learning Consortium (IMS GLC) [26], y es una organización internacional sin ánimo de lucro que surgió en 1995 y persigue fomentar a nivel global el crecimiento y el impacto de las tecnologías del aprendizaje en los ámbitos educativos y de formación en empresas. Su principal actividad consiste en desarrollar estándares (normas) para la creación, difusión e interoperabilidad de Tecnologías Educativas para el Aprendizaje. Han conseguido aprobar y publicar más de 20 estándares gratuitos que pueden ser usados libremente sin restricciones por derechos de autor.

Según E-LEARN MAGAZINE [27] con la adopción de los estándares IMS, las organizaciones reducen costos y mejoran la integración de sistemas en un factor de 10 a 1000 en comparación con las interfaces de programación de aplicaciones personalizadas de conexión (API). Esto se debe a que todas las aplicaciones certificadas por IMS utilizan los mismos estándares abiertos para conectarse.

2.1.4 Arquitectura LTI

Para poder establecer una comunicación LTI, es necesario que en el TC se configuren las credenciales de dominio de un TP. Dichas credenciales se aplicarán a los enlaces LTI que se creen en el *Tool Consumer*.

Estas credenciales consisten en una clave y un secreto, que han debido ser creadas previamente por el superusuario del TP. Este par Clave-Secreto debe ser conocido tanto por el TC como por el TP y será usado por los mismos para firmar el contenido de los paquetes OAuth [28].

Nuestra aplicación utiliza la biblioteca Oscelot para gestionar la comunicación con los consumidores por LTI. Esta biblioteca, desarrollada por Stephen P. Vickers, encapsula el código necesario para que un proveedor de herramientas compatible con LTI 1 se comunique con un consumidor de herramientas. Incluye soporte para LTI 1.1 y las extensiones no oficiales de Basic LTI. Los beneficios de usar una librería de clases como esta son:

- la capa de abstracción proporcionada por las clases mantiene las comunicaciones LTI separadas del código de la aplicación.
- el código puede reutilizarse entre varios proveedores de herramientas.
- los datos LTI se transforman en objetos útiles y los datos que faltan se sustituyen automáticamente por valores predeterminados razonables.
- la función *Outcomes service* (servicio de resultados) utiliza LTI 1.1 o la extensión no oficial de resultados, según lo que admita el consumidor de la herramienta.
- se admiten las extensiones no oficiales para los servicios de afiliación y configuración.
- se incluye funcionalidad adicional para:
 - activar/desactivar una clave de consumidor.
 - establecer horas de inicio y fin para habilitar el acceso para cada clave de consumidor.
 - establecer acuerdos para que los usuarios de diferentes enlaces de recursos puedan colaborar juntos en un único enlace de proveedor de herramientas.
- los proveedores de herramientas pueden aprovechar las actualizaciones de LTI con un impacto mínimo en el código de sus aplicaciones.

Una vez configurados el TP y el TC un ejemplo de comunicación LTI [29] sería el mostrado en el siguiente diagrama de comunicación:

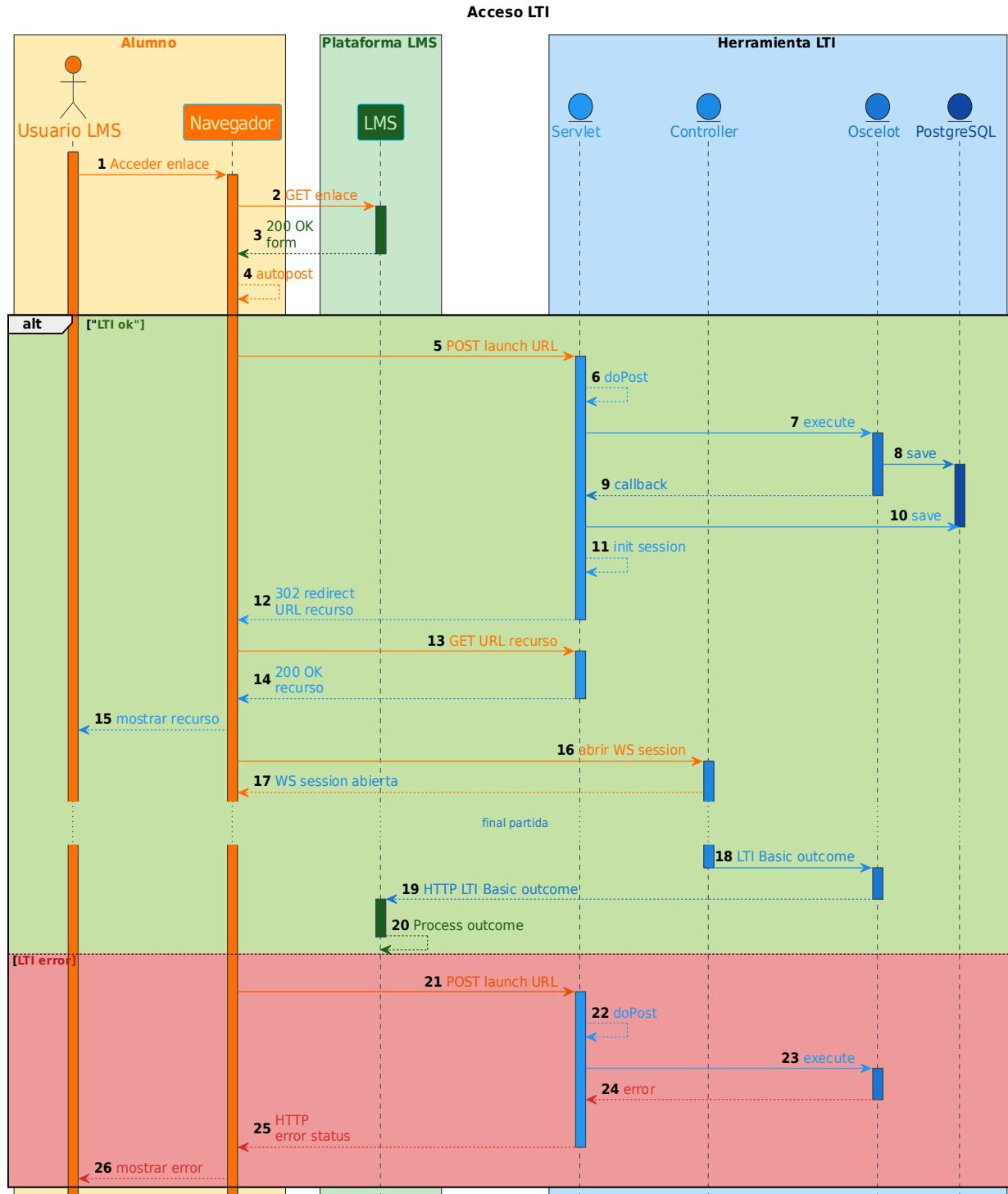


Figura 4: Ejemplo conexión LTI Usuario LMS

El paso a paso de la comunicación LTI mostrada en el diagrama anterior sería la siguiente:

1. Un usuario pincha en un enlace LTI alojado en el TC.
2. El navegador realiza una petición GET HTTP o HTTPS al LMS.
3. El LMS envía respuesta 200 OK con un formulario con los datos necesarios para acceder al TP.
4. El navegador hace automáticamente una petición POST.

En el caso de que la petición sea válida:

5. Se envía la petición al TP.
6. Se ejecuta el método `doPost` en nuestro TP.
7. Desde este método se invoca al método `execute` de la biblioteca LTI Oscelot, pasando como parámetro *callback* nuestro método `execute`. Para comprobar la validez de una petición LTI, tanto en lo relativo a la firma de los datos como a no caducidad y no duplicidad de la misma se invoca al método `execute` de esta biblioteca al que se le pasa como referencia un método de nuestra aplicación que se ejecutará en caso de la petición sea correcta. En este caso la propia biblioteca actualiza la información correspondiente en la base de datos.
8. Se comprueba que la petición es válida y se actualiza la base de datos con la información recibida: consumidor, usuario, *nonce* (número arbitrario que garantiza que la petición no está repetida ni caducada).
9. El método `execute` ejecuta como *callback* el método `execute` de nuestro *servlet*.
10. Desde el *servlet* se actualiza en la base de datos información adicional no contemplada en la biblioteca Oscelot.
11. El TP inicia una sesión HTTP para este cliente.
12. Una vez iniciada la sesión el TP envía una respuesta 302 REDIRECT al navegador para acceder al recurso solicitado.
13. El navegador realiza una petición GET para acceder al recurso en el TP.
14. El TP envía la respuesta a la petición del navegador (200 OK).
15. El navegador muestra el recurso al usuario.
16. El navegador envía una petición de apertura de comunicación WebSocket.
17. El TP completa el establecimiento de la sesión WebSocket.
18. Una vez finalizada la partida, nuestro TP envía, si procede, un mensaje a la biblioteca Oscelot con la calificación obtenida por los jugadores.
19. La biblioteca Oscelot envía un mensaje LTI de tipo *Basic outcome* al LMS para actualizar la calificación del jugador.
20. El LMS procesa el mensaje y almacena la calificación.

En caso de que haya un error en la petición:

21. Se envía la petición del punto 4 al TP.
22. Se ejecuta el método `doPost` en nuestro TP.
23. Desde este método se invoca al método `execute` de la biblioteca LTI Oscelot, pasando como parámetro *callback* nuestro método `execute`.
24. El método `execute` de la biblioteca Oscelot indica que la petición no es válida.
25. El *servlet* envía al navegador la respuesta a la petición (21) con un código de error.

26. El navegador muestra al usuario información del error.

2.1.5 Bibliotecas y Ejemplos LTI

La documentación para el uso del estándar LTI en sus distintas versiones se puede encontrar en la página de IMS (1edtech) [30].

Para la aplicación desarrollada en este proyecto el intercambio de mensajes entre TC y TP se realiza empleando el código de la biblioteca Oscelot. Este código esta programado en Java, soporta hasta la versión LTI 1.1 y además proporciona un código de ejemplo utilizando el servicio *Outcomes*.

2.2 Jakarta EE

Para el desarrollo de la aplicación se ha utilizado fundamentalmente la plataforma Jakarta EE, que es el nombre de las nuevas versiones de Java EE, una vez que Oracle cedió el control de la misma a la Eclipse Foundation.

Jakarta EE consiste en un conjunto de especificaciones para los distintos aspectos de una aplicación. Las principales especificaciones utilizadas en el desarrollo de esta aplicación son las siguientes:

1. *Jakarta Expression Language 3.0 (EL)* [31]

El *Unified Expression Language* de Java es un lenguaje de programación de propósito especial utilizado principalmente en aplicaciones web en Java para incrustar expresiones en páginas web.

2. *Jakarta Server Faces 2.3 (JSF)* [6]

Jakarta Server Faces es una tecnología y *framework* para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Jakarta EE.

3. *Jakarta Managed Beans 1.0* [32]

Esta especificación define un conjunto de servicios básicos para objetos gestionados por el contenedor con requisitos mínimos, también conocidos bajo el acrónimo "POJOs" (*Plain Old Java Objects*). Se basa en la especificación JavaBeans [33].

4. *Jakarta Contexts and Dependency Injection 2.0 (CDI)* [34]

La inyección de dependencias (en inglés *Dependency Injection*, DI) es un patrón de diseño orientado a objetos, en el que se suministran objetos a una clase en lugar de ser la propia clase la que cree dichos objetos.

Los objetos que pueden ser inyectados son *Managed Beans* definidos con el ámbito declarado en su clase, y se inyecta en cada caso el objeto que este en el ámbito al que tiene acceso el objeto que lo inyecta. En caso de no existir el objeto en ese ámbito, se crea un nuevo objeto previamente a la inyección.

En CDI se definen cinco tipos de ámbito estándar.

- *Application*, ámbito común a toda la aplicación.
- *Session*, corresponde a la sesión del usuario, en caso de que exista.
- *Request*, corresponde al transcurso del procesamiento de la respuesta a una petición.
- *Conversation*, que representa el ámbito de conversación definido en *Conversation context lifecycle*.
- *Dependent*, pseudoámbito para objetos dependientes.

5. Jakarta Persistence 2.2 (JPA) [35]

Jakarta Persistence una API de persistencia desarrollada para la plataforma Jakarta EE, que permite manejar datos relacionales en aplicaciones desarrolladas para esta plataforma.

La persistencia proporcionada cubre tres áreas:

- La API en sí misma, definida en el paquete `javax.persistence`.
- El lenguaje de consulta Java Persistence Query Language (JPQL).
- Metadatos objeto/relacional.

El objetivo que persigue el diseño de esta API es no perder las ventajas de la orientación a objetos al interactuar con una base de datos (siguiendo el patrón de mapeo objeto-relacional) y permitir usar objetos regulares (conocidos como POJO).

2.3 Ajax

Se trata de una técnica de desarrollo web para crear aplicaciones web asíncronas. La aplicación que se ejecuta en el cliente puede establecer una conexión asíncrona con el servidor en segundo plano, de forma que se interactúa con el servidor sin necesidad de recargar la página o recargando únicamente determinados elementos de la misma.

Una vez que se recibe la respuesta del servidor, la aplicación javascript que se ejecuta en el navegador puede modificar la página utilizando para ello la información recibida del servidor.

2.4 WebSocket

WebSocket es una tecnología que proporciona un canal de comunicación bidireccional y *full-duplex* sobre una única conexión HTTP.

El protocolo está definido por parte del IETF (*Internet Engineering Task Force*) [36] en el RFC 6455 [7]. Al ser *full-duplex* permite a los servidores web iniciar la transmisión de información a los navegadores sin necesidad de una petición previa por parte de éstos.

2.5 ADJA

El Proyecto "Aplicación de Juegos Didácticos en el Aula" AJDA [5], es un proyecto abierto y sin ánimo de lucro, perteneciente a la Red Educativa Digital Descartes, cuyo principal objetivo es ofrecer a la comunidad educativa más de 500 juegos didácticos y los recursos necesarios para su puesta en práctica en el aula.

Se ofrecen juegos que son los que se podrán usar en nuestra aplicación. El número de jugadores que pueden participar en una partida depende de cómo se haya desarrollado el juego que se usa en la misma.

2.5.1 Juegos

Se trata de juegos interactivos que se ejecutan en la ventana de un navegador web. Los jugadores tienen que responder las preguntas que se les van planteando durante el transcurso del juego. Algunos ejemplos de los juegos disponibles son: artificieros, batalla naval y bombilla.

2.5.2 Equipos y grupos

En las partidas multijugador los jugadores se organizan en grupos, los cuales a su vez se organizan en equipos.

Las preguntas se realizan a los miembros de un determinado grupo, y mediante la aplicación de un algoritmo a las respuestas de los integrantes del grupo, se determina si la respuesta del grupo es correcta o no.

Desde el punto de vista del juego, cada grupo es lo que el juego considera un jugador, corresponde a la aplicación una vez que termina la partida asignar la calificación obtenida por el grupo a cada uno de los integrantes del mismo.

2.5.3 Configuración de partida

Se trata de un “molde” de partida que puede estar incompleto y a partir del cual se pueden definir las partidas.

2.5.4 Partida

Una partida permite que uno o varios jugadores participen en uno de los juegos existentes en nuestro TP. Los parámetros son los definitivos para jugar a partir –modificando lo deseado- de una ‘Configuración de Partida’.

En función de los parámetros de la partida, estas se pueden clasificar en:

- Monojugador

Este tipo de partidas se desarrollan por cada jugador individualmente, por un único jugador. Aunque una partida monojugador puede ser ejecutada independientemente por varios alumnos, de forma que nunca participa más de un jugador.

- Multijugador

Para este tipo de partidas se necesitará la intervención de un iniciador y dos o más jugadores que participen en la misma. Se dispone también de la modalidad de Espectador para este tipo de partidas.

Este tipo de partidas esta pensado para realizarlas en el aula, donde el profesor actuará de iniciador, mostrando su pantalla con el desarrollo de la partida y los alumnos podrán ir respondiendo cada uno desde su panel de respuestas.

- Multijugador distribuida

En estas partidas participan también varios jugadores, pero cada uno dispone de una copia del juego y no existe iniciador. La partida comienza automáticamente cuando se alcanza el número de jugadores necesario y la misión de la aplicación es coordinar las distintas copias del juego para que funcionen coordinadamente mediante el procesado de los mensajes que se van recibiendo del juego.

3 DISEÑO

En este apartado se describirá el *software* desarrollado en este trabajo, así como los objetivos y funcionalidades.

3.1 Introducción

El objetivo de diseño de la aplicación es la actualización a tecnología JSF de la página de administración de la plataforma, habilitar el acceso por LTI desde el LMS (ver Figura 4) de los profesores para administrar las partidas sin necesidad de creación de usuario en la misma, la posibilidad de partidas multijugador LTI y distribuidas.

Para mejorar la gestión de datos del proyecto se ha realizado la migración de la base de datos existente en SQLite a PostgreSQL.

También se ha desarrollado la funcionalidad de un Chat entre los distintos usuarios de una partida.

3.2 Análisis de requisitos y especificaciones

En este apartado vamos a realizar un análisis pormenorizado de los requisitos que debe cumplir nuestra aplicación. El formato es el descrito a continuación:

Tabla 1. Caracterización de los requisitos del Sistema

IDENTIFICADOR	Nombre del requisito
Dependencias	
Descripción	
Importancia	
Prioridad	

- **Dependencias:** Nombre de el/los requisitos que son necesarios para la implementación de este requisito o que están íntimamente relacionados.
- **Descripción:** Explicación breve del objetivo a alcanzar con ese requisito.
- **Importancia:** Indica el grado de relevancia que tiene el requisito para el cliente.
- **Prioridad:** Indica el grado de relevancia que tiene el requisito para el desarrollador.

3.2.1 Requisitos generales

Tabla 2. Requisito general 01 “Sistema de gestión de la plataforma de gamificación”

RG-01	Sistema de gestión de la plataforma de gamificación
Dependencias	
Descripción	Acceso a las opciones de gestión y configuración de la plataforma.
Importancia	Alta
Prioridad	Alta

Tabla 3. Requisito general 02 “Acceso al módulo de administración mediante Login”

RG-02	Acceso a la administración mediante Login
Dependencias	RG-01
Descripción	Acceso a la plataforma de Administración mediante login con usuario y contraseña.
Importancia	Alta
Prioridad	Alta

Tabla 4. Requisito general 03 “Acceso LTI al módulo de administración”

RG-03	Acceso LTI al módulo de administración
Dependencias	RG-01
Descripción	Acceso mediante LTI al módulo de Administración en los casos en los que el LMS informa el rol de instructor.
Importancia	Alta
Prioridad	Alta

Tabla 5. Requisito general 04 “Acceso LTI a la partidas”

RG-04	Acceso LTI a la partidas
Dependencias	
Descripción	Acceso mediante LTI a las partidas desde la plataforma LMS
Importancia	Alta
Prioridad	Alta

Tabla 6. Requisito general 05 “Entorno Gestión JSF”

RG-05	Entorno Gestión JSF
Dependencias	
Descripción	Migración del Modulo de Gestión a tecnología JSF.
Importancia	Alta
Prioridad	Alta

Tabla 7. Requisito general 06 “Implementación Partidas Multijugador”

RG-06	Implementación Partidas Multijugador
Dependencias	
Descripción	Implementación de partidas en las que varios jugadores participan en la misma y responden en un panel de respuestas cuando les corresponde el turno.
Importancia	Alta
Prioridad	Alta

Tabla 8. Requisito general 07 “Implementación Partidas distribuidas”

RG-07	Implementación Partidas distribuidas
Dependencias	
Descripción	Implementación de partidas en las que todos los jugadores visualizan el juego en su terminal y responde directamente en el Juego.
Importancia	Alta
Prioridad	Alta

Tabla 9. Requisito general 08 “Implementación espectadores en las partidas”

RG-08	Implementación espectadores en las partidas
Dependencias	RG-06, RG-07
Descripción	Implementación de participantes en la partida, los cuales pueden visualizar el desarrollo de la misma pero no pueden intervenir en ella.
Importancia	Alta
Prioridad	Alta

Tabla 10. Requisito general 09 “Versión LTI”

RG-09	Versión LTI
Dependencias	
Descripción	Se usará la versión LTI 1.1. que permite la comunicación con las plataformas LMS.
Importancia	Alta
Prioridad	Alta

Tabla 11. Requisito general 10 “Migración Base de datos”

RG-10	Migración Base de datos
Dependencias	
Descripción	Migración de SQLite a PostgreSQL de la base de datos LTI
Importancia	Alta
Prioridad	Alta

Tabla 12. Requisito general 11 “Configuración de la aplicación en base de datos”

RG-11	Configuración de la aplicación en base de datos
Dependencias	RG-10
Descripción	Los parámetros de configuración de la aplicación se almacenarán en la base de datos.
Importancia	Alta
Prioridad	Alta

Tabla 13: Requisito general 12 “Control de acceso mediante roles”

RG-12	Control de acceso mediante roles
Dependencias	
Descripción	Los parámetros de configuración de la aplicación se almacenarán en la base de datos. Para el control del acceso a las distintas páginas de la aplicación se definirán los siguientes roles: <ul style="list-style-type: none"> • Superusuario: Puede acceder a todas las opciones de administración de la aplicación. • Profesor: Tiene un acceso limitado a las opciones de administración. • Alumno: Sólo puede acceder a las partidas.
Importancia	Alta
Prioridad	Alta

3.2.2 Requisitos funcionales

3.2.2.1 Casos de uso

Tabla 14. Caso de uso 01 “Iniciar sesión”

RCU-01	Iniciar sesión
Dependencias	RG-01, RG-02, RG-03
Descripción	<p>Para el acceso a la aplicación el usuario debe iniciar sesión, tanto si está identificado como si no. Al iniciar la sesión el sistema asigna los roles que permitirán el control del acceso a los distintos elementos de la aplicación.</p> <p>La sesión se puede iniciar en</p> <ul style="list-style-type: none"> • Página de login • Acceso mediante LTI • Acceso a partida por IDP (identificador de partida)
Importancia	Alta
Prioridad	Alta

Tabla 15: Caso de uso 02 “Consulta de la configuración de la aplicación”

RCU-02	Consulta de la configuración de la aplicación
Dependencias	RCU-01, RG-01, RG-02, RG-11, RG-12
Descripción	El rol de superusuario permite consultar la configuración en el módulo de Administración de la aplicación.
Importancia	Alta
Prioridad	Alta

Tabla 16: Caso de uso 03 “Modificación de la configuración de la aplicación”

RCU-03	Modificación de la configuración de la aplicación
Dependencias	RCU-01, RG-01, RG-02, RG-11, RG-12
Descripción	El rol de superusuario permite modificar la configuración en el módulo de Administración de la aplicación.
Importancia	Alta
Prioridad	Alta

Tabla 17: Caso de uso 04 “Consulta de consumidores LTI”

RCU-04	Consulta de consumidores LTI
Dependencias	RCU-01, RG-01, RG-02, RG-12
Descripción	El rol de superusuario permite consultar los consumidores registrados.
Importancia	Alta
Prioridad	Alta

Tabla 18: Caso de uso 05 “Registro de un consumidor LTI”

RCU-05	Registro de un consumidor LTI
Dependencias	RCU-01, RG-01, RG-02, RG-12
Descripción	El rol de superusuario permite registrar un nuevo consumidor LTI.
Importancia	Alta
Prioridad	Alta

Tabla 19: Caso de uso 06 “Borrado de un consumidor LTI”

RCU-06	Borrado de un consumidor LTI
Dependencias	RCU-01, RG-01, RG-02, RG-12
Descripción	El rol de superusuario permite borrar un consumidor LTI registrado previamente.
Importancia	Alta
Prioridad	Alta

Tabla 20: Caso de uso 07 “Modificación de un consumidor LTI”

RCU-07	Modificación de un consumidor LTI
Dependencias	RCU-01, RG-01, RG-02, RG-12
Descripción	El rol de superusuario permite modificar los datos de un consumidor LTI registrado previamente.
Importancia	Alta
Prioridad	Alta

Tabla 21: Caso de uso 08 “Consulta de usuarios”

RCU-08	Consulta de usuarios
Dependencias	RCU-01, RG-01, RG-02, RG-03, RG-12
Descripción	El rol de superusuario permite consultar todos los usuarios registrados. Los profesores pueden consultar los de su mismo consumidor LTI.
Importancia	Alta
Prioridad	Alta

Tabla 22: Caso de uso 09 “Registro de un usuario no LTI”

RCU-09	Registro de un usuario no LTI
Dependencias	RCU-01, RG-01, RG-02, RG-12
Descripción	El rol de superusuario permite registrar un nuevo usuario no LTI.
Importancia	Alta
Prioridad	Alta

Tabla 23: Caso de uso 10 “Borrado de un usuario no LTI”

RCU-10	Borrado de un usuario no LTI
Dependencias	RCU-01, RG-01, RG-02, RG-12
Descripción	El rol de superusuario permite borrar un usuario no LTI registrado previamente.
Importancia	Alta
Prioridad	Alta

Tabla 24: Caso de uso 11 “Modificación de un usuario no LTI”

RCU-11	Modificación de un usuario no LTI
Dependencias	RCU-01, RG-01, RG-02, RG-12
Descripción	El rol de superusuario permite modificar los datos de un usuario no LTI registrado previamente.
Importancia	Alta
Prioridad	Alta

Tabla 25: Caso de uso 12 “Consulta de juegos”

RCU-12	Consulta de juegos
Dependencias	RCU-01, RG-01, RG-02, RG-12
Descripción	El rol de superusuario permite consultar los juegos.
Importancia	Alta
Prioridad	Alta

Tabla 26: Caso de uso 13 “Registro de un juego”

RCU-13	Registro de un juego
Dependencias	RCU-01, RG-01, RG-02, RG-12
Descripción	El rol de superusuario permite registrar un nuevo juego.
Importancia	Alta
Prioridad	Alta

Tabla 27: Caso de uso 14 “Borrado de un juego”

RCU-14	Borrado de un juego
Dependencias	RCU-01, RG-01, RG-02, RG-12
Descripción	El rol de superusuario permite borrar un juego registrado previamente.
Importancia	Alta
Prioridad	Alta

Tabla 28: Caso de uso 15 “Modificación de un juego”

RCU-15	Modificación de un juego
Dependencias	RCU-01, RG-01, RG-02, RG-12
Descripción	El rol de superusuario permite modificar un juego registrado previamente.
Importancia	Alta
Prioridad	Alta

Tabla 29: Caso de uso 16 “Consulta de ficheros de preguntas”

RCU-16	Consulta de ficheros de preguntas
Dependencias	RCU-01, RG-01, RG-02, RG-12
Descripción	Tanto el rol de superusuario como el de profesor permiten consultar los ficheros de preguntas.
Importancia	Alta
Prioridad	Alta

Tabla 30: Caso de uso 17 “Registro de un fichero de preguntas”

RCU-17	Registro de un fichero de preguntas
Dependencias	RCU-01, RG-01, RG-02, RG-12
Descripción	Tanto el rol de superusuario como el de profesor permiten registrar un nuevo fichero de preguntas.
Importancia	Alta
Prioridad	Alta

Tabla 31: Caso de uso 18 “Borrado de un fichero de preguntas”

RCU-18	Borrado de un fichero de preguntas
Dependencias	RCU-01, RG-01, RG-02, RG-11, RG-12
Descripción	Tanto el rol de superusuario como el de profesor permiten borrar un fichero de preguntas.
Importancia	Alta
Prioridad	Alta

Tabla 32: Caso de uso 19 “Modificación de un fichero de preguntas”

RCU-19	Modificación de un fichero de preguntas
Dependencias	RCU-01, RG-01, RG-02, RG-12
Descripción	Tanto el rol de superusuario como el de profesor permiten modificar un fichero de preguntas registrado previamente.
Importancia	Alta
Prioridad	Alta

Tabla 33: Caso de uso 20 “Consulta de configuraciones de partida”

RCU-20	Consulta de configuraciones de partida
Dependencias	RCU-01, RG-01, RG-02, RG-03, RG-12
Descripción	Tanto el rol de superusuario como el de profesor permiten consultar las configuraciones de partida.
Importancia	Alta
Prioridad	Alta

Tabla 34: Caso de uso 21 “Registro de una configuración de partida”

RCU-21	Registro de una configuración de partida
Dependencias	RCU-01, RG-01, RG-02, RG-03, RG-12
Descripción	Tanto el rol de superusuario como el de profesor permiten registrar una nueva configuración de partida.
Importancia	Alta
Prioridad	Alta

Tabla 35: Caso de uso 22 “Borrado de una configuración de partida”

RCU-22	Borrado de una configuración de partida
Dependencias	RCU-01, RG-01, RG-02, RG-03, RG-12
Descripción	Tanto el rol de superusuario como el de profesor permiten borrar una configuración de partida registrada previamente.
Importancia	Alta
Prioridad	Alta

Tabla 36: Caso de uso 23 “Modificación de una configuración de partida”

RCU-23	Modificación de una configuración de partida
Dependencias	RCU-01, RG-01, RG-02, RG-03, RG-12
Descripción	Tanto el rol de superusuario como el de profesor permiten modificar una configuración de partida registrada previamente.
Importancia	Alta
Prioridad	Alta

Tabla 37: Caso de uso 24 “Consulta de partidas”

RCU-24	Consulta de partidas
Dependencias	RCU-01, RG-01, RG-02, RG-03, RG-12
Descripción	Tanto el rol de superusuario como el de profesor permiten consultar las partidas.
Importancia	Alta
Prioridad	Alta

Tabla 38: Caso de uso 25 “Registro de una partida”

RCU-25	Registro de una partida
Dependencias	RCU-01, RG-01, RG-02, RG-03, RG-12
Descripción	Tanto el rol de superusuario como el de profesor permiten registrar una nueva partida.
Importancia	Alta
Prioridad	Alta

Tabla 39: Caso de uso 26 “Borrado de una partida”

RCU-26	Borrado de una partida
Dependencias	RCU-01, RG-01, RG-02, RG-03
Descripción	Tanto el rol de superusuario como el de profesor permiten borrar una partida registrada previamente.
Importancia	Alta
Prioridad	Alta

Tabla 40: Caso de uso 27 “Modificación de una partida”

RCU-27	Modificación de una partida
Dependencias	RCU-01, RG-01, RG-02, RG-03, RG-12
Descripción	Tanto el rol de superusuario como el de profesor permiten modificar una partida registrada previamente.
Importancia	Alta
Prioridad	Alta

Tabla 41: Caso de uso 28 “Acceso del iniciador de partida multijugador”

RCU-28	Acceso del iniciador de partida multijugador
Dependencias	RCU-01, RG-01, RG-02, RG-03, RG-06
Descripción	Acceso del iniciador para gestionar el desarrollo de una partida multijugador.
Importancia	Alta
Prioridad	Alta

Tabla 42: Caso de uso 29 “Jugar en una partida multijugador”

RCU-29	Jugar en una partida multijugador
Dependencias	RCU-01, RG-01, RG-02, RG-03, RG-06
Descripción	Acceso para jugar una partida multijugador.
Importancia	Alta
Prioridad	Alta

Tabla 43: Caso de uso 30 “Seleccionar equipo y grupo”

RCU-30	<i>Seleccionar equipo y grupo</i>
Dependencias	RCU-01, RG-01, RG-02, RG-03, RG-06
Descripción	Los jugadores seleccionarán equipo y grupo para acceder a una partida multijugador.
Importancia	Alta
Prioridad	Alta

Tabla 44: Caso de uso 31 “Cambio de equipo y grupo”

RCU-31	<i>Cambio de equipo y grupo</i>
Dependencias	RCU-01, RG-01, RG-02, RG-03, RG-06, RCU-28
Descripción	El iniciador de la partida puede cambiar el equipo y grupo de un jugador antes del inicio de la misma.
Importancia	Alta
Prioridad	Alta

Tabla 45: Caso de uso 32 “Espectador en una partida multijugador”

RCU-32	Espectador en una partida multijugador
Dependencias	RCU-01, RG-01, RG-02, RG-03, RG-06
Descripción	Acceso como espectador de una partida multijugador.
Importancia	Alta
Prioridad	Alta

Tabla 46: Caso de uso 33 “Jugar en una partida distribuida multijugador”

RCU-33	Jugar en una partida distribuida multijugador
Dependencias	RCU-01, RG-01, RG-02, RG-03, RG-07
Descripción	Acceso para jugar una partida distribuida multijugador.
Importancia	Alta
Prioridad	Alta

Tabla 47: Caso de uso 34 “Espectador en una partida distribuida multijugador”

RCU-34	Espectador en una partida distribuida multijugador
Dependencias	RCU-01, RG-01, RG-02, RG-03, RG-07
Descripción	Acceso como espectador de una partida distribuida multijugador.
Importancia	Alta
Prioridad	Alta

Tabla 48: Caso de uso 35 “Enviar mensaje”

RCU-35	Enviar mensaje
Dependencias	RCU-01, RG-01, RG-02, RG-03, RG-06, RG-07
Descripción	Los participantes en una partida pueden enviar mensajes a otros participantes en la misma.
Importancia	Alta
Prioridad	Alta

Tabla 49: Caso de uso 36 “Recibir mensajes”

RCU-36	Recibir mensajes
Dependencias	RCU-01, RG-01, RG-02, RG-03, RG-06, RG-07
Descripción	Los participantes en una partida recibirán los mensajes que se dirijan a ellos.
Importancia	Alta
Prioridad	Alta

3.2.2.2 Reglas de negocio

Tabla 50: Regla de negocio 01 “Administración con acceso por LTI”

RN-01	Administración con acceso por LTI
Dependencias	RG-03, RG-12, RCU-16, RCU-17, RCU-18, RCU-19, RCU-20, RCU-21, RCU-22, RCU-23, RCU-24, RCU-25, RCU-26, RCU-27, RCU-28
Descripción	El acceso por LTI a la gestión de usuarios, partidas y configuraciones de partidas estará limitado a los datos correspondientes a la institución del usuario.
Importancia	Alta
Prioridad	Alta

Tabla 51: Regla de negocio 02 “Envío calificaciones LTI”

RN-02	Envío calificaciones LTI
Dependencias	RG-04, RG-06, RG-07
Descripción	Se enviarán las calificaciones de los jugadores de la partida al LMS en el caso de que el jugador haya accedido por LTI.
Importancia	Alta
Prioridad	Alta

3.2.2.3 Requisitos de conducta

Tabla 52: Requisitos de conducta 01 “Reconexión de un jugador”

RC-01	Reconexión de un jugador
Dependencias	RG-04, RG-06, RG-07
Descripción	Un mismo jugador podrá reconnectarse a la partida en la que estaba participando si por algún motivo pierde la conexión.
Importancia	Alta
Prioridad	Alta

Tabla 53: Requisitos de conducta 02 “Reconexión del iniciador”

RC-02	Reconexión del iniciador
Dependencias	RG-06, RCU-28
	Si se produce una nueva conexión de un iniciador, éste se convierte en el iniciador de la partida y se cierra la sesión del iniciador anterior.
Importancia	Alta
Prioridad	Alta

3.2.2.4 Requisitos de información

Tabla 54: Requisito de información 01: Sistema de gestión de la plataforma de gamificación

RI-01	Sistema de gestión de la plataforma de gamificación
Dependencias	RG-01, RG-02, RG-05, RG-10, RCU-02, RCU-03, RCU-04, RCU-05, RCU-06, RCU-07
Descripción	Es necesario poder configurar la plataforma y los consumidores autorizados a conectarse.
Importancia	Alta
Prioridad	Alta

3.2.3 Requisitos no funcionales

3.2.3.1 Usabilidad

Tabla 55: Requisito de usabilidad 01: Entrega de mensajes del chat

RU-01	Entrega de mensajes del chat
Dependencias	RG-06, RG-07
Descripción	La recepción de los mensajes del chat será inmediata e irá acompañada de una señal acústica.
Importancia	Media
Prioridad	Media

3.2.3.2 Portabilidad

Tabla 56: Requisito de portabilidad 01: Aplicación multiplataforma

RP-01	Aplicación multiplataforma
Dependencias	
Descripción	Se utilizarán tecnologías que permitan su despliegue en diversas plataformas de <i>hardware</i> , sistema operativo y servidor de aplicaciones.
Importancia	Alta
Prioridad	Alta

3.2.3.3 Seguridad

La seguridad de la aplicación se basa en roles. Cada usuario que inicia sesión en la aplicación, independientemente de cómo inicie la sesión, tiene asignado uno o varios de los siguientes roles:

- **Superusuario:** Puede acceder a todas las opciones de administración de la aplicación. Sólo se asigna a usuarios que acceden por la página de *login*, en caso de que les corresponda.
- **Profesor:** Tiene un acceso limitado a las opciones de administración. El acceso puede ser por página de *login* o por LTI. En este último caso, sólo puede acceder a los datos de su institución.
- **Alumno:** Sólo puede acceder a las partidas, y puede iniciar sesión de cualquier forma.

En caso de acceso LTI, si el profesor que ha creado la partida lo ha definido como iniciador de la partida, accederá a la página del iniciador en lugar de acceder para participar en la partida como jugador.

En los inicios de sesión por IDP de la partida, el rol asignado es siempre el de alumno.

En los inicios de sesión por LTI el rol LIS de *Learner* corresponde con el de alumno, mientras que el de *Instructor* corresponde con el de profesor. Ambos roles pueden estar activos simultáneamente en la misma sesión si el LMS comunica ambos en el lanzamiento.

En el caso de que la sesión se haya iniciado por la página de *login*, el rol asignado es el que tenga definido en la tabla de usuarios de la aplicación.

Tabla 57: Requisito de seguridad 01: Control del acceso a los recursos

RS-01	Control del acceso a los recursos
Dependencias	RG-01, RG-12
Descripción	Los usuarios sólo podrán acceder los recursos para los que estén autorizados.
Importancia	Alta
Prioridad	Alta

3.3 Arquitectura y funcionamiento básico de la aplicación

3.3.1 Arquitectura de capas de la aplicación

Podemos agrupar el funcionamiento de la aplicación en las siguientes capas.

1. Base de datos

Su misión es almacenar y garantizar la consistencia de la información.

2. Aplicación basada en Jakarta EE

Se encarga de procesar la información recibida desde los clientes, que en nuestro caso serán los navegadores de los usuarios. A su vez, podemos distinguir dentro de ella las siguientes subcapas:

- Vista

Interfaz de usuario construida con la tecnología *Facelets* de JSF, a su vez una de las tecnologías de Jakarta EE.

- Controlador

Dan soporte a los *Facelets*, tanto proporcionando la información solicitada como realizando las acciones solicitadas por los usuarios.

- Servicio

Permiten realizar las operaciones sobre las entidades, incluyendo la transaccionalidad de las operaciones.

- DAO

Operaciones básicas realizadas en la base de datos: consulta, inserción, actualización y borrado de datos.

En el siguiente diagrama se muestra la comunicación entre las distintas capas.

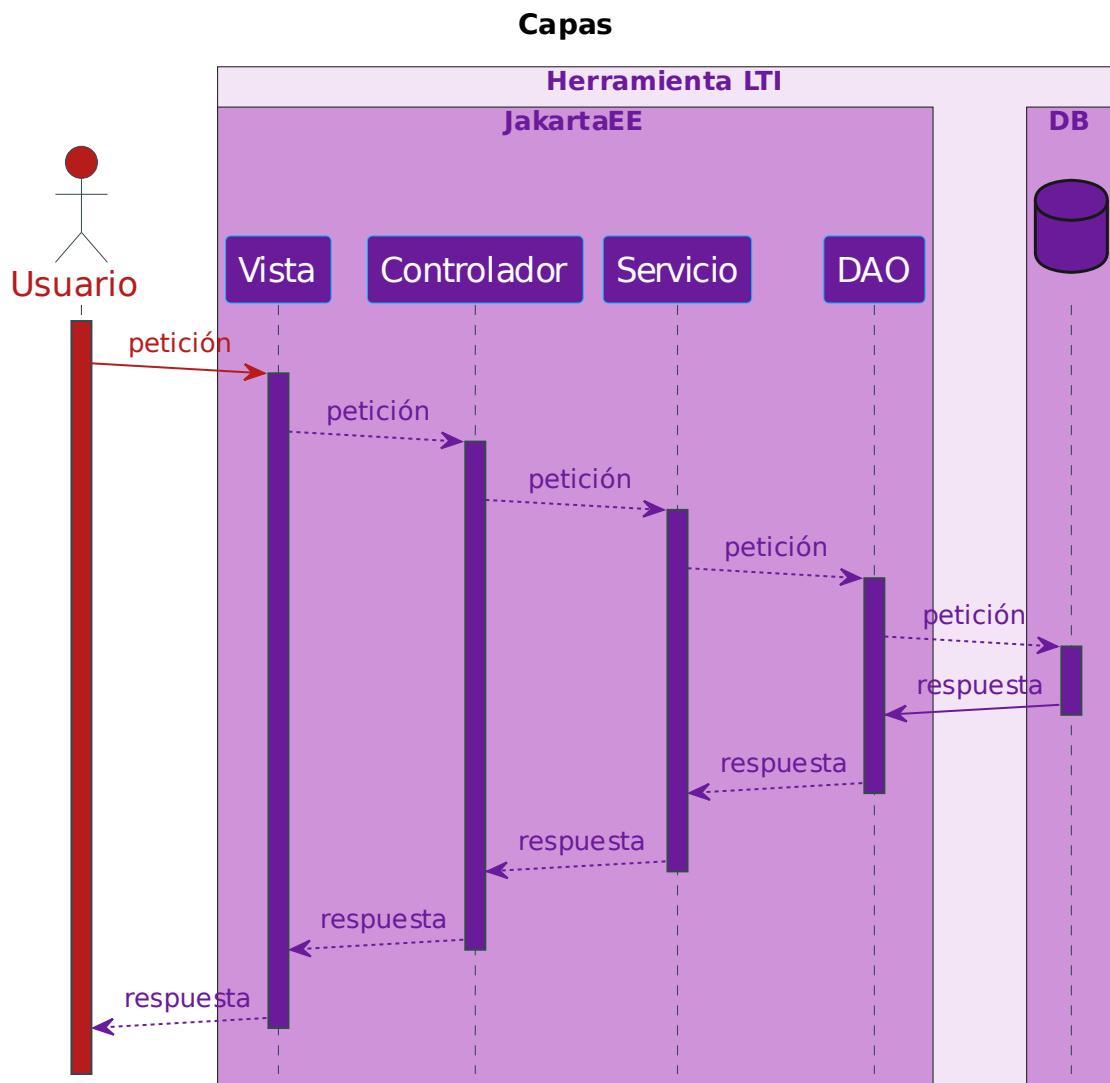


Figura 5: Diagrama de comunicación entre las capas de la aplicación

3.3.2 Arquitectura de componentes

La aplicación se puede organizar en componentes que proporcionan un subconjunto de la funciones requeridas para el funcionamiento y que se comunican entre ellos para dar soporte al funcionamiento de la aplicación.

Podemos ver estos componentes y sus relaciones en el siguiente diagrama.

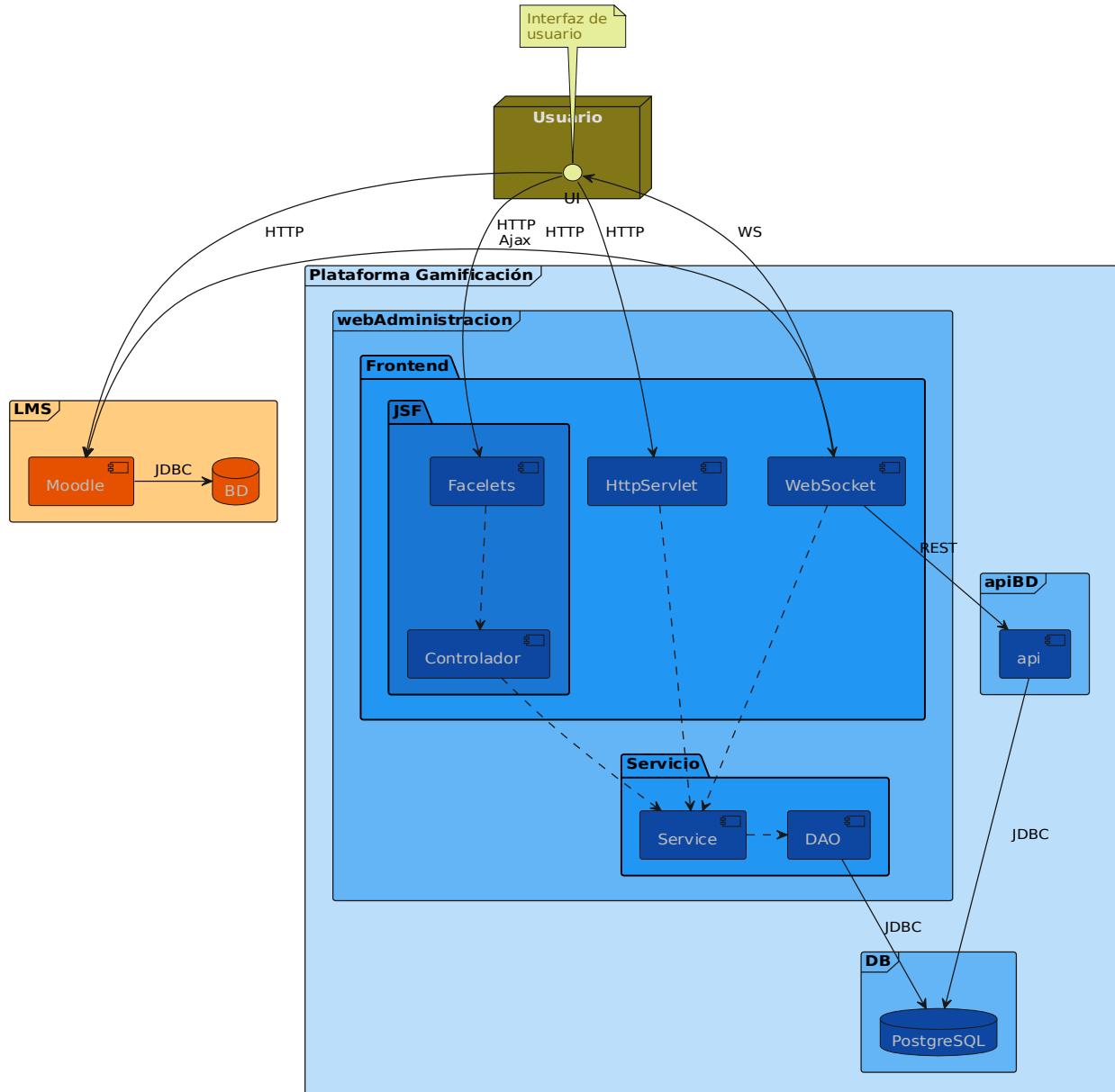


Figura 6: Diagrama de componentes

A continuación vamos a ver los componentes del sistema

1. Interfaz de usuario

Está compuesto por las páginas web de nuestra aplicación a las que acceder el usuario desde su navegador. El navegador se comunica con el LMS y con la Plataforma de gamificación mediante conexiones HTTP y WebSocket para mostrar el contenido de estas páginas y enviar la información correspondiente a las acciones del usuario.

2. LMS

Herramienta de enseñanza virtual de la institución a la que pertenece el usuario. En un entorno LTI actúa como *Tool Consumer* (TC). Proporciona al usuario los enlaces para acceder a la plataforma de gamificación.

3. Plataforma de gamificación

Proporciona acceso a recursos de gamificación que pueden ser incorporados para su uso desde distintos LMS. En un entorno LTI actúa como *Tool Provider* (TP).

A su vez, la plataforma podemos subdividirla en:

- webAdministracion

En este componente nos encontramos tanto con la web de administración como con el soporte a la disputa de partidas en la plataforma.

También se encarga del acceso a los datos almacenados en la base de datos utilizando para ello JPA, salvo en el caso de la persistencia de los resultados de las partidas en las que el acceso a la base de datos está mediado por apiBD.

Igualmente se encarga del envío al LMS de la información relativa a las calificaciones obtenidas por los alumnos en las partidas en las que han participado.

- apiBD

Se trata de una aplicación auxiliar que proporciona acceso mediante servicios REST a la base de datos. Se trata de una aplicación que proviene de uno de los trabajos anteriores y se usa para guardar los resultados de las partidas.

- Base de datos

Se trata de una base de datos PostgreSQL en la que se han unificado las bases de datos de los distintos trabajos anteriores que han servido de punto de partida para éste, que actúa como el almacén de datos persistentes de la aplicación.

Como hemos visto, el acceso a la misma se hace desde webAdministracion bien directamente con JPA o bien mediado por apiBD en cuyo caso se invocan los servicios REST correspondientes.

3.3.3 Funcionamiento básico de la aplicación

En este apartado se abordará y analizará el funcionamiento y estructura del proyecto desarrollado. Esto se puede dividir en dos partes, la parte del módulo de configuración de las partidas y la parte donde se desarrolla la misma.

- El módulo de configuración de partidas, es el encargado de administrar los usuarios, subir los juegos, crear las partidas y dar de alta los consumidores.

Se muestra a continuación la estructura de las opciones de menú.

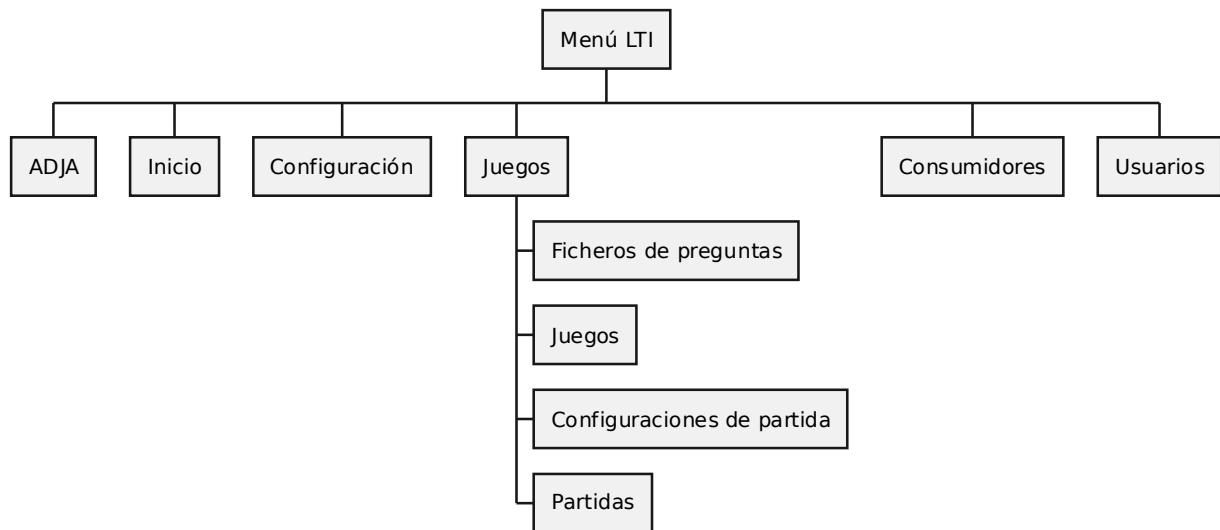


Figura 7: Estructura del menú de la aplicación

- Partiendo de una situación inicial en la que no tengamos creados ni usuarios ni partidas el funcionamiento sería el siguiente:
 - El superusuario, crea un nuevo consumidor con el nombre de la institución correspondiente, los campos clave LTI y secreto son generados automáticamente por la plataforma, aunque el campo secreto es editable. Para que el consumidor este activo se debe marcar Activo→“Sí” e indicar el intervalo de tiempo en el que se desea que este activado, debe indicarse al menos la fecha de inicio de la activación del consumidor.

La captura de pantalla muestra la interfaz de usuario para crear un nuevo consumidor. Los campos principales incluyen:

- Nombre:** Nombre (campo obligatorio)
- Clave LTI:** 1e468614-cd18-448e-8bb7-2423d58f7ab7
- Usuario LMS:** IMS LTI standard
- Activo:** Botón "no" (se recomienda marcarlo como "Sí")
- Activación:** Campos "Desde" y "Hasta" para establecer el período de actividad.
- Protegido:** Botón "no"
- Ruta CSS LTI:** Ruta CSS LTI
- URL herramienta:** http://gamifica.fjij.es:80/gamificacion-tfg_AlvaroSantiago_Cuenca/games
- Parámetros personalizados:**
 - management:** true (Acceso a la administración)
 - codigo_partida:** Campo código de la partida (Acceso a la partida)
 - espectador:** true (Espectador)

Figura 8: Añadir consumidor

- El superusuario deberán cargar los juegos si se desea utilizar juegos que no vengan precargados en el módulo. Para la subida de un juego, este debe estar en formato ZIP y se le debe asignar un nombre con el que aparecerá en la plataforma. Se podrán indicar datos adicionales del juego como la etapa educativa a la que corresponde, el tipo de respuesta, el número mínimo de preguntas...

(puede consultarse la descripción completa de estos datos en el apartado Base de Datos). El archivo ZIP a subir consta de un fichero HTML que recibe el nombre de “**jugar.html**” y opcionalmente de un archivo de texto llamado “**properties.txt**” en el que se encuentran diversos parámetros de configuración del juego. Este fichero viene incluido en el archivo ZIP donde está empaquetado el juego.

Añadir Juego

Figura 9: Añadir juego

3. El profesor o el superusuario subirá el fichero de preguntas, en el formato adecuado para el juego en el que se desea utilizar, habitualmente TXT. Se indicará el nombre del fichero, así como otros datos: asignatura, curso, observaciones, tipo de respuestas. La descripción completa de los datos de cada entidad aparece en el apartado Base de Datos.

Figura 10: Añadir fichero de preguntas

4. El superusuario creará una configuración de partida, que servirá de plantilla para crear una o varias partidas. En ella se indicarán valores que se usarán en la definición de partidas basadas en la misma. Los valores definidos en la configuración se toman como valores iniciales al crear la partida, aunque pueden modificarse. Los datos que no se cumplimenten en la configuración habrá que cumplimentarlos al crear la partida. Entre los datos que se pueden definir están el juego, el factor de corrección, el tipo de respuesta, los ficheros de preguntas... Puede consultarse la descripción completa de los datos en el apartado Base de Datos.

The screenshot shows a web-based configuration interface for game settings. At the top, there's a navigation bar with links to ADJA, Inicio, Configuración, Juegos, Consumidores, and Usuarios. On the right, there are links for Preferencias and Cerrar sesión (super). The main title is "Añadir Configuración de partida #". Below it, there are several input fields and dropdown menus:

- Juego:** A dropdown menu set to "Seleccionar".
- Profesor:** A dropdown menu set to "Seleccionar".
- Factor de corrección:** A slider set to 50 %.
- Curso:** A dropdown menu set to "Curso".
- Tema:** A dropdown menu set to "Tema".
- Asignatura:** A dropdown menu set to "Asignatura".
- Etapa:** A dropdown menu with options GEN, EI, EP, ESO, BAC, and UNIV.
- Tiempo de respuesta:** A slider set to 60 segundos.

Below these are two sections:

- Ficheros de preguntas:** A section for question files.
- Jsons de configuración:** A section for configuration JSONs.

At the bottom, there are tabs for "Equipos" and "Parámetros conf.", and a table for team management with columns: #, Nombre, Número de jugadores por equipo, and Acciones. The table currently shows "Sin datos".

Figura 11: Añadir configuración de partida

5. El superusuario o el profesor creará una partida en base a una configuración de partida existente. Para ello deberá seleccionar una configuración de partida, el juego con el que vayamos a realizar la actividad, el iniciador en caso de partidas multijugador, y otros parámetros como si es activa o no, la fecha de inicio y fin, si se desea que tenga espectadores, el modo de partida (Centralizada sin terminal, Centralizada LTI, Centralizada Multijugador, Distribuida).

The screenshot shows a web-based application for managing game configurations. At the top, there's a navigation bar with links to ADJA, Inicio, Configuración, Juegos, Consumidores, and Usuarios. On the right, there are links for Preferencias and Cerrar sesión (sign out). The main title is "Añadir Partida". Below it, there are several dropdown menus and input fields:

- Configuración de partida:** A dropdown menu set to "Seleccionar".
- Juego:** A dropdown menu set to "Seleccionar".
- Profesor:** A dropdown menu set to "Seleccionar".
- Iniciador:** A dropdown menu set to "Seleccionar".
- Título:** A text input field containing "Título".
- Clave de entrega:** A text input field containing "WYXxrAIH".
- Fecha actualización:** A date input field.
- Inicio / Fin:** Two date input fields for start and end dates.
- Finalizada:** A checkbox labeled "no".
- Permisos para iniciador:** A checkbox labeled "no".
- LTI:** A checkbox labeled "no".
- Cuenta:** A dropdown menu set to "1".
- Actividad:** A checkbox labeled "no".
- Con espectadores:** A checkbox labeled "no".
- status:** A checkbox labeled "no".
- Outcome:** A checkbox labeled "no".
- Pública:** A checkbox labeled "no".
- Máx. repeticiones:** A slider set to 3.
- Modo partida:** A dropdown menu with options CST, CLTI, CMJ, and DST.

At the bottom, there are tabs for Estado (Estado), Conf (Configuración), REG (Registro), Jugando (Jugando), and Fin (Finalizado). Below the tabs, there's a section for "Ficheros de preguntas" (Question files) with checkboxes for La Célula 1, Matemáticas, Cultura general 1, and Cultura general 1b. The footer includes copyright information for 2024 Álvaro Santiago, a link to the Department of Telematic Engineering, and the University of Seville logo.

Figura 12: Añadir partida

- Como segunda parte, tenemos la implementación de la partida LTI en una plataforma de enseñanza virtual:
 - El administrador del curso deberá configurar el enlace LTI en su LMS. Esta parte de configuración de un enlace LTI en una plataforma se detallará más adelante en el Anexo B Manual de usuario En el mismo se indicará el proceso en la plataforma Moodle donde se han realizado las pruebas.
 - El alumno o alumnos acceden a la partida a través del enlace LTI que les aparecerá en su LMS.
 - Una vez finalizada la partida, la calificación será enviada a la plataforma LMS para su almacenamiento.

En los próximos capítulos se ahondará en la implementación y se mostrarán más casos de funcionamiento.

3.3.4 Base de Datos

La estructura de la base de datos puede observarse en la siguiente Figura:

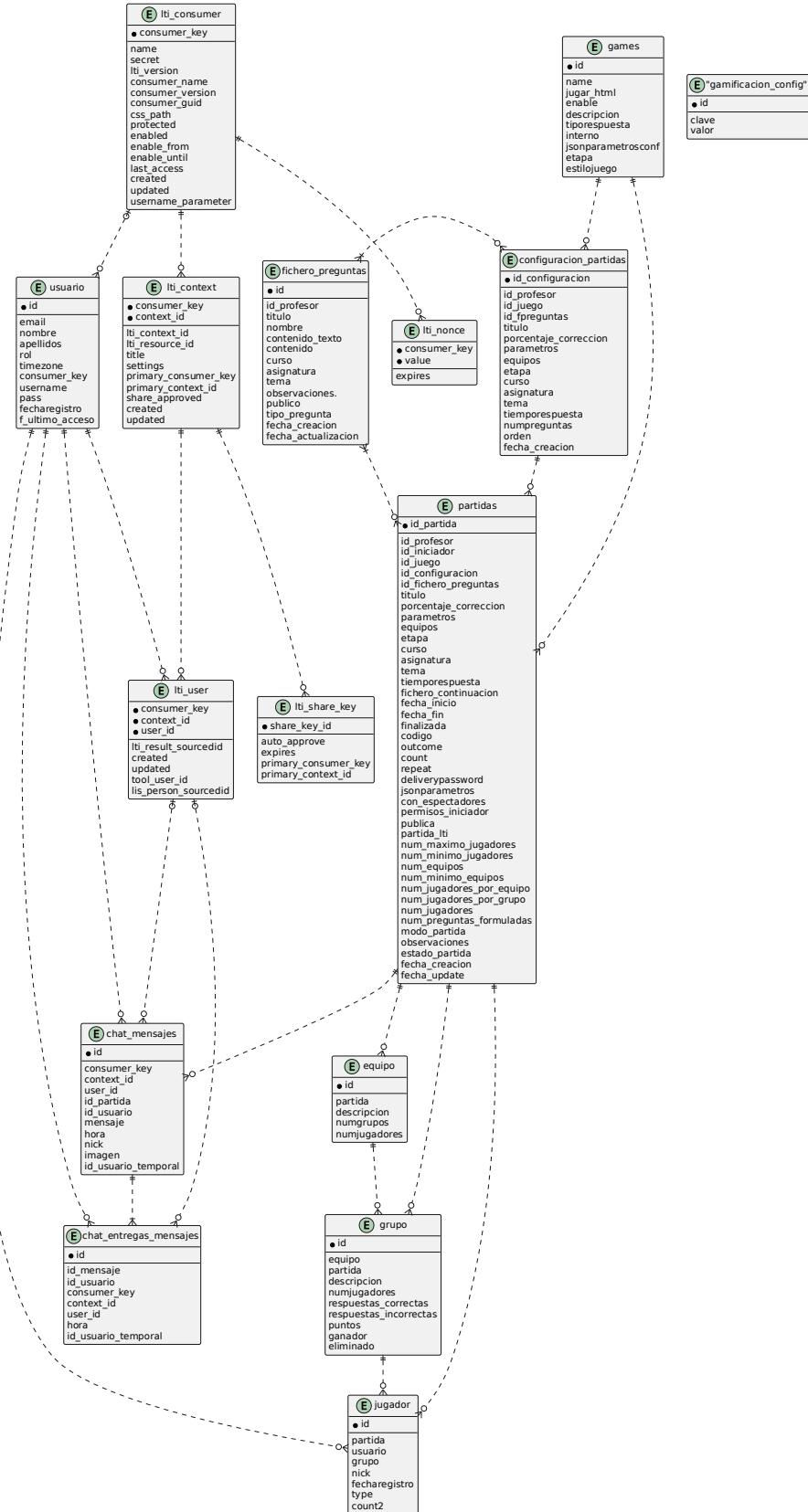


Figura 13: Estructura de la base de datos

Se han unificado en una sola base de datos en PostgreSQL la base de datos correspondiente a la información de los juegos con la que previamente estaba en SQLite correspondiente a la información necesaria para la comunicación LTI.

Se han fusionado las tablas con información relativa a los mismos elementos que estaban en ambas bases de datos.

Además se han añadido tablas para almacenar la información correspondiente a los mensajes entre los participantes de una partida.

3.3.4.1 Tablas de información de juegos y partidas

3.3.4.1.1 `gamification_config`

Tabla que contiene la información relativa a configuración del TP.

- **id**: clave sustituta del dato de configuración.
- **clave**: clave del dato de configuración.
- **valor**: valor del dato de configuración.

3.3.4.1.2 `games`

Tabla que contiene la información relativa a los juegos.

- **id**: clave sustituta del juego.
- **name**: nombre del juego.
- **jugar_html**: ubicación del fichero `jugar.html` del juego.
- **enable**: indica si el juego está activo o no.
- **descripcion**: descripción del juego.
- **tiporespuesta**: tipo de respuesta que se usa en el juego.
- **interno**: indicador si se trata de un juego incluido en la plataforma o cargado posteriormente.
- **jsonparametrosconf**: JSON con parámetros del juego.
- **etapa**: etapa educativa a la que está dirigido el juego.
- **estilojuego**: estilo del juego.

3.3.4.1.3 `configuracion_partidas`

Tabla que contiene la información relativa a las plantillas para facilitar la definición de partidas.

- **id_configuracion**: clave sustituta de la plantilla.
- **id_profesor**: identificador del profesor al que corresponde la plantilla (clave foránea a usuario).
- **id_juego**: identificador del juego de la plantilla (clave foránea a games).
- **id_fpreguntas**: identificador del fichero de preguntas utilizado en la plantilla (clave foránea a fichero_preguntas).
- **titulo**: título de la plantilla.
- **porcentaje_correccion**: porcentaje de respuestas correctas para considerar correcta la respuesta global del grupo.

- **parametros**: parámetros de la plantilla.
- **equipos**: información en formato JSON de los equipos y grupos.
- **etapa**: etapa educativa a la que está dirigida la plantilla.
- **curso**: curso al que está dirigida la plantilla.
- **asignatura**: asignatura para la que se hace la plantilla.
- **tema**: tema del que trata la plantilla.
- **tiemporespuesta**: tiempo de respuesta de los jugadores para las preguntas y votaciones.
- **numpreguntas**: número de preguntas.
- **orden**: orden aleatorio de las preguntas.
- **fecha_creacion**: fecha de creación de la plantilla.

3.3.4.1.4 fichero_preguntas

Tabla que contiene la información relativa a los ficheros de preguntas para las partidas.

- **id**: clave sustituta del fichero de preguntas.
- **id_profesor**: identificador del profesor que ha subido el fichero de preguntas (clave foránea a usuario).
- **titulo**: título del fichero de preguntas.
- **nombre**: nombre del fichero de preguntas.
- **contenido_texto**: contenido del fichero de preguntas en formato texto.
- **contenido**: contenido del fichero de preguntas sin convertir a formato texto.
- **curso**: curso al que está dirigido el fichero de preguntas.
- **asignatura**: asignatura para la que se hace el fichero de preguntas.
- **tema**: tema del que trata el fichero de preguntas.
- **observaciones**: observaciones relativas al fichero.
- **publico**: indica si se trata de un fichero de preguntas de uso público.
- **tipo_pregunta**: tipo de respuesta de la preguntas de este fichero.
- **fecha_creacion**: fecha de creación del fichero de preguntas.
- **fecha_actualizacion**: fecha de la última modificación del fichero de preguntas.

3.3.4.1.5 partidas

Tabla que contiene la información relativa a las partidas.

- **id_partida**: clave sustituta de la partida.
- **id_profesor**: identificador del profesor al que corresponde la partida (clave foránea a usuario).
- **id_iniciador**: identificador del iniciador de la partida (clave foránea a usuario).
- **id_juego**: identificador del juego de la partida (clave foránea a games).

- **id_configuracion**: identificador de la plantilla en la que se basa la partida (clave foránea a configuracion_partidas).
- **id_fichero_preguntas**: identificador del fichero de preguntas utilizado en la partida (clave foránea a fichero_preguntas).
- **titulo**: título de la partida.
- **porcentaje_correccion**: porcentaje de respuestas correctas para considerar correcta la respuesta global del grupo.
- **parametros**: parámetros de la partida.
- **equipos**: información en formato JSON de los equipos y grupos de la partida.
- **etapa**: etapa educativa a la que está dirigida la partida.
- **curso**: curso al que está dirigida la partida.
- **asignatura**: asignatura para la que se hace la partida.
- **tema**: tema del que trata la partida.
- **tiemporespuesta**: tiempo de respuesta de los jugadores para las preguntas y votaciones.
- **fichero_continuacion**: datos para continuación de la partida.
- **fecha_inicio**: fecha de inicio a partir de la cual se puede jugar la partida.
- **fecha_fin**: fecha de fin a partir de la cual ya no es posible jugar la partida.
- **finalizada**: indicador de partida finalizada.
- **codigo**: IDP (identificador de la partida). Se utiliza para acceder a la partida, bien introduciéndolo en la pantalla o como parámetro del enlace LTI en el LMS.
- **outcome**: indicador si la partida enviará mensaje *outcome* al LMS con las calificaciones en caso de que el jugador haya accedido por LTI.
- **count**: número de veces que se ha jugado la partida.
- **repeat**: máximo número de veces que un alumno puede jugar la partida.
- **deliverypassword**: clave para acceder a la partida como jugador sin necesidad de hacer *login*.
- **jsonparametros**: parámetros fijos y opcionales de la partida.
- **con_espectadores**: indicador de posibilidad de espectadores en la partida.
- **permisos_iniciador**: indica si el iniciador está autorizado a modificar algunos valores de la partida.
- **publica**: indicador de si es una partida pública.
- **partida_lti**: indicador de si es una partida a la que se accede mediante LTI.
- **num_maximo_jugadores**: número máximo de jugadores en la partida.
- **num_minimo_jugadores**: número mínimo de jugadores en la partida.
- **num_equipos**: número de equipos en la partida.
- **num_minimo_equipos**: número mínimo de equipos en la partida.

- **num_jugadores_por_equipo**: número de jugadores por equipo.
- **num_jugadores_por_grupo**: número de jugadores por grupo.
- **num_jugadores**: número de jugadores.
- **num_preguntas_formuladas**: número de preguntas formuladas la última vez que se ha jugado la partida.
- **modo_partida**: modo de juego para esta partida.
- **observaciones**: observaciones relativas a la partida.
- **estado_partida**: estado de la partida.
- **fecha_creacion**: fecha de creación de la partida.
- **fecha_update**: fecha de la última modificación de la partida.

3.3.4.1.6 **equipo**

Tabla que contiene la información relativa a los equipos participantes en las partidas.

- **id**: clave sustituta del equipo.
- **partida**: identificador de la partida en la que ha participado (clave foránea a partidas).
- **descripcion**: nombre del equipo.
- **numgrupos**: número de grupos del equipo.
- **numjugadores**: número de jugadores en el equipo.

3.3.4.1.7 **grupo**

Tabla que contiene la información relativa a los grupos participantes en las partidas.

- **id**: clave sustituta del grupo.
- **equipo**: identificador de la equipo al que pertenece el grupo (clave foránea a equipo).
- **partida**: identificador de la partida en la que ha participado (clave foránea a partidas).
- **descripcion**: nombre del grupo.
- **numjugadores**: número de jugadores en el grupo.
- **respuestas_correctas**: número de respuestas contestadas correctamente por el grupo.
- **respuestas_incorrectas**: número de respuestas contestadas incorrectamente por el grupo.
- **puntos**: puntuación del grupo en la partida.
- **ganador**: indicador de si el grupo ha ganado la partida.
- **eliminado**: indicador de si el grupo ha sido eliminado de la partida.

3.3.4.1.8 **jugador**

Tabla que contiene la información relativa a los jugadores participantes en las partidas.

- **id**: clave sustituta del jugador.
- **partida**: identificador de la partida en la que ha participado (clave foránea a partidas).

- **usuario**: identificador del usuario (clave foránea a usuario).
- **grupo**: identificador del grupo al que pertenece (clave foránea a grupo).
- **nick**: nick (apodo) del jugador en la partida.
- **fecharegistro**: fecha de registro del jugador.
- **type**: rol de partida con el que ha participado el usuario.
- **count2**: número de veces que ha jugado esta partida.

3.3.4.1.9 **usuario**

Tabla que contiene la información relativa a los usuarios de la plataforma de gamificación.

- **id**: clave sustituta del usuario.
- **email**: correo electrónico de este usuario.
- **nombre**: nombre de este usuario.
- **apellidos**: apellidos de este usuario.
- **rol**: rol en la aplicación de este usuario.
- **timezone**: huso horario de este usuario.
- **consumer_key**: identificador del consumidor LTI, sólo aparecerá cuando se trate de usuarios correspondientes a conexiones LTI (clave foránea a `lti_consumer`).
- **username**: nombre de usuario.
 - En el caso de usuarios correspondientes a conexiones LTI, se almacena el valor del campo `user_id` comunicado por el LMS en el lanzamiento LTI.
 - En el resto de caso se asigna manualmente al dar de alta el usuario y es el que hay que poner para hacer *login*.
- **pass**: *hash* de la contraseña del usuario codificada usando PBKDF2 (*password-based key derivation function*) [37].
- **fecharegistro**: fecha de registro del usuario.
- **f_ultimo_acceso**: fecha del último acceso del usuario.

3.3.4.2 Tablas para comunicación LTI

Son las tablas cuyo nombre empieza por `lti_` y son requeridas para el funcionamiento de la biblioteca Oscelot que implementa el estándar LTI 1.1.

3.3.4.2.1 **lti_consumer**

`lti_c`

Tabla que contiene la información relativa a los distintos consumidores (*Tool Consumer*) que pueden utilizar esta herramienta.

- **consumer_key**: es la identificación del TC en este TP y se utiliza para la autenticación OAuth en el mensaje `basic-lti-launch-request`. Generado automáticamente por el TP al dar de alta un nuevo TC.
- **name**: nombre del TC.

- **secret**: secreto compartido con el TC para la firma de los mensajes OAuth.
- **lti_version**: versión de LTI, se actualiza automáticamente con la información contenida en el mensaje **basic-lti-launch-request**.
- **consumer_name**: nombre del TC recibido del TC, se actualiza automáticamente con la información contenida en el mensaje **basic-lti-launch-request**.
- **consumer_version**: versión del *software* utilizado por el LMS recibida del TC, se actualiza automáticamente con la información contenida en el mensaje **basic-lti-launch-request**.
- **consumer_guid**: identificador del LMS recibido del TC, se actualiza automáticamente con la información contenida en el mensaje **basic-lti-launch-request**.
- **css_path**: hoja de estilos para usar con este TC.
- **protected**: indica si la conexión con este TC está protegida.
- **enabled**: indica si la conexión con este TC está habilitada.
- **enable_from**: fecha de inicio del período en que la conexión con este TC está habilitada.
- **enable_until**: fecha de fin del período en que la conexión con este TC está habilitada.
- **last_access**: fecha del último acceso desde este TC, se actualiza automáticamente con la recepción de mensajes **basic-lti-launch-request**.
- **created**: fecha de creación del consumidor.
- **updated**: fecha de la última actualización del consumidor.
- **username_parameter**: nombre del parámetro donde el LMS indica el nombre del usuario conectado, puesto que varía dependiendo del LMS que actúa como TC [38].

3.3.4.2.2 **ontext**

lti_c

Tabla que contiene la información relativa al contexto recibida en el mensaje LTI. El contexto LTI corresponde con el curso en el que está ubicado el enlace.

- **consumer_key**: identificador del TC (clave foránea a lti_consumer).
- **context_id**: identificador local en el TP del contexto del LMS.
- **lti_context_id**: identificador del contexto recibido del TC.
- **lti_resource_id**: identificador en el TC del último recurso accedido.
- **title**: título del contexto en el TC, se actualiza automáticamente con la información contenida en el mensaje **basic-lti-launch-request**.
- **settings**: ajustes recibidos en el último mensaje **basic-lti-launch-request**.
- **primary_consumer_key**: usado en caso de compartición de recursos.
- **primary_context_id**: usado en caso de compartición de recursos.
- **share_approved**: indica si la compartición está habilitada.
- **created**: fecha de creación del contexto.
- **updated**: fecha de la última actualización del contexto.

3.3.4.2.3 lti_user

Tabla que contiene la información relativa al recurso LTI, éste incluye tanto información del enlace como del usuario que efectúa el lanzamiento LTI.

- **consumer_key**: identificador del TC.
- **context_id**: identificador del contexto (junto con la columna anterior forman clave foránea a lti_context).
- **user_id**: identificador del usuario en el TC.
- **lti_result_sourcedid**: URL en el TC para la comunicación del resultado de la actividad.
- **created**: fecha de creación del usuario.
- **updated**: fecha de la última actualización del usuario.
- **tool_user_id**: identificador del usuario asignado en la plataforma de gamificación (clave foránea a usuario).
- **lis_person_sourcedid**: nombre del usuario en el TC.

3.3.4.2.4 lti_nonce

Tabla que contiene la información relativa al nonce del mensaje **basic-lti-launch-request**, que evita que se procese un mensaje caducado o en más de una ocasión. Un nonce es un número arbitrario que sólo se puede utilizar una vez en un canal de comunicación.

- **consumer_key**: identificador del TC (clave foránea a lti_consumer).
- **value**: valor del nonce.
- **expires**: hora límite de validez del nonce.

3.3.4.2.5 lti_s hare_key

Tabla que contiene la información relativa a compartición de recursos LTI. Es una tabla utilizada por la biblioteca Oscelot, aunque no se ha utilizado en este trabajo.

- **share_key_id**: identificador de la compartición.
- **auto_approve**: indica si la aprobación de la compartición es automática.
- **expires**: hora límite de validez de la compartición.
- **primary_consumer_key**: identificador del TC del contexto compartido.
- **primary_context_id**: identificador del contexto compartido.

3.3.4.3 Tablas de mensajería

3.3.4.3.1 chat_ mensajes

Tabla que contiene la información relativa a los mensajes enviados en el chat.

- **id**: clave sustituta del mensaje.
- **consumer_key**: identificador del TC del remitente.
- **context_id**: identificador del contexto del TC del remitente.

- **user_id**: identificador del usuario del TC del remitente (junto con las dos columnas anteriores forman clave foránea a lti_user).
- **id_partida**: identificador de la partida en la que se envía el mensaje (clave foránea a partidas).
- **id_usuario**: identificador del remitente en el el TP (clave foránea a usuario).
- **mensaje**: texto del mensaje.
- **hora**: hora de envío del mensaje.
- **nick**: nick utilizado por el remitente en la partida y que se muestra a los destinatarios como remitente del mensaje.
- **imagen**: imagen del usuario para mostrar a los destinatarios.
- **id_usuario_temporal**: identificador de usuario temporal del remitente en caso de acceder a la partida por IDP sin hacer *login* ni conectar por LTI.

3.3.4.3.2 **chat_entregas_mensajes**

Tabla que contiene la información relativa a los destinatarios de los mensajes. Al almacenarse en esta tabla se pueden mostrar al destinatario cuando se conecte si en el momento del envío no está conectado al chat. Se recoge la hora real de entrega de los mensajes, tanto si es inmediata porque el destinatario está conectado como si lo recibe posteriormente al conectarse al chat.

- **id**: clave sustituta de la entrega.
- **id_mensaje**: identificador del mensaje (clave foránea a chat_mensajes).
- **id_usuario**: identificador del destinatario en el el TP (clave foránea a usuario).
- **consumer_key**: identificador del TC del destinatario.
- **context_id**: identificador del contexto del TC del destinatario.
- **user_id**: identificador del destinatario en el el TP (junto con las dos columnas anteriores forman clave foránea a lti_user).
- **hora**: hora real de entrega del mensaje.
- **id_usuario_temporal**: identificador de usuario temporal del destinatario en caso de acceder a la partida por IDP sin hacer *login* ni conectar por LTI.

4 IMPLEMENTACIÓN

4.1 Código

4.1.1 Bibliotecas utilizadas

4.1.1.1 Jakarta EE Platform API

Contiene el API de Jakarta EE, que define una plataforma estándar para alojar aplicaciones Jakarta EE [15]. El servidor WildFly proporciona la implementación de Jakarta EE.

Es el núcleo principal de nuestra aplicación, ya que se trata de una aplicación desarrollada para la plataforma Jakarta EE.

No es necesaria su inclusión en el desplegable, puesto que el propio servidor WildFly proporciona una implementación de la misma.

4.1.1.2 PrimeFaces

Biblioteca [16] con componentes para la construcción de páginas JSF.

Proporciona componentes adicionales para la construcción de páginas web con mejoras sobre los proporcionados de forma estándar por JSF.

Se han utilizado elementos de entrada de datos para los formularios y elementos de visualización de datos. Estos últimos aportan la posibilidad de usar fuentes de datos de tipo *lazy*, que evitan la carga innecesaria de datos que no se van a mostrar al usuario desde la capa de persistencia.

4.1.1.3 Primeflex

Biblioteca [39] ligera para interfaces de usuario responsivas, compatible con PrimeFaces.

Facilita la creación de interfaces de usuario responsivas, es decir, que se adaptan automáticamente al tamaño y resolución del elemento de visualización que está utilizando el usuario.

4.1.1.4 OmniFaces

Biblioteca [40] de utilidades para JSF que se centra en utilidades que facilitan las tareas diarias con la API estándar de JSF, sin incluir componentes visuales.

Se ha utilizado en algunas páginas de la aplicación para formatear valores de fecha y para solucionar algunos problemas en los formularios con la codificación de caracteres, para lo que hemos declarado el filtro `org.omnifaces.filter.CharacterEncodingFilter` en el fichero `web.xml`.

4.1.1.5 Font Awesome

Biblioteca [41] con más de 30.000 iconos con para utilizar en páginas web, de los cuales más de 2.000 son de uso libre, compatible con PrimeFaces.

Algunos de los iconos de esta biblioteca se utilizan en las páginas web de la aplicación.

4.1.1.6 PrimeFaces Bootstrap Theme

Biblioteca [42] para PrimeFaces con un tema para la construcción de interfaces de usuario responsivas basado en la biblioteca bootstrap [43], que a su vez es una biblioteca que proporciona CSS y JS para la construcción de interfaces responsivas.

Se utiliza en algunas de las páginas web de la aplicación.

4.1.1.7 PostgreSQL JDBC Driver

Controlador JDBC [44] que permite el acceso a bases de datos PostgreSQL desde aplicaciones que se ejecutan en una JVM.

Se usa para acceder a la base de datos PostgreSQL, tanto en los casos en que se usa directamente el mismo, como cuando se hace mediante JPA.

4.1.1.8 Hibernate ORM Hibernate Core

Biblioteca [45] para mapeo relacional de objetos que implementa la especificación JPA compatible con aplicaciones basadas en la biblioteca Jakarta EE Platform API.

La especificación JPA es una API que necesita una implementación de su funcionalidad. Hibernate ORM es una de las implementaciones de JPA más usadas y es la incluida en el servidor WildFly.

4.1.1.9 IMS BasicLTI Utilities

Biblioteca [46] que ofrece un conjunto de clases de utilidad para ayudar en el desarrollo de los consumidores y proveedores BasicLTI. Se ocupan de gran parte del trabajo pesado y hacen que el proceso sea más opaco para el desarrollador.

Se utiliza para consultar los nombres de los parámetros incluidos en los lanzamientos LTI.

4.1.1.10 Gson

Gson [47] es una biblioteca Java que se puede utilizar para convertir objetos Java en su representación JSON. También se puede utilizar para convertir una cadena JSON en un objeto Java equivalente. Gson puede trabajar con objetos Java arbitrarios, incluidos objetos preexistentes de los que no tiene código fuente.

Se utiliza para tratar el contenido de datos en formato JSON (*JavaScript Object Notation*) en varios módulos de la aplicación.

El formato JSON es sintácticamente idéntico al código para crear objetos JavaScript. Debido a esta similitud, un programa JavaScript puede convertir fácilmente datos JSON en objetos JavaScript nativos.

La sintaxis JSON se deriva de la sintaxis de la notación de objetos de JavaScript, pero el formato JSON es sólo texto. El código para leer y generar datos JSON puede escribirse en cualquier lenguaje de programación.

4.1.1.11 FasterXML Jackson

Biblioteca [48] que permite leer y escribir documentos JSON.

Se usa para procesar los mensajes enviados por la conexión WebSocket del chat que van en formato JSON y para conversores JPA que nos permiten almacenar objetos en base de datos en formato JSON.

4.1.1.12 JDOM

Biblioteca [49] que permite acceder, manipular y generar datos en formato XML.

Se utiliza en la biblioteca Oscelot para procesar los mensajes intercambiados en las conexiones LTI.

4.1.1.13 OpenCSV

Biblioteca [50] de analizadores CSV (*comma-separated values*, valores separados por comas) fácil de usar para Java.

Se utiliza para procesar datos en formato CSV almacenados en tablas de la base de datos.

4.1.1.14 IText Core

Biblioteca [51] que proporciona clases para lectura y escritura de ficheros en formato PDF.

Se utiliza en el generador de ficheros de preguntas incluido de un trabajo anterior.

4.1.1.15 Docx4J

Biblioteca Java [52] de código abierto (ASLv2) para crear y manipular archivos Microsoft Open XML (Word docx, Powerpoint pptx y Excel xlsx).

Se utiliza en el generador de ficheros de preguntas incluido de un trabajo anterior.

4.1.1.16 SLF4J API Module

Simple Logging Facade for Java (SLF4J) [53] es una fachada o abstracción para diversas bibliotecas de registro, permitiendo al usuario final conectar la biblioteca de registro deseada en el momento del despliegue.

Permite una configuración de registro independiente que puede aplicarse a una variedad de bibliotecas de registro en función de la utilizada en el despliegue de la aplicación.

Utilizada para posibilitar la escritura de mensajes en el registro por parte de la aplicación.

4.1.1.17 Apache Commons Lang

Biblioteca [54] que forma parte del proyecto Apache Commons [55], proporciona una serie de utilidades de ayuda para la API `java.lang`, en particular métodos de manipulación de cadenas, métodos numéricos básicos, reflexión de objetos, concurrencia, creación y serialización y propiedades del sistema. Además, contiene mejoras básicas de `java.util.Date` y una serie de utilidades dedicadas a ayudar con la construcción de métodos, como `hashCode`, `toString` y `equals`.

Se utiliza fundamentalmente para detectar si una cadena es nula, vacía o contiene sólo caracteres en blanco: espacios, tabuladores, saltos de línea...

4.1.1.18 Apache Commons BeanUtils

Biblioteca [56] que forma parte del proyecto Apache Commons [55], centrado en proporcionar envoltorios fáciles de usar para acceder a las API de reflexión e introspección proporcionadas de forma nativa por el JDK.

En la aplicación se ha utilizado para acceder mediante programación a las propiedades de determinadas clases.

4.1.1.19 JUniversalChardet

Implementación [57] en Java de universalchardet, la biblioteca de detección de codificación de texto de Mozilla.

Se utiliza para detectar la codificación de los ficheros de texto subidos por los usuarios a la plataforma.

4.1.1.20 Spring Framework

Biblioteca básica [58] para Spring que, junto con Spring Beans, proporciona inyección de dependencia y funciones de IoC (*Inversion of Control*).

Utilizada en la aplicación apiBD y en las llamadas a sus servicios REST desde la aplicación principal.

Uno de los componentes es Spring Web, que proporciona características de integración, como la funcionalidad de carga de archivos multiparte y la inicialización del contenedor IoC mediante escuchas de *servlets* y un contexto de aplicación orientado a la web. También contiene un cliente HTTP y las partes relacionadas con la web del soporte remoto de Spring.

4.1.1.21 JSON In Java

Los archivos de este paquete [59] implementan codificadores/decodificadores JSON en Java. También incluye la capacidad de convertir entre JSON y XML, encabezados HTTP, *cookies* y CDL. Esta es una implementación de referencia.

Se usa en la aplicación para la construcción de los mensajes que se envían por la conexión WebSocket.

4.1.1.22 Oscelot

La biblioteca Java Oscelot [60] (*Open Source Community for Educational Learning Objects and Tools*), creada por Stephen P. Vickers, encapsula el código necesario para que un proveedor de herramientas compatible con LTI 1 se comunique con un consumidor de herramientas. Incluye soporte para LTI 1.1 y las extensiones no oficiales de Basic LTI.

Se usa para procesar las peticiones de lanzamiento LTI recibidas de los TC, así como para enviar a los LMS las calificaciones obtenidas.

4.1.2 Clases

Según su finalidad, podemos agrupar las clases en los siguientes paquetes.

4.1.2.1 Definición del Modelo

En JPA (*Java Persistence API*, en castellano API de persistencia para Java, actualmente renombrado a *Jakarta Persistence*) [35] se definen clases que mapean las tablas de la base de datos para permitir el acceso a los datos almacenados. También pueden definirse clases auxiliares para la definición de datos complejos y la conversión entre las representaciones de los datos en la aplicación y en la base de datos.

1. Entidades

Son las encargadas de realizar el ORM (*Object Relational Mapping*, en español Mapeo relacional de objetos) entre la aplicación (objeto) y la base de datos donde se almacenan los datos (relacional), tanto las correspondientes a la plataforma de juegos como las correspondientes a la integración LTI.

2. Clases embebidas

Corresponden a objetos compuestos que se mapean en más de una columna en la tabla de la base de datos. Pueden corresponder a claves primarias compuestas o a otros datos.

3. Enumeraciones

Se definen para los casos en que determinadas propiedades sólo pueden tomar un número limitado de valores.

- EstadoPartida: enumeración con los posibles estados de la partida.

- Etapa: enumeración de las posibles etapas educativas.
- ModoPartida: enumeración de los posibles modos de juego de una partida.
- RolPartida: enumeración de los posibles roles de un usuario en una partida.
- RolUsuario: enumeración de los posibles roles de un usuario que ha iniciado sesión.
- TipoRespuesta: enumeración de los posibles tipos de respuesta que se pueden utilizar en los juegos.

4.1.2.2 Controlador

Son las clases que se vinculan desde los ficheros JSF, y que proporcionan los datos a mostrar en las mismas y realizan las acciones que el usuario solicita.

1. LoginController: usado en la página de *login*.
2. SelectController: proporciona los elementos de las listas de valores para los datos correspondientes a enumeraciones.
3. Controladores para entidades: hay un controlador para cada entidad gestionada en el módulo de administración, que da soporte a las páginas relacionadas del módulo de administración.
4. Controladores para las vistas de la partida: son los controladores para las páginas que se utilizan para mostrar el desarrollo de las partidas.

4.1.2.3 Servicio

Las clases de esta capa definen las operaciones que se realizan con los datos. Estas operaciones con su mayor o menor grado de complejidad invocarán a métodos de las clases DAO de las entidades del modelo.

Para realizar las operaciones, éstas se ejecutan dentro de una transacción definida en estas clases.

1. Servicio

Clase abstracta genérica que proporciona las operaciones básicas que se pueden realizar sobre una entidad.

2. Servicios de entidades

Una clase para cada una de las entidades del modelo que extiende Servicio y que concreta la clase del modelo a la que se aplica. Define la clase para los objetos gestionados y puede proporcionar servicios adicionales para la entidad a la que corresponde.

Además se definen el resto de posibles operaciones a realizar sobre la entidad y se crea la transacción dentro de la que se realizan las operaciones.

3. LazyEntityDataModel

Implementación de **LazyDataModel** de PrimeFaces que obtiene la información desde una base de datos. Permite que una tabla de datos en PrimeFaces sólo recupere los datos que se están mostrando en pantalla y acceda a la base de datos sólo cuando sea necesario el acceso a registros adicionales. Utilizable en los componentes de PrimeFaces que permiten un *datamodel* (componente que se usa para la construcción de las páginas web y permite la navegación por un conjunto de filas avanzando, retrocediendo o filtrando) de tipo *lazy* (la carga de datos desde la base de datos está diferida hasta que estos son necesarios para mostrarlos al usuario).

4.1.2.4 DAO

1. Dao

Clase abstracta genérica con la definición de las operaciones básicas a realizar en cada entidad (*find, save, create, update, delete*).

2. DaoDB

Clase genérica que añade la conexión a base de datos a las operaciones definidas en la clase Dao. Se instancia en cada servicio.

3. DaoEntityManager

Clase que proporciona a DaoDB el *entityManager* para realizar las operaciones en la base de datos. El *entityManager* es el encargado en JPA de realizar las operaciones de las entidades del modelo en la base de datos.

4. Resultado

Clase genérica que contiene el resultado de una consulta, tanto los datos cargados de la base de datos como el total de filas que cumplen las condiciones de la consulta. Utilizada en la clase LazyEntityDataModel, descrita anteriormente en la capa de Servicio.

4.1.2.5 Filtros

1. LoginFilter

Clase que comprueba que el usuario conectado tiene acceso a la página solicitada. Si no hay un usuario conectado redirige a la página de *login*.

4.1.2.6 Servlets

1. Facelet Servlet

Proporcionado por la plataforma Jakarta EE, se encarga de responder las peticiones de los clientes para acceder a las páginas JSF de la aplicación, así como recursos adicionales definidos para las mismas como hojas de estilos, javascript, imágenes.

2. Launch, LaunchRequestInfo

Launch es el *servlet* que recibe las peticiones iniciales POST de conexión mediante LTI. Comprueba usando la biblioteca Oscelot que la petición es válida y redirige a la URL correspondiente al recurso solicitado. LaunchRequestInfo mantiene los datos necesarios en Launch para la ejecución del Callback una vez comprobada la petición en la biblioteca Oscelot.

3. ResourceFolderServlet, InternoResourceFolderServlet, JuegoInternoResourceFolderServlet

Permiten que los ficheros de los juegos incluidos en el desplegable de la plataforma accedan a los recursos necesarios, toda vez que su ubicación no es accesible directamente.

4. ExternStaticFolder, Interfaz

Permiten que los ficheros de los juegos cargados en el TP mediante el módulo de administración accedan a los recursos necesarios, toda vez que su ubicación no es accesible directamente.

4.1.2.7 Autenticación y autorización

1. AuthenticatedUser

Interfaz con el comportamiento de un usuario autenticado. Usuario, LtiUser y UsuarioTemporal implementan esta interfaz.

2. LoginController

Usado en la página de *login*.

3. DatosSesion

Clase que permite acceder a los datos almacenados en la sesión del usuario.

4. UserSession

Managed Bean que extiende DatosSesion (pero gestionada por CDI) con ámbito de sesión de forma que se puede utilizar en los ficheros JSF.

5. UsuarioTemporal

Clase para asignar a los usuarios conectados a una partida mediante el IDP de la partida sin hacer *login* ni conectar por LTI.

4.1.2.8 Conversores (JSF)

Implementan la interfaz **Converter** [61], y realizan conversiones de **Object** a **String** y de **String** a **Object** entre objetos de datos del modelo y una representación de cadena de esos objetos que sea adecuada para su visualización.

1. LocalDateTimeConverter

Permite la utilización de objetos de la clase **LocalDateTime** en las páginas JSF.

2. LocalDateConverter

Permite la utilización de objetos de la clase **LocalDate** en las páginas JSF.

4.1.2.9 Conversores (ORM)

Implementan la interfaz **AttributeConverter** [62], y pueden utilizarse para convertir el estado de los atributos de la entidad en una representación de columna de base de datos y viceversa. Hay que tener en cuenta que se pueden realizar tanto conversiones en las que los tipos de datos de origen y destino son distintos como iguales.

1. BooleanToShortConverter

Convierte entre valores lógicos (**boolean**) en los objetos y entero corto (**Short**) en la base de datos.

2. EstadoPartidaToIntegerConverter

Convierte entre valores de la enumeración **EstadoPartida** en los objetos y entero (**Integer**) en la base de datos.

3. LocalTimeToLongConverter

Convierte entre valores **LocalTime** en los objetos y entero largo (**Long**) en la base de datos.

4. MapStringStringConverter

Convierte entre valores **Map<String, String>** en los objetos y **String** con formato JSON en la base de datos.

5. ModoPartidaToIntegerConverter

Convierte entre valores de la enumeración **ModoPartida** en los objetos y entero (**Integer**) en la base de datos.

6. RolToIntegerConverter

Convierte entre valores de la enumeración **RolUsuario** en los objetos y entero (**Integer**) en la base de datos.

7. TimeZoneConverter

Convierte entre valores `TimeZone` en los objetos y `String` en la base de datos.

4.1.2.10 Desarrollo de la partida

1. `WebsocketController`

Se encarga de la comunicación por WebSocket, tanto la recepción y procesamiento de los mensajes recibidos como del envío de mensajes.

2. `Cronometro`

Temporizador para controlar el tiempo de respuesta a las preguntas y votaciones.

3. `PartidaGlobal`

Datos de las partidas activas, bien en juego o bien activadas por el iniciador para el registro de jugadores previo al inicio.

4. `Participacion`

Datos de los participantes en una partida.

4.1.2.11 Utilidad

1. `AppConfiguration`

Bean de aplicación con los datos de configuración de la aplicación.

2. `AppLoader`

`ServletContextListener` que se inicializa la base de datos en caso de que sea necesario.

3. `Constants`

Clase con valores de constantes usados en el resto de la aplicación.

4. `DateUtil`

Clase con métodos para la conversión entre distintas clases que representan información de fecha y hora.

5. `Descomprime`

Clase con métodos para leer los ficheros de un archivo en formato ZIP.

6. `Encryption`

Clase con métodos para generar, cifrar y comprobar contraseñas.

7. `Messages`

Clase que gestiona la localización (l10n) de los mensajes utilizados en las clases Java de la aplicación.

8. `StringUtil`

Clase con métodos para codificar y decodificar un objeto serializable en formato base64.

4.1.3 Ficheros JSF

Ficheros con *facelets* que proporcionan la interfaz de usuario. Los *facelets* permiten componer una página web mediante el uso de plantillas (*templates*) y con la posibilidad de incorporar a la misma fragmentos reutilizables y parametrizables.

Según el uso de las páginas generadas por estos ficheros podemos clasificarlos en los siguientes grupos:

4.1.3.1 Login

4.1.3.1.1 Página de *login* para inicio de sesión

Página desde la cual se accede al módulo, esta página permite el acceso desde el navegador web, para poder acceder sin LTI.



Figura 14: Inicio de Sesión

- Inicio de sesión correcto: página que se muestra cuando se realiza correctamente el inicio de sesión, en ella se muestra un mensaje de bienvenida.

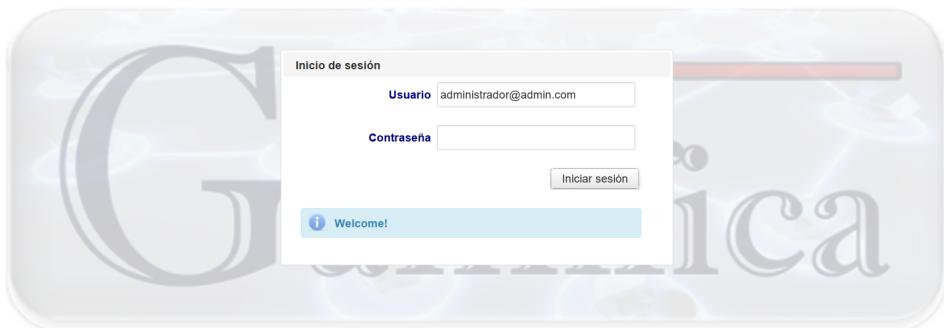


Figura 15: Inicio de Sesión Correcto

- Error en inicio de sesión: página que se muestra cuando no se realiza correctamente el inicio de sesión, en ella se muestra un mensaje de error en el *login*, en el idioma seleccionado por el usuario.

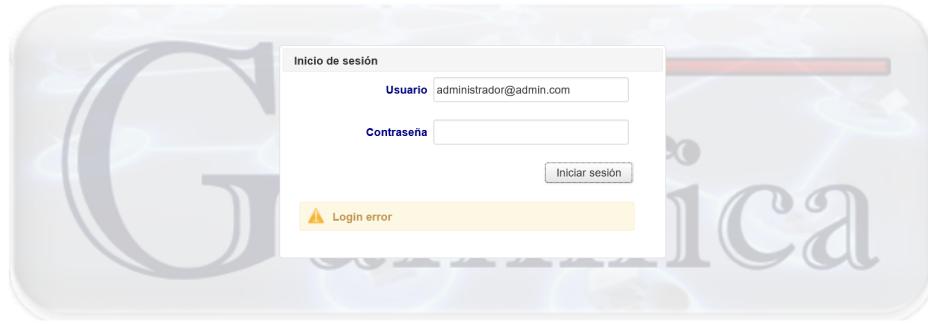


Figura 16: Error en inicio de Sesión

4.1.3.2 Utilidades de usuario

4.1.3.2.1 Preferencias

Menú de preferencias que se puede abrir desde cualquier página del módulo, una vez realizado el login. Esto permite la edición de algunas opciones como el idioma, el huso horario o la contraseña.

ID	Juego	Profesor	Acciones
130	SYSjug-batalla_naval	super@admin.com	[Actions]
129	SYSjug-batalla_naval	super@admin.com	[Actions]
128	SYSjug-batalla_naval	super@admin.com	[Actions]
127	SYSjug-batalla_naval	super@admin.com	[Actions]
126	SYSjug-batalla_naval	jesusman1 @ IES Joaquín Turina	[Actions]
125	SYSjug-batalla_naval	alira-05 @ IES Joaquín Turina	[Actions]
123	SYSjug-batalla_naval	super@admin.com	[Actions]
122	SYSjug-batalla_naval	super@admin.com	[Actions]
121	SYSjug-batalla_naval	super@admin.com	[Actions]
120	SYSjug-test	super@admin.com	[Actions]

Figura 17: Preferencias

4.1.3.3 Administración

4.1.3.3.1 Estructura del menú

Para el módulo administración se ha creado un menú donde se han agrupado las distintas páginas de administración con la estructura que se muestra a continuación.

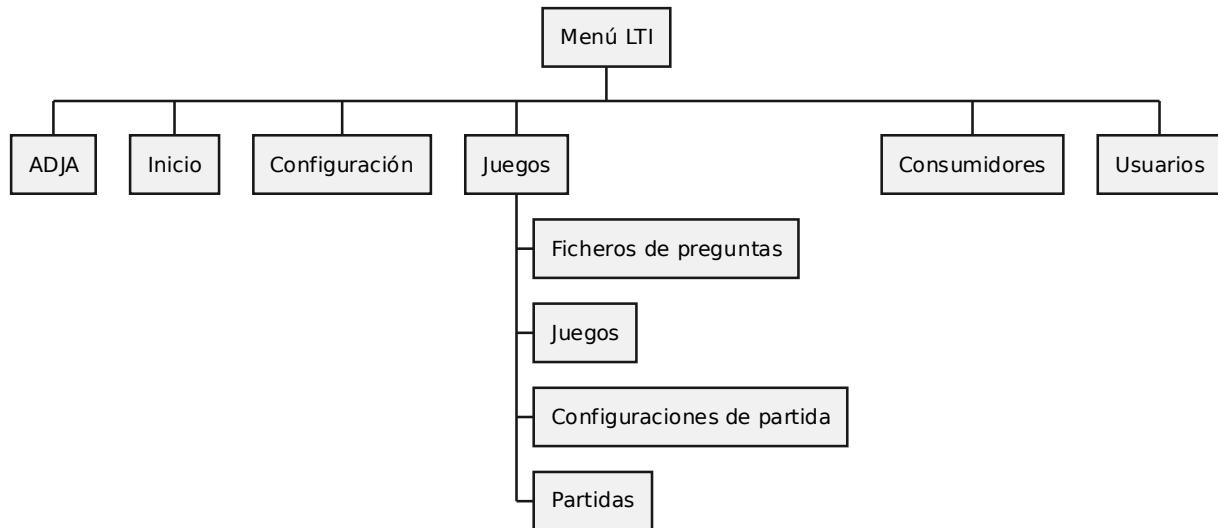


Figura 18: Estructura del menú de administración

A continuación se describen las páginas a las que se puede acceder desde el menú.

4.1.3.3.2 Configuración

4.1.3.3.2.1 Lista de propiedades

Muestra la lista de propiedades, rootPath (ubicación en el servidor de los ficheros que son necesarios para el funcionamiento de los juegos) y gamePath (ubicación en el servidor, como subdirectorio de rootPath de los juegos subidos mediante el módulo de administración), visibles y editables solo desde el perfil de superusuario. Se accede mediante la opción de menú Configuración.

Clave	Valor	Acciones
gamePath	00-juegos-on-line/juegos	
rootPath	/home/itssh/tg_AlvaroSantiago_Cuenca	

Figura 19: Lista de propiedades

4.1.3.3.2.2 Visualización y edición de propiedades

Editar Propiedad: página desde la que se permite la visualización y edición de gamePath y rootPath.

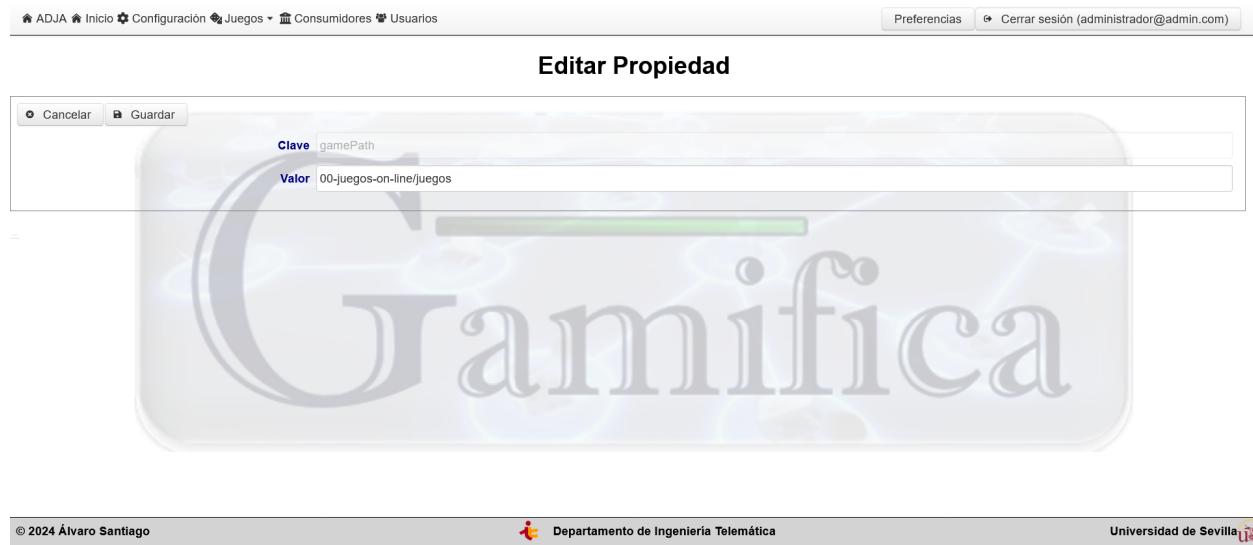


Figura 20: Editar Propiedades

Ver propiedad: página que permite ver una propiedad pero sin poder editarla.

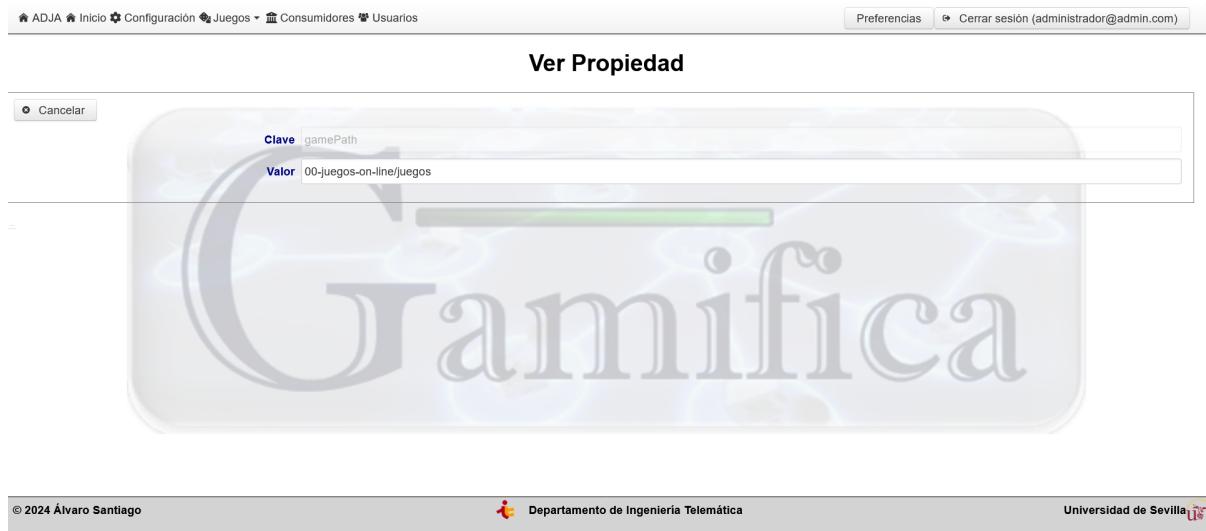


Figura 21: Ver Propiedad

4.1.3.3.3 Juegos

Dentro del desplegable Juegos, encontramos las siguientes opciones:

- Ficheros de preguntas.
- Juegos.
- Configuraciones de partida.
- Partidas.

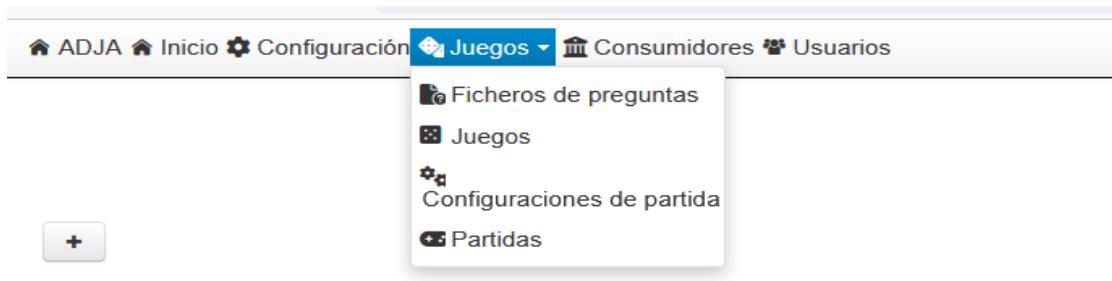


Figura 22: Menú desplegable Juegos

4.1.3.3.4 Ficheros de preguntas

4.1.3.3.4.1 Lista de ficheros de preguntas

Página que muestra los ficheros de preguntas que están cargados en la plataforma y que podrán ser seleccionados como fichero de preguntas para los juegos y partidas. Sólo se muestran los ficheros de preguntas a los que el usuario conectado tiene acceso.

La captura de pantalla muestra una lista de ficheros de preguntas cargados en la plataforma. Los datos se presentan en una tabla con las siguientes columnas: Id, Profesor, Título, Tipo de respuestas, Público, Consumidor y Acciones. Se observan cuatro registros:

ID	Profesor	Título	Tipo de respuestas	Público	Consumidor	Acciones
4	teacher @ IES Joaquin Turina	Cultura general 1	Opciones (A.B.C...)	✓	IES Joaquin Turina	
6	teacher @ IES Joaquin Turina	Cultura general 1b	Opciones (A.B.C...)	✗	IES Joaquin Turina	
1	teacher @ IES Joaquin Turina	La Célula 1	Opciones (A.B.C...)	✗	IES Joaquin Turina	
5	super	La Célula 1	Opciones (A.B.C...)	✓		

En la parte inferior de la pantalla se ven los créditos: © 2024 Álvaro Santiago, Departamento de Ingeniería Telemática y Universidad de Sevilla.

Figura 23: Lista de ficheros de preguntas

4.1.3.3.4.2 Añadir Fichero de preguntas:

Página desde la cual se pueden añadir ficheros de preguntas. En esta página se selecciona tanto el fichero a cargar como las características del mismo, siendo la más importante Tipo de Respuestas ya que indicará que tipo de respuesta está configurada en el fichero de preguntas cargado.

Añadir Fichero de preguntas

Título: Título
Profesor: administrador@admin.com

Características

Asignatura: Asignatura
Tema: Tema
Curso: Curso
Tipo de respuestas: -o- ABCD 123 TXT Rel
Público: No

Observaciones: Observaciones

Fichero

Nombre de fichero: Nombre
Subir fichero: + Seleccionar fichero... Subir Cancelar

© 2024 Álvaro Santiago | Departamento de Ingeniería Telemática | Universidad de Sevilla

Figura 24: Añadir Fichero de preguntas

4.1.3.3.4.3 Editar fichero de preguntas

Página desde la que se permite editar un fichero de preguntas previamente cargado.

Editar Fichero de preguntas #4

Título: Cultura general 1
Profesor: teacher @ IES Joaquin Turina

Características

Asignatura: Asignatura
Tema: Tema
Curso: Curso
Tipo de respuestas: -o- ABCD 123 TXT Rel
Público: Si

Observaciones: Observaciones

Fecha de creación: 2022-05-30 **F. actualización:** 2024-02-16

Fichero

Nombre de fichero: preguntas2.txt
Contenido: AUTOR/A: Pedro García

© 2024 Álvaro Santiago | Departamento de Ingeniería Telemática | Universidad de Sevilla

```
Esclavos
Plebeyos
Patricios
```

Figura 25: Editar Fichero de Preguntas

4.1.3.3.4.4 Ver Fichero de Preguntas

Página desde la cual se visualiza un fichero de preguntas ya cargado pero sin mostrar la posibilidad de editarlo.

Ver Fichero de preguntas #4

Características

Título: Cultura general 1
Profesor: teacher @ IES Joaquin Turina

Asignatura: Asignatura
Curso: Curso
Tipo de respuestas: ABCD, 123, TXT, Rel
Tema: Tema
Público: Sí

Observaciones: Observaciones

Fecha de creación: 2022-05-30 **F. actualización:** 2024-02-16

Fichero

Nombre de fichero: preguntas2.txt
Contenido: AUTOR/A: Pedro Garcia

© 2024 Álvaro Santiago Departamento de Ingeniería Telemática Universidad de Sevilla

Esclavos
Plebeyos
Patricios

Figura 26: Ver Fichero de Preguntas

4.1.3.3.5 Juegos

4.1.3.3.5.1 Lista de Juegos

Página en la cual se muestra la lista de juegos disponibles para crear una partida.

Lista de juegos

ID	Nombre	Activo	Acciones
15	artificeros	x	
17	jug-artificeros	x	
18	jug-batalla_naval	x	
20	jug-batalla_naval-LTI	x	
22	jug-batalla_naval-LTI (b)	x	
19	jug-bombilla	x	
21	jug-bombilla-LTI	x	
23	jug-bombilla-LTI (b)	x	
102	SYSjug-50x15	x	
109	SYSjug-50x15-lite	x	

© 2024 Álvaro Santiago Departamento de Ingeniería Telemática Universidad de Sevilla

Figura 27: Lista de Juegos

4.1.3.3.5.2 Añadir Juego

Página a través de la cual podemos añadir un nuevo juego a la plataforma, así como configurarle datos adicionales.

The screenshot shows the 'Añadir Juego' (Add Game) form. At the top, there are navigation links: ADJA, Inicio, Configuración, Juegos, Consumidores, Usuarios, Preferencias, and Cerrar sesión (super). The main title is 'Añadir Juego'. The form has several sections:

- Datos adicionales:** Includes fields for Nombre (Nombre), Descripción (Descripción), Etapa (GEN, EI, EP, ESO, BAC, UNIV), Preguntas ilimitadas (no), Carga completa (no), Número mínimo preguntas, and Estilo de Juego (Clásico, TV, Deporte, AJDA).
- Ubicación de los recursos:** A section for Fichero Zip with fields for Nombre and Ruta, and a 'Subir fichero' button.
- Parámetros:** Sections for Respuestas (JSON), Fijos, Opcionales (with a table for Clave and Valor), Equipos (JSON), and Otros datos (JSON).

Figura 28: Añadir Juego

4.1.3.3.5.3 Editar Juego

Página que permite la edición de un juego previamente cargado.

ADJA Inicio Configuración Juegos Consumidores Usuarios Preferencias Cerrar sesión (super)

Editar Juego #20

Cancelar Guardar Activar

Nombre: jug-batalla_naval-LTI
Descripción: Descripción
página juego: jug-batalla_naval-LTI/jugar.html

Datos adicionales

Etapas: GEN (selected), EI, EP, ESO, BAC, UNIV
Preguntas ilimitadas: no
N.º de ficheros: entre 1 y 1
Tipo de respuestas: 0-, ABCD, 123, TXT, Rel
Carga completa: no
N.º min. preguntas:
Estilo de juego: Clásico, TV, Deporte, AJDA

Ubicación de los recursos

Fichero Zip:
Nombre: jug-batalla_naval-LTI.zip
Ruta: **Subir fichero:**

Parámetros

Respuestas (JSON):
Fijos:
Opcionales:

#	Clave	Valor
Sin datos		
Añadir		

Equipos (JSON):
Otros datos (JSON):

Partidas:

ID	Profesor	Título	Acciones
Sin datos			

© 2024 Álvaro Santiago  Departamento de Ingeniería Telemática Universidad de Sevilla 

Figura 29: Editar Juego

4.1.3.3.5.4 Ver Juego

Página que permite la visualización de un juego cargado, pero sin permitir la edición del mismo.

The screenshot displays the 'Ver Juego #20' (View Game #20) configuration page. At the top, there are navigation links: ADJA, Inicio, Configuración, Juegos, Consumidores, Usuarios, Preferencias, and Cerrar sesión (super). The main title is 'Ver Juego #20'.

Datos principales:

- Nombre: jug-batalla_naval-LTI
- Descripción: Descripción
- página juego: jug-batalla_naval-LTI/jugar.html

Opciones adicionales:

- Etapa: GEN (selected), EI, EP, ESO, BAC, UNIV
- Preguntas ilimitadas: no
- Tipo de respuestas: 0-, ABCD, 123, TXT, Rel
- Carga completa: no
- Núm. mínimo preguntas: (dropdown)
- Núm. de ficheros: entre 1 y 1
- Estilo de juego: Clásico, TV, Deporte, AJDA

Ubicación de los recursos:

Fichero Zip

- Nombre: jug-batalla_naval-LTI.zip
- Ruta: (empty)
- Subir fichero: + Seleccionar fichero ..., Subir, Cancelar

Parámetros:

- Respuestas (JSON): (empty)
- Fijos: (empty)
- Opcionales: (table)

#	Clave	Valor
Sin datos		
Añadir		
- Equipos (JSON): (empty)
- Otros datos (JSON): (empty)

Partidas:

+ (button)

ID	Profesor	Título	Acciones
Sin datos			

© 2024 Álvaro Santiago Departamento de Ingeniería Telemática Universidad de Sevilla

Figura 30: Ver Juego

4.1.3.3.6 Configuraciones de Partida

4.1.3.3.6.1 Lista Configuraciones de Partidas

Página en la cual se muestra las diferentes configuraciones para crear partidas en la plataforma a las que el usuario conectado tiene acceso.

Id	Título	Juego	Profesor	Acciones
12	Plantilla 2	Test	super@admin.com	
11	Plantilla 1	Test	super@admin.com	

Figura 31: Lista Configuraciones de Partidas

4.1.3.3.6.2 Añadir Configuración de Partida

Página desde la cual creamos una nueva configuración de partida, donde le asignamos los parámetros necesarios.

Figura 32: Añadir Configuración de Partida

4.1.3.3.6.3 Editar Configuración de Partida

Página desde la cual se edita una configuración de partida existente.

The screenshot displays the 'Editar Configuración de partida #' (Edit Game Configuration) interface. At the top, there's a navigation bar with links to ADJA, Inicio, Configuración, Juegos, Consumidores, and Usuarios. On the right, there are buttons for Preferencias and Cerrar sesión (super).

Juego: Test
Id: 12
Factor de corrección: 30 %
Curso: 2
Tema: Ondas
Fecha de creación: 2023-08-18

Profesor: jesusman1 @ IES Joaquin Turina
Titulo: Plantilla 2
Etapa: GEN EI EP ESO BAC UNIV
Asignatura: Fisica
Tiempo de respuesta: 65 segundos

Ficheros de preguntas:

- La Célula 1
- Matemáticas
- Cultura general 1
- La Célula 1b
- Cultura general 1b

Equipos:

#	Nombre	Número de jugadores por equipo	Acciones
Sin datos			

Partidas:

#	Juego	Profesor	Título	Acciones
Sin datos				

At the bottom, there are copyright notices: © 2024 Álvaro Santiago, Departamento de Ingeniería Telemática, and Universidad de Sevilla.

Figura 33: Editar Configuración de Partida

4.1.3.3.6.4 Ver Configuración de Partida

Página que permite la visualización de una partida pero sin poder editar ningún parámetro.

Figura 34: Ver Configuración de Partida

4.1.3.3.7 Partidas

4.1.3.3.7.1 Lista de Partidas

Página que muestra las partidas configuradas a las que el usuario conectado tiene acceso.

Id	Juego	Profesor	Título	Acciones
130	SYSjug-batalla_naval	super@admin.com	(LTi\alva-02)[34436797] Plantilla 2 2023-08-18 15:17:37	
129	SYSjug-batalla_naval	super@admin.com	(LTi\alvaro)[34436797] Plantilla 2 2023-08-18 15:17:37	
128	SYSjug-batalla_naval	super@admin.com	(LTi\jesusman1)[34436797] Plantilla 2 2023-08-18 15:17:37	
127	SYSjug-batalla_naval	super@admin.com	(LTi\alva-01)[34436797] Plantilla 2 2023-08-18 15:17:37	
126	SYSjug-batalla_naval	jesusman1 @ IES Joaquin Turina	Prueba centralizada multijugador	
125	SYSjug-batalla_naval	alva-05 @ IES Joaquin Turina	Prueba distribuida	
123	SYSjug-batalla_naval	super@admin.com	(LTi\teacher)[34436797] Plantilla 2 2023-08-18 15:17:37	
122	SYSjug-batalla_naval	super@admin.com	(LTi\learner1)[34436797] Plantilla 2 2023-08-18 15:17:37	
121	SYSjug-batalla_naval	super@admin.com	(LTi\teacher)[34436797] Plantilla 2 2023-08-18 15:17:37	
120	SYSjug-test	super@admin.com	(LTi\teacher)[34436797] Plantilla 2 2023-08-18 15:17:37	

Figura 35: Lista de Partidas

4.1.3.3.7.2 Añadir Partida

Página desde la cual se permite añadir una partida nueva, indicando la configuración de partida a utilizar, el juego y los demás parámetros necesarios.

Añadir Partida

Configuración de partida: Seleccionar

Juego: Seleccionar

Profesor: Seleccionar

Iniciador: Seleccionar

Título: Título

Clave de entrega: DDRr=#c@

Inicio / Fin: Fecha actualización

Finalizada: no Activo: no Outcome: no

Permisos para iniciador: no Con espectadores: no Pública: no

LTI: no status: no Máx. repeticiones: 3

Cuenta: 1 Modo partida: CST CLTI CMJ DST

Estado: Conf REG Jugando Fin

Ficheros de preguntas:

- La Célula 1
- Matematicas
- Cultura general 1
- La Célula 1
- Cultura general 1b

Datos adicionales:

Código: 4434095792	Etapa: GEN EI EP ESO BAC UNIV
Curso:	Asignatura:
Tema:	Fichero continuación:
Factor de corrección: 50 %	Tiempo de respuesta: 60 segundos
Equipos: no	Núm. jugadores: entre 1 y 4

Figura 36: Añadir Partida

4.1.3.3.7.3 Editar Partida

Página desde la cual se permite la edición de una partida existente.

ADJA Inicio Configuración Juegos Consumidores Usuarios Preferencias Cerrar sesión (super)

Editar Partida #125

Cancelar Guardar

Juego	jug-bombilla_video	Iniciador	learner2 @ IES Joaquin Turina																				
Profesor	alva-05 @ IES Joaquin Turina																						
Título	Prueba distribuida																						
Clave de entrega	asd																						
Inicio / Fin		Fecha actualización	2024-07-31 21:32:46																				
Finalizada	no	Activo	sí																				
Permisos para iniciador	sí	Con espectadores	sí																				
LTI	sí	status	no																				
Cuenta	1	Máx. repeticiones	20																				
Estado	Conf REG Jugando Fin	Modo partida	CST CLTI CMJ DST																				
Ficheros de preguntas																							
<input type="checkbox"/> La Célula 1 <input checked="" type="checkbox"/> Matematicas <input type="checkbox"/> Cultura general 1 <input type="checkbox"/> La Célula 1 <input type="checkbox"/> Cultura general 1b																							
Datos adicionales <table border="1"> <tr> <td>Código</td> <td>28178104813</td> <td>Etapa</td> <td>GEN EI EP ESO BAC UNIV</td> </tr> <tr> <td>Curso</td> <td></td> <td>Asignatura</td> <td></td> </tr> <tr> <td>Tema</td> <td></td> <td>Fichero continuación</td> <td>Partida del juego BATALLA NAVAL1alva-01alva-03020C</td> </tr> <tr> <td>Factor de corrección</td> <td>50 %</td> <td>Tiempo de respuesta</td> <td>20 segundos</td> </tr> <tr> <td>Equipos</td> <td>no</td> <td>Núm. jugadores</td> <td>2 jugadores</td> </tr> </table>				Código	28178104813	Etapa	GEN EI EP ESO BAC UNIV	Curso		Asignatura		Tema		Fichero continuación	Partida del juego BATALLA NAVAL1alva-01alva-03020C	Factor de corrección	50 %	Tiempo de respuesta	20 segundos	Equipos	no	Núm. jugadores	2 jugadores
Código	28178104813	Etapa	GEN EI EP ESO BAC UNIV																				
Curso		Asignatura																					
Tema		Fichero continuación	Partida del juego BATALLA NAVAL1alva-01alva-03020C																				
Factor de corrección	50 %	Tiempo de respuesta	20 segundos																				
Equipos	no	Núm. jugadores	2 jugadores																				
Parámetros <table border="1"> <tr> <td>Fijos</td> <td></td> </tr> <tr> <td>Opcionales</td> <td> <table border="1"> <thead> <tr> <th>#</th> <th>Clave</th> <th>Valor</th> </tr> </thead> <tbody> <tr> <td>Sin datos</td> <td></td> <td></td> </tr> <tr> <td><input type="button"/> Añadir</td> <td></td> <td></td> </tr> </tbody> </table> </td> </tr> </table>				Fijos		Opcionales	<table border="1"> <thead> <tr> <th>#</th> <th>Clave</th> <th>Valor</th> </tr> </thead> <tbody> <tr> <td>Sin datos</td> <td></td> <td></td> </tr> <tr> <td><input type="button"/> Añadir</td> <td></td> <td></td> </tr> </tbody> </table>	#	Clave	Valor	Sin datos			<input type="button"/> Añadir									
Fijos																							
Opcionales	<table border="1"> <thead> <tr> <th>#</th> <th>Clave</th> <th>Valor</th> </tr> </thead> <tbody> <tr> <td>Sin datos</td> <td></td> <td></td> </tr> <tr> <td><input type="button"/> Añadir</td> <td></td> <td></td> </tr> </tbody> </table>	#	Clave	Valor	Sin datos			<input type="button"/> Añadir															
#	Clave	Valor																					
Sin datos																							
<input type="button"/> Añadir																							
<input type="button"/> Jugadores		<input type="button"/> Usuarios LTI																					
<table border="1"> <thead> <tr> <th>Id</th> <th>Usuario</th> <th>Acciones</th> </tr> </thead> <tbody> <tr> <td>36</td> <td>learner2 @ IES Joaquin Turina</td> <td></td> </tr> <tr> <td>34</td> <td>learner1 @ IES Joaquin Turina</td> <td></td> </tr> </tbody> </table>				Id	Usuario	Acciones	36	learner2 @ IES Joaquin Turina		34	learner1 @ IES Joaquin Turina												
Id	Usuario	Acciones																					
36	learner2 @ IES Joaquin Turina																						
34	learner1 @ IES Joaquin Turina																						

Figura 37: Editar Partida

4.1.3.3.7.4 Ver Partida

Página desde la cual se permite la visualización de una partida y de los parámetros configurados en la misma pero sin permiso para edición.

ADJA Inicio Configuración Juegos Consumidores Usuarios Preferencias Cerrar sesión (super)

Ver Partida #125

Juego	jug-bombilla_video	Iniciador	learner2 @ IES Joaquin Turina																					
Profesor	alva-05 @ IES Joaquin Turina																							
Título	Prueba distribuida																							
Clave de entrega	asd																							
Inicio / Fin																								
	Fecha actualización 2024-07-31 21:32:46																							
Finalizada	no	Activo	sí																					
Outcome	sí																							
Permisos para iniciador	sí	Con espectadores	sí																					
LTI	sí	status	no																					
Cuenta	1	Máx. repeticiones	20																					
Estado	Conf REG Jugando Fin																							
Ficheros de preguntas																								
<input type="checkbox"/> La Célula 1 <input checked="" type="checkbox"/> Matematicas <input type="checkbox"/> Cultura general 1 <input type="checkbox"/> La Célula 1 <input type="checkbox"/> Cultura general 1b																								
- Datos adicionales <table border="0"> <tr> <td>Código</td> <td>28178104813</td> <td>Etapa</td> <td>GEN EI EP ESO BAC UNIV</td> </tr> <tr> <td>Curso</td> <td></td> <td>Asignatura</td> <td></td> </tr> <tr> <td>Tema</td> <td></td> <td>Fichero continuación</td> <td>Partida del juego BATALLA NAVAL1alva-01alva-03020C</td> </tr> <tr> <td>Factor de corrección</td> <td>50 %</td> <td>Tiempo de respuesta</td> <td>20 segundos</td> </tr> <tr> <td>Equipos</td> <td>no</td> <td>Núm. jugadores</td> <td>2 jugadores</td> </tr> </table>					Código	28178104813	Etapa	GEN EI EP ESO BAC UNIV	Curso		Asignatura		Tema		Fichero continuación	Partida del juego BATALLA NAVAL1alva-01alva-03020C	Factor de corrección	50 %	Tiempo de respuesta	20 segundos	Equipos	no	Núm. jugadores	2 jugadores
Código	28178104813	Etapa	GEN EI EP ESO BAC UNIV																					
Curso		Asignatura																						
Tema		Fichero continuación	Partida del juego BATALLA NAVAL1alva-01alva-03020C																					
Factor de corrección	50 %	Tiempo de respuesta	20 segundos																					
Equipos	no	Núm. jugadores	2 jugadores																					
- Parámetros <table border="0"> <tr> <td>Fijos</td> <td></td> </tr> <tr> <td>Opcionales</td> <td> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>#</th> <th>Clave</th> <th>Valor</th> </tr> </thead> <tbody> <tr> <td colspan="3">Sin datos</td> </tr> </tbody> </table> <input type="button" value="Añadir"/> </td> </tr> </table>					Fijos		Opcionales	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>#</th> <th>Clave</th> <th>Valor</th> </tr> </thead> <tbody> <tr> <td colspan="3">Sin datos</td> </tr> </tbody> </table> <input type="button" value="Añadir"/>	#	Clave	Valor	Sin datos												
Fijos																								
Opcionales	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>#</th> <th>Clave</th> <th>Valor</th> </tr> </thead> <tbody> <tr> <td colspan="3">Sin datos</td> </tr> </tbody> </table> <input type="button" value="Añadir"/>	#	Clave	Valor	Sin datos																			
#	Clave	Valor																						
Sin datos																								
<input type="button" value="Jugadores"/> <input type="button" value="Usuarios LTI"/> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th>Id</th> <th>Usuario</th> <th>Acciones</th> </tr> </thead> <tbody> <tr> <td>36</td> <td>learner2 @ IES Joaquin Turina</td> <td></td> </tr> <tr> <td>34</td> <td>learner1 @ IES Joaquin Turina</td> <td></td> </tr> </tbody> </table>					Id	Usuario	Acciones	36	learner2 @ IES Joaquin Turina		34	learner1 @ IES Joaquin Turina												
Id	Usuario	Acciones																						
36	learner2 @ IES Joaquin Turina																							
34	learner1 @ IES Joaquin Turina																							

Figura 38: Ver Partida

4.1.3.3.8 Consumidores

4.1.3.3.8.1 Lista Consumidores

Página accesible sólo para el superusuario, en ella se muestran los consumidores configurados en la plataforma.

Nombre	Nombre LTI	Versión consumidor LTI	Versión LTI	Acciones
IES Joaquin Turina	mTurina formación	moodle-2019111803.02	LTI-1p0	
turina Test	mTurina formación	moodle-2019111803.02	LTI-1p0	

Figura 39: Lista de Consumidores

4.1.3.3.8.2 Añadir Consumidor

Página accesible sólo para el superusuario, página desde la cual se nos permite añadir un nuevo consumidor a la plataforma.

Figura 40: Añadir Consumidor

4.1.3.3.8.3 Editar Consumidor

Página accesible sólo para el superusuario, en la cual se nos permite editar un consumidor ya existente.

The screenshot displays the 'Editar Consumidor' (Edit Consumer) page. At the top, there are tabs for 'Cancelar' (Cancel) and 'Guardar' (Save). The main form fields include:

- Nombre:** IES Joaquin Turina
- Clave LTI:** 073aeaf-fab7-49cc-846a-87280060d68d
- Secreto:** tlPtxxs4X5
- Usuario LMS:** Moodle
- Activo:** Sí
- Activo desde/hasta:** 2023-06-04 09:00:00
- Protegido:** No
- Ruta CSS LTI:** Ruta CSS LTI
- URL herramienta:** http://82.223.18.147:80/gamificación-tfg_AlvaroSantiago_Cuenca/games

Below the main form, there's a section for 'Parámetros personalizados' (Personalized parameters) with a 'management' field set to true. The footer contains copyright information for 2024 Álvaro Santiago, the Department of Telematic Engineering, and the University of Seville.

The bottom part of the screenshot shows a table titled 'Información LTI' (LTI Information) with the following data:

GUID LTI	mfor.iesjoaquiturina.es	Nombre LTI	mTurina formación	
Versión consumidor LTI	moodle-2019111803.02	Versión LTI	LTI-1p0	
Fecha de creación	2023-11-09 08:00:43	F. actualización	2024-07-02 21:40:18	
			Último acceso	2024-07-02 00:00:00

At the very bottom, there are tabs for 'Usuarios', 'Contextos LTI', and 'Nonces LTI', with the 'Usuarios' tab currently selected. A table below shows a list of LTI users:

Id usuario LTI	Rol	Acciones
103	LTI	<input type="button" value=""/> <input type="button" value="i"/>
118	LTI	<input type="button" value=""/> <input type="button" value="i"/>
119	LTI	<input type="button" value=""/> <input type="button" value="i"/>
120	LTI	<input type="button" value=""/> <input type="button" value="i"/>
121	LTI	<input type="button" value=""/> <input type="button" value="i"/>
122	LTI	<input type="button" value=""/> <input type="button" value="i"/>
123	LTI	<input type="button" value=""/> <input type="button" value="i"/>
2	LTI	<input type="button" value=""/> <input type="button" value="i"/>
3	LTI	<input type="button" value=""/> <input type="button" value="i"/>
4	LTI	<input type="button" value=""/> <input type="button" value="i"/>

Figura 41: Editar Consumidor

4.1.3.3.8.4 Ver Consumidor

Página accesible sólo para el superusuario, desde la cual se permite la visualización de un consumidor existente pero sin opción de edición.

ADJA Inicio Configuración Juegos Consumidores Usuarios Preferencias Cerrar sesión (administrador@admin.com)

Ver Consumidor

Cancelar

Nombre	IES Joaquin Turina	Clave LTI	073aeaf-fab7-49cc-846a-87280060d68d
Secreto	tiPfxs4X5	Usuario LMS	Moodle
Activo	sí	Activo desde/hasta	2023-06-04 09:00:00
- Datos adicionales			
Protegido	<input type="checkbox"/> no	Ruta CSS LTI	Ruta CSS LTI
- Configuración LTI			
URL herramienta	http://82.223.18.147:80/gamificación-tfg_AlvaroSantiago_Cuenca/games		

Parámetros personalizados

management	true	Acceso a la administración de la plataforma
------------	------	---

© 2024 Álvaro Santiago  Departamento de Ingeniería Telemática Universidad de Sevilla 

[- Información LTI](#)

GUID LTI	mfor.iesjoaquinTurina.es	Nombre LTI	mTurina formación	
Versión consumidor LTI	moodle-2019111803.02	Versión LTI	LTI-1p0	
Fecha de creación	2023-11-09 08:00:43	F. actualización	2024-07-02 21:40:18	
			Último acceso	2024-07-02 00:00:00

[Usuarios](#) [Contextos LTI](#) [Nonce LTI](#)

[+](#)

1 de 2	1 2	10
Id usuario LTI	Rol	Acciones
103	LTI	 
118	LTI	 
119	LTI	 
120	LTI	 
121	LTI	 
122	LTI	 
123	LTI	 
2	LTI	 
3	LTI	 
4	LTI	 

1 de 2 1 2 10

Figura 42: Ver Consumidor

4.1.3.3.9 Usuarios

4.1.3.3.9.1 Lista Usuarios

Página desde la cual se muestran los usuarios, para un usuario profesor se le mostrarán únicamente los usuarios de su institución, para un superusuario se mostrarán todos.

Consumidor	Id usuario LTI	Rol	Acciones
IES Joaquin Turina	103	LTI	
turina Test	103	LTI	
turina Test	118	LTI	
IES Joaquin Turina	118	LTI	
IES Joaquin Turina	119	LTI	
IES Joaquin Turina	120	LTI	
IES Joaquin Turina	121	LTI	
IES Joaquin Turina	122	LTI	
IES Joaquin Turina	123	LTI	
IES Joaquin Turina	2	LTI	

1 de 2 | 1 2 | 10 | © 2024 Álvaro Santiago | Departamento de Ingeniería Telemática | Universidad de Sevilla

Figura 43: Lista de usuarios

4.1.3.3.9.2 Añadir Usuario

Página desde la que se permite añadir un nuevo usuario a la plataforma.

Añadir Usuario

<input type="button" value="Cancelar"/>	<input type="button" value="Guardar"/>
Consumidor	Username
Contraseña	Repetir contraseña
Rol	Zona horaria
Superusuario	Seleccionar
Nombre	Apellidos
E-mail	País
Fecha de registro	Último acceso

© 2024 Álvaro Santiago | Departamento de Ingeniería Telemática | Universidad de Sevilla

Figura 44: Añadir Usuario

4.1.3.3.9.3 Editar Usuario

Página desde la cual se permite la edición de un usuario existente.

Editar Usuario

Consumidor: IES Joaquín Turina
Rol: LTI
Nombre: Alvaro
E-mail: alvarosanttiagocuenca@gmail.com
Fecha de registro: 2024-05-22 17:58:06

Username: 103
Zona horaria: Seleccionar
Apellidos: Santiago Cuenca
País:
Último acceso: 2024-07-05 10:01:59

Datos plataforma EV

Usuario LMS: alvaro
Avatar:

Consumidor	ID	Nombre	Gamificación	Detalles
IES Joaquín Turina	103	alvaro	Gamificación: Partida distribuida	i
IES Joaquín Turina	103	alvaro	Gamificación: Partida multijugador	i
IES Joaquín Turina	103	alvaro	Gamificación: Partida LTI/mono	i

Figura 45: Editar Usuario

4.1.3.3.9.4 Ver Usuario

Página desde la que se permite la visualización pero no la edición de un usuario existente.

Ver Usuario

Consumidor: IES Joaquín Turina
Rol: LTI
Nombre: Alvaro
E-mail: alvarosanttiagocuenca@gmail.com
Fecha de registro: 2024-05-22 17:58:06

Username: 103
Zona horaria: Seleccionar
Apellidos: Santiago Cuenca
País:
Último acceso: 2024-07-05 10:01:59

Datos plataforma EV

Usuario LMS: alvaro
Avatar:

Consumidor	ID	Nombre	Gamificación	Detalles
IES Joaquín Turina	103	alvaro	Gamificación: Partida distribuida	i
IES Joaquín Turina	103	alvaro	Gamificación: Partida multijugador	i
IES Joaquín Turina	103	alvaro	Gamificación: Partida LTI/mono	i

Figura 46: Ver Usuario

4.1.3.4 Juego

Páginas utilizadas para el desarrollo de las partidas. Las partidas pueden desarrollarse en las modalidades de monojugador y multijugador, como se indica en el apartado ADJA de del capítulo Estado del arte. Vamos a ver a continuación el funcionamiento de cada modalidad para los distintos participantes en las mismas.

4.1.3.4.1 Multijugador

4.1.3.4.1.1 Iniciador

Es el encargado de la gestión de la partida, será habitualmente el profesor. Es el que debe iniciar la partida una vez estén todos los jugadores preparados, además podrá realizar cambios en la configuración de los equipos realizada por los jugadores.

Durante el desarrollo de la partida será el encargado de ir avanzando con las preguntas e ir avanzando el desarrollo del juego.

En su pantalla se mostrará la pantalla del juego con las posibles respuestas y el desarrollo del mismo.



Figura 47: Pantalla inicial Iniciador Partida Multijugador

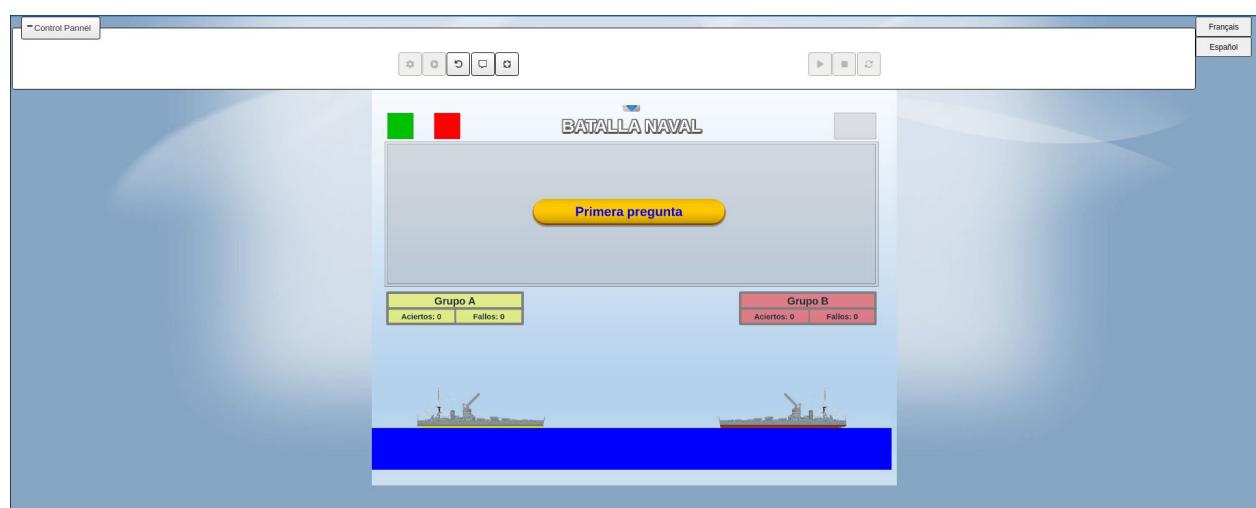


Figura 48: Pantalla Iniciador Partida Multijugador

4.1.3.4.1.2 Jugador

Serán habitualmente los alumnos. Les aparecerá un panel con las posibles respuestas a las preguntas que se vayan mostrando en la pantalla del iniciador.



Figura 49: Pantalla Inicial Alumno/Jugador Partida Multijugador



Figura 50: Pantalla Panel de respuestas Alumno/Jugador Partida Multijugador



Figura 51: Pantalla resultado Pregunta Alumno/Jugador en Partida Multijugador

4.1.3.4.1.3 Espectador

En su pantalla se podrá ver el desarrollo de la partida pero no podrá intervenir en la misma.

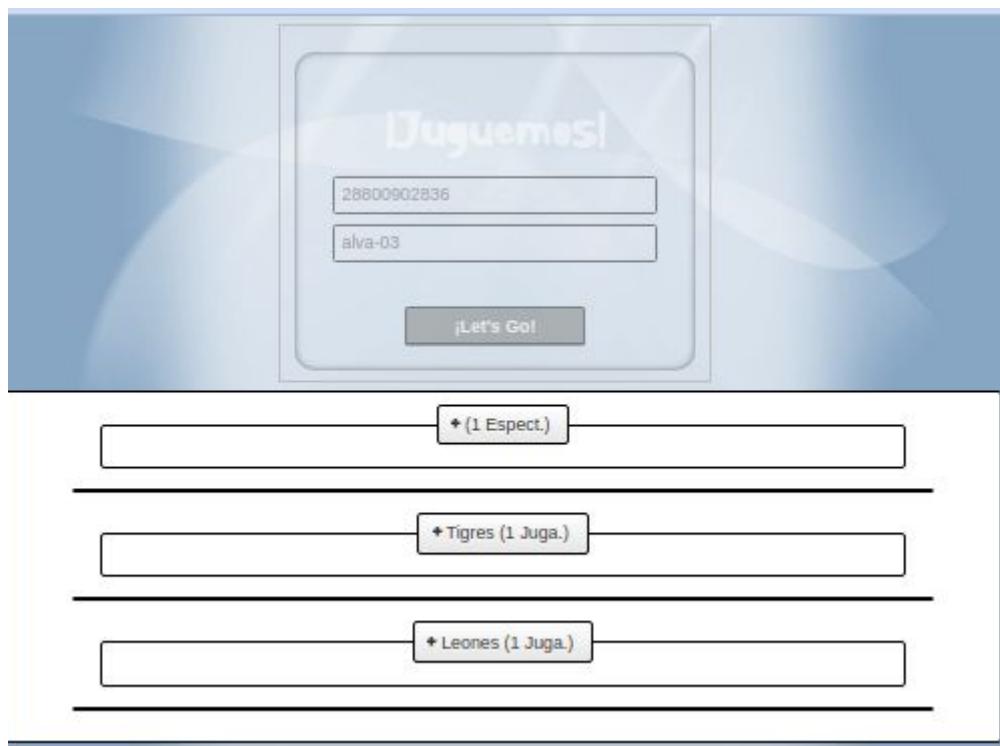


Figura 52: Pantalla Inicial Espectador Partida Multijugador

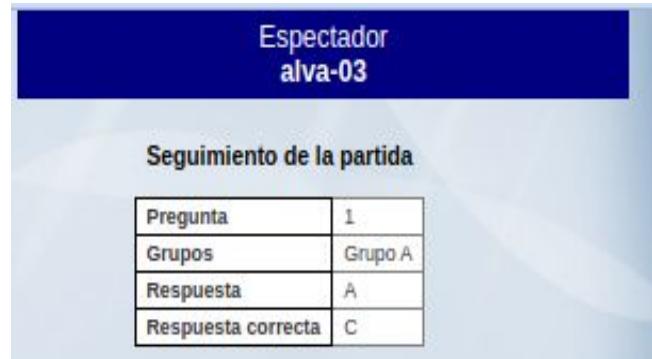


Figura 53: Pantalla Seguimiento Espectador Partida Multijugador I

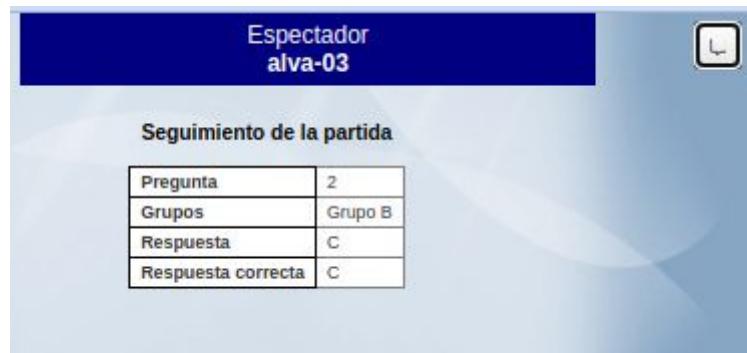


Figura 54: Pantalla Seguimiento Espectador Partida Multijugador II

4.1.3.4.2 Distribuidas

Este tipo de partidas se desarrollarán sin la necesidad de un iniciador, la partida se inicia automáticamente cuando se conecta el número de jugadores necesario.

4.1.3.4.2.1 Jugador

Serán habitualmente los alumnos. A cada jugador se le mostrará la pantalla del juego en donde responderán la pregunta cuando sea su turno, de manera que se podrán realizar partidas entre dos jugadores indistintamente del lugar en el que se encuentren.

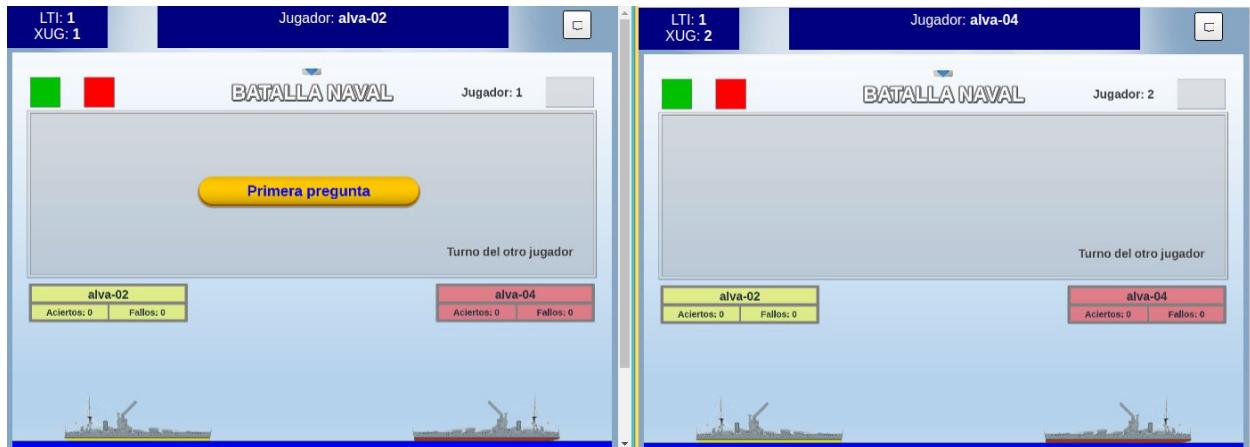


Figura 55: Pantallas Jugadores Partida Distribuida I

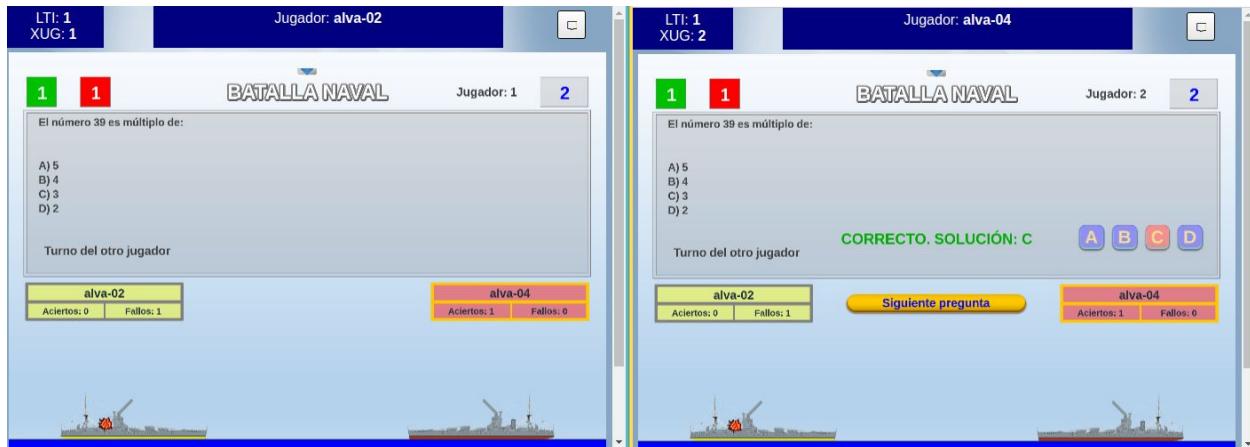


Figura 56: Pantallas Jugadores Partida Distribuida II

4.1.3.4.2.2 Espectador

En su pantalla se podrá ver el desarrollo de la partida pero no podrá intervenir en la misma.



Figura 57: Pantalla Espectador Partida Distribuida

4.1.3.5 Chat

Página en la que se muestran los mensajes recibidos de otros participantes en la partida, que puede ser tanto distribuida como multijugador, y permite a su vez el envío de mensajes.

Dentro del chat que se inicia automáticamente en una ventana emergente al iniciar la partida, o bien manualmente pulsando en el botón indicado para ello en el juego, podemos encontrar varios tipos de mensajes.

Por un lado tenemos un chat común a todos los participantes de la partida, en la que los mensajes enviados serán recibidos por todos los participantes.



Figura 58: Ejemplo chat común

Por otro lado tendremos un chat con el profesor, en la que los alumnos podrán mandar mensajes individualizados al profesor (iniciador) de la partida, esto sólo se da en el caso de partidas multijugador puesto que en las distribuidas no existe la figura del iniciador.



Figura 59: Ejemplo mensaje Alumno-Profesor



Figura 60: Ejemplo mensaje Profesor-Alumno

Por ultimo tenemos mensajes entre equipos, estos mensajes serán recibidos por los miembros del equipo que el participante que envía el mensaje ponga como destinatario.

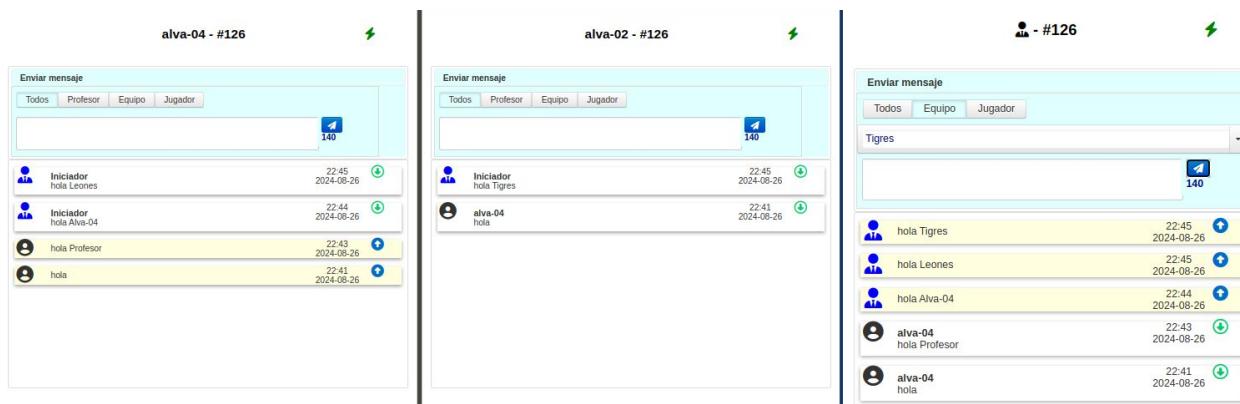


Figura 61: Ejemplo mensaje Equipos

4.1.3.6 Plantillas

4.1.3.6.1 template

Plantilla principal para las páginas del módulo de administración de la aplicación, que contiene:

- La barra superior con el menú, el botón para activar el diálogo con las preferencias y el botón para hacer *login* o desconectar de la sesión.
- Los elementos gráficos del fondo de la página.
- El pie de la página.

4.1.3.6.2 datatable

Se trata de una plantilla que muestra una tabla con los datos de una entidad recuperados de la base de datos, que permite la visualización paginada de los mismos. Esta plantilla se parametriza al incluirla en la página de forma que muestra las columnas apropiadas para la entidad correspondiente.

4.1.4 Ficheros localización – I10n

Los datos de localización se incluyen en ficheros de nombre `translations*.properties`, habiendo uno por cada idioma soportado y uno con los valores por defecto.

Permiten que los textos se muestren a los usuarios en el idioma seleccionado en su navegador.

Los idiomas soportados son español, inglés y francés.

4.1.5 Hojas de estilo (CSS)

Se usan para establecer el diseño visual de las páginas de la interfaz de usuario, tanto las del módulo de administración, como las de desarrollo de las partidas y el chat entre jugadores.

4.1.6 Javascript (JS)

Permite realizar operaciones en la interfaz del usuario en respuesta a las interacciones del usuario con la página o a la recepción de mensajes desde el servidor por la conexión WebSocket. Se detalla a continuación la funcionalidad principal de cada uno.

4.1.6.1 Desarrollo de la partida

1. `padre.js`

Para efectuar la carga de los ficheros javascript necesarios.

2. `padre-codigo.js`

Para la inicialización una vez completada la carga de la página.

3. `comunicacion.js`

Permite la comunicación entre el juego y la página que lo alberga en un elemento `iframe`.

4. `envio-variables.js`

Envía los datos recibidos del juego al TP a través de la conexión WebSocket (usando funciones de comunicación en `websocket.js`).

5. `websocket.js`

Gestiona tanto los mensajes recibidos como el envío de mensajes en la conexión WebSocket.

6. `jugador.js`

Para jugadores, con funciones para el acceso a la partida y permitir la respuesta a las preguntas recibidas.

7. `espectador.js`

Para espectadores, con funciones para realizar el seguimiento del desarrollo de la partida.

8. `jugador_o_espectador.js`

Común para espectadores y jugadores, con funciones para efectuar el acceso a la partida.

9. `chat.js`

Para abrir y cerrar el chat con el resto de participantes de la partida.

4.1.6.2 Chat

1. `chat.js`

Inicializa el WebSocket de la página de chat por el que se reciben los mensajes y muestra al usuario los mensajes recibidos.

4.1.7 Mensajes WebSocket y LTI

La coordinación entre los distintos participantes en una partida se realiza con el intercambio de mensajes a través de la conexión WebSocket.

Esta coordinación es necesaria cuando en la partida participa más de un jugador, es decir, en las partidas multijugador, tanto si son distribuidas como si no. De todas formas, vamos a ver que los mensajes intercambiados varían en función de esta circunstancia.

Igualmente se utiliza una conexión WebSocket para el envío de los mensajes intercambiados en el chat entre los participantes en una partida, para que se muestren inmediatamente a los destinatarios de los mismos.

Detallamos a continuación los mensajes intercambiados en cada caso.

4.1.7.1 Partida Centralizada Multijugador

4.1.7.1.1 Registro de partida

Cuando el iniciador se conecta a la página desde la que controla el desarrollo de la partida, se envían estos mensajes:

1. El iniciador envía el mensaje de registro identificando la partida.
2. La aplicación responde indicando que se ha producido la conexión del iniciador.

En la siguiente figura podemos ver el diagrama de secuencia con los mensajes intercambiados.

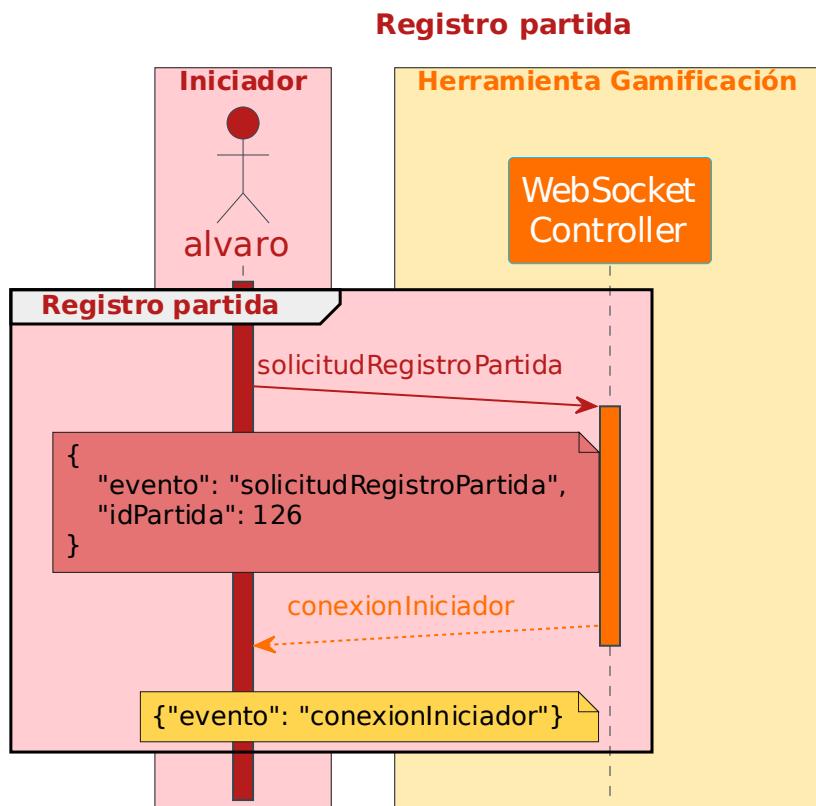


Figura 62: Registro partida

4.1.7.1.2 Conexión de jugadores y espectadores

Cuando se conectan tanto jugadores como espectadores se envían los siguientes mensajes (los dos primeros únicamente en el caso de jugadores que son los que tienen que seleccionar el grupo y equipo):

1. El jugador envía un mensaje para solicitar la lista de grupos de una partida.
2. La aplicación envía la lista de grupos y equipos correspondiente a esa partida.
3. El jugador/espectador solicita el acceso a la partida.
4. La aplicación responde con los datos de registro en la partida.
5. La aplicación envía un mensaje a todos los participantes en la partida para que actualicen la lista de jugadores y espectadores.

En las siguientes figuras se muestran los diagramas de secuencia con el intercambio de mensajes a través la conexión WebSocket entre la aplicación y los jugadores y espectadores conforme éstos se van conectando a la partida. Los diagramas se muestran en el mismo orden en que sucedieron en la prueba real de la partida.

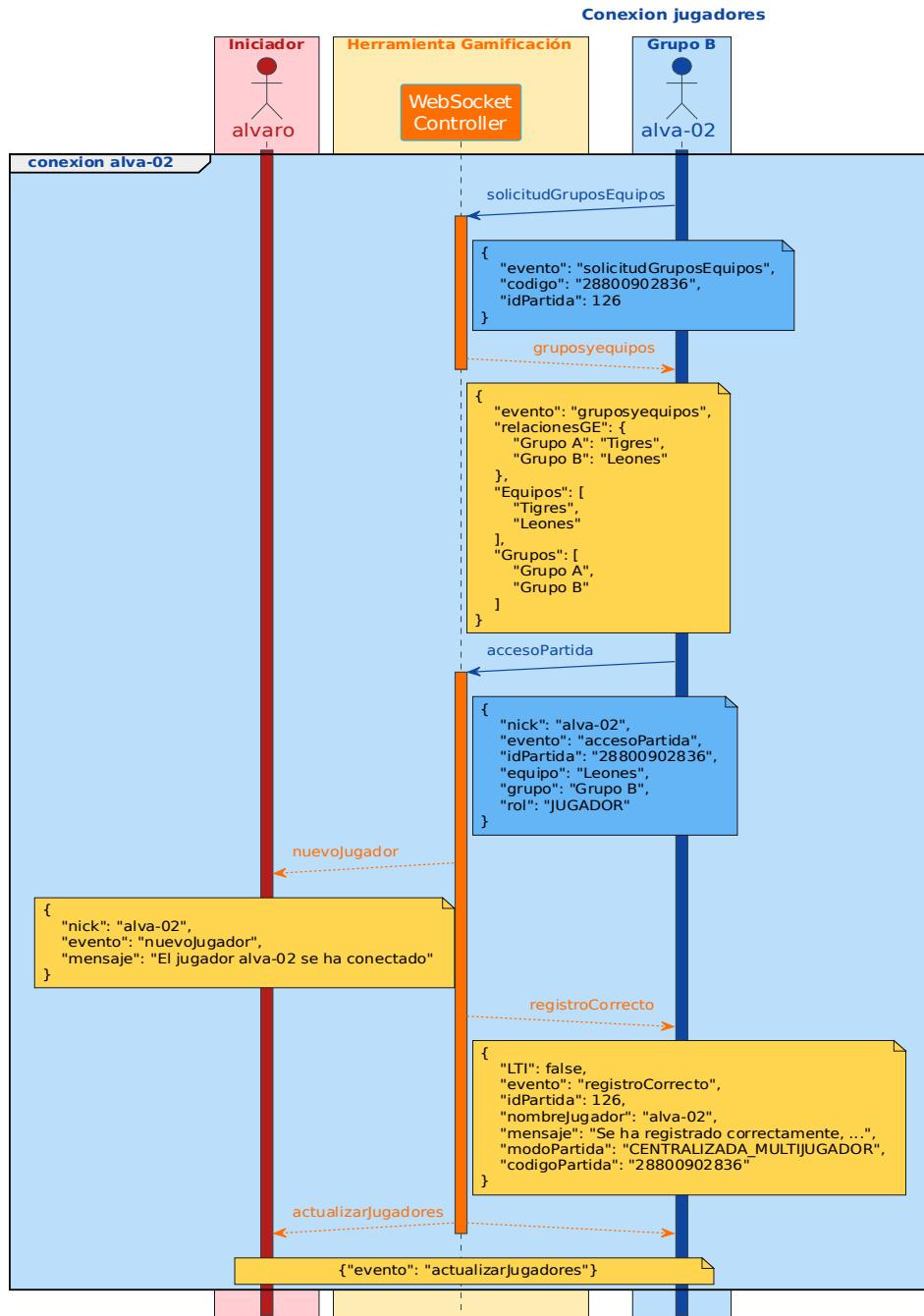


Figura 63: Conexión jugador alva-02

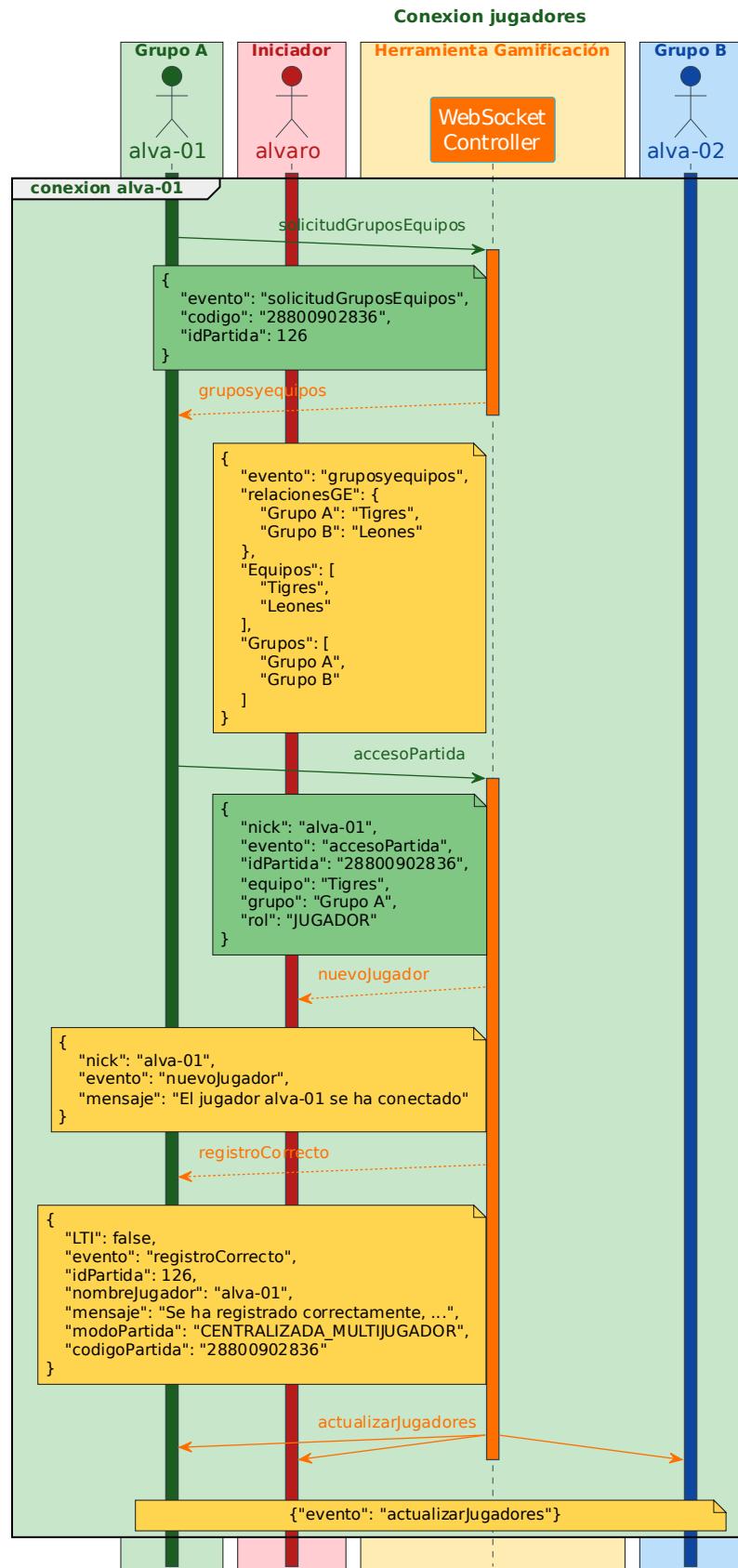


Figura 64: Conexión jugador alva-01

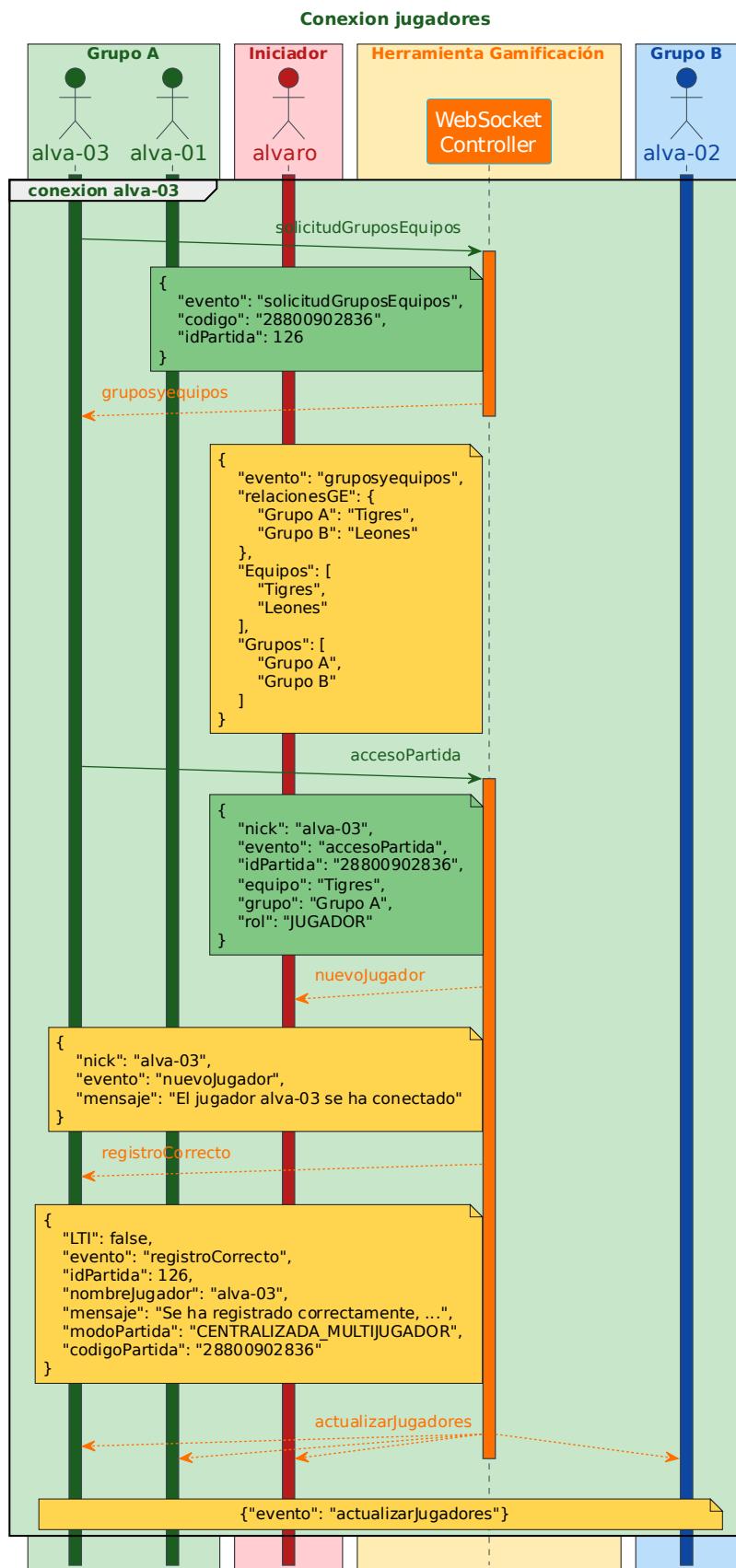


Figura 65: Conexión jugador alva-03

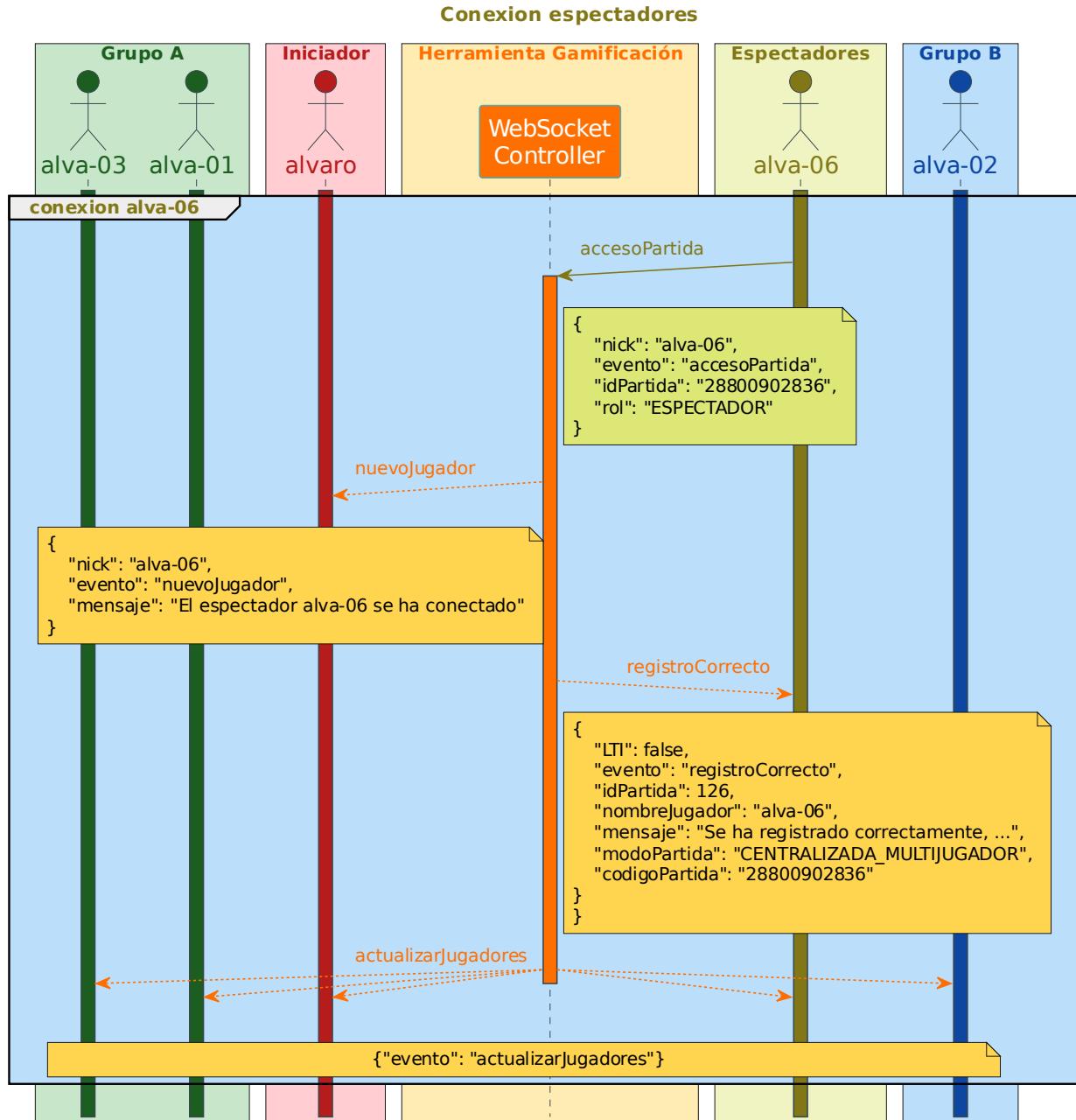


Figura 66: Conexión espectador alva-06

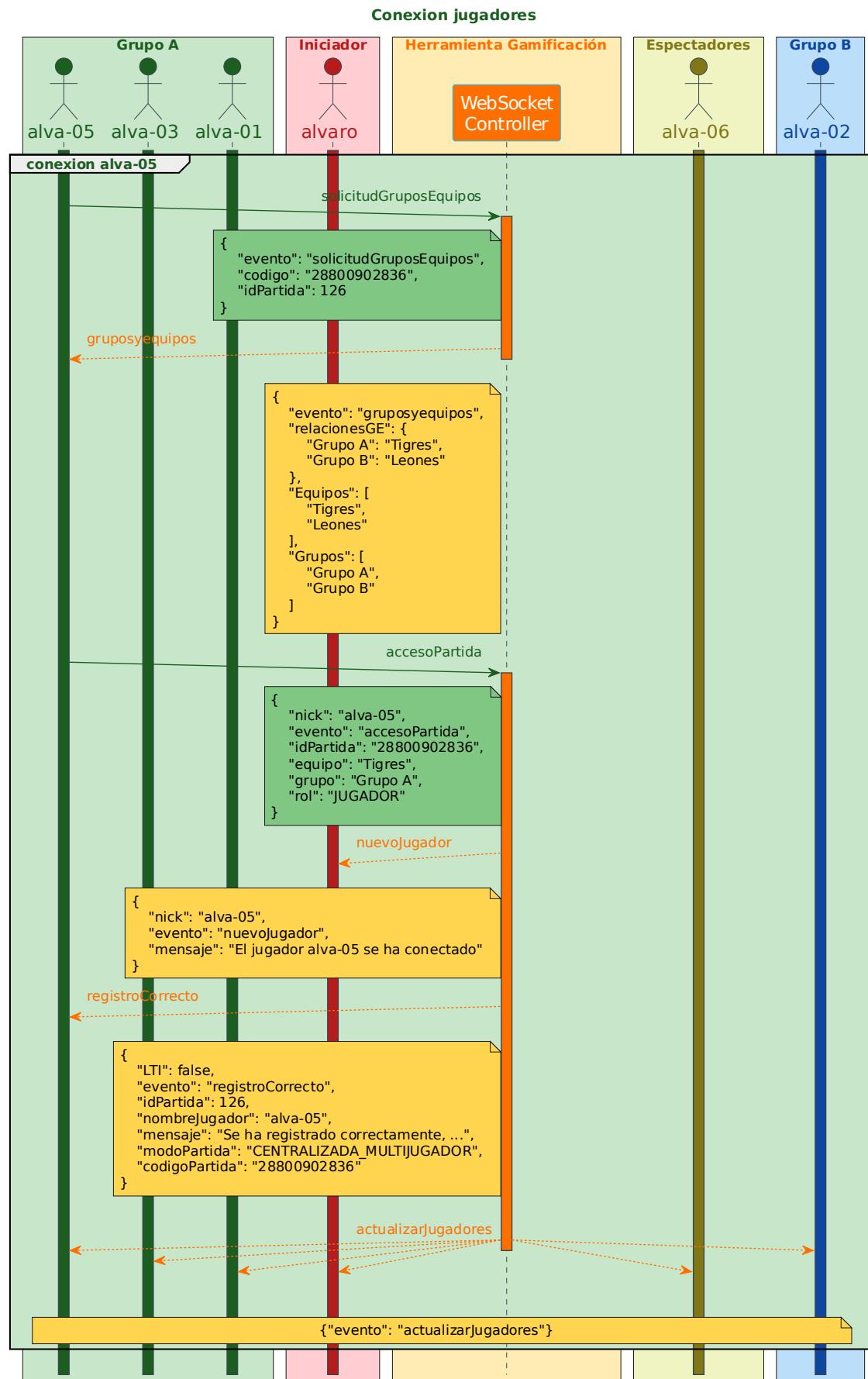


Figura 67: Conexión jugador alva-05

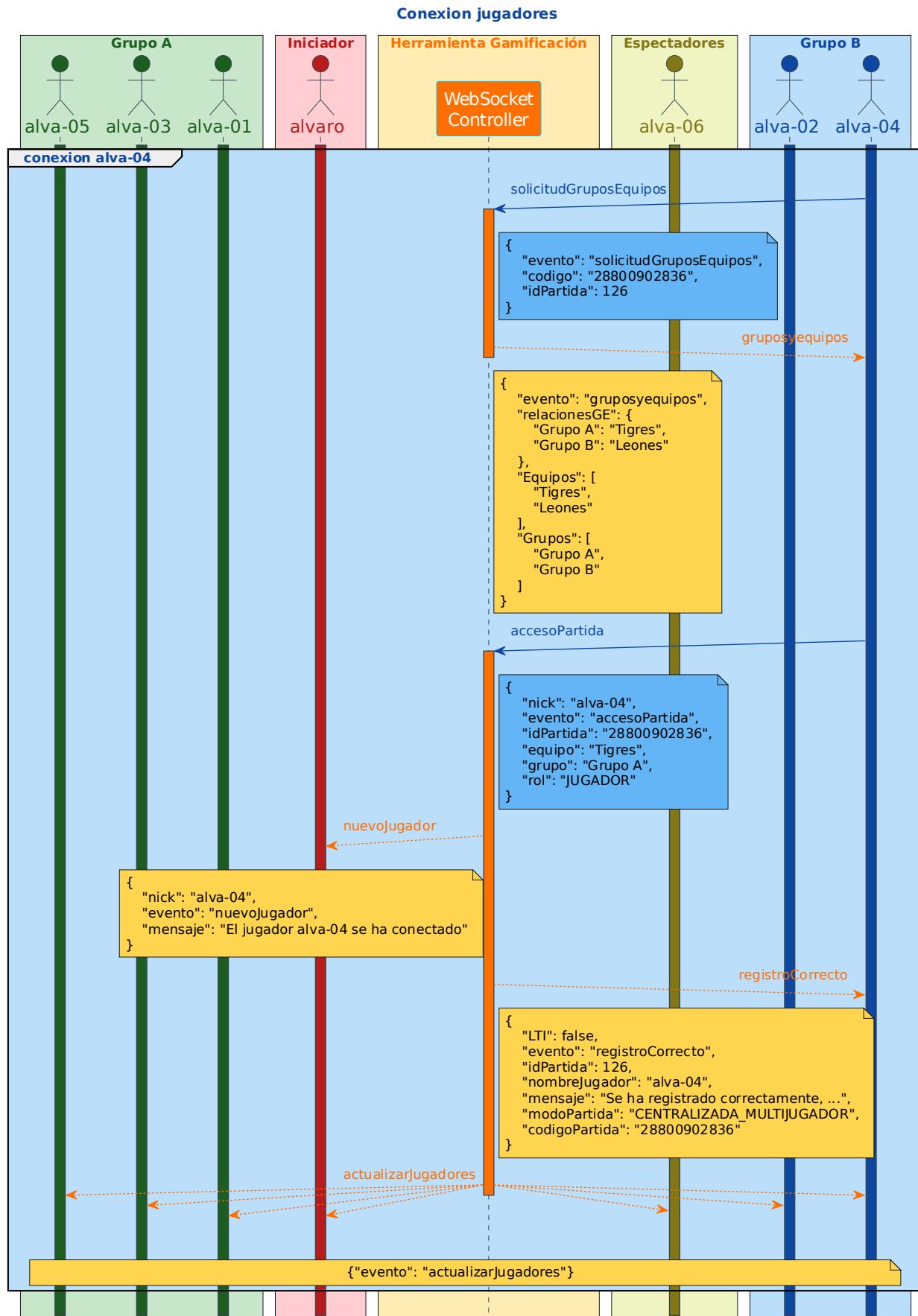


Figura 68: Conexión jugador alva-04

4.1.7.1.3 Cambio de grupo/equipo

El iniciador de la partida puede reasignar a un jugador a un grupo o equipo distinto del actual, para lo cual se intercambian los siguientes mensajes:

1. Envío de comando JSF para realizar el cambio.
2. El controlador JSF comunica el cambio al controlador del WebSocket.
3. Se envía un mensaje al jugador con el cambio de grupo.
4. Se envía un mensaje a todos los participantes en la partida para que actualicen las listas de jugadores y espectadores.

A continuación se muestra el diagrama de secuencia con los mensajes intercambiados para el cambio de grupo y equipo del jugador alva-04.

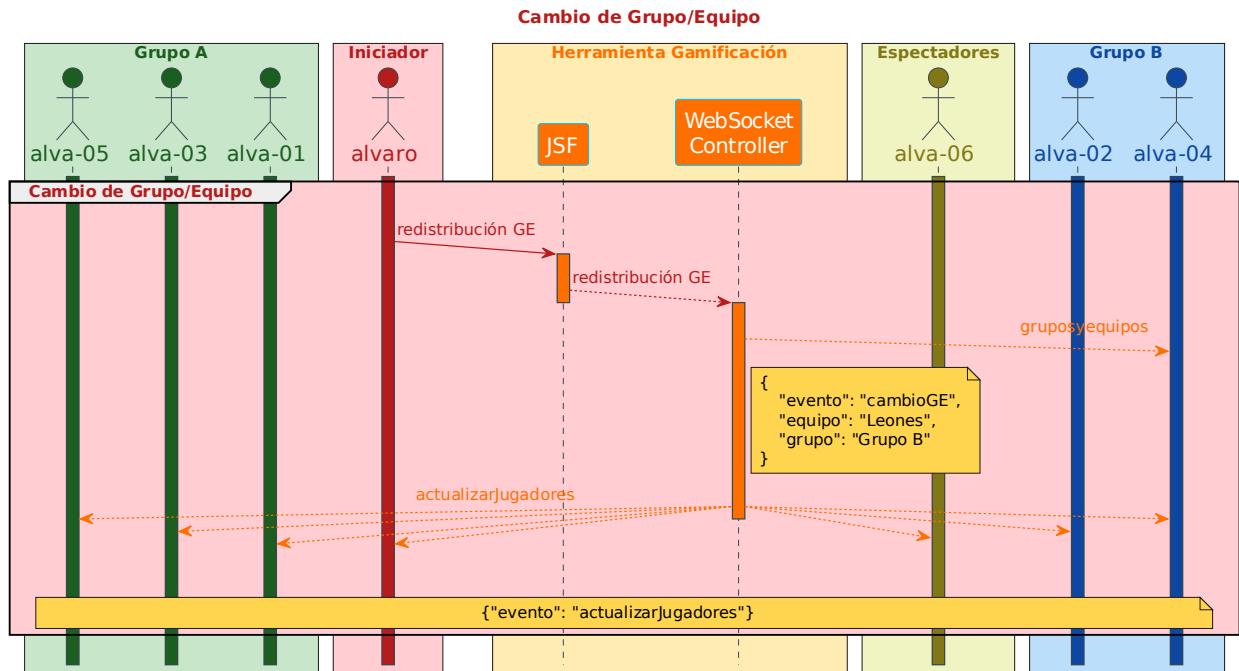


Figura 69: Cambio grupo y equipo

4.1.7.1.4 Inicio de la partida

Cuando el iniciador de la partida da comienzo a la partida se intercambian los siguientes mensajes:

1. Envío de comando JSF para iniciar la partida.
2. El controlador JSF lo comunica al controlador del WebSocket.
3. Se envía un mensaje de inicio a todos los participantes, que incluye la ruta para acceder al juego.
4. El iniciador carga el juego en el *iframe* de su página y envía un mensaje cuando la carga está completa.
5. La aplicación envía al iniciador los datos iniciales para el funcionamiento del juego.

En la siguiente figura podemos ver el diagrama de secuencia con los mensajes intercambiados.

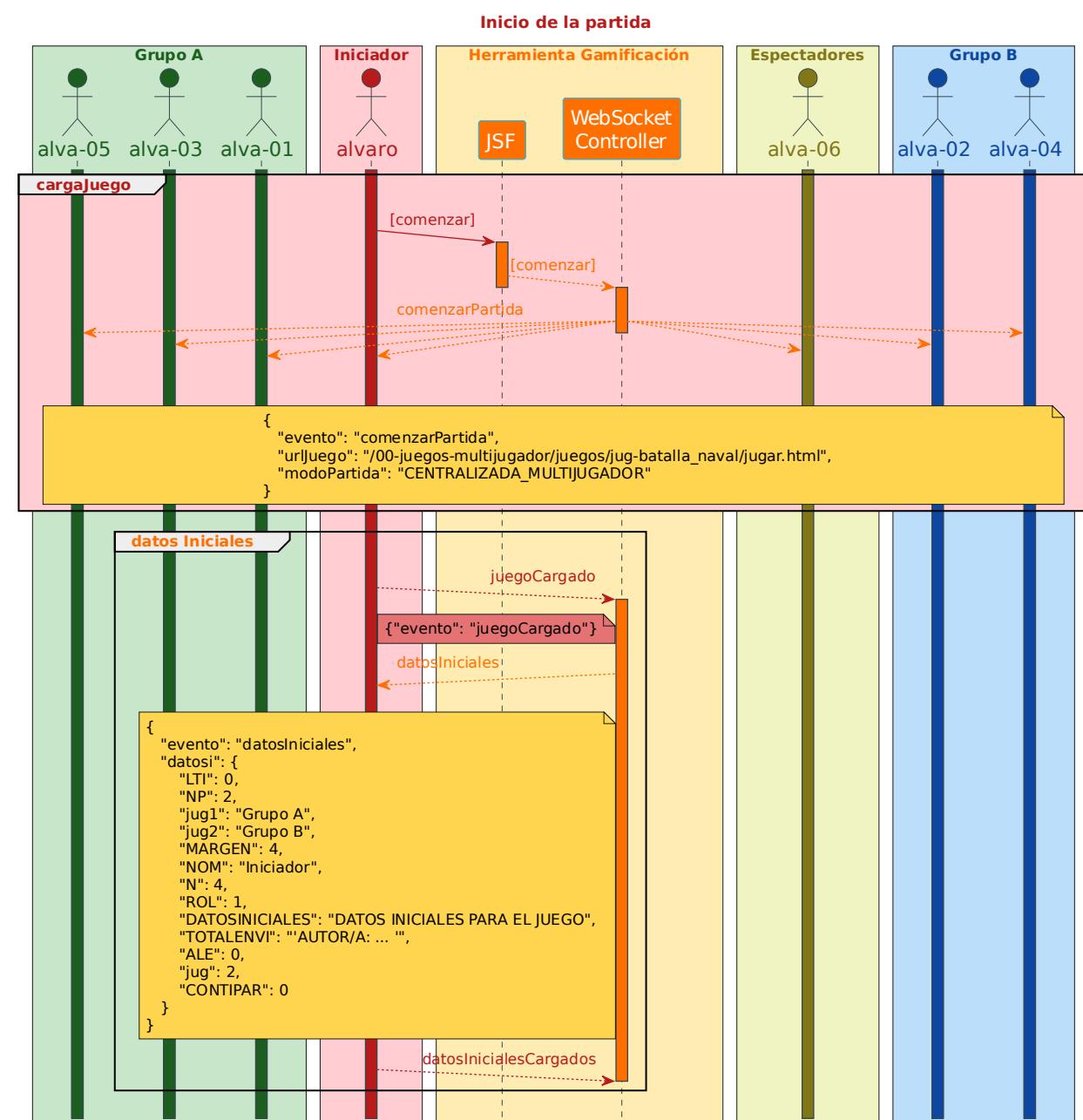


Figura 70: Inicio de la partida

4.1.7.1.5 Pregunta

El proceso de realizar una pregunta es el siguiente:

1. El iniciador solicita pasar a una nueva pregunta.
2. Se reenvía la pregunta a los jugadores del grupo correspondiente, así como a los espectadores.
3. Se arranca un temporizador que envía un mensaje cada segundo con el tiempo que resta para contestar la pregunta.
4. Los jugadores envían sus respuestas.
5. Se notifica al iniciador y a los espectadores que se ha recibido una respuesta.
6. Cuando se acaba el tiempo o responden todos los jugadores, se envía la respuesta del grupo.
7. Se envía un mensaje de notificación de tiempo agotado a los participantes.
8. El iniciador envía un mensaje con el estado de la partida actualizado.

Vemos a continuación los diagramas de secuencia con los mensajes intercambiados para el desarrollo del juego, en primer lugar el envío de la pregunta a los jugadores del grupo A y al espectador.

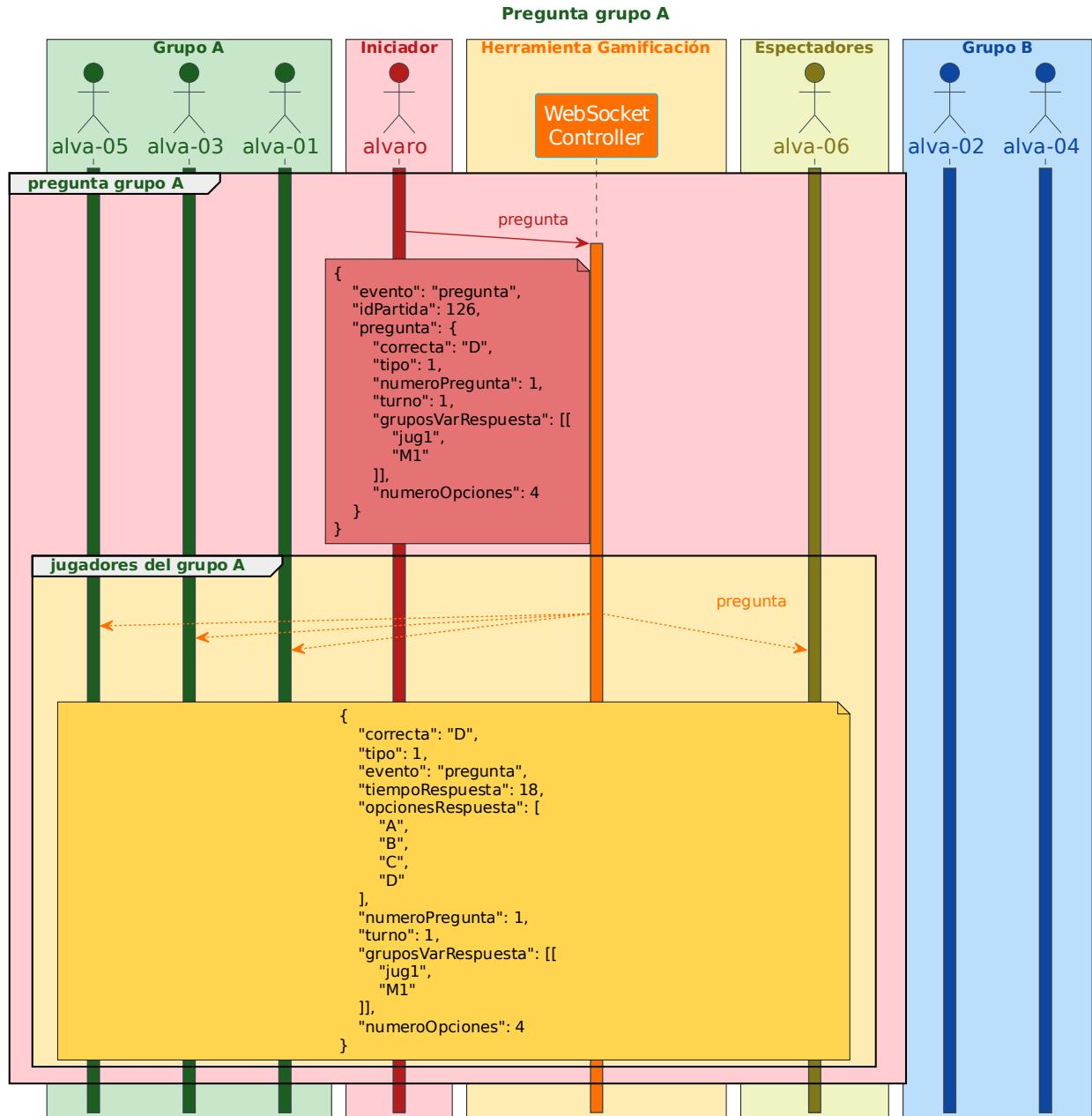


Figura 71: Pregunta grupo A

A continuación tenemos los diagramas de secuencia con los mensajes intercambiados para la recepción de las respuestas de los jugadores del grupo A y el control del tiempo de respuesta.

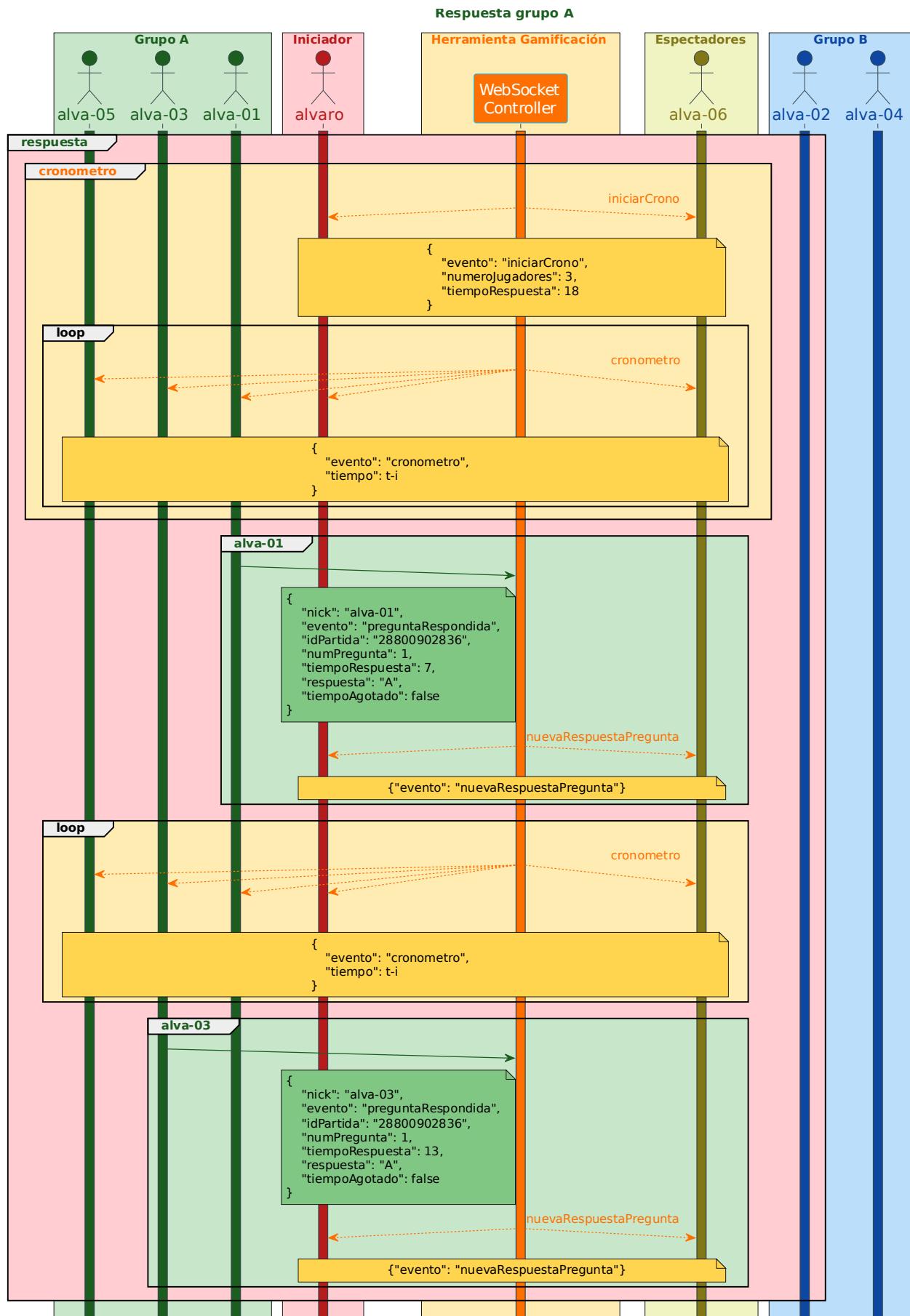


Figura 72: Respuesta grupo A (I)

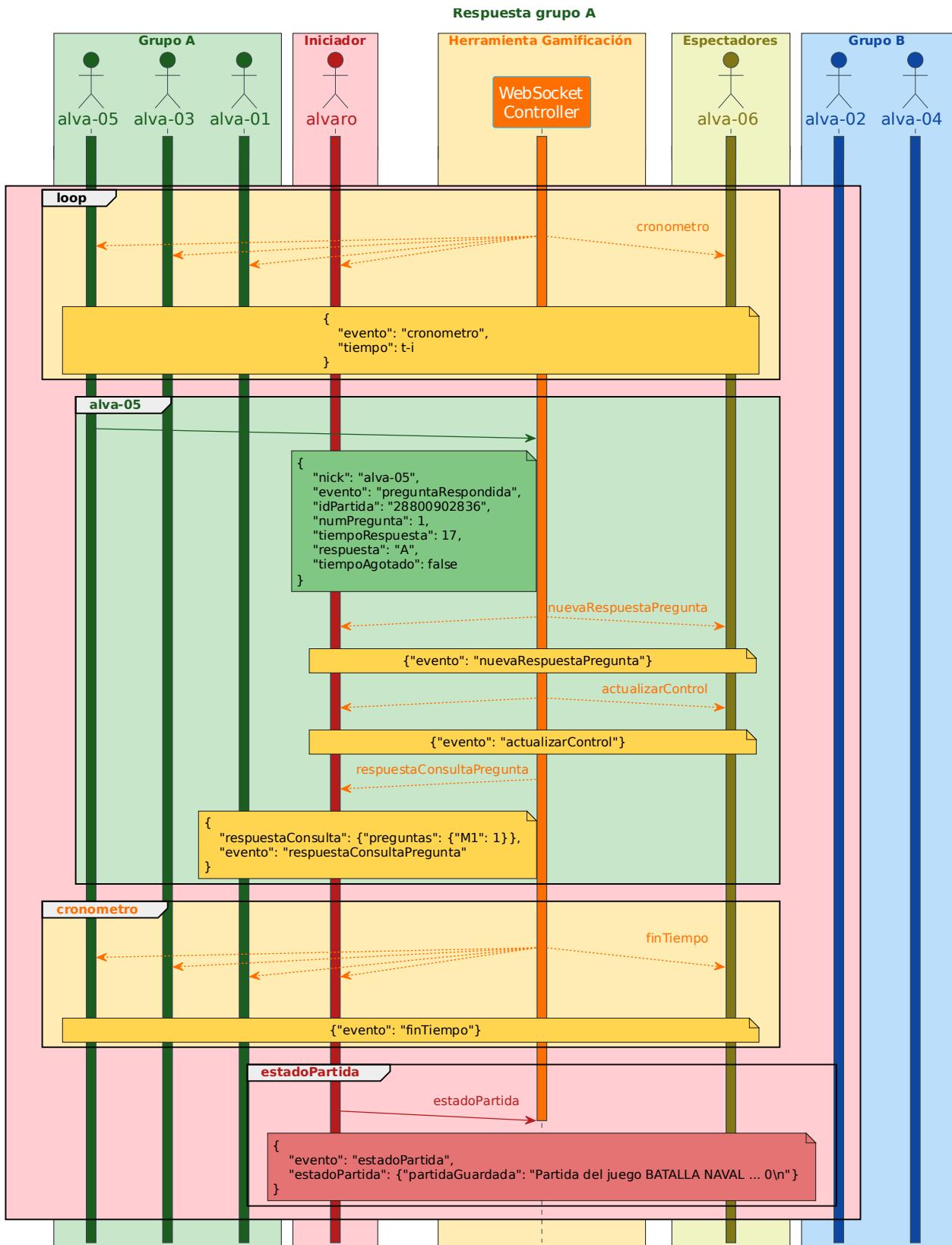


Figura 73: Respuesta grupo A (II)

A continuación se muestra el diagrama de secuencia con la pregunta al grupo B.

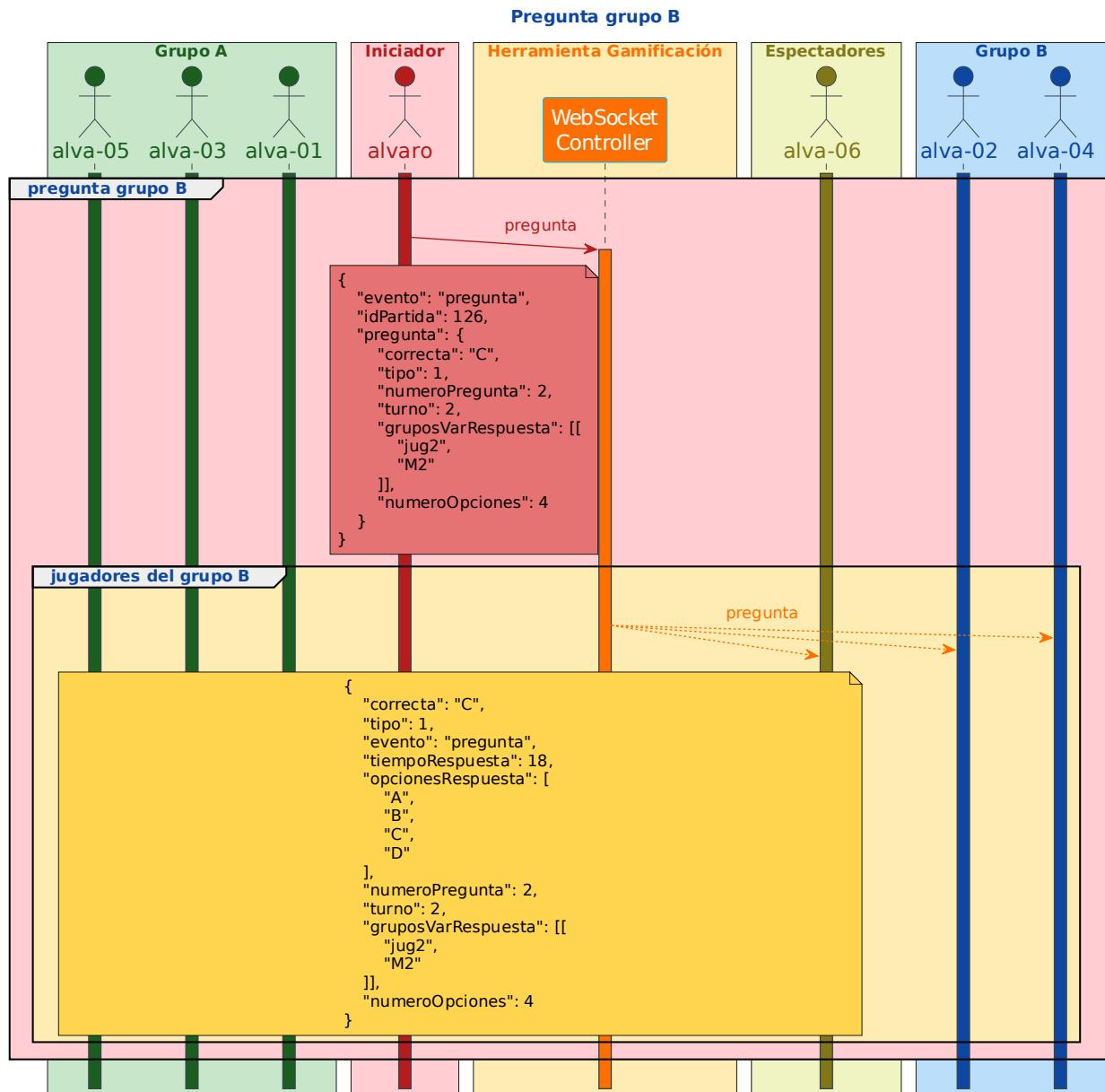


Figura 74: Pregunta grupo B

Y a continuación el diagrama de secuencia con los mensajes intercambiados para las respuesta de los jugadores del grupo B.

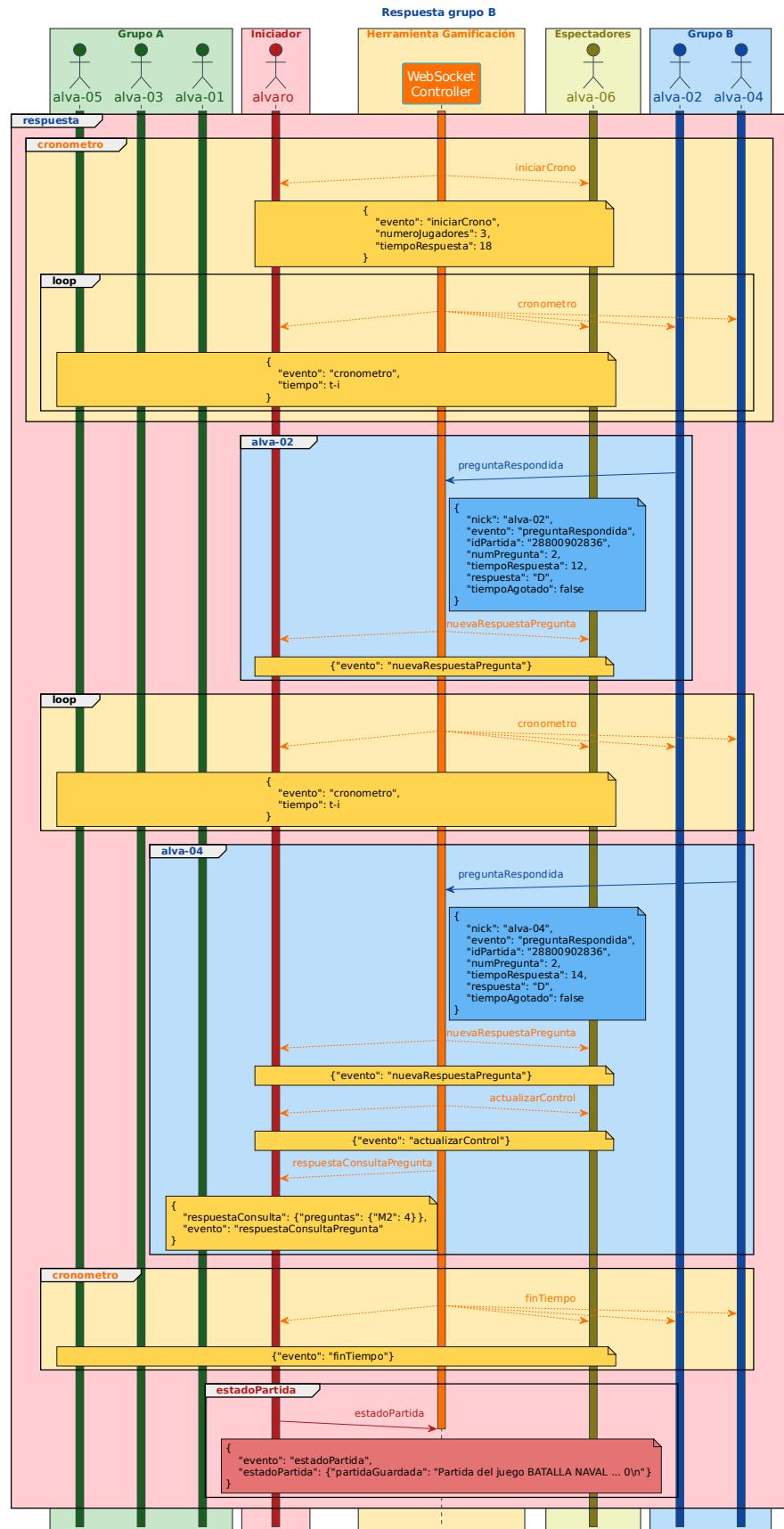


Figura 75: Respuesta grupo B

4.1.7.1.6 Final de la partida

Cuando finaliza la partida, desde el juego se envía un mensaje con el resultado obtenido por cada uno de los grupos.

Por último, se muestra el diagrama de secuencia con los mensajes intercambiados cuando acaba la partida.

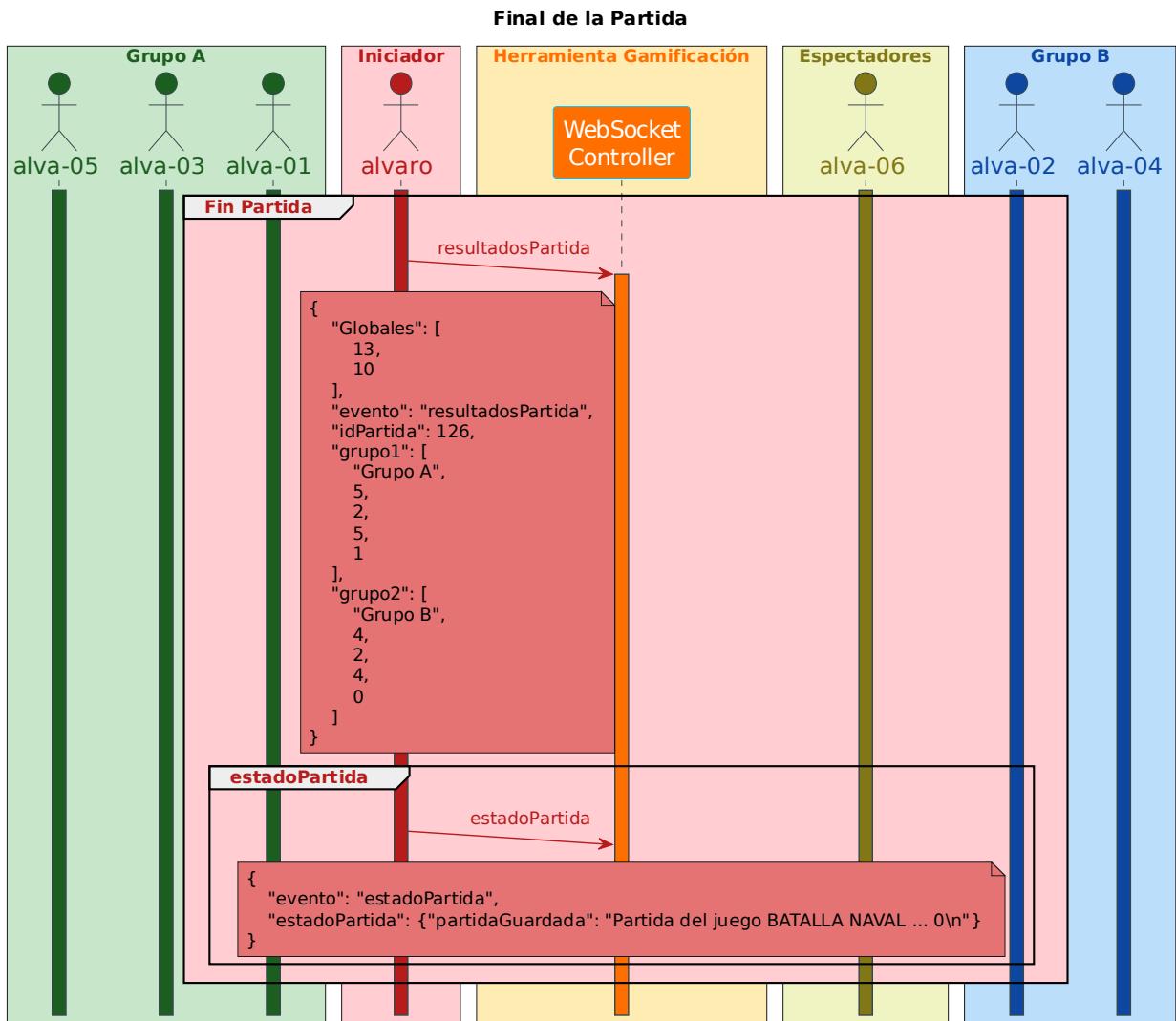


Figura 76: Final de la partida

4.1.7.2 Partida Distribuida Multijugador

4.1.7.2.1 Registro de jugadores y espectadores

Cuando se conectan tanto jugadores como espectadores se envían los siguientes mensajes:

1. El jugador/espectador solicita el acceso a la partida.
2. La aplicación responde con los datos de registro en la partida.
3. La aplicación envía un mensaje a todos los participantes en la partida para que actualicen la lista de jugadores y espectadores.

Se muestran a continuación los diagramas de secuencia con los mensajes intercambiados en la conexión de los jugadores y espectadores.

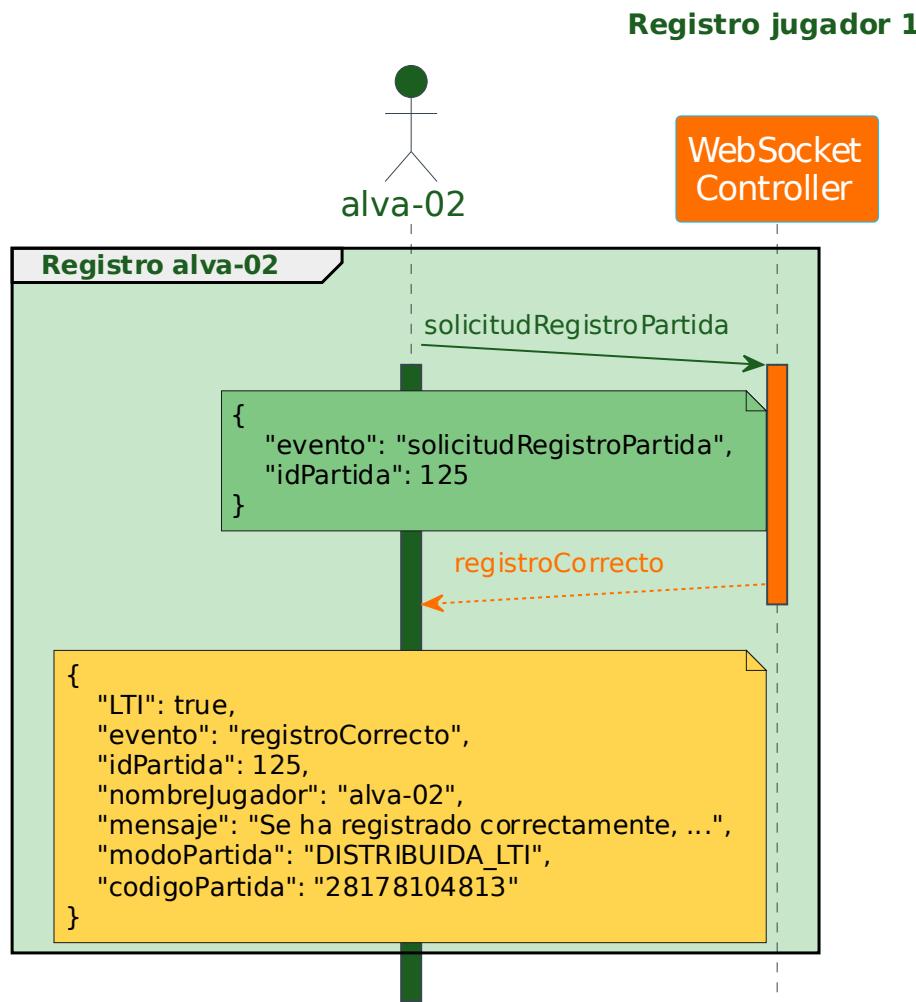


Figura 77: Registro y conexión jugador alva-02

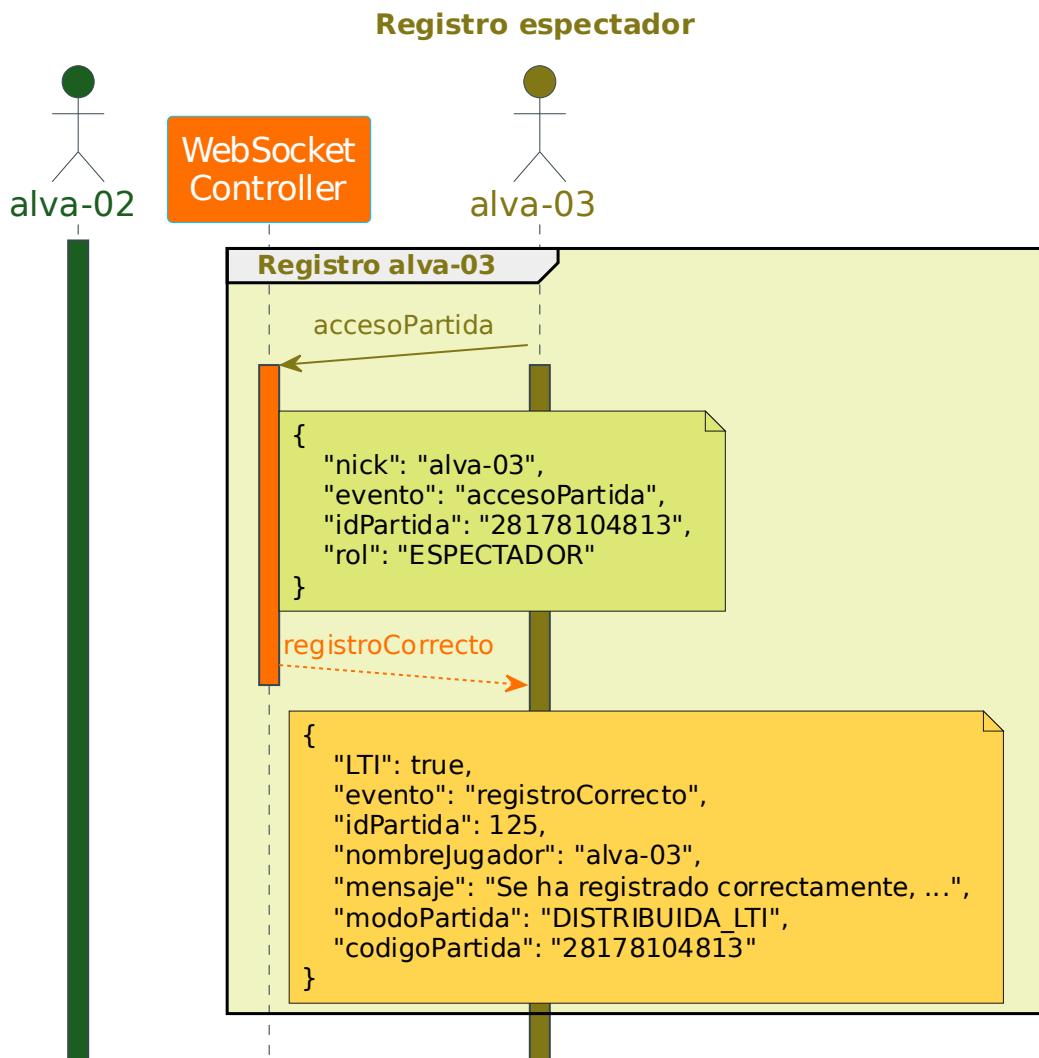


Figura 78: Conexión espectador alva-03

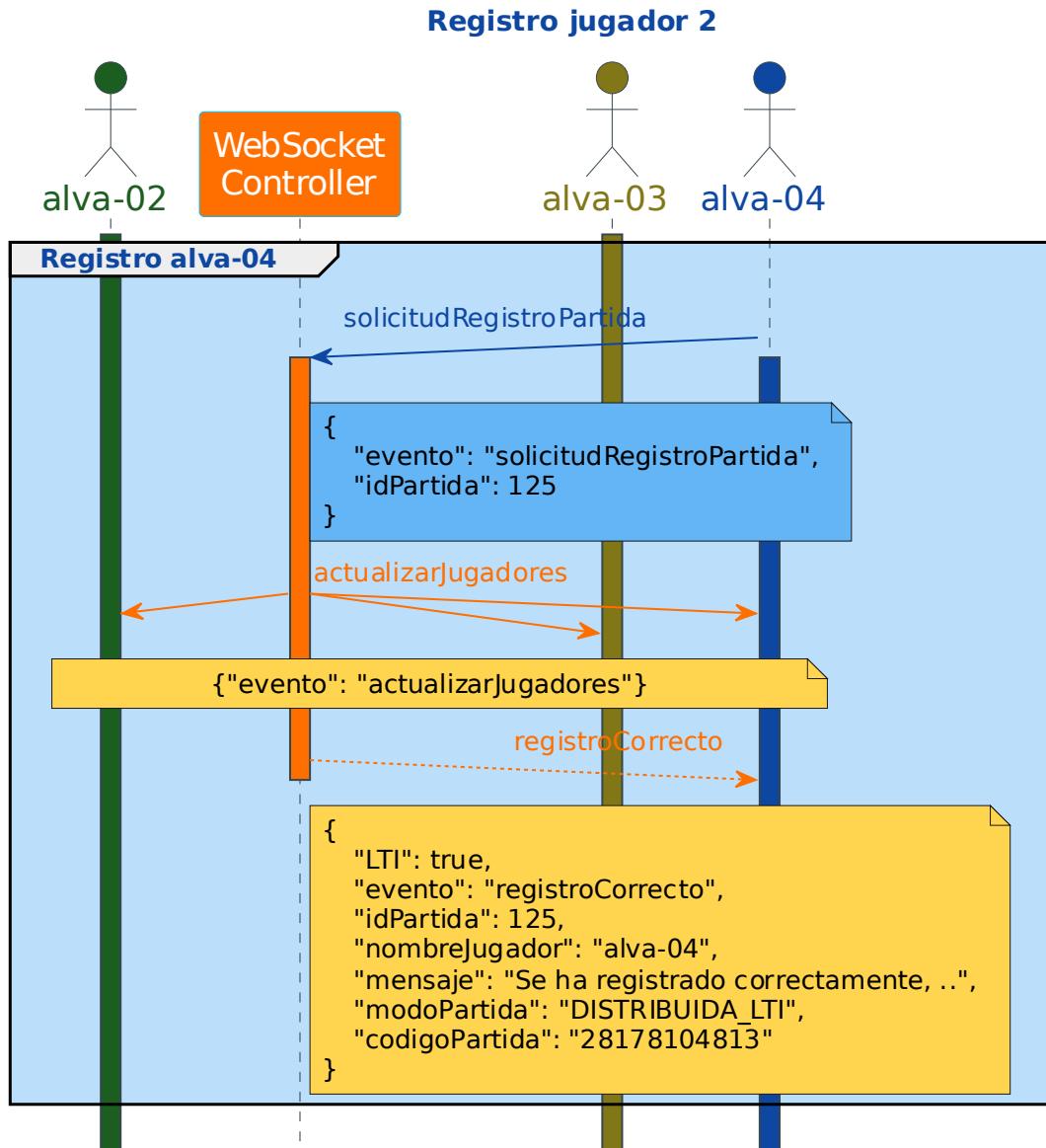


Figura 79: Registro y conexión jugador alva-04

4.1.7.2.2 Inicio de la partida

El inicio de la partida se produce automáticamente cuando se terminan de conectar los jugadores.

1. Se envía un mensaje de inicio a todos los participantes, que incluye la ruta para acceder al juego.
2. Los participantes cargan el juego en el *iframe* de su página.
3. Cada participante envía un mensaje comunicando que la carga del juego está completa.
4. La aplicación responde al mensaje anterior enviando los datos iniciales de la partida personalizados para ese participante.

A continuación, se muestran los diagramas de secuencia con los mensajes intercambiados para notificar a los participantes el comienzo de la partida, y la carga e inicialización consiguientes del juego por parte de jugadores y espectador.

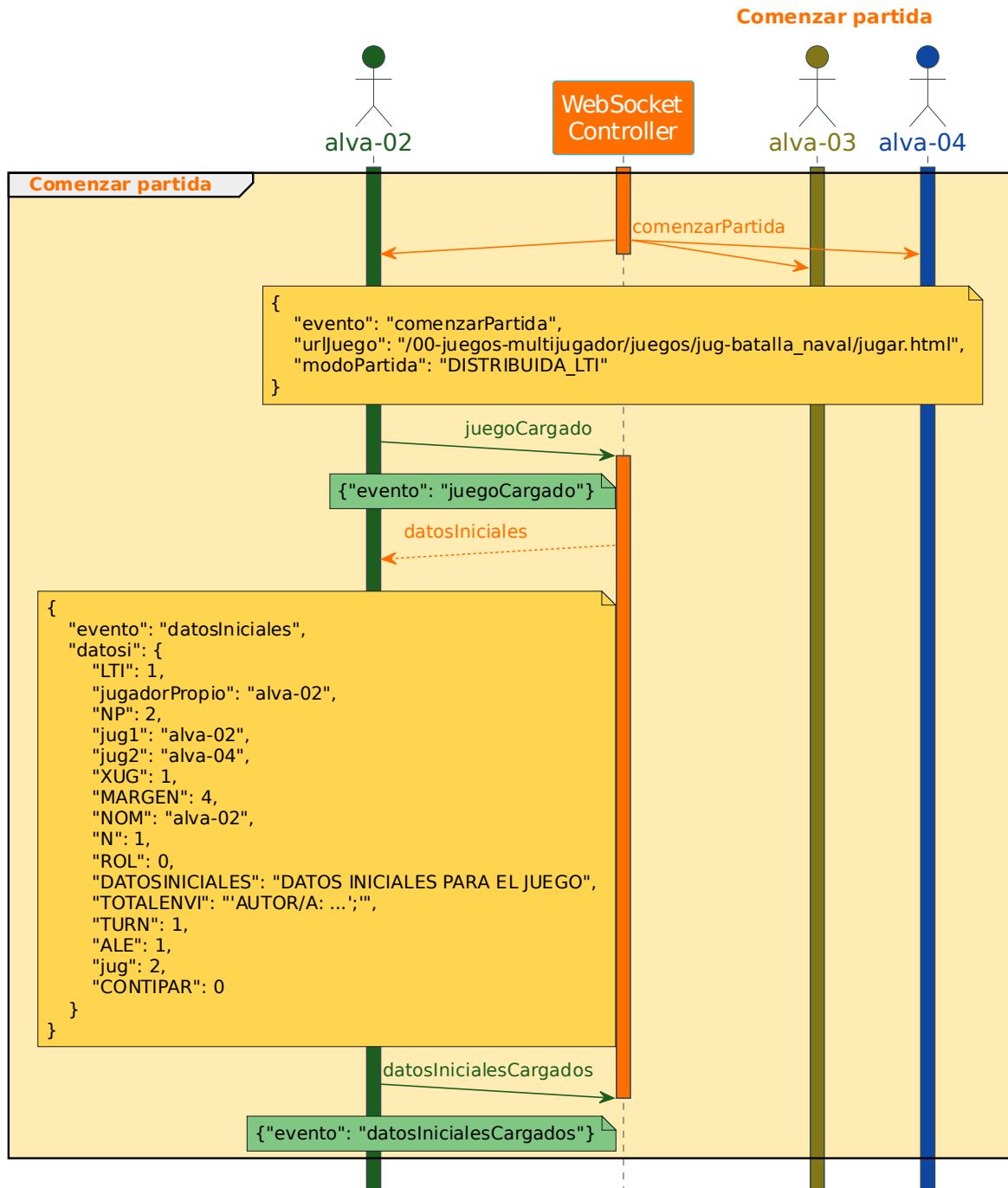


Figura 80: Comenzar partida - jugador alva-02

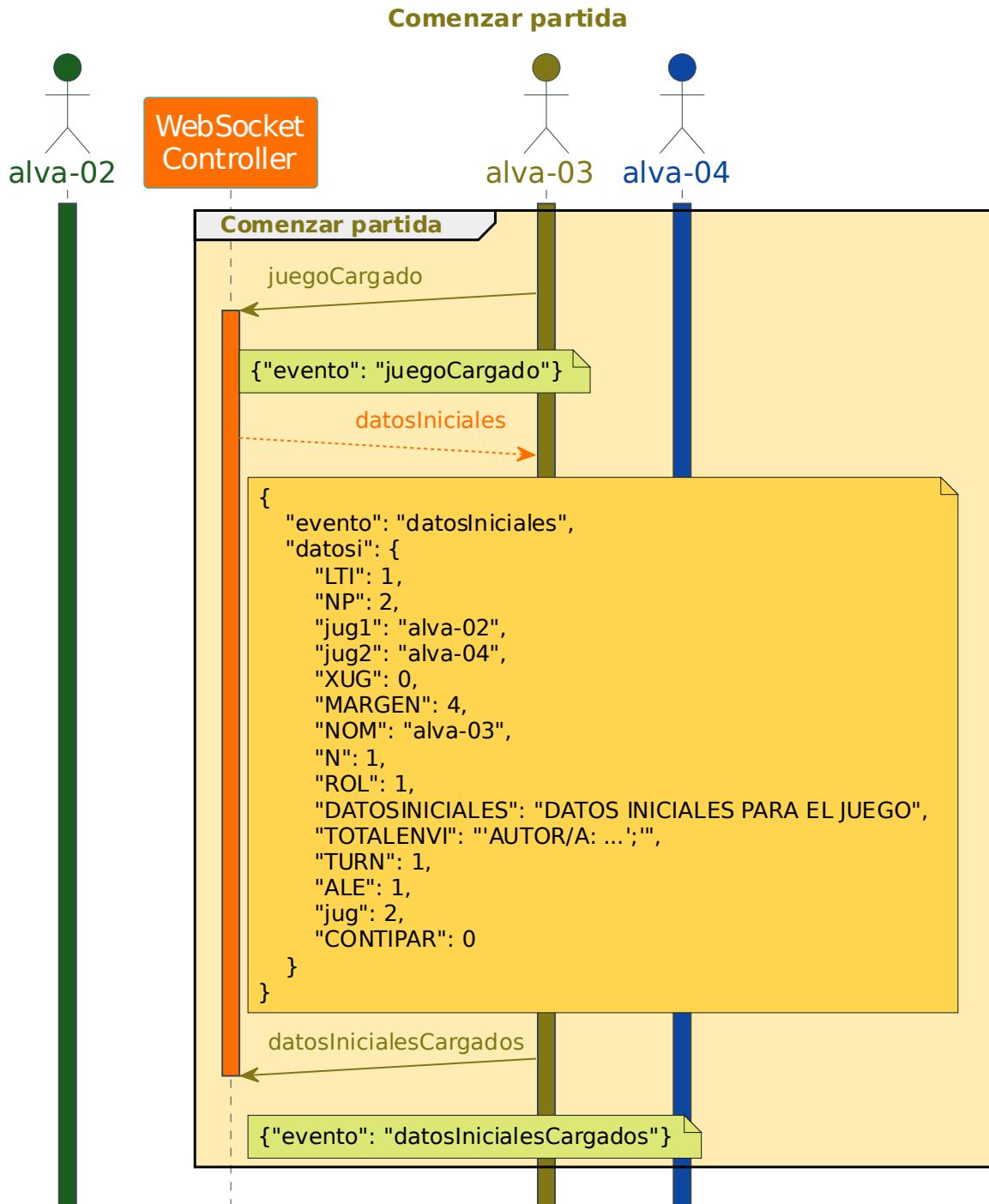


Figura 81: Comenzar partida - espectador alva-03

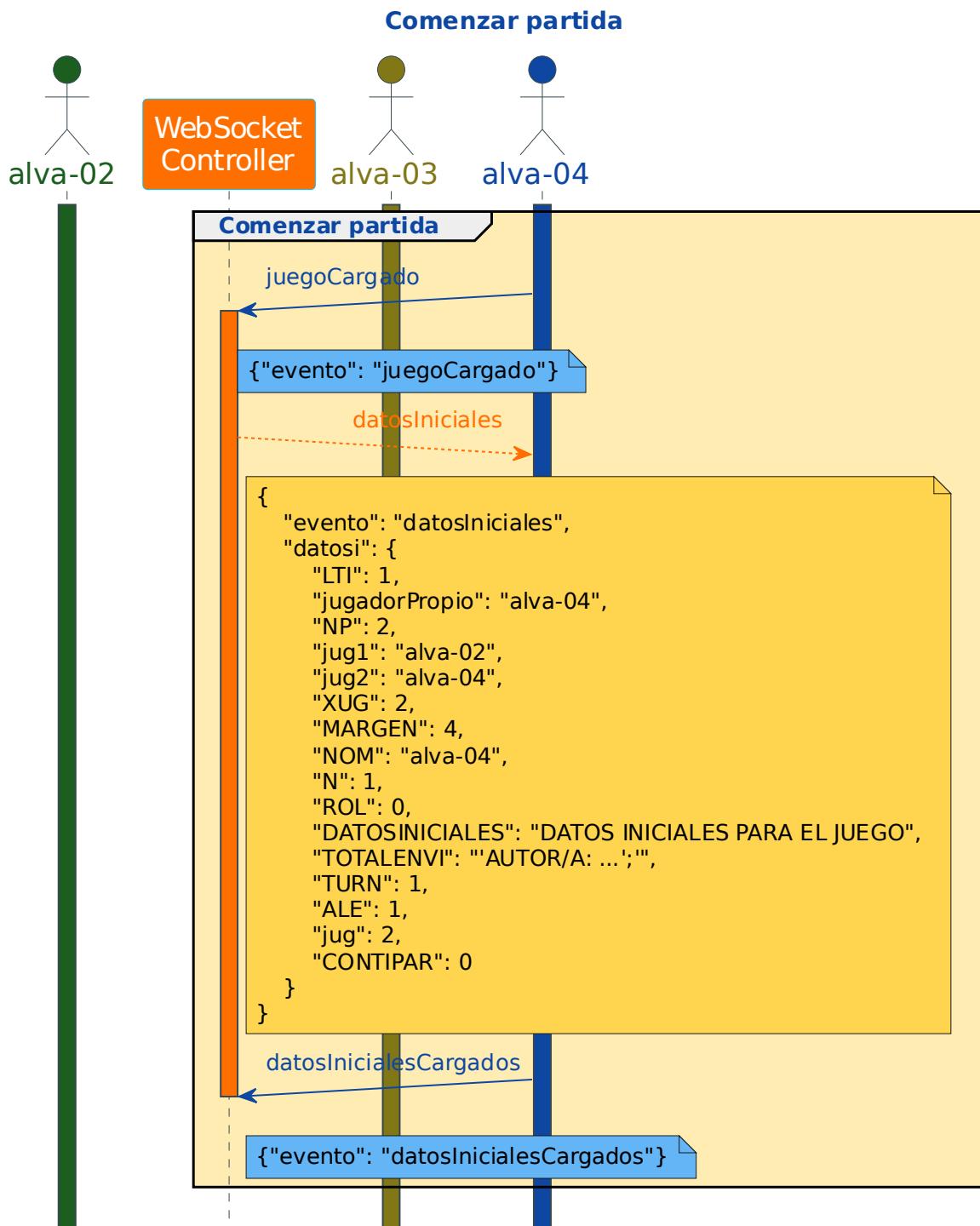


Figura 82: Comenzar partida - jugador alva-04

4.1.7.2.3 Reconexión

En el caso de reconexión de un jugador, se le envían inmediatamente los mensajes de inicio de la partida y en los datos iniciales se incluye la información del estado actual de la partida, tal como se muestra en el siguiente diagrama de secuencia.

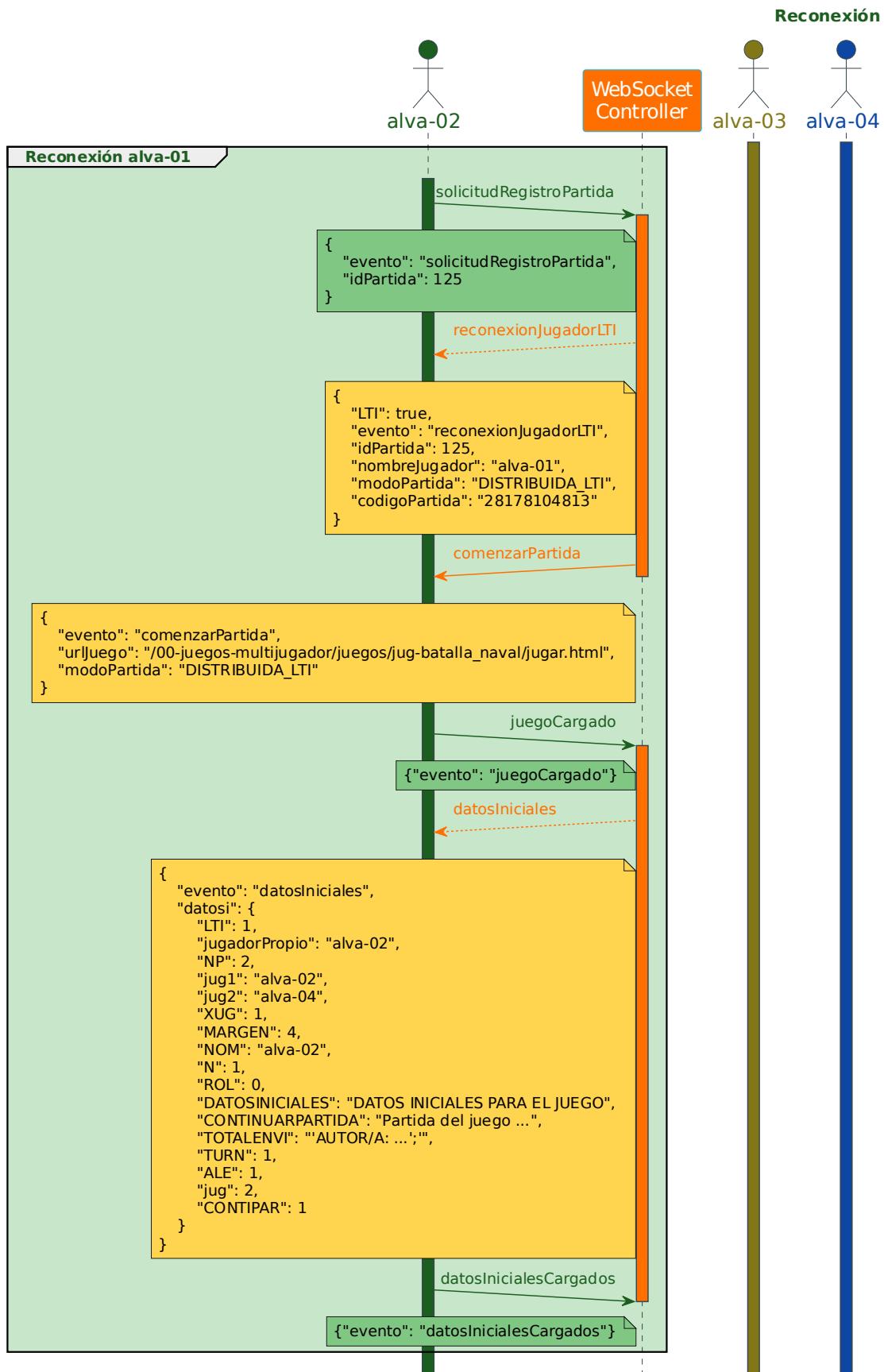


Figura 83: Reconexión

4.1.7.2.4 Preguntas

En el caso de las partidas distribuidas, el desarrollo de la partida se controla desde el juego que se ejecuta para cada uno de los jugadores, y la función de la aplicación es transmitir las actualizaciones del estado de la partida recibidas al resto de participantes.

Se muestra en primer lugar el diagrama de secuencia con los mensajes correspondientes a la primera pregunta.

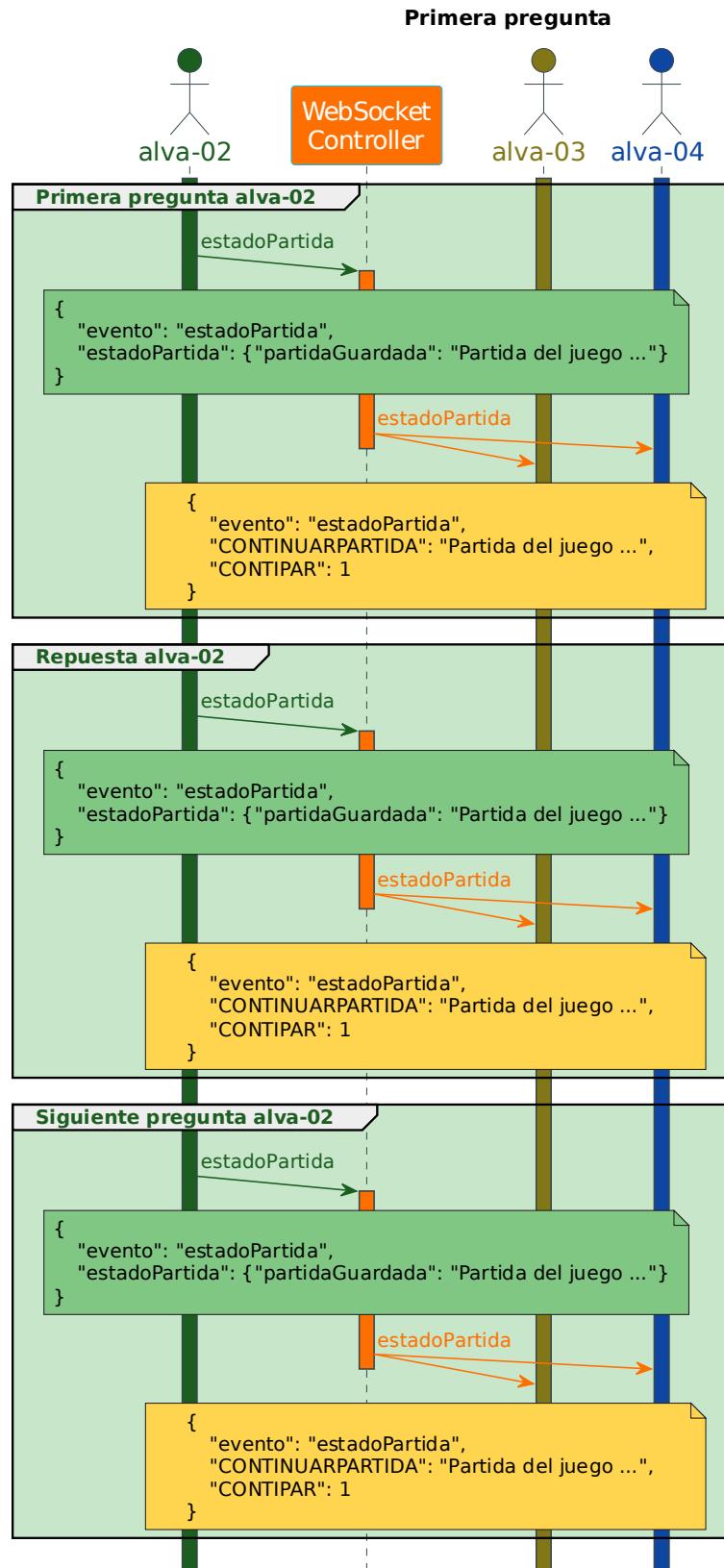


Figura 84: Primera pregunta

Y a continuación los diagramas para las siguientes preguntas, alternándose ambos jugadores.

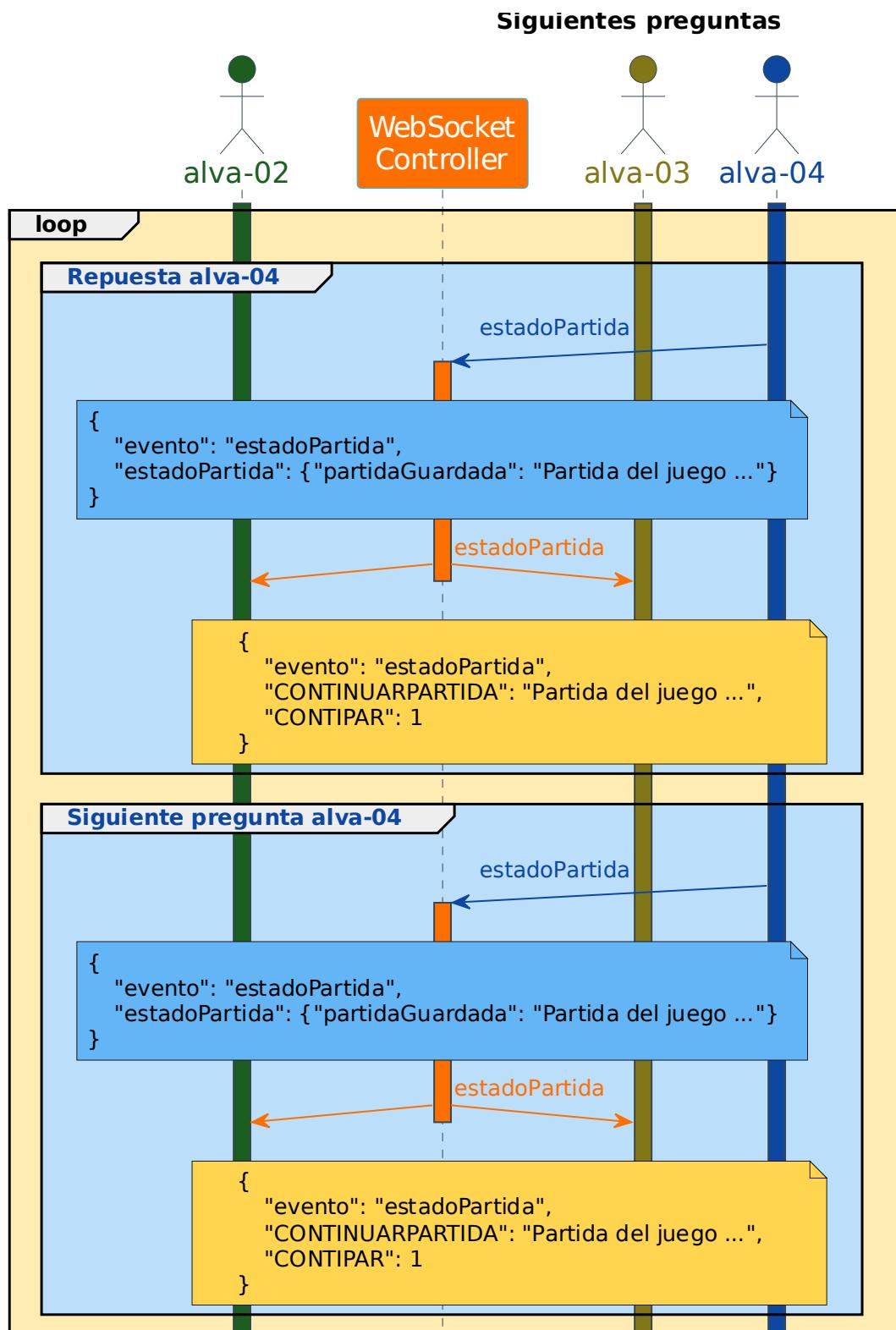


Figura 85: Siguientes preguntas - jugador alva-04

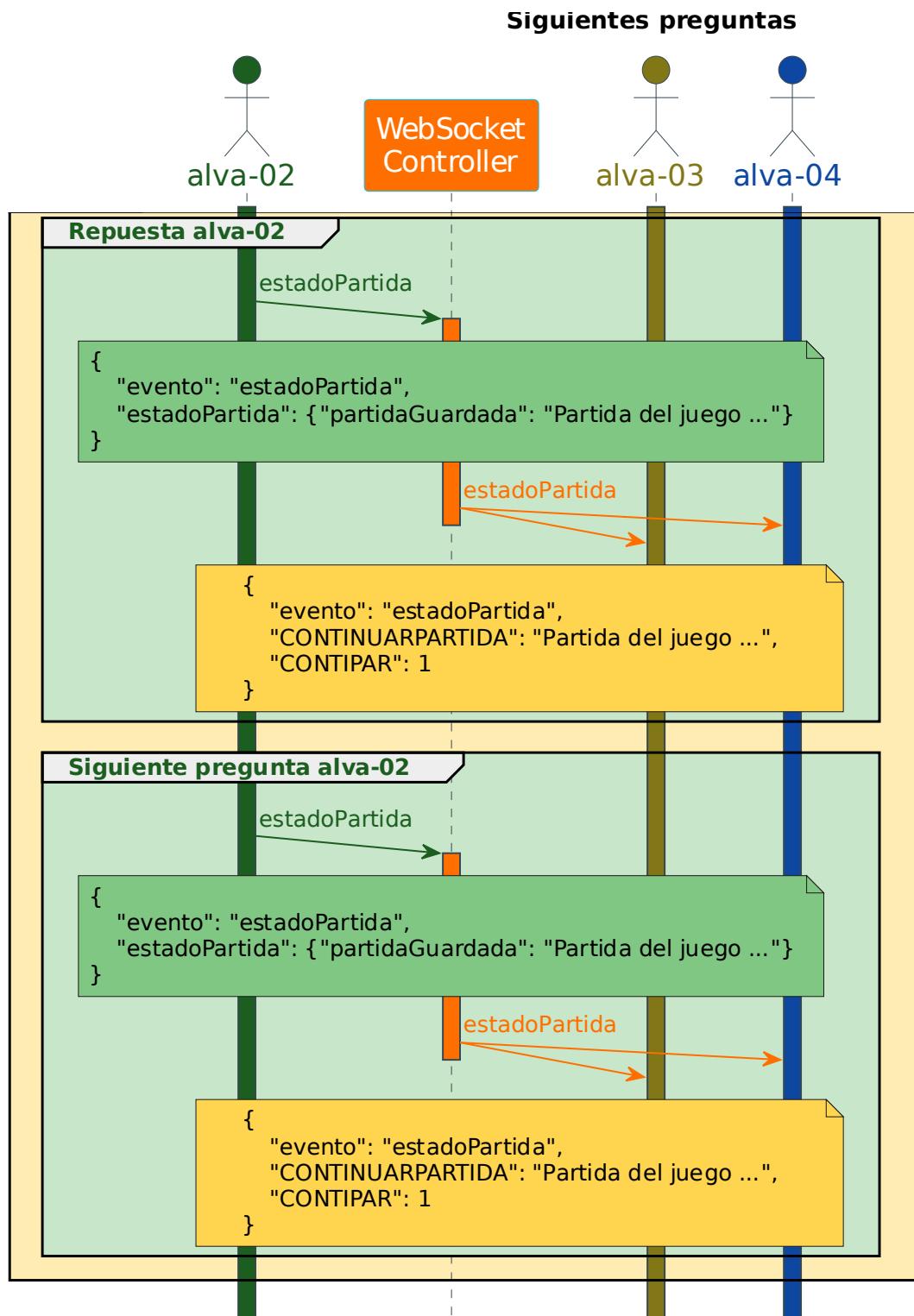


Figura 86: Siguientes preguntas - jugador alva-02

4.1.7.2.5 Final de la partida

Cuando finaliza la partida:

1. Se envía un mensaje desde el juego con el resultado obtenido por cada uno de los grupos y a continuación otro mensaje actualizando el estado de la partida.
2. La plataforma reenvía el mensaje de actualización del estado de la partida al resto de participantes.

3. Al recibir esta actualización de estado, el resto de participantes envían un mensaje con el resultado de la partida que es ignorado por la plataforma, al tratarse de un mensaje con datos idénticos al primer mensaje con el resultado de la partida.

En primer lugar, el diagrama de secuencia con los mensajes intercambiados con el jugador en cuyo juego se produce el final de la partida.

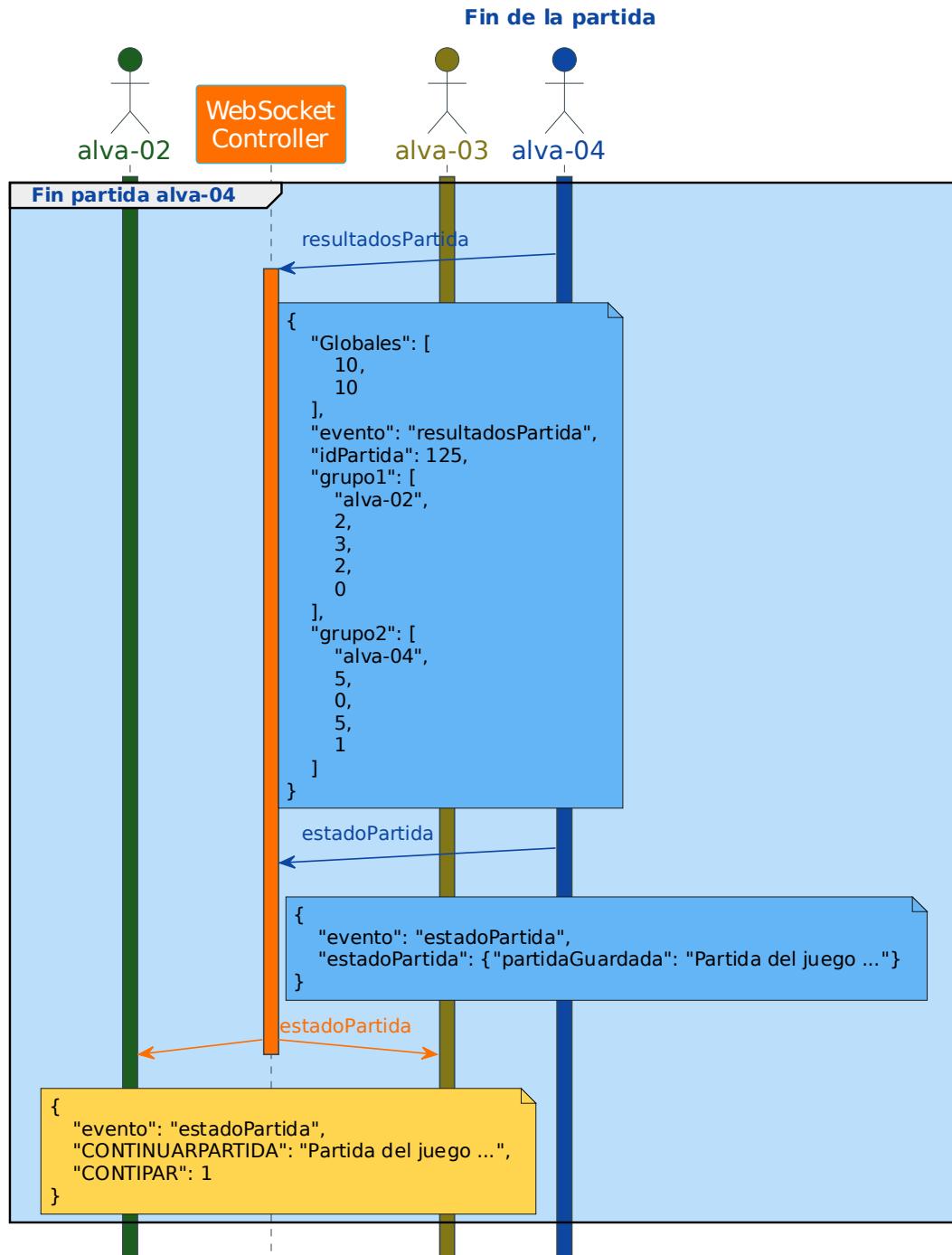


Figura 87: Final partida - jugador alva-04

A continuación, al sincronizarse el estado del juego con el resto de participantes, éstos detectan el final de la partida y envían nuevamente el mismo mensaje con el resultado de la participa.

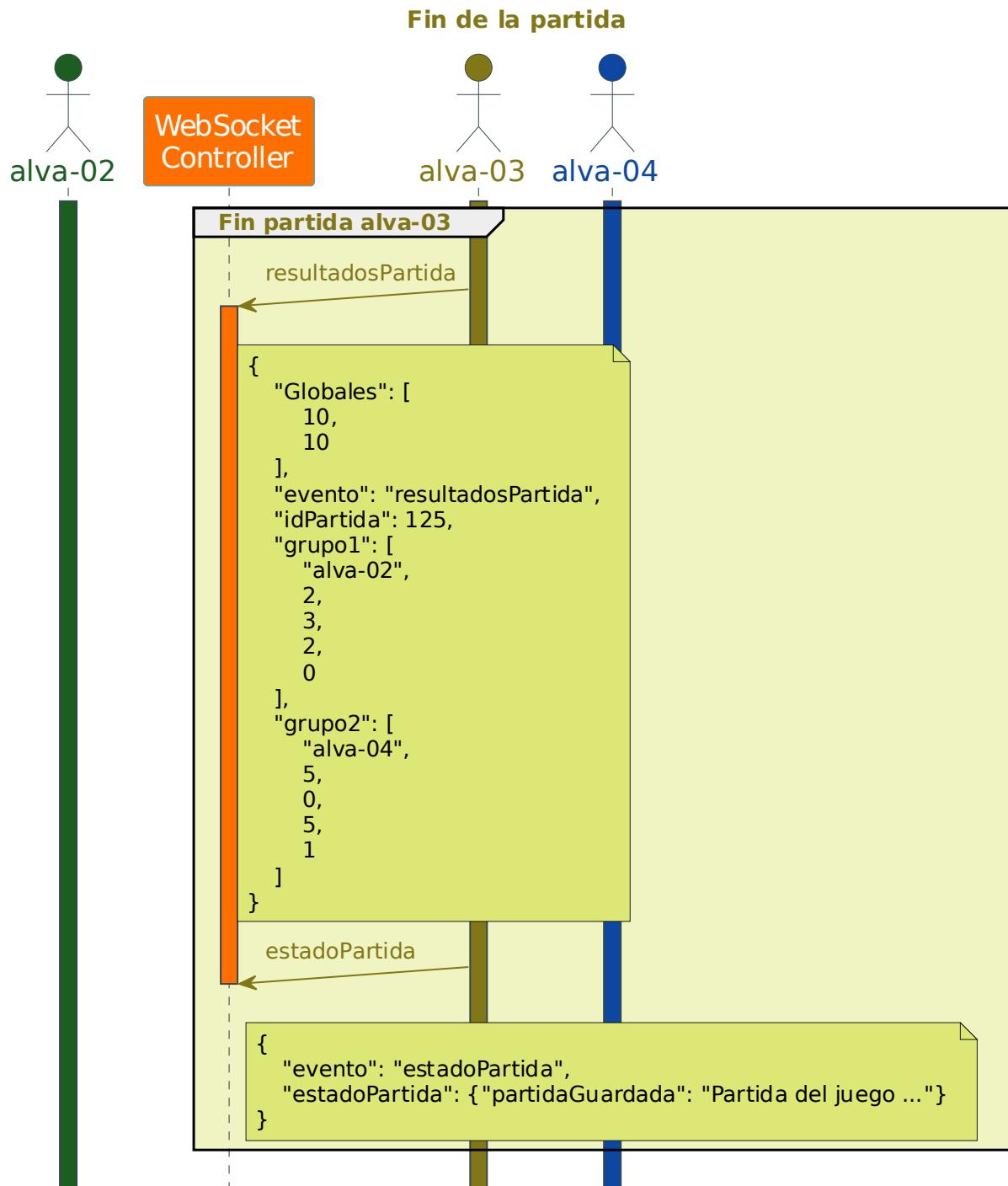


Figura 88: Final partida – espectador alva-03

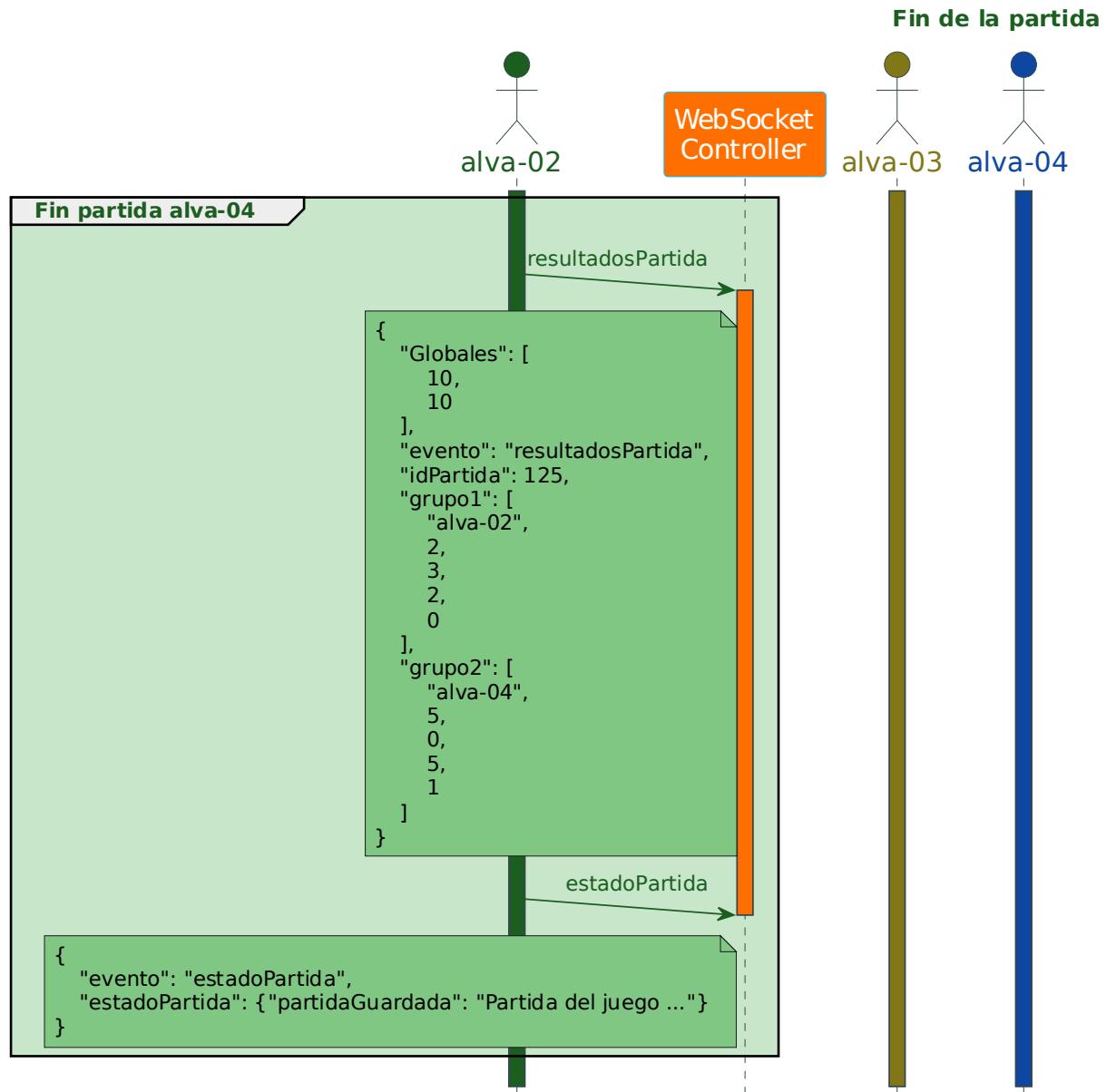


Figura 89: Final partida - jugador alva-02

4.1.7.3 Chat

En el caso de los mensajes utilizados en el chat entre los participantes de una partida, la solicitud de envío de un mensaje por el chat se realiza mediante JSF, pero una vez recibida la petición en la plataforma se envía por WebSocket un mensaje a los destinatarios para que actualicen su lista de mensajes.

Se muestra el diagrama de secuencia del envío de mensaje en el chat de un jugador a otro. El mensaje se envía también al remitente para que actualice su terminal y le aparezca el mensaje como enviado.

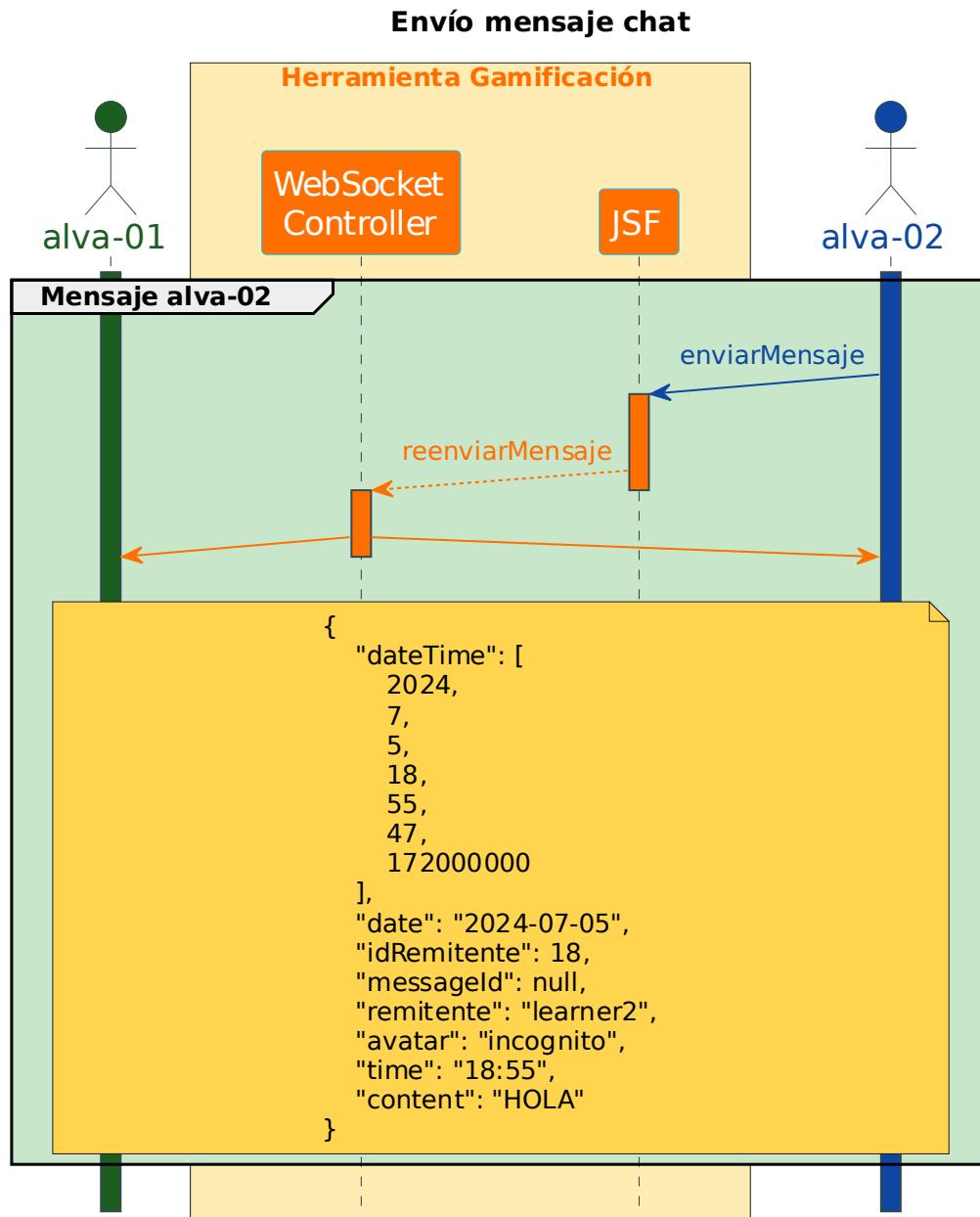


Figura 90: Envío mensaje de chat

4.1.8 Envío calificaciones al LMS

Cuando finaliza la partida se envían al LMS las calificaciones obtenidas por los jugadores, sólo a aquéllos conectados a la partida por LTI, mediante la función *Outcomes service* de la biblioteca Oscelot. A diferencia de los mensajes intercambiados por la conexión WebSocket que transportan los datos en formato JSON, en este caso los datos se envían al LMS en un formato XML definido en la especificación LTI.

Podemos ver el diagrama de secuencia del envío de este mensaje.

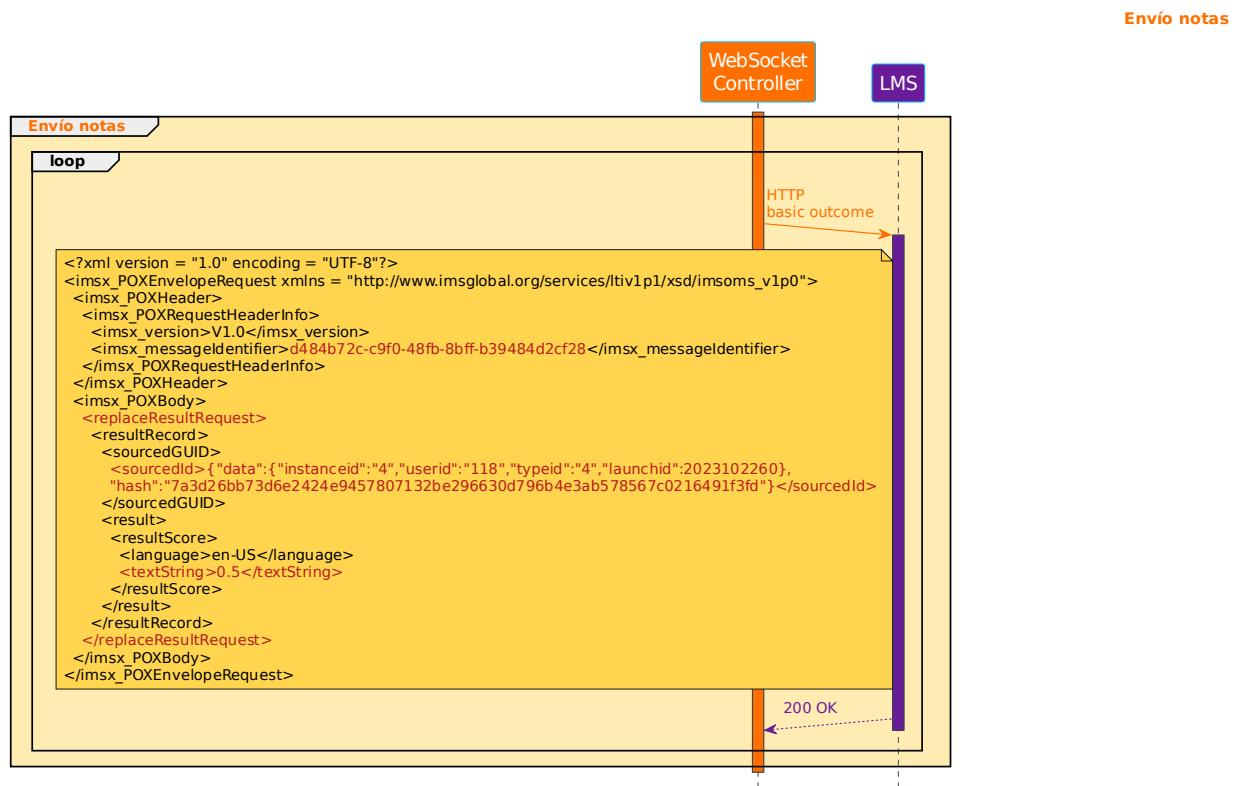


Figura 91: Envío calificaciones al LMS

4.1.9 Configuración

4.1.10 config.properties

Define los parámetros necesarios para la conexión directa por JDBC a la base de datos, necesario para la biblioteca Oscelot.

4.1.11 Descriptores

Se trata de ficheros normalizados para aplicaciones Java con información necesaria para la construcción de la aplicación o su funcionamiento una vez desplegada en un servidor.

4.1.11.1 pom.xml

Fichero *maven* [63] para la construcción de los archivos JAR y WAR usados en la aplicación.

4.1.11.2 web.xml

Se definen parámetros de funcionamiento de JSF y PrimeFaces.

4.1.11.3 jboss-web.xml

Permite definir un contexto de despliegue distinto del nombre del fichero WAR cuando el despliegue de la aplicación se realiza en servidores WildFly o JBoss.

4.1.11.4 persistence.xml

Se define la unidad de persistencia. En nuestro caso se definen los parámetros de conexión a la base de datos, pero también se podría utilizar una conexión definida en el servidor Jakarta EE (WildFly en nuestro caso) usando su nombre JNDI (*Java Naming and Directory Interface* [64], en español Interfaz de Nombrado y Directorio Java).

4.1.11.5 faces-config.xml

Definimos la variable translations [65] para efectuar la internacionalización (i18n) de los ficheros JSF.

4.2 Otras funcionalidades

4.2.1 Autenticación LTI

Cuando se recibe el mensaje LTI de tipo `basic-lti-launch-request`, este contiene información relativa al usuario que se está conectando firmada por el LMS usando el secreto compartido entre TC y TP para garantizar la validez de la misma.

La información para identificar al usuario que se incluye consiste en [66]:

- **user_id:** identifica únicamente al usuario. No debe contener ninguna información que permita conocer la identidad real del usuario. La mejor práctica es que este campo sea una «clave primaria» a largo plazo generada por el TC para el registro de usuario, no la «clave lógica». Como mínimo, este valor debe ser único dentro de un TC.
- **roles:** una lista separada por comas de valores URN (*Uniform Resource Name*, en español Nombre de Recurso Uniforme) para roles. Si esta lista no está vacía, debe contener al menos un rol de los vocabularios *LIS System Role*, *LIS Institution Role* o *LIS Context Role*. El espacio de nombres supuesto de estos URNs es el vocabulario LIS de Roles de Contexto LIS [66]. Si el TC desea incluir un rol de otro espacio de nombres, se debe utilizar un URN completamente cualificado.
- **lis_person_name_given, lis_person_name_family, lis_person_name_full, lis_person_contact_email_primary:** estos campos contienen información sobre la cuenta de usuario que está realizando este lanzamiento. Los nombres de estos datos proceden de LIS. El significado preciso del contenido de estos campos está definido por LIS. Estos parámetros se recomiendan a menos que se supriman debido a la configuración de privacidad.
- **lis_person_sourcedid:** este campo contiene el identificador LIS de la cuenta de usuario que está realizando este lanzamiento. lis_person_sourcedid es simplemente un identificador único (es decir, una cadena normalizada). Este campo es opcional y su contenido y significado están definidos por LIS.

La mayor parte de los LMS utilizados habitualmente utilizan otro campo para enviar esta información al TP. En [38] tenemos información sobre el nombre del campo utilizado por los LMS con más implantación.

- Canvas: **custom_lis_user_username**

- Desire2Learn: **ext_d2l_username**
- Moodle: **ext_user_username**
- BlackBoard: **custom_lis_user_username**
- Sakai: **ext_sakai_eid** or **lis_person_sourcedid**

Con la información recibida en el mensaje de lanzamiento el TP puede identificar al usuario y si accede con el rol de alumno (*Learner*), profesor (*Instructor*) o con ambos.

La plataforma asigna un usuario en la misma a cada valor de **user_id** de un TC (si el mismo **user_id** se repite en varios TC se consideran usuarios distintos) y le asigna un rol de usuario de tipo LTI, lo que indica que el rol con el que actúa depende de la información recibida en el lanzamiento, ya que un mismo usuario de un TC puede actuar como alumno en un curso y como profesor en otro.

Con estos datos se inicia una sesión HTTP que permite tener identificado al usuario que ha accedido al TP mediante un enlace LTI.

La información indicada, junto con información adicional suministrada por el TC en el mensaje de lanzamiento y que es necesaria para la comunicación al mismo del resultado de la partida, se almacena en los datos de la sesión.

Una vez que termina la partida, el TP envía al TC las calificaciones de los jugadores con un mensaje LTI de tipo *Basic Outcome*.

5 PRUEBAS DE EJECUCIÓN

En este apartado se explicará el entorno en el que se han realizado las pruebas y el desarrollo de las mismas.

5.1 Entorno LMS utilizado

Para las pruebas de ejecución se ha utilizado la plataforma Moodle. Para ello hemos utilizado el Moodle del IES Joaquín Turina de Sevilla [67], el cual Jesús Muñoz Calle (autor de los juegos) emplea como profesional de la docencia durante su día a día. En ella se nos ha habilitado un curso llamado Gamificación en el cual se han ejecutado las pruebas con las partidas.

Para ello se han creado un perfil de profesor y 6 alumnos ficticios:

- Profesor:
 - alvaro
- Alumnos:
 - alva-01
 - alva-02
 - alva-03
 - alva-04
 - alva-05
 - alva-06

Se ha escogido este entorno al ser un entorno real en el cual realizar las pruebas.

5.2 Plataforma Gamificación

Para el uso de la plataforma de gamificación necesitamos que esté accesible desde Internet, de forma que se puedan conectar los usuarios, tanto los que se conectan directamente como los que lo hacen mediante un enlace LTI.

Los elementos necesarios para el funcionamiento son una instalación de WildFly para desplegar la aplicación y un servidor de base de datos PostgreSQL. En nuestro caso se ha utilizado un servidor alojado en IONOS en el que se dispone de ambos elementos.

5.3 Pruebas de funcionamiento realizadas y otros datos de interés

Para la ejecución de las pruebas se han creado dentro del curso Gamificación, temas en los que se han clasificado los enlaces para las distintas opciones a las que se puede acceder en la plataforma de gamificación, como se muestra en la siguiente figura.

Figura 92: Contenido del curso de Moodle para realizar las pruebas

Las pruebas que se han realizado con resultado satisfactorio han sido:

- Subida, Edición y Eliminación de Ficheros de preguntas.
- Subida, Activación/Desactivación, Edición y Eliminación de juegos.
- Creación, Edición y Eliminación de Configuraciones de Partida.
- Creación, Edición y Eliminación de Partidas.
- Creación, Edición y Eliminación de Consumidores.
- Creación, Edición y Eliminación de Usuarios.
- Edición de valores de configuración de la plataforma.
- Acceso mediante usuario y contraseña a la plataforma.
- Acceso al módulo de administración por LTI desde el LMS.
- Acceso del iniciador de la partida por LTI desde el LMS.
- Modificación de datos de la partida por parte del iniciador.
- Control de la partida por parte del iniciador, desde su apertura hasta su finalización.
- Acceso de espectadores de la partida por LTI desde el LMS.
- Participación de espectadores en partidas en modo multijugador en las que puede hacer un seguimiento de las respuestas similar al del iniciador.
- Participación de espectadores en partidas en modo distribuido en las que puede ver el funcionamiento del juego, aunque sin poder participar en el mismo.
- Acceso de jugadores de la partida por LTI desde el LMS.
- Participación de jugadores en partidas en modo multijugador en las que puede responder en su propio panel de respuestas.

- Control del tiempo máximo para responder, de forma que si finalizado el tiempo el jugador no ha respondido se envía automáticamente un mensaje de pregunta no respondida.
- Participación de espectadores en partidas en modo distribuido en las que responden en los propios controles del juego cuando les corresponde el turno de respuesta.
- Envío de mensajes por un chat entre los participantes de una partida, con posible selección de los destinatarios.
- Los mensajes se reciben por los destinatarios seleccionados, bien inmediatamente si están conectados al chat o en el momento de conectarse al chat.
- Cuando finaliza la partida se envía el mensaje LTI Basic Outcome al TC para la grabación de la calificación obtenida en la actividad. Este mensaje se envía correctamente y la respuesta del TC es 200 OK. La grabación de la calificación en el LMS depende de cómo éste esté configurado y los parámetros de la actividad en el LMS.

5.4 Pruebas de Rendimiento

El programa responde de forma fluida tanto para el acceso al módulo de Administración como durante el desarrollo de la partida.

Los mensajes del chat se comprueba que se reciben de forma instantánea cuando no hay problemas de conectividad.

5.5 Pruebas de Seguridad

En cuanto a la seguridad se han comprobado satisfactoriamente los siguientes aspectos:

- No se permite el acceso directo a la plataforma a través de URL sin pasar por un *login* previo.
- En el acceso al módulo de administración por LTI únicamente son accesibles los datos correspondientes al TC desde el que se origina la conexión.

Si por error se deja visible el módulo de administración a los alumnos de LMS, éste rechaza correctamente el intento de acceso al no tener el rol autorizado.

6 CONCLUSIONES Y LÍNEAS DE FUTURO

6.1 Conclusiones

Para poder llegar a las conclusiones finales del proyecto, debemos basarnos en los objetivos iniciales del proyecto. Podemos afirmar que hemos conseguido los principales propósitos planteados:

- Actualizar la interfaz de la web de administración a tecnología JSF.
- Migración Base de Datos de SQLite a PostgreSQL.
- Implementación partidas distribuidas.
- Implementación chat entre usuarios.

Una vez cumplidos los objetivos marcados en el proyecto, podemos plantearnos las conclusiones del mismo.

La actualización a JSF, va a permitir una herencia más sencilla del proyecto, ya que al ser un lenguaje basado en plantillas permite una comprensión más sencilla del código desarrollado.

A través de la introducción en el aula de la Gamificación llevada a cabo en este proyecto, se consigue darle una visión diferente a la educación en las clases. El enfoque que el hecho de poder impartir la enseñanza a través de juegos, permite darle un matiz muy interesante en el aprendizaje.

Con la implementación de las partidas distribuidas añadimos una funcionalidad que permite multitud de posibilidades, como por ejemplo, que los alumnos puedan realizar las partidas desde casa sin necesidad de la intervención del profesor durante la partida, el cual podrá ver los resultados de las mismas y evaluarlo en función de los objetivos que busque.

Para este funcionalidad de partidas distribuidas era muy necesario la implementación de un chat entre jugadores, de manera que pudieran comunicarse y compartir información durante el transcurso de la partida.

Este enfoque puede ayudar a muchos alumnos a conectar con la materia y motivarlos a aprenderla. Además a través de los juegos multijugador se fomenta el trabajo en grupo y la colaboración de unos alumnos con otros, favoreciendo la integración entre ellos.

Desde el punto de vista del profesor, el modulo de administración le proporciona una gran capacidad de crear actividades LTI, introduciendo una manera de evaluación alternativa, más amena y dinámica para el alumno. Además simplifica el proceso de evaluación ya que es la propia herramienta la que calcula la calificación y la envía a la plataforma LMS, de modo que se puede evaluar de forma ágil.

Otro punto a destacar es que al estar basado en un estándar como LTI, es fácilmente integrable en multitud de plataformas, lo que la dota de un carácter portable muy destacable.

Personalmente, en la elaboración de este trabajo he tenido la ocasión de aplicar tecnologías actuales en la construcción de una aplicación, que resumo a continuación:

1. Programación funcional

En la versión 8 de Java aparecen expresiones lambda [68], que permiten encapsular el código de una función en una expresión sin necesidad de la declaración de un método. Además incorpora algunas interfaces funcionales predefinidas, entre las que podemos destacar:

- *Supplier* (suministrador): devuelve un dato sin recibir ningún parámetro, corresponde con una función sin parámetros.
- *Consumer* (consumidor): realiza una acción para la que recibe un parámetro, pero no devuelve ninguna información. Se corresponde con un procedimiento con un sólo parámetro.
- *Function* (función): fragmento de código que recibe un parámetro y devuelve un valor, corresponde con una función que sólo recibe un parámetro.

2. Java Streams

Se trata la nueva *Stream API* [69], que apareció en la versión 8 de Java. Se trata de un API para el procesamiento de secuencias de elementos. Para extender su uso a otros objetos es posible realizar la conversión de listas y mapas a secuencias de objetos y viceversa.

3. CDI [34]

Permite acceder fácilmente a datos presentes en el contexto de ejecución, ya que es el contenedor el que se encarga de gestionar los objetos adecuados en cada momento.

4. JSF [6]

Permite la construcción de interfaces de usuario de forma que se puede dar un aspecto homogéneo a la aplicación fácilmente y reutilizar fragmentos de código en varias páginas.

También ofrece soporte para internacionalización (i18n) y localización (l10n) [70] de las interfaces de usuario. La internacionalización consiste en la adecuación de una aplicación para que soporte diversos idiomas, y la localización en la definición de los textos que tienen que aparecer en cada uno de los posibles idiomas utilizados en la aplicación.

Además, permite una interacción fluida gracias a su uso de Ajax para actualizaciones de la página de forma transparente.

5. JPA [35]

Simplifica el acceso a la base de datos, encapsulando gran parte de las operaciones necesarias para establecer una conexión JDBC con una base de datos relacional.

6. WebSockets [7]

Permite la comunicación bidireccional entre el servidor y el navegador en una aplicación web, de forma que se pueden iniciar operaciones en el navegador iniciadas por un evento en el servidor sin que sea necesaria la intervención del usuario.

6.2 Líneas de Futuro

Como principales líneas de futuro, sería muy interesante la adaptación a nuevas versiones de LTI, actualmente la versión utilizada en el proyecto es la 1.1.

Una actualización a la versión de LTI 1.3 [71] nos permitiría la utilización de OAuth 2.0 [72] para tener una autenticación más segura.

Por otro lado LTI 1.3 ofrece un mejor soporte para dispositivos móviles, lo que lo hace más adaptable a los entornos de aprendizaje actuales.

Otra línea de actuación es la actualización de la tecnología de Jakarta EE de la versión 8 a la actual versión 10, ya que es la versión soportada por las últimas versiones de los servidores de aplicaciones.

7 ANEXOS

7.1 Anexo A Manual de instalación

7.1.1 Preparación del entorno

1. Instalación de Eclipse: para este proyecto se ha utilizado la versión de Eclipse 2021-12, el cual se puede descargar desde la página de eclipse [73].
2. Instalación de Java: para este proyecto se ha utilizado la versión de JDK 8. La cual se puede descargar desde la página de OpenJDK [74].
3. Instalación de WildFly: para este proyecto se ha utilizado la versión WildFly 23. La cual se puede descargar desde la página de WildFly [75].
4. Opcionalmente se puede instalar un visualizador de base de datos, para este proyecto hemos utilizado: SQL Workbench/J [76] y PgAdmin 4 [77].

7.1.2 Fichero de configuración “config.properties”

Previamente a iniciar la aplicación será necesario configurar el fichero “config.properties” situado en WEB-INF. Los parámetros a configurar son los siguientes son los siguientes:

- dbHost: equipo en el que se encuentra el servidor de bases de datos.
- dbName: nombre de la base de datos utilizada en el proyecto.
- dbUsername: usuario con el que se accede a la base de datos.
- dbPassword: contraseña del usuario con el que se accede a la base de datos.

7.1.3 Configuración Eclipse y WildFly

Una vez tengamos eclipse instalado en nuestro equipo debemos importar el proyecto, para ello pulsaremos “File → Import → Projects from folder or Archive” y buscaremos el directorio donde tengamos guardado el proyecto. Una vez importado con éxito, nos aparecerá la pestaña “package explorer” .

Una vez importado nuestro proyecto, procederemos a la instalación del servidor: “File → new → server”, e instalaremos la versión WildFly 23 indicada en el apartado anterior. Nos deberá aparecerá ahora el servidor que acabamos de instalar en la pestaña “Servers”. Para iniciar lo pulsaremos con el botón derecho sobre el servidor y seleccionamos “start”, para poder acceder a la aplicación desde un navegador.

7.1.4 Creación de base de datos

Es necesario crear una base de datos en el servidor PostgreSQL, que configuraremos en la aplicación (config.properties y persistence.xml). La aplicación crea automáticamente los parámetros de configuración y un superusuario para permitir acceder a la misma, en caso de que no existan en las tablas correspondientes.

7.2 Anexo B Manual de usuario

La gestión del módulo de administración, tanto la gestión de usuarios, consumidores, partidas, juegos queda detallada en el apartado 4.1.3 Ficheros JSF.

En lo que respecta a la configuración del LMS, explicaremos el caso de Moodle, ya que es la plataforma donde se han realizado las pruebas:

Dentro del Curso donde se vaya a realizar la actividad se deberá activar la edición:

The screenshot shows the Moodle 'Gamificación' page. At the top right, a context menu is open with the following options: 'Editar ajustes', 'Activar edición' (which is checked), 'Finalización del curso', 'Filtros', 'Configuración Calificaciones', 'Copia de seguridad', 'Restaurar', 'Importar', 'Reiniciar', 'Competencias', and 'Más ...'. Below the menu, the course structure is visible with four main sections: 'TEMA 1 - MODULO ADMINISTRACIÓN', 'TEMA 2 - PARTIDA LTI MONOJUGADOR', 'TEMA 3 - PARTIDA LTI MULTIJUGADOR', and 'TEMA 4 - PARTIDA LTI DISTRIBUIDA'. Each section contains various activity icons like 'Avisos', 'Consola gamificación', 'Partida LTI/mono', 'Partida multijugador', 'Espectador multijugador', and 'Partida distribuida'. Each section has an 'Editar' button at the top right.

Figura 93: Moodle - Activar Edición

En el Tema donde se desee realizar la actividad se hace click en “+ Añadir una actividad o un recurso”

This screenshot shows the same Moodle 'Gamificación' page as Figure 93, but with a specific action highlighted. In each of the four course sections ('TEMA 1 - MODULO ADMINISTRACIÓN', 'TEMA 2 - PARTIDA LTI MONOJUGADOR', 'TEMA 3 - PARTIDA LTI MULTIJUGADOR', and 'TEMA 4 - PARTIDA LTI DISTRIBUIDA'), the '+ Añadir una actividad o un recurso' button is circled in red. This indicates where the user should click to add a new activity or resource to the respective section.

Figura 94: Moodle - Añadir una actividad o un recurso

Se nos abrirá el siguiente menú e indicaremos “Herramienta Externa” y hacemos click en “Añadir”

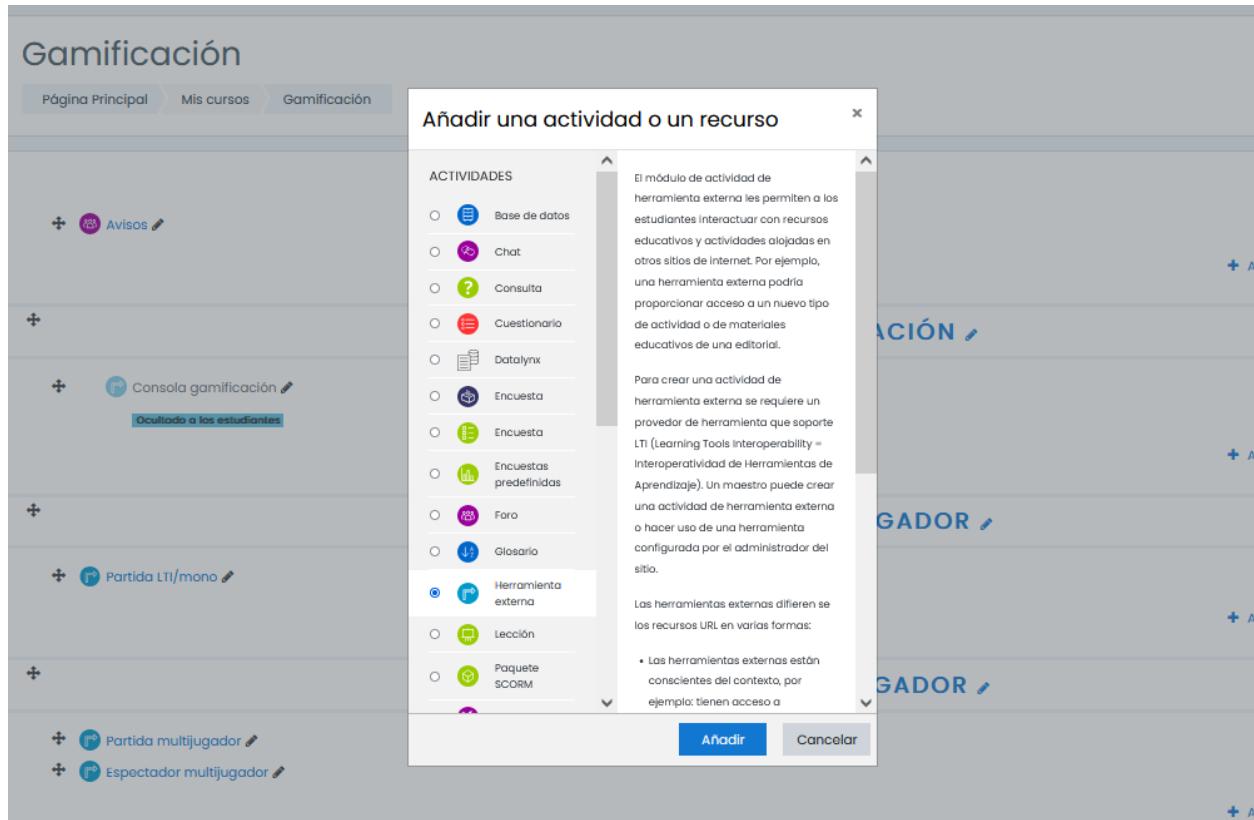


Figura 95: Moodle - Añadir Herramienta Externa

Esto nos llevará a la siguiente pagina:

Figura 96: Moodle - Configuración Herramienta Externa

En ella deberemos configurar los siguientes campos:

- Nombre de la actividad: indicaremos el nombre de la actividad con la que aparecerá en el curso.

- URL de herramienta segura: indicaremos la URL de nuestra herramienta, en nuestro caso será: https://gamifica.fjfj.es/gamificacion-tfg_AlvaroSantiago_Cuenca/games
- Iniciar el contenedor: seleccionaremos → “Nueva ventana”.
- Clave de cliente: la indicada por el superusuario de la aplicación de Gamificación.
- Secreto compartido: la indicada por el superusuario de la aplicación de Gamificación.
- Parámetros personalizados: aquí dependiendo de lo que necesitemos indicaremos diversos parámetros:
 - management=true para el acceso LTI al módulo de administración.
 - codigo_partida=XXXXXXX, donde XXXXXXXX será el código (IDP) de la partida creada en el módulo de administración.
 - espectador=true, este parámetro se añadirá adicionalmente al código de la partida, para indicar que este enlace será solo de Espectador.

Quedando de la siguiente forma:

The screenshot shows the 'Actualizando Herramienta externa en TEMA 1 - MODULO ADMINISTRACIÓN' section of the Moodle administrator interface. Under the 'General' tab, the 'Nombre de la actividad' is set to 'Consola gamificación'. The 'Descripción de la actividad' field contains a rich text editor. Below these, there are three checkboxes: 'Muestra la descripción en la página del curso' (unchecked), 'Nombre para mostrar la actividad cuando se inicia' (checked), and 'Mostrar la descripción de la actividad cuando se inicia' (unchecked). The 'Herramienta preconfigurada' dropdown is set to 'Automático, basado en la URL de la herramienta'. The 'URL de la herramienta' field contains 'https://gamifica.fjfj.es/gamificacion-tfg_AlvaroSantiago_Cuenca/games'. The 'Iniciar el contenedor' dropdown is set to 'Nueva ventana'. The 'Clave de cliente' and 'Secreto compartido' fields are filled with '073aa0faf-feb7-49cc-846a-8728006c' and '.....' respectively. The 'Parámetros personalizados' field contains 'management=true'. The 'URL de icono' and 'URL de icono seguro' fields are empty. At the bottom, there is a 'Ver menos...' link.

Figura 97: Moodle - Configuración Herramienta Externa para acceso al módulo de Administración

Actualizando Herramienta externa en TEMA 3 – PARTIDA LTI MULTIJUGADOR

General

Nombre de la actividad: Partida multijugador

Descripción de la actividad:

Herramienta preconfigurada: Automático, basado en la URL de la herramienta

URL de la herramienta: https://gamifica.tff.es/gamificacion-ttg_AlvaroSantiago_Cuenca/games

URL de herramienta segura:

Iniciar el contenedor: Nueva ventana

Clave de cliente: 073aeafaf-feb7-49cc-846a-9728006c

Secreto compartido:@*

Parámetros personalizados: codigo_partida=28800902838

URL de ícono:

URL de ícono seguro:

[Ver menos...](#)

Figura 98: Moodle - Configuración Herramienta Externa para acceso a la partida

Actualizando Herramienta externa en TEMA 3 – PARTIDA LTI MULTIJUGADOR

General

Nombre de la actividad: Espectador multijugador

Descripción de la actividad:

Herramienta preconfigurada: Automático, basado en la URL de la herramienta

URL de la herramienta: https://gamifica.tff.es/gamificacion-ttg_AlvaroSantiago_Cuenca/games

URL de herramienta segura:

Iniciar el contenedor: Defecto

Clave de cliente: 073aeafaf-feb7-49cc-846a-9728006c

Secreto compartido:@*

Parámetros personalizados: codigo_partida=28800902838
spectador=true

URL de ícono:

Figura 99: Moodle - Configuración Herramienta Externa para acceso a la partida como espectador

REFERENCIAS

- [1] *Qué es la gamificación en el aula, ventajas y cómo aplicarla* | UNIR, 2020 UNIVERSIDAD INTERNACIONAL DE LA RIOJA, S.A.: <https://www.unir.net/educacion/revista/gamificacion-en-el-aula/>
- [2] Lobón Roldán, A.M., *Aplicación de gestión de juegos para la educación con framework Spring y Primefaces*, 2020 Universidad de Sevilla: <https://biblus.us.es/bibing/proyectos/abreproj/93013>
- [3] Jiménez Vázquez, A., *APLICACIÓN WEB MULTIUUSUARIO PARA LA GAMIFICACIÓN EN EL AULA BASADA EN WEBSOCKET*, 2019 Universidad de Sevilla: <https://biblus.us.es/bibing/proyectos/abreproj/92225>
- [4] García Frutos, P., *INTEGRACIÓN DE HERRAMIENTAS DE GAMIFICACIÓN EN PLATAFORMAS DE ENSEÑANZA VIRTUAL*, 2018 Universidad de Sevilla: <https://biblus.us.es/bibing/proyectos/abreproj/91933>
- [5] Proyecto ADJA, Red Educativa Digital Descartes: https://proyectodescartes.org/newton/juegosdidacticos/index.php?option=com_content&view=article&id=132&Itemid=469&lang=es
- [6] Jakarta Server Faces 2.3, Eclipse Foundation: <https://jakarta.ee/specifications/faces/2.3/>
- [7] Fette I., Melnikov A., *RFC 6455 - The WebSocket Protocol*, 2011 Internet Engineering Task Force (IETF): <https://datatracker.ietf.org/doc/html/rfc6455>
- [8] Kantor I., *WebSocket*, 2022 The Modern JavaScript Tutorial: <https://es.javascript.info/websocket>
- [9] Pavey C., *Protocolos de comunicación WebSocket vs. HTTP*, Sendbird: <https://sendbird.com/es/developer/tutorials/websocket-vs-http-communication-protocols>
- [10] ECMA-262, 2024 Ecma International: <https://ecma-international.org/publications-and-standards/standards/ecma-262/>
- [11] HTML Standard, 2024 WHATWG: <https://html.spec.whatwg.org/>
- [12] ISO 8879:1986/Cor 2:1999 *Information processing - Text and office systems - Standard Generalized Markup Language (SGML)*, 1999 International Organization for Standardization: <https://www.iso.org/standard/30235.html>
- [13] W3C, 2024 World Wide Web Consortium.: <https://www.w3.org/>
- [14] CSS Snapshot 2023, 2023 World Wide Web Consortium.: <https://www.w3.org/TR/css-2023/>
- [15] Jakarta EE Platform 8 | Standards for Jakarta EE Applications, 2019 The Eclipse Foundation: <https://jakarta.ee/specifications/platform/8/>
- [16] PrimeFaces - Ultimate UI Framework, 2024 PrimeTek Informatics: <https://www.primefaces.org/>
- [17] Tecnologías Web Actuales, 2022 PROUN MEDIA S.L.: <https://www.proun.es/blog/tecnologias-web-actuales/>
- [18] PHP: Hypertext Preprocessor, 2024 The PHP Group: <https://www.php.net/>
- [19] Welcome to Python.org, 2024 Python Software Foundation: <https://www.python.org/>
- [20] ISO/IEC 9899:2018 *Information technology - Programming languages - C*, 2018 International Organization for Standardization: <https://www.iso.org/standard/74528.html>
- [21] PostgreSQL: The World's Most Advanced Open Source Relational Database, 2024 The PostgreSQL Global Development Group: <https://www.postgresql.org/>
- [22] SQLite Home Page, 2024 SQLite Consortium: <https://www.sqlite.org/index.html>

- [23] Morgan, G., *Observations on the Higher Education LMS Market*, 2024 On EdTech Newsletter - Phil Hill & Associates LLC: <https://onetech.philhillaa.com/p/observations-on-higher-ed-lms-market>
- [24] *Learning Tools Interoperability (LTI)® Security Update*, 2019 1EdTech Consortium, Inc.: <https://www.imsglobal.org/specs/ltil1p1>
- [25] *About 1EdTech*, 2024 1EdTech Consortium, Inc.: <https://www.1edtech.org/about/1edtech>
- [26] *IMS Global*, 2020 Wikimedia Foundation: https://es.wikipedia.org/wiki/IMS_Global
- [27] *IMS Global Learning Consortium estandariza el conocimiento para la interoperabilidad de las tecnologías educativas*, OCT 2016 E-LEARN MAGAZINE
- [28] Parecki, A., *OAuth Community Site*, : <https://oauth.net/>
- [29] *Recipe for Making LTI® 1 Tool Providers*, 2023 1EdTech Consortium Inc.: <https://www.imsglobal.org/recipe-making-lti-1-tool-providers>
- [30] *Learning Tools Interoperability*, 2024 1EdTech Consortium Inc.: <https://www.1edtech.org/standards/lti>
- [31] *Jakarta Expression Language 3.0*, 2019 Eclipse Foundation: <https://jakarta.ee/specifications/expression-language/3.0/>
- [32] *Jakarta Managed Beans 1.0*, 2019 Eclipse Foundation: <https://jakarta.ee/specifications/managedbeans/1.0/>
- [33] *JavaBeans Spec*, 2019 Eclipse Foundation: <https://www.oracle.com/java/technologies/javase/javabeans-spec.html>
- [34] *Jakarta Contexts and Dependency Injection*, 2019 Eclipse Foundation: <https://jakarta.ee/specifications/cdi/2.0/>
- [35] *Introduction to Jakarta Persistence*, Eclipse Foundation: <https://jakarta.ee/learn/docs/jakartaee-tutorial/current/persist/persistence-intro/persistence-intro.html>
- [36] *IETF | Internet Engineering Task Force*, Internet Engineering Task Force: <https://www.ietf.org/>
- [37] Moriarty K., Kaliski B., Rusch A., *RFC 8018 - PKCS #5: Password-Based Cryptography Specification Version 2.1*, 2017 Internet Engineering Task Force (IETF): <https://www.rfc-editor.org/rfc/rfc2898#section-5.2>
- [38] *LMS username parameter - Ex Libris Developer Network*, Ex Libris Ltd.: <https://developers.exlibrisgroup.com/leganto/integrations/lti/troubleshooting/user-problems/lms-username-parameter/>
- [39] *PrimeFlex - Utility-First CSS Library*, 2023 PrimeTek Informatics: <https://primeflex.org/>
- [40] Scholtz B., Tijms A., *OmniFaces | To make JSF life easier*, OmniFaces: <https://omnifaces.org/>
- [41] *Font Awesome*, Fonticons, Inc.: <https://fontawesome.com/>
- [42] *PrimeFaces Bootstrap Theme*, : <https://mvnrepository.com/artifact/org.primefaces.themes/bootstrap/1.0.11>
- [43] *Bootstrap · The most popular HTML, CSS, and JS library in the world.*, Bootstrap team: <https://getbootstrap.com/>
- [44] *PostgreSQL JDBC Driver*, 2023 The PostgreSQL Global Development Group: <https://jdbc.postgresql.org/>
- [45] *Your relational data. Objectively. - Hibernate ORM*, Commonhaus Foundation: <https://hibernate.org/orm/>
- [46] *GitHub - 1EdTech/basiclti-util-java*, 2014 IMS Global Learning Consortium, Inc.: <https://github.com/1EdTech/basiclti-util-java>
- [47] *GitHub - google/gson: A Java serialization/deserialization library to convert Java Objects into JSON and back*, : <https://github.com/google/gson>

- [48] *GitHub - FasterXML/jackson: Main Portal page for the Jackson project*, Jackson Project: <https://github.com/FasterXML/jackson>
- [49] *JDOM, JDOM*: <https://mvnrepository.com/artifact/org.jdom/jdom2/2.0.6.1>
- [50] Conway S., Jones A.R., *OpenCSV*, : <https://opencsv.sourceforge.net/>
- [51] *The Leading PDF Library for Developers | iText*, Apryse: <https://itextpdf.com/>
- [52] *Docx4J*, Plutext Pty Ltd: <https://www.docx4java.org/trac/docx4j>
- [53] *Simple Logging Facade for Java (SLF4J)*, 2024 QOS.ch Sarl: <https://www.slf4j.org/>
- [54] *Commons Lang*, 2024 The Apache Software Foundation: <https://commons.apache.org/proper/commons-lang/>
- [55] *Apache Commons*, 2024 The Apache Software Foundation: <https://commons.apache.org/>
- [56] *Commons BeanUtils*, 2024 The Apache Software Foundation: <https://commons.apache.org/proper/commons-beanutils/>
- [57] *Java port of universalchardet*, : <https://code.google.com/archive/p/juniversalchardet/>
- [58] *Spring Framework*, Broadcom Inc.: <https://spring.io/projects/spring-framework>
- [59] Leary S., *GitHub - stleary/JSON-java: A reference implementation of a JSON package in Java.*, 2024 : <https://github.com/stleary/JSON-java>
- [60] Vickers S., *Java LTI Tool Provider package*, 2014 SPV Software Products: http://www.spvsoftwareproducts.com/java/lti_tool_provider/
- [61] *javax.faces.convert.Converter*, 2015 Oracle America, Inc.: <https://docs.oracle.com/javaee/7/api/javax/faces/convert/Converter.html>
- [62] *javax.persistence.AttributeConverter*, 2015 Oracle America, Inc.: <https://docs.oracle.com/javaee/7/api/javax/persistence/AttributeConverter.html>
- [63] *Apache Maven Project*, 2024 The Apache Software Foundation: <https://maven.apache.org/>
- [64] *Java Naming and Directory Interface (JNDI)*, 2024 Oracle and/or its affiliates: <https://docs.oracle.com/javase/8/docs/technotes/guides/jndi/index.html>
- [65] Kumar P., *JSF Resource Bundle, Custom Messages Example Tutorial*, 2016 JournalDev IT Services Private Limited: <https://web.archive.org/web/20220521211653/https://www.journaldev.com/6694/jsf-resource-bundle-custom-messages-example-tutorial>
- [66] *Learning Tools Interoperability v1.1 Implementation Guide | IMS Global Learning Consortium*, 1EdTech Consortium, Inc.: <https://www.imsglobal.org/specs/ltiv1p1/implementation-guide>
- [67] *mTurina - Plataforma educativa del IES Joaquín Turina de Sevilla*, IES Joaquín Turina: <https://mfor.iesjoaquinTurina.es/>
- [68] *Lambda Expressions (The Java™ Tutorials > Learning the Java Language > Classes and Objects)*, Oracle and/or its affiliates.: <https://docs.oracle.com/javase/tutorial/java/javaOO/lambdaexpressions.html>
- [69] *Package java.util.stream*, Oracle and/or its affiliates: <https://docs.oracle.com/javase/8/docs/api/java/util/stream/package-summary.html>
- [70] Ishida R., Miller S.K., *Localization vs. Internationalization*, 2005 W3C: <https://web.archive.org/web/20160403134943/http://www.w3.org/International/questions/qa-i18n/>
- [71] *Learning Tools Interoperability Core Specification 1.3 | IMS Global Learning Consortium*, 2019 1EdTech Consortium Inc.: <https://www.imsglobal.org/spec/lti/v1p3>
- [72] *¿Qué es OAuth 2.0 y para qué sirve? - Auth0*, Okta, Inc: <https://auth0.com/es/intro-to-iam/what-is-oauth-2>
- [73] *Eclipse IDE 2021-12 R Packages*, 2021 Eclipse Foundation: <https://www.eclipse.org/downloads/packages/release/2021-12/r>

[74] *OpenJDK*., 2014 Oracle Corporation and/or its affiliates: <https://openjdk.org/projects/jdk8/>

[75] Stansberry B., *WildFly 23 is released!*, 2021 Red Hat Middleware Community Projects: <https://www.wildfly.org/news/2021/03/11/WildFly23-Final-Released/>

[76] Kellerer T., *SQL Workbench/J*, 2024 : <https://www.sql-workbench.eu/>

[77] *pgAdmin - PostgreSQL Tools*, 2024 pgAdmin Development Team:

<https://www.pgadmin.org/download/pgadmin-4-windows/>