

Auditoría de seguridad a un entorno realista simulado

Trabajo de Fin de Grado

GRADO EN INGENIERÍA INFORMÁTICA

Anexo III. Diseño del Sistema Software



**VNiVERSiDAD
D SALAMANCA**

Julio de 2023

Autor

Marcos Panero Calles

Tutor/a

Ángel Luis Sánchez Lázaro

Tabla de contenido

1. Introducción	4
2. Modelo de diseño.....	5
2.1. Patrones arquitectónicos.....	6
2.1.1 Patrón Cliente-Servidor	6
2.1.2 Patrón Layers (Capas)	7
2.2 Clases de diseño	8
2.2.1 Capa de Presentación.....	8
2.2.1 Capa de Aplicación	9
2.2.1 Capa de Datos.....	9
3. Realización de casos de uso	10
3.1 Gestión de Usuarios	10
3.1.1 Caso de Uso Iniciar Sesión.....	10
3.1.2 Caso de Uso Registrarse	10
3.1.3 Caso de Uso Cerrar Sesión.....	11
3.2 Gestión de Plataforma.....	11
3.2.1 Caso de Uso Subir Tarea	11
3.2.2 Caso de Uso Crear Tarea.....	11
3.2.3 Caso de Uso Corregir Tarea	12
3.2.4 Caso de Uso Mostrar Área Alumno	12
3.2.5 Caso de Uso Mostrar Área Profesor	12
3.2.6 Caso de Uso Mostrar Área Administrador.....	13
3.2.7 Caso de Uso Enviar Código Registro	13
3.2.8 Caso de Uso Mostrar Área Tienda	13
3.2.9 Caso de Uso Gestionar Compra.....	14
3.3 Gestión de Accesos No Autorizados.....	14
3.3.1 Caso de Uso Reportar Acceso No Autorizado	14
4. Diagrama de despliegue	15
5. Bibliografía.....	16

Tabla de Ilustraciones

Ilustración 1: Esquema patrón Cliente-Servidor.....	6
Ilustración 2: Arquitectura web de 3 niveles.....	7
Ilustración 3: Modelo arquitectónico de 3 capas.....	8
Ilustración 4: Capa de Presentación.....	8
Ilustración 5: Capa de Aplicación.....	9
Ilustración 6: Capa de Datos.....	9
Ilustración 7: Diagrama Secuencia Iniciar Sesión.....	10
Ilustración 8: Diagrama Secuencia Registrarse.....	10
Ilustración 9: Diagrama Secuencia Cerrar Sesión.....	11
Ilustración 10: Diagrama Secuencia Subir Tarea.....	11
Ilustración 11: Diagrama Secuencia Crear Tarea.....	11
Ilustración 12: Diagrama Secuencia Corregir Tarea.....	12
Ilustración 13: Diagrama Secuencia Mostrar Área Alumno.....	12
Ilustración 14: Mostrar Área Profesor.....	12
Ilustración 15: Diagrama Secuencia Mostrar Área Administrador.....	13
Ilustración 16: Diagrama Secuencia Enviar Código Registro.....	13
Ilustración 17: Mostrar Área Tienda.....	13
Ilustración 18: Diagrama Secuencia Gestionar Compra.....	14
Ilustración 19: Diagrama Secuencia Reportar Acceso No Autorizado.....	14
Ilustración 20: Diagrama de despliegue.....	15

1. Introducción

El objetivo del presente documento es documentar el diseño del sistema software implementado, siguiendo las directrices establecidas en el Anexo II: Especificación y análisis de requisitos. En este documento se proporcionará una explicación detallada de los aspectos relacionados con el diseño del sistema.

Se abordarán diversos temas, entre los cuales se incluyen el modelo de diseño utilizado, los patrones arquitectónicos seguidos y las clases de diseño definidas. Además, se realizarán y describirán los diagramas de secuencia correspondientes a los diferentes casos de uso del sistema. Por último, se presentará y se explicará el diagrama de despliegue de la aplicación.

2. Modelo de diseño

A la hora de comenzar el desarrollo de cualquier proyecto un aspecto importante a tener en cuenta es la realización de un análisis del patrón arquitectónico que se va a utilizar durante toda la implementación.

Al analizar un patrón arquitectónico, se busca identificar cómo se organiza el sistema, qué componentes lo componen, cómo interactúan entre sí y cómo se cumplen los requisitos funcionales y no funcionales.

Durante el desarrollo de la aplicación UsalCoins se utilizarán las siguientes tecnologías:

- **Apache:** Es un servidor web de código abierto ampliamente utilizado. Proporciona un entorno para alojar y servir archivos web a través del protocolo HTTP. Define una arquitectura basada en el patrón Cliente-Servidor.
- **PHP:** Lenguaje de programación del lado del servidor. Permite implementar las funciones encargadas de llevar a cabo las distintas tareas lógicas que debe realizar el sistema.
- **MySQL:** Sistema de gestión de bases de datos relacional de código abierto. Permite la creación de estructuras de base de datos, consultas y manipulación de datos de manera eficiente y segura.
- **HTML, CSS y JavaScript:** Tecnologías utilizadas para la implementación de las diferentes vistas de las páginas web de la plataforma.

2.1. Patrones arquitectónicos

Un patrón arquitectónico es una solución probada y comúnmente aceptada para un problema recurrente en el diseño de software. Es como un plan o una estructura general que se utiliza para organizar y construir sistemas de software. Los patrones arquitectónicos proporcionan pautas y mejores prácticas que han sido desarrolladas y refinadas a lo largo del tiempo por expertos en el campo de la ingeniería de software.

2.1.1 Patrón Cliente-Servidor

En el patrón Cliente-Servidor, el sistema se divide en dos componentes principales: el cliente y el servidor.

El cliente es la parte de la aplicación que interactúa directamente con el usuario final. Puede ser una aplicación de escritorio, una aplicación móvil o un navegador web. El cliente envía solicitudes al servidor y espera recibir las respuestas correspondientes.

El servidor, por otro lado, es la parte de la aplicación que procesa las solicitudes del cliente y envía las respuestas. Puede ser un servidor web, un servidor de base de datos u otro tipo de servidor especializado. El servidor tiene la capacidad de recibir solicitudes de múltiples clientes y responderles de manera adecuada.

La comunicación entre el cliente y el servidor se realiza a través de un protocolo de red, como HTTP. El cliente envía una solicitud al servidor con una determinada acción o petición, y el servidor procesa esa solicitud y envía una respuesta al cliente con los resultados o los datos solicitados.

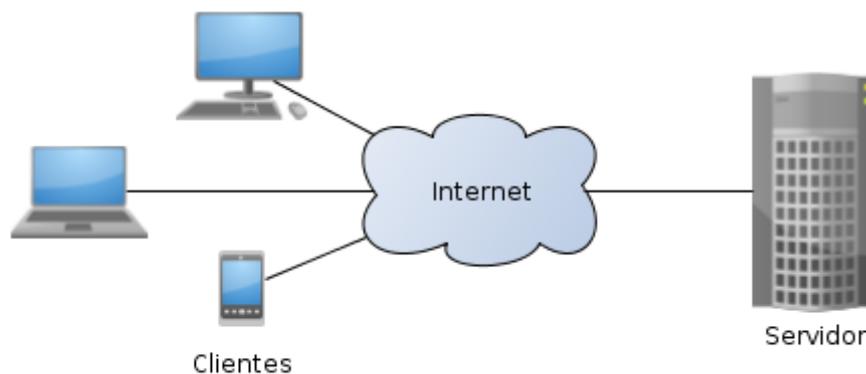


Ilustración 1: Esquema patrón Cliente-Servidor

2.1.2 Patrón Layers (Capas)

Para organizar la estructura de la aplicación web en diferentes capas y separar responsabilidades también se considera el uso del patrón arquitectónico Layers. Dividiendo la aplicación en 3 capas o niveles:

1. **Capa de Presentación:** Esta capa se encarga de la interacción con el usuario final. Es responsable de recibir las solicitudes del usuario y mostrar los resultados de manera adecuada. Su objetivo es presentar la información de una manera comprensible y amigable para el usuario. Puede incluir elementos como interfaces gráficas de usuario, páginas web o servicios de API.
2. **Capa de Aplicación:** Esta capa contiene la lógica y reglas de negocio de la aplicación. Aquí se procesan y manipulan los datos recibidos de la capa de presentación. La capa de lógica de la aplicación se encarga de realizar las operaciones necesarias para cumplir con los requisitos funcionales y las reglas establecidas. Puede incluir componentes como controladores, servicios y clases que implementen la lógica empresarial.
3. **Capa de Datos:** Esta capa se encarga de gestionar y almacenar los datos utilizados por la aplicación. Puede ser una base de datos relacional, una base de datos NoSQL u otras fuentes de datos como archivos XML. La capa de datos proporciona los mecanismos para acceder, recuperar, almacenar y actualizar la información necesaria para el funcionamiento de la aplicación.

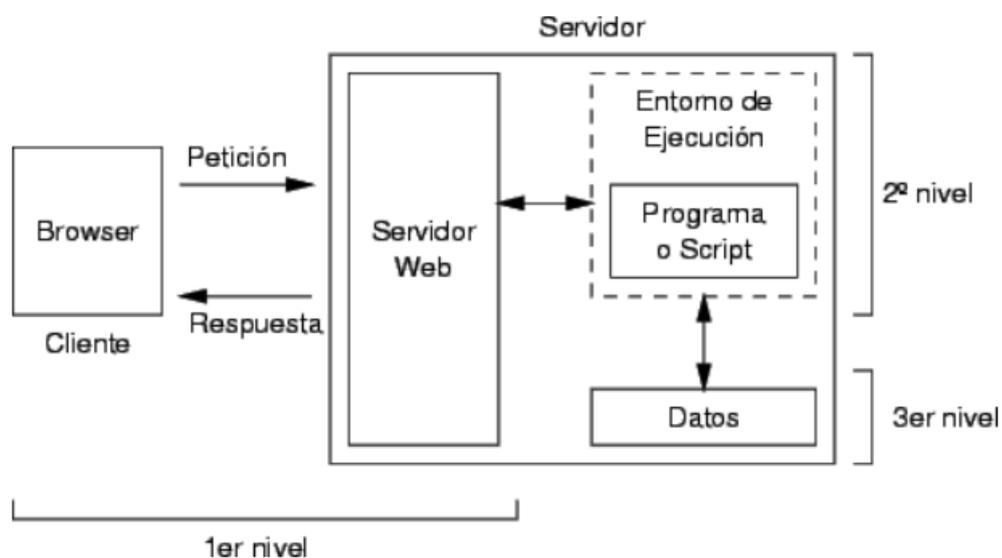


Ilustración 2: Arquitectura web de 3 niveles

2.2 Clases de diseño

En este apartado se detallarán las clases contenidas en cada uno de los paquetes del diagrama del modelo de diseño que se muestra a continuación.

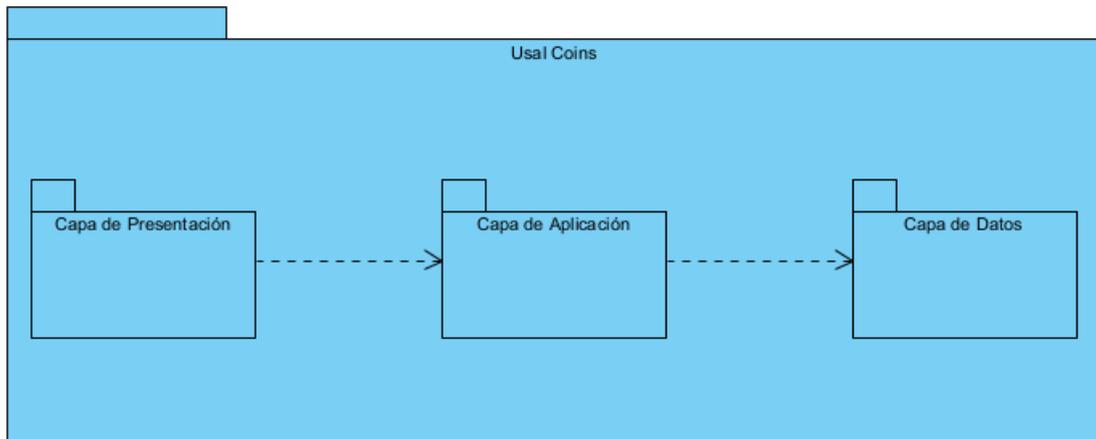


Ilustración 3: Modelo arquitectónico de 3 capas

2.2.1 Capa de Presentación

En la siguiente figura se especifican las clases relacionadas con las vistas de la aplicación web. Estas clases se han agrupado en distintos paquetes con la intención de organizarlas.

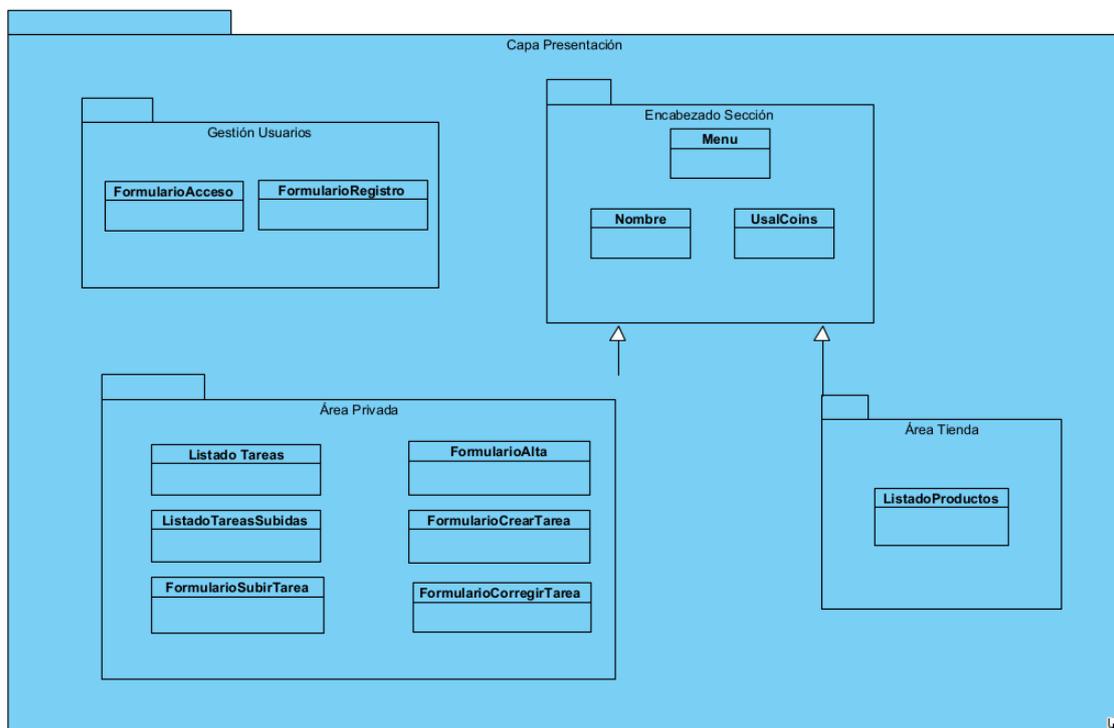


Ilustración 4: Capa de Presentación

2.2.1 Capa de Aplicación

En la siguiente figura se especifica cuales son las funciones que implementan la lógica de la aplicación web. Estas funciones son las encargadas de procesar los datos introducidos por los usuarios en la capa de presentación.

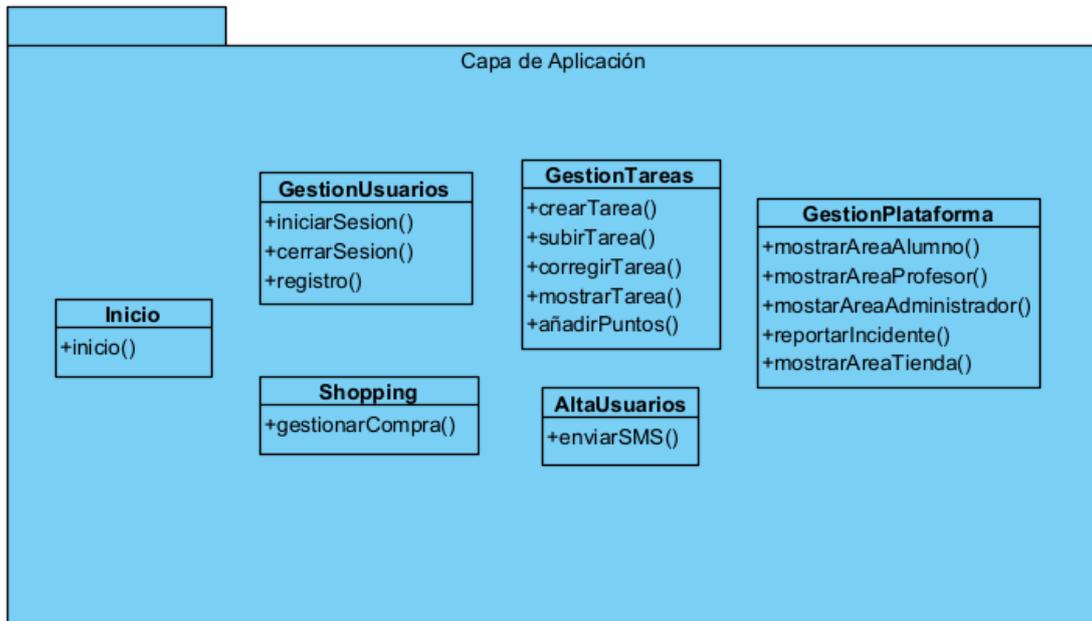


Ilustración 5: Capa de Aplicación

2.2.1 Capa de Datos

Por último, en la siguiente figura se recogen todas las operaciones referentes a la manipulación de la base de datos del sistema.

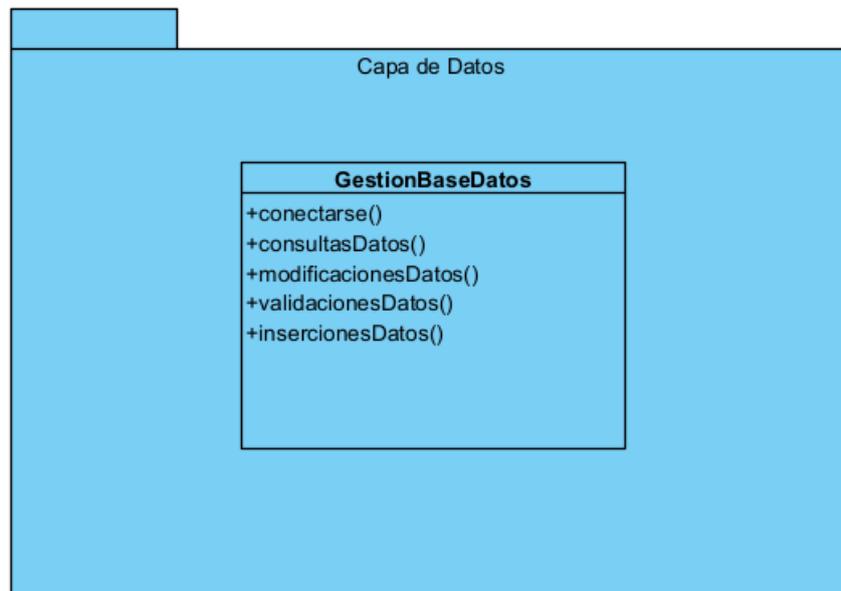


Ilustración 6: Capa de Datos

3. Realización de casos de uso

Con el objetivo de comprender cómo comporta el sistema en términos de interacciones entre sus componentes se van a realizar los diagramas de secuencia correspondientes a los casos de uso predefinidos en el Anexo II.

3.1 Gestión de Usuarios

3.1.1 Caso de Uso Iniciar Sesión

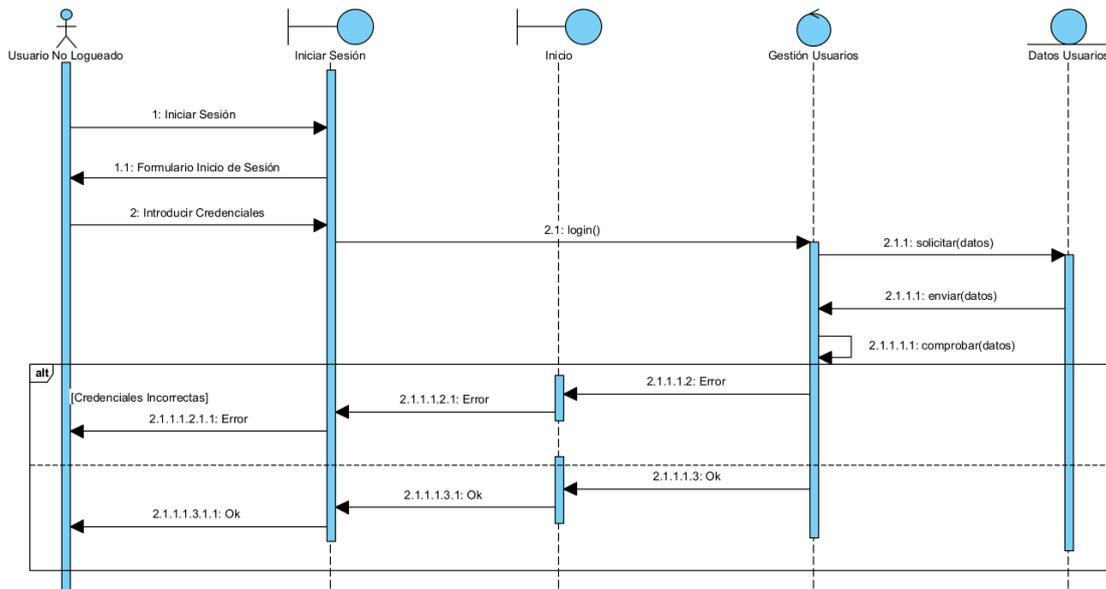


Ilustración 7: Diagrama Secuencia Iniciar Sesión

3.1.2 Caso de Uso Registrarse

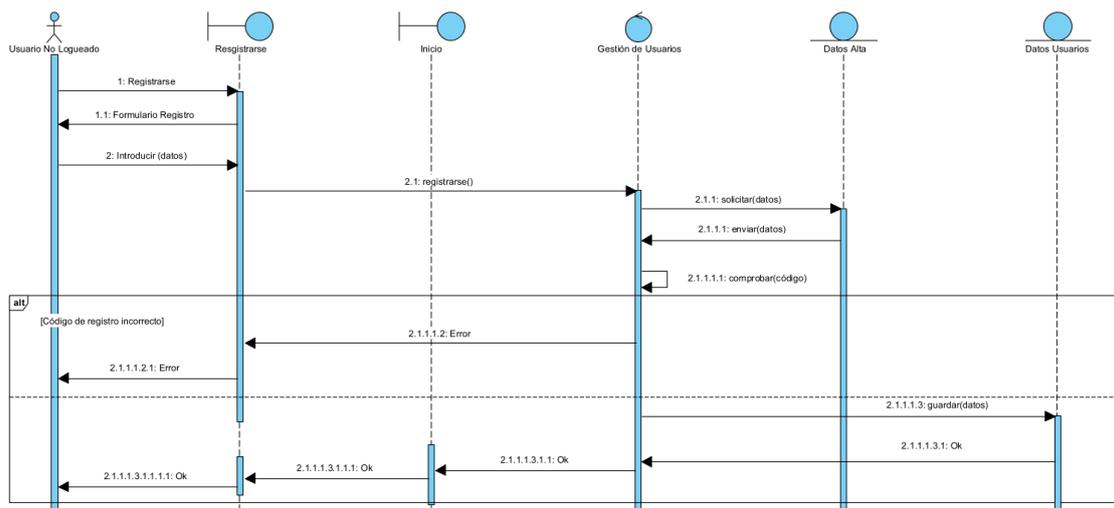


Ilustración 8: Diagrama Secuencia Registrarse

3.1.3 Caso de Uso Cerrar Sesión

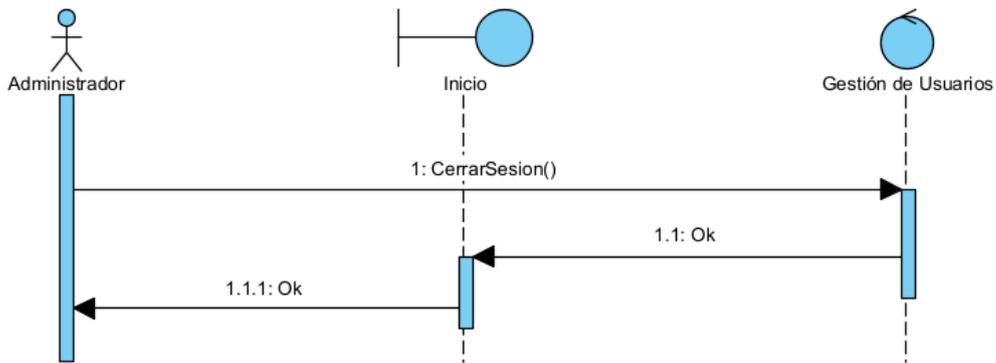


Ilustración 9: Diagrama Secuencia Cerrar Sesión

3.2 Gestión de Plataforma

3.2.1 Caso de Uso Subir Tarea

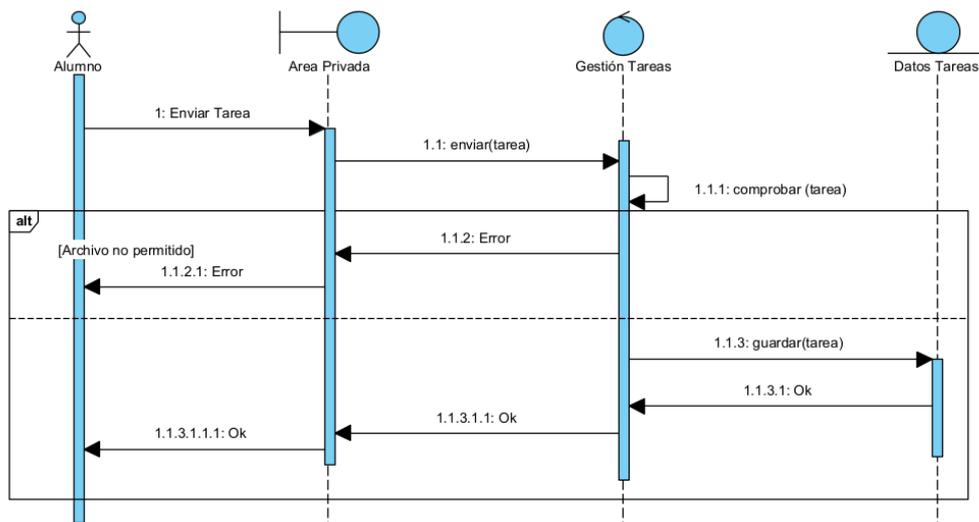


Ilustración 10: Diagrama Secuencia Subir Tarea

3.2.2 Caso de Uso Crear Tarea

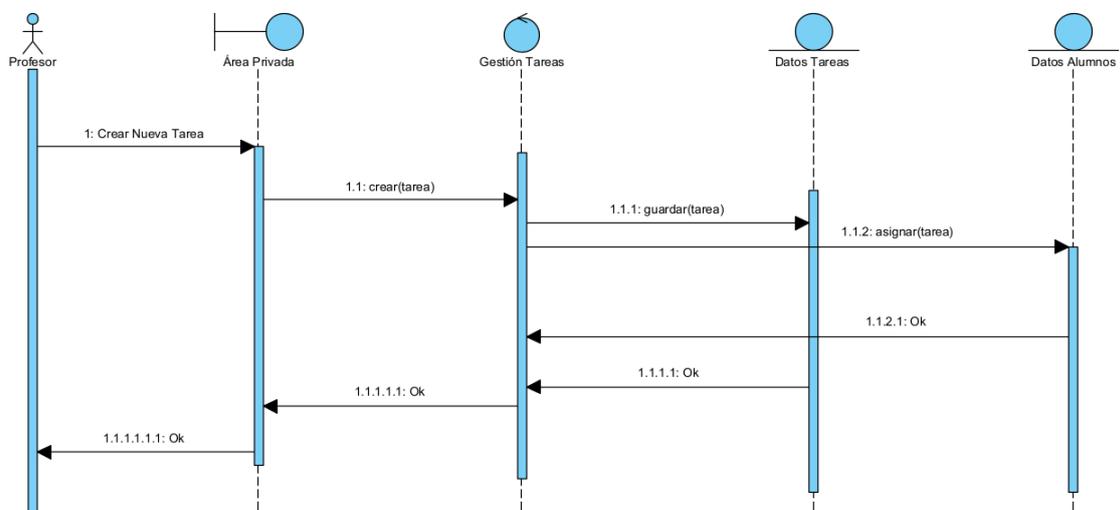


Ilustración 11: Diagrama Secuencia Crear Tarea

3.2.3 Caso de Uso Corregir Tarea

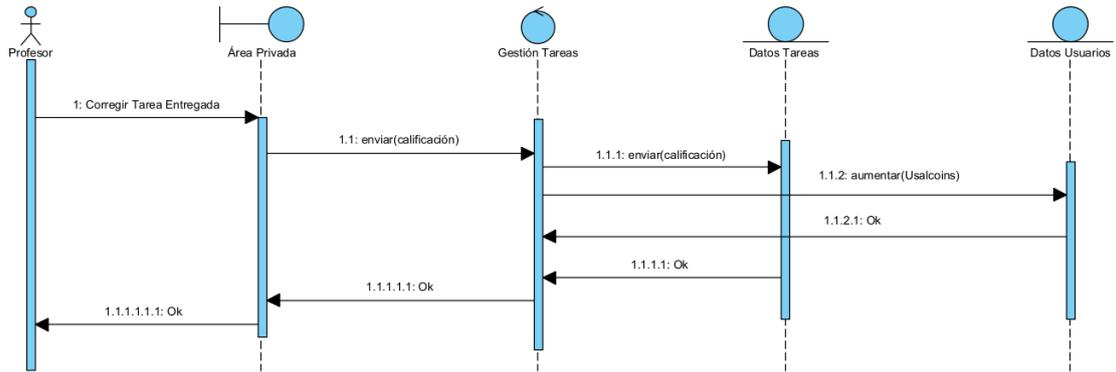


Ilustración 12: Diagrama Secuencia Corregir Tarea

3.2.4 Caso de Uso Mostrar Área Alumno

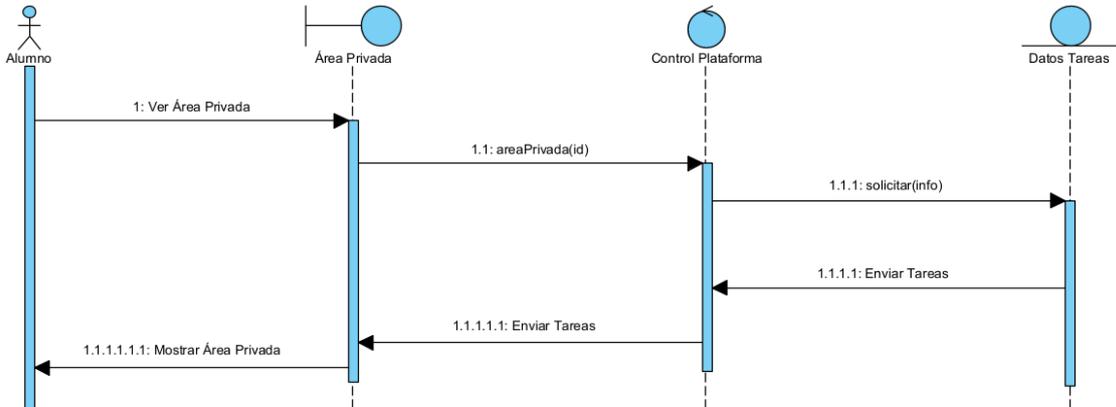


Ilustración 13: Diagrama Secuencia Mostrar Área Alumno

3.2.5 Caso de Uso Mostrar Área Profesor

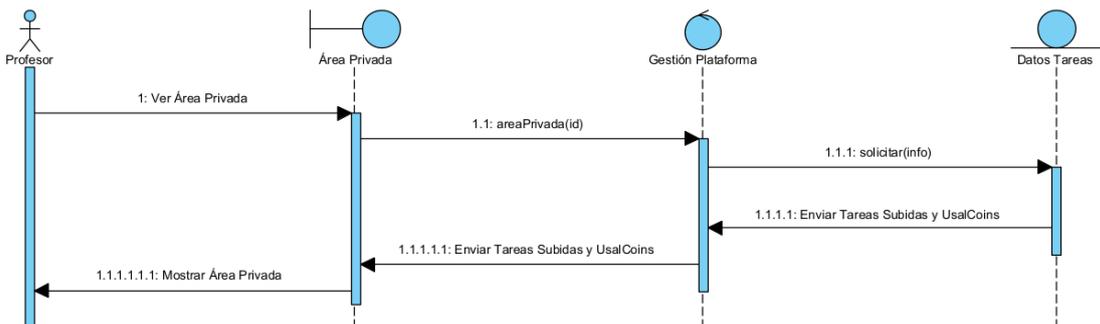


Ilustración 14: Mostrar Área Profesor

3.2.6 Caso de Uso Mostrar Área Administrador

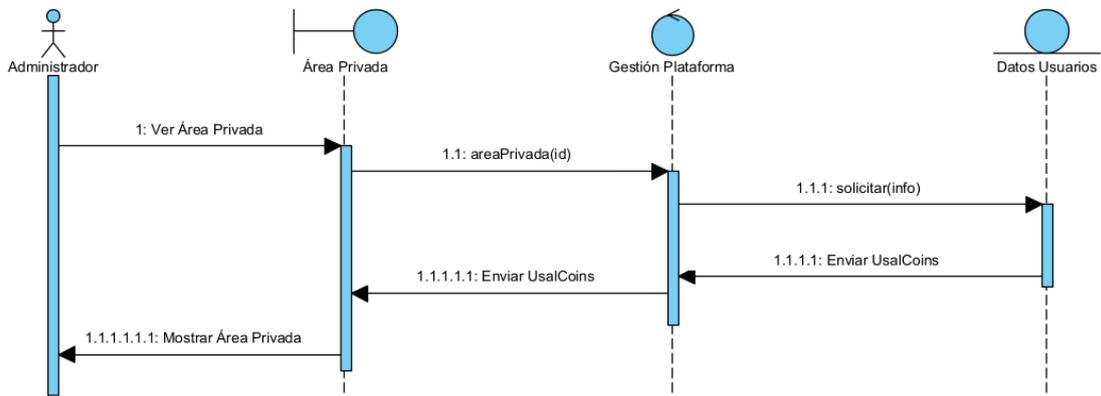


Ilustración 15: Diagrama Secuencia Mostrar Área Administrador

3.2.7 Caso de Uso Enviar Código Registro

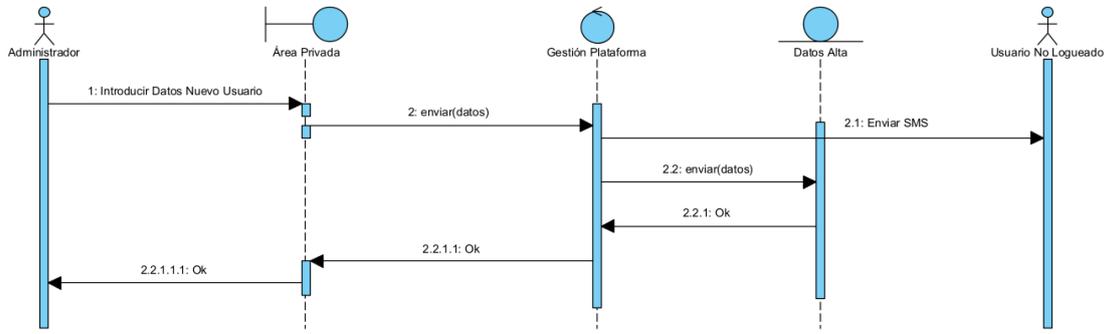


Ilustración 16: Diagrama Secuencia Enviar Código Registro

3.2.8 Caso de Uso Mostrar Área Tienda

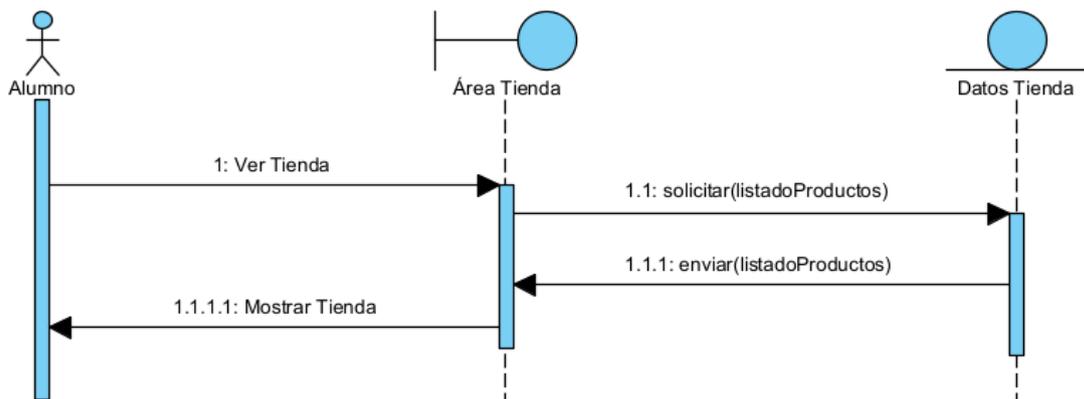


Ilustración 17: Mostrar Área Tienda

3.2.9 Caso de Uso Gestionar Compra

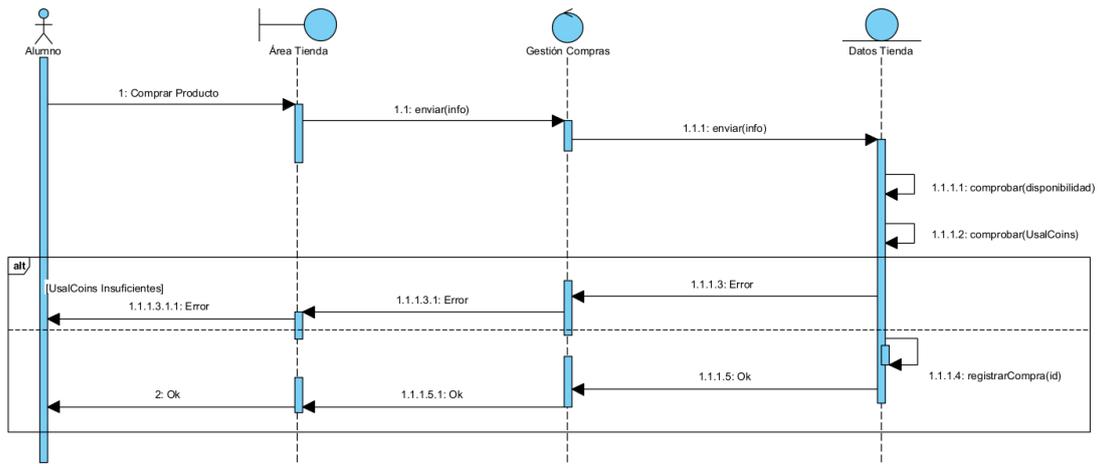


Ilustración 18: Diagrama Secuencia Gestionar Compra

3.3 Gestión de Accesos No Autorizados

3.3.1 Caso de Uso Reportar Acceso No Autorizado

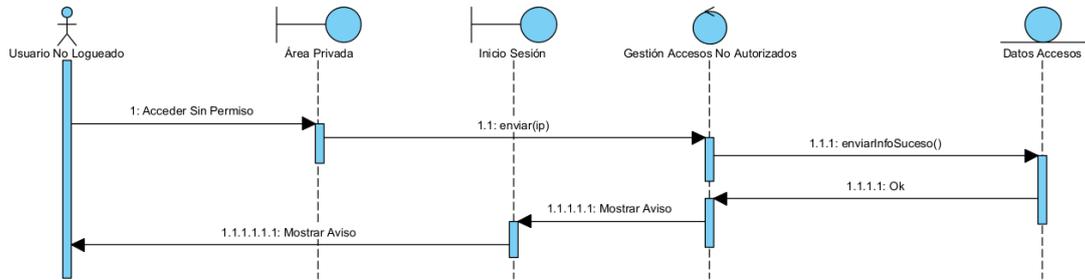


Ilustración 19: Diagrama Secuencia Reportar Acceso No Autorizado

4. Diagrama de despliegue

El diagrama de despliegue proporciona una representación visual de cómo se distribuyen los componentes de software y hardware en un entorno de implementación. Cada nodo en el diagrama de despliegue representa un componente o recurso físico, como servidores, dispositivos de red, bases de datos, etc. A continuación, se expone la justificación de cada nodo:

Cliente: Representa al usuario que accede a la plataforma, en la que dependiendo del rol que tenga asignado podrá interactuar de diferentes maneras con la misma. Para conectarse a la aplicación utiliza su navegador web.

Servidor Aplicación Web: Representa a la plataforma UsalCoins, donde los usuarios que se conectan podrán subir, publicar y corregir tareas además de canjear los puntos obtenidos en la propia tienda de la plataforma.

Base de Datos: Este nodo representa la base de datos utilizada por la aplicación. La creación de un nodo dedicado para la base de datos es importante por varias razones. En primer lugar, separar la base de datos del servidor web ayuda a mejorar el rendimiento y la escalabilidad, ya que los recursos del servidor web no se ven afectados por las operaciones de la base de datos. Además, tener un nodo separado para la base de datos facilita las tareas de administración y mantenimiento, así como la aplicación de medidas de seguridad específicas.

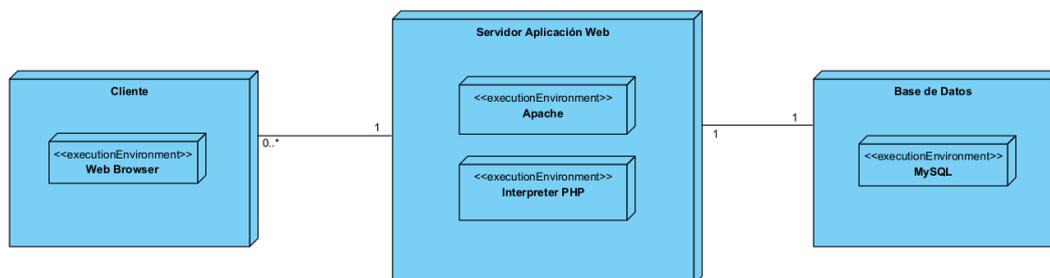


Ilustración 20: Diagrama de despliegue

5. Bibliografía

Jesús Vegas, Universidad de Valladolid, Aplicaciones Multinivel - <https://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node21.html>

Wikipedia, Patrón Cliente-Servidor - <https://es.wikipedia.org/wiki/Cliente-servidor>

Francisco José García Peñalvo, Alicia García Holgado, Transparencias de la Asignatura Ingeniería del Software – Modelo de Domino