



Universidad de Valladolid

Escuela de Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

Mención en Ingeniería de Software

**Desarrollo y despliegue de una
aplicación escalable usando IBM
Cloud (Bluemix) y servicios
Watson**

Autor:

D. Luis Higuero Casado



Universidad de Valladolid

Escuela de Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

Mención en Ingeniería de Software

**Desarrollo y despliegue de una
aplicación escalable usando IBM
Cloud (Bluemix) y servicios
Watson**

Autor:

D. Luis Higuero Casado

Tutor:

D. Benjamín Sahelices Fernández

Resumen

El proyecto tiene como objetivo principal la utilización tanto de servicios Watson de inteligencia artificial como de otros servicios proporcionados por IBM Cloud (Bluemix). Para ello crearemos una aplicación de almacenamiento de imágenes, utilizando de esta manera Watson para el análisis de estas imágenes y así poder implementar de manera sencilla un buscador de imágenes en función del contenido de las mismas.

La aplicación será desplegada como servicio utilizando servicios de aplicaciones disponibles en IBM Cloud y las imágenes se almacenarán en una CDN (red de distribución de contenidos) externa. Por último, el cliente final será móvil con una aplicación para dispositivos Android.

Índice

Resumen	5
1. Introducción y objetivos.....	9
2. Contexto tecnológico.....	10
2.1. Servicios web:	10
2.2. IBM Cloud:	12
2.3. NodeJs:	13
2.4. Android.....	14
2.5. Entorno de desarrollo	15
3. Análisis del proyecto	17
3.1. Introducción.....	17
3.2. Roles	17
3.3. Planificación	18
3.4. Riesgos	22
3.5. Análisis de costes y presupuestos	25
3.6. Descripción de actores.....	27
3.7. Objetivos del sistema	27
3.8. Requisitos funcionales	28
3.9. Requisitos no funcionales.....	30
3.10. Requisitos de información	32
4. Diseño de la aplicación	34
4.1. Diagrama de casos de uso.....	34
4.2. Descripción de los casos de uso	35
4.3. Diagramas de secuencia	48
4.4. Arquitectura.....	61
5. Implementación y pruebas	62
5.1. Introducción.....	62
5.2. Implementación	62
5.3. Contexto de las pruebas	71
5.4. Pruebas realizadas.....	72
6. Conclusiones	81
Bibliografía	82
Apéndices	83
Manual de despliegue	83
Manual de usuario.....	85

1. Introducción y objetivos

Introducción

Hoy en día existe cada vez más una gran integración entre la nube y nuestros dispositivos propios, por lo que la correcta utilización de los servicios que esta ofrece se hace prioritario si queremos sobrevivir en el mercado cada vez más competitivo de hoy en día.

Con el apoyo de los servicios ofrecidos en la nube conseguimos una gran cantidad de servicios especializados que de otra manera sería muy complicados de obtener, a su vez, aportan una gran versatilidad a la hora de adquirir únicamente aquello que necesitas en el momento en que es necesario.

En nuestro caso vamos a aplicar estas técnicas al almacenamiento y análisis de fotos, utilizando servicios como el analizador de imágenes de Watson. Todos estos servicios ofrecen una interfaz sencilla de utilizar y una gran potencia con una complejidad y coste relativamente bajo, facilitando la construcción de aplicaciones más complejas.

Objetivos

El objetivo principal de este Trabajo de Fin de Grado es el desarrollo de una aplicación que permita el almacenamiento y análisis de fotos utilizando distintos servicios web, tanto los proporcionados por la plataforma cloud de IBM como otras de terceros.

Esta aplicación permitirá a los usuarios el almacenamiento y gestión de sus fotos desde su propio dispositivo móvil con acceso a internet junto con el etiquetado de dichas fotos para su posterior búsqueda utilizando el servicio Watson de análisis de imagen.

Se han considerado a su vez los siguientes objetivos de manera individual para la consecución del siguiente proyecto:

- Construcción de una aplicación que aproveche los servicios proporcionados por IBM Cloud, especialmente los módulos de inteligencia artificial Watson y que utilice la integración entre distintos módulos de dicha plataforma.
- Familiarización con la tecnología a utilizar, tanto desde el lado del cliente final como desde el lado del servidor.
- Utilización de una metodología de trabajo que nos ayude a planificar los plazos y nos ayude a evitar los posibles riesgos que se pueden dar durante el desarrollo del proyecto
- Análisis y diseño de la aplicación a construir.
- Construcción tanto de la aplicación que tenemos como objetivo desarrollar como de toda la documentación perteneciente a dicha aplicación a desarrollar.

2. Contexto tecnológico

Para la consecución de este proyecto se han utilizado distintas tecnologías trabajando unidas, a continuación procedemos a la explicación de cada una de ellas de manera individualizada:

2.1. Servicios web:

Los servicios web son una tecnología que define una serie de protocolos y estándares que permiten el intercambio de información entre aplicaciones. De esta forma aplicaciones muy diversas, desarrolladas con tecnologías muy diferentes pueden interactuar y cooperar de manera mucho más sencilla, todo esto hace que las aplicaciones finales se vuelvan mucho más sencillas a la hora de construirse y nos permite que la funcionalidad final sea mucho mayor, ya que requiere un menor trabajo su implementación

Toda esta arquitectura se sustenta sobre esta serie de protocolos y estándares, estos están definidos por organizaciones como OASIS Y W3C, estas organizaciones también se encargan de definir muchos de los estándares que se utilizan en la web a día de hoy.

Existen distintos estándares tales como XML, SOAP, REST, etc., todos estos aportan distintas maneras de enfocar la utilización de dichos servicios con una serie de particularidades. A continuación explicaremos con mayor detenimiento los más utilizados, que son SOAP y REST.

SOAP

SOAP es un protocolo para la comunicación de procesos de manera estándar, deriva de un protocolo creado por Dave Winer en 1998, llamado XML-RPC y fue creado finalmente por un conjunto de empresas entre las que destaca IBM y Microsoft y utiliza el formato XML para el intercambio de información. La principal característica diferenciadora con el resto de protocolos es la definición de las operaciones como puertos WSDL, por lo tanto nos permite definir de manera muy concreta todos los parámetros de la comunicación, desde las funciones de la interfaz hasta los tipos de los datos intercambiados.

REST

REST es otro de los protocolos de comunicación estándar y sea el utilizado en nuestra aplicación. REST surgió a principios del 2000 a raíz de una tesis doctoral y utiliza el protocolo HTTP para proporcionar una API que utiliza cada uno de sus métodos. Rest permite la utilización tanto de JSON como de XML para el intercambio de información, al contrario que el protocolo anterior que solo permitía XML. Todas las operaciones que se implementen como servicio rest se solicitarán mediante las operaciones estándar GET,PUT,POST y DELETE lo que hace que no se requiera de una implementación especial a la hora de consumir dichos servicios.

Para acceder a la funcionalidad rest de otro dispositivo se utilizan las cadenas url, a continuación aparece un ejemplo de una cadena estándar:

Dirección del host: puerto/identificación de la operación/argumentos

A su vez, algunas operaciones requerirán también argumentos en el cuerpo de la petición HTTP, por lo que será necesaria su inclusión para que la operación pueda llevarse a cabo.

Por último, las operaciones REST devuelven una respuesta, estas respuestas siguen el código de respuestas HTTP para indicar el estado de la respuesta y a su vez pueden incluir otra serie de argumentos como respuesta de las operaciones, que será necesario extraer de la respuesta.

La utilización de servicios web se está extendiendo de manera progresiva, presentan diversas ventajas respecto a otras maneras de comunicación: utilizan el puerto 80 en lugar de otra serie de puertos, lo que hace que no puedan ser bloqueados por firewalls, aportan una gran independencia entre el servicio y la aplicación, nos permite pagar únicamente por aquello que utilizamos, etc., todas estas razones hacen que su uso aumente en detrimento de otra serie de fórmulas más convencionales.

En nuestro caso utilizamos como plataforma que nos proporciona servicios dos, IBM Cloud nos proporciona IBM Watson, un punto de despliegue del servidor rest y la base de datos y Cloudinary almacena las fotos de la aplicación.

Cloudinary



Cloudinary es el servicio web elegido para el almacenamiento de las fotos en la plataforma, nos permite el almacenamiento escalable de una gran cantidad de fotos con una interfaz muy sencilla.

Funciona como una CDN(Content Delivery Network o red de distribución de contenido), es decir, almacena múltiples copias de las mismas en distintos dispositivos lo que permite que la distribución sea mucho más rápida. A su vez, la utilización de este servicio web es muy sencillo ya que dispone de múltiples Apis nativas para distintas plataformas, incluyendo entre ellas NodeJs y Android, que son las que vamos a utilizar para el desarrollo de esta aplicación.

2.2. IBM Cloud:



IBM Cloud nos proporciona algo que se va haciendo común entre las principales distribuidoras de servicios web, toda una serie de servicios accesibles y combinables en una única plataforma. Está basado en vitalización y permite a los desarrolladores el acceso a los numerosos servicios que posee sea muy sencilla. Los distintos servicios que podemos adquirir son de múltiples categorías y van desde despliegue de instancias, monitorización, etc. hasta servicios de inteligencia artificial y blockchain, todos ellos combinables de manera muy sencilla y nos ofrecen distintas posibilidades para su desarrollo tanto online como en nuestro propio dispositivo.

A continuación hablaremos de los distintos servicios que hemos utilizado para nuestra aplicación:

Watson

El primero de ellos es IBM Watson, el cual es un servicio web que posee varias variantes todas ellas centradas en inteligencia artificial, para ello utiliza técnicas de deep learning junto con redes neurales. Nos ofrece múltiples servicios basados en inteligencia artificial tales como análisis de voz, de texto, imágenes, video, metadatos, etc., la lista es extensa y poco a poco se siguen incluyendo nuevos servicios basados en Watson.

En nuestro caso hemos seleccionado el servicio de análisis de imágenes el cual permite tanto analizar fotos de manera genérica, la utilidad que nosotros vamos a utilizar, como la creación de clasificadores y el entrenamiento de ellos mediante el paso de fotos y la posterior utilización de este clasificador para un análisis más específico de las fotos.

La primera de las utilidades es la que nos permite la creación de clasificadores, esta utilidad nos permite coger una serie de fotos y agruparlas en torno a buenas y malas, para así crear una herramienta llamada clasificador, la cual podrá ser utilizada posteriormente para analizar fotos. Este clasificador es más preciso cuantas más fotos se le pasen y se recomienda que, como mínimo, se le proporcione 1000 fotos buenas y 200 malas, una vez hecho esto podemos empezar a utilizarlo y el clasificador irá ampliando su conocimiento del problema a medida que va analizando más fotos, volviéndose de esta manera más preciso.

La segunda utilidad de Watson con el analizador de imágenes es el análisis de imágenes en sí, el cual se puede realizar con el clasificador creado en el apartado anterior o con el clasificador por defecto dado por ellos, el cuál será el que utilizemos nosotros y nos permite un análisis de fotos que pueden ser de cualquier tipo, sin adaptarse a un único problema. Para ello hay dos posibilidades, la primera es subir las fotos a la aplicación de manera directa para el análisis y la segunda es con la url, utilizaremos la segunda ya que requiere menos recursos. Por último, se nos devolverá una serie de resultados, los cuales vienen formateados como un array en un json y nos devolverán los atributos, la probabilidad de acierto y la jerarquía del atributo si la hubiere.

Despliegue de aplicaciones NodeJs

La segunda funcionalidad utilizada ha sido el despliegue de una aplicación NodeJs en una instancia web. Esto nos permite el desarrollo de la misma en local y una vez concluida desplegarla en dicha instancia, mediante una interfaz definida por ellos muy sencilla de utilizar. Una vez concluida y colocados todos los parámetros requeridos para su despliegue en la instancia web se procede a ello, una vez desplegada nos permite conservar una dirección de host de manera constante sin tener que hacernos cargo de nada, de una manera muy sencilla.

Compose for MySQL

Por último, la base de datos que hemos utilizado es un servicio proporcionado por IBM Cloud llamado Compose for MySQL, el cual nos ofrece un host con una base de datos relacional MySQL con acceso total a toda la funcionalidad de ella como si la tuviésemos en el mismo dispositivo, sin tener que hacernos cargo de nada.

2.3. NodeJs:



NodeJs es un entorno de ejecución para Java Script construido a partir del motor Java Script V8 de Chrome. Tiene entrada/salida asíncrona y una arquitectura orientada a eventos con el objetivo de poder crear programas altamente escalables.

NodeJs, a pesar de estar hecho a partir de Java Script, no se ejecuta en el cliente final, sino que se ejecuta en el servidor, como podemos ver en la especificado en la figura 1. Fue creado en el año 2009 y ha ido evolucionando desde entonces

La principal fortaleza de NodeJs es npm. Npm es la biblioteca de módulos de NodeJs y es la mayor biblioteca de módulos de código abierto del planeta. Npm permite al usuario la búsqueda, descarga, gestión de dependencias, etc. de sus módulos de manera muy sencilla, a su vez ofrece un entorno de ejecución para java script.

Esto hace que la adquisición de módulos y la combinación de ellos sea muy sencillo para el usuario, a su vez podemos crear nuestros propios módulos y subirlos a la plataforma sin ningún inconveniente.

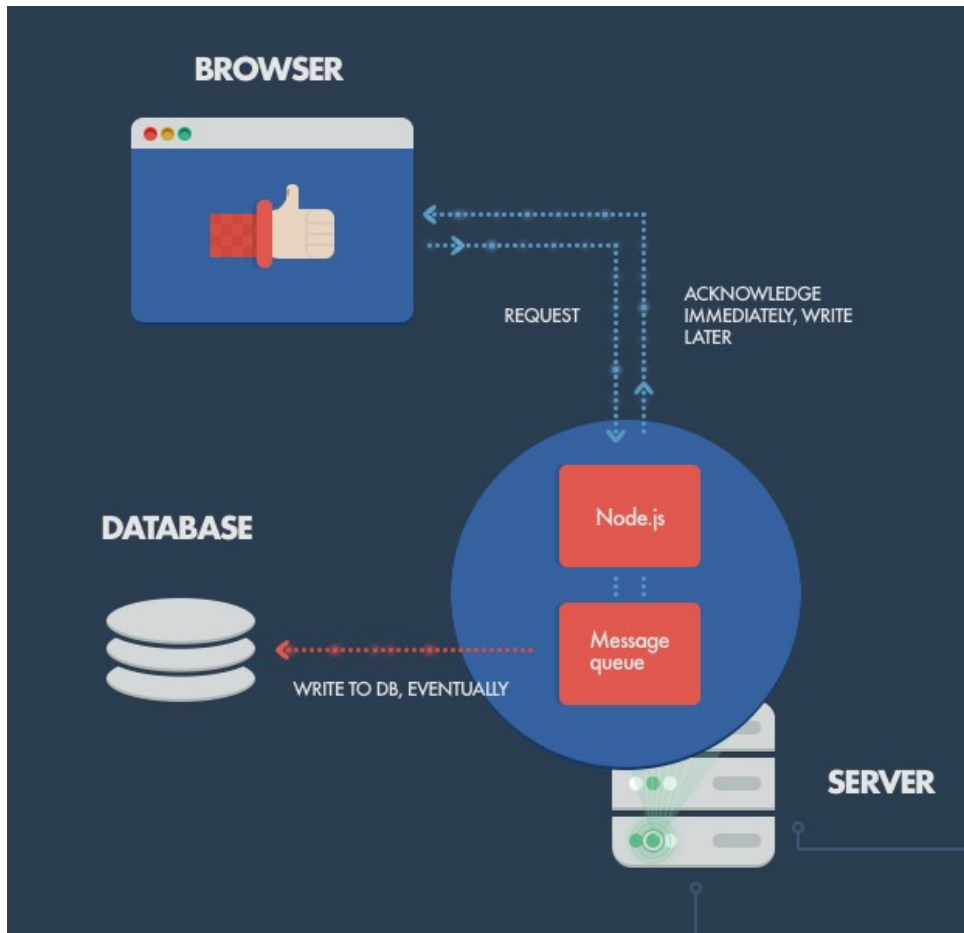


Figura 1: Arquitectura de una aplicación con NodeJs como servidor.

2.4. Android



Android es un sistema operativo basado en el núcleo Linux. Este sistema operativo fue diseñado principalmente para dispositivos móviles tales como teléfonos inteligentes, tabletas, relojes, etc. Actualmente es el sistema operativo más utilizado en dispositivos móviles con cuotas de mercado superiores al 80 % en el año 2017 por lo que se puede afirmar que es el sistema operativo móvil más utilizado del planeta.

Este sistema operativo fue desarrollado por Android Inc. y posteriormente comprado por Google, en el año 2008 fue puesto a la venta el primer dispositivo con Android como sistema operativo. Una de sus principales ventajas es que un software de código abierto, lo cual le ha permitido tener una clara ventaja respecto a sus competidores y convertirse en el más utilizado.

Una de las principales desventajas de Android es su fragmentación en versiones. La primera versión android fue Android A, actualmente la última versión es android O, la actualización entre versiones no es obligatoria y depende del fabricante, por lo que entre los dispositivos android hay una gran variedad de versiones en uso como podemos ver en la figura 2. Por lo que a la hora de realizar aplicaciones hay que tener en cuenta para que versiones se va a hacer, aumentando la complejidad si lo que queremos es que sea compatible entre múltiples versiones.

Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	0.3%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.4%
4.1.x	Jelly Bean	16	1.5%
4.2.x		17	2.2%
4.3		18	0.6%
4.4	KitKat	19	10.3%
5.0	Lollipop	21	4.8%
5.1		22	17.6%
6.0	Marshmallow	23	25.5%
7.0	Nougat	24	22.9%
7.1		25	8.2%
8.0	Oreo	26	4.9%
8.1		27	0.8%

Figura 2: Porcentaje de usuarios por versión android instalada a día 08/01/2018

Otro de los problemas de android respecto a sus competidores es las vistas que se han de mostrar en dispositivos con tamaños de pantalla muy diferentes, lo cual hace que no se puedan definir unas proporciones genéricas para todas las pantallas. Esto hace que haya que definir las dimensiones en para múltiples pantallas e incluso redefinir la interfaz entera, en android se ha resuelto este problema dividiendo las pantallas en tipos, para que así aunque no quede ajustado de manera exacta para cada dispositivo el resultado final no es tan malo como las medidas genéricas.

2.5. Entorno de desarrollo

En este caso, toda la creación del código ha sido realizada con un IDE, pero cada uno de los dos partes ha sido construida con un IDE distinto, el código Android ha sido escrito utilizando la Android Studio y el código Node se ha escrito utilizando Visual Studio Code. El diseño de la aplicación se ha realizado con la herramienta Astah y la documentación ha sido escrita con Microsoft Office Word.

A continuación hablaremos brevemente de cada uno de ellos:

- **Android Studio:** Android studio es un IDE desarrollado por Google para el desarrollo de aplicaciones Android. Este IDE permite al usuario escribir tanto código Java como

Kotlin para el desarrollo de aplicaciones y fue creado para sustituir a Eclipse como IDE de desarrollo Android

- Visual Studio Code: Visual Studio es el IDE de Microsoft para el desarrollo de aplicaciones en múltiples lenguajes. Ofrece múltiples módulos de compatibilidad para distintos lenguajes y en nuestro caso, ofrece compatibilidad plena con NodeJs y npm.
- Astah: Astah es una herramienta que nos permite la creación de múltiples diagramas en UML. Esta herramienta ha sido utilizada a lo largo de la carrera y es por eso por lo que, a pesar de haber herramientas mejores a nuestra disposición, hemos optado por la utilización de esta herramienta
- Microsoft Office Word: Microsoft Office Word es un popular editor de texto propiedad de Microsoft, posee múltiples funcionalidades y cumple con todos los requisitos a la hora de escribir un TFG.

3. Análisis del proyecto

3.1. Introducción

En este capítulo se tratará los apartados relacionados con la planificación, asignación de recursos, y seguimiento de las fases del proyecto.

Para la elaboración del siguiente proyecto se ha recurrido al Proceso unificado como proceso de desarrollo de software, este proceso está dividido en cuatro fases (Inicio, elaboración, construcción y transición) que , a su vez, pueden estar divididas en una o varias iteraciones.

3.2. Roles

En este proyecto el único integrante ha sido Luis Higuero Casado, el cual ha ejercido todos los roles y responsabilidades del proyecto. Sin embargo, cabe destacar que ha contado con el apoyo y experiencia de su tutor Don Benjamín Sahelices Fernández durante la elaboración del mismo.

A continuación, en la tabla 1 procedemos a explicar de manera breve los distintos roles del proyecto, junto con las personas encargadas de ellos a lo largo del proyecto.

Rol	Responsabilidades	Persona encargada
<i>Gestor del proyecto</i>	-Planificación y organización del proyecto. -Elaboración del documento explicativo del proyecto	Luis Higuero Casado
<i>Analista</i>	-Análisis del problema a resolver -Elaboración de los requisitos y funciones a implementar	Luis Higuero Casado
<i>Diseñador</i>	-Elaboración de la solución a implementar	Luis Higuero Casado
<i>Programador</i>	-Implementación de la solución	Luis Higuero Casado
<i>Testeo</i>	-Prueba del resultado final y corrección de los fallos detectados	Luis Higuero Casado

Tabla 1: Distintos roles en el desarrollo de la aplicación

3.3. Planificación

En este proyecto se ha utilizado el proceso unificado como método de planificación del desarrollo, las distintas fases en las que se ha dividido el desarrollo están explicadas a continuación. Dado que la metodología utilizada es el proceso unificado, el final de cada fase viene marcado por un hito, en el cual se libera una de las partes del proyecto junto con su documentación y prototipos.

Dado el análisis previo se ha estimado el coste del proyecto en 300 horas de trabajo, repartiéndose a lo largo de 15 semanas de trabajo a razón de 20 horas de trabajo semanales .

A continuación explicamos los detalles de las iteraciones, en las tabla 2 explicamos la división de las distintas iteraciones, y en las tablas 3, 4, 5, 6 y 7 explicamos las tareas a desarrollar en cada una de ellas de manera individualizada.

Iteración	Descripción	Elementos liberados	Fecha de Inicio	Fecha de finalización
<i>Iteración 1,</i> Fase de Inicio	-Análisis de viabilidad -Planificación inicial del proyecto	Plan de proyecto Documento de requisitos	12/02/2018	25/02/2018
<i>Iteración 2,</i> Fase de Elaboración	-Análisis y diseño de la aplicación a construir	Documento de análisis Documento de diseño de la aplicación	26/02/2018	18/03/2018
<i>Iteración 3,</i> Fase de Construcción	-Implementación del servidor Rest	Código del servidor Rest	19/03/2018	08/04/2018
<i>Iteración 4,</i> Fase de Construcción	-Implementación del cliente Android	Código del cliente Android	09/04/2018	20/05/2018
<i>Iteración 5,</i> Fase de Transición	-Documentación -Testeo -Corrección de bugs	Aplicación completa ya desplegada Manuales de usuario Documentación completa	21/05/2018	03/06/2018

Tabla 2: División de las iteraciones por fechas y tareas a desarrollar

Cada una de estas fases definidas anteriormente tiene una serie de tareas, cada una de ellas con una duración planificada de manera concreta

Iteración 1, Fase de Inicio

Nombre de la tarea	Duración prevista	Inicio	Fin	Duración real
Investigación previa	4 horas	12/02/2018	12/02/2018	4 horas
Organización del proyecto	4 horas	13/02/2018	13/02/2018	5 horas
Creación del plan de proyecto	8 horas	14/02/2018	15/02/2018	6 horas
Identificación de las fases	2 horas	16/02/2018	16/02/2018	2 horas
Identificación de los roles necesarios y costes	1 hora	16/02/2018	16/02/2018	2 hora
Identificación de requisitos	7 horas	16/02/2018	20/02/2018	8 horas
Identificación de los casos de uso	2 horas	20/02/2018	20/02/2018	3 horas
Identificación de los riesgos asociados	8 horas	21/02/2018	22/02/2018	7 horas
Planificación de la siguiente fase	4 horas	23/02/2018	25/02/2018	5 horas
TOTAL	40 horas	12/02/2018	25/02/2018	42 horas

Tabla 3: Tareas a desarrollar en la iteración 1

Como podemos ver en la tabla 3, la planificación en la iteración 1 ha sido ligeramente más optimista que la realidad, provocando una divergencia de 2 h entre la duración total esperada de la fase y la duración real.

Iteración 2, Fase de elaboración

Nombre de la tarea	Duración prevista	Inicio	Fin	Duración real
Revisión de la documentación previa	4 horas	26/02/2018	26/02/2018	3 horas
Definición de requisitos	14 horas	27/02/2018	02/03/2018	12 horas
Definición de los casos de uso	14 horas	02/03/2018	07/03/2018	16 horas
Elaboración de diagramas asociados a los casos de uso	16 horas	08/03/2018	13/03/2018	16 horas
Diseño de la arquitectura a implementar	8 horas	14/03/2018	15/03/2018	8 horas
Planificación de la siguiente fase	4 horas	16/03/2018	18/03/2018	4 horas
TOTAL	60 horas	26/02/2018	18/03/2018	59 horas

Tabla 4: Tareas a desarrollar en la iteración 2

Como podemos ver en la tabla 4, en la iteración 2 la planificación ha sido mínimamente pesimista, provocando una divergencia de 1 h entre la duración total y la esperada.

Iteración 3, Fase de construcción

Nombre de la tarea	Duración prevista	Inicio	Fin	Duración real
Revisión de la documentación previa	4 horas	19/03/2018	19/03/2018	3 horas
Diseño de las operaciones rest a implementar	4 horas	20/03/2018	20/03/2018	3 horas
Implementación del código	28 horas	21/03/2018	29/03/2018	30 horas
Planificación de la siguiente fase	4 horas	30/03/2018	08/04/2018	4 horas
TOTAL	40 horas	19/03/2018	08/04/2018	40 horas

Tabla 5: Tareas a desarrollar en la iteración 3

Como podemos ver en la tabla 5, en la iteración 3 la duración planificada y la final son las misma, a pesar de las discrepancias individuales se ha conseguido que el tiempo total concuerde con la cantidad planificada para la iteración.

Iteración 4, Fase de construcción

Nombre de la tarea	Duración prevista	Inicio	Fin	Duración real
Revisión de la documentación previa	4 horas	09/04/2018	09/04/2018	4 horas
Diseño de las vistas a implementa	10 horas	10/04/2018	12/04/2018	20 horas
Implementación del código	102 horas	12/04/2018	17/05/2018	100 horas
Planificación de la siguiente fase	4 horas	18/05/2018	20/05/2018	4 horas
TOTAL	120 horas	09/04/2018	20/05/2018	128 horas

Tabla 6: Tareas a desarrollar en la iteración 4

Como podemos ver en la tabla 6, en la iteración 4 se ha producido una clara desviación entre la duración planificada y la real, debido a un claro sobrecoste en la fase de diseño de las vistas. Esto es debido a que se detectaron fallos en un primer diseño cuya corrección era complicada, por lo que se optó por cambiar la totalidad del diseño de la interfaz para corregir este problema.

Iteración 5, Fase de transición

Nombre de la tarea	Duración prevista	Inicio	Fin	Duración real
Testeo de la aplicación	14 horas	21/05/2018	24/05/2018	10 horas
Corrección de problemas detectados	6 horas	24/05/2018	25/05/2018	4 horas
Despliegue final de la aplicación	4 horas	26/05/2018	28/05/2018	4 horas
Conclusión de la memoria	10 horas	29/05/2018	31/05/2018	12 horas
Elaboración del manual de usuario	6 horas	31/05/2018	03/06/2018	6 horas
TOTAL	40 horas	21/05/2018	03/06/2018	38 horas

Tabla 7: Tareas a desarrollar en la iteración 5

Como podemos ver en la tabla 7, en la iteración 5 la duración planificada y la final son muy similares, a pesar de las discrepancias individuales se ha conseguido que el tiempo total varié muy poco entre lo planificado y lo real.

3.4. Riesgos

Formato de los riesgos

Los riesgos expuestos a continuación presentan el siguiente formato:

- **Probabilidad de ocurrencia:** Este campo se divide en cuatro rangos de posibilidad de ocurrencia que van desde 1 hasta 4, siendo el 1 un riesgo muy poco probable y el 4 un riesgo casi seguro.
- **Impacto de la ocurrencia:** Este campo, al igual que el campo anterior, va a ser dividido en cuatro rangos que van desde 1 hasta 4, siendo el 1 un riesgo muy poco peligroso para el proyecto y el 4 un riesgo crítico.
- **Protección contra el riesgo:** En este campo aparecerán las medidas que se han de tomar para minimizar la probabilidad de que dicho riesgo tenga lugar
- **Plan de contingencia:** En este campo aparecerán las medidas que se han de tomar una vez dicho riesgo se haya manifestado

Riesgos

Riesgo 1	Borrado de datos
Impacto	Dependiente del tipo de datos perdidos, pudiendo ir desde 1 si son datos fácilmente restaurarles hasta 4 si son críticos
Probabilidad	2
Protección contra el riesgo	Utilización de un sistema de control de versiones
Plan de contingencia	Restauración de una versión anterior si es posible, si no reelaboración del contenido perdido

Figura 3: Riesgo de borrado de datos

Riesgo 2	Maquina averiada
Impacto	Dado que este proyecto requiere de distintos tipos de máquinas para su consecución, el impacto será dependiente de la maquina averiada y de la gravedad de su avería
Probabilidad	1
Protección contra el riesgo	Aplicación de un sistema de mantenimiento adecuado a las máquinas que forman parte del desarrollo del proyecto
Plan de contingencia	<p>Dependerá del impacto del riesgo:</p> <ul style="list-style-type: none"> • Impacto 2 (Avería leve): Reparación de la máquina • Impacto 3 (Avería no recuperable): Rescatado de los datos almacenados y sustitución de la maquina por una nueva • Impacto 4 (Avería crítica): Recuperación de una versión anterior en una nueva máquina y continuación del proyecto en esta

Figura 4: Riesgo de maquina averiada

Riesgo 3	Enfermedad
Impacto	3
Probabilidad	1
Protección contra el riesgo	En este caso no existe plan de protección contra este riesgo.
Plan de contingencia	Replanificación del proyecto en la medida de lo posible y, si dicha replanificación no es posible, aplazamiento de la fecha de entrega del proyecto.

Figura 5: Riesgo de enfermedad

Riesgo 4	Planificación demasiado optimista
Impacto	Dependerá de la diferencia entre la planificación y la realidad, siendo de 1 cuando varia en horas hasta 3 cuando varia en semanas
Probabilidad	1
Protección contra el riesgo	Realización de una planificación pesimista para que dicha planificación solo pueda ser peor que la realidad
Plan de contingencia	Replanificación del proyecto en la medida de lo posible y, si dicha replanificación no es posible, aplazamiento de la fecha de entrega del proyecto.

Figura 6: Riesgo de borrado de planificación demasiado optimista

Riesgo 5	Fallos en la etapa de análisis
Impacto	4
Probabilidad	2
Protección contra el riesgo	Revisión del trabajo realizado en esta fase a la hora de su conclusión y prestar especial cuidado y atención a la hora de su elaboración
Plan de contingencia	Replanificación del proyecto en la medida de lo posible y ,si dicha replanificación no es posible, aplazamiento de la fecha de entrega del proyecto, evaluación de las distintas partes elaboradas hasta el momento para su reutilización.

Figura 7: Riesgo de borrado de fallos en la etapa de análisis

Riesgo 6	Fallos en la etapa de diseño
Impacto	Desde 1 hasta 3 en función del tamaño de los módulos afectados y las dependencias con otros
Probabilidad	1
Protección contra el riesgo	Utilización de un desarrollo progresivo, de mas general a más específico, para que en caso de detección de un fallo mas general el tiempo de reparación pueda ser mayor
Plan de contingencia	Replanificación del proyecto en la medida de lo posible, si dicha replanificación no es posible aplazamiento de la fecha de entrega del proyecto.

Figura 8: Riesgo de borrado de fallos en la etapa de diseño

Riesgo 7	Fallos en la etapa de implementación
Impacto	Dependerá del tiempo restante y el tamaño de las funcionalidades afectadas, yendo desde 1 hasta 2
Probabilidad	2
Protección contra el riesgo	Utilización de un desarrollo progresivo, de mas general a más específico, para que en caso de detección de un fallo mas general el tiempo de reparación pueda ser mayor
Plan de contingencia	Replanificación del proyecto en la medida de lo posible, si dicha replanificación no es posible aplazamiento de la fecha de entrega del proyecto.

Figura 9: Riesgo de fallos en la etapa de implementación

Riesgo 8	Fallos en la etapa de despliegue
Impacto	1
Probabilidad	1
Protección contra el riesgo	Realización de un testeo extenso de la aplicación
Plan de contingencia	Replanificación del proyecto en la medida de lo posible, si dicha replanificación no es posible aplazamiento de la fecha de entrega del proyecto.

Figura 10: Riesgo de fallos en la etapa de despliegue

Riesgo 8	Fallos en el software de terceros
Impacto	2
Probabilidad	2
Protección contra el riesgo	Realización de una evaluación previa de las tecnologías para la detección de estos posibles puntos de fallo y la reparación antes de que ocurran
Plan de contingencia	<p>Ante la ocurrencia de dicho tipo de riesgos, se tratará de solucionarlos siguiendo los siguientes pasos en orden, siendo los primeros menos perjudiciales para el desarrollo que los últimos.</p> <ol style="list-style-type: none"> 1. Reparación del problema 2. Evitar la utilización de dicha característica 3. Sustituir el software utilizado por otro que si que cumpla las características requeridas 4. Desestimación de la inclusión de dicha funcionalidad en la aplicación, con las correspondientes consecuencias

Figura 11: Riesgo de fallos en el software de terceros

3.5. Análisis de costes y presupuestos

Tecnología y equipos

En este caso, la tecnología a utilizar es libre, por lo que no requiere de coste adicional para su utilización, por lo que el coste asociado a esta será nulo en este caso

En cuanto a los equipos, es necesario la utilización de un ordenador y un dispositivo móvil con android instalado. El coste del dispositivo móvil es de 200€ y tiene una vida útil de 2 años, mientras que el del ordenador es de 500€, y tiene una vida útil de 4 años. Por lo que el coste de este proyecto, cuya duración queda fijada en 4 meses, será de 33€ en el dispositivo móvil y de 41€ en el ordenador de trabajo, lo que hace un total de 74€.

Personal

En este caso, dado que el proyecto a desarrollar es un trabajo de fin de grado el coste de personal es nulo, ya que es elaborado por el alumno sin ningún tipo de coste como complemento de su formación, pero dado que este es un supuesto hipotético aproximaremos el coste de personal utilizando como base el sueldo que percibe un programador junior de media que es aproximadamente unos 40€/ hora, por lo tanto, el coste del proyecto el cual se estima en una duración de 300 horas queda fijado en 12000€ de manera aproximada.

Coste de funcionamiento

Esta aplicación tiene un coste de funcionamiento y mantenimiento dado que utiliza servicios de pago de IBM cloud, en concreto hace utilización de 3 servicios.

1. Servicio de despliegue de una aplicación NodeJS: en este caso el coste es por GB-hora, ascendiendo a 0.0526 €/GB-hora
2. Servicio de base de datos MySQL: en este caso depende de la cantidad de datos que se tenga almacenado en el servidor.

GB	Tarifas
1 - 9	€13.54 EUR/GB
10 - 24	€12.18 EUR/GB
25 - 49	€10.84 EUR/GB
50 - 99	€9.48 EUR/GB
100 - 499	€8.12 EUR/GB
500 - 999	€6.77 EUR/GB
1,000 - 4,999	€5.42 EUR/GB
5,000+	€4.06 EUR/GB

Figura 12: Costes de las bases de datos Compose for MySql de IBM Cloud en función del tamaño por cada mes

Como podemos ver en la figura 12, en las primeras fases el volumen de datos correspondería al rango 1, y el coste ascendería a 13,54€/ GB, por lo que el coste variaría en función del plan que tengamos asignado de entre los mostrados y los datos que almacenemos.

3. Servicio de análisis de imágenes Watson: en este caso, el coste viene asociado al número de imágenes que analizamos. Dado que utilizamos el clasificador estándar el coste sería de 0.001504 € por imagen analizada

Dado que también se utiliza una CDN para el almacenamiento físico de las fotos, este coste habrá que incluirlo también, la utilizada actualmente es Cloudinary utilizando el plan gratuito, pero una vez esto escale será necesario llegar a un acuerdo con una CDN para el almacenamiento de dichos datos, con su correspondiente coste

Coste total

En la tabla 8, analizamos los costes de desarrollo totales de la aplicación, los costes de mantenimiento no pueden ser calculados de manera concreta porque dependerán del tráfico que tenga la aplicación y de la cantidad de datos, por lo que es muy complicado hacer una aproximación medianamente coherente con lo que nos espera en la realidad.

Elemento	Coste
Dispositivo móvil	33 €
Ordenador de trabajo	41 €
Personal	12000 €
Total	12074 €

Tabla 8: Costes totales de la aplicación

Dado el carácter de trabajo de fin de grado que tiene este proyecto, el concepto de coste no tiene demasiado sentido, por lo que este apartado solamente es de carácter especulativo y en ningún supone un coste real de la aplicación.

3.6. Descripción de actores

En este caso, solo un actor genérico va a ser el que utilice esta aplicación:

Usuario: Tendrá acceso a toda la aplicación y todos los casos de uso, desde el inicio de sesión y modificación de los datos asociados a un usuario hasta la subida, modificación, búsqueda y descarga de las fotos.

Hemos tomado esta decisión debido a que para la funcionalidad actual implementada no es necesario un usuario administrador, eso habría de ser implementado en el futuro para tareas de mantenimiento y monitorización y ha sido incluido en la lista de mejoras futuras.

3.7. Objetivos del sistema

A continuación se detallarán los principales objetivos de la aplicación a implementar:

1. La aplicación tendrá un sistema de usuarios y sesiones, que permitirá que cada usuario pueda acceder a su contenido desde distintos puntos.
2. La aplicación permitirá la gestión de las fotos de los distintos usuarios.
3. La aplicación permitirá el análisis y etiquetado de las distintas fotos que entren al sistema.

3.8. Requisitos funcionales

Requisito funcional 1	Registro de usuarios
Descripción	El sistema permitirá el registro de nuevos usuarios en el sistema
Caso de uso relacionado	Registrar usuario

Figura 12: Requisito funcional número 1

Requisito funcional 2	Modificación de usuarios
Descripción	El sistema permitirá la modificación de las credenciales de los usuarios
Caso de uso relacionado	Modificar credenciales

Figura 13: Requisito funcional número 2

Requisito funcional 3	Dstrucción de usuarios
Descripción	El sistema permitirá la destrucción de usuarios
Caso de uso relacionado	Destruir usuario

Figura 14: Requisito funcional número 3

Requisito funcional 4	Inicio de sesión
Descripción	El sistema permitirá el inicio de sesión a los usuarios registrados
Caso de uso relacionado	Iniciar sesión

Figura 15: Requisito funcional número 4

Requisito funcional 5	Salida de sesión
Descripción	El sistema permitirá la destrucción de las sesiones
Caso de uso relacionado	Cerrar sesión

Figura 16: Requisito funcional número 5

Requisito funcional 6	Subida de fotos
Descripción	El sistema permitirá la subida de fotos
Caso de uso relacionado	Subir foto de la cámara y subir foto de la galería

Figura 17: Requisito funcional número 6

Requisito funcional 7	Análisis de fotos
Descripción	El sistema permitirá el análisis de fotos
Caso de uso relacionado	Subir foto de la cámara y subir foto de la galería

Figura 18: Requisito funcional número 7

Requisito funcional 8	Búsqueda de fotos
Descripción	El sistema permitirá la búsqueda de fotos y su visualización
Caso de uso relacionado	Buscar foto, ver mis fotos y ver foto

Figura 19: Requisito funcional número 8

Requisito funcional 9	Borrado de fotos
Descripción	El sistema permitirá la destrucción de las fotos
Caso de uso relacionado	Borrar foto

Figura 20: Requisito funcional número 9

Requisito funcional 10	Guardar foto en local
Descripción	El sistema permitirá descargar fotos almacenadas en el servidor
Caso de uso relacionado	Guardar foto en local

Figura 21: Requisito funcional número 10

Requisito funcional 11	Modificar datos de las fotos
Descripción	El sistema permitirá modificar los datos asociados a las distintas fotos
Caso de uso relacionado	Modificar datos

Figura 22: Requisito funcional número 11

3.9. Requisitos no funcionales

Requisito no funcional 1	Sesión utilizando token
Descripción	El sistema garantizará que el mantenimiento de la sesión abierta se realice con una token

Figura 23: Requisito no funcional 1

Requisito no funcional 2	Integridad token
Descripción	El sistema garantizará la existencia de una única token por usuario

Figura 24: Requisito no funcional número 2

Requisito no funcional 3	Numero de sesiones
Descripción	El sistema garantizará que el usuario solo pueda tener una única sesión abierta simultáneamente

Figura 25: Requisito no funcional número 3

Requisito no funcional 4	Construcción de la token
Descripción	El sistema utilizará un algoritmo sha para la creación de las token del sistema

Figura 26: Requisito no funcional número 4

Requisito no funcional 5	Formato de las fotos de la galería
Descripción	Las fotos a subir desde la galería podrán ser en formato jpg o png.

Figura 27: Requisito no funcional número 5

Requisito no funcional 6	Filtros de búsqueda
Descripción	El sistema solo permitirá utilizar como parámetros de filtro parámetros existentes en la base de datos

Figura 28: Requisito no funcional número 6

Requisito no funcional 7	Fotos privadas
Descripción	El sistema permitirá establecer fotos propias como privadas, solo visibles al propietario

Figura 29: Requisito no funcional número 7

Requisito no funcional 8	Fotos públicas
Descripción	El sistema permitirá establecer fotos propias como públicas, visibles para todos

Figura 30: Requisito no funcional número 8

Requisito no funcional 9	Arquitectura del servidor
Descripción	El servidor será desplegado en IBM Cloud como una aplicación

Figura 31: Requisito no funcional número 9

Requisito no funcional 10	Cliente
Descripción	El cliente será una aplicación android

Figura 32: Requisito no funcional número 10

Requisito no funcional 11	Análisis de las fotos
Descripción	El sistema utilizara IBM Watson para el análisis y etiquetado de las fotos

Figura 33: Requisito no funcional número 11

Requisito no funcional 12	Base de datos
Descripción	El sistema utilizara la base de datos MySQL de IBM como almacenamiento de datos del sistema

Figura 34: Requisito no funcional número 12

Requisito no funcional 13	Almacenamiento de las fotos
Descripción	El sistema utilizara una CDN externa a IBM como ubicación de almacenamiento de las fotos

Figura 35: Requisito no funcional número 13

3.10. Requisitos de información

Requisito de información 1	Usuarios
Descripción	El sistema almacenará la información referente a los distintos usuarios
Detalles	<ul style="list-style-type: none"> • Nombre de usuario • Contraseña • Token de sesión

Figura 36: Requisito de información número 1

IRQ 2	Fotos
Descripción	El sistema almacenará información referente a las distintas fotos
Detalles	<ul style="list-style-type: none"> • Path a la foto • Usuario al que pertenece • Pública o privada • Nombre de la foto

Figura 37: Requisito de información número 2

IRQ 3	Parámetros de búsqueda
Descripción	El sistema almacenará información referente a las distintas parámetros de búsqueda que han sido detectados en las fotos
Detalles	<ul style="list-style-type: none"> • Nombre del parámetro

Figura 38: Requisito de información número 3

IRQ 3	Detecciones de parámetros
Descripción	El sistema almacenará información referente a las distintas detecciones de parámetros que se han efectuado en las fotos subidas
Detalles	<ul style="list-style-type: none"> • Nombre del parámetro • Foto en la que aparece • Porcentaje de acierto

Figura 39: Requisito de información número 4

4. Diseño de la aplicación

4.1. Diagrama de casos de uso

En la figura 40 podemos ver el diagrama de casos de uso de la aplicación. Los casos de uso se pueden dividir en dos grupos, los relacionados con la sesión que son los casos de uso número 1, 2, 3, 4 y 5 y los relacionados con las fotos, que son los casos de uso número 6, 7, 8, 9, 10, 11, 12 y 13.

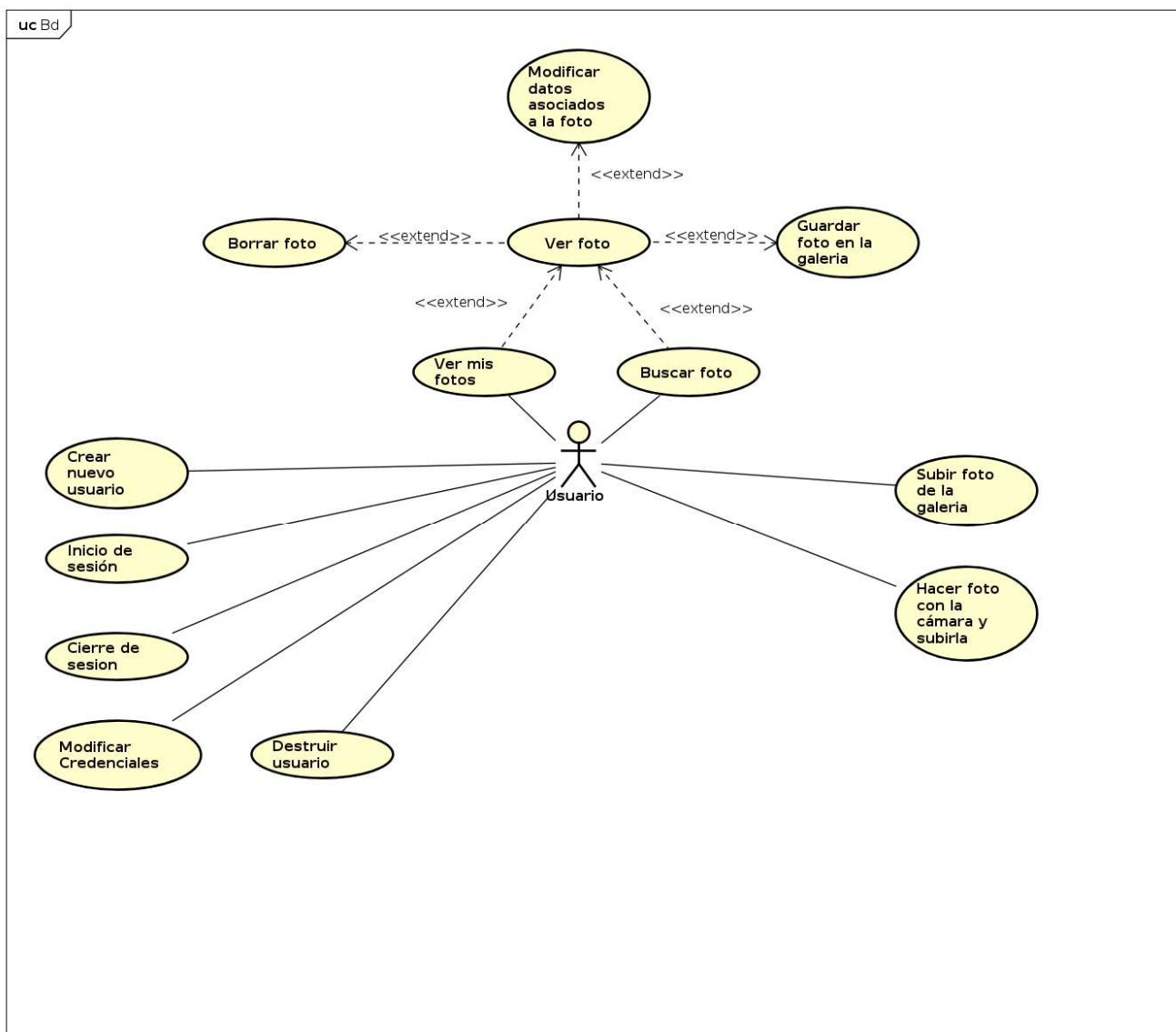


Figura 40: Diagrama de casos de uso de la aplicación

4.2. Descripción de los casos de uso

Caso de uso 1: Registrar usuario

Este caso de uso nos permite añadir nuevos usuarios a la aplicación, el actor aportará los credenciales del nuevo usuario y estos se almacenarán en la base de datos.

Caso de uso 1	Registrar Usuario
Actores	Usuario del sistema.
Descripción	El actor solicita al sistema la creación de un nuevo usuario en el sistema.
Precondiciones	Ninguna.
Flujo normal	<ol style="list-style-type: none">1.-El actor usuario solicita al sistema la creación de un nuevo usuario.2.-El sistema solicita al actor usuario que introduzca los datos necesarios para la creación del nuevo usuario (nombre de usuario y contraseña).3.-El actor usuario introduce los datos requeridos por el sistema.4.-El sistema comprueba que los datos introducidos son válidos, crea un nuevo usuario con dichos valores y el caso de uso concluye.
Postcondiciones	Se ha creado un nuevo usuario en el sistema con los parámetros introducidos.
Flujos alternativos	<ol style="list-style-type: none">3a. El actor decide cancelar el caso de uso en curso. En este caso el caso de uso queda sin efecto.4a. Los datos introducidos no son válidos o colisionan con otros. En este caso el sistema notifica al actor usuario que los valores introducidos no son válidos y el caso de uso continua por el paso número 2.

Figura 41: Caso de uso número 1

Caso de uso 2: Iniciar sesión

Este caso de uso permite al usuario entrar en la aplicación con su cuenta, al ejecutarse se verifican las credenciales suministradas y si dichas credenciales son válidas recibe una token de sesión

Caso de uso 2	Iniciar sesión
Actores	Usuario del sistema.
Descripción	El actor solicita al sistema el inicio de sesión en el sistema con una cuenta.
Precondiciones	No hay una sesión activa en el sistema.
Flujo normal	<ol style="list-style-type: none">1.-El actor usuario solicita al sistema el inicio de una nueva sesión en el sistema.2.-El sistema solicita al actor usuario sus credenciales de acceso al sistema.3.-El actor usuario introduce los datos solicitados por el sistema (nombre de usuario y contraseña).4.-El sistema crea una sesión con los datos proporcionados por el usuario, se la da al usuario y el caso de uso concluye.
Postcondiciones	El usuario ha iniciado sesión en el sistema y ha recibido su clave de sesión.
Flujos alternativos	<ol style="list-style-type: none">3a. El actor decide cancelar el caso de uso en curso. En este caso el caso de uso queda sin efecto.4a. Los datos introducidos no son válidos. En este caso el sistema notifica al actor usuario que los valores introducidos no son válidos y el caso de uso continua por el paso número 2.

Figura 42: Caso de uso número 2

Caso de uso 3: Cerrar sesión

Este caso de uso permite al usuario destruir la clave de sesión actual que posee en el dispositivo y da la posibilidad de crear una nueva cuenta o de iniciar sesión de nuevo.

Caso de uso 3	Cerrar sesión
Actores	Usuario del sistema.
Descripción	El actor solicita al sistema el cierre de la sesión que actualmente tiene abierta.
Precondiciones	El actor usuario tiene una sesión abierta.
Flujo normal	1.-El actor usuario solicita el cierre de la sesión que actualmente tiene iniciada. 2.-El sistema cierra la sesión asociada a ese usuario y destruye la clave de sesión asociada a esa sesión.
Postcondiciones	La sesión asociada al usuario del sistema se ha cerrado.
Flujos alternativos	Ninguno

Figura 43: Caso de uso número 3

Caso de uso 4: Modificar credenciales

Este caso de uso permite al usuario modificar los credenciales de acceso a la cuenta que tiene abierta actualmente, los nuevos credenciales proporcionados si estos son válidos serán los utilizados para acceder a partir de entonces.

Caso de uso 4	Modificar credenciales
Actores	Usuario del sistema.
Descripción	El actor solicita al sistema la modificación de los credenciales de la cuenta que actualmente esta iniciada.
Precondiciones	El actor usuario tiene una sesión abierta.
Flujo normal	1.-El actor usuario solicita la modificación de las credenciales de acceso de la cuenta actual. 2.-El sistema solicita al usuario que introduzca las nuevas credenciales asociadas a la cuenta actual. 3.-El actor usuario introduce las nuevas credenciales asociadas a la cuenta actual. 4.-El sistema comprueba que las credenciales son válidas y no colisionan con otras credenciales, modifica las credenciales actuales y el caso de uso concluye.
Postcondiciones	Las credenciales de la cuenta han sido modificadas.
Flujos alternativos	3a. El actor solicita cancelar el caso de uso en curso. En este caso el caso de uso queda sin efecto. 4a. Los datos introducidos no son válidos o colisionan con otros. En este caso el sistema notifica al actor usuario que los valores introducidos no son válidos y el caso de uso continua por el paso número 2.

Figura 44: Caso de uso número 4

Caso de uso 5: Destruir usuario

Este caso de uso permite al usuario destruir su cuenta, junto con todas las fotos y análisis de estas asociados a la cuenta.

Caso de uso 5	Destruir usuario
Actores	Usuario del sistema.
Descripción	El actor solicita al sistema la destrucción del usuario que actualmente está activo
Precondiciones	El actor usuario tiene una sesión abierta.
Flujo normal	1.-El actor usuario solicita la destrucción del usuario que actualmente está activo 2.-El sistema destruye el usuario, junto con todas las fotos y resultados asociados a esa cuenta y nos muestra la pantalla de inicio de la aplicación
Postcondiciones	El usuario actual ha sido destruido y la aplicación nos muestra la pantalla de inicio
Flujos alternativos	Ninguno

Figura 45: Caso de uso número 5

Caso de uso 6: Subir foto de la cámara

Este caso de uso permite al usuario subir una nueva foto a la aplicación directamente desde la cámara de su dispositivo móvil, el usuario además deberá proporcionar una serie de datos adicionales para poder subir la foto tales como nombre o privacidad.

Caso de uso 6	Subir una foto de la cámara
Actores	Usuario del sistema.
Descripción	El actor solicita al sistema realizar una foto con la cámara y subirla al servidor.
Precondiciones	El usuario tiene una sesión abierta.
Flujo normal	<ol style="list-style-type: none">1.-El actor usuario solicita al sistema realizar una foto y subirla al servidor.2.-El sistema solicita al actor usuario que tome una foto con la cámara.3.-El actor usuario toma la foto con la cámara.4.-El sistema muestra la foto al actor usuario y solicita la confirmación.5.-El actor usuario confirma la subida de la foto.6.-El sistema analiza la foto, la almacena y muestra el resultado del análisis al actor usuario.
Postcondiciones	La foto ha sido subida, almacenada y analizada por el sistema y estará disponible para futuras operaciones.
Flujos alternativos	<p>3a,5a. El actor decide cancelar el caso de uso en curso. En este caso el caso de uso queda sin efecto.</p> <p>3a. El sistema no dispone de cámara de fotos. En este caso se notifica al usuario que no se puede llevar a cabo dicha acción y el caso de uso queda sin efecto</p>

Figura 46: Caso de uso número 6

Caso de uso 7: Subir foto de la galería

Este caso de uso permite al usuario subir una nueva foto a la aplicación directamente desde la galería de su dispositivo móvil, el usuario además deberá proporcionar una serie de datos adicionales para poder subir la foto tales como nombre o privacidad.

Caso de uso 7	Subir una foto de la galería
Actores	Usuario del sistema.
Descripción	El actor solicita al sistema subir una foto almacenada en la galería de su dispositivo
Precondiciones	El usuario tiene una sesión abierta.
Flujo normal	<ol style="list-style-type: none">1.-El actor usuario solicita al sistema subir una foto de la galería del dispositivo2.-El sistema muestra las distintas fotos del dispositivo y pide que seleccione una de ellas3.-El actor usuario selecciona una de ellas4.-El sistema almacena la foto
Postcondiciones	La foto ha sido subida, almacenada y analizada por el sistema y estará disponible para futuras operaciones.
Flujos alternativos	<ol style="list-style-type: none">2a. El dispositivo local no cuenta actualmente con fotos disponibles para subirse. En este caso el sistema notifica al usuario que no hay archivos disponibles y el caso de uso queda sin efecto3a. El actor decide cancelar el caso de uso en curso. En este caso el caso de uso queda sin efecto.

Figura 47: Caso de uso número 7

Caso de uso 8: Ver mis fotos

Este caso de uso permite al usuario ver la totalidad de las fotos que tiene subidas a la aplicación en el momento actual.

Caso de uso 8	Ver mis fotos
Actores	Usuario del sistema.
Descripción	El actor solicita al sistema ver las fotos que tiene subidas
Precondiciones	El usuario tiene una sesión abierta.
Flujo normal	1.-El actor usuario solicita al sistema que le muestre las fotos que tiene subidas 2.-El sistema muestra una lista con todas las fotos que tiene actualmente subidas al sistema
Postcondiciones	El sistema muestra una lista con todas las fotos que tiene subidas al sistema
Flujos alternativos	2a. La cuenta que actualmente tiene abierta la sesión no tiene fotos que puedan ser seleccionadas. En este caso el sistema mostraría un mensaje indicando que no hay fotos disponibles y el caso de uso queda sin efecto.

Figura 48: Caso de uso número 8

Caso de uso 9: Buscar foto

Este caso de uso permite al usuario buscar fotos dentro de la totalidad de fotos que están subidas a la aplicación, permitirá tanto buscar por categorías o por nombre y entre las todas las fotos públicas o solo las propias.

Caso de uso 9	Buscar foto
Actores	Usuario del sistema.
Descripción	El actor solicita al sistema la búsqueda de fotos de acuerdo a una serie de parámetros
Precondiciones	El usuario tiene una sesión abierta.
Flujo normal	<ol style="list-style-type: none">1.-El actor usuario solicita al sistema la búsqueda de una foto en el sistema.2.-El sistema muestra los posibles filtros de selección y solicita al usuario que los seleccione.3.-El actor usuario selecciona los filtros que considere para realizar su búsqueda.4.-El sistema muestra una lista con todos los resultados de acuerdo a esos filtros.
Postcondiciones	Las fotos que cumplen las condiciones seleccionadas aparecen listadas al actor usuario
Flujos alternativos	<ol style="list-style-type: none">2a. La cuenta que actualmente tiene abierta la sesión no tiene fotos que puedan ser seleccionadas o filtradas. En este caso el sistema mostraría un mensaje indicando que no hay fotos disponibles y el caso de uso queda sin efecto.3a. El actor decide cancelar el caso de uso en curso. En este caso el caso de uso queda sin efecto.4a. No existen fotos que cumplan con las condiciones seleccionadas en el paso 2. En este caso el sistema mostraría un mensaje indicando que no hay fotos que cumplan con las condiciones y el caso de uso volvería al paso 3.

Figura 49: Caso de uso número 9

Caso de uso 10: Ver foto

Este caso de uso permite la visualización de la foto seleccionada previamente, junto con todos los atributos asociados a ella.

Caso de uso 10	Ver foto
Actores	Usuario del sistema.
Descripción	El actor solicita al sistema ver una de las fotos que tiene almacenadas en el sistema.
Precondiciones	El usuario tiene una sesión abierta, buscar foto ha sido ejecutado con éxito.
Flujo normal	<ol style="list-style-type: none">1.-El actor usuario solicita al sistema visualizar una de las fotos que tiene almacenada.2.-El sistema solicita al usuario que seleccione una de las fotos mostradas como resultado del caso de uso anterior3.-El actor usuario elige una de las fotos listadas.4.-El sistema muestra la imagen que el actor usuario ha seleccionado.
Postcondiciones	La foto seleccionada por el usuario aparece mostrada por pantalla
Flujos alternativos	<ol style="list-style-type: none">2a. La cuenta que actualmente tiene abierta la sesión no tiene fotos almacenadas. En este caso el sistema mostraría un mensaje indicando que no hay fotos disponibles y el caso de uso queda sin efecto.3a. El actor decide cancelar el caso de uso en curso. En este caso el caso de uso queda sin efecto.

Figura 50: Caso de uso número 10

Caso de uso 11: Borrar foto

Este caso de uso permite al propietario de una foto la destrucción de dicha foto del almacenamiento de la aplicación.

Caso de uso 11	Borrar foto
Actores	Usuario del sistema.
Descripción	El actor solicita al sistema la eliminación de una de las fotos que tiene almacenadas
Precondiciones	El usuario tiene una sesión abierta, se ha ejecutado con éxito el caso de uso ver foto y la foto pertenece al usuario actual
Flujo normal	1.-El actor usuario solicita al sistema la eliminación de una de las fotos que tiene almacenada 2.-El sistema elimina la foto que el actor usuario esta visualizando actualmente
Postcondiciones	La foto seleccionada por el usuario es eliminada del almacenamiento del sistema
Flujos alternativos	2a. La cuenta que actualmente tiene abierta la sesión no tiene fotos almacenadas. En este caso el sistema mostraría un mensaje indicando que no hay fotos disponibles y el caso de uso queda sin efecto. 3a. El actor decide cancelar el caso de uso en curso. En este caso el caso de uso queda sin efecto.

Figura 51: Caso de uso número 11

Caso de uso 12: Guardar foto en local

Este caso de uso permite guardar en la galería del dispositivo que está actualizando en el momento actual.

Caso de uso 12	Guardar foto en local
Actores	Usuario del sistema.
Descripción	El actor solicita al sistema la descarga de una de las fotos que dicho usuario tiene almacenado
Precondiciones	El usuario tiene una sesión abierta, ver foto ha sido ejecutado con éxito
Flujo normal	1.-El actor usuario solicita que la foto que actualmente está viendo sea almacenada en local. 2.-El sistema descarga la foto en el dispositivo local del actor usuario.
Postcondiciones	La foto seleccionada está disponible en la galería local del dispositivo utilizado.
Flujos alternativos	2a. La cuenta que actualmente tiene abierta la sesión no tiene fotos que puedan ser seleccionadas o filtradas. En este caso el sistema mostraría un mensaje indicando que no hay fotos disponibles y el caso de uso queda sin efecto. 3a. El actor decide cancelar el caso de uso en curso. En este caso el caso de uso queda sin efecto.

Figura 52: Caso de uso número 12

Caso de uso 13: Modificar datos de una foto

Este caso de uso permite la modificación de los datos de la foto que actualmente esta visualizando al propietario de dicha foto.

Caso de uso 13	Modificar datos de una foto
Actores	Usuario del sistema.
Descripción	El actor solicita modificar los datos asociados a una foto
Precondiciones	El usuario tiene una sesión abierta, ver foto ha sido ejecutado con éxito y la foto pertenece al usuario actual
Flujo normal	1.-El actor usuario solicita modificar los datos asociados a una foto 2.-El sistema solicita al usuario que introduzca los nuevos datos 3.-El actor usuario introduce los nuevos datos 4.-El sistema almacena los nuevos datos como datos de la foto
Postcondiciones	Los datos asociados a la foto han sido modificados
Flujos alternativos	3a. El actor decide cancelar el caso de uso en curso. En este caso el caso de uso queda sin efecto. 4a. Los datos proporcionados como datos de la foto se solapan con otros de otra foto. En este caso el sistema notificará al sistema que esos datos se solapan y el caso de uso volverá al paso número 2.

Figura 53: Caso de uso número 13

4.3. Diagramas de secuencia

-Caso de uso 1: Registrar usuario

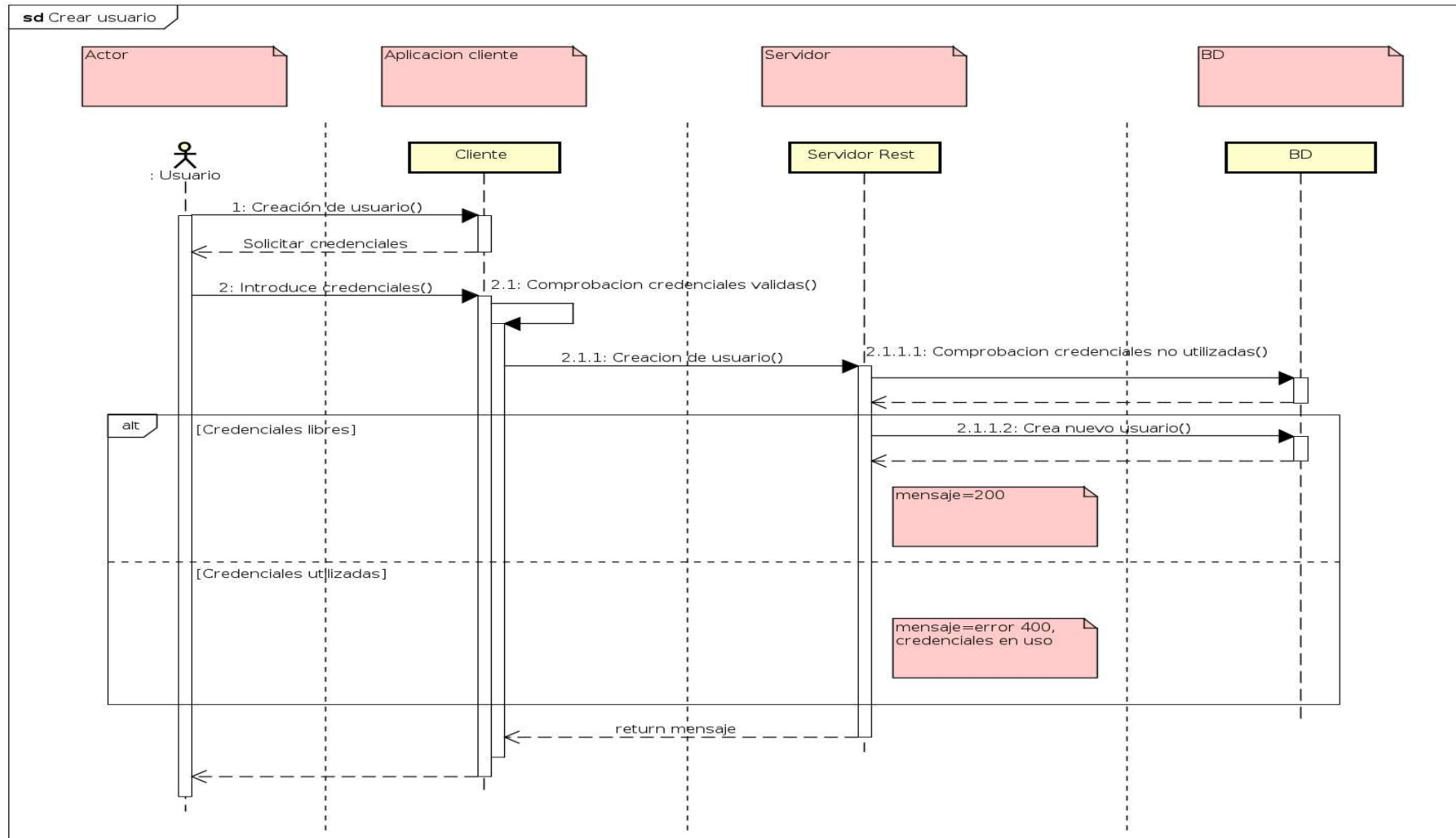


Figura 54: Diagrama de secuencia del caso de uso 1

-Caso de uso 2: Iniciar sesión

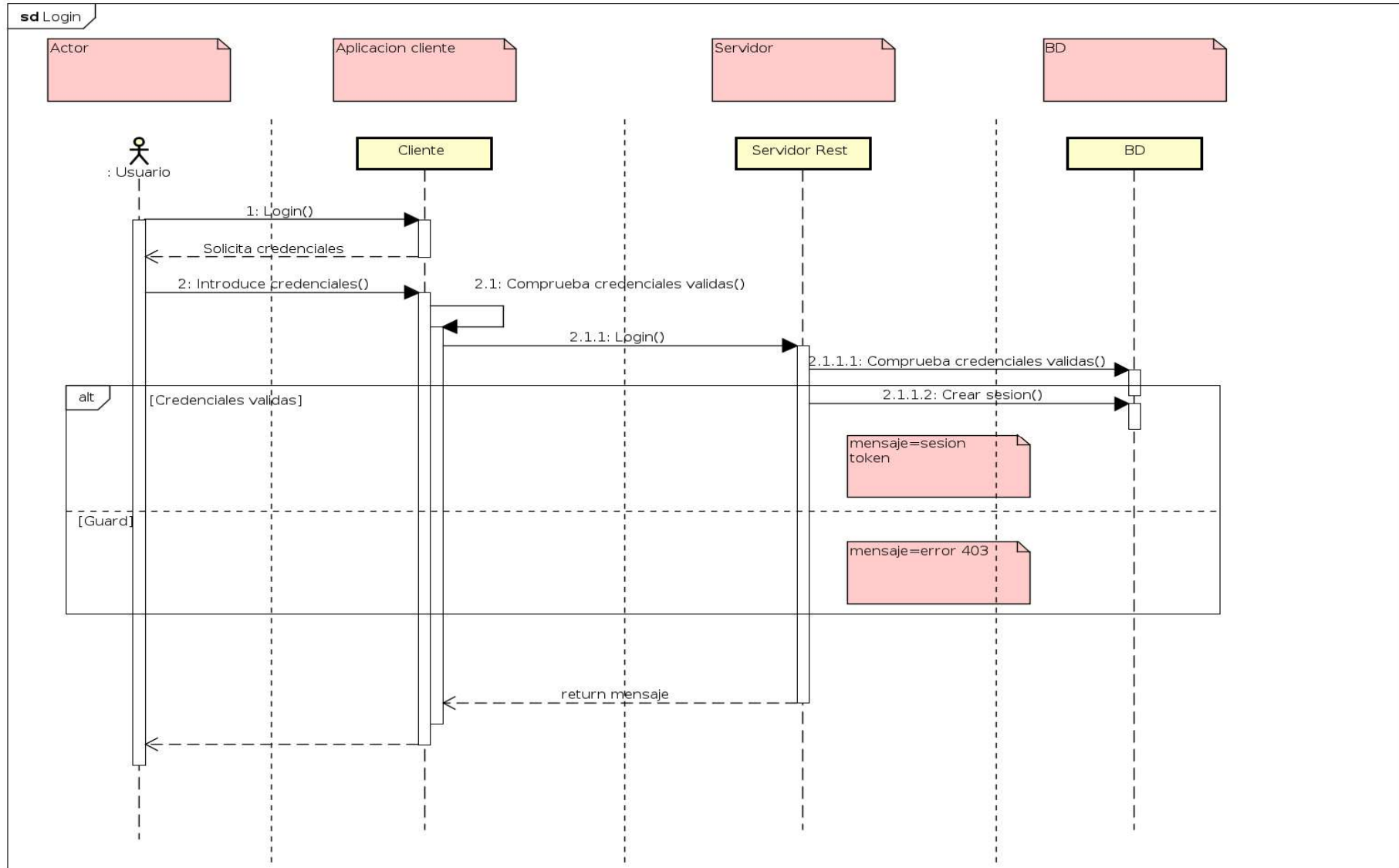


Figura 55: Diagrama de secuencia del caso de uso 2

-Caso de uso 3: Cerrar sesión

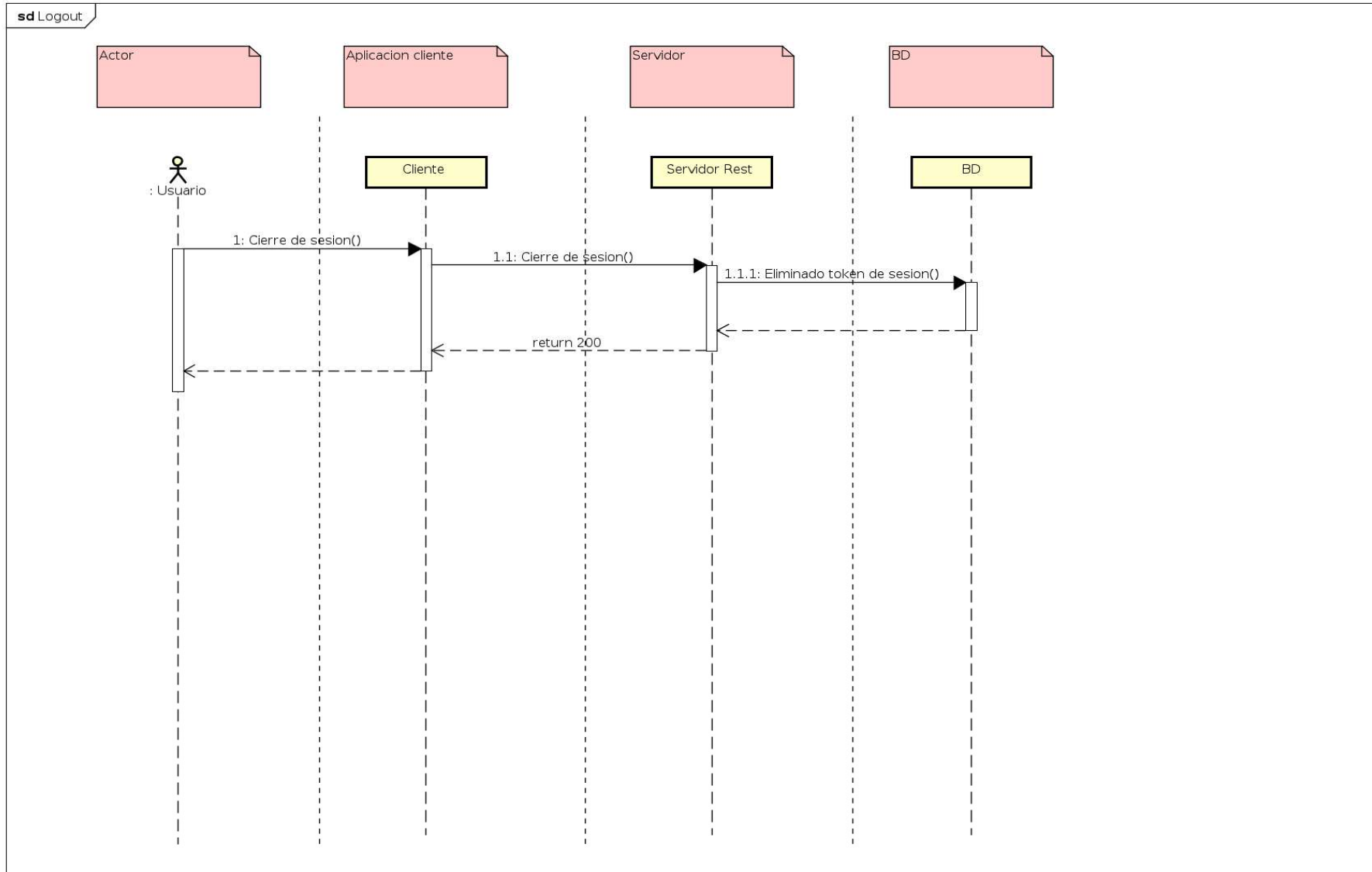


Figura 56: Diagrama de secuencia del caso de uso 3

-Caso de uso 4: Modificar credenciales

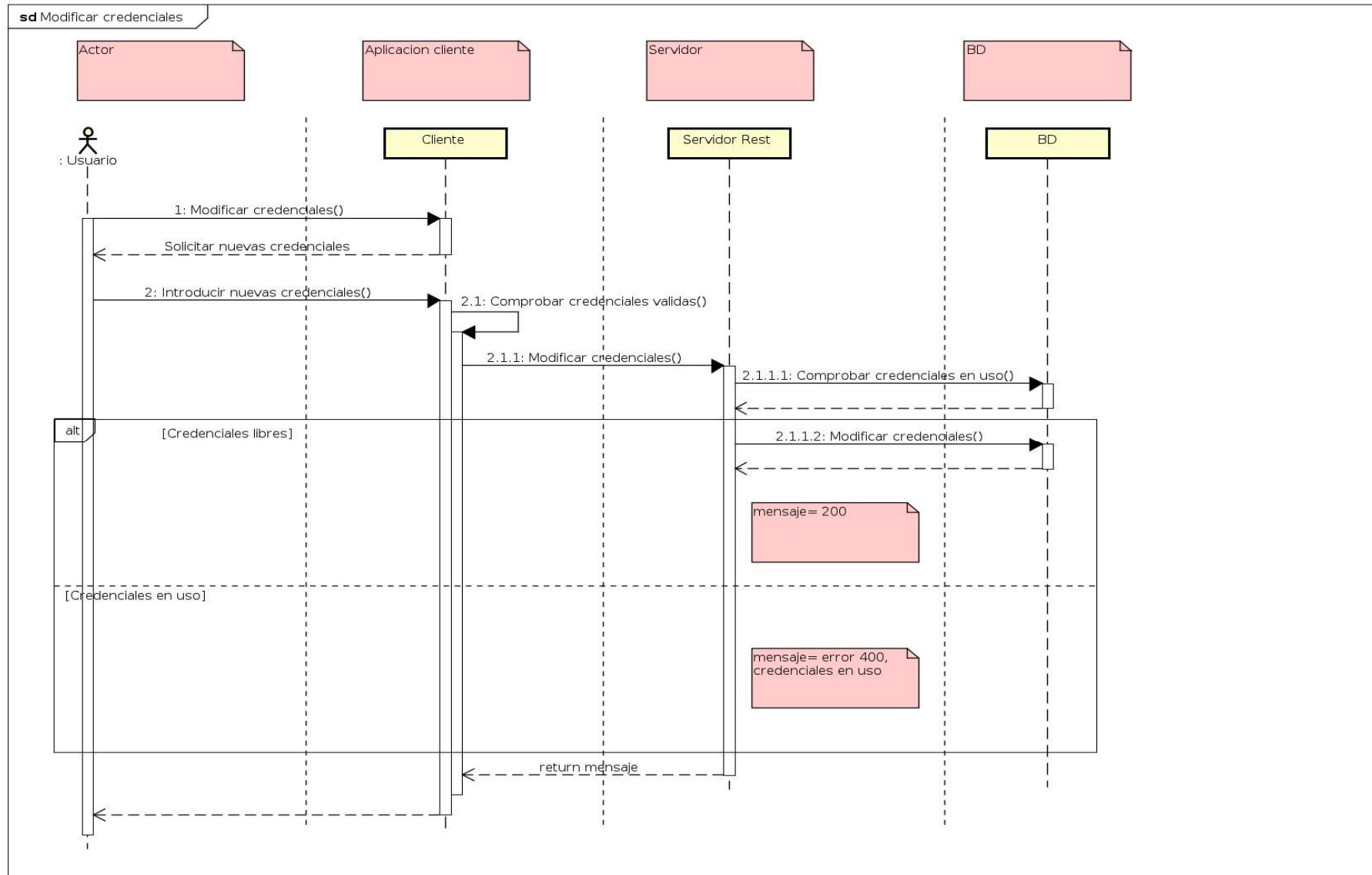


Figura 57: Diagrama de secuencia del caso de uso 4

-Caso de uso 5: Destruir Usuario

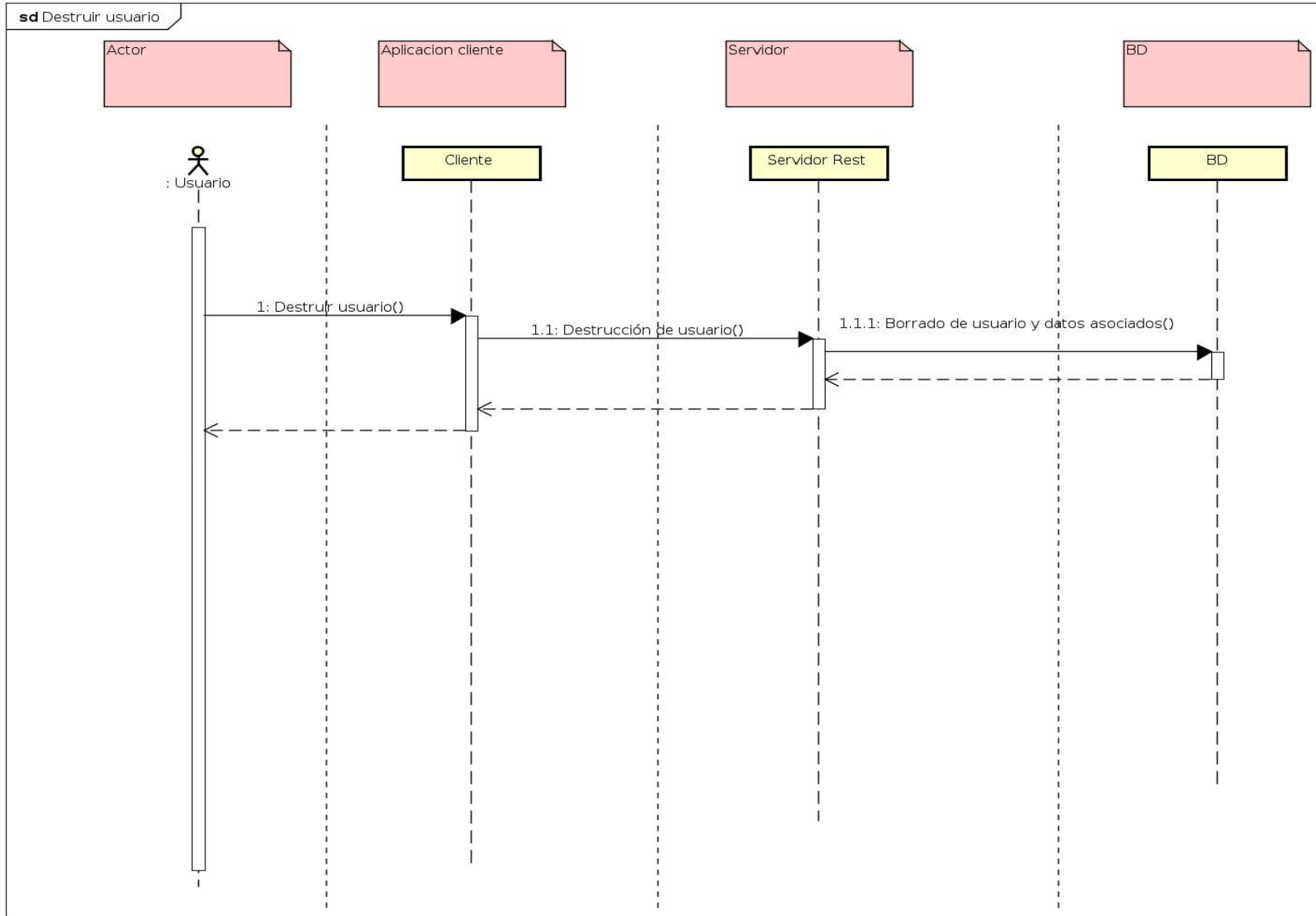


Figura 58: Diagrama de secuencia del caso de uso 5

-Caso de uso 6: Subir foto de la cámara

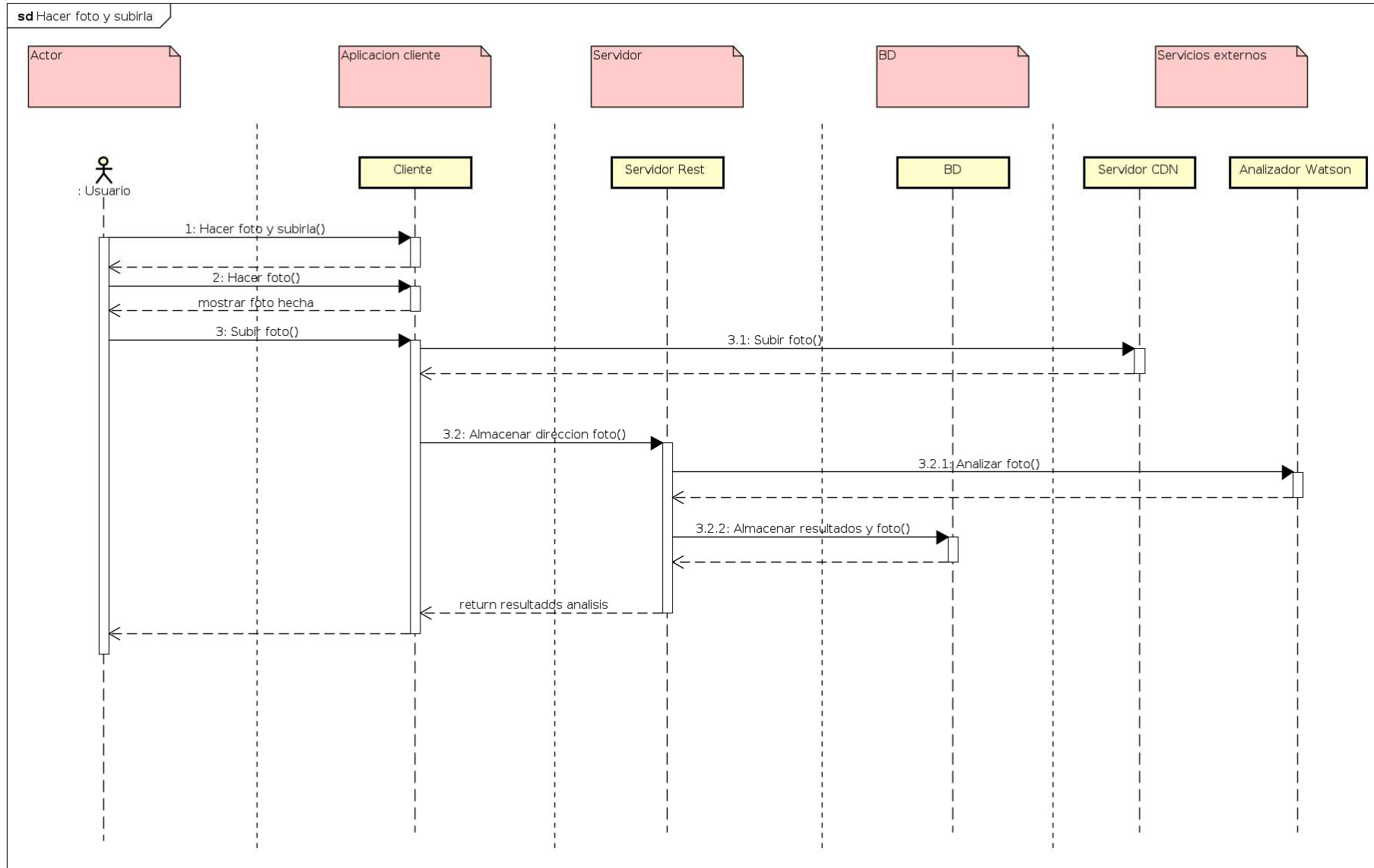


Figura 59: Diagrama de secuencia del caso de uso 6
53

-Caso de uso 7: Subir foto de la galería

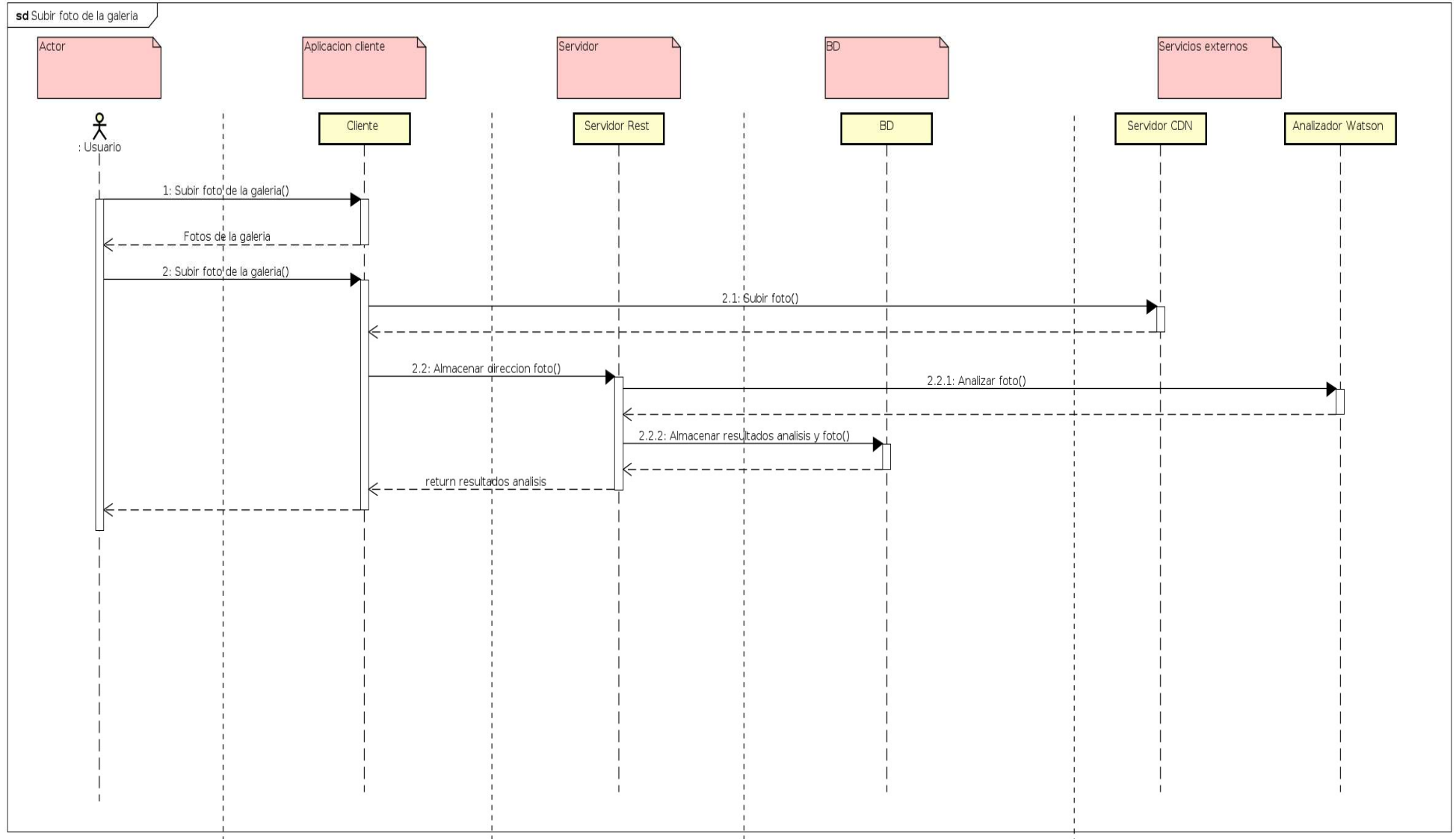


Figura 59: Diagrama de secuencia del caso de uso 7

-Caso de uso 8: Ver mis fotos

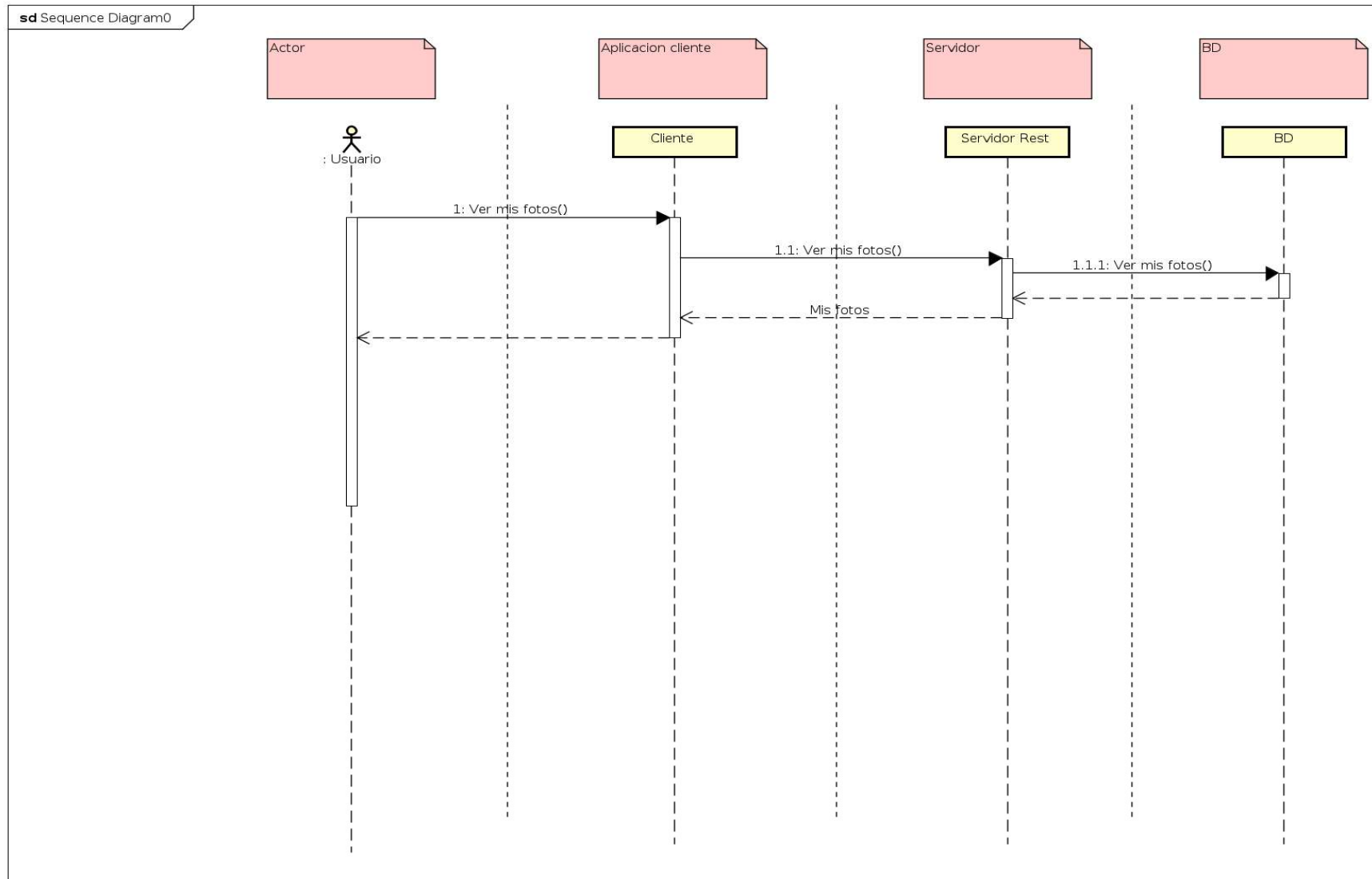


Figura 60: Diagrama de secuencia del caso de uso 8

-Caso de uso 9: Buscar fotos

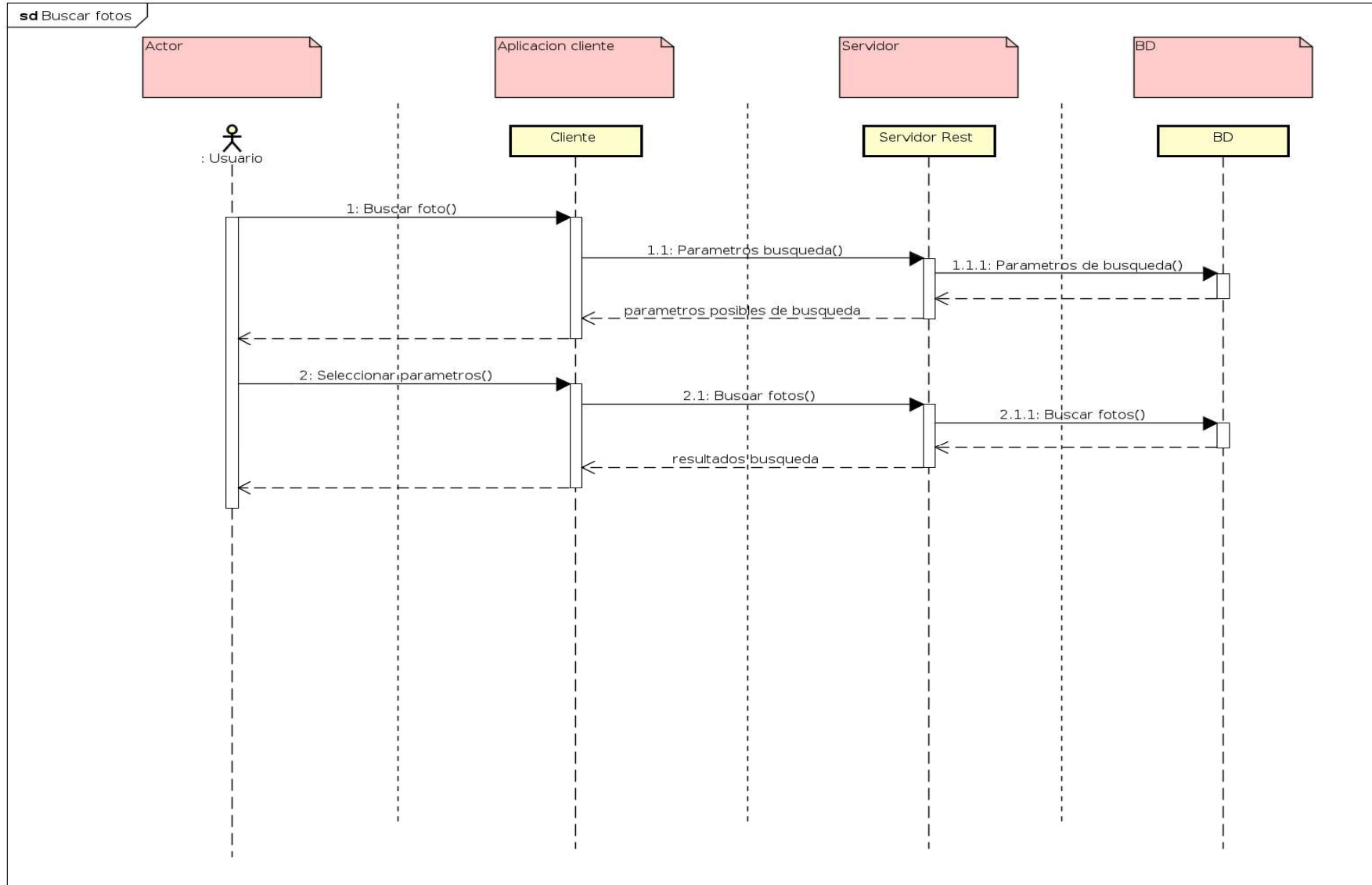


Figura 61: Diagrama de secuencia del caso de uso 9

-Caso de uso 10: Ver foto

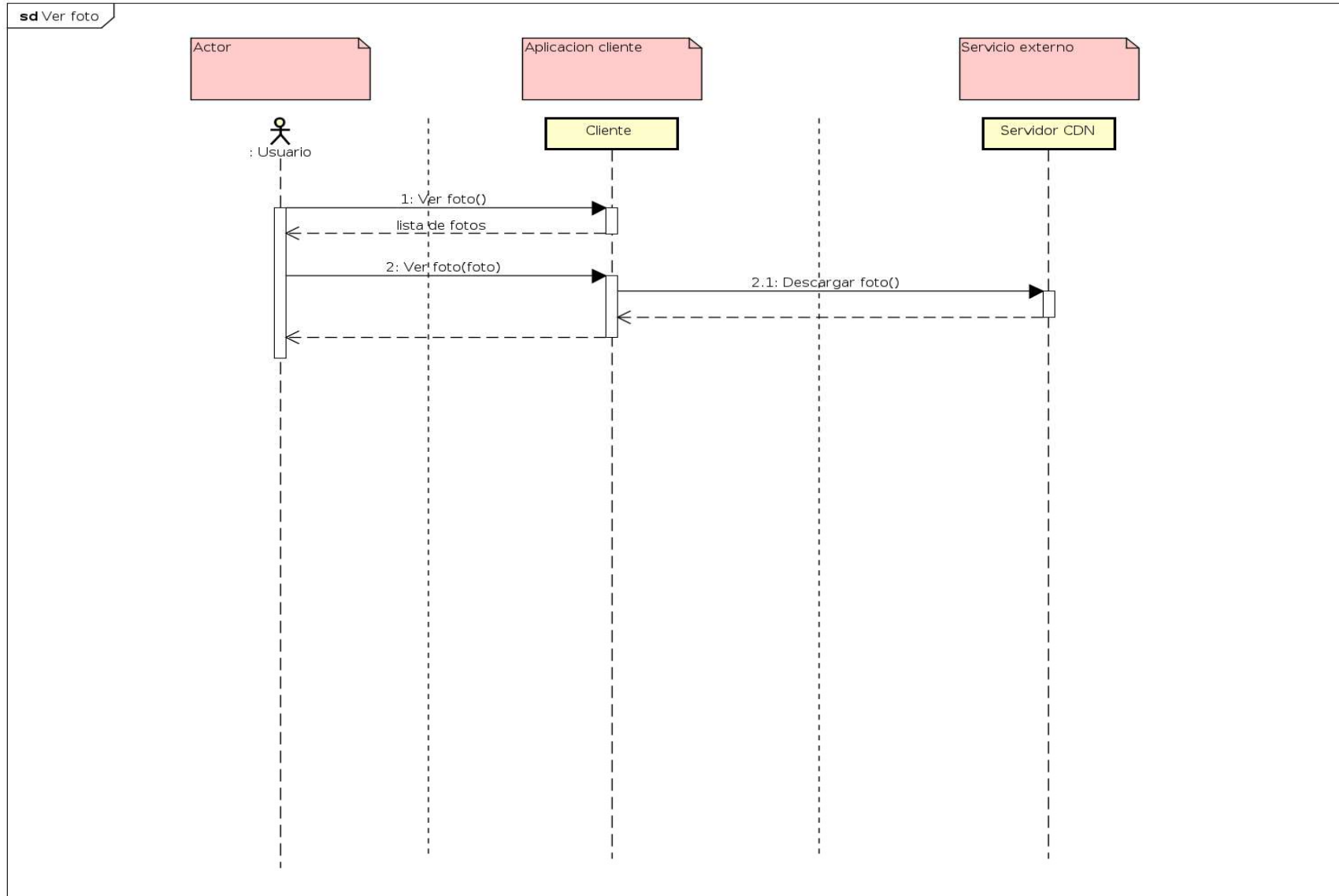


Figura 62: Diagrama de secuencia del caso de uso 10

-Caso de uso 11: Descargar foto

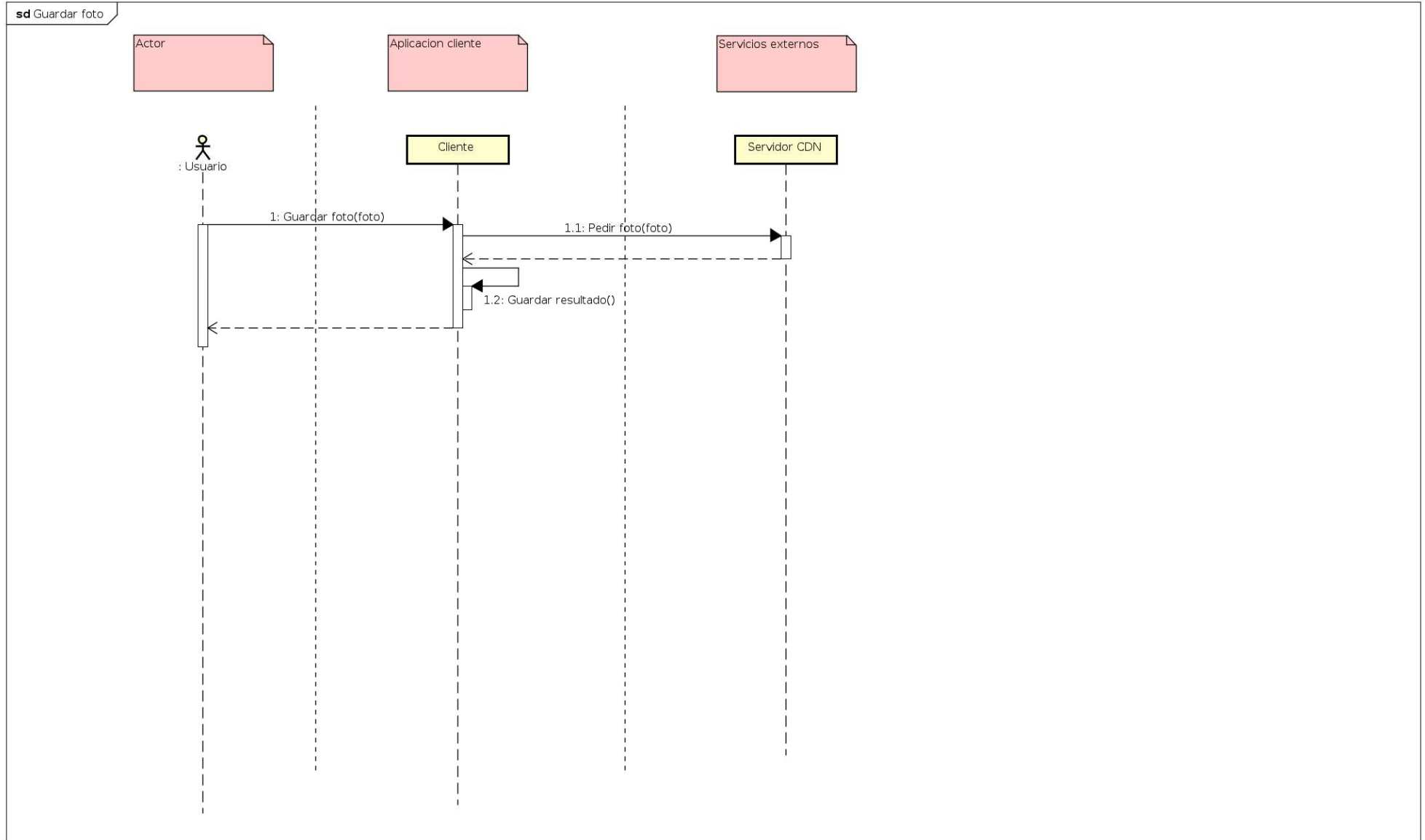


Figura 63: Diagrama de secuencia del caso de uso 11

-Caso de uso 12: Borrar foto

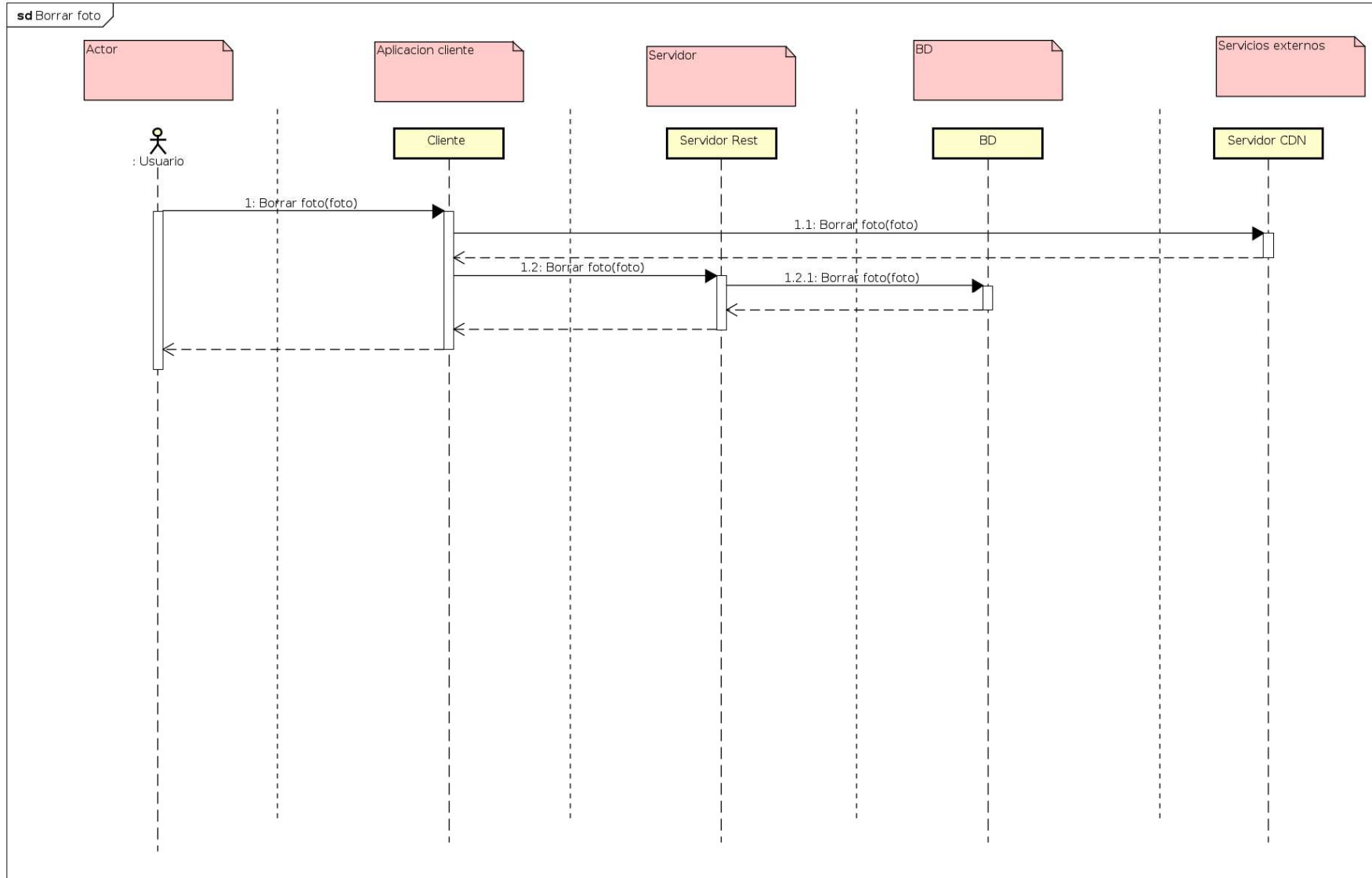


Figura 64: Diagrama de secuencia del caso de uso 12

-Caso de uso 13: Modificar datos de una foto

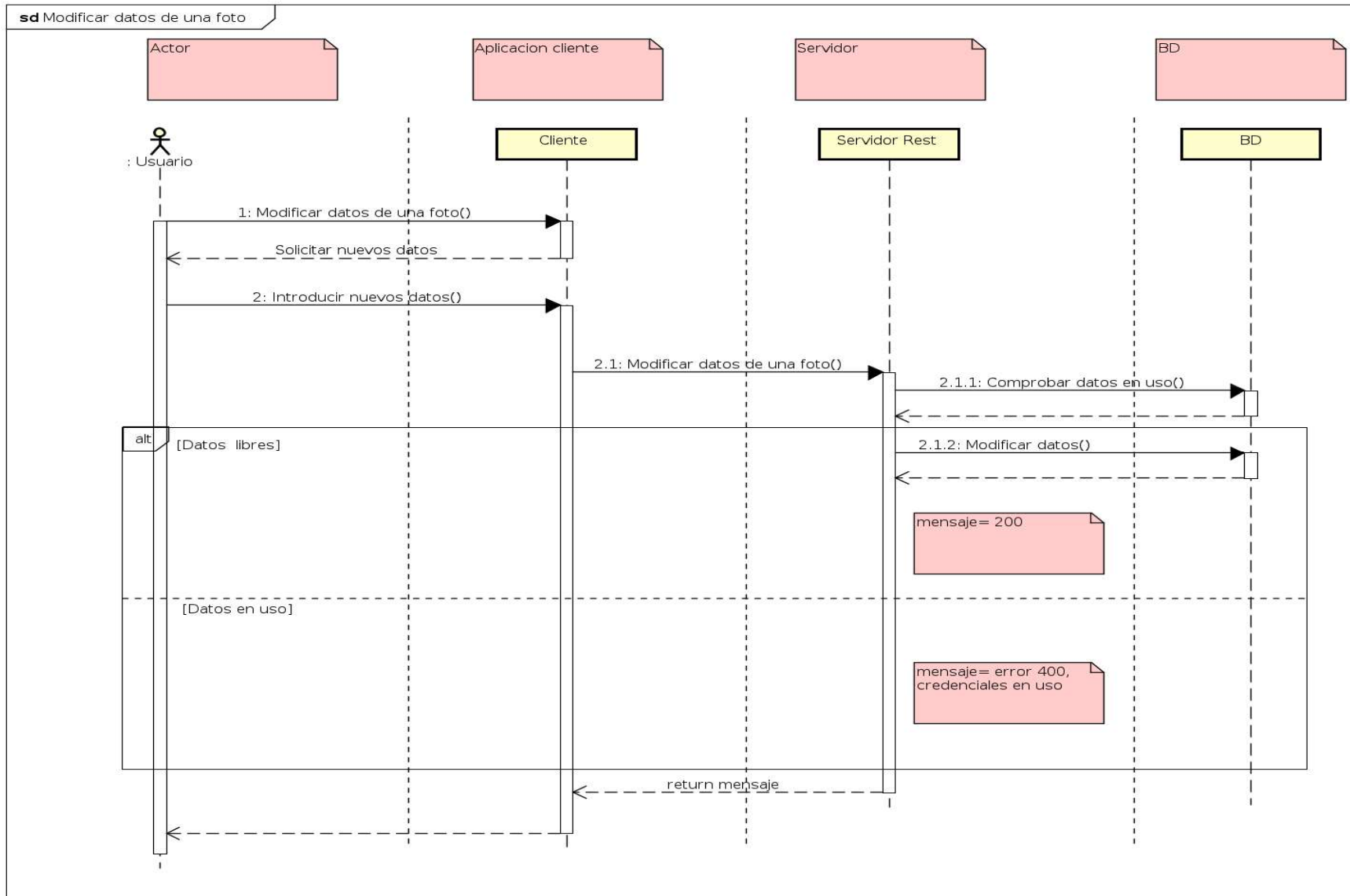


Figura 65: Diagrama de secuencia del caso de uso 13

4.4. Arquitectura

Como podemos ver en la figura 65, la aplicación dispone de cinco módulos distintos:

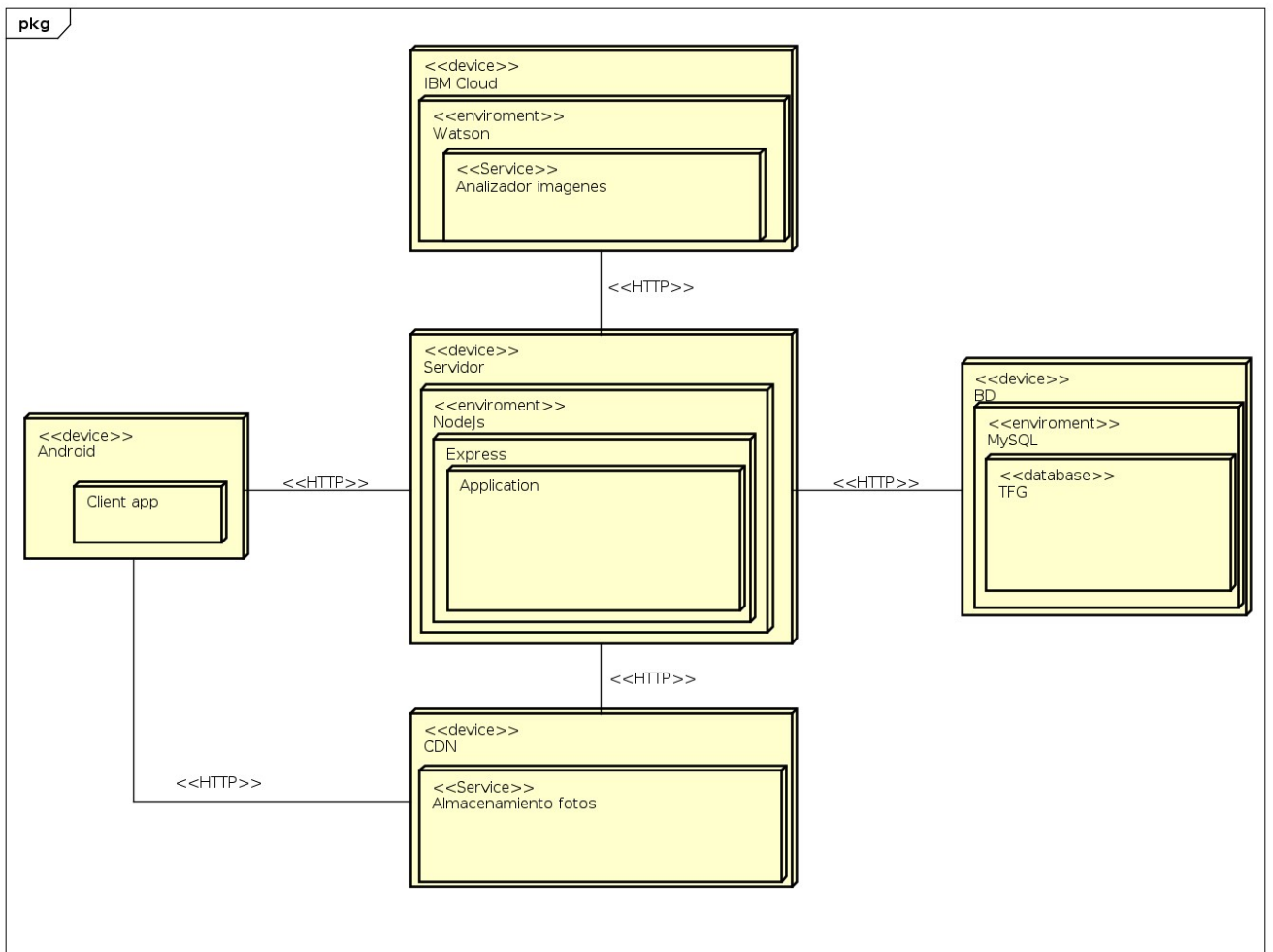


Figura 65: Esquema de la arquitectura de la aplicación

Primeramente disponemos de un cliente android desplegado en el cliente final que es el encargado de implementar los casos de uso a los que tendrá acceso el usuario final. Este módulo se comunica con el módulo servidor mediante las operaciones rest vía Http y con la CDN que hemos elegido para almacenar las fotos.

La CDN es el segundo módulo, este módulo es el encargado de brindar acceso a las fotos almacenadas y de almacenar fotos nuevas. La CDN escogida ofrece un acceso público mediante una url publica a las distintas fotos, por lo que almacenamos dicha url para los casos de uso de análisis y descarga en lugar de utilizar las fotos directamente.

El servidor rest ha sido construido con Nodejs, y es desplegado como una aplicación en IBM Cloud. Aquí se implementan una serie de operaciones rest que se utilizarán como interfaz de acceso entre el cliente android con sus casos de uso y los datos almacenados en la base de datos. Dado que utiliza la tecnología rest, permitiría la inclusión o sustitución del cliente rest por uno distinto, lo que permitiría el crecimiento o mejora de la aplicación. Este modulo utiliza peticiones Http como método de comunicación con el resto de módulos.

Los módulos de servicios web son el cuarto módulo, en este grupo se podrían agrupar también la CDN y la base de datos, pero hemos decidido explicarlos por separado para hablar más en detalle. En este módulo utilizamos el servicio Watson de IBM Cloud para el análisis de las imágenes. Este servicio nos permite tanto subir una imagen como seleccionar una url y nos devuelve los resultados de dicha imagen como un Json. Los resultados de este análisis son los que almacenaremos en la base de datos como resultados del análisis de la foto para su posterior búsqueda.

Por último, el último módulo es la base de datos. Dicha base de datos ha sido desplegada como servicio en IBM Cloud, en ella se almacenan los datos de los usuarios, sus fotos y las detecciones de elementos en dichas fotos.

5. Implementación y pruebas

5.1. Introducción

En esta sección describiremos tanto el funcionamiento mismo de la aplicación como detalles de la implementación que se ha considerado importante explicar de manera más detallada.

5.2. Implementación

Todo el código desarrollado se divide en dos partes, la parte del cliente Android y la parte del servidor REST, primero hablaremos de la parte del servidor.

Servidor REST

El servidor REST ha sido desarrollado en NodeJs como hemos indicado anteriormente, ofrece una interfaz con una serie de operaciones que el cliente Android invocará para ejecutar las operaciones, utiliza el framework express para la creación de la interfaz que ofrece al exterior. Para el acceso a la base de datos se ha utilizado el ORM Sequelize que ofrece una interfaz de acceso a la base de datos muy sencilla a la hora de acceder a las operaciones de acceso a la base de datos.

A continuación enumeraremos las operaciones que han sido implementadas en el servidor para el funcionamiento de la aplicación y explicaremos de manera breve para que se usa cada una de ellas, para ello utilizaremos las operaciones Android que las llaman en lugar del código NodeJs ya que es mucho más visual.

- Operaciones relacionadas con la sesión

```
@POST("/login/register")
Call<Void> register(@Header("username") String username, @Header("password") String password);

@GET("/login/login")
Call<Token> login(@Header("username") String username, @Header("password") String password);

@GET("/login/verifyToken")
Call<Void> verifyToken(@Header("token") String token);
```

Figura 66: Operaciones Rest relacionadas con la sesión

Este módulo está centrado en el inicio de sesión y creación de nuevos usuarios, han sido implementadas un total de tres operaciones:

1. Register: Esta operación permite la creación de nuevos usuarios en la aplicación, el usuario pasado ha de ser único y la contraseña se almacena cifrada con sha-256 para mejorar la seguridad de la aplicación.
2. Login: Esta operación permite la creación de sesiones y el acceso a la aplicación verifica que existe el usuario y que la contraseña corresponde con la proporcionada y si esto se cumple crea una token única de sesión y la devuelve al usuario.
3. VerifyToken: Esta operación es utilizada para comprobar al iniciar la aplicación cuando ya hay una cuenta almacenada en el dispositivo, comprueba que la token de sesión almacenada sigue siendo válida y si lo es inicia la aplicación, si no remitirá al inicio de la aplicación.

- Operaciones relacionadas con el usuario

```
@PUT("/user/updateUsername")
Call<Void> updateUsername(@Header("token") String token,@Header("username") String username);

@PUT("/user/updatePassword")
Call<Void> updatePassword(@Header("token") String token,@Header("password") String password);

@DELETE("/user/delete")
Call<Void> deleteUser(@Header("token") String token);

@GET("/user/getUsernameFromToken")
Call<User> getUserFromToken(@Header("token") String token);
```

Figura 67: Operaciones Rest relacionadas con el usuario

Este módulo está centrado en operaciones de gestión de las cuentas de los usuarios, han sido implementadas un total de cuatro operaciones:

1. UpdateUsername: Esta operación permite modificar el nombre de usuario. Para ello comprueba que el nuevo nombre no está en uso y si es así modifica el nombre de usuario asociado a la token proporcionada.
2. UpdatePassword: Esta operación permite modificar la contraseña del usuario. Para ello cifra con sha-256 la nueva contraseña y la coloca en el campo correspondiente en el usuario asociado a la token proporcionada.

3. DeleteUser: Esta operación permite borrar un usuario. Para ello busca el usuario asociado a la token proporcionada y lo destruye.
4. GetUserFromToken: Esta operación permite obtener los datos de un usuario. Para ello busca el usuario asociado a la token de sesión proporcionada y lo elimina de la base de datos

- Operaciones relacionadas con las fotos y su gestión

```

@POST("/upload/uploadImage")
Call<Void> uploadImage(@Header("token") String token,@Header("id") String id, @Header("url") String url,@Header("nombre") String name,@Header("privada") boolean privada)

@GET("/photos/getFotosCategoria")
Call<List<Photo>> getFotosCategoria(@Header("token") String token,@Header("filtro") String subcategory);

@GET("/photos/getFotosUsuarioCategoria")
Call<List<Photo>> getFotosUsuarioCategoria(@Header("token") String token,@Header("filtro") String subcategory);

@GET("/photos/getFotosNombre")
Call<List<Photo>> getFotosNombre(@Header("token") String token,@Header("nombre") String name);

@GET("/photos/getFotosUsuarioNombre")
Call<List<Photo>> getFotosUsuarioNombre(@Header("token") String token,@Header("nombre") String name);

@GET("/photos/getMisFotos")
Call<List<Photo>> getMyPhotos(@Header("token") String token);

@PUT("/photos/updateFoto")
Call<Void> updateFoto(@Header("token") String token, @Header("privada") int privada,@Header("nombre") String name, @Header("id") String id);

@DELETE("/photos/deleteFoto")
Call<Void> deleteFoto(@Header("token") String token, @Header("id") String idFoto);

@GET("/photos/getCategorias")
Call<List<Subcategory>> getCategorias(@Header("token") String token);

@GET("/photos/getCategoriasFoto")
Call<List<Subcategory>> getCategoriasFoto(@Header("token") String token,@Header("id")String id);

@GET("/photos/cdnCredentials")
Call<CDNCredentials> getCredentials(@Header("token") String token);

```

Figura 68: Operaciones Rest relacionadas con las fotos y su gestión

Este módulo se centra en operaciones de gestión de las fotos de la aplicación, cuenta con un total de once operaciones:

1. UploadImage: Esta operación de registrar nuevas fotos en la aplicación. Para ello crea una nueva foto en el almacenamiento interno con los parámetros proporcionados y pasa al servicio Watson la dirección de la foto almacenada en Clouinary, Watson devuelve los resultados que también se almacenan en la base de datos.
2. GetPhotosCategoria: Esta operación es una de las encargadas de buscar fotos en la aplicación. Esta en concreto busca fotos a partir de una de las categorías seleccionadas pudiendo ser propias o de otros usuario marcadas como públicas.
3. GetPhotosUsuarioCategoria: Esta operación es una de las encargadas de buscar fotos en la aplicación. Esta en concreto es la encargada de buscar fotos a partir de una de las categorías, pero solo buscara aquellas que pertenecen al usuario que esta logueado en esa sesión.
4. GetPhotosNombre: Esta operación es una de las encargadas de buscar fotos en la aplicación. Esta en concreto buscará fotos a partir del nombre proporcionado a la foto, pudiendo ser tanto fotos mías como de otros usuarios marcadas como públicas.
5. GetPhotosUsuarioNombre: Esta operación es una de las encargadas de buscar fotos en la aplicación. En este caso buscará fotos a partir del nombre proporcionado, pero solo buscará entre las fotos que pertenecen al usuario logueado actualmente.
6. GetMisFotos: Esta operación es una de las encargadas de buscar fotos en la aplicación. En este caso buscará todas las fotos pertenecientes al usuario que esta logueado actualmente en la aplicación.

7. UpdatePhoto: Esta operación permite la modificación de los parámetros asociados a una foto. Para ello se requiere que la foto pertenezca al usuario logueado y modificará los parámetros actuales sustituyéndolos por los nuevos parámetros proporcionados.
8. DeletePhoto: Esta operación permite la destrucción de fotos. Para ello busca la foto con la id asociada y si pertenece al usuario de la token proporcionada la destruye.
9. GetCategorias: Esta operación permite obtener las categorías existentes actualmente en la aplicación. Para ello busca en las detecciones y devuelve los distintos nombres registrados en la aplicación.
10. GetCategoriasPhoto: Esta operación busca todas las categorías presentes en una foto. Para ello busca todas las detecciones de la foto proporcionada y las devuelve.
11. GetCredentials: Esta operación devuelve los credenciales de acceso de la CDN. Dado que la aplicación accede directamente a Cloudinary es necesario proporcionar los credenciales para que esto pueda ocurrir.

Por último, cabe destacar la función encargada de comprobar que el usuario está en la aplicación, para esto existe una función de comprobación de la token de sesión que comprueba que el usuario esta logueado y la token es correcta para las funciones no relacionadas con inicio de sesión y registro.

Ciente Android

El cliente Android ha sido realizado utilizando Android Studio, en concreto ha sido realizado en java en lugar de kotlin y algunos de los ficheros auxiliares que utiliza android están en XML. Para la realización de las comunicaciones con el servidor se ha utilizado unas librerías auxiliares encargadas de ello, en este caso hemos optado por Retrofit2 que permite de una manera muy sencilla parametrizar las operaciones rest para así poder realizar las consultas de manera sencilla. Por último cabe destacar que la aplicación y las vistas han sido desarrolladas utilizando el modelo vista presentador, para cada una de las vistas creadas.

La aplicación desarrollada se estructura en torno a 3 vistas principales, la vista asociada al inicio de sesión, la asociada a la subida de fotos, la búsqueda de fotos y las opciones y la vista asociada a una foto concreta, a continuación explicaremos cada una de manera individual.

- Inicio de sesión

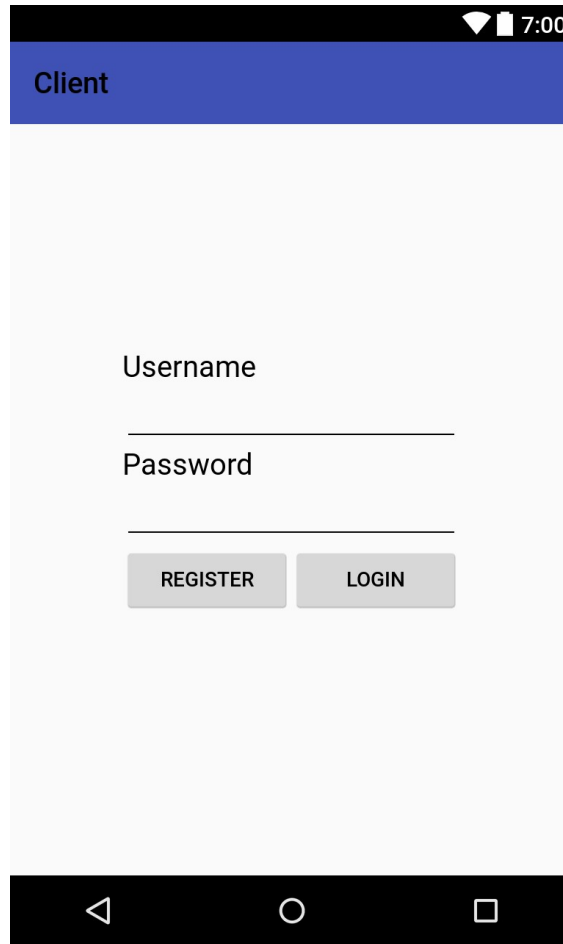


Figura 69: Vista de inicio de sesión

En esta vista están aplicados dos de los casos de uso, el registro y el inicio de sesión. Para ello se coloca el nombre de usuario y la contraseña y se solicita si es un inicio de sesión o un registro. Si no se cumplen las condiciones notificará al usuario de ello. Por último, cabe destacar que esta vista solo se muestra si no hay una token de sesión almacenada para la aplicación o si la que esta almacenada no es válida ya, si esto no se cumple y la token es válida se salta directamente a la ventana principal de la sesión con la token de sesión almacenada.

- Ventana principal de la aplicación

La ventana principal de la aplicación cuenta con cuatro subventanas cada una de ellas para una función concreta, estas son la ventana relacionada con subir fotos, ver mis fotos, buscar fotos y por último modificar los credenciales de la cuenta.

Subir foto

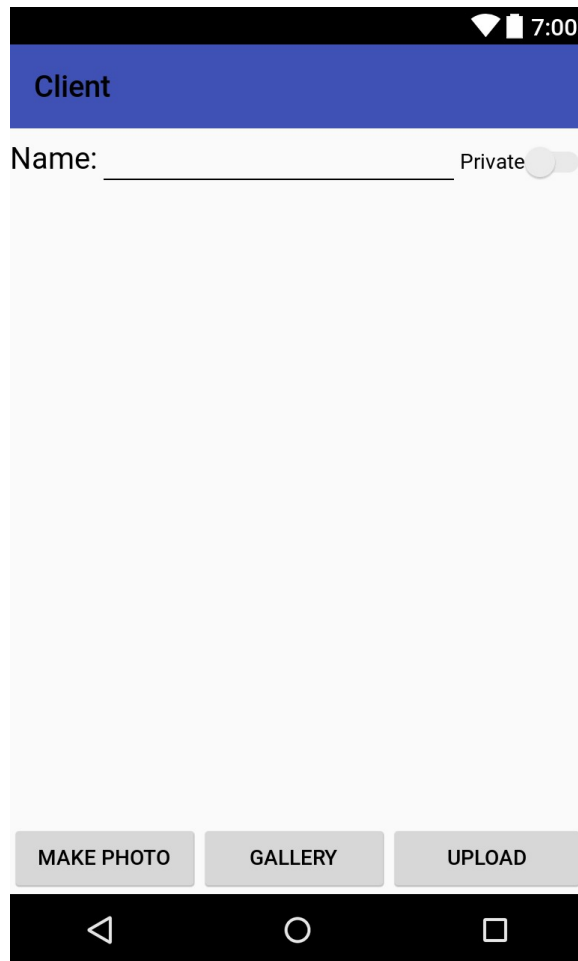


Figura 70: Vista de subida de fotos

En esta ventana se implementan dos de los casos de uso, subir foto desde la cámara y subir foto de la galería. Para ello tenemos tres botones.

-El primero al pulsarlo nos abre la aplicación de de la cámara del teléfono, una vez que este hecha la foto la manda de nuevo a la aplicación, que realiza un postprocesado de la imagen ya que requiere de rotación y la coloca en el centro de la vista.

-El segundo nos abre el menú de selección de fotos de la galería, nos permite seleccionar una foto de la galería y esta aparecerá como en el caso anterior en el centro de la vista.

-El último nos permite subir la foto a la aplicación. Primero subimos la foto a Clouinary que nos devolverá la url pública, después subiremos dicha url junto con el nombre y si es privada o no, la foto se analiza y almacena las direcciones a dicha foto en la base de datos.

Mis fotos

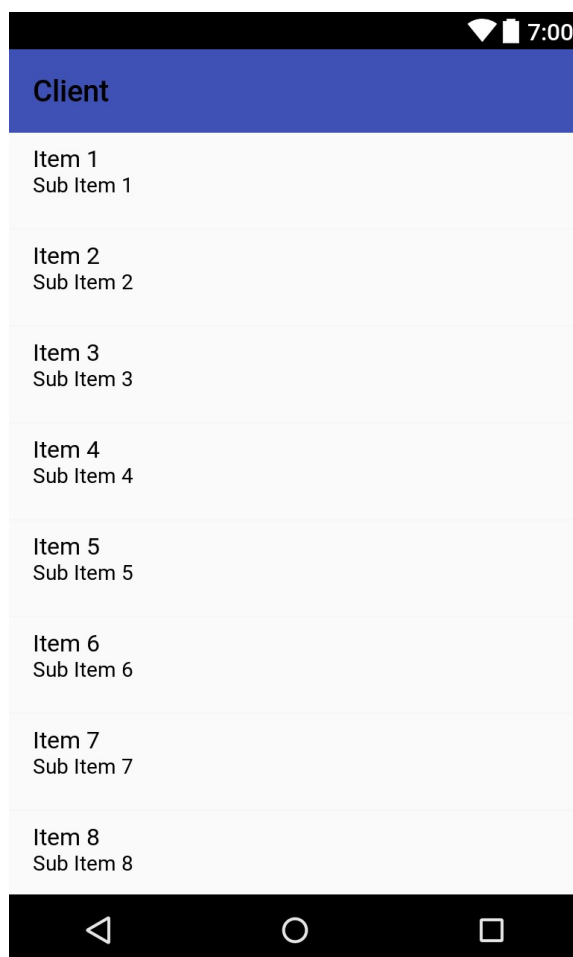


Figura 71: Vista de mis fotos

Esta ventana muestra una lista de nuestras fotos, junto con la fecha en que fue subida y nos permite una visualización de la foto con un botón por entrada.

Buscar foto

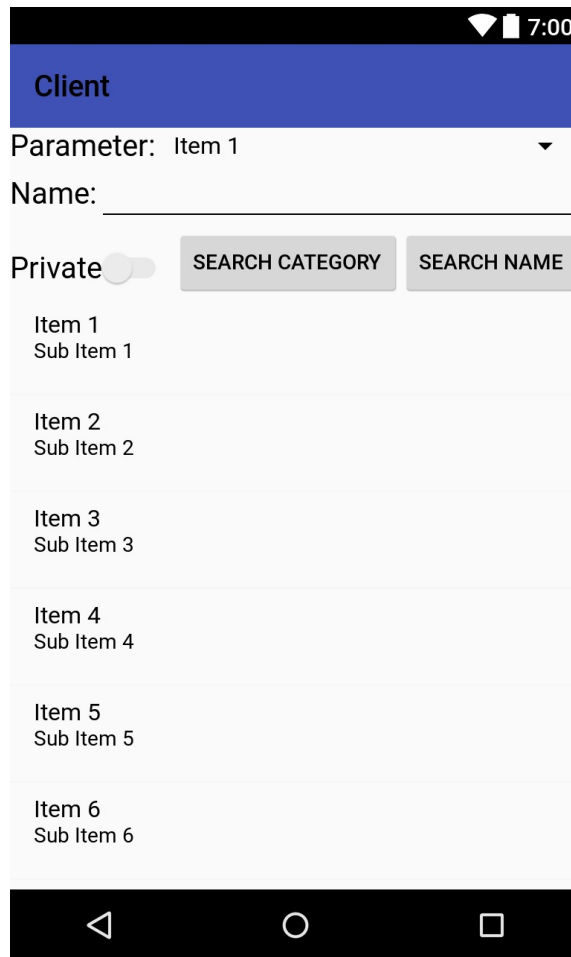


Figura 72: Vista de búsqueda de fotos

En esta ventana se implementa el caso de uso buscar foto, el cual nos permite buscar foto no solo por parámetro sino también por el nombre de la foto que se ha proporcionado. Los resultados aparecerán en la lista de la parte inferior.

El primer botón nos permite buscar solo las fotos que tengan la categoría seleccionada mientras que el segundo nos permitirá buscar por el nombre que hemos seleccionado. Por último, podemos buscar solo entre nuestras fotos o entre todas las fotos seleccionando el botón de privado o no.

Los resultados obtenidos se mostrarán en la parte inferior, donde se mostrará el nombre de la foto, el autor y la fecha y una entrada para ver el resto de opciones de la foto.

Ajustes de la cuenta

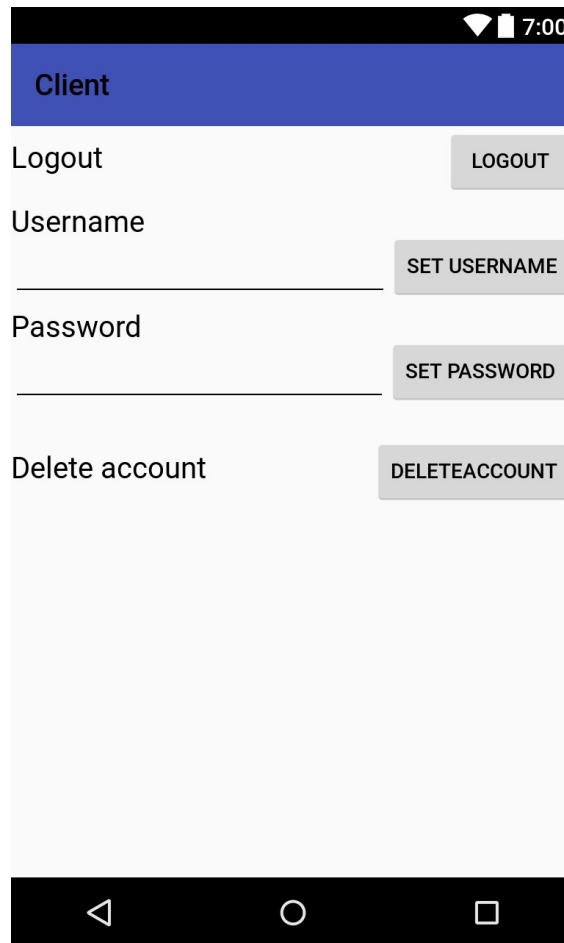


Figura 73: Vista de ajustes

En esta foto se implementan los casos de uso de modificar los credenciales asociados a la cuenta y cerrar sesión, se han desgajado en cuatro subapartados:

- El primero nos permite cerrar sesión, destruyendo la clave de sesión asociada y mostrando de nuevo la vista de inicio de sesión
- El segundo nos permite modificar el nombre del usuario, solo permitirá modificarlo si el nuevo nombre de usuario proporcionado no está en uso
- El tercero nos permite modificar la contraseña de acceso a la cuenta del usuario
- El último nos permite destruir la cuenta, destruirá todas las fotos y todos los resultados asociados a ellas.

- Ventana de detalles de la foto

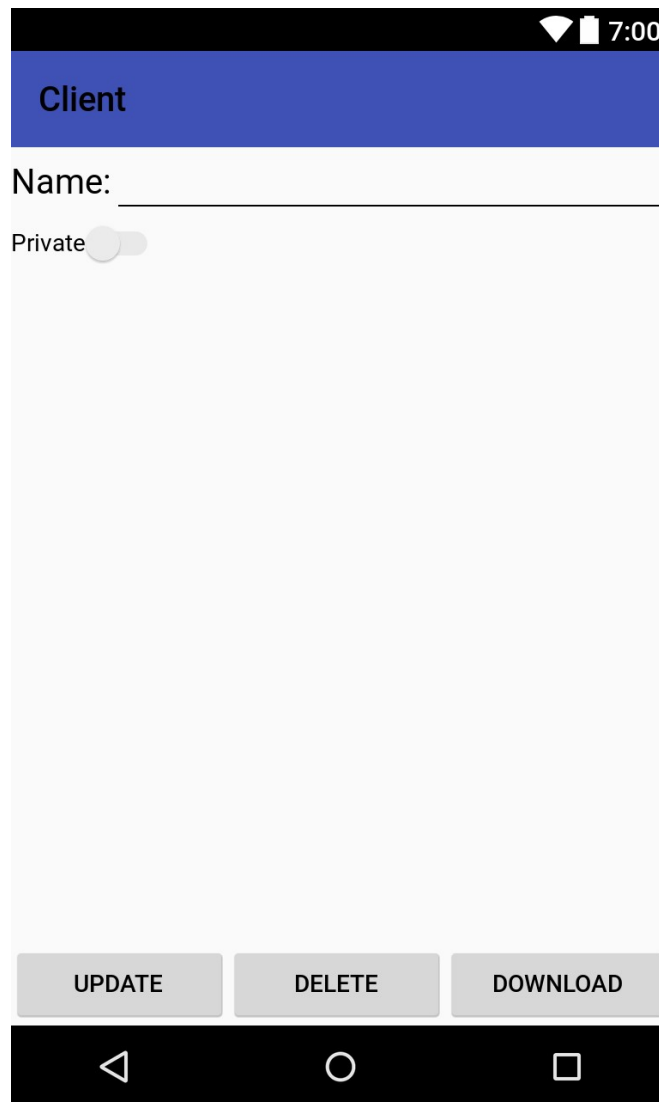


Figura 74: Vista de detalles de una foto

Esta es la última ventana, es la encargada de implementar los casos de uso de actualizar foto, borrar foto y descargar foto.

El primer botón nos permite modificar tanto el nombre como el estado de privacidad de la foto, el segundo nos permite destruir la foto de la aplicación y el tercero nos permite almacenar una copia de la foto en nuestro dispositivo.

Por último, cabe destacar que el los botones de actualizar y borrar solo se mostrarán en caso de ser nosotros el propietario de la foto, y que el nuevo nombre ha de ser único, si no nos dejara colocarlo como nuevo nombre de la foto

5.3. Contexto de las pruebas

En este caso, hemos utilizado únicamente pruebas de caja negra, dado el escaso tiempo que tenemos para la realización de las mismas y la corrección de los posibles fallos detectados.

Las pruebas realizadas han sido sobre el cliente final, dado que de esta manera podemos probar de manera conjunta las operaciones rest del servidor y la funcionalidad de las distintas ventanas de la aplicación.

5.4. Pruebas realizadas

Prueba 1	Cargar página inicial sin una sesión almacenada
Descripción	El usuario inicia la aplicación en el cliente sin tener ninguna clave de sesión almacenada en el dispositivo
Entrada	Ninguna
Resultado esperado	Se muestra la pantalla de inicio de sesión
Resultado obtenido	Se muestra la pantalla de inicio de sesión
¿Es correcto?	Resultado esperado

Figura 75: Prueba número 1

Prueba 2	Cargar página inicial con una sesión almacenada
Descripción	El usuario inicia la aplicación en el cliente teniendo ya una clave de sesión almacenada en el dispositivo
Entrada	Ninguna
Resultado esperado	Se muestra la pantalla principal de la aplicación, sin pasar por la pantalla de inicio de sesión
Resultado obtenido	Se muestra la pantalla de inicio en lugar de la ventana principal de la aplicación
Corrección	No se guardaba correctamente en Shared Preferences, reparando el método se consigue que esto funcione de manera adecuada

Figura 76: Prueba número 2

Prueba 3	Inicio de sesión con un usuario no creado
Descripción	El usuario solicita el inicio de sesión en la aplicación con unos credenciales que no han sido registrados en la base de datos
Entrada	Credenciales no registrados
Resultado esperado	Se comunica al usuario que esos credenciales no son correctos
Resultado obtenido	Se comunica al usuario que esos credenciales no son correctos
¿Es correcto?	Resultado esperado

Figura 77: Prueba número 3

Prueba 4	Inicio de sesión con un usuario creado, pero contraseña de acceso incorrecta
Descripción	El usuario solicita el inicio de sesión en la aplicación con unos credenciales que han sido registrados en la base de datos, pero la contraseña de acceso proporcionada no coincide con la registrada en la base de datos
Entrada	Credenciales registrados, pero contraseña inválida
Resultado esperado	Se comunica al usuario que esos credenciales no son correctos
Resultado obtenido	Se comunica al usuario que esos credenciales no son correctos
¿Es correcto?	Resultado esperado

Figura 78: Prueba número 4

Prueba 5	Inicio de sesión con un usuario válido
Descripción	El usuario solicita el inicio de sesión en la aplicación con unos credenciales correctos
Entrada	Credenciales válidos
Resultado esperado	Se muestra la pantalla principal de la aplicación y se almacena la clave de sesión en el dispositivo
Resultado obtenido	Se muestra la pantalla de inicio y la notificación de que los credenciales no son válidos
Corrección	Se trata de un problema de flujo del programa, tras reajustar la cadena del switch en función del código de la respuesta funciona de manera adecuada

Figura 78: Prueba número 5

Prueba 6	Registro de un nuevo usuario utilizando unos credenciales que ya han sido registrados
Descripción	El usuario la creación de un nuevo usuario proporcionando unos credenciales que ya han sido registrados
Entrada	Credenciales ya registrados
Resultado esperado	Se notifica al usuario que los credenciales proporcionados no se pueden registrar porque ya están en uso
Resultado obtenido	Se notifica al usuario que los credenciales proporcionados no se pueden registrar porque ya están en uso
¿Es correcto?	Resultado esperado

Figura 79: Prueba número 6

Prueba 7	Registro de un nuevo usuario utilizando unos credenciales que aun no han sido registrados
Descripción	El usuario solicita la creación de un nuevo usuario proporcionando unos credenciales que aun no han sido registrados por la aplicación
Entrada	Credenciales válidos
Resultado esperado	Se notifica al usuario que se ha creado un nuevo usuario y se almacenan los nuevos datos en la base de datos
Resultado obtenido	Se notifica al usuario que se ha creado un nuevo usuario y se almacenan los nuevos datos en la base de datos
¿Es correcto?	Resultado esperado

Figura 80: Prueba número 7

Prueba 8	Subir una foto cuando no se ha cargado aún en la aplicación
Descripción	El usuario solicita subir una nueva foto a la aplicación cuando aún no se ha cargado una foto en la aplicación
Entrada	Datos asociados a la nueva foto
Resultado esperado	La aplicación no realiza ninguna acción
Resultado obtenido	Se almacena una foto vacía en la aplicación
Corrección	Utilización de un booleano para asegurarse de que solo se sube una foto una vez que se ha cargado

Figura 81: Prueba número 8

Prueba 9	Subir una foto cuando se ha cargado una foto ya en la aplicación
Descripción	El usuario solicita subir una nueva foto a la aplicación una vez que ya se ha cargado en la vista principal
Entrada	Datos asociados a la nueva foto
Resultado esperado	La aplicación sube la foto a la CDN, la registra en la BD y notifica del éxito de la operación al usuario
Resultado obtenido	La aplicación sube la foto a la CDN, la registra en la BD y notifica del éxito de la operación al usuario
¿Es correcto?	Resultado esperado

Figura 82: Prueba número 9

Prueba 10	Cargar una foto en la aplicación desde la cámara del dispositivo
Descripción	El usuario solicita cargar una nueva foto a partir de la cámara del dispositivo que está utilizando
Entrada	Ninguna
Resultado esperado	Se abre la aplicación de la cámara del dispositivo, la foto tomada aparece cargada dentro de la aplicación del cliente
Resultado obtenido	Se abre la aplicación de la cámara del dispositivo, la foto tomada aparece cargada dentro de la aplicación del cliente
¿Es correcto?	Resultado esperado

Figura 83: Prueba número 10

Prueba 11	Cargar una foto en la aplicación desde la galería
Descripción	El usuario solicita cargar una nueva foto a partir de las fotos que tiene almacenadas en la galería del dispositivo que está utilizando
Entrada	Ninguna
Resultado esperado	Se abre la aplicación de selección de fotos de la galería, la foto seleccionada aparece cargada dentro de la aplicación del cliente
Resultado obtenido	Se abre la aplicación de selección de fotos de la galería, la foto seleccionada aparece cargada dentro de la aplicación del cliente
¿Es correcto?	Resultado esperado

Figura 84: Prueba número 11

Prueba 12	Ver mis fotos cuando no hay fotos propias subidas ya en la aplicación
Descripción	El usuario solicita ver las fotos que tiene subidas cuando aún no tiene cargada en la aplicación ninguna foto
Entrada	Ninguna
Resultado esperado	Se muestra la vista de mis fotos almacenadas con ninguna entrada apareciendo en ella
Resultado obtenido	Se muestra la lista por defecto de Android en lugar de una lista vacía
¿Es correcto?	Modificar el adaptador por defecto para que muestre una lista vacía

Figura 85: Prueba número 12

Prueba 13	Ver mis fotos cuando hay fotos propias subidas ya en la aplicación
Descripción	El usuario solicita ver las fotos que tiene subidas teniendo al menos una foto ya subida a la aplicación
Entrada	Ninguna
Resultado esperado	Se muestra la vista de mis fotos almacenadas mostrando en sus entradas las distintas fotos que el usuario tiene almacenadas
Resultado obtenido	Se muestra la vista de mis fotos almacenadas mostrando en sus entradas las distintas fotos que el usuario tiene almacenadas
¿Es correcto?	Resultado esperado

Figura 86: Prueba número 13

Prueba 14	Buscar fotos por categoría cuando no se tiene acceso a ninguna foto con los parámetros dados
Descripción	El usuario busca fotos seleccionando categoría pero no tiene acceso a ninguna de ellas con los parámetros proporcionados
Entrada	Parámetros de búsqueda
Resultado esperado	No se muestra ninguna foto en la lista de fotos
Resultado obtenido	No se muestra ninguna foto en la lista de fotos
¿Es correcto?	Resultado esperado

Figura 87: Prueba número 14

Prueba 15	Buscar fotos por categoría cuando hay fotos que concuerdan con la categoría y parámetros seleccionados
Descripción	El usuario busca fotos seleccionando categoría y tiene al menos una foto que concuerda con los parámetros seleccionados
Entrada	Parámetros de búsqueda
Resultado esperado	Se muestra en la vista las distintas fotos que se corresponden con los parámetros seleccionados
Resultado obtenido	Se muestra en la vista las distintas fotos que se corresponden con los parámetros seleccionados
¿Es correcto?	Resultado esperado

Figura 88: Prueba número 15

Prueba 16	Buscar fotos por nombre cuando no se tiene acceso a ninguna foto con los parámetros dados
Descripción	El usuario busca fotos seleccionando categoría pero no tiene acceso a ninguna de ellas con los parámetros proporcionados
Entrada	Parámetros de búsqueda
Resultado esperado	No se muestra ninguna foto en la lista de fotos
Resultado obtenido	No se muestra ninguna foto en la lista de fotos
¿Es correcto?	Resultado esperado

Figura 89: Prueba número 16

Prueba 17	Buscar fotos por nombre cuando hay fotos que concuerdan con la categoría y parámetros seleccionados
Descripción	El usuario busca fotos seleccionando nombre y tiene al menos una foto que concuerda con los parámetros seleccionados
Entrada	Parámetros de búsqueda
Resultado esperado	Se muestra en la vista las distintas fotos que se corresponden con los parámetros seleccionados
Resultado obtenido	No se muestran todas las fotos que concuerdan con los parámetros seleccionados
Corrección	El problema es la sentencia SQL utilizada en el servidor, modificándola conseguimos que funcione correctamente

Figura 90: Prueba número 17

Prueba 18	Descargar una foto
Descripción	El usuario descarga la foto que está viendo actualmente en la vista de visualización de fotos
Entrada	Ninguno
Resultado esperado	La foto es almacenada en la galería del dispositivo actual y se notifica al usuario actual del éxito de la operación
Resultado obtenido	La foto es almacenada en la galería del dispositivo actual y se notifica al usuario actual del éxito de la operación
¿Es correcto?	Resultado esperado

Figura 91: Prueba número 18

Prueba 19	Modificar los parámetros de una foto
Descripción	El usuario solicita modificar los parámetros de una foto con los actuales almacenados en la aplicación
Entrada	Parámetros nuevos
Resultado esperado	Se almacenan los nuevos parámetros de la foto seleccionada y se notifica al usuario del éxito de la operación
Resultado obtenido	Se almacenan los nuevos parámetros de la foto seleccionada y se notifica al usuario del éxito de la operación
¿Es correcto?	Resultado esperado

Figura 92: Prueba número 19

Prueba 20	Borrar una foto
Descripción	El usuario solicita borrar la foto que esta visualizando actualmente de la aplicación
Entrada	Ninguno
Resultado esperado	La foto es eliminada de la aplicación y se notifica al usuario el éxito de la operación
Resultado obtenido	La foto es eliminada de la aplicación y se notifica al usuario el éxito de la operación
¿Es correcto?	Resultado esperado

Figura 93: Prueba número 20

Prueba 21	Cerrar sesión
Descripción	El usuario solicita cerrar la sesión que tiene abierta actualmente en el dispositivo que está utilizando
Entrada	Ninguno
Resultado esperado	Se elimina la sesión actual del dispositivo y se muestra la pantalla de inicio de sesión
Resultado obtenido	Se elimina la sesión actual del dispositivo y se muestra la pantalla de inicio de sesión
¿Es correcto?	Resultado esperado

Figura 94: Prueba número 21

Prueba 22	Modificar el nombre de usuario cuando el nombre de usuario nuevo proporcionado ya está en uso
Descripción	El usuario solicita modificar su nombre de usuario y proporciona uno nuevo, pero ese nuevo nombre de usuario ya está en uso por otro miembro de la aplicación
Entrada	Nombre de usuario
Resultado esperado	El nombre de usuario no es modificado, se notifica al usuario que dicho nombre ya está siendo utilizado por otro miembro de la aplicación
Resultado obtenido	El nombre de usuario no es modificado, se notifica al usuario que dicho nombre ya está siendo utilizado por otro miembro de la aplicación
¿Es correcto?	Resultado esperado

Figura 95: Prueba número 22

Prueba 23	Modificar el nombre de usuario cuando el nombre de usuario nuevo proporcionado está disponible
Descripción	El usuario solicita modificar su nombre de usuario, proporciona uno nuevo y dicho nombre está disponible para su uso
Entrada	Nombre de usuario
Resultado esperado	El nombre de usuario es modificado, se notifica al usuario el éxito en la operación
Resultado obtenido	El nombre de usuario es modificado, se notifica al usuario el éxito en la operación
¿Es correcto?	Resultado esperado

Figura 94: Prueba número 23

Prueba 24	Modificar la contraseña del usuario que tiene abierta la sesión el dispositivo
Descripción	El usuario solicita modificar la contraseña del usuario que tiene abierta la sesión
Entrada	Nueva contraseña
Resultado esperado	La nueva contraseña es almacenada como contraseña del usuario, se notifica al usuario el éxito en la operación
Resultado obtenido	La nueva contraseña es almacenada como contraseña del usuario, se notifica al usuario el éxito en la operación
¿Es correcto?	Resultado esperado

Figura 95: Prueba número 24

Prueba 25	Borrar un usuario
Descripción	El usuario solicita destruir el usuario que tiene abierto en el dispositivo actual
Entrada	Ninguno
Resultado esperado	El usuario es eliminado de la aplicación y se muestra la pantalla de inicio de sesión
Resultado obtenido	El usuario es eliminado de la aplicación y se muestra la pantalla de inicio de sesión
¿Es correcto?	Resultado esperado

Figura 96: Prueba número 25

6. Conclusiones

Tras la realización del proyecto hemos obtenido una aplicación funcional que responde a toda una serie de requisitos especificados anteriormente. Esta aplicación permite que los usuarios almacenen y compartan sus fotos de manera sencilla desde su dispositivo móvil. A su vez, esta aplicación hace uso de IBM Cloud para el despliegue de la aplicación y de IBM Watson para el análisis de las fotos, dos de los puntos clave del desarrollo de este trabajo.

En cuanto a la funcionalidad implementada, se logró la consecución de la totalidad de los casos de uso, de los cuales la mayor parte de la funcionalidad de dichos casos de uso se ha concluido con éxito.

En cuanto a los plazos, se ha logrado que la planificación previa se ajuste en gran medida a la planificación preestablecida. Afortunadamente, no ha tenido lugar ninguno de los riesgos predefinidos y a pesar de que algunos de las tareas individuales si se han desviado de lo planificado, la desviación total del proyecto no es elevada.

Finalmente, podemos concluir que la aplicación diseñada y la implementada son similares, hemos logrado que los distintos puntos diseñados e implementados sean similares, y las planificaciones, los riesgos tenidos en cuenta, y el resto de decisiones de planificación y diseño nos han permitido ajustarnos a los plazos y que el resultado obtenido sea similar al esperado.

Trabajo futuro

La aplicación final obtenida responde a los requisitos proporcionados, no obstante, a continuación detallaremos algunas inclusiones que podrían ser interesantes para mejorar la aplicación:

- Inclusión de un súper usuario, para labores de mantenimiento y gestión de la aplicación
- Inclusión de jerarquías en el mecanismo de clasificación de las fotos
- Sustitución de la base de datos relacional por una no relacional
- Limitaciones a las fotos que puede subir el usuario
- Inclusión de algún tipo de monetización para el mantenimiento de la aplicación
- Cambio estético para que sea mucho más atractiva

Bibliografía

[1] IBM Redbooks ,Building Getting Started With IBM Bluemix, 2015

[2] IBM Redbooks ,Building Cognitive Applications with IBM Watson Services: Volume 1 Getting Started, 2017

[3] IBM Redbooks, Building Cognitive Applications with IBM Watson Services: Volume 3 Visual Recognition, 2017

[4] IBM Redbooks, Developing Node.js Applications on IBM Cloud, 2017

[5] IBM Redbooks, Essentials of Application Development on IBM Cloud, 2017

[6] Api de NodeJs, visto por última vez 15/05/2018

<https://nodejs.org/api/index.html>

[7] Api de Android, visto por última vez 03/06/2018

<https://developer.android.com/>

[8] Api de Java 1.8, visto por última vez 01/06/2018

<https://docs.oracle.com/javase/8/docs/api/>

[9] Tutorial de SQL de W3SCHOOLS, visto por última vez 15/05/2018

<https://www.w3schools.com/sql/default.asp>

[10] Listado de módulos de npm, visto por última vez 12/05/2018

<https://www.npmjs.com/>

[11] Manual de usuario de Sequelize, visto por última vez 15/05/2018

<http://docs.sequelizejs.com/>

[12] Api de Retrofit2, visto por última vez 28/04/2018

<http://square.github.io/retrofit/>

Apéndices

Manual de despliegue

A continuación explicamos como se ha de desplegar la aplicación:

1. Primeramente, creamos el servicio de analisis de imagenes en IBM Cloud y el servicio de la base de datos, guardamos sus credenciales

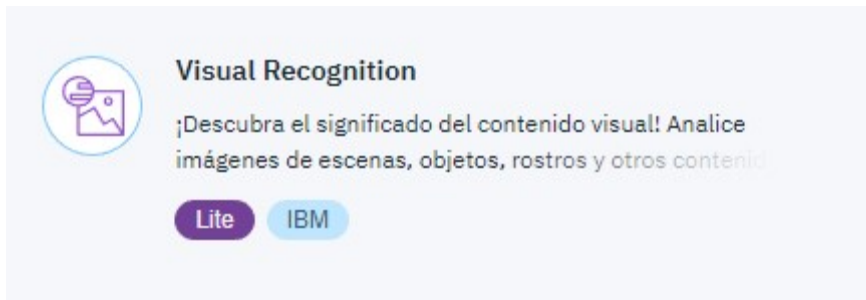


Figura 97: Icono del servicio de analisis de imagenes

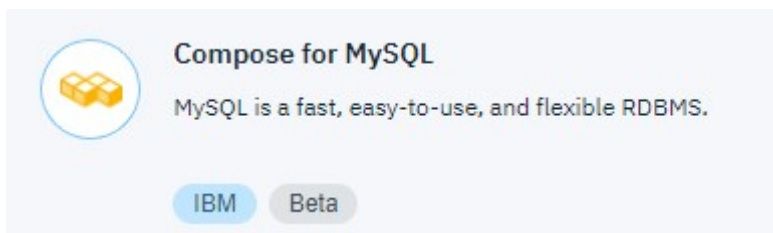


Figura 98: Icono del servicio de base de datos

2. A continuación colocamos los datos de los servicios en nuestro programa NodeJs, los datos de la base de datos se colocan en el modulo sequelize, los datos del analizador de imagenes se colocan en el módulo de analisis de imagenes y los de el servicio Clouinary se colocan en el módulo de las fotos, en la operación getCDNCredentials.

```
4 EDITORES ABIERTOS
  {} package.json.. M
  JS Sequelize.js a... U
  ! manifest.y... M
4 TFGFINAL
  api-node
  models
  JS Deteccion.js U
  JS Foto.js U
  JS Models.js U
  JS Sequelize.js U
  JS User.js U
1  const Sequelize = require('sequelize');
2  const sequelize = new Sequelize('prueba1', 'admin', 'prueba1234', {
3
4    host: 'sl-us-south-1-portal.14.dblayer.com',
5    dialect: 'mysql',
6    port : 28507,
7    logging:false,
8    operatorsAliases:false,
9    pool: {
10     max: 5,
11     min: 0,
12     idle: 10000
13   });
14  sequelize.authenticate().then()
```

Figura 99: Ubicación donde colocar los datos de acceso a la base de datos

JS Deteccion.js	U	9	var visual_recognition = new watson.VisualRecognitionV3({
JS Foto.js	U	10	api_key: 'cf86a1489cffd897126796049063f903f2ffaec5',
JS Models.js	U	11	version: 'v3',
JS Sequelize.js	U	12	version_date: '2016-05-20'
JS User.js	U	13	});
node_modules		14	
routes		15	router.post('/uploadImage', function (req, res) {
JS fotos.js	U	16	models.User.findOne({where:{token:req.headers.token}})
JS login.js	U	17	.then((user)=>{
JS user.js	U	18	models.Foto.create({
JS visualRecog...	U	19	id:req.headers.id,
		20	url: req.headers.url,
		21	});

Figura 100: Ubicación donde colocar los datos de acceso al servicio Watson

JS User.js	U	146	
node_modules		147	router.get('/cdnCredentials', function(req, res){
routes		148	var credentials={
JS fotos.js	U	149	"cloud_name": "ddzt3hsec",
JS login.js	U	150	"api_key": "998911677189156",
JS user.js	U	151	"api_secret": "E0MUX1gK3HzlD-sKAE7qFgbMMY",
JS visualRecog...	U	152	"notification_url": "https://requestb.in/12345abcd"};
views		153	res.send(credentials);
		154	});
		155	});

Figura 101: Ubicación donde colocar los datos de acceso a clouinary

3. Posteriormente procedemos a subir la instancia a IBM Cloud, para ello necesitamos una máquina con linux y descargar el módulo de ibm cloud. Una vez hecho esto ejecutamos los siguientes comandos:

Este primer comando nos pedirá los credenciales de acceso, una vez validados ejecutamos el siguiente comando.

- Este primer comando nos pedirá los credenciales de acceso, una vez validados ejecutamos el siguiente comando, en función de donde queramos desplegar la aplicación elegiremos una de las cuatro opciones.

- US South and US East

```
$ ibmcloud login -a api.ng.bluemix.net
```

- Sydney and AP North

```
$ ibmcloud login -a api.au-syd.bluemix.net
```

- Germany

```
$ ibmcloud login -a api.eu-de.bluemix.net
```

- United Kingdom

```
$ ibmcloud login -a api.eu-gb.bluemix.net
```

Figura 102: Lista de posible comandos de login para IBM Cloud

- Finalmente situados en la carpeta donde tenemos la aplicación ejecutamos el siguiente comando, que creará la instancia de la aplicación y nos devolverá los datos de la misma, los anotamos.

```
ibmcloud cf push
```

4. Por último, colocamos la url a nuestra aplicación en la aplicación Android en como se puede ver en la figura 101, con ello compilamos la aplicación y la instalamos en los dispositivos que deseemos, requiere que al menos el dispositivo tenga la Api 26 de Android, o no será posible instalarlo



Figura 103: Ubicación donde colocar la url a nuestra aplicación

Manual de usuario

Una vez que la aplicación ha sido desplegada con éxito y tenemos en nuestro dispositivo móvil la aplicación instalada podemos arrancarla, al ejecutarla primeramente nos muestra la pantalla de inicio, donde podemos iniciar sesión o crear un nuevo usuario.

Una vez iniciada la sesión se nos mostrará la ventana principal donde podremos navegar entre las distintas subventanas, en cada una de las cuales se implementa los diversos casos.

Todas y cada una de las ventanas de la aplicación móvil están explicadas con detenimiento 5.2, en el subapartado del cliente Android.