Especificación de Requerimientos de Software

para

Aplicación Social Network

Versión 1.0

Hecho por José Alberto Espinoza, Jethran Enrique Gómez

Universidad Veracruzana

20/10/2019

Table of Contents

Table of Contents ii

Historial de Revisión iii

1. Introducción 1

1.1 Propósito 1

1.2 Convenciones del Documento 1

1.3 Alcance del proyecto 1

2. Descripción General 2

2.1 Perspectiva del producto 2

2.2 Clases de usuarios y sus características 2

2.3 Ambiente de operación 2

2.4 Restricciones de implementación y de diseño 2

3. Características del Sistema 3

3.1 Modelo de dominio 3

3.2 Diagramas de casos de uso 4

3.2.1 Diagrama de paquetes de casos de uso 4

3.2.2 Diagrama de casos de uso por actor - Moderador 4

3.2.3 Diagrama de casos de uso por actor – Usuario 5

3.3 Casos de uso de Usuario 6

3.3.1 Registro de usuario 6

3.3.2 Autenticación de usuario 7

3.3.3 Modificar perfil de usuario 8

3.3.4 Subir foto 10

3.3.5 Comentar foto 11

3.3.6 Reaccionar a foto 11

3.3.7 Eliminar foto 12

3.3.8 Guardar foto en favoritos 13

3.3.9 Consultar fotos de favoritos 14

3.3.10 Chat personal 14

3.3.11 Chat grupal 16

3.3.12 Enviar mensaje 17

3.3.13 Eliminar mensaje 18

3.3.14 Editar mensaje 19

3.3.15 Buscar usuario 20

3.3.16 Ver información de usuario 21

3.3.17 Revisar solicitudes 22

3.3.18 Ver feed de fotos 23

3.3.19 Consultar chats recientes 23

3.4 Casos de uso de moderador 24

3.4.1 Revisar contenido 24

3.4.2 Eliminar contenido 25

3.4.3 Consultar cuentas 26

3.4.4 Suspender cuenta 26

3.4.5 Activar cuenta 27

4. Características del Servidor 28

4.1 Diagrama de casos de uso del Servidor RestAPI 28

4.1.1 Registrar nuevo usuario 28

4.1.2 Guardar foto de usuario 29

4.1.3 Guardar comentario de foto 29

4.1.4 Eliminar foto de usuario 30

4.1.5 Guardar foto en favoritos de usuario 30

4.1.6 Consultar fotos de favoritos de usuario 31

4.1.7 Crear nuevo chat personal 31

4.1.8 Crear nuevo chat grupal 32

4.1.9 Registrar mensaje 33

4.1.10 Modificar mensaje 33

4.1.11 Consultar usuarios 33

4.1.12 Consultar información de usuario 34

4.1.13 Registrar solicitud de amistad a usuario 35

4.1.14 Modificar solicitud 36

4.1.15 Consultar solicitudes 37

4.1.16 Consultar feed de fotos 38

4.1.17 Revisar contenido 39

4.1.18 Eliminar contenido 39

4.1.19 Consultar cuentas 40

4.1.20 Suspender cuenta 40

4.1.21 Activar cuenta 41

4.1.22 Registrar nuevo moderador 42

4.1.23 Autenticación de usuario 42

5. Requerimientos de datos 43

5.1 Modelo de datos relacional 43

5.2 Diccionario de datos 43

5.3 Reportes 44

5.4 Data Acquisition, Integrity, Retention, and Disposal 44

6. External Interface Requirements 44

6.1 User Interfaces 44

6.2 Software Interfaces 44

6.3 Hardware Interfaces 45

6.4 Communications Interfaces 45

7. Quality Attributes 45

7.1 Usability 45

7.2 Performance 45

7.3 Security 45

7.4 Safety 45

7.5 [Others as relevant] 46

8. Internationalization and Localization Requirements 46

9. Other Requirements 46

Appendix A: Glossary 46

Appendix B: Analysis Models 46

Historial de Revisión

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre** | **Fecha** | **Razón de cambios** | **Versión** |
| Creación | 21/10/2019 | Primera revisión | 1.0 |
|  |  |  |  |

# Introducción

La presente Especificación de Requerimientos de Software (ERS) tiene una estructura distribuida de la siguiente manera: En la parte introductoria del documento se localiza el propósito de este, junto a detalles para entender mejor la redacción de la ERS. En el cuerpo del documento (Descripción general, Características del sistema, Requerimientos de datos, Requerimientos de interfaz externa y Atributos de calidad) se localizan las características esenciales acerca de la funcionalidad y modo de operación del producto de software. Por último, en los Apéndices del documento se encontrarán modelos de análisis que permitirán identificar con mayor facilidad los atributos de calidad y requerimientos funcionales, así como la relación entre otros aspectos del producto de software.

## Propósito

En está ERS daremos una descripción detallada de los requerimientos funcionales, atributos de calidad, y modelo de datos de la aplicación “Social Network”, este documento esta enfocado a ser utilizado por desarrolladores para conocer la estructura estática del proyecto, el comportamiento y las relaciones de datos que existen entre ellos. Para los encargados lideres de proyecto con la intención de conocer las restricciones, estructura de datos y del sistema, comportamiento de este y las consideraciones de atributos de calidad necesarios para alcanzar el nivel de calidad esperado.

## Convenciones del Documento

ERS: Especificación de requerimientos de software

FN: Flujo normal

FA: Flujo alterno

EX: Excepciones

Sistema: Se refiere a la Aplicación “Social Network”

BD: Hace referencia la Base de datos

## Alcance del proyecto

La aplicación “Social Network” tiene como propósito ser una red social en las cual múltiples usuarios podrán registrar y posteriormente ingresar para así subir publicaciones con fotos, al mismo tiempo incluir un servicio de mensajería instantánea para comunicación entre dos personas o un grupo de personas.

# Descripción General

## Perspectiva del producto

Primera versión de la Aplicación “Social Network”, una nueva alternativa a otras redes sociales existentes. Tenemos que tener en cuenta que el presente Sistema presenta una arquitectura de Distribuida

## Clases de usuarios y sus características

* Usuario de la aplicación. (A excepción del usuario moderador):

Hablamos de cualquier usuario capaz de acceder a la aplicación, esté podrá tener acceso a las funcionalidades principales del sistema tales como, manejar fotos, agregar otros usuarios a su lista de amigos, enviar mensajes personales y grupales.

* Moderador de la aplicación.

Encargado de la gestión de la aplicación y de sus Usuarios con la capacidad de suspenderlos y de revisar el contenido subido a la aplicación

## Ambiente de operación

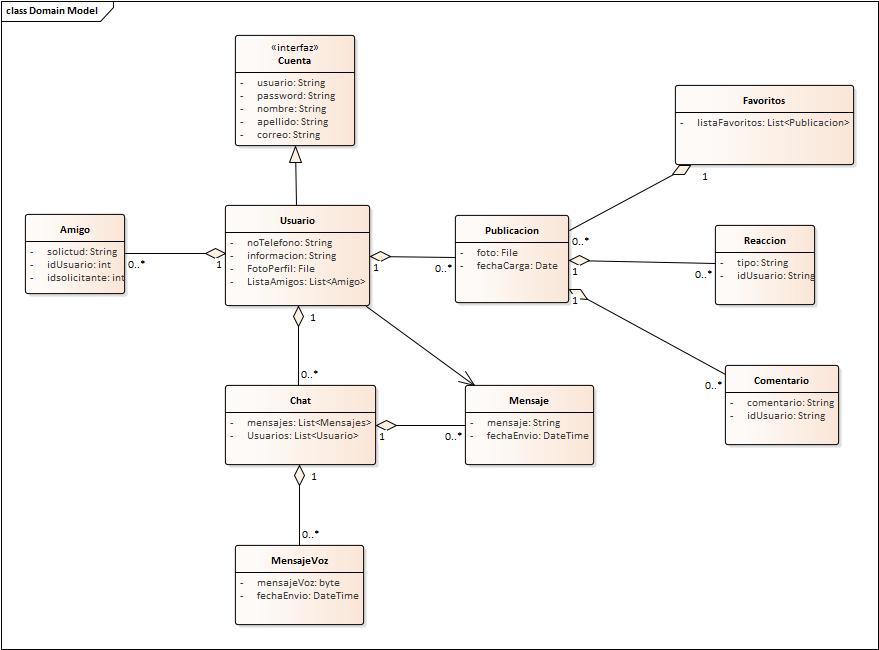
Lo clientes de la aplicación serán instaladas en dispositivos moviles Android con una versión superior a Nougat 7.0

## Restricciones de implementación y de diseño

* La aplicación funcionara mediante una arquitectura de sistemas distribuidos.
* Implementara estrategias de integración continua.
* Estará implementado un Cliente de Usuario, Cliente de Moderador, Servidor de fotos y un Servidor gRPC
* La mensajería instantánea contara una cola de mensajes sin conexión para ser enviados en cuanto la conexión a internet se recupere.
* Se ocupará la metodología ICONIX para el diseño de la aplicación.

# Características del Sistema

## Modelo de dominio



## Diagramas de casos de uso

### Diagrama de paquetes de casos de uso

Imagen que contiene captura de pantalla

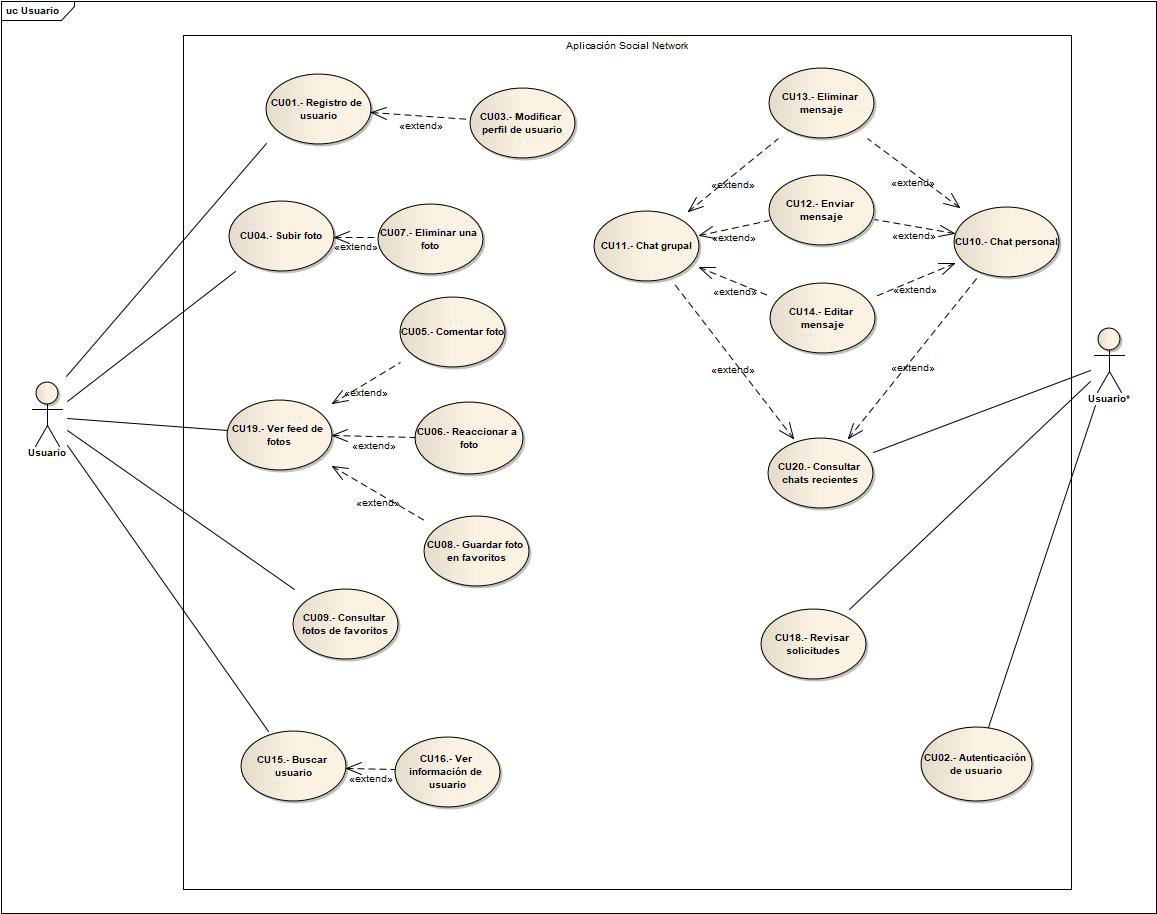
Descripción generada automáticamente

### Diagrama de casos de uso por actor - Moderador

Imagen que contiene texto, mapa

Descripción generada automáticamente

### Diagrama de casos de uso por actor – Usuario



## Casos de uso de Usuario

### Registro de usuario

#### Descripción y flujos

|  |  |
| --- | --- |
| **­­ID:** | CU01 |
| **Nombre:** | Registro de cuenta usuario |
| **Autor(es):** | José Alberto Espinoza Castillo |
| **Fecha de creación:** | 18/10/2019 |
| **Fecha de actualización:** | 20/10/2019 – Se modificaron algunas entidades |
| **Actor(es):** | Usuario sin cuenta. En adelante será referenciado como Usuario para este CU. |
| **Descripción:** | Permitirá añadir un nuevo evento al sistema |
| **Precondiciones:** | Pre01.- |
| **Disparador:** | El Usuario hace clic en “Registrarme” en la ventana **GUILogin**. |
| **Flujo Normal:** | 1. El Sistema despliega **GUIRegistroCuenta** con un formulario con los siguiente campos:    1. Nombre(s)    2. Apellido(s)    3. Nombre de Usuario    4. Correo Electrónico    5. Numero de celular    6. Contraseña    7. Repetir Contraseña    8. Fecha de nacimiento con DatePicker 2. El Usuario termina de responder el formulario y hace clic en “Terminar”. 3. El Sistema verifica que los campos estén completos y valida la información requerida. 4. El Sistema registra la nueva CUENTA de usuario y envía un correo de verificación. 5. Extiende al “CU0 .- Validar código de verificación” y termina caso de uso |
| **Flujos Alternos:** | **FA-1. Existen campos obligatorios vacíos.**   1. El Sistema despliega un popup con el mensaje: “Por favor rellene todos los campos” 2. Continua en el paso 2 del flujo normal.   **FA-2. El numero de celular ingresado no coincide con el formato de 10 digitos**   1. El Sistema resalta en rojo el campo del celular con mensaje debajo: “El numero de celular debe ser de 10 números”. 2. Continua en el paso 2 del flujo normal.   **FA-3. El correo electrónico ingresado no coincide con el formato estándar de un correo.**   1. El Sistema resalta en rojo el campo de correo electrónico con un mensaje debajo: “El correo ingresado no es valido”. 2. Continua en el paso 2 del flujo normal |
| **Excepciones:** | **Error 500: No se ha podido registrar la CUENTA en el Sistema.**   1. El Sistema muestra un popup con el mensaje “Ocurrió un error en el servidor. Inténtelo mas tarde”. 2. Termina caso de uso |
| **Postcondiciones:** | PosCO01 – Si el flujo concluyo normalmente debe haber una CUENTA registrada. |
| **Prioridad:** | Media |

#### Diagrama de robustez

### Autenticación de usuario

#### Descripción y flujos

|  |  |
| --- | --- |
| **­­ID:** | CU02 |
| **Nombre:** | Autenticación de usuario |
| **Autor(es):** | José Alberto Espinoza Castillo |
| **Fecha de creación:** | 18/10/2019 |
| **Fecha de actualización:** | 20/10/2019 – Se modificaron algunas entidades |
| **Actor(es):** | Usuario |
| **Descripción:** | Permitirá al usuario acceder al Sistema |
| **Precondiciones:** | Pre01.- No debe hCUaber una cuenta abierta al momento de la autenticación |
| **Disparador:** | El Usuario accede al Sistema |
| **Flujo Normal:** | 1. El Sistema despliega **GUILogin** con los siguientes componentes:    1. Un formulario que requiere los siguientes campos:       1. Correo electrónico       2. Contraseña 2. El Usuario completa los campos solicitados y hace clic en “Iniciar sesión” 3. El Sistema verifica que los campos estén completos y valida la información requerida. 4. El Sistema consulta las CUENTAS registradas usando la combinación de correo electrónico y contraseña ingresados por el Usuario. 5. El Sistema asigna al Usuario un token y lo redirige a **GUIFeedPrincipal** |
| **Flujos Alternos:** | **FA-1. Existen campos obligatorios vacíos.**   1. El Sistema despliega un pop-up con el mensaje: “Por favor rellene todos los campos” 2. Continua en el paso 2 del flujo normal.   **FA-2. El correo electrónico ingresado no coincide con el formato estándar de un correo.**   1. El Sistema resalta en rojo el campo de correo electrónico con un mensaje debajo: “El correo ingresado no es valido”. 2. Continua en el paso 2 del flujo normal.   **FA-3. El correo ingresado no está asociado con ninguna cuenta**   1. El Sistema despliega un pop-up con el mensaje: “El correo ingresado no está asociado con ninguna cuenta” 2. Continua en el paso 2 del flujo normal.   **FA-4. La contraseña ingresada es incorrecta.**   1. El Sistema despliega un pop-up con el mensaje: “La contraseña ingresada es incorrecta” 2. Continua en el paso 2 del flujo normal   **FA-5. Usuario sin verificación de correo.**   1. Extiende al “CU0 .- Validar código de verificación” y termina caso de uso |
| **Excepciones:** | **Error 500: No se ha podido acceder a las cuentas del Sistema.**   1. El Sistema muestra un pop-up con el mensaje “Ocurrió un error en el servidor. Inténtelo mas tarde”. 2. Termina caso de uso |
| **Postcondiciones:** | PosCO01 – Si el flujo concluyo normalmente el usuario deberá ser redirigido a “GUIFeedPrincipal” |
| **Prioridad:** | Alta |

#### Diagrama de robustez

### Modificar perfil de usuario

#### Descripción y flujos

|  |  |
| --- | --- |
| **­­ID:** | CU03 |
| **Nombre:** | Modificar perfil de usuario |
| **Autor(es):** | José Alberto Espinoza Castillo |
| **Fecha de creación:** | 18/10/2019 |
| **Fecha de actualización:** | 20/10/2019- Se añadieron flujos alternos |
| **Actor(es):** | Usuario |
| **Descripción:** | Permitirá al Usuario modificar la información de su cuenta |
| **Precondiciones:** | Pre01.- Debe haber una cuenta de usuario activa. |
| **Disparador:** | El Usuario hace clic en “Información de cuenta” del menú desplegable principal. |
| **Flujo Normal:** | 1. El Sistema despliega “GUILogin” con los siguientes componentes:    1. Un formulario que requiere los siguientes campos:       1. Correo electrónico       2. Contraseña |
| **Flujos Alternos:** | **FA-1. Existen campos obligatorios vacíos.**   1. El Sistema despliega un pop-up con el mensaje: “Por favor rellene todos los campos” 2. Continua en el paso 2 del flujo normal.   **FA-2. El correo electrónico ingresado no coincide con el formato estándar de un correo.**   1. El Sistema resalta en rojo el campo de correo electrónico con un mensaje debajo: “El correo ingresado no es valido”. 2. Continua en el paso 2 del flujo normal.   **FA-3. El correo ingresado no está asociado con ninguna cuenta**   1. El Sistema despliega un pop-up con el mensaje: “El correo ingresado no está asociado con ninguna cuenta” 2. Continua en el paso 2 del flujo normal.   **FA-4. La contraseña ingresada es incorrecta.**   1. El Sistema despliega un pop-up con el mensaje: “La contraseña ingresada es incorrecta” 2. Continua en el paso 2 del flujo normal   **FA-5. Usuario sin verificación de correo.**   1. Extiende al “CU0 .- Validar código de verificación” y termina caso de uso |
| **Excepciones:** | **Error 500: No se ha podido acceder a las cuentas del Sistema.**   1. El Sistema muestra un pop-up con el mensaje “Ocurrió un error en el servidor. Inténtelo mas tarde”. 2. Termina caso de uso |
| **Postcondiciones:** | PosCO01 – Si el flujo concluyo normalmente el usuario deberá ser redirigido a “GUIFeedPrincipal” |
| **Prioridad:** | Alta |

#### Diagrama de robustez

### Subir foto

#### Descripción y flujos

|  |  |
| --- | --- |
| **­­ID:** | CU04 |
| **Nombre:** | Subir foto |
| **Autor(es):** | Ulises González Robles |
| **Fecha de creación:** | 21/10/2019 |
| **Fecha de actualización:** | Fecha en la que se modifica (opcional) |
| **Actor(es):** | Usuario |
| **Descripción:** | El usuario podrá subir a la aplicación una fotografía desde su galería o tomar una foto en ese momento con la cámara. |
| **Precondiciones:** | Tener una aplicación de cámara, o tener una foto en la galería. Previamente haber iniciado la aplicación. |
| **Flujo Normal:** | 1.- El usuario selecciona el ícono de la cámara en la aplicación.  2.- El sistema abre la aplicación de la cámara.  3.- El usuario toma la fotografía. [FA1]  4.- El sistema muestra una vista previa de la fotografía y pide la  confirmación del usuario para proceder a subirla. [FA2]  5.- El usuario confirma la subida de la fotografía  6.- El sistema sube la fotografía al servidor. [EX1] |
| **Flujos Alternos:** | FA1: 1.- El usuario cancela la toma de la fotografía.  2.- El sistema cierra la aplicación de la cámara y regresa a  la ventana principal.  FA2: 1.- El usuario selecciona cancelar.  2.- El sistema cancela la subida de la fotografía y regresa a  la ventana principal. |
| **Excepciones:** | EX1: 1.- El sistema no logra establecer una conexión con el  Servidor. |
| **Poscondiciones:** | La fotografía es publicada en el servidor y puede ser vista por el usuario. |
| **Entradas:** | El archivo de la fotografía. |
| **Salidas:** | EL archivo de la fotografía almacenado en el servidor.  El sistema despliega el mensaje: “Fotografía subida con éxito” |
| **Incluye:**  **(relación Include)** | Haga clic aquí para escribir texto. |
| **Extiende:**  **(relación Extend)** | Haga clic aquí para escribir texto. |
| **Prioridad:** | Elija un elemento. |

#### Diagrama de robustez

### Comentar foto

#### Descripción y flujos

|  |  |
| --- | --- |
| **­­ID:** | CU05 |
| **Nombre:** | Comentar foto |
| **Autor(es):** | Ulises González Robles |
| **Fecha de creación:** | 21/10/2019 |
| **Fecha de actualización:** | Fecha en la que se modifica (opcional) |
| **Actor(es):** | Usuario |
| **Descripción:** | El usuario podrá comentar una fotografía. |
| **Precondiciones:** | Debe existir una fotografía en el servidor. |
| **Flujo Normal:** | 1.- El usuario selecciona la opción comentar en la fotografía que  se desea comentar.  2.- El sistema abre un cuadro de texto para poder realizar el  comentario.  3.- El usuario escribe el comentario y selecciona publicar. [FA1]  4.- El sistema agrega el comentario a la fotografía deseada y lo  guarda en el servidor. [EX1] |
| **Flujos Alternos:** | FA1: 1.- El usuario cancela el comentario.  2.- El sistema eliminar el texto y cierra el cuadro de texto. |
| **Excepciones:** | EX1: 1.- El sistema no logra establecer una conexión con el  Servidor. |
| **Poscondiciones:** | El comentario es publicado en el servidor y puede ser visto por el usuario. |
| **Entradas:** | Texto del comentario. |
| **Salidas:** | EL archivo de la fotografía almacenado en el servidor con un comentario añadido. |
| **Incluye:**  **(relación Include)** | Haga clic aquí para escribir texto. |
| **Extiende:**  **(relación Extend)** | Haga clic aquí para escribir texto. |
| **Prioridad:** | Elija un elemento. |

#### Diagrama de robustez

### Reaccionar a foto

#### Descripción y flujos

|  |  |
| --- | --- |
| **­­ID:** | CU06 |
| **Nombre:** | Reaccionar a foto |
| **Autor(es):** | Ulises González Robles |
| **Fecha de creación:** | 21/10/2019 |
| **Fecha de actualización:** | Fecha en la que se modifica (opcional) |
| **Actor(es):** | Usuario |
| **Descripción:** | El usuario podrá reaccionar a una fotografía. |
| **Precondiciones:** | Debe existir una fotografía en el servidor. |
| **Flujo Normal:** | 1.- El usuario selecciona la opción reaccionar en la fotografía  que se desea comentar. [FA1]  2.- El sistema agrega la reacción a la fotografía deseada y lo  guarda en el servidor. [EX1] |
| **Flujos Alternos:** | FA1: 1.- El usuario selecciona reaccionar a una fotografía en la  que ya había reaccionado previamente.  2.- El sistema eliminar la reacción de la fotografía. |
| **Excepciones:** | EX1: 1.- El sistema no logra establecer una conexión con el  Servidor. |
| **Poscondiciones:** | La reacción es publicada en el servidor y puede ser vista por el usuario. |
| **Entradas:** | Incremento en el contador de reacciones de una fotografía. |
| **Salidas:** | Contador de reacciones en la fotografía incrementado en una unidad. |
| **Incluye:**  **(relación Include)** | Haga clic aquí para escribir texto. |
| **Extiende:**  **(relación Extend)** | Haga clic aquí para escribir texto. |
| **Prioridad:** | Elija un elemento. |

#### Diagrama de robustez

### Eliminar foto

#### Descripción y flujos

|  |  |
| --- | --- |
| **­­ID:** | CU07 |
| **Nombre:** | Eliminar una foto |
| **Autor(es):** | Ulises González Robles |
| **Fecha de creación:** | 21/10/2019 |
| **Fecha de actualización:** | Fecha en la que se modifica (opcional) |
| **Actor(es):** | Usuario |
| **Descripción:** | El usuario eliminar una fotografía de su cuenta. |
| **Precondiciones:** | Debe existir una fotografía en el servidor. |
| **Flujo Normal:** | 1.- El usuario selecciona la opción eliminar en la fotografía  que se desea eliminar.  2.- El sistema despliega un mensaje de confirmación.  3.- El usuario confirma la eliminación de la fotografía. [FA1]  4.- El sistema elimina la fotografía del servidor. [EX1] |
| **Flujos Alternos:** | FA1: 1.- El usuario cancela la eliminación de la fotografía.  2.- El sistema cierra el cuadro de dialogo y no elimina la  fotografía. |
| **Excepciones:** | EX1: 1.- El sistema no logra establecer una conexión con el  Servidor. |
| **Poscondiciones:** | La fotografía es eliminada del servidor y no puede ser vista por el usuario. |
| **Entradas:** | Fotografía para eliminar. |
| **Salidas:** | Fotografía eliminada |
| **Incluye:**  **(relación Include)** | Haga clic aquí para escribir texto. |
| **Extiende:**  **(relación Extend)** | Haga clic aquí para escribir texto. |
| **Prioridad:** | Elija un elemento. |

#### Diagrama de robustez

### Guardar foto en favoritos

#### Descripción y flujos

|  |  |
| --- | --- |
| **­­ID:** | CU08 |
| **Nombre:** | Guardar foto en favoritos |
| **Autor(es):** | Ulises González Robles |
| **Fecha de creación:** | 21/10/2019 |
| **Fecha de actualización:** | Fecha en la que se modifica (opcional) |
| **Actor(es):** | Usuario |
| **Descripción:** | El usuario guardar una fotografía en favoritos. |
| **Precondiciones:** | Debe existir una fotografía en el servidor. |
| **Flujo Normal:** | 1.- El usuario selecciona guardar en favoritos en la fotografía  deseada. [FA1]  2.- El sistema agrega la fotografía a los favoritos del usuario.  [EX1] |
| **Flujos Alternos:** | FA1: 1.- El usuario selecciona guardar en favoritos en una  fotografía que ya se encuentra en favoritos.  2.- El sistema elimina la fotografía de favoritos. |
| **Excepciones:** | EX1: 1.- El sistema no logra establecer una conexión con el  Servidor. |
| **Poscondiciones:** | La fotografía es eliminada de los favoritos y no puede ser vista por el usuario en su lista de favoritos. |
| **Entradas:** | Fotografía para agregar a favoritos. |
| **Salidas:** | Fotografía almacenada en la lista de favoritos del usuario. |
| **Incluye:**  **(relación Include)** | Haga clic aquí para escribir texto. |
| **Extiende:**  **(relación Extend)** | Haga clic aquí para escribir texto. |
| **Prioridad:** | Elija un elemento. |

#### Diagrama de robustez

### Consultar fotos de favoritos

#### Descripción y flujos

|  |  |
| --- | --- |
| **­­ID:** | CU09 |
| **Nombre:** | Consultar fotos favoritas |
| **Autor(es):** | Ulises González Robles |
| **Fecha de creación:** | 21/10/2019 |
| **Fecha de actualización:** | Fecha en la que se modifica (opcional) |
| **Actor(es):** | Usuario |
| **Descripción:** | El usuario podrá consultar sus fotografías favoritas. |
| **Precondiciones:** | Debe existir una fotografía guardada en favoritos en su cuenta. |
| **Flujo Normal:** | 1.- El usuario mostrar fotografías guardadas en favoritos.  2.- El sistema despliega todas las fotografías guardadas en  favoritos. [EX1] |
| **Flujos Alternos:** |  |
| **Excepciones:** | EX1: 1.- El sistema no logra establecer una conexión con el  Servidor. |
| **Poscondiciones:** | Las fotografías guardadas en favoritos son desplegadas para la visualización por el usuario. |
| **Entradas:** | Haga clic aquí para escribir texto. |
| **Salidas:** | Visualización de Fotografías almacenadas en la lista de favoritos del usuario. |
| **Incluye:**  **(relación Include)** | Haga clic aquí para escribir texto. |
| **Extiende:**  **(relación Extend)** | Haga clic aquí para escribir texto. |
| **Prioridad:** | Elija un elemento. |

#### Diagrama de robustez

### Chat personal

#### Descripción y flujos

|  |  |
| --- | --- |
| **­­ID:** | CU10 |
| **Nombre:** | Iniciar Chat personal |
| **Autor(es):** | Jethran Enrique Gómez San Gabriel |
| **Fecha de creación:** | 19/10/2019 |
| **Fecha de actualización:** | Fecha en la que se modifica (opcional) |
| **Actor(es):** | Usuario |
| **Descripción:** | El usuario podrá iniciar un chat personal con algunos de sus seguidores |
| **Precondiciones:** | PRE1.- El usuario del sistema deberá estar autenticado dentro del sistema |
| **Disparador:** | El usuario da clic en el icono “Nuevo mensaje” |
| **Flujo Normal:** | FN1: Iniciar chat   1. El sistema muestra la **GUINuevoChat** con:  * un campo de busqueda: * Un botón de “Nuevo grupo” * Debajo todos los contactos que tiene  1. El usuario da clic en un contacto (ver FA1, Error 500) 2. El sistema abre un nuevo chat con el usuario seleccionado   Fin del CU |
| **Flujos Alternos:** | **FA1: Buscar usuario para nuevo chat**   1. El usuario ingresa el nombre de un usuario en el campo de busqueda 2. El sistema recupera los contactos similares con el nombre del usuario ingresado (ver FA2) 3. El usuario da clic en el contacto   **FA2: No se encontró el usuario**   1. El sistema manda un mensaje “No se encontró” + el nombre de contacto buscado 2. El usuario da clic en cancelar |
| **Excepciones:** | **Error 500: No se ha podido acceder a la información solicitada.**   1. El Sistema muestra un pop-up con el mensaje “Ocurrió un error en el servidor. Inténtelo más tarde”. 2. Termina caso de uso |
| **Poscondiciones:** | POST1.- Se crea un nuevo chat |
| **Entradas:** | Ninguna |
| **Salidas:** | Ninguna |
| **Incluye:**  **(relación Include)** | Ninguna |
| **Extiende:**  **(relación Extend)** | CU11.- Nuevo Grupo |
| **Prioridad:** | ALTA |

#### Diagrama de robustez

### Chat grupal

#### Descripción y flujos

|  |  |
| --- | --- |
| **­­ID:** | CU11 |
| **Nombre:** | Iniciar chat grupal |
| **Autor(es):** | José Alberto Espinoza Castillo |
| **Fecha de creación:** | 18/10/2019 |
| **Fecha de actualización:** | 20/10/2019- Se añadieron flujos alternos |
| **Actor(es):** | Usuario |
| **Descripción:** | Permitirá al Usuario iniciar un chat grupal con los usuarios que elija |
| **Precondiciones:** | Pre01.- Debe haber una cuenta de usuario activa. |
| **Disparador:** | El Usuario hace clic en “Nuevo Grupo” en la pantalla **GUINuevoChat** |
| **Flujo Normal:** | 1. El Sistema despliega **GUINuevoGrupoNombre** el cual contiene un formulario con los siguientes campos:    1. Nombre de grupo 2. El Usuario completa los campos solicitados y hace clic en “Siguiente” 3. El Sistema verifica que los campos estén completos y despliega **GUINuevoGrupoIntegrantes** el cual contiene una lista de AMIGOS del Usuario para seleccionar. 4. El Usuario selecciona los AMIGOS para agregar al grupo. 5. El Sistema crea el nuevo grupo, con los integrantes seleccionados. 6. El Sistema despliega **GUIChat** con los datos del grupo creado. |
| **Flujos Alternos:** | **FA-1. Existen campos obligatorios vacíos.**   1. El Sistema despliega un pop-up con el mensaje: “Por favor rellene todos los campos” 2. Continua en el paso 2 del flujo normal.   **FA-2. No se han seleccionado integrantes**   1. El Sistema despliega un pop-up con el mensaje: “Seleccione al menos un integrante” 2. Continua en el paso 4 del flujo normal |
| **Excepciones:** | **Error 500: No se ha podido acceder a la información solicitada.**   1. El Sistema muestra un pop-up con el mensaje “Ocurrió un error en el servidor. Inténtelo más tarde”. 2. Termina caso de uso   **Error 500: No se ha podido registrar el grupo**   1. El Sistema muestra un pop-up con el mensaje “Ocurrió un error en el servidor. Inténtelo más tarde”. 2. Termina caso de uso |
| **Postcondiciones:** | PosCO01 – Si el flujo concluyo normalmente el nuevo CHATGRUPAL deberá estar registrado en el Sistema  PosCO02 – Si el flujo concluyo normalmente el usuario deberá ser redirigido al chat actualmente creado. |
| **Prioridad:** | Alta |

#### Diagrama de robustez

### Enviar mensaje

#### Descripción y flujos

|  |  |
| --- | --- |
| **­­ID:** | CU12 |
| **Nombre:** | Enviar mensaje |
| **Autor(es):** | José Alberto Espinoza Castillo |
| **Fecha de creación:** | 20/10/2019 |
| **Fecha de actualización:** | Haga clic aquí para escribir una fecha. |
| **Actor(es):** | Usuario |
| **Descripción:** | Permitirá al Usuario enviar un mensaje a un chat o grupo del Sistema |
| **Precondiciones:** | Pre01.- Debe haber una cuenta de usuario activa.  Pre02.- Debe haber una conexión a Internet |
| **Disparador:** | Usuario hace clic en un chat de **GUIChatsRecientes** |
| **Flujo Normal:** | 1. El Sistema desplegara **GUIChat** con:    1. Input para mensajes    2. Botón de enviar    3. Botón para chat de voz    4. Mensajes ordenados del más reciente hacia el más antiguo. 2. El Usuario escribe un mensaje y hace clic en “Enviar”. 3. El Sistema envía el mensaje a los integrantes del chat. 4. El Sistema despliega el mensaje enviado arriba del input de mensajes. 5. Termina caso de uso |
| **Flujos Alternos:** | **FA-1. Usuario hace clic en “Mensaje de voz”**   1. El Sistema empieza la captura de voz y muestra un cronometro con el tiempo transcurrido del mensaje 2. El Usuario termina de grabar y selecciona “Enviar” 3. El Sistema envía el mensaje a los integrantes del chat (Uno para el chat personal o varios para el chat grupal) 4. El Sistema despliega el mensaje enviado arriba del input de mensajes. 5. Termina caso de uso |
| **Excepciones:** | **Error 500: No se ha podido acceder a la información solicitada.**   1. El Sistema muestra un pop-up con el mensaje “Ocurrió un error en el servidor. Inténtelo mas tarde”. 2. Termina caso de uso   **Error 500: No se ha podido enviar el mensaje por falla con la conexión**   1. El Sistema almacena los mensajes enviados al cliente en una cola de mensajes pendientes 2. Cuando se recupere la conexión a internet el Sistema deberá enviar los mensajes de la cola de mensajes pendientes |
| **Postcondiciones:** | PosCO01 – Si el flujo normal concluyo sin errores deberá haber un nuevo mensaje de voz o texto enviado a los integrantes del chat. |
| **Prioridad:** | Alta |

#### Diagrama de robustez

### Eliminar mensaje

#### Descripción y flujos

|  |  |
| --- | --- |
| **­­ID:** | CU13 |
| **Nombre:** | Eliminar mensaje |
| **Autor(es):** | Jethran Enrique Gómez San Gabriel |
| **Fecha de creación:** | 19/10/2019 |
| **Fecha de actualización:** | Fecha en la que se modifica (opcional) |
| **Actor(es):** | Usuario |
| **Descripción:** | El usuario podrá borrar un mensaje de un chat |
| **Precondiciones:** | PRE1.- El usuario deberá estar autenticado dentro del sistema |
| **Flujo Normal:** | **FN1: Eliminar mensaje**   1. El usuario da clic en un mensaje 2. El sistema muestra un pequeño de cuadro con dos opciones:  * Editar * Eliminar  1. El usuario da clic en eliminar (ver FA1) 2. El sistema manda un mensaje de confirmación 3. El usuario da clic en “Eliminar” (ver FA2) 4. El sistema quita el mensaje del chat y lo elimina de la base de datos   Fin CU |
| **Flujos Alternos:** | **FA1: Editar mensaje**   1. El usuario da clic en Editar 2. Redirige el CU13   **FA2: Cancelar**   1. El usuario da clic en “Cancelar” 2. El sistema regresa al chat |
| **Excepciones:** | **Error 500: No se ha podido acceder a la información solicitada.**   1. El Sistema muestra un pop-up con el mensaje “Ocurrió un error en el servidor. Inténtelo más tarde”. 2. Termina caso de uso |
| **Poscondiciones:** | POST1.- El chat debe estar eliminado de la base de datos y no debe volverse a mostrar en el chat |
| **Entradas:** | Ninguno |
| **Salidas:** | Ninguno |
| **Incluye:**  **(relación Include)** | Ninguno |
| **Extiende:**  **(relación Extend)** | CU13.- Editar mensaje |
| **Prioridad:** | ALTA |

#### Diagrama de robustez

### Editar mensaje

#### Descripción y flujos

|  |  |
| --- | --- |
| **­­ID:** | CU14 |
| **Nombre:** | Editar mensaje |
| **Autor(es):** | Jethran Enrique Gómez San Gabriel |
| **Fecha de creación:** | 19/10/2019 |
| **Fecha de actualización:** | Fecha en la que se modifica (opcional) |
| **Actor(es):** | Usuario |
| **Descripción:** | El usuario podrá editar un mensaje enviado dentro de un chat |
| **Precondiciones:** | PRE1.- El usuario deberá estar autenticado dentro del sistema |
| **Flujo Normal:** | **FN1: Editar mensaje**   1. El usuario da clic en un mensaje 2. El sistema muestra un pequeño de cuadro con dos opciones:  * Editar * Eliminar  1. El usuario da clic en editar (ver FA1) 2. El sistema hable el campo de mensaje con el texto del mensaje a editar 3. El usuario ingresa el texto y da clic en enviar (ver FA2) 4. El sistema cambia el mensaje al nuevo y marca que fue editado   Fin del CU |
| **Flujos Alternos:** | **FA1: Eliminar mensaje**   1. El usuario da clic en Eliminar 2. Redirige el CU12   **FA2: Cancelar**   1. El usuario da clic en “Cancelar” 2. El sistema regresa al chat |
| **Excepciones:** | **Error 500: No se ha podido acceder a la información solicitada.**   1. El Sistema muestra un pop-up con el mensaje “Ocurrió un error en el servidor. Inténtelo más tarde”. 2. Termina caso de uso |
| **Poscondiciones:** | POST1.- El mensaje deberá estar marcado como editado |
| **Entradas:** | Ninguno |
| **Salidas:** | Ninguno |
| **Incluye:**  **(relación Include)** | Ninguno |
| **Extiende:**  **(relación Extend)** | CU12.- Eliminar mensaje |
| **Prioridad:** | ALTO |

#### Diagrama de robustez

### Buscar usuario

#### Descripción y flujos

|  |  |
| --- | --- |
| **­­ID:** | CU15 |
| **Nombre:** | Buscar usuario |
| **Autor(es):** | José Alberto Espinoza Castillo |
| **Fecha de creación:** | 18/10/2019 |
| **Fecha de actualización:** | 20/10/2019- Se añadieron flujos alternos |
| **Actor(es):** | Usuario |
| **Descripción:** | Permitirá al Usuario buscar otros usuarios |
| **Precondiciones:** | Pre01.- Debe haber una cuenta de usuario activa. |
| **Disparador:** | El Usuario hace clic en “Buscar” en el menú desplegable principal |
| **Flujo Normal:** | 1. El Sistema despliega **GUIBuscarUsuario** con un único input para agregar la búsqueda. 2. El Usuario ingresa los parámetros de búsqueda 3. El Sistema despliega debajo del input los usuarios relacionados con los parámetros de búsqueda. 4. Termina caso de uso |
| **Flujos Alternos:** | **FA-1. No hay resultados que coincidan con los parámetros ingresados**   1. El Sistema despliega un pop-up con el mensaje: “No hay resultados” 2. Continua en el paso 2 del flujo normal.   **FA-2. Clic en un resultado de la consulta**   1. Extiende al “CU0.- Ver información de usuario”. |
| **Excepciones:** | **Error 500: No se ha podido acceder a la información solicitada.**   1. El Sistema muestra un pop-up con el mensaje “Ocurrió un error en el servidor. Inténtelo más tarde”. 2. Termina caso de uso |
| **Postcondiciones:** | PosCO01 – Si el flujo concluyo normalmente deberá mostrar un listado de los resultados de la consulta. |
| **Prioridad:** | Alta |

#### Diagrama de robustez

### Ver información de usuario

#### Descripción y flujos

|  |  |
| --- | --- |
| **­­ID:** | CU16 |
| **Nombre:** | Ver información de usuario |
| **Autor(es):** | José Alberto Espinoza Castillo |
| **Fecha de creación:** | 20/10/2019 |
| **Fecha de actualización:** | Haga clic aquí para escribir una fecha. |
| **Actor(es):** | Usuario |
| **Descripción:** | Permitirá al Usuario ver la información de otro usuario del Sistema |
| **Precondiciones:** | Pre01.- Debe haber una cuenta de usuario activa.  Pre02.- Debe haberse seleccionado un Usuario de los resultados de la búsqueda. |
| **Disparador:** | Usuario hace clic en un resultado de **GUIBuscarUsuario** |
| **Flujo Normal:** | 1. El Sistema despliega **GUIInfoUsuario** con la información del usuario seleccionado. 2. Termina caso de uso |
| **Flujos Alternos:** | **FA-1. Usuario hace clic en “Añadir amigo”**   1. El Sistema registra la solicitud en las solicitudes del Usuario seleccionado. 2. Termina caso de uso |
| **Excepciones:** | **Error 500: No se ha podido acceder a la información solicitada.**   1. El Sistema muestra un pop-up con el mensaje “Ocurrió un error en el servidor. Inténtelo más tarde”. 2. Termina caso de uso   **Error 500: No se ha podido registrar la solicitud de amistad**   1. El Sistema muestra un pop-up con el mensaje “Ocurrió un error en el servidor. Inténtelo más tarde”. 2. Termina caso de uso |
| **Postcondiciones:** | PosCO01 – Si el el curso se procedió por el “**FA-1”** deberá haber registrada una nueva solicitud en el las solicitudes del Usuario seleccionado. |
| **Prioridad:** | Alta |

#### Diagrama de robustez

### Revisar solicitudes

#### Descripción y flujos

|  |  |
| --- | --- |
| **­­ID:** | CU18 |
| **Nombre:** | Revisar solicitudes |
| **Autor(es):** | Jethran Enrique Gómez San Gabriel |
| **Fecha de creación:** | 20/10/2019 |
| **Fecha de actualización:** | Fecha en la que se modifica (opcional) |
| **Actor(es):** | Usuario |
| **Descripción:** | El usuario podrá revisar las solicitudes que le han sido enviadas para poder aceptarlas o cancelarlas |
| **Precondiciones:** | PRE1.- El usuario deberá estar autenticado dentro del sistema |
| **Flujo Normal:** | FN1: Aceptar solitud de amistad   1. El usuario da clic en el icono de ver solicitudes 2. El sistema recupera todas las solicitudes del servidor 3. El usuario da clic en una solicitud 4. El sistema muestra un mensaje para confirmación de solicitud (ver FA2) 5. El usuario da clic en Aceptar (ver FA1) 6. El sistema muestra que el amigo ya ha sido agregado   Fin CU |
| **Flujos Alternos:** | FA1: Cancelar solicitud   1. El usuario da clic en cancelar 2. El sistema borra la solicitud de amistad de la base de datos y ya no la muestra   FA2: Solicitud no disponible   1. El sistema manda un mensaje que la solicitud ya no se encuentra disponible 2. El usuario da clic en aceptar |
| **Excepciones:** | **Error 500: No se ha podido acceder a la información solicitada.**   1. El Sistema muestra un pop-up con el mensaje “Ocurrió un error en el servidor. Inténtelo más tarde”. 2. Termina caso de uso |
| **Poscondiciones:** | POST1.- Se le debe sumar un amigo al usuario en la base de datos |
| **Entradas:** | Ninguno |
| **Salidas:** | Ninguno |
| **Incluye:**  **(relación Include)** | Ninguno |
| **Extiende:**  **(relación Extend)** | Ninguno |
| **Prioridad:** | ALTO |

#### Diagrama de robustez

### Ver feed de fotos

#### Descripción y flujos

#### Diagrama de robustez

### Consultar chats recientes

#### Descripción y flujos

#### Diagrama de robustez

## Casos de uso de moderador

### Revisar contenido

#### Descripción y flujos

|  |  |
| --- | --- |
| **­­ID:** | CU21 |
| **Nombre:** | Revisar contenido |
| **Autor(es):** | Jethran Enrique Gómez San Gabriel |
| **Fecha de creación:** | 19/10/2019 |
| **Fecha de actualización:** | Fecha en la que se modifica (opcional) |
| **Actor(es):** | Moderador |
| **Descripción:** | El moderado podrá consultar todo el contenido que se haya subido a la aplicación |
| **Precondiciones:** | PRE1.- El moderado deberá estar autenticado dentro del sistema |
| **Flujo Normal:** | FN1: consultar contenido   1. El moderador da clic en consultar contenido 2. El sistema se conecta con el servidor y recupera todas las fotos subidas 3. El moderador selecciona un filtro (ver FA1, FA2) 4. El sistema muestra las imágenes del tipo de filtro seleccionado |
| **Flujos Alternos:** | FA1: Filtro reportadas   1. El moderado selecciona el filtro de fotos reportadas 2. El sistema recupera todas las fotos reportadas desde el servidor y las muestra en mosaicos   FA2: Filtro por usuario   1. El moderado ingresa el nombre de un usuario y clic en buscar 2. El sistema recupera todas las fotos del usuario buscado y las muestra en mosaicos |
| **Excepciones:** | EX01.- Error al conectarse con el servidor   1. El sistema manda un mensaje que hubo un error al conectarse con el servidor 2. El moderado da clic en aceptar 3. El sistema cierra la apliación |
| **Poscondiciones:** | POST1.- Recuperar todas las fotos y mostrarlas en mosaicos. |
| **Entradas:** | Ninguno |
| **Salidas:** | Ninguno |
| **Incluye:**  **(relación Include)** | Ninguno |
| **Extiende:**  **(relación Extend)** | CU15 Eliminar contenido |
| **Prioridad:** | ALTA |

#### Diagrama de robustez

### Eliminar contenido

#### Descripción y flujos

|  |  |
| --- | --- |
| **­­ID:** | CU22 |
| **Nombre:** | Eliminar contenido |
| **Autor(es):** | Jethran Enrique Gómez San Gabriel |
| **Fecha de creación:** | 19/10/2019 |
| **Fecha de actualización:** | Fecha en la que se modifica (opcional) |
| **Actor(es):** | Moderador |
| **Descripción:** | El moderado podrá eliminar una publicación que inflija las normas de la empresa |
| **Precondiciones:** | PRE1.- El moderador deberá estar autenticado dentro del sistema  PRE2.- El moderador deberá haber consultado antes las publicaciones |
| **Flujo Normal:** | FN1: Eliminar contenido   1. El moderador selecciona una publicación (ver FA2) 2. El sistema muestra la publicación en pantalla completa 3. El moderado da clic en eliminar publicación 4. El sistema manda un mensaje de confirmación de eliminación de publicación 5. El moderador da clic en Eliminar (ver FA1) 6. El sistema elimina la publicación de la base de datos 7. El sistema manda un mensaje de que se envió un mensaje al dueño de la publicación |
| **Flujos Alternos:** | FA1: Cancelar   1. El moderador da clic en cancelar 2. El sistema regresa a la guiconsultarpublicaciones   FA2: No existe esa publicación   1. El sistema manda un mensaje que la publicación ya no se encuentra disponible 2. El moderador da clic en aceptar 3. El sistema recarga las publicaciones |
| **Excepciones:** | EX01.- Error al conectarse con el servidor   1. El sistema manda un mensaje que hubo un error al conectarse con el servidor 2. El moderador da clic en aceptar 3. El sistema cierra la aplicación |
| **Poscondiciones:** | POST1.- El sistema eliminara una publicación de la base de datos  POST2.- El sistema deberá notificarle al usuario de que su publicación ha sido eliminada |
| **Entradas:** | Ninguno |
| **Salidas:** | Ninguno |
| **Incluye:**  **(relación Include)** | Ninguno |
| **Extiende:**  **(relación Extend)** | Ninguno |
| **Prioridad:** | ALTA |

#### Diagrama de robustez

### Consultar cuentas

#### Descripción y flujos

#### Diagrama de robustez

### Suspender cuenta

#### Descripción y flujos

|  |  |
| --- | --- |
| **­­ID:** | CU24 |
| **Nombre:** | Suspender cuenta |
| **Autor(es):** | Jethran Enrique Gómez San Gabriel |
| **Fecha de creación:** | 20/10/2019 |
| **Fecha de actualización:** | Fecha en la que se modifica (opcional) |
| **Actor(es):** | Moderador |
| **Descripción:** | El moderador podrá suspender una cuenta de usuario que inflija las reglas |
| **Precondiciones:** | PRE1.- El moderador deberá estar autenticado dentro del sistema |
| **Disparador** | El moderador da clic en suspender cuenta |
| **Flujo Normal:** | FN1: Suspender cuenta   1. El sistema muestra la guisuspendercuenta, con un campo de busqueda 2. El sistema recupera las cuentas que ya le hayan sido eliminada 3 imágenes 3. El moderador escoge un usuario (ver FA1) 4. El sistema lo manda al perfil del usuario mostrando dos botonees; suspender y cancelar 5. El moderador da clic en suspender (ver FA2) 6. El sistema manda un mensaje de confirmación de que ya ha sido suspendido su cuenta 7. El moderador da clic en aceptar |
| **Flujos Alternos:** | FA1: Buscar usuario   1. El moderador ingresa el nombre del usuario 2. El sistema recupera los usuarios similares con el nombre ingresado 3. El moderado escoge un usuario 4. Pasa al paso 4 del flujo normal   FA2: Cancelar   1. El moderador da clic en “Cancelar” 2. El sistema regresa al paso 1 del FN1 |
| **Excepciones:** | EX01.- Error al conectarse con el servidor   1. El sistema manda un mensaje que hubo un error al conectarse con el servidor 2. El moderador da clic en aceptar 3. El sistema cierra la aplicación |
| **Poscondiciones:** | POST1.- Una cuenta pasa a suspendida en la base de datos  POST2.- Se notifica al usuario que la cuenta ha sido suspendida |
| **Entradas:** | Ninguna |
| **Salidas:** | Ninguno |
| **Incluye:**  **(relación Include)** | Ninguno |
| **Extiende:**  **(relación Extend)** | Ninguno |
| **Prioridad:** | MEDIA |

#### Diagrama de robustez

### Activar cuenta

#### Descripción y flujos

#### Diagrama de robustez

# Características del Servidor

## Diagrama de casos de uso del Servidor RestAPI

### Registrar nuevo usuario

|  |  |
| --- | --- |
| **­­ID:** | SCU01 |
| **Nombre:** | Registro de nuevo usuario |
| **Autor(es):** | José Alberto Espinoza Castillo |
| **Fecha de creación:** | 06/11/2019 |
| **Fecha de actualización:** | 06/11/2019 |
| **Actor(es):** | Servidor RestAPI |
| **Descripción:** | Permitirá añadir un nuevo usuario al Sistema |
| **Precondiciones:** | Pre01.- No hay un usuario con el mismo nombre de usuario registrado  Pre02.- No hay un usuario con el mismo correo registrado |
| **Flujo Normal:** | 1. El Servidor RestAPI recibe una petición POST para registrar un nuevo usuario, con el siguiente “body”:    1. Nombre    2. Apellido    3. Nombre usuario    4. Correo electrónico    5. Numero de celular    6. Contraseña 2. El Servidor RestAPI verifica que los atributos del body no se encuentren vacíos y los almacena en la Base de datos. 3. El Servidor RestAPI retorna un código 201 y termina caso de uso |
| **Flujos Alternos:** | **FA-1. Existen atributos del body que están vacíos**   1. El Servidor RestAPI retorna un código 400, con el mensaje “parámetros inválidos o incompletos”. 2. Termina caso de uso. |
| **Excepciones:** | **EX-1. No se ha podido registrar la cuenta de usuario en el Sistema.**   1. El Servidor RestAPI retorna un código 500, con el mensaje “No se pudo ejecutar el registro”. 2. Termina caso de uso. |
| **Postcondiciones:** | PosCO01 – Si el flujo concluyo normalmente debe haber una CUENTA registrada. |

### Guardar foto de usuario

|  |  |
| --- | --- |
| **­­ID:** | SCU02 |
| **Nombre:** | Guardar foto de usuario |
| **Autor(es):** | Ulises González Robles |
| **Fecha de creación:** | 07/11/2019 |
| **Fecha de actualización:** | Fecha en la que se modifica (opcional) |
| **Actor(es):** | Servidor RestAPI |
| **Descripción:** | El servidor RestAPI guardará la foto subida por el usuario |
| **Precondiciones:** | Haga clic aquí para escribir texto. |
| **Flujo Normal:** | 1.- El Servidor RestAPI recibe una petición POST para registrar una nueva fotografía del usuario.  2.- El servidor verifica que el body no sea nulo. [FA1]  3.- El servidor guarda la fotografía en la base de datos. [EX1] |
| **Flujos Alternos:** | FA1: 1.- El body es nulo, el Servidor RestAPI regresa un código 400 con el mensaje “Parámetros inválidos” |
| **Excepciones:** | EX1: 1.- El servidor RestAPI no logra guardar la fotografía en la base de datos regresa un código 500 con el mensaje “Error de conexión con la base de datos” |
| **Poscondiciones:** | Haga clic aquí para escribir texto. |
| **Prioridad:** | Alta |

### Guardar comentario de foto

|  |  |
| --- | --- |
| **­­ID:** | SCU03 |
| **Nombre:** | Guardar comentario de foto |
| **Autor(es):** | Ulises González Robles |
| **Fecha de creación:** | 07/11/2019 |
| **Fecha de actualización:** | Fecha en la que se modifica (opcional) |
| **Actor(es):** | Servidor RestAPI |
| **Descripción:** | El servidor RestAPI guardará la foto subida por el usuario |
| **Precondiciones:** | Debe existir una fotografía almacenada en la base de datos. |
| **Flujo Normal:** | 1.- El Servidor RestAPI recibe una petición POST para registrar un comentario en una fotografía.  2.- El servidor verifica que el body no sea nulo. [FA1]  3.- El servidor guarda el comentario en la base de datos. [EX1] |
| **Flujos Alternos:** | FA1: 1.- El body es nulo, el Servidor RestAPI regresa un código 400 con el mensaje “Parámetros inválidos” |
| **Excepciones:** | EX1: 1.- El servidor RestAPI no logra guardar el comentario de fotografía en la base de datos regresa un código 500 con el mensaje “Error de conexión con la base de datos” |
| **Poscondiciones:** |  |

### Eliminar foto de usuario

|  |  |
| --- | --- |
| **­­ID:** | SCU04 |
| **Nombre:** | Eliminar foto del usuario |
| **Autor(es):** | Ulises González Robles |
| **Fecha de creación:** | 07/11/2019 |
| **Fecha de actualización:** | Fecha en la que se modifica (opcional) |
| **Actor(es):** | Servidor RestAPI |
| **Descripción:** | El servidor RestAPI Eliminará la foto subida por el usuario |
| **Precondiciones:** | Debe existir una fotografía almacenada en la base de datos. |
| **Flujo Normal:** | 1.- El Servidor RestAPI recibe una petición DEL para eliminar una fotografía.  2.- El servidor verifica que el body no sea nulo. [FA1]  3.- El servidor eliminar la foto de la base de datos. [EX1] |
| **Flujos Alternos:** | FA1: 1.- El body es nulo, el Servidor RestAPI regresa un código 400 con el mensaje “Parámetros inválidos” |
| **Excepciones:** | EX1: 1.- El servidor RestAPI no logra eliminar la fotografía en la base de datos, regresa un código 500 con el mensaje “Error de conexión con la base de datos” |

### Guardar foto en favoritos de usuario

|  |  |
| --- | --- |
| **­­ID:** | SCU05 |
| **Nombre:** | Guardar foto en favoritos del usuario |
| **Autor(es):** | Ulises González Robles |
| **Fecha de creación:** | 07/11/2019 |
| **Fecha de actualización:** | Fecha en la que se modifica (opcional) |
| **Actor(es):** | Servidor RestAPI |
| **Descripción:** | El servidor RestAPI guardará una foto del usuario en favoritos |
| **Precondiciones:** | Debe existir una fotografía almacenada en la base de datos. |
| **Flujo Normal:** | 1.- El Servidor RestAPI recibe una petición POST para agregar una fotografía en favoritos  2.- El servidor verifica que el body no sea nulo. [FA1]  3.- El servidor modifica el atributo favorito a True en la foto de la base de datos. [EX1] |
| **Flujos Alternos:** | FA1: 1.- El body es nulo, el Servidor RestAPI regresa un código 400 con el mensaje “Parámetros inválidos” |
| **Excepciones:** | EX1: 1.- El servidor RestAPI no logra eliminar la fotografía en la base de datos, regresa un código 500 con el mensaje “Error de conexión con la base de datos” |

### Consultar fotos de favoritos de usuario

|  |  |
| --- | --- |
| **­­ID:** | SCU05 |
| **Nombre:** | Guardar foto en favoritos del usuario |
| **Autor(es):** | Ulises González Robles |
| **Fecha de creación:** | 07/11/2019 |
| **Fecha de actualización:** | Fecha en la que se modifica (opcional) |
| **Actor(es):** | Servidor RestAPI |
| **Descripción:** | El servidor RestAPI guardará una foto del usuario en favoritos |
| **Precondiciones:** | Debe existir una fotografía almacenada en la base de datos. |
| **Flujo Normal:** | 1.- El Servidor RestAPI recibe una petición POST para agregar una fotografía en favoritos  2.- El servidor verifica que el body no sea nulo. [FA1]  3.- El servidor modifica el atributo favorito a True en la foto de la base de datos. [EX1] |
| **Flujos Alternos:** | FA1: 1.- El body es nulo, el Servidor RestAPI regresa un código 400 con el mensaje “Parámetros inválidos” |
| **Excepciones:** | EX1: 1.- El servidor RestAPI no logra eliminar la fotografía en la base de datos, regresa un código 500 con el mensaje “Error de conexión con la base de datos” |

### Crear nuevo chat personal

|  |  |
| --- | --- |
| **­­ID:** | SCU07 |
| **Nombre:** | Crear nuevo chat personal |
| **Autor(es):** | Jethran Enrique Gómez San Gabriel |
| **Fecha de creación:** | 05/11/2019 |
| **Actor(es):** | Servidor RestAPI |
| **Descripción:** | El Servidor RestAPI crea un nuevo chat personal entre dos usuarios |
| **Precondiciones:** | PRE1.- El usuario deberá estar autenticado |
| **Flujo Normal:** | 1. El Servidor RestAPI hace un método POST en el chat con:  * idUsuario * idAmigo  1. Verifica que los atributos no estén vacíos 2. El Servidor RestAPI crea un nuevo chat con los dos usuarios 3. El Servidor RestAPI regresa un estatus 200 (ver FA1, EX1) |
| **Flujos Alternos:** | FA1: Bad Request   1. El Servidor RestAPI regresa un estatus 400 |
| **Excepciones:** | EX1.- Internal Server Error   1. No se pudo conectar con la base de datos 2. El Servidor RestAPI regresa un estatus 500 |

### Crear nuevo chat grupal

|  |  |
| --- | --- |
| **­­ID:** | SCU07 |
| **Nombre:** | Crear nuevo chat grupal |
| **Autor(es):** | José Alberto Espinoza Castillo |
| **Fecha de creación:** | 06/11/2019 |
| **Fecha de actualización:** | 06/11/2019 |
| **Actor(es):** | Servidor RestAPI |
| **Descripción:** | Permitirá a un usuario crear un nuevo chat grupal en el Sistema |
| **Precondiciones:** | Pre01.- No esta previamente autenticado en el Sistema |
| **Flujo Normal:** | 1. El Servidor RestAPI recibe una petición POST para registrar un nuevo chat grupal con el siguiente “body”:    1. Nombre chat    2. Un conjunto de identificadores de los usuarios miembros 2. El Servidor RestAPI verifica que existan mas de dos identificadores de usuarios miembros y el atributo nombre de chat no se encuentre vacío. 3. El Servidor RestAPI registra el nuevo chat grupal en la base de datos y regresa un código 201. |
| **Flujos Alternos:** | **FA-1. Atributo nombre chat está vacío o id miembros son menos de 3**   1. El Servidor RestAPI retorna un código 400, con el mensaje “parámetros inválidos o incompletos”. 2. Termina caso de uso. |
| **Excepciones:** | **EX-1. No se ha podido realizar el registro**   1. El Servidor RestAPI retorna un código 500, con el mensaje “No se pudo ejecutar el registro solicitado”. 2. Termina caso de uso. |
| **Postcondiciones:** | PosCO01 – Si el flujo concluyo normalmente debe haber un nuevo chat grupal registrado. |

### Registrar mensaje

|  |  |
| --- | --- |
| **­­ID:** | SCU09 |
| **Nombre:** | Registrar mensaje |
| **Autor(es):** | José Alberto Espinoza Castillo |
| **Fecha de creación:** | 06/11/2019 |
| **Fecha de actualización:** | 06/11/2019 |
| **Actor(es):** | Servidor RestAPI |
| **Descripción:** | Permitirá al Servidor RestAPI registrar un nuevo mensaje |
| **Precondiciones:** | Pre01.- Hay un usuario con una cuenta activa. |
| **Flujo Normal:** | 1. El Servidor RestAPI recibe una petición POST para registrar un nuevo mensaje con el siguiente “body”:    1. Texto del mensaje, ya sea string de un mensaje de texto o un string codificado en base64 para los mensajes de audio    2. Identificador del chat al que pertenece    3. Identificador del usuario que lo envió 2. El Servidor RestAPI verifica que los atributos del body estén completos y registra el nuevo mensaje en la base de datos 3. El Servidor RestAPI regresa un código 201 |
| **Flujos Alternos:** | **FA-1. Existen atributos del body que están vacíos**   1. El Servidor RestAPI retorna un código 400, con el mensaje “parámetros inválidos o incompletos”. 2. Termina caso de uso. |
| **Excepciones:** | **EX-1. No se ha podido realizar el registro**   1. El Servidor RestAPI retorna un código 500, con el mensaje “No se pudo ejecutar el registro solicitado”. 2. Termina caso de uso. |
| **Postcondiciones:** | PosCO01 – Si el flujo concluyo normalmente debe haber un nuevo mensaje registrado en el chat referenciado |

### Modificar mensaje

### Consultar usuarios

|  |  |
| --- | --- |
| **­­ID:** | SCU11 |
| **Nombre:** | Consultar usuarios |
| **Autor(es):** | José Alberto Espinoza Castillo |
| **Fecha de creación:** | 06/11/2019 |
| **Fecha de actualización:** | 06/11/2019 |
| **Actor(es):** | Servidor RestAPI |
| **Descripción:** | Permitirá al Servidor RestAPI devolver una lista con los usuarios consultados |
| **Precondiciones:** | Pre01.- Hay un usuario con una cuenta activa. |
| **Flujo Normal:** | 1. El Servidor RestAPI recibe una petición GET para devolver una lista de usuarios con un parámetro de búsqueda que hace referencia al nombre de usuario que desee buscar 2. El Servidor RestAPI revisa que el parámetro de búsqueda tenga al menos un carácter y realiza un consulta buscando los mejores resultados. 3. El Servidor RestAPI retorna la información de los usuarios resultantes de la consulta con un código 200 |
| **Flujos Alternos:** | **FA-1. Existen atributos del body que están vacíos**   1. El Servidor RestAPI retorna un código 400, con el mensaje “parámetros inválidos o incompletos”. 2. Termina caso de uso.   **FA-2. No se encontró ningún registra que coincidiera con el parámetro de búsqueda**   1. El Servidor RestAPI retorna un código 404, con el mensaje “no se encontró ningún resultado” 2. Termina caso de uso |
| **Excepciones:** | **EX-1. No se ha podido realizar el registro**   1. El Servidor RestAPI retorna un código 500, con el mensaje “No se pudo ejecutar el registro solicitado”. 2. Termina caso de uso. |
| **Postcondiciones:** | PosCO01 – Si el flujo concluyo normalmente debe haber un nuevo mensaje registrado en el chat referenciado |

### Consultar información de usuario

|  |  |
| --- | --- |
| **­­ID:** | SCU12 |
| **Nombre:** | Consultar información de usuario |
| **Autor(es):** | José Alberto Espinoza Castillo |
| **Fecha de creación:** | 06/11/2019 |
| **Fecha de actualización:** | 06/11/2019 |
| **Actor(es):** | Servidor RestAPI |
| **Descripción:** | Permitirá al Servidor RestAPI devolver un usuario |
| **Precondiciones:** | Pre01.- Hay un usuario con una cuenta activa. |
| **Flujo Normal:** | 1. El Servidor RestAPI recibe una petición GET para devolver la información de un usuario de acuerdo con un atributo id en el body de la petición 2. El Servidor verifica que los atributos del body estén completos y realizan una consulta a la base de datos 3. El Servidor RestAPI devuelve la información del usuario con un código 200 |
| **Flujos Alternos:** | **FA-1. Existen atributos del body que están vacíos**   1. El Servidor RestAPI retorna un código 400, con el mensaje “parámetros inválidos o incompletos”. 2. Termina caso de uso.   **FA-2. No se encontró ningún registra que coincidiera con el parámetro de búsqueda**   1. El Servidor RestAPI retorna un código 404, con el mensaje “no se encontró ningún resultado” 2. Termina caso de uso |
| **Excepciones:** | **EX-1. No se ha podido realizar el registro**   1. El Servidor RestAPI retorna un código 500, con el mensaje “No se pudo ejecutar el registro solicitado”. 2. Termina caso de uso. |
| **Postcondiciones:** | PosCO01 – Si el flujo concluyo normalmente debe haber un nuevo mensaje registrado en el chat referenciado |

### Registrar solicitud de amistad a usuario

|  |  |
| --- | --- |
| **­­ID:** | SCU13 |
| **Nombre:** | Registrar solicitud de amistad a usuario |
| **Autor(es):** | José Alberto Espinoza Castillo |
| **Fecha de creación:** | 07/11/2019 |
| **Fecha de actualización:** | 07/11/2019 |
| **Actor(es):** | Servidor RestAPI |
| **Descripción:** | Permitirá al Servidor RestAPI permitirá registrar una solicitud de amistad |
| **Precondiciones:** | Pre01.- Hay un usuario con una cuenta activa. |
| **Flujo Normal:** | 1. El Servidor RestAPI recibe una petición POST para registrar una nueva solicitud dado los atributos:    1. idUsuarioDestino    2. idUsuarioRemitente 2. El Servidor ResAPI verifica que ninguno esté vacío y registra la solicitud en la base de datos 3. El Servidor regresa un código 200 |
| **Flujos Alternos:** | **FA-1. Existen atributos del body que están vacíos**   1. El Servidor RestAPI retorna un código 400, con el mensaje “parámetros inválidos o incompletos”. 2. Termina caso de uso.   **FA-2. No se encontró ningún usuario con el identificador solicitado**   1. El Servidor RestAPI retorna un código 404, con el mensaje “no se encontró ningún resultado” 2. Termina caso de uso |
| **Excepciones:** | **EX-1. No se ha podido realizar el registro**   1. El Servidor RestAPI retorna un código 500, con el mensaje “No se pudo ejecutar el registro solicitado”. 2. Termina caso de uso. |
| **Postcondiciones:** | PosCO01 – Si el flujo concluyo normalmente debe haber una nueva solicitud registrada en la base de datos |

### Modificar solicitud

|  |  |
| --- | --- |
| **­­ID:** | SCU14 |
| **Nombre:** | Modificar solicitud |
| **Autor(es):** | José Alberto Espinoza Castillo |
| **Fecha de creación:** | 06/11/2019 |
| **Fecha de actualización:** | 06/11/2019 |
| **Actor(es):** | Servidor RestAPI |
| **Descripción:** | Permitirá al Servidor RestAPI cambiar el estado de la solicitud entre aceptado, rechazado o pendiente |
| **Precondiciones:** | Pre01.- Hay un usuario con una cuenta activa.  Pre02.- Existe la solicitud a la que se hace referencia  Pre03.- El estado actual es diferente al estado entrante |
| **Flujo Normal:** | 1. El Servidor RestAPI recibe una petición PUT con el id de la solicitud a la que se hace referencia y el estado al que se desea cambiarla dentro del body 2. El Servidor verifica que los atributos del body esten completos, y verifica que las siguientes condiciones:    1. Si tiene estado “Pendiente”, puede cambiar a “Aceptada o Rechazada”    2. Si tiene estado “Rechazado” puede cambiar a pendiente    3. Si tiene estado “Aceptado puede cambiar a pendiente” 3. El Servidor realiza la modificación y regresa un codigo 201 |
| **Flujos Alternos:** | **FA-1. Existen atributos del body que están vacíos**   1. El Servidor RestAPI retorna un código 400, con el mensaje “parámetros inválidos o incompletos”. 2. Termina caso de uso.   **FA-2. No se encontró ningún registro de solicitud que coincidiera con el parámetro de búsqueda**   1. El Servidor RestAPI retorna un código 404, con el mensaje “no se encontró ningún resultado” 2. Termina caso de uso |
| **Excepciones:** | **EX-1. No se ha podido realizar la modificación**   1. El Servidor RestAPI retorna un código 500, con el mensaje “No se pudo ejecutar la modificación solicitada”. 2. Termina caso de uso. |
| **Postcondiciones:** | PosCO01 – Si el flujo concluyo normalmente debe haber un nuevo mensaje registrado en el chat referenciado |

### Consultar solicitudes

|  |  |
| --- | --- |
| **­­ID:** | SCU16 |
| **Nombre:** | Consultar solicitudes |
| **Autor(es):** | José Alberto Espinoza Castillo |
| **Fecha de creación:** | 06/11/2019 |
| **Fecha de actualización:** | 06/11/2019 |
| **Actor(es):** | Servidor RestAPI |
| **Descripción:** | Permitirá al Servidor RestAPI devolver una lista con las solicitudes del usuario activo |
| **Precondiciones:** | Pre01.- Hay un usuario con una cuenta activa. |
| **Flujo Normal:** | 1. El Servidor RestAPI recibe una petición GET para devolver una lista de solicitudes de acuerdo al identificador de un usuario enviado en los parámetros de la petición 2. El Servidor RestAPI revisa que el parámetro de búsqueda no esté vacío y realiza la consulta de solicitudes a la base de datos 3. El Servidor RestAPI retorna la información de las solicitudes resultantes de la consulta con un código 200 |
| **Flujos Alternos:** | **FA-1. Los parámetros están vacíos o no coinciden con la información solicitada**   1. El Servidor RestAPI retorna un código 400, con el mensaje “parámetros inválidos o incompletos”. 2. Termina caso de uso.   **FA-2. No se encontró ningún registro que coincidiera con el parámetro de búsqueda**   1. El Servidor RestAPI retorna un código 404, con el mensaje “no se encontró ningún resultado” 2. Termina caso de uso |
| **Excepciones:** | **EX-1. No se ha podido realizar el registro**   1. El Servidor RestAPI retorna un código 500, con el mensaje “No se pudo ejecutar el registro solicitado”. 2. Termina caso de uso. |

### Consultar feed de fotos

|  |  |
| --- | --- |
| **­­ID:** | SCU17 |
| **Nombre:** | Consultar feed de fotos |
| **Autor(es):** | José Alberto Espinoza Castillo |
| **Fecha de creación:** | 07/11/2019 |
| **Fecha de actualización:** | 07/11/2019 |
| **Actor(es):** | Servidor RestAPI |
| **Descripción:** | Permitirá al Servidor RestAPI devolver una lista con el feed de fotos ordenados cronológicamente. |
| **Precondiciones:** | Pre01.- Hay un usuario con una cuenta activa. |
| **Flujo Normal:** | 1. El Servidor RestAPI recibe una petición GET para devolver una lista con el feed de fotos conformado por las publicaciones de los amigos del usuario que lo solicita. 2. El Servidor crea una consulta solicitando las información de las fotos y devuelve un codigo 200 con las fotos solicitadas |
| **Flujos Alternos:** | **FA-1. Los parámetros están vacíos o no coinciden con la información solicitada**   1. El Servidor RestAPI retorna un código 400, con el mensaje “parámetros inválidos o incompletos”. 2. Termina caso de uso. |
| **Excepciones:** | **EX-1. No se ha podido realizar la consulta**   1. El Servidor RestAPI retorna un código 500, con el mensaje “No se pudo ejecutar la consulta solicitada”. 2. Termina caso de uso. |

### Revisar contenido

|  |  |
| --- | --- |
| **­­ID:** | CU101 |
| **Nombre:** | Revisar Contenido |
| **Autor(es):** | Jethran Enrique Gómez San Gabriel |
| **Fecha de creación:** | 05/11/2019 |
| **Actor(es):** | Servidor RestAPI |
| **Descripción:** | El Servidor RestAPI recuperara las publicaciones de todos los usuarios |
| **Precondiciones:** | PRE1.- Deberá estar autenticado anteriormente |
| **Flujo Normal:** | 1. El Servidor RestAPI hace una petición GET en las **publicaciones** 2. El Servidor RestAPI recupera todas las **publicaciones** de la base de datos 3. El Servidor RestAPI regresa un estatus 200 (ver FA1, EX1) |
| **Flujos Alternos:** | FA1.-Bad Request-No se pudo encontrar la ruta   1. El Servidor RestAPI regresa un estatus 404 |
| **Excepciones:** | EX1.- Internal Server Error   1. No se pudo conectar con la base de datos 2. El Servidor RestAPI regresa un estatus 500 |
| **Poscondiciones:** | POST1.- El Servidor RestAPI regresara una lista de publicaciones |

### Eliminar contenido

|  |  |
| --- | --- |
| **­­ID:** | CU102 |
| **Nombre:** | Eliminar contenido |
| **Autor(es):** | Jethran Enrique Gómez San Gabriel |
| **Fecha de creación:** | 05/11/2019 |
| **Actor(es):** | Servidor RestAPI |
| **Descripción:** | El Servidor RestAPI elimina una publicación de la base de datos |
| **Precondiciones:** | PRE1.- Deberá estar autenticado anteriormente |
| **Flujo Normal:** | 1. El Servidor RestAPI hace una petición DELETE de una **publicación** con el **id** de la publicación 2. Verifica que el atributo **id** no este vacío 3. El Servidor RestAPI elimina la **publicación** de la base de datos 4. El Servidor RestAPI regresa un estatus 200 (ver FA1, EX1) |
| **Flujos Alternos:** | FA1.- Bad Request   1. El Servidor RestAPI regresa un estatus 400 |
| **Excepciones:** | EX1.- Internal Server Error   1. No se pudo conectar con la base de datos 2. El Servidor RestAPI regresa un estatus 500 |
| **Poscondiciones:** | POST1.- El Servidor RestAPI elimina una publicación |

### Consultar cuentas

|  |  |
| --- | --- |
| **­­ID:** | CU103 |
| **Nombre:** | Consultar cuentas |
| **Autor(es):** | Jethran Enrique Gómez San Gabriel |
| **Fecha de creación:** | 05/11/2019 |
| **Actor(es):** | Servidor RestAPI |
| **Descripción:** | El Servidor RestAPI recupera todas las cuentas registradas en la base de datos |
| **Precondiciones:** | PRE1.- Deberá estar autenticado anteriormente |
| **Flujo Normal:** | 1. El Servidor RestAPI hace una petición GET de todas las cuentas 2. El Servidor RestAPI recupera todas las **cuentas** de la base de datos 3. El Servidor RestAPI regresa un estatus 200 (ver FA1, EX1) |
| **Flujos Alternos:** | FA1.-Bad Request-Error al encontrar la ruta   1. El Servidor RestAPI regresa un estatus 404 |
| **Excepciones** | EX1.- Internal Server Error   1. No se pudo conectar con la base de datos 2. El Servidor RestAPI regresa un estatus 500 |
| **Poscondiciones:** | POST1.- El Servidor RestAPI regresara una lista de cuentas |
| **Extiende:** | CU104.- Suspender cuenta  CU105.- Activar cuenta |

### Suspender cuenta

|  |  |
| --- | --- |
| **­­ID:** | CU104 |
| **Nombre:** | Suspender cuenta |
| **Autor(es):** | Jethran Enrique Gómez San Gabriel |
| **Fecha de creación:** | 05/11/2019 |
| **Actor(es):** | Servidor RestAPI |
| **Descripción:** | El Servidor RestAPI suspenderá una cuenta |
| **Precondiciones:** | PRE1.- Deberá estar autenticado anteriormente  PRE2.- Consultar las cuentas |
| **Flujo Normal:** | 1. El Servidor RestAPI hace una petición PUT hacia la **Cuenta** con el **id** de la cuenta 2. Verifica que el atributo **id** no este vacío 3. El Servidor RestAPI edita el campo de cuenta activa a suspendida en la base datos 4. El Servidor RestAPI regresa un estatus 200 (ver FA1, EX1) |
| **Flujos Alternos:** | FA1: Bad Request   1. El atributo se encuentra vacío o esta erróneo 2. El Servidor RestAPI regresa un estatus 400 |
| **Excepciones** | EX1.- Internal Server Error   1. No se pudo conectar con la base de datos 2. El Servidor RestAPI regresa un estatus 500 |
| **Poscondiciones:** | POST1.- El Servidor RestAPI suspende una cuenta en la base de datos |

### Activar cuenta

|  |  |
| --- | --- |
| **­­ID:** | CU105 |
| **Nombre:** | Activar cuenta |
| **Autor(es):** | Jethran Enrique Gómez San Gabriel |
| **Fecha de creación:** | 05/11/2019 |
| **Actor(es):** | Servidor RestAPI |
| **Descripción:** | El Servidor RestAPI podrá activar una cuenta suspendida |
| **Precondiciones:** | PRE1.- Deberá estar autenticado anteriormente  PRE2.- Consultar las cuentas |
| **Flujo Normal:** | 1. El Servidor RestAPI recibe una petición PUT hacia la **Cuenta** suspendida con el **id** de la cuenta 2. Verifica que el atributo id no este vacío y realiza la activación de la cuenta 3. El Servidor RestAPI regresa un estatus 200 (ver FA1, EX1) |
| **Flujos Alternos:** | FA1: Bad Request- Atributos vacíos   1. El Servidor RestAPI regresa un estatus 400 |
| **Excepciones:** | EX1.- Internal Sever Error   1. No se pudo conectar con la base de datos 2. El Servidor RestApi regresa un estatus |
| **Poscondiciones:** | POST1.- El Servidor RestAPI activa una cuenta |

### Registrar nuevo moderador

|  |  |
| --- | --- |
| **­­ID:** | SCU106 |
| **Nombre:** | Registro de nuevo moderador |
| **Autor(es):** | José Alberto Espinoza Castillo |
| **Fecha de creación:** | 06/11/2019 |
| **Fecha de actualización:** | 06/11/2019 |
| **Actor(es):** | Usuario sin cuenta. En adelante será referenciado como Usuario para este CU. |
| **Descripción:** | Permitirá añadir un nuevo evento al sistema |
| **Precondiciones:** | Pre01.- No hay un usuario con el mismo nombre de usuario registrado  Pre02.- No hay un usuario con el mismo correo registrado |
| **Flujo Normal:** | 1. El Servidor RestAPI recibe una petición POST para registrar un nuevo moderador, con el siguiente “body”:    1. Nombre    2. Apellido    3. Nombre usuario    4. Correo electrónico    5. Numero de celular    6. Contraseña 2. El Servidor RestAPI verifica que los atributos del body no se encuentren vacíos y los almacena en la Base de datos 3. El Servidor RestAPI retorna un código 201 y termina caso de uso |
| **Flujos Alternos:** | **FA-1. Existen atributos del body que están vacíos**   1. El Servidor RestAPI retorna un código 400, con el mensaje “parámetros inválidos o incompletos”. 2. Termina caso de uso. |
| **Excepciones:** | **EX-1. No se ha podido registrar la cuenta de moderador en el Sistema.**   1. El Servidor RestAPI retorna un código 500, con el mensaje “No se pudo ejecutar el registro”. 2. Termina caso de uso. |
| **Postcondiciones:** | PosCO01 – Si el flujo concluyo normalmente debe haber una CUENTA registrada. |

### Autenticación de usuario

|  |  |
| --- | --- |
| **­­ID:** | SCU21 |
| **Nombre:** | Autenticar usuario |
| **Autor(es):** | José Alberto Espinoza Castillo |
| **Fecha de creación:** | 06/11/2019 |
| **Fecha de actualización:** | 06/11/2019 |
| **Actor(es):** | Servidor RestAPI |
| **Descripción:** | Permitirá a un usuario acceder al Sistema |
| **Precondiciones:** | Pre01.- No esta previamente autenticado en el Sistema |
| **Flujo Normal:** | 1. El Servidor RestAPI recibe una petición GET para buscar credenciales de usuario, con el siguiente “body”:    1. Correo electrónico    2. Contraseña 2. El Servidor RestAPI verifica que los atributos del body no sean nulos y realiza la búsqueda de las credenciales ingresadas 3. El Servidor regresa un código 200 y retorna el token de autenticación |
| **Flujos Alternos:** | **FA-1. Existen atributos del body que están vacíos**   1. El Servidor RestAPI retorna un código 400, con el mensaje “parámetros inválidos o incompletos”. 2. Termina caso de uso. |
| **Excepciones:** | **EX-1. No se ha podido realizar la consulta**   1. El Servidor RestAPI retorna un código 500, con el mensaje “No se pudo ejecutar la consulta solicitada”. 2. Termina caso de uso. |
| **Postcondiciones:** | PosCO01 – Si el flujo concluyo normalmente debe haber una CUENTA registrada. |

# Requerimientos de datos

<This section describes various aspects of the data that the system will consume as inputs, process in some fashion, or create as outputs.>

## Modelo de datos relacional

<A data model is a visual representation of the data objects and collections the system will process and the relationships between them. Include a data model for the business operations being addressed by the system, or a logical representation for the data that the system itself will manipulate. Data models are most commonly created as an entity-relationship diagram.>

## Diccionario de datos

<The data dictionary defines the composition of data structures and the meaning, data type, length, format, and allowed values for the data elements that make up those structures. In many cases, you're better off storing the data dictionary as a separate artifact, rather than embedding it in the middle of an SRS. That also increases its reusability potential in other projects.>

## Reportes

<If your application will generate any reports, identify them here and describe their characteristics. If a report must conform to a specific predefined layout you can specify that here as a constraint, perhaps with an example. Otherwise, focus on the logical descriptions of the report content, sort sequence, totaling levels, and so forth, deferring the detailed report layout to the design stage.>

## Data Acquisition, Integrity, Retention, and Disposal

<If relevant, describe how data is acquired and maintained. State any requirements regarding the need to protect the integrity of the system's data. Identify any specific techniques that are necessary, such as backups, checkpointing, mirroring, or data accuracy verification. State policies the system must enforce for either retaining or disposing of data, including temporary data, metadata, residual data (such as deleted records), cached data, local copies, archives, and interim backups.>

# External Interface Requirements

<This section provides information to ensure that the system will communicate properly with users and with external hardware or software elements.>

## User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

## Software Interfaces

<Describe the connections between this product and other software components (identified by name and version), including other applications, databases, operating systems, tools, libraries, websites, and integrated commercial components. State the purpose, formats, and contents of the messages, data, and control values exchanged between the software components. Specify the mappings of input and output data between the systems and any translations that need to be made for the data to get from one system to the other. Describe the services needed by or from external software components and the nature of the intercomponent communications. Identify data that will be exchanged between or shared across software components. Specify nonfunctional requirements affecting the interface, such as service levels for responses times and frequencies, or security controls and restrictions.>

## Hardware Interfaces

<Describe the characteristics of each interface between the software and hardware (if any) components of the system. This description might include the supported device types, the data and control interactions between the software and the hardware, and the communication protocols to be used. List the inputs and outputs, their formats, their valid values or ranges, and any timing issues developers need to be aware of. If this information is extensive, consider creating a separate interface specification document.>

## Communications Interfaces

<State the requirements for any communication functions the product will use, including e-mail, Web browser, network protocols, and electronic forms. Define any pertinent message formatting. Specify communication security or encryption issues, data transfer rates, handshaking, and synchronization mechanisms. State any constraints around these interfaces, such as whether e-mail attachments are acceptable or not.>

# Quality Attributes

## Usability

<Specify any requirements regarding characteristics that will make the software appear to be “user-friendly.” Usability encompasses ease of use, ease of learning; memorability; error avoidance, handling, and recovery; efficiency of interactions; accessibility; and ergonomics. Sometimes these can conflict with each other, as with ease of use over ease of learning. Indicate any user interface design standards or guidelines to which the application must conform.>

## Performance

<State specific performance requirements for various system operations. If different functional requirements or features have different performance requirements, it's appropriate to specify those performance goals right with the corresponding functional requirements, rather than collecting them in this section.>

## Security

<Specify any requirements regarding security or privacy issues that restrict access to or use of the product. These could refer to physical, data, or software security. Security requirements often originate in business rules, so identify any security or privacy policies or regulations to which the product must conform. If these are documented in a business rules repository, just refer to them.>

## Safety

<Specify requirements that are concerned with possible loss, damage, or harm that could result from use of the product. Define any safeguards or actions that must be taken, as well as potentially dangerous actions that must be prevented. Identify any safety certifications, policies, or regulations to which the product must conform.>

## [Others as relevant]

<Create a separate section in the SRS for each additional product quality attribute to describe characteristics that will be important to either customers or developers. Possibilities include availability, efficiency, installability, integrity, interoperability, modifiability, portability, reliability, reusability, robustness, scalability, and verifiability. Write these to be specific, quantitative, and verifiable. Clarify the relative priorities for various attributes, such as security over performance.>

# Internationalization and Localization Requirements

<Internationalization and localization requirements ensure that the product will be suitable for use in nations, cultures, and geographic locations other than those in which it was created. Such requirements might address differences in: currency; formatting of dates, numbers, addresses, and telephone numbers; language, including national spelling conventions within the same language (such as American versus British English), symbols used, and character sets; given name and family name order; time zones; international regulations and laws; cultural and political issues; paper sizes used; weights and measures; electrical voltages and plug shapes; and many others.>

# Other Requirements

<Examples are: legal, regulatory or financial compliance, and standards requirements; requirements for product installation, configuration, startup, and shutdown; and logging, monitoring and audit trail requirements. Instead of just combining these all under "Other," add any new sections to the template that are pertinent to your project. Omit this section if all your requirements are accommodated in other sections. >

Appendix A: Glossary

<Define any specialized terms that a reader needs to know to understand the SRS, including acronyms and abbreviations. Spell out each acronym and provide its definition. Consider building a reusable enterprise-level glossary that spans multiple projects and incorporating by reference any terms that pertain to this project.>

Appendix B: Analysis Models

<This optional section includes or points to pertinent analysis models such as data flow diagrams, feature trees, state-transition diagrams, or entity-relationship diagrams. You might prefer to insert certain models into the relevant sections of the specification instead of collecting them at the end.>