

2021

# Documentación Ducker



Universidad Veracruzana  
Desarrollo de Aplicaciones  
Equipo 1



# Índice

<b>1. Introducción.....</b>	<b>2</b>
<b>2. Requerimientos.....</b>	<b>2</b>
Contexto.....	2
Clases de usuario.....	3
Requerimientos funcionales.....	3
Requerimientos no funcionales.....	4
<b>3. Análisis y diseño .....</b>	<b>5</b>
Casos de uso.....	5
Diagramas de actividad .....	20
Diagrama de clases .....	25
Diagrama E R.....	26
Prototipo UI Cliente .....	27
<b>4. Construcción.....</b>	<b>32</b>
Selección de lenguajes de programación .....	32
Estándar de codificación.....	32
Selección de tecnologías .....	32
<b>5. Anexos.....</b>	<b>33</b>



# Introducción

## Requerimientos

### Contexto

Bajo las especificaciones dadas en las experiencias educativas de “Desarrollo de sistemas en red” y “Desarrollo de aplicaciones” se nos pidió elaborar un sistema que fungiera como una red social para los integrantes de la Facultad de estadística e informática, este sistema tendría características similares a las encontradas en la plataforma social de “Twitter” ya que podría proporcionar interacciones con un flujo e acción similar a este, tener la opción de seguir a una cuenta para que el sistema pueda compartir su actividad contigo, dar “like”, hacer un retweet o crear un hashtag para etiquetar las publicaciones hechas por una cuenta.

El sistema derivado de las especificaciones dadas por las experiencias educativas anteriormente mencionadas, y desarrollado por el equipo anteriormente mencionado en la portada de este documento, se ha denominado como “Ducker” el cual ha sido construido usando tecnologías API REST como la solución al problema de la implementación y construcción de la API usada para el mismo esto con el propósito de la que API pueda ser usada en un ambiente multi-entorno (multi-plataforma) para brindar un aspecto fuerte de portabilidad en el sistema.

## Clases de usuario



## Requerimientos no funcionales

- El servidor debe de ejecutarse en una distribución de Linux.
- Utilizar únicamente tecnologías contempladas en el programa del curso (Web API, Sockets, RPC).



# Análisis y diseño

## Casos de uso

<b>ID:</b>	CU-01-RegistrarUsuario
<b>Nombre:</b>	RegistrarUsuario
<b>Autor(es):</b>	Equipo 1
<b>Descripción:</b>	El <b>Usuario</b> registra una nueva cuenta en la base de API
<b>Actor(es):</b>	<b>Usuario</b>
<b>Frecuencia:</b>	Por lo menos una vez al día
<b>Disparador:</b>	El <b>Usuario</b> da click en "Crear cuenta"
<b>Precondiciones:</b>	No existe una precondición
<b>Flujo Normal:</b>	<ol style="list-style-type: none"><li>1. El <b>Sistema</b> muestra los campos a llenar en la ventana "xml_registro_usuario" con los botones "Registrar" y "Cancelar" (Ver EX1)</li><li>2. El <b>Usuario</b> ingresa los datos requeridos en los textField correspondientes y da clic en el botón "Registrar" (Ver FA1)</li><li>3. El <b>Sistema</b> manda la información ingresada a la <b>API</b> (Ver EX1)</li><li>4. La <b>API</b> responde exitosamente (Ver FA2) (Ver FA3) (Ver EX1)</li><li>5. El <b>Sistema</b> crea una nueva cuenta con los datos del Usuario, muestra un mensaje de éxito y muestra el menú principal</li><li>6. Fin del caso de uso</li></ol>
<b>Flujos Alternos:</b>	<p>FA1.- El <b>Usuario</b> cancela el registro de la cuenta</p> <p>1.1.- El <b>Usuario</b> da clic en el botón "Cancelar" y" y regresa al paso 1 del Flujo Normal</p> <p>FA2.- El <b>Usuario</b> dejo campos vacíos</p> <p>2.1.- El <b>Sistema</b> muestra el mensaje "Se han dejado campos vacíos. Favor de llenar todos los campos"</p> <p>2.2.- El <b>Sistema</b> regresa a la ultima instancia del paso 2 del Flujo normal</p>
<b>Excepciones:</b>	<p>EX1.- El <b>Sistema</b> no pudo comunicarse con la API</p> <p>1.- El <b>Sistema</b> muestra el mensaje "Ocurrió un problema, intente más tarde."</p> <p>2.- El <b>Sistema</b> regresa a la ventana anterior.</p> <p>3.- Termina el caso de uso.</p>
<b>Postcondiciones:</b>	<p>POST1.- El <b>Sistema</b> se comunicó exitosamente con la API y guardó un nuevo Usuario</p> <p>POST2.- El <b>Sistema</b> muestra la página de inicio del <b>Usuario</b></p>



<b>ID:</b>	CU-02-IniciarSesión
<b>Nombre:</b>	IniciarSesión
<b>Autor(es):</b>	Equipo 1
<b>Descripción:</b>	El <b>Usuario</b> inicia sesión en el sistema
<b>Actor(es):</b>	<b>Usuario</b>
<b>Frecuencia:</b>	Por lo menos una vez al día
<b>Disparador:</b>	Se ingresa al menú principal de la aplicación
<b>Precondiciones:</b>	El <b>Usuario</b> se debió de haber registrado previamente
<b>Flujo Normal:</b>	<ol style="list-style-type: none"><li>1. El <b>Sistema</b> muestra los campos a llenar en la ventana "xml_login" con los botones "Iniciar sesión" y "Registrar Cuenta" (Ver EX1)</li><li>2. El <b>Usuario</b> ingresa los datos requeridos en los textField correspondientes y da clic en el botón "Iniciar sesión" (Ver FA1)</li><li>3. El <b>Sistema</b> manda la información ingresada a la <b>API</b> (Ver EX1)</li><li>4. La <b>API</b> responde exitosamente (Ver FA2) (Ver FA3) (Ver FA4) (Ver EX1)</li><li>5. El <b>Sistema</b> muestra el mensaje de éxito</li><li>6. Fin del caso de uso</li></ol>
<b>Flujos Alternos:</b>	<p>FA1.- El <b>Usuario</b> quiere registrar un CUENTA</p> <p>1.1.- El <b>Sistema</b> redirecciona al <b>Usuario</b> a la ventana de registro de cuenta</p> <p>1.2.- El flujo alternativo sigue en el primer paso del Flujo normal del CU-01</p> <p>FA2.- El <b>Usuario</b> dejó campos vacíos</p> <p>2.1.- El <b>Sistema</b> muestra el mensaje "Se han dejado campos vacíos. Favor de llenar todos los campos"</p> <p>2.3.- El <b>Sistema</b> regresa a la última instancia del paso 2 del Flujo normal</p> <p>FA3.- El <b>Usuario</b> ingreso información incorrecta</p> <p>3.1.- El <b>Sistema</b> muestra el mensaje "Contraseña o correo incorrectos. Favor de revisar los campos"</p> <p>3.3.- El <b>Sistema</b> regresa a la última instancia del paso 2 del Flujo normal</p>
<b>Excepciones:</b>	<p>EX1.- El <b>Sistema</b> no pudo comunicarse con la API</p> <p>1.- El <b>Sistema</b> muestra el mensaje "Ocurrió un problema, intente más tarde."</p> <p>2.- El <b>Sistema</b> regresa a la ventana anterior.</p> <p>4.- Termina el caso de uso.</p>



<b>Postcondiciones:</b>	POST1.- El <b>Sistema</b> muestra la pagina de inicio del <b>Usuario</b>
-------------------------	--





<b>ID:</b>	CU-03-PublicarQuack
<b>Nombre:</b>	PublicarQuack
<b>Autor(es):</b>	Equipo 1
<b>Descripción:</b>	El <b>Usuario</b> publica un QUACK desde su CUENTA
<b>Actor(es):</b>	<b>Usuario</b>
<b>Frecuencia:</b>	Por lo menos una vez al día
<b>Disparador:</b>	Se da clic en el botón "Nuevo quack"
<b>Precondiciones:</b>	El <b>Usuario</b> tiene su CUENTA iniciada
<b>Flujo Normal:</b>	<ol style="list-style-type: none"><li>1. El <b>Sistema</b> muestra los campos a llenar en la ventana "xml_nuevo_quack" con los botones "Publicar" y "Cancelar" (Ver EX1)</li><li>2. El <b>Usuario</b> ingresa los datos requeridos en los textField correspondientes y da clic en el botón "Publicar" (Ver FA1)</li><li>3. El <b>Sistema</b> manda la información ingresada a la <b>API</b> (Ver EX1)</li><li>4. La <b>API</b> responde exitosamente (Ver FA2) (Ver EX1)</li><li>5. El <b>Sistema</b> muestra el mensaje de éxito</li><li>6. El <b>Usuario</b> clic en el mensaje de éxito</li><li>7. Fin del caso de uso</li></ol>
<b>Flujos Alternos:</b>	<p>FA1.- El <b>Usuario</b> cancela la publicación de QUACK</p> <p>1.1.- El <b>Sistema</b> muestra el mensaje "¿Desea cancelar el quack?" y los botones "Sí" y "No"</p> <p>1.2.- El <b>Usuario</b> da clic en el botón "Sí" y termina el caso de uso, si no, da clic en el botón "no" y regresa al paso 1 del Flujo Normal</p> <p>FA2.- El <b>Usuario</b> dejo campos vacíos</p> <p>2.1.- El <b>Sistema</b> muestra el mensaje "Se han dejado campos vacíos. Favor de llenar todos lo campos" y el botón "Aceptar"</p> <p>2.2.- El <b>Usuario</b> da clic en "Aceptar"</p> <p>2.3.- El <b>Sistema</b> regresa a la ultima instancia del paso 2 del Flujo normal</p>
<b>Excepciones:</b>	<p>EX1.- El <b>Sistema</b> no pudo comunicarse con la API</p> <p>1.- El <b>Sistema</b> muestra el mensaje "Ocurrió un problema, intente más tarde." Y el botón "Aceptar"</p> <p>2.- El <b>Usuario</b> selecciona el botón "Aceptar".</p> <p>3.- El <b>Sistema</b> regresa a la ventana anterior.</p> <p>4.- Termina el caso de uso.</p>



<b>Postcondiciones:</b>	POST1.- El <b>Sistema</b> se comunicó exitosamente con la API y publico el nuevo QUACK  POST2.- El <b>Sistema</b> muestra la pagina de inicio del <b>Usuario</b>
-------------------------	--

<b>ID:</b>	CU-05-VerSeguidores
<b>ID:</b>	CU-04-VerSeguidos
<b>Nombre:</b>	VerSeguidores
<b>Autor(es):</b>	Equipo 1
<b>Descripción:</b>	El <b>Usuario</b> ingresa a su apartado de "Seguidores" para ver los <b>Usuarios</b> que lo siguen
<b>Nombre:</b>	VerSeguidos
<b>Actor(es):</b>	<b>Usuario</b>
<b>Autor(es):</b>	Equipo 1
<b>Frecuencia:</b>	Por lo menos una vez al día
<b>Disparador:</b>	Se ingresa al perfil de <b>Usuario</b>
<b>Precondiciones:</b>	El <b>Usuario</b> debe de tener por lo menos a un <b>Usuario</b> más como seguidor
<b>Flujo Normal:</b>	<ol style="list-style-type: none"><li>1. El <b>Sistema</b> muestra la ventana "xml_perfil" (Ver EX1)</li><li>2. El <b>Usuario</b> da clic en el apartado</li></ol>
<b>Descripción:</b>	El <b>Usuario</b> ingresa al apartado de "Seguidos" para ver los <b>Usuarios</b> que lo siguen
<b>Actor(es):</b>	<b>Usuario</b>
<b>Frecuencia:</b>	Por lo menos una vez al día
<b>Flujos Alternos:</b>	No existen flujos alternos para este caso de uso
<b>Disparador:</b>	Se ingresa al perfil de <b>Usuario</b>
<b>Precondiciones:</b>	El <b>Usuario</b> debe seguir por lo menos a un <b>Usuario</b> más
<b>Excepciones:</b>	EX1 - El <b>Sistema</b> no pudo comunicarse con la API
<b>Flujo Normal:</b>	<ol style="list-style-type: none"><li>1. El <b>Sistema</b> muestra la ventana "xml_perfil" (Ver EX1)</li><li>1.- El <b>Sistema</b> muestra el mensaje "Ocurrió un problema, intente más tarde."</li><li>2. El <b>Usuario</b> da clic en el apartado "Seguidos"</li><li>3. El <b>Sistema</b> manda la petición a la API (Ver EX1)</li><li>2.- El <b>Usuario</b> selecciona el botón "Aceptar".</li><li>4. La API responde exitosamente (Ver EX1)</li><li>3.- El <b>Sistema</b> regresa a la ventana anterior.</li><li>3. El <b>Sistema</b> muestra la lista de seguidos</li><li>4. Termina el caso de uso.</li><li>6. Fin del caso de uso</li></ol>
<b>Flujos Alternos:</b>	No existen flujos alternos para este caso de uso
<b>Postcondiciones:</b>	No existen postcondiciones para este caso de uso
<b>Excepciones:</b>	EX1.- El <b>Sistema</b> no pudo comunicarse con la API
	<ol style="list-style-type: none"><li>1.- El <b>Sistema</b> muestra el mensaje "Ocurrió un problema, intente más tarde."</li><li>2.- El <b>Sistema</b> regresa a la ventana anterior.</li><li>4.- Termina el caso de uso.</li></ol>
<b>Postcondiciones:</b>	No existen precondiciones para este caso de uso

<b>Nombre:</b>	CU-06: Responder Quack
<b>Autor:</b>	Equipo 1
<b>Fecha creación:</b>	5 de noviembre de 2021
<b>Fecha actualización:</b>	7 de diciembre de 2021
<b>Descripción:</b>	Permite que el <b>Usuario</b> responda un QUACK desde su CUENTA.
<b>Actor:</b>	<b>Usuario, API</b>
<b>Disparador:</b>	El <b>Usuario</b> da clic en el botón “Responder quack” dentro de la vista “feed.xml” o “feed_descubrir.xml”.
<b>Precondiciones:</b>	Pre-1. Debe existir una sesión iniciada.
<b>Flujo normal:</b>	<ol style="list-style-type: none"> <li>1. El Sistema muestra la vista “quack_respuesta.xml” donde se muestran los datos del usuario al que se está respondiendo y solicita el texto de respuesta junto con los botones “Publicar”, “Regresar”, “Agregar archivo multimedia” y “Dar like”.</li> <li>2. El <b>Usuario</b> ingresa la información y da clic en el botón “Publicar”. (Ver FA1) (Ver FA2) (Ver FA3)</li> <li>3. El Sistema manda la información ingresada a la API. (Ver EX1)</li> <li>4. La <b>API</b> responde exitosamente.</li> <li>5. El Sistema muestra la RESPUESTA al QUACK de manera exitosa.</li> <li>6. Fin de caso de uso.</li> </ol>
<b>Flujos alternos:</b>	<p>FA 1: Cancelar acción.</p> <ol style="list-style-type: none"> <li>1. El <b>Usuario</b> da clic en el botón “Regresar”.</li> <li>2. El Sistema muestra el diálogo “Dialogo_Confirmacion” con el mensaje “¿Seguro que desea cancelar?” No se guardará ningún cambio.” junto con botones “Sí” y “No”.</li> <li>3. Si el <b>Usuario</b> da clic en el botón “Sí” termina el caso de uso, sino vuelve al FN-2 con la información ya ingresada anteriormente.</li> </ol> <p>FA 2: Campos vacíos.</p> <ol style="list-style-type: none"> <li>1. El <b>Usuario</b> no llena todos los campos solicitados obligatorios.</li> <li>2. El Sistema muestra el diálogo “Dialogo_CamposVacios” con el mensaje "Debe llenar todos los campos solicitados obligatorios". junto con el botón "Aceptar".</li> <li>3. El <b>Usuario</b> da clic en el botón “Aceptar”.</li> <li>4. El Sistema regresa al FN-1 con la información ya ingresada anteriormente.</li> </ol> <p>FA 3: Dar like al Quack.</p> <ol style="list-style-type: none"> <li>1. El <b>Usuario</b> da clic en el botón “Dar like”.</li> <li>2. El Sistema manda la información ingresada a la API. (Ver EX1)</li> <li>3. La <b>API</b> responde exitosamente.</li> <li>4. El Sistema muestra el LIKE en el QUACK de manera exitosa y regresa al FN-1.</li> </ol>
<b>Excepciones:</b>	<p>EX 1. Error al comunicarse con la API.</p> <ol style="list-style-type: none"> <li>1. El Sistema muestra el mensaje “Ocurrió un problema, intente más tarde.” Junto con el botón “Aceptar”.</li> <li>2. El <b>Usuario</b> selecciona el botón “Aceptar”.</li> <li>3. El Sistema regresa a la ventana anterior.</li> <li>4. Termina el caso de uso.</li> </ol>
<b>Postcondiciones:</b>	Post-1: El Sistema guardó una RESPUESTA al QUACK en la base de datos.



<b>Incluye:</b>	No incluye.
<b>Extiende:</b>	No extiende.



<b>Nombre:</b>	CU-07: Visualizar Quack
<b>Autor:</b>	Equipo 1
<b>Fecha creación:</b>	5 de noviembre de 2021
<b>Fecha actualización:</b>	7 de diciembre de 2021
<b>Descripción:</b>	Permite que el <b>Usuario</b> visualice un QUACK desde su CUENTA.
<b>Actor:</b>	<b>Usuario, API</b>
<b>Disparador:</b>	El <b>Usuario</b> da clic en un QUACK dentro de la vista “feed.xml” o “feed_descubrir.xml”.
<b>Precondiciones:</b>	Pre-1. Debe existir una sesión iniciada.
<b>Flujo normal:</b>	<ol style="list-style-type: none"><li>1. El Sistema se comunica con la <b>API</b> para mostrar la información solicitada. (Ver EX1)</li><li>2. La <b>API</b> responde exitosamente.</li><li>3. El Sistema muestra los datos del QUACK seleccionado junto con sus RESPUESTA.</li><li>4. El <b>Usuario</b> visualiza la información y da clic en el botón “Regresar”. (Ver FA1)</li><li>5. El Sistema muestra la información QUACK de manera exitosa.</li><li>6. Fin de caso de uso.</li></ol>
<b>Flujos alternos:</b>	FA 1: Dar like al Quack. <ol style="list-style-type: none"><li>1. El <b>Usuario</b> da clic en el botón “Dar like”.</li><li>2. El Sistema manda la información ingresada a la API. (Ver EX1)</li><li>3. La <b>API</b> responde exitosamente.</li><li>4. El Sistema muestra el LIKE en el QUACK de manera exitosa y regresa al FN-3.</li></ol>
<b>Excepciones:</b>	EX 1. Error al comunicarse con la API. <ol style="list-style-type: none"><li>1. El Sistema muestra el mensaje “Ocurrió un problema, intente más tarde.” Junto con el botón “Aceptar”.</li><li>2. El <b>Usuario</b> selecciona el botón “Aceptar”.</li><li>3. El Sistema regresa a la ventana anterior.</li><li>4. Termina el caso de uso.</li></ol>
<b>Postcondiciones:</b>	Post-1: El Sistema mostró un QUACK en la interfaz del usuario.
<b>Incluye:</b>	No incluye.
<b>Extiende:</b>	No extiende.



<b>Nombre:</b>	CU-08: Modificar perfil
<b>Autor:</b>	Equipo 1
<b>Fecha creación:</b>	5 de noviembre de 2021
<b>Fecha actualización:</b>	7 de diciembre de 2021
<b>Descripción:</b>	Permite que el <b>Usuario</b> modifique la información de su perfil.
<b>Actor:</b>	<b>Usuario, API</b>
<b>Disparador:</b>	El <b>Usuario</b> da clic en el botón “Editar” dentro de la vista “perfil_usuario.xml”
<b>Precondiciones:</b>	Pre-1. Debe existir una sesión iniciada.
<b>Flujo normal:</b>	<ol style="list-style-type: none"><li>1. El Sistema obtiene los datos del perfil y muestra la vista “modificación_de_usuario.xml” donde se muestran los datos del usuario y los campos que se permiten modificar junto con los botones “Cambiar contraseña”, “Eliminar cuenta”, “Guardar” y “Cancelar”. (Ver EX 1)</li><li>2. El <b>Usuario</b> ingresa la nueva información y da clic en el botón “Guardar”. (Ver FA1) (Ver FA2) (Ver FA3) (Ver FA3) (Ver FA3)</li><li>3. El Sistema manda la información ingresada a la API. (Ver EX1)</li><li>4. La <b>API</b> responde exitosamente.</li><li>5. El Sistema guarda los nuevos datos exitosamente y muestra un diálogo de éxito junto con el botón “Aceptar”.</li><li>6. El <b>Usuario</b> da clic en el botón “Aceptar”.</li><li>7. Fin de caso de uso.</li></ol>
<b>Flujos alternos:</b>	<p>FA 1: Cancelar acción</p> <ol style="list-style-type: none"><li>1. El <b>Usuario</b> da clic en el botón “Regresar”.</li><li>2. El Sistema muestra el diálogo “Dialogo_Confirmacion” con el mensaje “¿Seguro que desea cancelar?” No se guardará ningún cambio.” junto con botones “Sí” y “No”.</li><li>3. Si el <b>Usuario</b> da clic en el botón “Sí” termina el caso de uso, sino vuelve al FN-2 con la información ya ingresada anteriormente.</li></ol> <p>FA 2: Campos vacíos</p> <ol style="list-style-type: none"><li>1. El <b>Usuario</b> no llena todos los campos solicitados obligatorios.</li><li>2. El Sistema muestra el diálogo “Dialogo_CamposVacios” con el mensaje "Debe llenar todos los campos solicitados obligatorios". junto con el botón "Aceptar".</li><li>3. El <b>Usuario</b> da clic en el botón “Aceptar”.</li><li>4. El Sistema regresa al FN-1 con la información ya ingresada anteriormente.</li></ol> <p>FA 3: Información ingresada inválida.</p> <ol style="list-style-type: none"><li>1. El <b>Usuario</b> ingresa información inválida en los campos solicitados.</li><li>2. El Sistema muestra el diálogo “Dialogo_InformacionInvalida” con el mensaje "Se ha ingresado información no válida, por favor ingrese los datos correctamente". junto con el botón "Aceptar".</li><li>3. El <b>Usuario</b> da clic en el botón “Aceptar”.</li><li>4. El Sistema regresa al FN-1 con la información ya ingresada anteriormente.</li></ol> <p>FA 4: Cambiar contraseña</p> <ol style="list-style-type: none"><li>1. El <b>Usuario</b> da clic en el botón “Cambiar contraseña”.</li></ol>



	<ol style="list-style-type: none"><li>2. El Sistema muestra el campo para ingresar la nueva contraseña junto con el botón “Aceptar”.</li><li>3. El <b>Usuario</b> ingresa la información solicitada y da clic en el botón “Aceptar”.</li><li>4. El Sistema regresa al FN-1 con la información ya ingresada anteriormente.</li></ol> <p>FA 5: Eliminar cuenta</p> <ol style="list-style-type: none"><li>1. El <b>Usuario</b> da clic en el botón “Eliminar cuenta”.</li><li>2. El Sistema muestra el diálogo “Dialogo_Confirmacion” con el mensaje “¿Seguro que desea eliminar su cuenta? Este cambio es irreversible” junto con botones “Sí” y “No”.</li><li>3. Si el <b>Usuario</b> da clic en el botón “Sí” la CUENTA se elimina y termina el caso de uso. Si da clic en el botón “No” el Sistema regresa al FN-1.</li></ol>
<b>Excepciones:</b>	<p>EX 1. Error al comunicarse con la API.</p> <ol style="list-style-type: none"><li>1. El Sistema muestra el mensaje “Ocurrió un problema, intente más tarde.” Junto con el botón “Aceptar”.</li><li>2. El <b>Usuario</b> selecciona el botón “Aceptar”.</li><li>3. El Sistema regresa a la ventana anterior.</li><li>4. Termina el caso de uso.</li></ol>
<b>Postcondiciones:</b>	Post-1: El Sistema modificó un CUENTA en la base de datos.
<b>Incluye:</b>	No incluye.
<b>Extiende:</b>	No extiende.

<b>Nombre:</b>	CU-09: Visualizar perfil
<b>Autor:</b>	Equipo 1
<b>Fecha creación:</b>	5 de noviembre de 2021
<b>Fecha actualización:</b>	7 de diciembre de 2021
<b>Descripción:</b>	Permite que el <b>Usuario</b> visualice una CUENTA desde su CUENTA.
<b>Actor:</b>	<b>Usuario, API</b>
<b>Disparador:</b>	El <b>Usuario</b> da clic en una CUENTA (incluida la propia) dentro de la vista “feed.xml” o “feed_descubrir.xml” en su menú principal.
<b>Precondiciones:</b>	Pre-1. Debe existir una sesión iniciada.
<b>Flujo normal:</b>	<ol style="list-style-type: none"><li>1. El Sistema se comunica con la API para mostrar la información solicitada. (Ver EX1)</li><li>2. La <b>API</b> responde exitosamente.</li><li>3. El Sistema muestra la vista “perfil_usuario.xml” donde se muestran los datos del usuario seleccionado junto con el botón “Regresar”.</li></ol>



	<ol style="list-style-type: none"><li>4. El <b>Usuario</b> visualiza la información y da clic en el botón “Regresar”.</li><li>5. El Sistema muestra la información de la CUENTA exitosamente.</li><li>6. Fin de caso de uso.</li></ol>
<b>Flujos alternos:</b>	
<b>Excepciones:</b>	<p>EX 1. Error al comunicarse con la API.</p> <ol style="list-style-type: none"><li>1. El Sistema muestra el mensaje “Ocurrió un problema, intente más tarde.” Junto con el botón “Aceptar”.</li><li>2. El <b>Usuario</b> selecciona el botón “Aceptar”.</li><li>3. El Sistema regresa a la ventana anterior.</li><li>4. Termina el caso de uso.</li></ol>
<b>Postcondiciones:</b>	Post-1: El Sistema mostró la información de una CUENTA en la interfaz del usuario de manera exitosa.
<b>Incluye:</b>	No incluye.
<b>Extiende:</b>	No extiende.





<b>Nombre:</b>	CU-10: Seguir usuario
<b>Autor:</b>	Equipo 1
<b>Fecha creación:</b>	5 de noviembre de 2021
<b>Fecha actualización:</b>	7 de diciembre de 2021
<b>Descripción:</b>	Permite que el <b>Usuario</b> siga una CUENTA desde su CUENTA.
<b>Actor:</b>	<b>Usuario, API</b>
<b>Disparador:</b>	El <b>Usuario</b> da clic en un perfil de otra persona dentro de la vista “feed.xml” o “feed_descubrir.xml”.
<b>Precondiciones:</b>	Pre-1. Debe existir una sesión iniciada.
<b>Flujo normal:</b>	<ol style="list-style-type: none"><li>1. El Sistema se comunica con la API para mostrar la información solicitada. (Ver EX1)</li><li>2. La <b>API</b> responde exitosamente.</li><li>3. El Sistema muestra la vista “perfil_usuario.xml” donde se muestran los datos del usuario seleccionado junto con los botones “Regresar”, “Seguir” o “Dejar de seguir” según sea el caso.</li><li>4. El <b>Usuario</b> da clic en el botón “Seguir”. (VER FA1) (VER FA2)</li><li>5. El Sistema manda la información ingresada a la API. (Ver EX1)</li><li>6. La <b>API</b> responde exitosamente.</li><li>7. El Sistema guarda el SEGUIDO y SEGUIDOR de manera exitosa.</li><li>8. Fin de caso de uso.</li></ol>
<b>Flujos alternos:</b>	<p>FA 1: Cancelar acción.</p> <ol style="list-style-type: none"><li>1. El <b>Usuario</b> da clic en el botón “Regresar”.</li><li>2. El Sistema regresa a la ventana anterior.</li><li>3. Fin de caso de uso.</li></ol> <p>FA 2: Dejar de seguir cuenta.</p> <ol style="list-style-type: none"><li>1. El Sistema muestra el botón de “Dejar de seguir” en lugar de “Seguir”.</li><li>2. Si el <b>Usuario</b> da clic en el botón “Dejar de seguir” entonces va al FN-4 del CU-11: Dejar de seguir usuario. Si no, termina el caso de uso.</li></ol>
<b>Excepciones:</b>	<p>EX 1. Error al comunicarse con la API.</p> <ol style="list-style-type: none"><li>1. El Sistema muestra el mensaje “Ocurrió un problema, intente más tarde.” Junto con el botón “Aceptar”.</li><li>2. El <b>Usuario</b> selecciona el botón “Aceptar”.</li><li>3. El Sistema regresa a la ventana anterior.</li></ol>



	4. Termina el caso de uso.
<b>Postcondiciones:</b>	Post-1: El Sistema guardó un SEGUIDO y SEGUIDOR correspondientes en la base de datos.
<b>Incluye:</b>	No incluye.
<b>Extiende:</b>	No extiende.



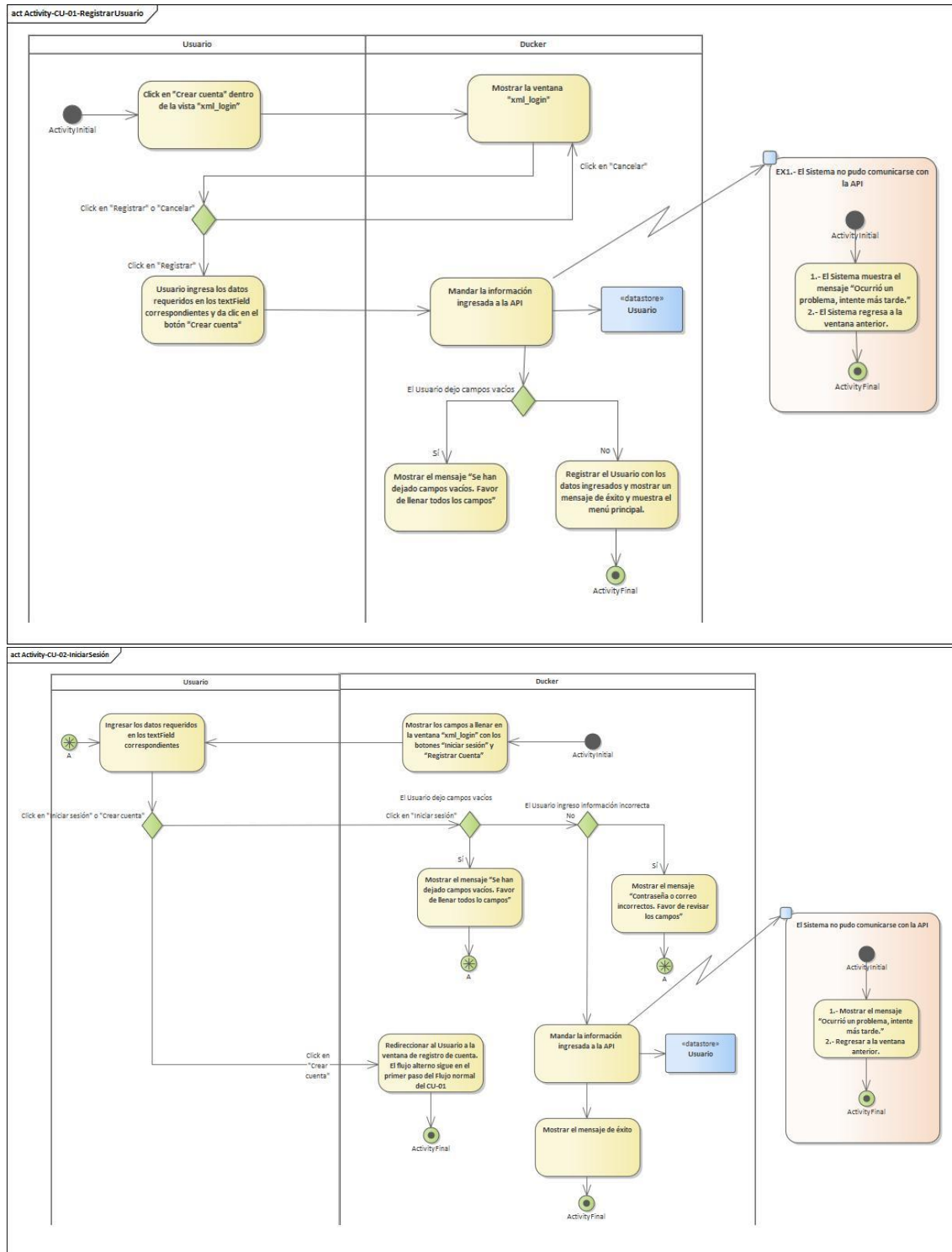
<b>Nombre:</b>	CU-11: Dejar de seguir usuario
<b>Autor:</b>	Equipo 1
<b>Fecha creación:</b>	5 de noviembre de 2021
<b>Fecha actualización:</b>	7 de diciembre de 2021
<b>Descripción:</b>	Permite que el <b>Usuario</b> deje de seguir una CUENTA desde su CUENTA.
<b>Actor:</b>	<b>Usuario, API</b>
<b>Disparador:</b>	El <b>Usuario</b> da clic en un perfil de otra persona dentro de la vista “feed.xml” o “feed_descubrir.xml”.
<b>Precondiciones:</b>	Pre-1. Debe existir una sesión iniciada.
<b>Flujo normal:</b>	<ol style="list-style-type: none"><li>1. El Sistema se comunica con la API para mostrar la información solicitada. (Ver EX1)</li><li>2. La <b>API</b> responde exitosamente.</li><li>3. El Sistema muestra la vista “perfil_usuario.xml” donde se muestran los datos del usuario seleccionado junto con los botones “Regresar”, “Seguir” o “Dejar de seguir” según sea el caso.</li><li>4. El <b>Usuario</b> da clic en el botón “Dejar de seguir”. (VER FA1) (VER FA2)</li><li>5. El Sistema manda la información ingresada a la API. (Ver EX1)</li><li>6. La <b>API</b> responde exitosamente.</li><li>7. El Sistema guarda el SEGUIDO y SEGUIDOR de manera exitosa.</li><li>8. Fin de caso de uso.</li></ol>
<b>Flujos alternos:</b>	<p>FA 1: Cancelar acción.</p> <ol style="list-style-type: none"><li>1. El <b>Usuario</b> da clic en el botón “Regresar”.</li><li>2. El Sistema regresa a la ventana anterior.</li><li>3. Fin de caso de uso.</li></ol> <p>FA 2: Seguir cuenta.</p> <ol style="list-style-type: none"><li>1. El Sistema muestra el botón de “Seguir” en lugar de “Dejar de seguir”.</li><li>2. Si el <b>Usuario</b> da clic en el botón “Seguir” entonces va al FN-4 del CU-10: Seguir usuario. Si no, termina el caso de uso.</li></ol>
<b>Excepciones:</b>	<p>EX 1. Error al comunicarse con la API.</p> <ol style="list-style-type: none"><li>1. El Sistema muestra el mensaje “Ocurrió un problema, intente más tarde.” Junto con el botón “Aceptar”.</li><li>2. El <b>Usuario</b> selecciona el botón “Aceptar”.</li></ol>



	3. El Sistema regresa a la ventana anterior. 4. Termina el caso de uso.
<b>Postcondiciones:</b>	Post-1: El Sistema guardó un SEGUIDO y SEGUIDOR correspondientes en la base de datos.
<b>Incluye:</b>	No incluye.
<b>Extiende:</b>	No extiende.

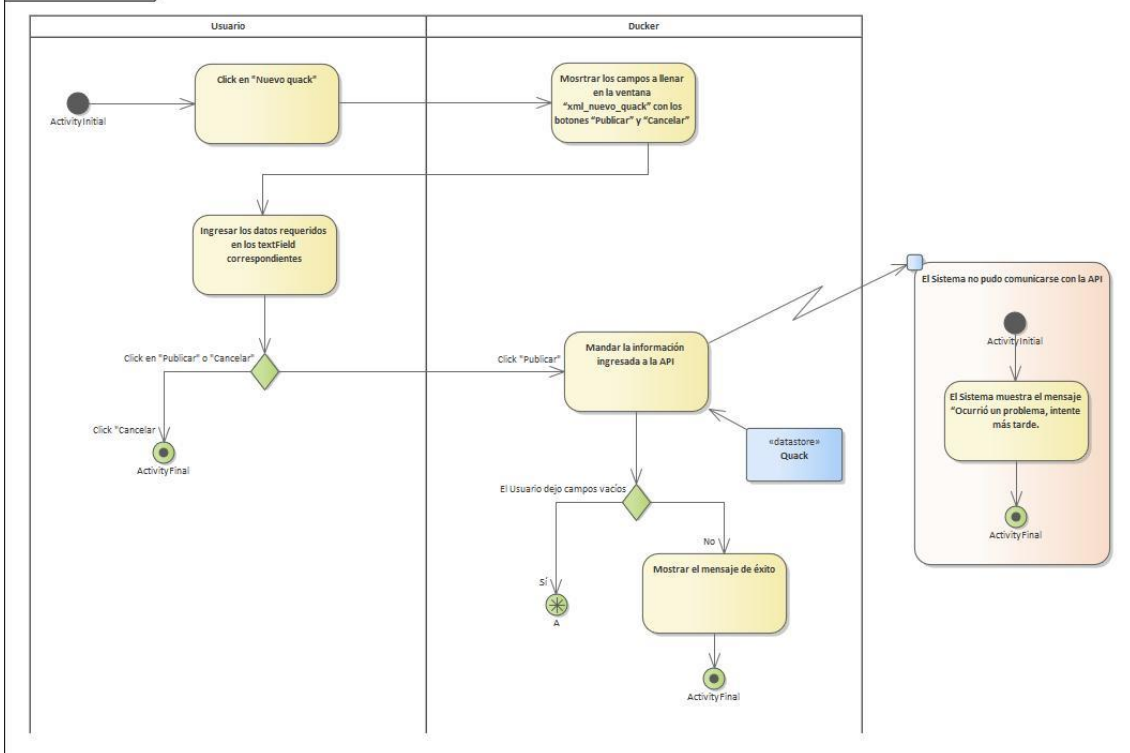


## Diagramas de actividad

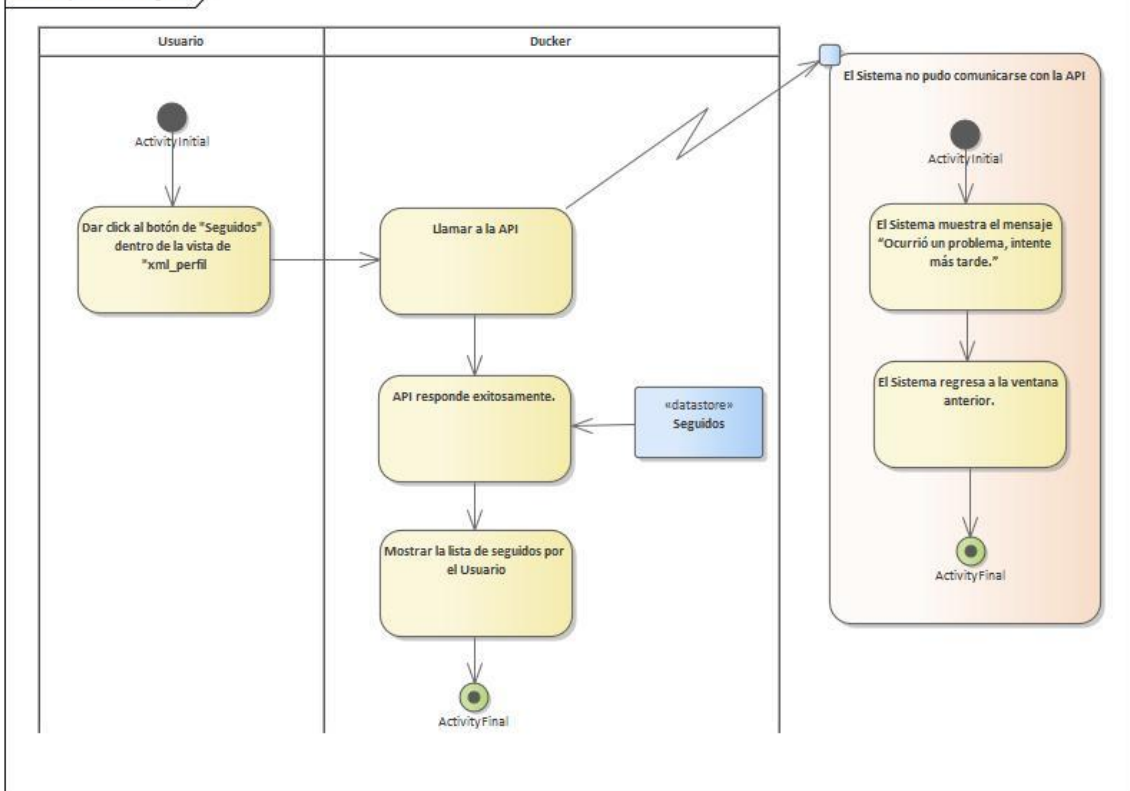




#### act Activity-CU-03-PublicarQuack

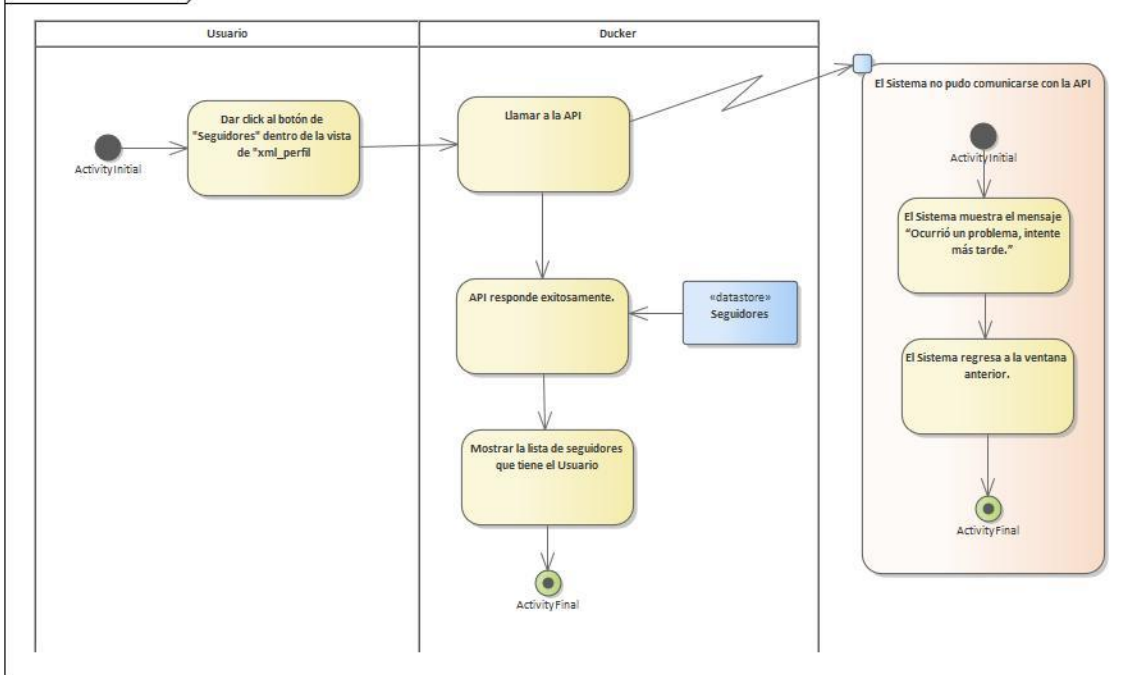


#### act Activity-CU-04-VerSeguidos

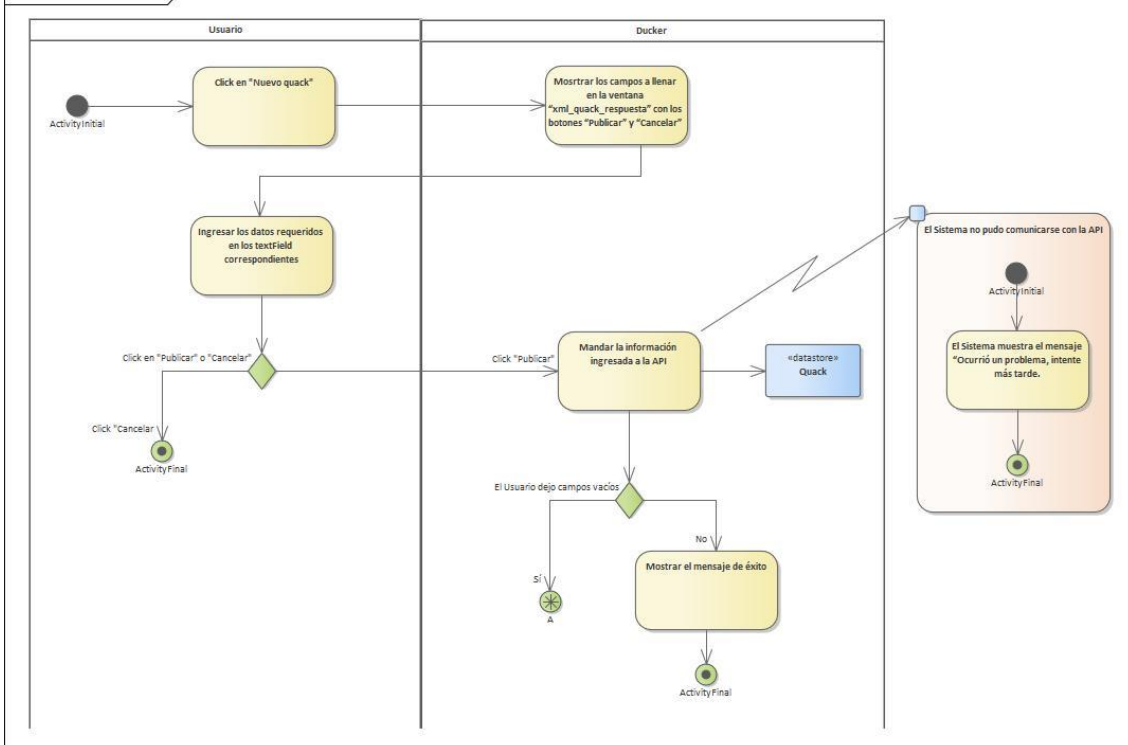




#### act Activity-CU-05-VerSeguidores

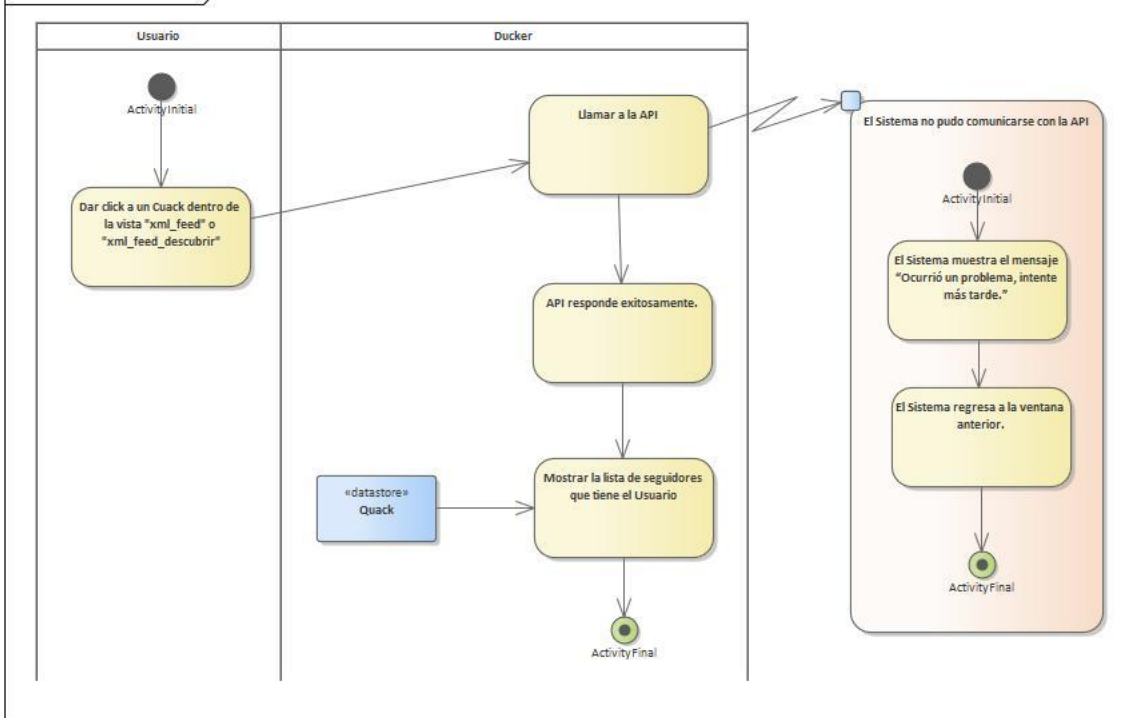


#### act Activity-CU-06-Responder Quack

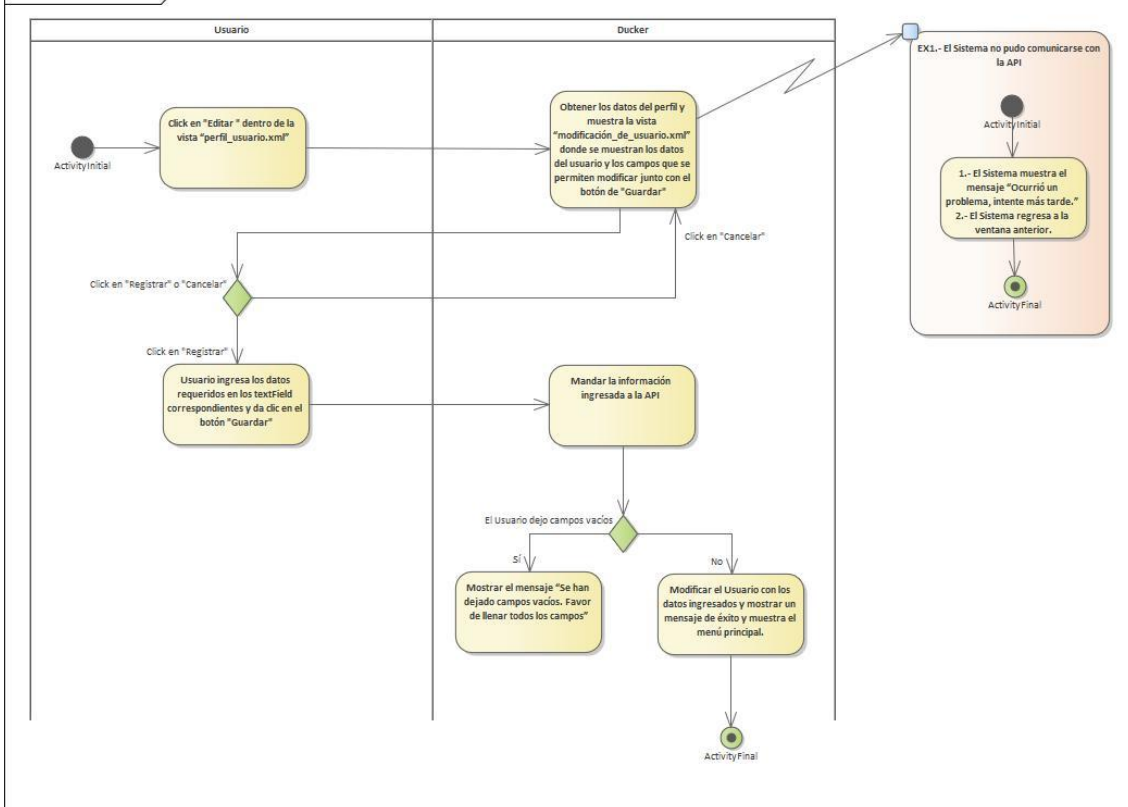




act Activity-CU-07-Visualizar Quack



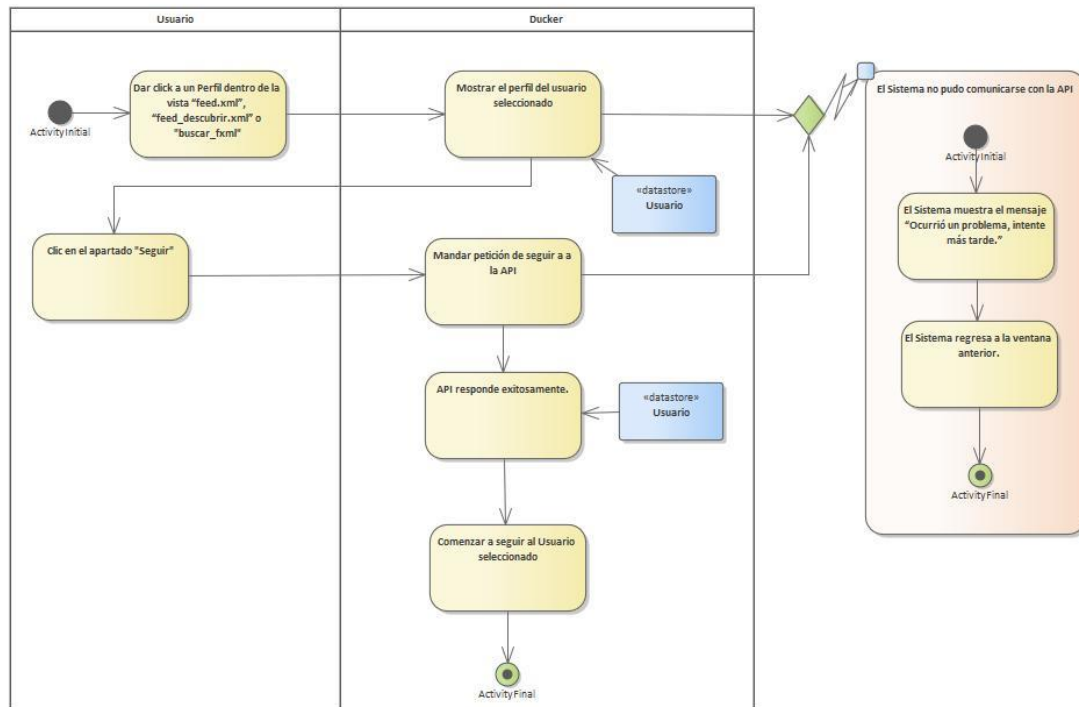
act Activity-CU-08-Modificar perfil



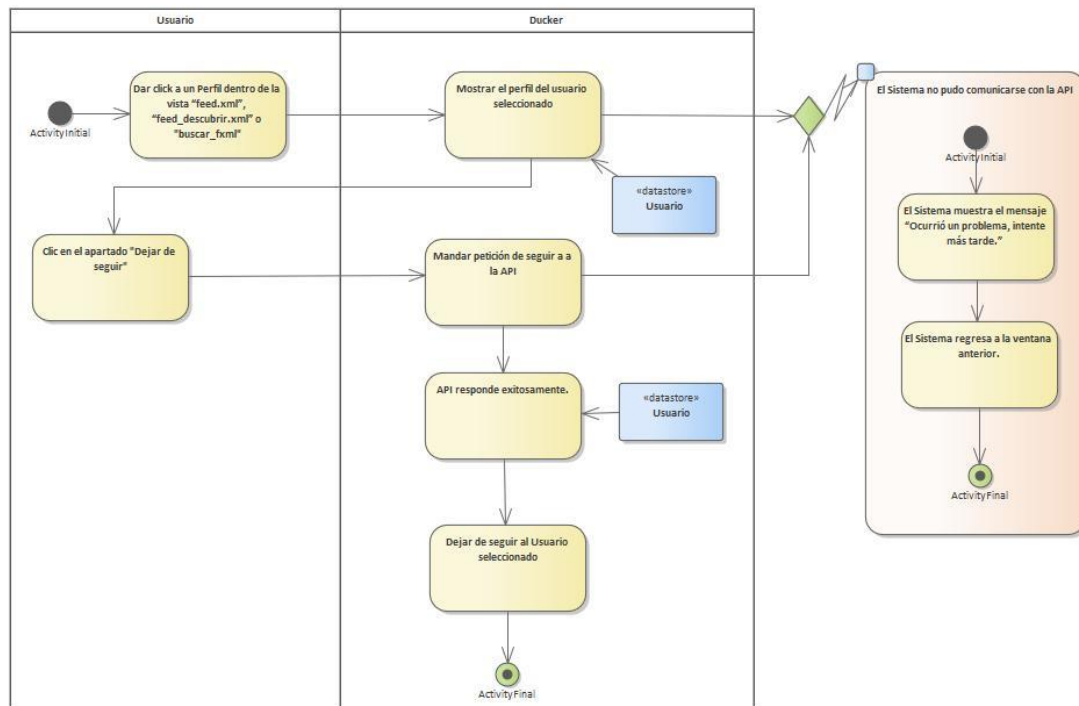




# act ActivityCU-10-Seguir usuario

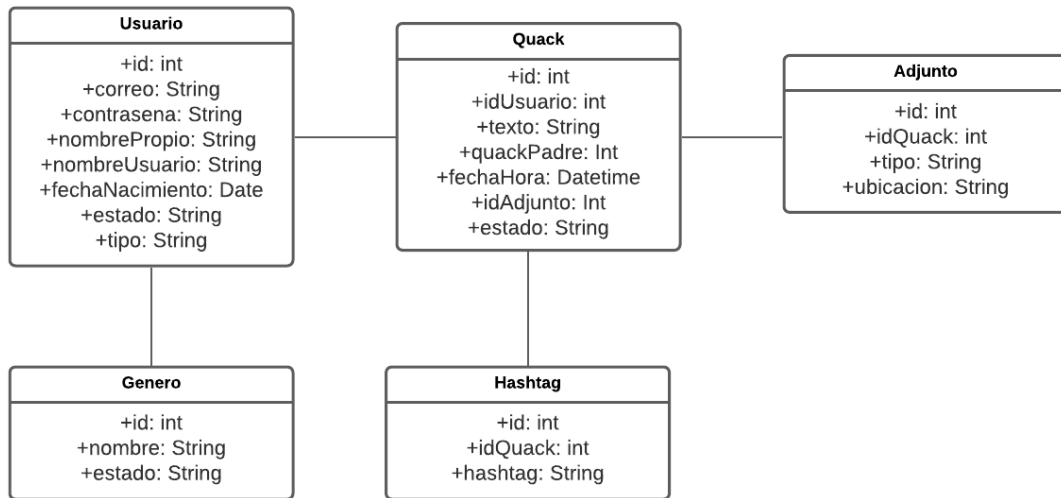


# act ActivityCU-10-Seguir usuario



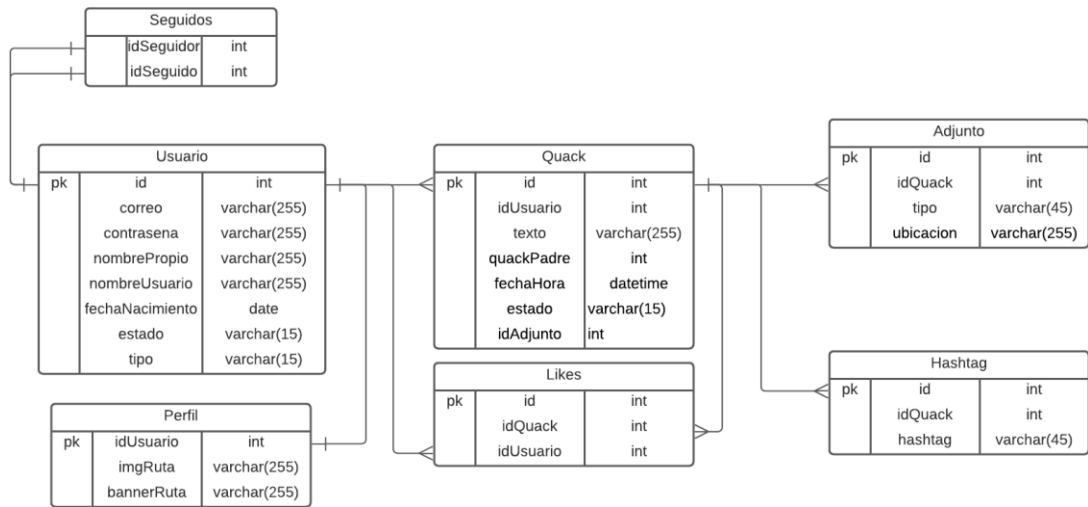


## Diagrama de clases



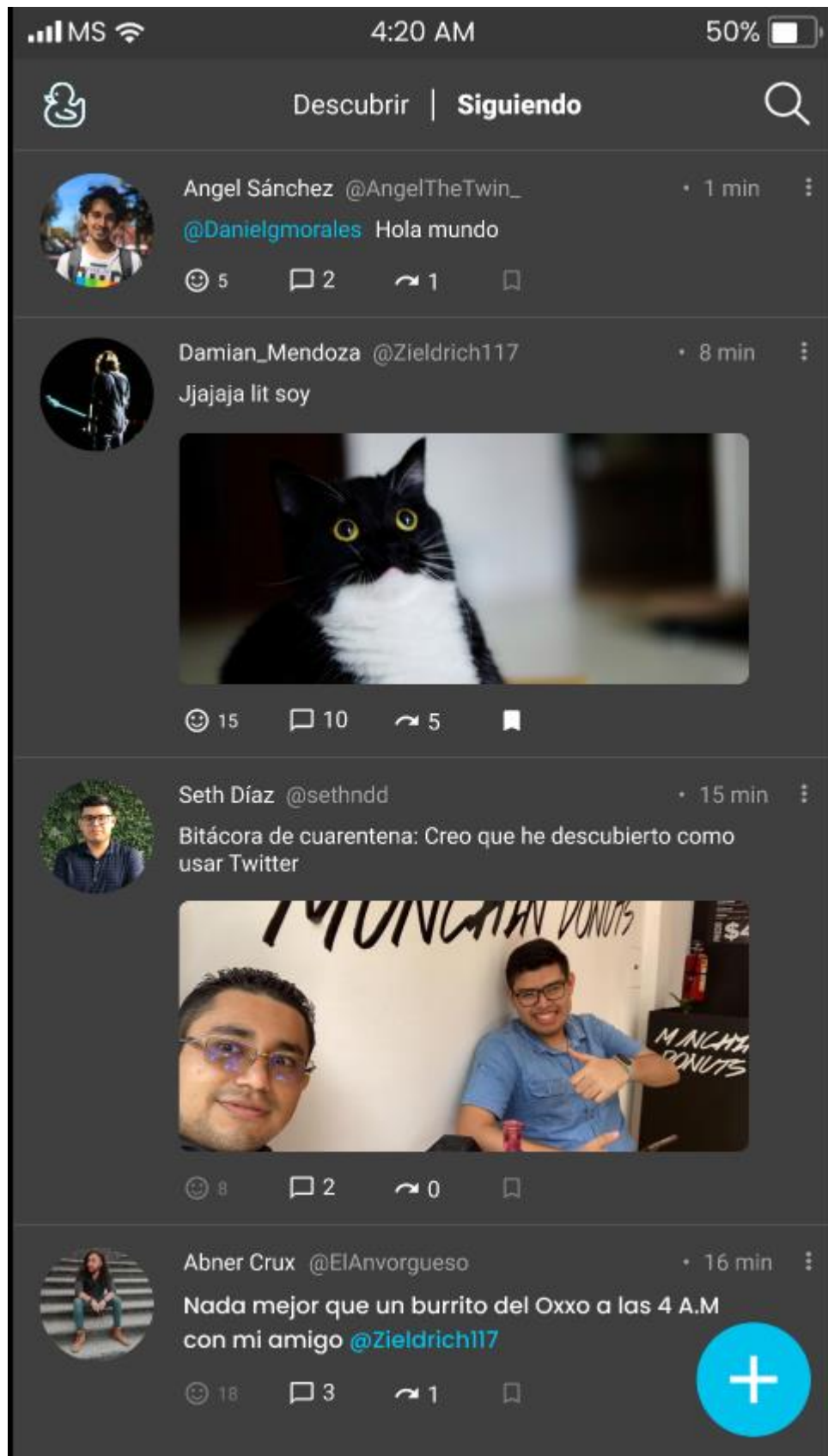


## Diagrama E R





## Prototipo UI Cliente





MS

4:20 AM

50%

Descubrir | Siguiendo

Adam Rodgers @Steamtrooper

• 1 min

"Leeeeeeroyyyy Jenkins"

140K

39K

28K

Bernadette Peters @OfficialBPeters

• 2 min

AA

15K

38K

5K

Ran @wandasolsen

• 5 min

Me analyzing the No Way Home trailer trying to see if Tobey and Andrey were edited out:

8K

204

32

Octavio Ocharán @Ocharan

• 8 min

"A rational person can find peace by cultivating indifference to things outside of their control"

10

6

0



MS

4:20 AM

50%



Yes daddy?

Trakalosa

demomutty



Editar perfil

Daniel Morales

@Danielgmorales

13 Siguiendo 48 Seguidores



Yes daddy?

Seth Díaz @sethndd

• 15/11/21

AMAAAAAAAAAAAAAAAAAAAAA, ME EQUIVOQUÉ DE CARRERA #MeQuieroDarDeBaja



15

2

0



Yes daddy?

Seth Díaz @sethndd

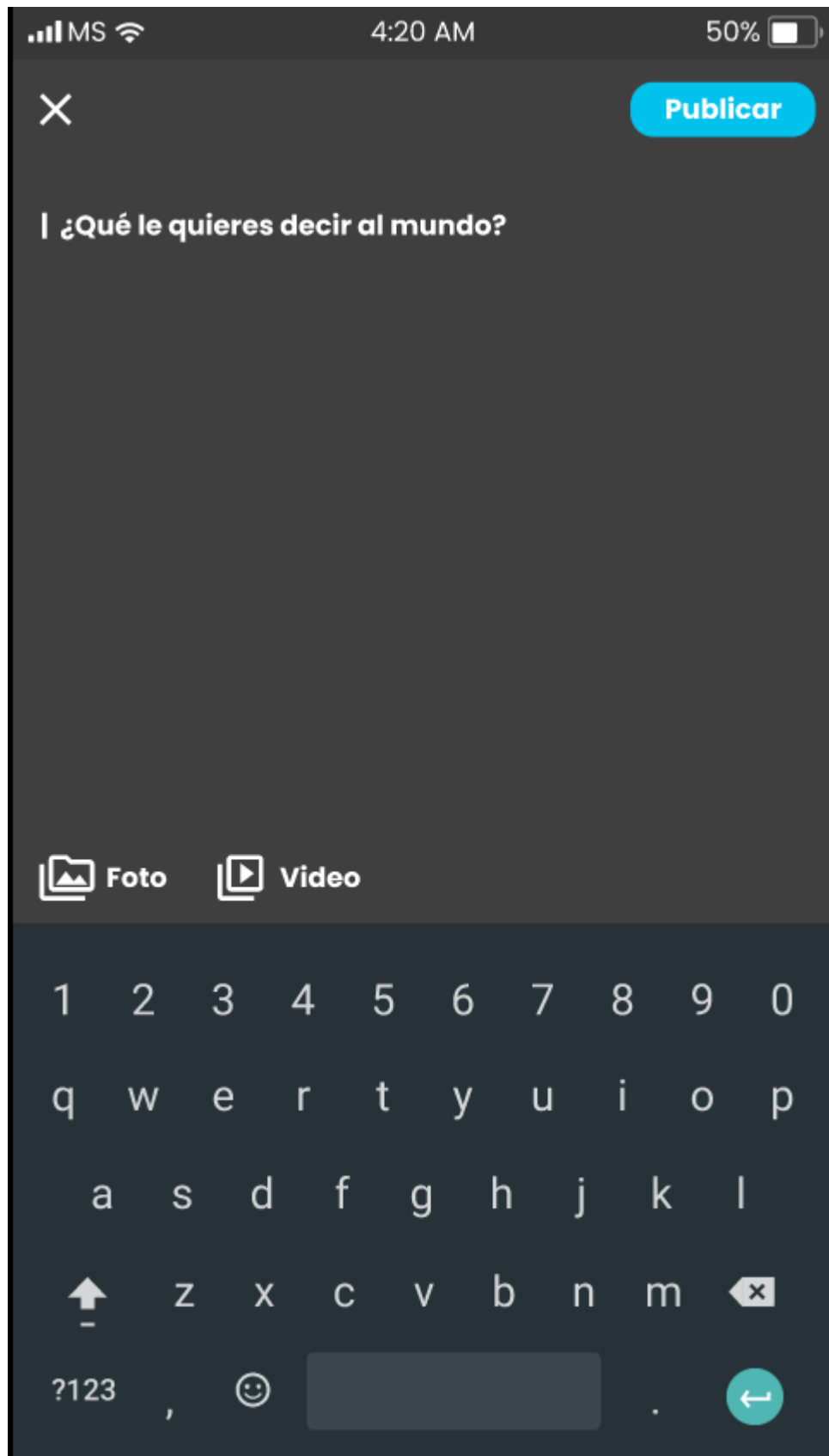
• 15/11/21

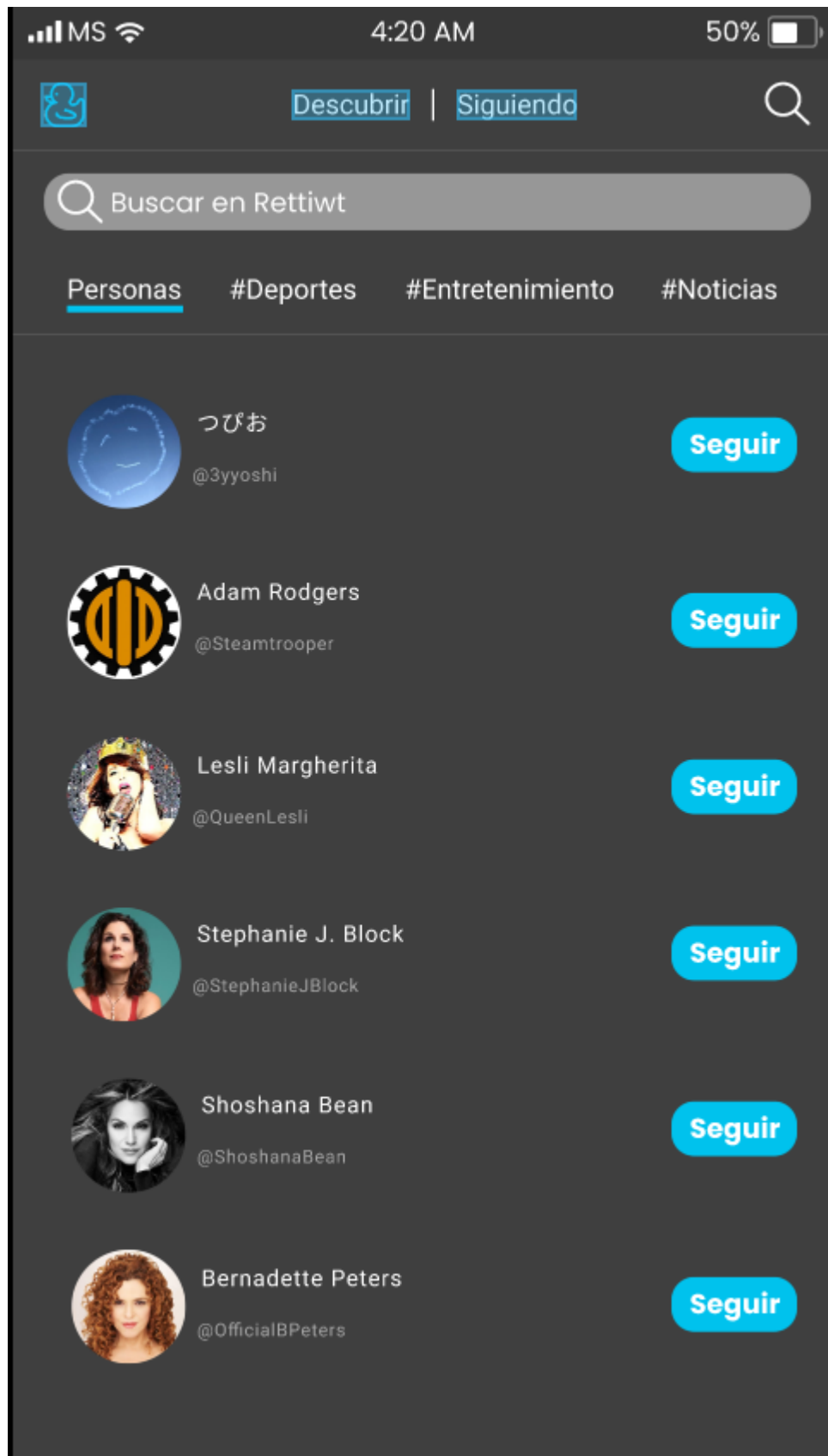
Cuando toca hacer 4 proyectos en menos de 15 días @sethndd @AngelTheTwin\_

Ayno



+









# Construcción

## Selección de lenguajes de programación

En la realización de este proyecto se utilizaron 3 lenguajes de programación, esto debido a que cada uno tiene un propósito y responsabilidad diferente dentro del sistema.

Como base de datos se utilizó MySQL, por lo que el lenguaje usado para este fin fue SQL. En el caso de la API se utilizó NodeJS (Javascript). Finalmente, en el cliente Android, el desarrollo se realizó en Kotlin.

## Estándar de codificación

Para este proyecto utilizamos el estándar de codificación para JavaScript de Engine Core. Para mas información, consulte el enlace en la sección de anexos.

## Selección de tecnologías

Para la elaboración de la API se usó NodeJS(Javascript) puesto que su librería express.js nos permite crear APIs de una forma fácil y entendible. Además, la sintaxis basada en promesas y callbacks, nos permitió mantener un mejor control sobre cuando se reciben los datos.

También se usaron las siguientes librerías:

- BCrypt: Permite cifrar los datos privados. Además, provee un Token cifrado por el cual podemos identificar a los usuarios.
- Express: Permite crear herramientas del lado del servidor que se comunican en tiempo real con un cliente por medio del protocolo HTTP. Fue la base para nuestra API.
- Multer: Middleware para manejar la carga de archivos a nuestro sistema.
- MySQL: Nos permite comunicarnos con MySQL desde NodeJs

Para la implementación de la API decidimos desplegarlo en un servidor propio, donde podemos asegurar su disponibilidad. Para este fin también “contenerizamos” nuestra API por medio de Docker.

En el caso del cliente utilizamos Kotlin y Android Studio, además de las siguientes librerías:

Picasso: Nos permite cargar imágenes alojadas en la web

Retrofit: Para manejar las peticiones HTTP y realizar el cambio de la información



# Anexos

Documentación OPENAPI (Swagger)

<https://app.swaggerhub.com/apis-docs/Sethndd/ducker/1.0.0#/>

Prototipos interactivos: (Figma)

<https://www.figma.com/proto/BaRQ9pVqbttmjMJpwA3RZb/Twitter-Smartphone?node-id=0%3A1&scaling=contain&page-id=0%3A1&starting-point-node-id=7%3A124&show-prot-sidebar=1>

Estándar de codificación

<https://enginecore.com.mx/assets/images/Estandar-Desarrollo-Plataforma-Engine-Core.pdf>