

Curso de Ingeniería de Software

Unidad 5

Requerimientos de software

Guadalupe Ibargüengoitia

Hanna Oktaba

Objetivos

- Especificar los requerimientos de software para la iteración, a partir del *Planteamiento de necesidades*, describiendo los detalles de los casos de uso, proponiendo los prototipos de interfaz y definiendo casos de prueba para diferentes flujos de control.
- El propósito es que el equipo logre el entendimiento común con el Cliente sobre las funcionalidades esperadas del software.
- La especificación de requerimientos servirá de base para el diseño de software.

Entradas a esta unidad

Condiciones

- Todos conocen y entienden los elementos del plan del proyecto.
- Todos conocen y entienden el alcance de la iteración y sus compromisos.
- Todos saben como ejecutar el plan de la iteración, dar el seguimiento usando el tablero y comprenden cómo resolver las dificultades.

Productos de trabajo

- Herramienta(s) colaborativa(s) que incluye una carpeta para la iteración
- Plan del proyecto documentado y resguardado
- Tablero de la iteración creado para el alcance de la iteración
- *Planteamiento de Necesidades*
- Plantilla de *Especificación de requerimientos de software*

DEFINICIONES DE CONCEPTOS

¿Qué es un requerimiento de software?

- Los **requerimientos de software** expresan las **funcionalidades y restricciones** que debe tener un producto de software para que contribuya a la solución de un problema en el mundo real (SWEBOK, 2014).
- Los **requerimientos** de un producto de software son las descripciones de **lo que hará, los servicios** que proporcionará y las **restricciones** de su operación (Sommerville I., 2011).
- Un **requerimiento o necesidad** es lo que el **cliente o un usuario desean** que **haga el software** para resolver un **problema**.

¿Qué se hace para identificar los requerimientos de software?

- A fin de entender el problema para poder especificar los requerimientos se pueden:
 - Hacer **entrevistas** con el Cliente y los posibles usuarios
 - Aplicar **cuestionarios** a los usuarios
 - **Observar** a los futuros usuarios al realizar las tareas que apoyará el software
 - **Revisar documentos o sistemas** ya existentes que se pretenden mejorar
 - Entender **los términos que usa el Cliente y los usuarios (dominio del problema).**

¿Cómo se especifican los requerimientos?

- Lenguaje natural – texto
- Forma gráfica – diagramas
- Prototipo de interfaz de usuario – pantallas

¿Para qué se especifican los requerimientos?

- Crear un **modelo de las funcionalidades de software** comprensible para **el cliente** y para **los desarrolladores**.
- **Obtener criterios para verificar y validar** el producto de software terminado.
- Guiar el **proceso de desarrollo**.

¿Qué tiene que ver la especificación de requerimientos con la calidad de software?

- La calidad de un producto de software se define como “**capacidad** de un producto de software de **satisfacer las necesidades** expresas e implícitas en **condiciones específicas**” (SWEBOOK, 2014).
- Al definir los requerimientos se busca **claridad, precisión y no ambigüedad** tanto para el desarrollador como para el cliente.

¿Cómo asegurar la calidad de la especificación de los requerimientos?

- El cliente **evalúa si** los desarrolladores **comprendieron correctamente** las funcionalidades esperadas del futuro software **revisando y avalando la especificación** de los requerimientos.
- La **validación final**, de que el **software cumple con los requerimientos especificados**, se hace hasta que el **software esté construido**.
- El **problema** que hay que afrontar es que los requerimientos pueden **cambiar**.

Tipos de requerimientos

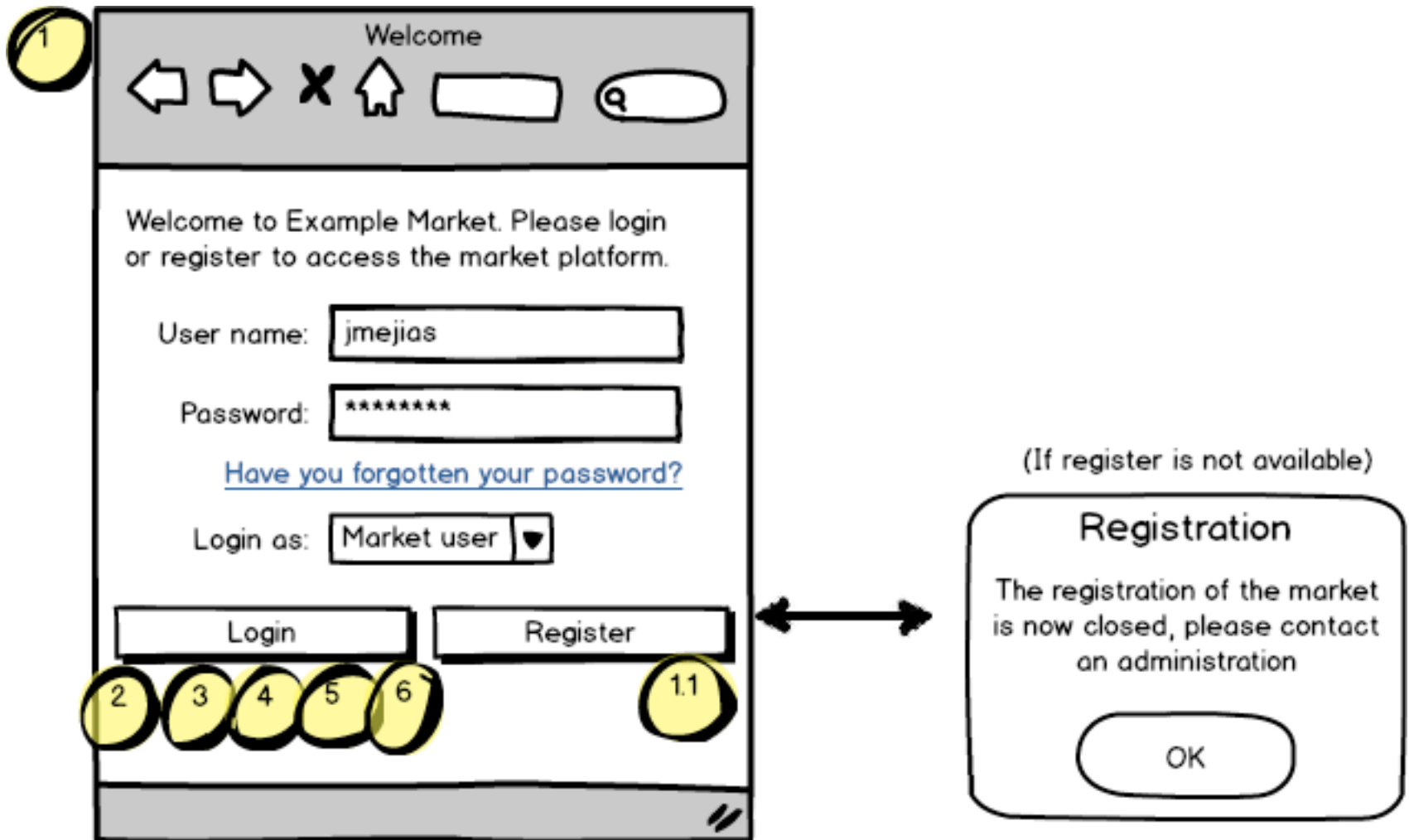
- Funcionales
 - Son los servicios, la forma en que el software debe reaccionar a entradas particulares y cómo debe comportarse en situaciones particulares
- No-Funcionales
 - Son restricciones que los servicios y funciones de software deben cumplir (ver Unidad 4)

Actividades para definir los requerimientos

- **Obtener** – Entender (desarrollador/cliente)
- **Analizar** (desarrollador)
- **Especificar** – Documentar texto, diagramas, prototipos (desarrollador)
- **Validar** (por el cliente)
- **Administrar** – Cambios a requerimientos (cliente/desarrollador)

Prototipo de interfaz de usuario

- Se diseñan **prototipos de pantallas** para facilitar al **Cliente la visualización** de la **interacción** con el futuro sistema de software.
- Para diseñar las pantallas se puede usar desde **papel y lápiz**, código **HTML**, o una **herramienta** que lo facilite.



Casos de prueba

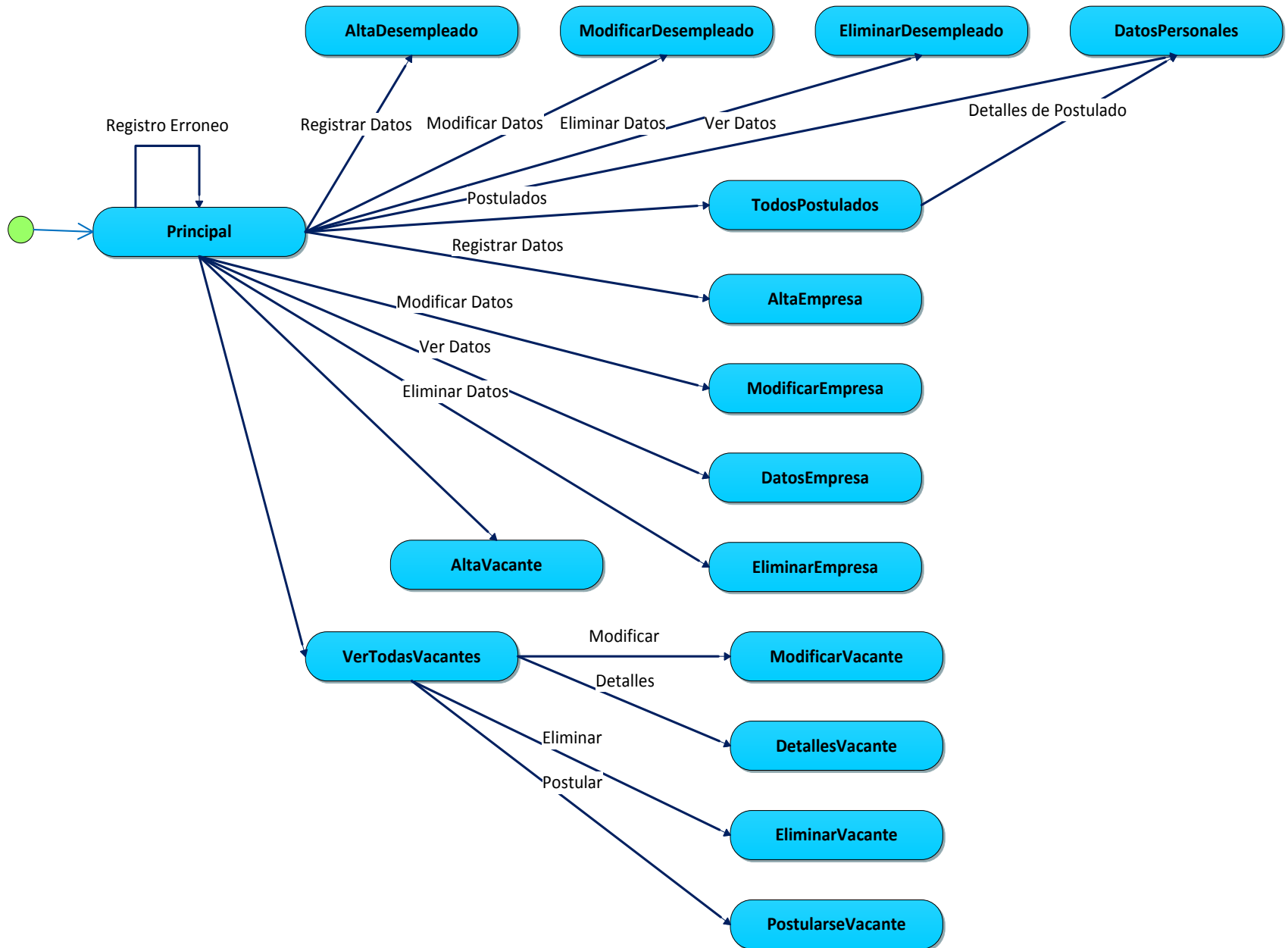
- Para todos los **casos de uso** se definen **casos de prueba**. Es decir, se definen **ejemplos de datos de entrada** y los **resultados esperados**.
- Los **casos de prueba** se usarán cuando el **caso de uso ya esté implementado** para confirmar el cumplimiento de los **requerimientos especificados**.
- Los **casos de prueba** sirven para **verificar** los requerimientos.

Diagramas de estados para la navegación entre las pantallas (1/2)

- Un diagrama de estados modela una máquina de estados finitos o autómata, que enfatiza el flujo de control de un estado a otro.
- Es una gráfica con nodos que representan estados y por arcos dirigidos que representan transiciones entre los estados a causa de eventos.

Diagrama de estados para la navegación entre las pantallas (1/2)

- El **modelo de navegación** define el orden y las relaciones entre las funcionalidades (**casos de uso**), se documenta usando **diagramas de estados** de UML.
- Las **pantallas del sistema** se definen como estados, se **identifica el evento** que causa el cambio de pantalla (estado).
- Por ejemplo, si estamos en la **pantalla de inicio** de un sistema (**estado**) y **seleccionamos *entrar al sistema*** (**evento**) nos llevará a una pantalla que nos permite capturar datos de usuario y contraseña (**otro estado**).



¿Qué hemos aprendido?

- ¿Porqué se especifican los requerimientos?
- ¿Qué es un prototipo de la interfaz de usuario?
- ¿Qué es un caso de prueba para un caso de uso?
- ¿Paraqué sirven los diagramas de navegación?

PRACTICA PD1 REQUERIMIENTOS DE SOFTWARE

PD1 Requerimientos de software

Objetivos

- Comprender y especificar los requerimientos de software, basándose en el *Planteamiento de necesidades*.

PD1 Requerimientos de software

Entradas

Condiciones

- Equipos conformados

Productos de trabajo

- *Planteamiento de necesidades*
- Plantilla de *Especificación de requerimientos de software*
- Herramienta(s) colaborativa(s) inicializada (s)

PD1 Requerimientos de software

Actividades

1. Analizar y especificar los requerimientos funcionales en el alcance de la iteración.
2. Crear prototipos de la interfaz de usuario.
3. Definir casos de prueba.
4. Hacer el diagrama de estados para la navegación entre pantallas.
5. Documentar la *Especificación de requerimientos de software*.

TÉCNICAS PARA REALIZAR LAS ACTIVIDADES DE PD1

Actividad 1 Analizar y especificar los requerimientos funcionales en el alcance de la iteración.

(1/3)

- Para tener una **vista gráfica** de los **casos de uso** que deben implementarse en la iteración el **Responsable técnico** hace el **Diagrama de Casos de Uso** en el alcance de la iteración, definido durante la planificación de la iteración.
- El **Responsable de calidad** asegura que se integre el diagrama en la **plantilla personalizada de Especificación de Requerimientos de Software**.

Actividad 1 Analizar y especificar los requerimientos funcionales en el alcance de la iteración.

(2/3)

Para entender mejor cada caso de uso el equipo analiza las secciones correspondientes del *Planteamiento de necesidades* y trata de imaginar cómo el actor va a interactuar con el software para ejecutar un caso de uso.

El resultado de este análisis se describe con la estructura que se presenta en las siguientes láminas.

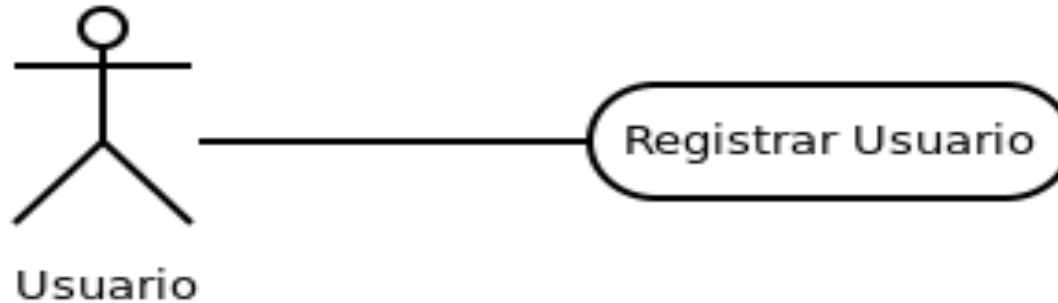
Estructura de la descripción detallada de un caso de uso (1/2)

- **Caso de uso** :“Nombre”
- **Diagrama del caso de uso**: dibujo del o los actores y el caso de uso en cuestión.
- **Descripción**: Texto breve describiendo la función que representa el caso de uso.
- **Precondiciones**: El estado en el que se encuentra el sistema para que el actor (o los actores) pueda iniciar el caso de uso.

Estructura de la descripción detallada de un caso de uso (2/2)

- **Flujo normal de eventos:** Tabla que describe el flujo de interacciones exitosas (*el camino feliz*) entre el actor y el sistema durante el caso de uso.
- **Flujo alternativo de eventos:** Tabla que describe el flujo de interacciones alternativas del flujo normal en el caso de uso que también permiten terminar con éxito.
- **Flujo excepcional de eventos:** Tabla con las acciones que ocurren en situaciones anormales o excepcionales.
- **Poscondiciones:** Define el estado en el que se encuentra el sistema después de la terminación exitosa del caso de uso.

Caso de uso: Registrar Usuario



Descripción: Permite al Usuario registrar sus datos en el sistema para poder hacer uso de éste.

Precondiciones: El Usuario no registrado está viendo la página principal de la red social con un botón de registro.

Flujo normal de eventos

Paso	Acción Actor	Paso	Acción Software	ID
1	Usuario presiona el botón de registrar.	2	Despliega campos para capturar correo electrónico y dos veces la contraseña.	
3	Usuario llena los campos del formulario con sus datos. Presiona el botón de enviar.	4	Valida si el correo tiene @ y si las contraseñas son idénticas. Si es así, presenta formulario adicional para agregar mas datos: sexo, fecha de nacimiento, foto (opcional) y pasatiempos.	A1 A2 E1
5	El Usuario llena los datos adicionales. Presiona el botón de enviar.	6	El sistema registra los datos del Usuario y despliega la página del inicio de Usuario registrado.	E1

Flujos alternativos

ID	Acción Actor	Acción Software
A1	El Usuario ingresa un correo sin @ y/o contraseñas no idénticas.	Informa al Usuario que los datos son incorrectos y le pide que vuelva a introducirlos.
A2	El Usuario omite llenar algún campo.	Informa al Usuario que todos los campos son obligatorios y le pide que los llene para poder terminar el proceso.

Flujo excepcional de eventos

ID	Nombre	Acción Sistema
E1	Ocurre un error de conexión con el servidor al momento de intentar guardar el registro.	Aborta el proceso, informa que hubo un problema y sugiere que intente mas tarde.

Poscondiciones: El Usuario queda registrado en el sistema y queda desplegada la página de inicio del Usuario. En caso del flujo excepcional el Usuario queda en la página principal sin cumplir con el registro.

Actividad 1 Analizar y especificar los requerimientos funcionales en el alcance de la iteración.

(3/3)

- El equipo designa, de común acuerdo, a los responsables de describir el detalle de todos los casos de uso dentro del alcance de la iteración.
- La asignación debe de reflejarse en el tablero Kanban con la fecha de entrega acorde con el guión del curso.
- El Responsable de calidad asegura que se incorporen los detalles de todos los casos de uso en la plantilla personalizada de *Especificación de requerimientos de software*.

Actividad 2 Crear prototipos de la interfaz de usuario.

- El **Responsable técnico** convoca al equipo a una reunión de toma de decisiones con el objetivo de definir lineamientos para los **prototipos de interfaz de usuario para los casos de uso**.
- Cada responsable del caso de uso diseña los prototipos de pantallas para el flujo normal, alternativo y excepcional, siguiendo los lineamientos acordados.
- Estas actividades quedan reflejadas en el **tablero Kanban**.
- El **Responsable de calidad** asegura que los prototipos de todos los casos de uso se incorporen en la plantilla personalizada de *Especificación de requerimientos de software*.

Actividad 3 Definir casos de prueba.

- Cada responsable de un caso de uso define los casos de prueba que corresponden a los flujos normales, alternativos y excepcionales. Estas actividades quedan reflejadas en el tablero Kanban.
- Los casos de prueba para cada caso de uso se documentan en una tabla (ver ejemplos en las siguientes láminas).
- Un caso de prueba identifica las entradas al caso de uso y los resultados esperados.
- Se consideran entradas para flujos normales, alternativos y excepcionales.
- El Responsable de calidad asegura que se integren los casos de prueba en la plantilla personalizada de *Especificación de requerimientos de software*.

Caso de prueba 1 (flujo normal)

Entradas	Resultado esperado
luisoto976@gmail.com fpEo34Sw fpEo34Sw masculino 25/09/1990 perfjl.jpg música futbol cocinar	Se espera que se registre exitosamente el Usuario nuevo.

Caso de prueba 2 (flujo alternativo)

Entradas	Resultado esperado
luisoto976@gmail.com fpEo34Sw q1u5r4kw3e3 masculino 25/09/1990 perfjl.jpg música futbol cocinar	Se despliega un aviso al Usuario de que la confirmación de contraseña no es correcta.

Caso de prueba 3 (flujo excepcional)

Entradas	Resultado esperado
Antes de enviar los datos completos se pierde la conexión a la base de datos.	Se despliega un aviso al usuario que hubo una falla y que intente más tarde.

Actividad 4 Hacer el diagrama de estados para la navegación entre pantallas.

- Para construir el diagrama de estados, que modela la navegación, se parte de los prototipos de la interfaz de usuario contruidos para los casos de uso.
- Las pantallas se representan como estados y los eventos, que ocasionan el cambio de una pantalla a otra, por las transiciones entre estados.
- El estado inicial del diagrama, apunta al estado que representa a la pantalla inicial. En esta pantalla, por lo general, se tienen varias opciones para navegar.
- Cada opción representa a un evento que puede llevar a otra pantalla.

Actividad 5 Documentar la Especificación de requerimientos de software.

- El **Responsable de calidad** revisa la *Especificación de requerimientos de software* asegurar que fueron integrados y están completos todos los elementos generados en esta práctica.
- El **Resposable de colaboración** asegura que el documento esté resguardado en la carpeta correspondiente del repositorio compartido.

PD1 Requerimientos de software

Resultados

Condiciones

- Todos conocen y entienden los requerimientos para la iteración.

Productos de trabajo

- *Especificación de requerimientos de software* documentada y resguardada

Resultados de esta unidad

- **Condiciones**
 - Requerimientos de software entendidos
- **Productos de trabajo**
 - *Especificación de requerimientos de software* que incluye:
 - Diagrama general de casos de uso de la iteración
 - Detalle de los casos de uso
 - Prototipos de interfaz de usuario
 - Casos de prueba para los casos de uso
 - Diagrama de estados para la navegación

Bibliografía

- Jacobson I., C. M. (1992). *Object-Oriented Software Engineering. A Use Case Driven Approach*. Addison-Wesley.
- Sommerville I. (2011). *Software Engineering*. Pearson.
- SWEBOK. (2014). *Guide to the Software Engineering Body of Knowledge v3.0*. IEEE Computer Society.