

Validación y Pruebas

(Práctica 1 – EncuestasUBU)



UNIVERSIDAD DE BURGOS

Autores:

Mario Ubierna San Mamés

Jorge Navarro González



Índice de Contenido

| | |
|---|-----|
| Descripción de la aplicación | 5 |
| Definición de los requisitos | 6 |
| Diseño de cada clase..... | 7 |
| Clase Usuario | 7 |
| Atributos | 7 |
| Métodos..... | 7 |
| Pruebas unitarias..... | 7 |
| Clase Encuesta | 8 |
| Atributos | 8 |
| Métodos..... | 8 |
| Pruebas unitarias..... | 8 |
| Clase Valoración..... | 9 |
| Atributos | 9 |
| Métodos..... | 9 |
| Pruebas unitarias | 9 |
| Clase DataBase..... | 10 |
| Atributos | 10 |
| Métodos..... | 10 |
| Pruebas de Integración..... | 11 |
| Ventanas de nuestra aplicación | 13 |
| Prototipo inicial de la aplicación (en papel) | 13 |
| Escuela Politécnica Superior | -2- |



| | |
|--|----|
| Usuario..... | 13 |
| Administrador | 14 |
| Prototipo final de la aplicación (en papel)..... | 15 |
| Usuario..... | 15 |
| Administrador | 15 |
| Diagrama Caso de Uso de la Aplicación | 16 |
| Historias de usuario (breve descripción)..... | 17 |
| Historia de Usuario (Usuario) | 17 |
| Historia de Usuario (Administrador)..... | 18 |
| Historias de Usuario | 19 |
| Historias de Usuario 1..... | 19 |
| Descripción corta..... | 19 |
| Descripción larga | 19 |
| Pruebas de aceptación | 19 |
| Diagrama de secuencia..... | 20 |
| Historia de Usuario 2 | 21 |
| Descripción corta..... | 21 |
| Descripción larga | 21 |
| Pruebas de aceptación | 21 |
| Diagrama de secuencia..... | 22 |
| Historia de Usuario 3 | 23 |
| Descripción corta..... | 23 |
| Descripción larga | 23 |



| | |
|-----------------------------|----|
| Pruebas de aceptación | 23 |
| Diagrama de secuencia..... | 24 |
| Historia de Usuario 4 | 25 |
| Descripción corta..... | 25 |
| Descripción larga | 25 |
| Pruebas de aceptación | 25 |
| Diagrama de secuencia..... | 26 |
| Historia de Usuario 5 | 27 |
| Descripción corta..... | 27 |
| Descripción larga | 27 |
| Pruebas de aceptación | 27 |
| Diagrama de secuencia..... | 28 |
| Historia de Usuario 6 | 29 |
| Descripción corta..... | 29 |
| Descripción larga | 29 |
| Pruebas de aceptación | 29 |
| Diagrama de secuencia..... | 30 |



Descripción de la aplicación

Se requiere implementar un sistema de recogida de la opinión de nuestros usuarios respecto a distintos servicios y al estado de nuestras instalaciones.

La aplicación se divide en dos partes: la de administración que previa autenticación nos permite acceder al apartado de administración de encuestas y el interfaz que utilizará el usuario.

La interfaz de administración permitirá como mínimo:

- Crear la encuesta con una sola pregunta cuya respuesta se obtiene de seleccionar una sola de las cuatro opciones visualizadas.
- Activar y desactivar encuestas.

La interfaz del usuario permitirá como mínimo:

- Selecciona de entre las encuestas activas aquella que quiera responder.
- Complimentar la pregunta e introducir de forma voluntaria los comentarios que desee.

Tal y como podemos observar en el texto anterior, tenemos cuatro requisitos (los cuatro puntos que hay). Por otro lado, nosotros hemos definido otros tres requisitos más:

- Modificar una encuesta determinada, es decir, poder seleccionar una encuesta y poder cambiar la descripción de dicha encuesta. (Esta opción solo la tiene disponible el administrador).
- Sugerir encuesta, es decir, que el usuario pueda además de votar, sugerir una encuesta interesante.
- Eliminar una encuesta, es decir, seleccionando una encuesta determinada que se pueda borrar. (Esta opción solo la tiene disponible el administrador).



Definición de los requisitos

| Requisitos | Descripción |
|------------|---|
| R1 | Se desea realizar la votación de una determinada encuesta activa. Además de seleccionar la encuesta y la votación que queremos realizar, se podrá introducir un comentario de forma opcional. |
| R2 | Se desea crear una encuesta, dicho requisito solo se cumple si eres un administrador. Para ello, tenemos que introducir el nombre de la encuesta y su descripción. |
| R3 | Se desea activar o desactivar una encuesta, dicho requisito solo se cumple si eres un administrador. Para ello, tendrás que seleccionar la encuesta que quieres activar o desactivar. |
| R4 | Se desea eliminar una encuesta, dicho requisito solo se cumple si eres un administrador. Para ello, se elige la encuesta que quieres eliminar. |
| R5 | Se desea modificar una encuesta, dicho requisito solo se cumple si eres un administrador. Para ello, seleccionará la encuesta que desea y podrá cambiar la descripción de dicha encuesta |
| R6 | Se desea sugerir una encuesta. Para ello, introducirá en la ventana correspondiente el nombre y la descripción de la encuesta. |



Diseño de cada clase

En cada clase hemos definido unos métodos equals y hashCode.

Clase Usuario

Atributos

Los atributos de la clase usuario que hemos puesto son:

- IdUsuario: es un número que se incrementa cada vez que se crea un usuario, un número único para cada usuario.
- Cuenta: es el correo de la persona que va a entrar en nuestra aplicación.
- Contraseña: la contraseña con la que va a entrar nuestro usuario a la aplicación, dicha contraseña está encriptada.

Métodos

En cuanto a los métodos, vamos a tener distintos, por un lado, vamos a tener unos métodos get y set para cada atributo que tengamos, además de dichos métodos también vamos a tener los siguientes:

- Encriptar(): sirve para encriptar la contraseña que pasamos por cabecera.
- ComprobarPass(): comprueba si la contraseña que pasamos por cabecera desencriptada, es la misma que la que tiene encriptada el usuario.

Pruebas unitarias

En cuanto a las pruebas unitarias que hemos realizado para la clase usuario, son las siguientes:

- Comprobar un get de los atributos de forma correcta, para ello hemos comprobado con assert que los gets están hechos de forma correcta, igualando lo que nos devuelve el get del atributo a lo que debería de salir y viendo si el assert es true.
- Comprobar un get de los atributos incorrecto, comprobamos en este caso un "assert.arenotequal" para ver si lo que nos devuelve el get no se corresponde con lo que ponemos nosotros a mano.
- Comprobar que el set se hace de manera correcta, para ello asignamos un valor con el set de cada uno de los atributos y comprobamos en el propio assert que se haya cambiado correctamente, si nos devuelve true se habrá hecho bien.



Clase Encuesta

Atributos

Los atributos de la clase encuesta que hemos puesto son:

- Id: que es un número único que identifica a cada encuesta.
- Nombre: en nuestro caso también el nombre va a ser único junto con el id, por tanto, no habría ninguno repetido.
- Descripción: que sería un párrafo para definir el tema de la encuesta.
- Activa: es un booleano que nos sirve para poder saber si dicha encuesta está activa o no está activa.
- Opiniones: es una lista con todas las opiniones que se han seleccionado para dicha encuesta.

Métodos

Los métodos de la clase Encuesta van a ser los get y set de todos los atributos que tenemos, en el caso de opiniones son también un get y set pero definidos por nosotros, ya que para el set, debíamos introducir la opinión en el caso de que el valor fuera distinto de 0 que es el valor por defecto que definimos, y también el método get es un método que devuelve toda la lista de opiniones.

Pruebas unitarias

En cuanto a las pruebas unitarias que hemos realizado para la clase encuesta, son las siguientes:

- Comprobar que los métodos get sean correctos, que es igual que en la clase usuario, pero con los de encuesta.
- Comprobar con un get que no sea correcto, al igual que antes igual que en usuario, pero con los de encuesta.
- Comprobar que los métodos set funcionen correctamente, para ello también hacemos lo mismo que con los de usuario, pero haciéndolo con encuesta.
- Que se haga bien el valor por defecto ya que hemos puesto en nuestro programa que cuando metes una encuesta y no especificas el atributo de activa, la encuesta empiece de inicio con el valor de false, con lo cual probamos a crear una encuesta y ver si su valor es false.
- Comprobamos que funciona correctamente el getopiniones y el setopiniones, para lo cual primero comprobamos que la lista tenga tamaño 0, luego introducimos algunas opiniones y vemos a ver si el tamaño de la lista va cambiando.



Clase Valoración

Atributos

Los atributos de la clase valoración que hemos puesto son:

- Valor: que es el valor que se le va a dar a una encuesta.
- Opinión: es la opinión opcional que se puede poner para una encuesta.

Métodos

Los métodos de la clase valoración son los métodos get y set de todos los atributos que tenemos en la clase valoración, también tenemos un método que se llama comprobarRangoValor(), al que le pasas un valor y si el valor está entre 1 y 4 devuelve ese valor, pero si el valor es distinto de ese rango devuelve un 0 por defecto.

Pruebas unitarias

En cuanto a las pruebas unitarias que hemos realizado para la clase valoración, son las siguientes:

- Test de valor correcto, el cual comprueba a crear un objeto valoración y comprueba que el valor se corresponde y la opinión también.
- Test de valor mayor que 4, el cual va a comprobar que si metes un valor mayor que 4 pone por defecto el valor de 0 en el valor.
- Test menor que 4, hace lo mismo que el anterior pero con un valor menor de 0.
- Dos tests más para el set de valor y el de opinión, el cual intentamos cambiar el valor y la opinión y comprobamos que se haya cambiado.



Clase DataBase

Atributos

Los atributos de la clase database que hemos puesto son:

- nextUser: que es el identificador para cada usuario que haya en nuestra base de datos.
- nextPoll: que es el identificador para cada encuesta que haya en nuestra base de datos.
- nextAnswer: que es un identificador para las respuestas que haya a las encuestas.
- Usuarios: que es una lista de usuarios con todos los usuarios que hay en nuestra base de datos.
- Encuestas: lo mismo, una lista con todas las encuestas que hay en la base de datos.
- Valoraciones: es una lista también con las valoraciones que hay.
- Sugerencias: toda la lista en la que se van añadiendo las encuestas que los propios usuarios que puntúan, van sugiriendo.

Métodos

Esta clase va a ser la que más métodos vamos a tener, ya que son los métodos que van a hacer que toda nuestra aplicación funcione, vamos a tener los siguientes:

- setNextPoll(): va a ser un método para incrementar el id de la encuesta que vamos a tener.
- getActivas(): que es un método que nos sirve para devolver una lista con todas las encuestas que tengamos activas.
- getNoActivas(): nos devuelve una lista con las encuestas que no están activas en ese momento.
- getUsuarios(): es un método que nos devuelve la lista con todos los usuarios que tenemos.
- getEncuestas(): es un método que nos devuelve una lista con todas las encuestas que tenemos.
- getValoraciones(): es un método que nos devuelve una lista con las valoraciones que tiene una determinada encuesta que le pasamos como parámetro al método.
- getUsuario(): es un método que nos devuelve un usuario entero, pasándole como parámetro solo la cuenta del usuario, en caso de que no encuentre ninguno, nos va a devolver un "null".



- `sugerirEncuesta()`: es un método al que le pasamos como parámetros el nombre y la descripción de una encuesta y lo que hace es crear un encuesta con ese nombre y descripción y añadirla a la lista de sugerencias.
- `getEncuesta()`: es un método al que pasándole un parámetro nombre, nos devuelve la encuesta completa que tenga ese nombre, al igual que el usuario en caso de no haber ninguna devolverá un "null".
- `addEncuesta()`: es un método que devuelve un bool, se le pasa como parámetros el nombre, la descripción y si está activa o no, si no se le pasa este parámetro por defecto saldrá en "false". Este método busca a ver si la encuesta con ese nombre está ya dentro o no y en caso de no haber ninguna con ese nombre, la añadirá y devolverá "true", sino devolverá "false".
- `modificarEncuesta()`: es un método al que le pasamos un nombre y una descripción, buscará el nombre de la encuesta en nuestra lista de encuestas y si lo encuentra modificará la descripción con la nueva descripción, en caso de no encontrarlo devolverá un "false".
- `borrarEncuesta()`: es un método que sirve para borrar una encuesta mediante un parámetro nombre que le pasamos al método, en caso de que encuentre la encuesta, la borrará y devolverá un "true", si no la encuentra o no consigue borrarla devolverá "false".
- `getSugeridas()`: es un método que nos devuelve todas las encuestas que están en la lista de sugerencias.

Pruebas de Integración

En esta clase hemos realizado las siguientes pruebas de integración:

- Prueba para el get de los usuarios: en esta prueba miramos a ver si el count de la lista de usuarios es igual al número que debería salir.
- Prueba para el get de un usuario concreto: lo que hacemos es ir haciendo el `getUsuario()` de una cuenta en concreto y ver si su nombre, id y contraseña se corresponden con las que deberían.
- Prueba del add y del get de una encuesta: añadimos encuestas y miramos a ver si se ha modificado el tamaño de la lista de encuestas con el get.
- Prueba para el get de las activas: aquí miramos a ver si una vez añadidas esas encuestas anteriores con el parámetro de activa en "false", sigue el tamaño siendo el mismo.
- Prueba para el modificar encuesta: intentamos modificar los campos de esa encuesta y miramos a ver con el get, si esos campos se han modificado o no.
- Prueba para el borrar encuesta: intentamos borrar la encuesta y mediante los get de toda la lista de encuestas comprobamos que se hayan borrado o no.
- Prueba para el sugerir una encuesta: probamos a añadir una sugerencia y ver si la lista de sugerencias ha crecido de tamaño.

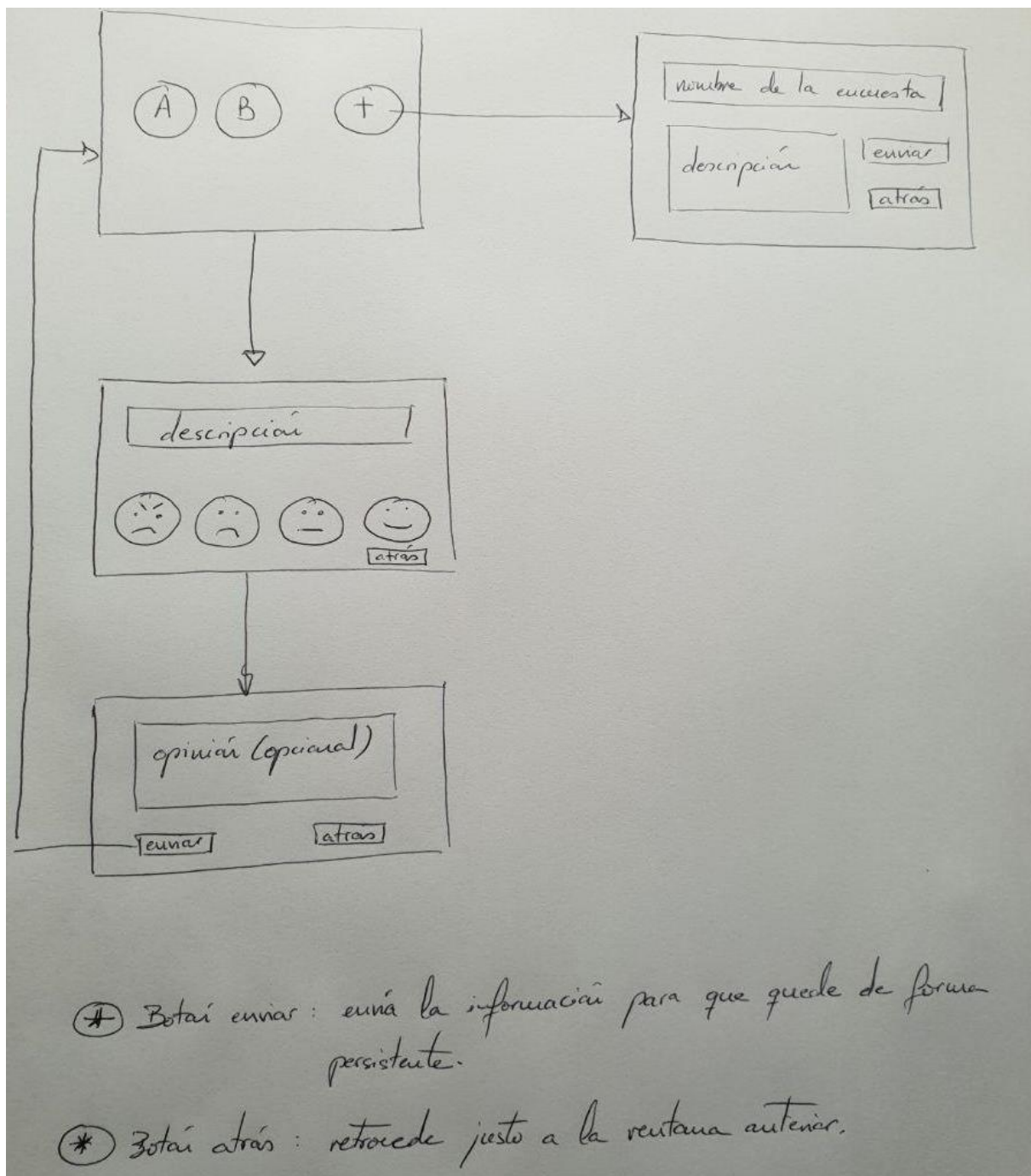


- Prueba para el get de las encuestas que no están activas: miramos a ver si añadiendo la encuesta que debería estar sin activar, se añade a la lista de no activas.

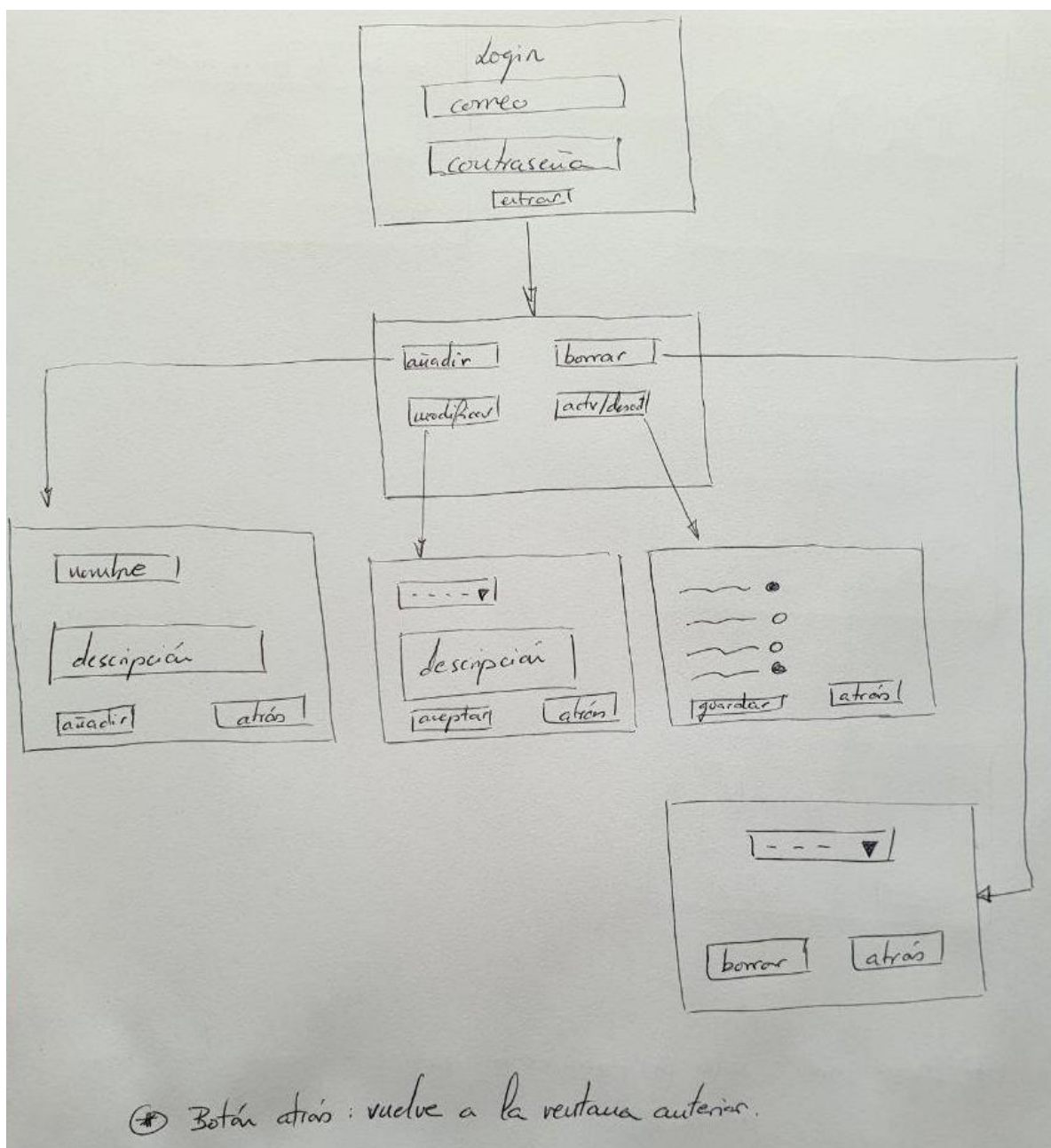
Ventanas de nuestra aplicación

Prototipo inicial de la aplicación (en papel)

Usuario

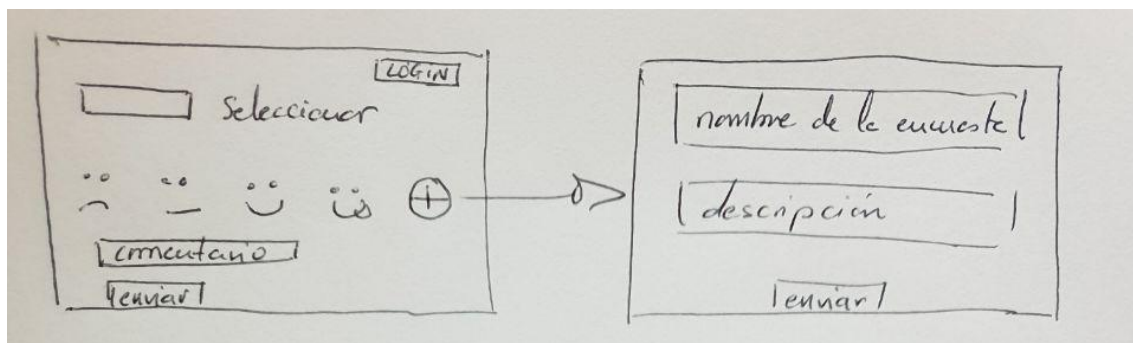


Administrador



Prototipo final de la aplicación (en papel)

Usuario



Administrador

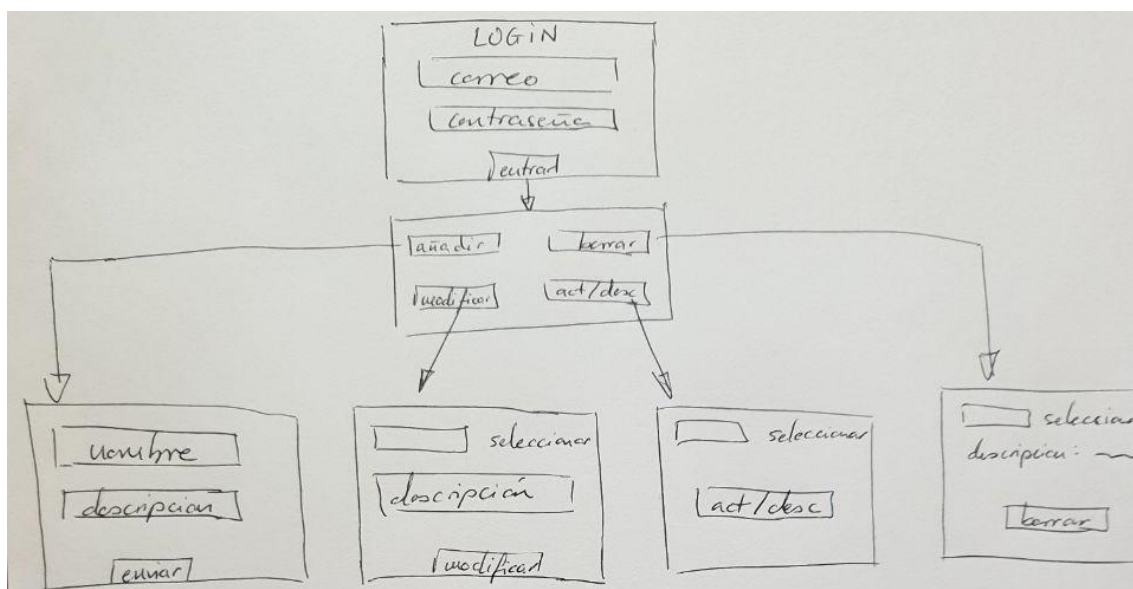
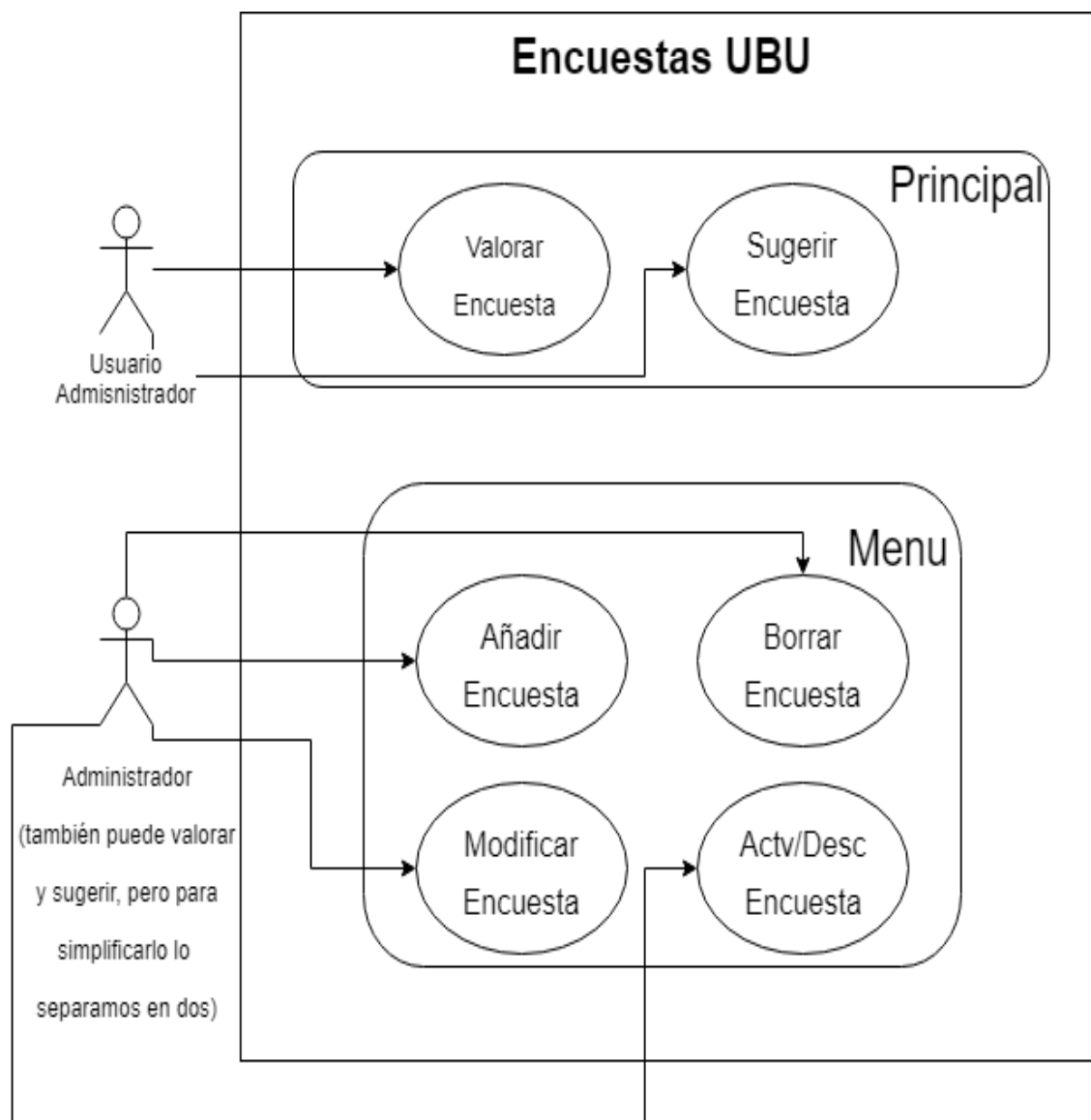


Diagrama Caso de Uso de la Aplicación





Historias de usuario (breve descripción)

Historia de Usuario (Usuario)

Descripción corta: el objetivo es valorar las encuestas que nos proporciona la aplicación, o sugerir encuestas a la aplicación.

Descripción larga: como usuario lo primero que tenemos que hacer es meter la URL de nuestra aplicación, posteriormente llegaremos a la primera ventana en la que podemos seleccionar las diferentes encuestas, o sugerir una nueva encuesta.

Si elegimos una encuesta ya hecha, nos mostrará una ventana en la que se nos mostrará una breve descripción de la encuesta junto con su nombre, y una serie de iconos que pueden ser seleccionados (4 opciones). Cuando elegimos una opción nos dejará escribir una opinión de manera opcional, esta última ventana tendrá el botón aceptar, el cual nos llevará otra vez a las encuestas que hay.

Por lo contrario, si seleccionamos la opción de sugerir accederemos a una nueva ventana, y en ella podremos escribir en un pequeño espacio, la encuesta sugerida.

Pruebas: para poder probar este apartado, hemos considerado las siguientes pruebas:

- Se explicarán en el siguiente punto, ya que haremos un análisis de historias de usuario más pequeñas.



Historia de Usuario (Administrador)

Descripción corta: el objetivo es la gestión de la aplicación.

Descripción larga: como administrador lo que tenemos que hacer es primero meter la URL de nuestra aplicación, posteriormente nos aparecerá un login, en el que tendremos que loguearnos de forma correcta con el correo y su contraseña cifrada.

Una vez logueado correctamente nos aparecerá una nueva ventana en la que tendremos las opciones de gestión, entre ellas están: añadir encuesta, borrar encuesta, modificar encuesta, activar/desactivar encuesta.

Dependiendo de la opción seleccionada nos iremos a una ventana u otra. Al seleccionar la ventana de añadir, nos aparecerá dos cuadros de texto en los que escribiremos el nombre de la encuesta y una breve descripción.

Si elegimos la opción borrar encuesta, entonces nos aparecerá una ventana en la que seleccionaremos la encuesta a borrar, y para que los cambios sean persistentes pulsaremos el botón.

Si seleccionamos la opción modificar encuesta, elegimos la encuesta a modificar y cambiaremos su descripción, finalmente para que los cambios sean persistentes pulsaremos el botón.

Finalmente, si seleccionamos la opción activar/desactivar, podremos elegir cual o cuales de las encuestas queremos que se activen o desactiven, y pulsaremos el botón para hacer los cambios persistentes.

Pruebas:

- Se explicarán en el siguiente punto, ya que haremos un análisis de historias de usuario más pequeñas.



Historias de Usuario

Historias de Usuario 1

Descripción corta: se desea valorar una encuesta.

Descripción larga: como usuario queremos poder ver todas las encuestas disponibles para así poder realizar una valoración.

Lo primero que debemos hacer es acceder a la página principal, ya que es aquí donde se encuentran las encuestas activas sobre las que se puede hacer la valoración.

Posteriormente, la página nos mostrará un listado de las encuestas activas, escribiremos en el cuadro de texto la encuesta correspondiente y daremos al botón seleccionar.

Una vez seleccionada la encuesta, nos mostrará la descripción de la misma, luego seleccionaremos la puntuación que queramos dar, yendo de menor a mayor (siendo la mínima puntuación un 1 y la máxima un 4). Luego, de manera opcional podremos escribir el comentario respectivo.

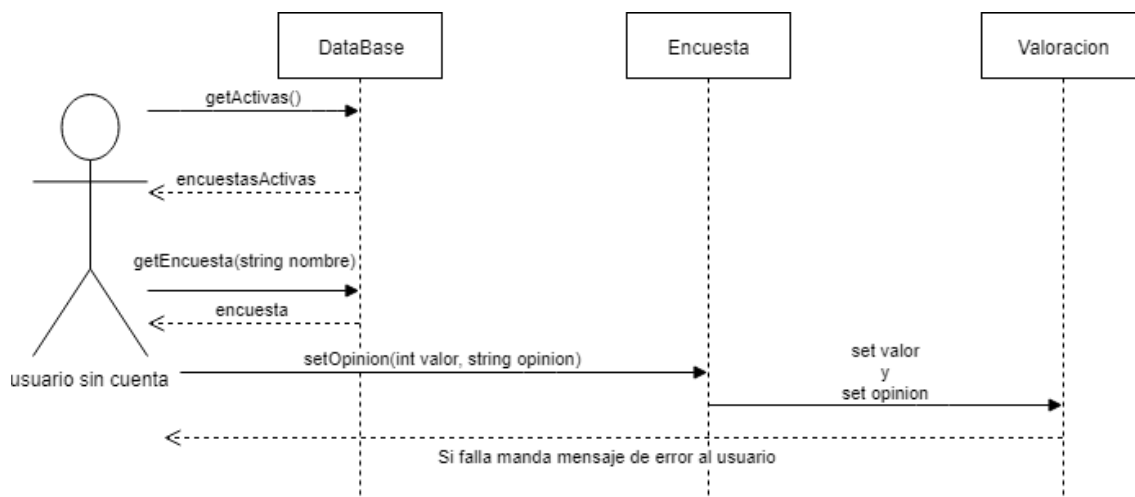
Finalmente, daremos un click al botón enviar, ya que si hemos introducido los campos correctamente habremos enviado la respuesta a la base de datos, de lo contrario nos mostrará un mensaje de error.

Pruebas de aceptación: para realizar este apartado hemos usado el IDE SELENIUM. Las pruebas de aceptación de esta historia de usuario se corresponden al UnitTest1.cs. En dicho fichero realizamos las siguientes pruebas:

- Comprobamos que accedemos a la página principal sin problemas.
- Comprobamos que introduciendo una encuesta activa correctamente, una valoración y una opinión se envía la respuesta de forma correcta.
- Comprobamos que introduciendo una encuesta activa correctamente y una valoración, el envío se hace de forma correcta.
- Comprobamos que introduciendo una encuesta que no existe y una valoración, la página nos muestra un mensaje de error.
- Comprobamos que si no introducimos ninguna encuesta y no valoramos, la aplicación nos muestra un mensaje de error.
- Comprobamos que introduciendo una encuesta activa diferente correctamente, una valoración y un comentario, la respuesta se envía de forma correcta.

Diagrama de secuencia

Aunque el `getActivas()` es algo que hace la propia base de datos hemos considerado que cuando el usuario entra a la página es como si lo pidiera, creemos que también se podría poner reflexivo en la propia base de datos.





Historia de Usuario 2

Descripción corta: se desea sugerir una encuesta.

Descripción larga: como usuario queremos poder realizar la sugerencia de una nueva encuesta.

Lo primero que debemos hacer es acceder a la página Principal, luego seleccionaremos el botón más.

Una vez realizado el paso anterior, accederemos a la ventana SugerirEncuesta, en ella nos aparecerá dos cuadros de texto, introduciremos el nombre de la encuesta en el primer cuadro de texto, y escribiremos la descripción de la encuesta en el segundo cuadro de texto.

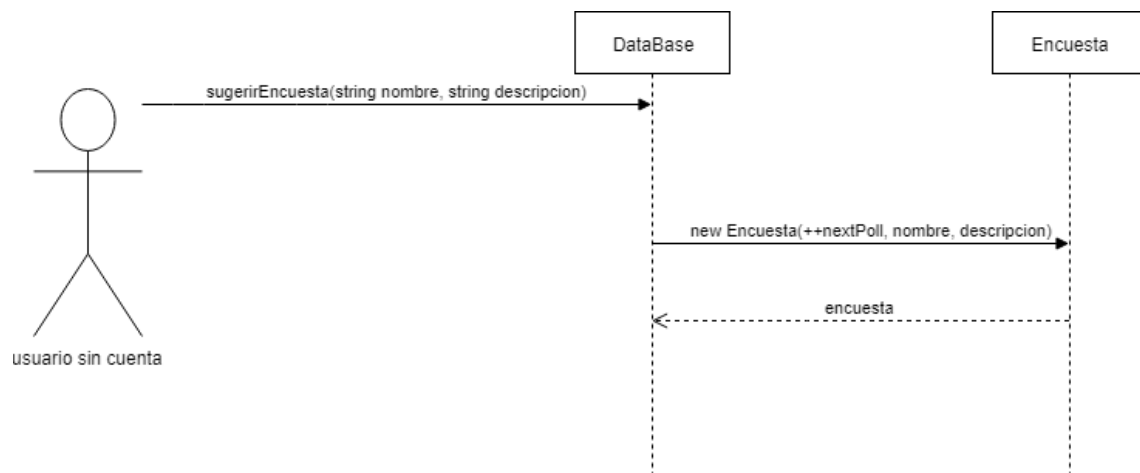
Finalmente, daremos al botón enviar para que se haga persistente la información en la base de datos.

Pruebas de aceptación: esta prueba se corresponde al fichero UnitTest2.cs. En dicho fichero realizamos las siguientes pruebas:

- Comprobamos que primero accedemos a la página principal y que pulsamos el botón más.
- Comprobamos que si introducimos una nueva encuesta y damos al botón enviar, la página nos muestra un mensaje de error, ya que no hemos introducido la descripción de la encuesta.
- Comprobamos que si introducimos una nueva encuesta y añadimos la descripción de la encuesta, se envía dicha respuesta de forma correcta.



Diagrama de secuencia





Historia de Usuario 3

Descripción corta: se desea añadir una encuesta.

Descripción larga: como administradores queremos primero logearnos en la aplicación, y posteriormente añadir una encuesta a la web.

Lo primero que debemos hacer es acceder a la página principal, en ella haremos click en el botón Iniciar Sesión, introduciremos los campos correo y contraseña de forma correcta para así poder acceder al menú de gestión.

Una vez que estamos dentro del menú, seleccionaremos el botón Añadir, de esta forma accederemos a la venta Añadir Encuesta.

Posteriormente, introduciremos el nombre que queremos de la encuesta y su descripción.

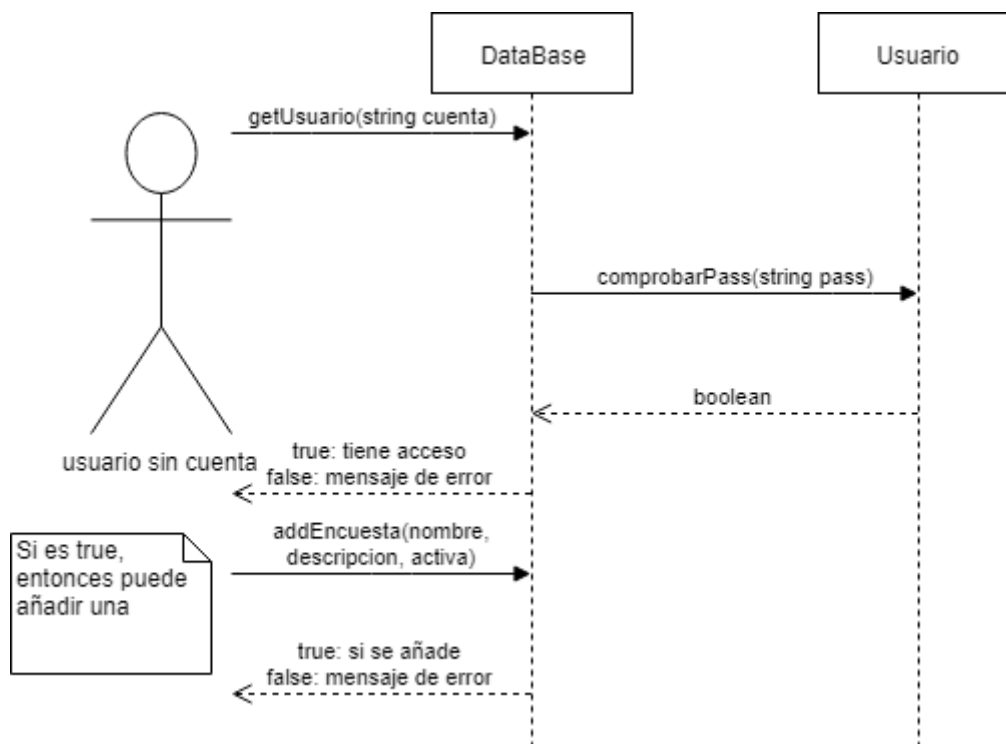
Finalmente, daremos al botón Añadir para así poder hacer que dicha encuesta esté en la base de datos.

Pruebas de aceptación: estas pruebas de aceptación se encuentran en el fichero UnitTest4.cs. Dicho fichero incluye las siguientes pruebas:

- Lo primero que hacemos es acceder a la página Principal, luego damos al botón Iniciar Sesión e introducimos los campos correspondientes, seleccionamos el botón Añadir y ya accedemos a la ventana.
- Comprobamos que si introducimos una encuesta sin descripción la aplicación nos muestra un mensaje de error.
- Comprobamos que si introducimos el nombre de una encuesta que ya existe, e introducimos la descripción, la web nos muestra un mensaje de error.
- Comprobamos que si solo metemos la descripción, la aplicación también nos muestra un mensaje de error.
- Comprobamos que si introducimos el nombre de una encuesta que no existe y su correspondiente descripción, luego pulsamos el botón añadir y la encuesta se introduce de forma correcta en la base de datos.

Diagrama de secuencia

En este caso sería usuario con cuenta, no usuario sin cuenta.





Historia de Usuario 4

Descripción corta: se desea borrar una determinada encuesta.

Descripción larga: como administrador queremos borrar una determinada encuesta de nuestra aplicación.

En primero lugar, tenemos que acceder a la página principal, iniciar sesión, accedemos al menú y seleccionamos el botón Borrar.

Una vez hecho lo anterior accederemos a la ventana de Borrar Encuesta, en dicha ventana nos mostrará todas las encuestas activas y no activas de nuestra aplicación. Posteriormente, introducimos en el cuadro de texto la encuesta que queremos, y damos al botón seleccionar.

Una vez seleccionada la encuesta, nos mostrará la descripción de la misma y daremos al botón borrar para que los cambios se realicen en la base de datos.

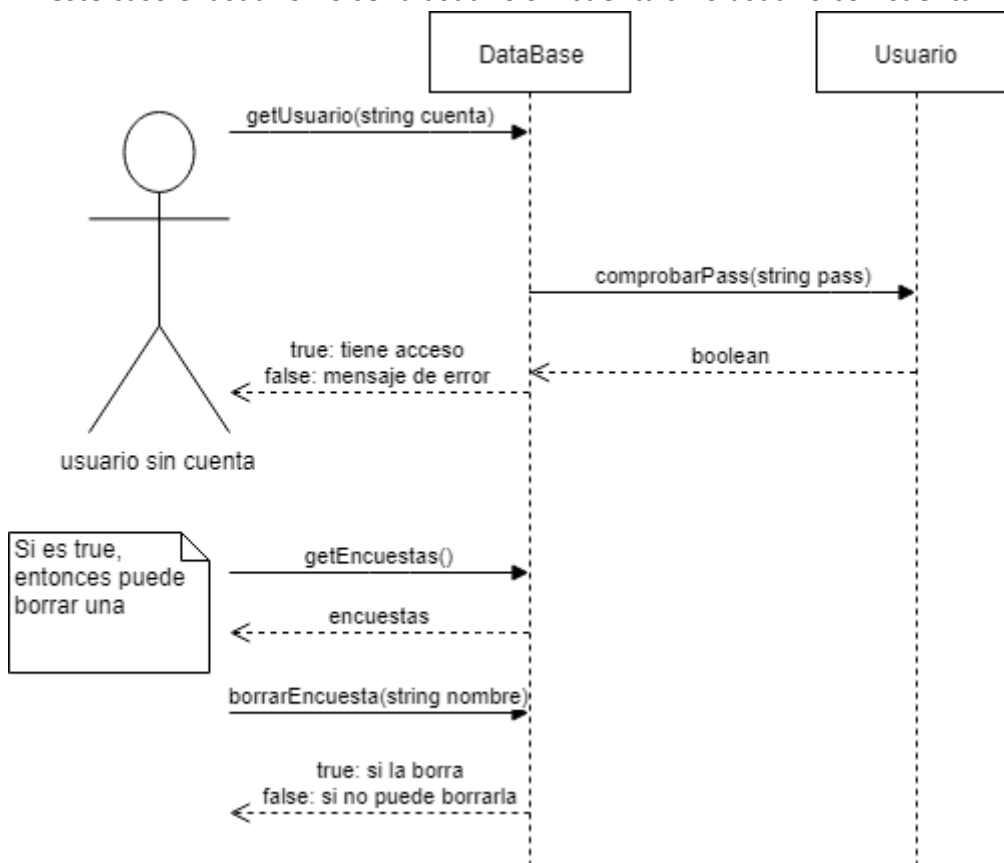
Pruebas de aceptación: dichas pruebas se encuentran en el fichero UnitTest5.cs. Dicho fichero incluye las siguientes pruebas:

- Primero accedemos a la ventana Principal, luego iniciamos sesión, seleccionamos el botón borrar y accederemos a la ventana correspondiente.
- Comprobamos que si no introducimos nada nos da un mensaje de error.
- Comprobamos que si pulsamos el botón seleccionar y no hay ninguna encuesta seleccionada, nos muestra un mensaje de error.
- Comprobamos que si introducimos una encuesta que no existe y damos al botón borrar, nos muestra un mensaje de error.
- Comprobamos que si introducimos una encuesta que existe en el sistema y damos al botón borrar se borra, de tal forma que la información queda persistente en la base de datos.

Diagrama de secuencia

Al igual que antes hemos considerado que el `getEncuestas()` nada más que el usuario accede a ese punto de la página es como si lo pidiera él a la base de datos, pero quizá se podría hacer reflexivo en la propia base de datos.

En este caso el usuario no sería usuario sin cuenta sino usuario con cuenta.





Historia de Usuario 5

Descripción corta: se desea modificar una encuesta.

Descripción larga: como administradores deseamos poder modificar la descripción de una encuesta que esté en el sistema.

Lo primero que tenemos que hacer es acceder a la página Principal, luego iniciamos sesión, y en el menú de gestión seleccionamos Modificar.

Una vez hecho lo anterior, accederemos a la ventana Modificar Encuesta, en la que nos mostrará las encuestas que hay en el sistema, luego escribiremos en el cuadro de texto la encuesta que queremos y damos al botón seleccionar.

Posteriormente introduciremos la descripción nueva que queremos para esa encuesta y daremos al botón modificar para que los cambios se hagan persistentes.

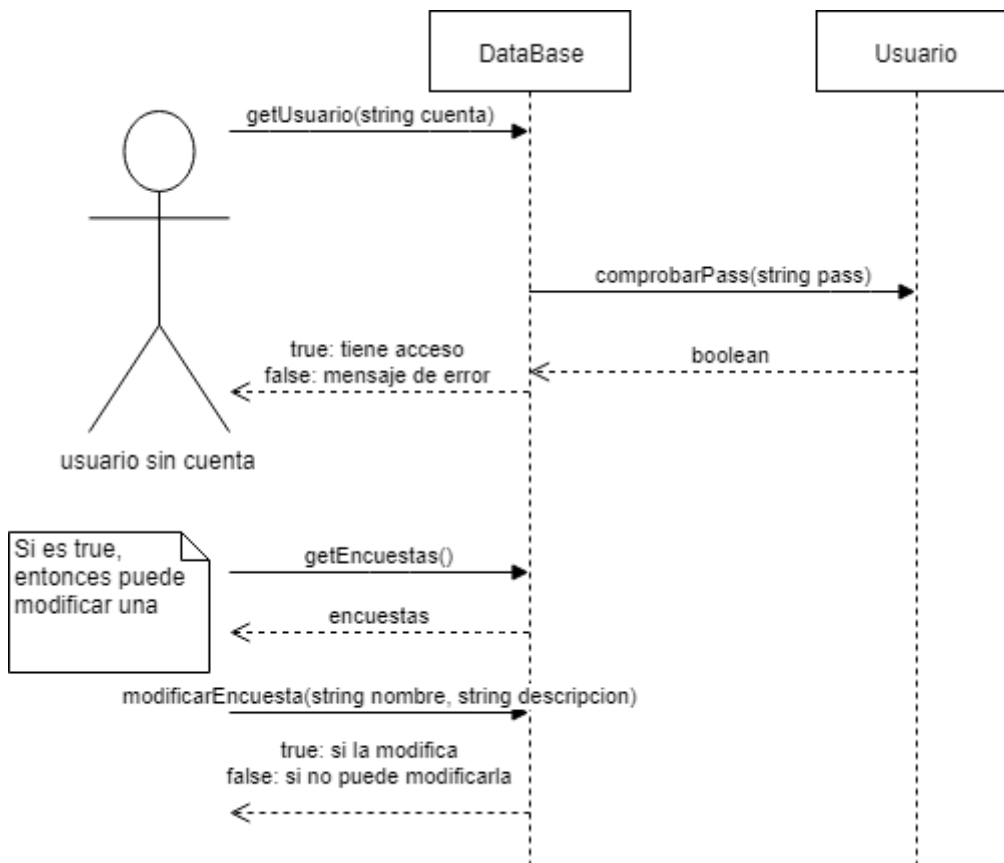
Pruebas de aceptación: estas pruebas están en el fichero UnitTest6.cs. Dicho fichero incluye las siguientes pruebas:

- Lo primero que hacemos es ir a la página Principal, iniciar sesión, y seleccionamos el botón Modificar.
- Comprobamos que si damos al botón Modificar sin introducir ningún campo, nos da un error.
- Comprobamos que si introducimos una encuesta que existe en el sistema, y damos al botón modificar nos muestra un mensaje de error.
- Comprobamos que si escribimos el nombre de una encuesta que existe el sistema e introducimos la descripción, y damos al botón modificar la información cambiada queda persistente en la base de datos.
- Comprobamos que si seleccionamos otra encuesta que exista en el sistema e introducimos su nueva descripción, y damos al botón modificar, la información queda persistente en la base de datos.

Diagrama de secuencia

Al igual que antes hemos considerado que el `getEncuestas()` nada más que el usuario accede a ese punto de la página es como si lo pidiera él a la base de datos, pero quizá se podría hacer reflexivo en la propia base de datos.

En este caso el usuario no sería usuario sin cuenta sino usuario con cuenta.





Historia de Usuario 6

Descripción corta: se desea activar o desactivar una encuesta.

Descripción larga: como administrador queremos activar o desactivar una determinada encuesta que exista en el sistema.

Lo primero que hacemos es acceder a la página Principal, iniciamos sesión, y seleccionamos el botón Activar/Desactivar.

Una vez hecho lo anterior, accedemos a la ventana correspondiente, nos mostrará dos listas, una de ellas son las encuestas activas y la otra las encuestas no activas.

Posteriormente, introduciremos en el cuadro de texto la encuesta seleccionada y daremos click al botón Seleccionar.

Finalmente, daremos click al botón Activar/Deasctivar, dicho botón realizará el cambio, es decir, si una encuesta era activa ahora será no activa, y si una encuesta era no activa ahora será activa.

Pruebas de aceptación: las pruebas correspondientes a esta historia de usuario se encuentran en el fichero UnitTest7.cs. Las pruebas que realizamos en este fichero son las siguientes:

- Primero accedemos a la ventana Principal, iniciamos sesión y accedemos a la ventana Activar/Desactivar encuestas.
- Comprobamos que si no introducimos la encuesta y damos al botón Activar/Desactivar, nos muestra un mensaje de error.
- Comprobamos que si seleccionamos una encuesta que no existe en el sistema, y hacemos click en el botón nos mostrará un mensaje de error.
- Comprobamos que si seleccionamos una encuesta que existe en el sistema, damos a seleccionar, y luego al botón Activar/Desactivar nos realiza el cambio correspondiente y lo hace persistente.
- Comprobamos lo mismo que en el punto anterior, y obtenemos la situación inicial, es decir, que el cambia lo deshace y lo hace persistente.
- Comprobamos que si introducimos otra encuesta que existe en el sistema y damos al botón Activar/Desactivar el cambio se hace persistente.

Diagrama de secuencia

Al igual que antes hemos considerado que el `getActivas()` y el `getNoActivas()` nada más que el usuario accede a ese punto de la página es como si lo pidiera él a la base de datos, pero quizá se podría hacer reflexivo en la propia base de datos.

En este caso el usuario no sería usuario sin cuenta sino usuario con cuenta.

