Departamento de Informática
Universidad Técnica Federico Santa María

Entregable III

Proyecto: BuscaTek

BUSCATEK

Integrantes:

Nombres y Apellidos	Email	ROL USM
Diego Jara Caballero	diego.jara.13@sansano.usm.cl	201304530-8
Tomás Gómez Molina	tomas.gomez.13@sansano.usm.cl	201373078-7

Modelo de Calidad

Los Stakeholders elegidos son: Profesor Administrador y Usuario final.

El Profesor Administrador sería el principal interesado en todo lo que involucre Funcionalidad, Confiabilidad, Administración y Usabilidad, esto según lo comunicado por él mismo en el intercambio de información que se ha tenido. Es el cliente que en este caso sería el profesor Maximiliano Rivera y por esto es esencial dentro de los Stakeholders. Para la funcionalidad, el cliente espera que el servicio contenga las funciones específicas de búsqueda con clasificación por perfiles, donde la información es recopilada desde sitios confiables. Por otro lado, el sistema debe tener una fácil administración y una usabilidad que permita que el usuario final pueda utilizar el servicio de búsqueda sin mayor problema.

El Usuario final es cada estudiante o usuario en general, que utilice el sistema de búsqueda con clasificación por personalidades, es importante que esté presente en el modelo de calidad, pues es el público objetivo que utilizará con frecuencia el servicio. Se considera que sus preocupaciones principales son la Usabilidad, Seguridad y la Eficiencia. Se consideró que estos tres atributos contribuyen a la constitución de un servicio de calidad para el usuario, entregando una experiencia en la que el servicio es fácil de utilizar, funciona de manera rápida y eficiente y protege toda su información sin peligro de mal uso de esta.

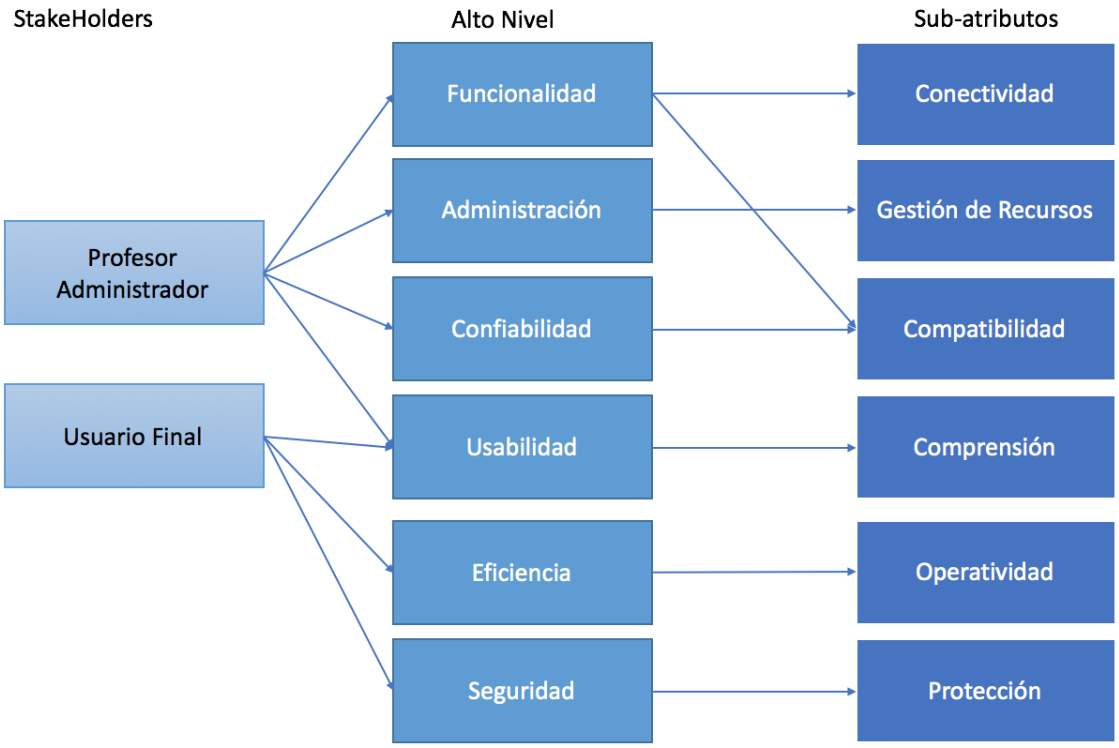
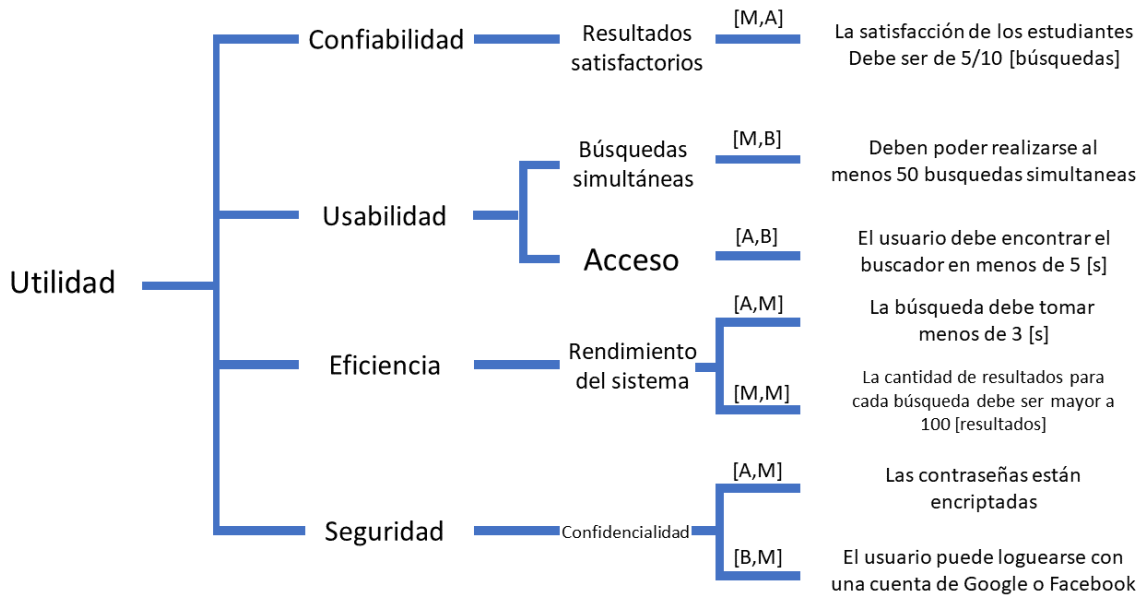


Ilustración 1: Modelo de Calidad del proyecto, con sus Stakeholders, atributos y sub-atributos.

Árbol de utilidad (actualización)



Las nuevas utilidades son:

- Encriptación de contraseñas
- Registro con una cuenta de RRSS.
- Acceso al buscador

El acceso al buscador tiene una prioridad alta ya que fue fácil de implementar y es de suma importancia ya que la claridad y rapidez con la que se encuentra esta funcionalidad le facilita la vida al usuario en gran medida. La encriptación de las contraseñas también es de alta prioridad, pero más difícil de implementar.

Pruebas de Software

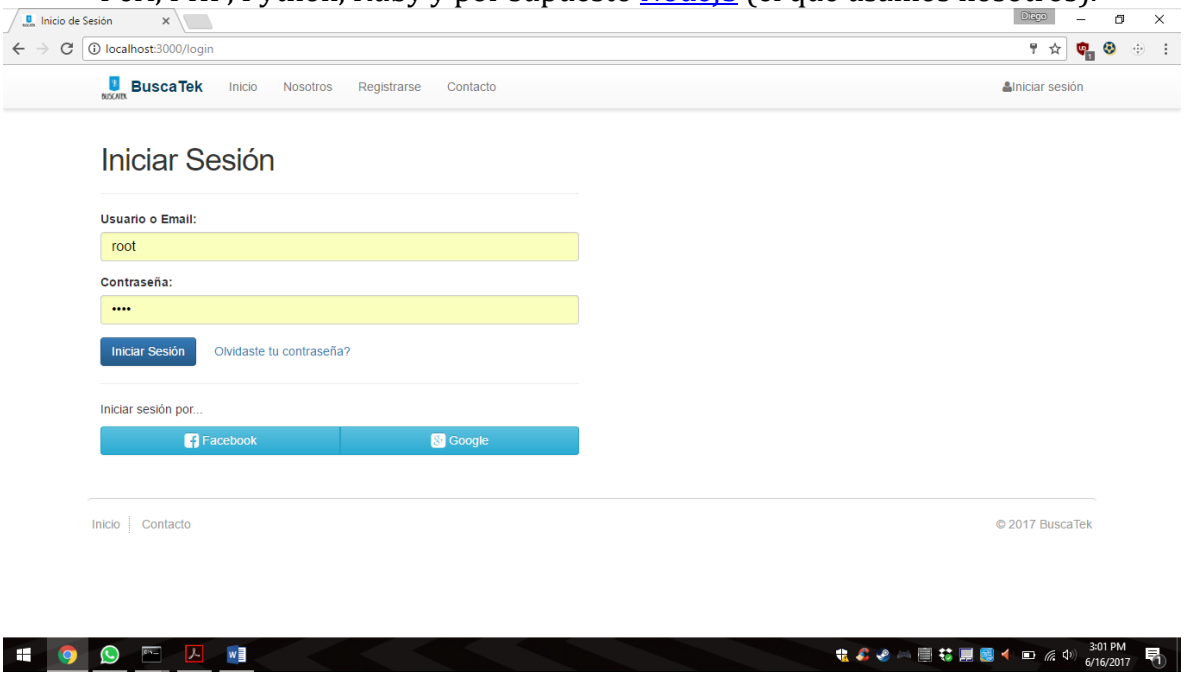
Para las pruebas del software nos contactamos con el QA Francisco Amaro Díaz Águila, nos reunimos y obtuvimos el feedback correspondiente, así como el detalle de las pruebas de software efectuadas. Los casos de prueba a realizarse son:

- Seguridad: encriptación de la contraseña, la razón de la exigencia de encriptación es para mantener un nivel de seguridad alto en cuanto a los datos del usuario.

- Rendimiento: tiempo de respuesta del servidor ante una búsqueda, esto para brindar un servicio rápido al usuario, sin que este tenga que perder tiempo en espera.
- Usabilidad/Seguridad: problemas de replicación de identidad (hacer consultas desde 2 pestañas), esto va en conjunto entre la usabilidad y la seguridad, si una persona quiere tener dos pestañas con el servicio en uso, este debiera mantener su sesión en ambas pestañas. Además, afecta a la seguridad del programa, ya que no se están chequeando las credenciales correspondientes.
- Resultados: que los resultados de la búsqueda sean mostrados en un orden correcto, esto es claro, ya que debe mostrarse de manera fácil y ordenada.
- Resultados: que la información mostrada sea relevante. Esto ya que se debe buscar en sitios web que sean confiables y relevantes con el tema tratado de manera académica.

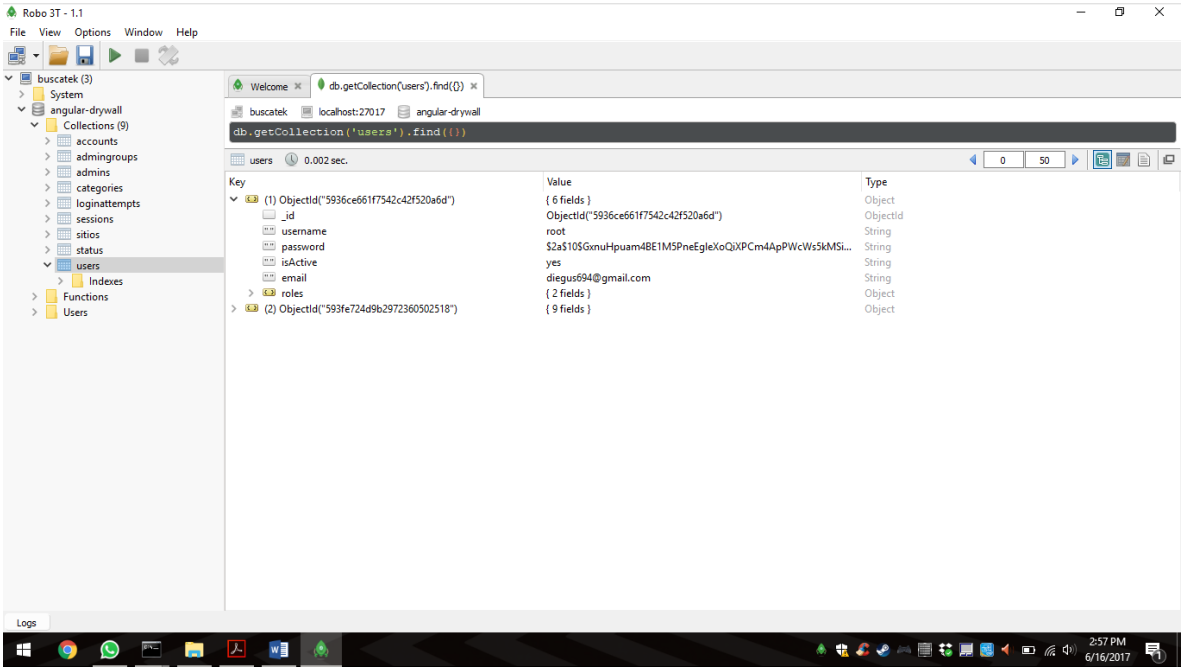
Resultados:

1. La prueba de la encriptación de la contraseña se pasó, ya que usamos bcrypt, el cual es una función generadora de contraseñas mediante hashing. Esta función se creó en 1999 pero hay implementaciones para C, C#, Go, Java, JavaScript, Perl, PHP, Python, Ruby y por supuesto NodeJS (el que usamos nosotros).



En este caso el administrador inicia sesión con,

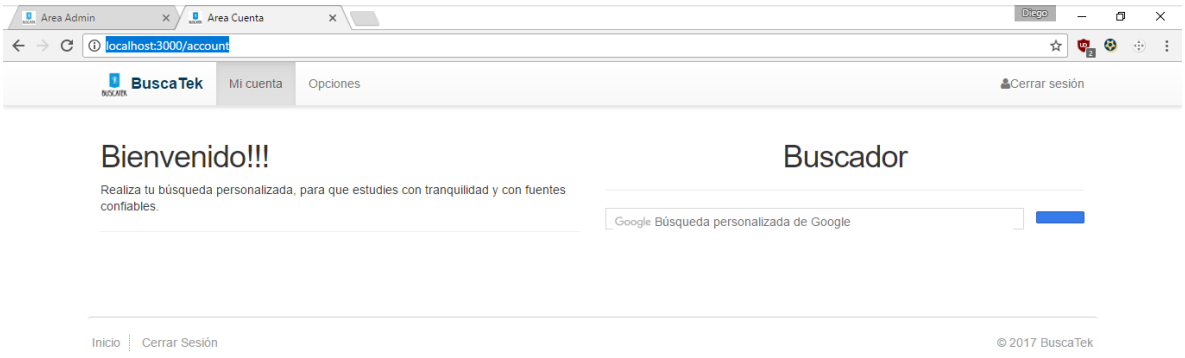
Usuario: root
Contraseña: root

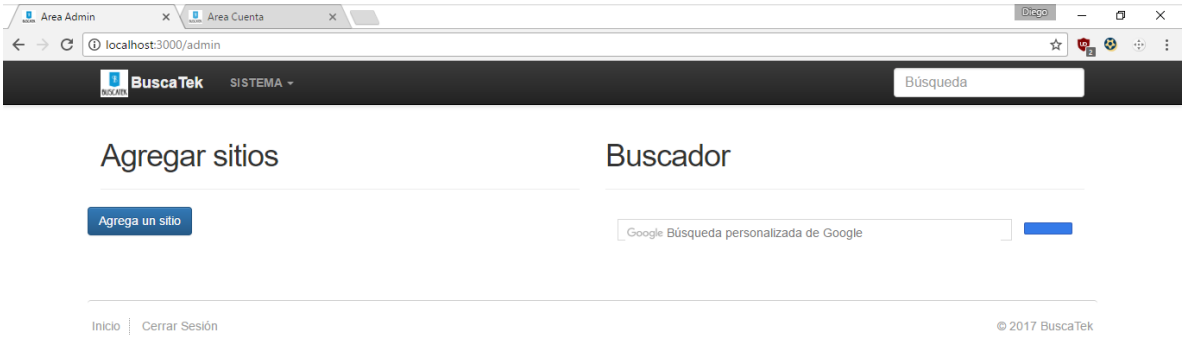


Al ver la "BD", se puede apreciar que el campo *password* es una clave hash generada con la función de bcrypt tomando como argumentos la contraseña ingresada por el usuario y una llave secreta (administrada por nosotros). Por lo tanto la contraseña está encriptada y cumple con la primera prueba.

2. La prueba del tiempo de respuesta de la búsqueda se pasó porque usamos una API de Google para buscar, por lo tanto, los tiempos de respuesta que tenemos en nuestra aplicación web son los mismos tiempos de respuesta que nos garantiza Google con esta herramienta. Por esto, el rendimiento de nuestra aplicación solo depende de la conexión a internet que el usuario tenga, y claramente eso no depende de nosotros.

3. La prueba de la "replicación de identidad" se falló, ya que en una misma ventana del navegador se pudo ingresar con dos cuentas distintas, como se puede ver en las imágenes.

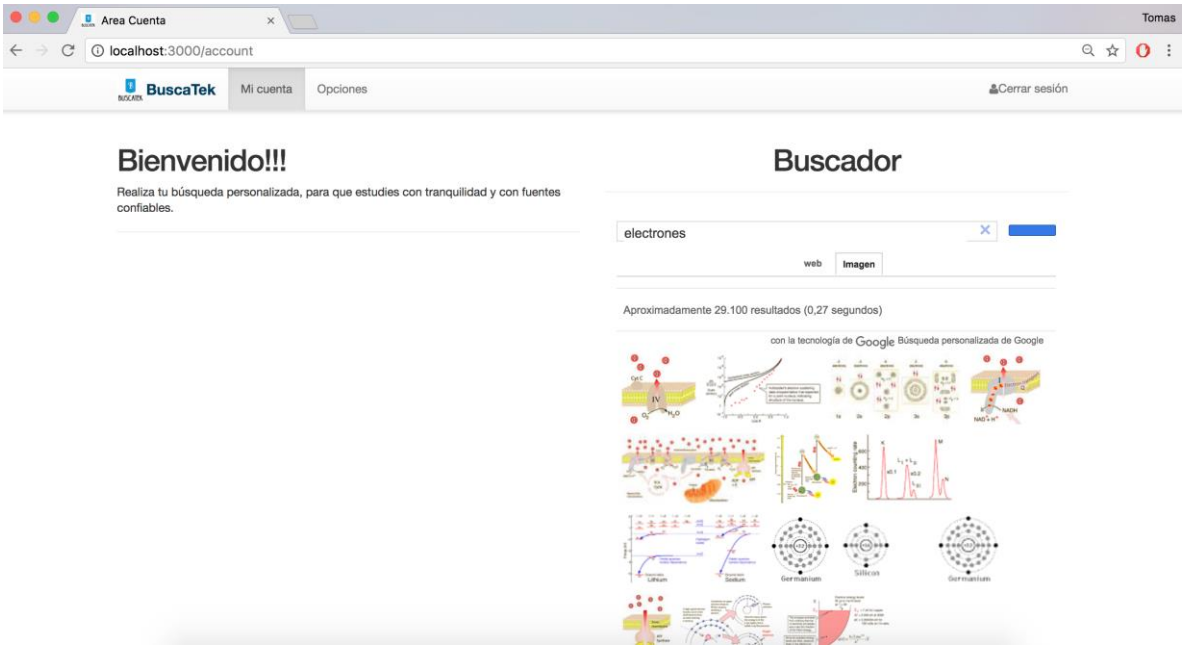


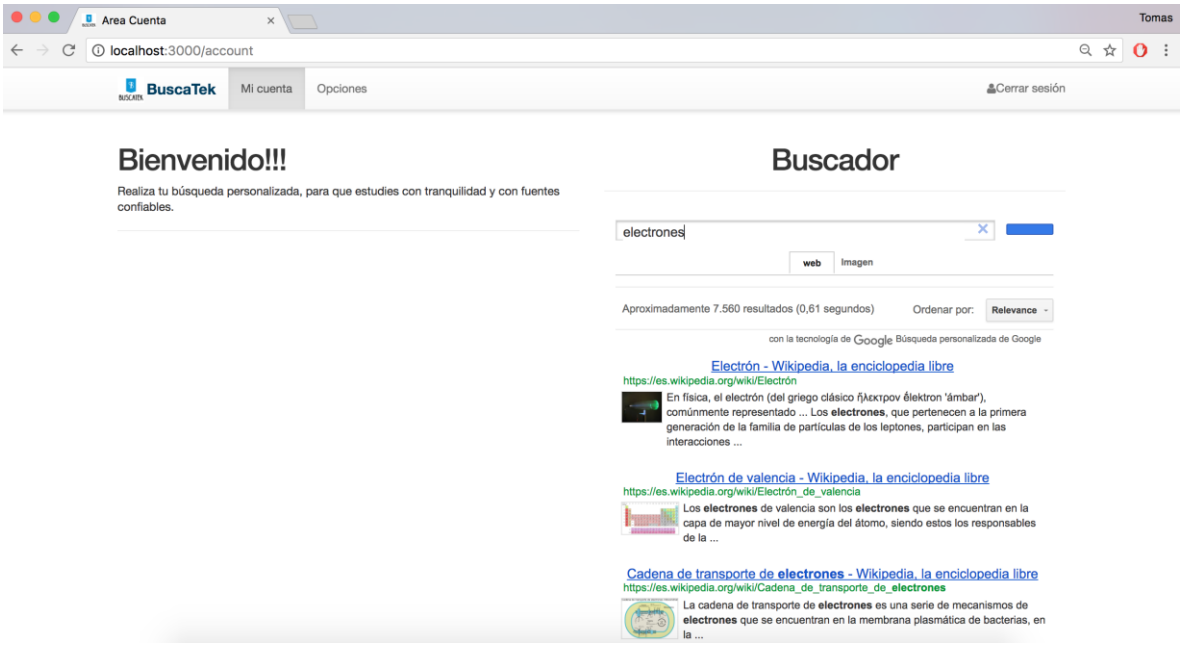


4. La prueba de búsqueda en la cual se requiere un orden correcto en los resultados mostrados cumplió con lo esperado. La muestra de los datos sigue un orden, el cual está definido por la API de Google Custom Search Engine.

5. La prueba de búsqueda en la cual se requiere que los datos sean relevantes, y por lo tanto confiables y respondan al requerimiento del usuario, es correcto. Como método de prueba se ajustó la búsqueda sobre páginas del dominio

"Wikipedia", aun así, se cumple que la información recopilada responde el tema que fue buscado.





Plan de Mejoras:

Dentro de todas las pruebas realizadas se encontraron pocas posibles mejoras, una de estas es la única prueba que falló, en la que se puede replicar la identidad en dos diferentes pestañas. Por lo tanto, se creó un plan de mejora enfocado directamente en

corregir este problema. Considerando que luego de varias pruebas e intentos no se ha corregido, se deja pendiente para la última entrega esta mejora, la cual es la única a realizar con respecto a las pruebas realizadas. Por supuesto, aún queda agregar la clasificación de la búsqueda según perfiles, pero considerando que eso no está implementado se considera como trabajo futuro y no mejora.

QA:

a) La participación del QA Francisco Díaz, fue buena, estuvo preocupado de probar el software y se concibió una reunión para observar y analizar los resultados de aquellas pruebas. Con el QA Andreas Aravena se mantuvo un par de conversaciones, las cuales no culminaron en mucha información de feedback.
b) Las pruebas realizadas fueron un total aporte, esto debido a que se encontraron posibles mejoras y se observó el funcionamiento desde una visión exterior de los desarrolladores del proyecto, lo que generó bastantes ideas acerca del futuro del proyecto, lo que falta, lo que se llevará finalmente a cabo y lo que no.
c) Para el QA Francisco Díaz, la calificación decidida por el grupo es de 5, ya que se preocupó de generar feedback y ayudar a mejorar el proyecto. Acerca del segundo QA Andreas Aravena, preferimos no calificar, ya que no hubo mayor contacto con este, pudiendo haberse comunicado entre QA's, por lo tanto, como grupo se omite la calificación.

Formulación de pruebas de software de requisitos no funcionales

Esta sección se omite, ya que las pruebas de software planteadas por el QA en la sección anterior se basaron en los requisitos no funcionales, por lo tanto, toda la descripción al respecto está en la sección anterior.

Listado de Pruebas/Requisitos

Paso 1:

ID	Nombre del Requerimiento	Roles asociados	Descripción
RF1	Ingreso y salida del sistema	1) Profesor Administrador 2) Usuario Final	Profesor Administrador: Un Profesor Administrador puede ingresar al sistema al sistema usando sus credenciales. Usuario Final: El usuario puede ingresar al sistema

			usando sus credenciales. Una vez que ambos ingresen, el sistema deberá mostrar el inicio dependiendo del tipo de usuario que haya ingresado.
RF2	Editado de información	3) Profesor Administrador 4) Usuario Final	Profesor Administrador: El Profesor Administrador puede editar los datos de cada usuario incluyendo aquellos que también sean administradores. También tiene la posibilidad de borrar cualquier usuario. Usuario Final: El Usuario Final puede editar en cualquier momento sus datos personales, en caso de necesitar actualizar alguna información. Ambos paneles de edición son diferentes, uno es personal, mientras el de administrador muestra todos los perfiles de usuarios creados.
RF3	Mostrar búsqueda	5) Usuario Final	Usuario Final: El Usuario Final, debe al realizar una búsqueda, poder ver los resultados de esta en su inicio de sesión.
RF4	Realizar búsqueda	6) Usuario Final 7) Profesor Administrador	Usuario Final: El Usuario Final, debe poder realizar una búsqueda de algún tema específico, dentro del dominio de sitios permitidos por el Administrador. Profesor Administrador: El Administrador, debe poder editar el dominio de sitios permitidos para realizar la búsqueda del tema escogido por el usuario. Por ahora, esto se está haciendo de forma manual dentro del panel de control en la API de Google Custom Search Engine.

Paso 2:

- RT1: El usuario no debe estar en blanco
- RT2: La contraseña no debe estar en blanco
- RT3: Si el usuario y la contraseña son válidos. Ingresar.

- RT4: Se puede editar cualquier dato así que se pueden dejar en blanco algunos campos. Si se quiere editar contraseña hay que ingresarla dos veces.
- RT5: La búsqueda no debe estar en blanco.
- RT6: La búsqueda depende de la conexión a internet. Hay que estar conectado a internet.

Paso 3:

Caso de Prueba #	RT#	Caso de Prueba	Pasos del caso	Datos utilizados	Resultados obtenidos
CP1	RT3	Verificar ingreso	1) Ingresar a la página de login 2) Ingresar usuario 3) Ingresar contraseña 4) Click en "Iniciar Sesión"	usuario= root Contraseña= root	Ingreso exitoso
CP2	RT4	Verificar edición	1) Iniciar sesión 2) Click en "Sistema" e ir a "Usuarios" o "Cuentas" 3) Hacer click en "Edit" 4) Ingresar nuevos datos 5) Hacer click en cualquiera de los botones para editar.	Usuario anterior: diego Usuario nuevo: diego2	Edición exitosa. Ahora para iniciar sesión requiere diego2 en vez de diego.
CP3	RT6	Verificar mostrado de búsqueda	1) Iniciar sesión 2) Escribir datos en buscador	Búsqueda: "campo eléctrico"	Búsqueda exitosa
CP4	RT5	Verificar búsqueda realizada	1) Iniciar sesión 2) Escribir datos en buscador	Búsqueda: ley de gauss	Búsqueda exitosa

Paso 4:

Caso de Prueba #	RF #	RT#	Caso de Prueba	Pasos del caso	Datos utilizados	Resultados obtenidos
CP1	RF1	RT3	Verificar ingreso	1) Ingresar a la página de login 2) Ingresar usuario 3) Ingresar contraseña 4) Click en "Iniciar Sesión"	usuario= root Contraseña= root	Ingreso exitoso
CP2	RF2	RT4	Verificar edición	1) Iniciar sesión 2) Click en "Sistema" e ir a "Usuarios" o "Cuentas" 3) Hacer click en "Edit" 4) Ingresar nuevos datos 5) Hacer click en cualquiera de los botones ("Actualizar", "Link", "Cambiar contraseña" y "Borrar") para editar.	Usuario anterior: diego Usuario nuevo: diego2	Edición exitosa. Ahora para iniciar sesión requiere diego2 en vez de diego.
CP3	RF3	RT6	Verificar mostrado de búsqueda	1) Iniciar sesión 2) Escribir datos en buscador	Búsqueda: "campo eléctrico"	Búsqueda exitosa
CP4	RF4	RT5	Verificar búsqueda realizada	1) Iniciar sesión 2) Escribir datos en buscador	Búsqueda: ley de gauss	Búsqueda exitosa