OpenGId.

Un servidor OpenId de autenticación gráfica.

Nombre Estudiante
Hernán Fernandez Lescano

Nombre Consultor
Cristina Pérez Solà

Fecha Entrega
Junio 2014

FICHA DEL TRABAJO FINAL

Título del Trabajo:	*OpenGid. Un servidor OpenId de autenticación gráfica*
Nombre del autor:	*Hernán Fernandez Lescano*
Nombre del consultor:	*Cristina Pérez Solà*
Fecha de entrega (mm/aaaa):	*06/2014*
Área del Trabajo Final:	*Seguridad Informática*
Titulación:	*Grado Ingeniería Informática*

Resumen del Trabajo (máximo 250 palabras):

El crecimiento de la Web 2.0 y las nuevas tendencias de computación en la nube, están incrementando considerablemente los servicios ofrecidos en internet, como redes sociales, espacio de almacenamiento, herramientas de comunicación y aplicaciones de trabajo colaborativas entre otras. Aquellos usuarios que utilicen estos servicios normalmente habrán de proporcionar sus datos personales y crear una contraseña para poder utilizarlos. Esta situación genera una serie de inconvenientes como el problema de memorización si el usuario utiliza contraseñas diferentes para cada uno de ellos o un problema de seguridad si se diseminara una misma contraseña en múltiples sitios o se utilizaran contraseñas simples fácilmente recordables pero vulnerables.

El presente trabajo propone una solución para estas problemáticas, mediante el uso de autenticación por tercera parte, para centralizar el proceso de autenticación y utilizar solo una contraseña para acceder a los distintos servicios y propone también el uso de contraseñas gráficas, para crear contraseñas robustas y fácilmente recordables.

El trabajo incluye el prototipo de un servidor de autenticación gráfica que utiliza el método PassPoint para contraseñas gráficas y el protocolo OpenId para la autenticación externa. El prototipo es usado para experimentar, analizar y evaluar la viabilidad de una implementación real de la solución propuesta.

Abstract (in English, 250 words or less):

The growth of Web 2.0 and the new trends in cloud computing are significantly increasing the services offered over the Internet, such as social networking, storage, communication tools and applications of collaborative work among others. People who use these services have to provide their personal information and create a password. This situation raises a number of problems such as the memorization difficulties if the user uses different passwords or the security problem of disseminating the same password on multiple sites, or the usage of simple passwords, which are easy to memorize but vulnerable.

This thesis proposes a solution to these problems, by using a third party authentication system, to centralize the authentication process and use an only password to access the different services. It also proposes the usage of graphical passwords to create strong ones but easy to memorize.

This thesis also includes a prototype of a graphical authentication server using the PassPoint method which is a graphical passwords method and the OpenId protocol for external authentication. The prototype is used to experiment, analyze and assess the feasibility of a real implementation of the proposed solution.

Palabras Clave (entre 4 y 8):

Autenticación, contraseñas, gráficas, OpenId, seguridad, servidor.

Índice

Lista de figuras

1. Introducción

1.1 Contexto y justificación del Trabajo

El presente trabajo consiste en la implementación de un prototipo de servidor de autenticación gráfica externa o por tercera parte (***third party authentication***), el cual será denominado OpenGId.

OpenGId ofrece una alternativa al método clásico de autenticación, que pretende resolver las siguientes problemáticas:

- Memorización de contraseñas: Con el objetivo de aumentar la robustez de contraseñas alfanuméricas, estas han de ser largas e incluir símbolos difícilmente recordables.

- Múltiples registros en sistemas: La mayoría de servicios requieren que el usuario se registre en el sistema para poder ser identificados al utilizarlos. El hecho de que un usuario tenga que distribuir sus datos personales en cada sistema utilizado incrementa la posibilidad de que un usuario malintencionado pueda acceder a ellos.

La autenticación externa o por tercera parte, permite a un usuario autenticarse frente a un servidor sin la necesidad de registrarse en el mismo. Para proveer el servicio de autenticación externa, se ha optado por utilizar el protocolo OpenId, el cual es un estándar abierto destinado a este fin y que está siendo ampliamente utilizado por organizaciones como Facebook, Yahoo o Google entre otras.
Lo novedoso en esta propuesta de sistema de autenticación es que ofrece servicios para poder identificarse mediante contraseñas gráficas, lo que permite disponer de contraseñas más robustas que las alfanuméricas y con mayor facilidad de memorización.
El sistema también garantiza la seguridad de las contraseña del usuario aplicando las medidas de seguridad pertinentes para prevenir los posibles ataques a sus bases de datos, como también la de proveer el poder computacional para llevar a cabo todo el proceso de autenticación.
A continuación se muestra un gráfico con las partes implicadas en el sistema:

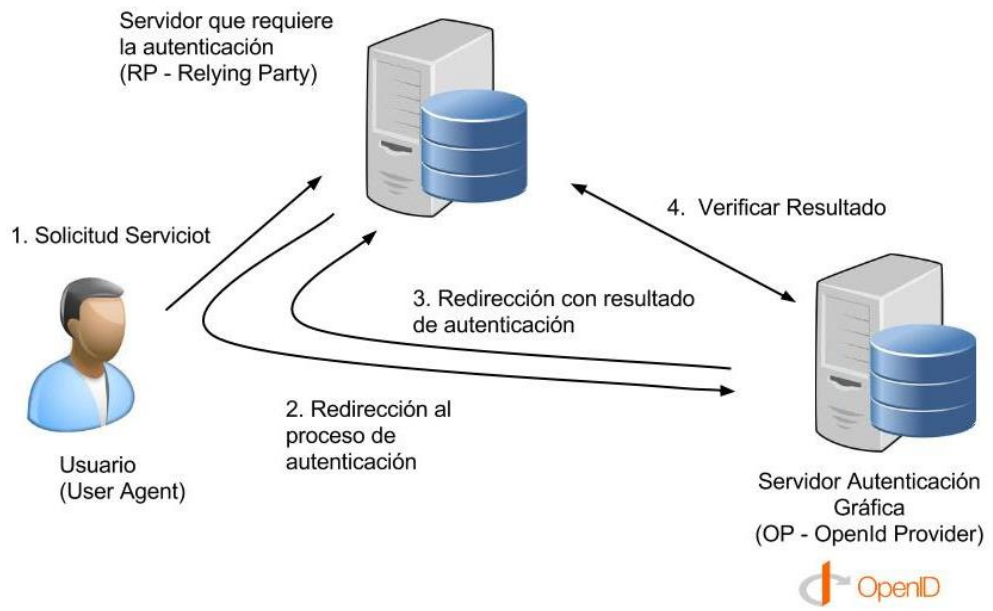


Ilustración 1. Interacción entre partes del sistema.

1.2 Objetivos del Trabajo

Mediante la implementación de este prototipo de sistema de seguridad, se espera cumplir con los siguientes objetivos:

● Realizar un análisis de viabilidad y aceptación de un esquema de autenticación como el propuesto.

● Evaluar las ventajas e inconvenientes de utilizar contraseñas gráficas.

● Evaluar las ventajas e inconvenientes de realizar una autenticación mediante una tercera parte.

● Analizar y estudiar los sistemas de contraseñas gráficas existentes.

1.3 Enfoque y método seguido.

La estrategia a seguir será la de adaptar un producto ya existente. Se adaptará el método de autenticación gráfica **PassPoint** y se lo integrará dentro del protocolo **OpenId** para realizar la autenticación.
De esta manera, se podrá usar la contraseña gráfica dentro de un estándar de autenticación, lo cual aumentará su utilidad.
Por otro lado, al ser un servidor dedicado, todo el espacio de almacenamiento y poder computacional requeridos caerá sobre el

servidor de autenticación liberando tanto al cliente como al proveedor del servicio que requiera la autenticación.

1.4 Planificación del Trabajo

		Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1	✓	Investigacion estado del arte	6 días?	lun 03/03/14	dom 09/03/14	
2	✓	Confección Pac1	5 días?	lun 10/03/14	dom 16/03/14	
3	✓	Entrega Pac1	1 día	lun 17/03/14	lun 17/03/14	
4		**Diseño**	**10 días**	**lun 17/03/14**	**vie 28/03/14**	
5	✓	Analisis de Requisitos	5 días	lun 17/03/14	vie 21/03/14	
6		Diseño del Sistema	5 días	lun 24/03/14	vie 28/03/14	5
7		**Implementacion**	**20 días?**	**lun 31/03/14**	**vie 25/04/14**	**4**
8	✓	Configuracion Servidores	1 día	lun 31/03/14	lun 31/03/14	
9		Implementacion OpenId	4,5 días?	mar 01/04/14	lun 07/04/14	8
10		Implementacion GUI	10 días	lun 07/04/14	vie 18/04/14	
11		Implementacion BD	7 días	mar 01/04/14	mié 09/04/14	8
12		Implementación Algoritmos	10 días	lun 07/04/14	vie 18/04/14	
13		Integración Componentes	5 días	lun 21/04/14	vie 25/04/14	10;11;12
14	✓	Confección Pac2	5 días	lun 07/04/14	vie 11/04/14	
15		Entrega Pac2	1 día	lun 14/04/14	lun 14/04/14	
16		Test y Encuestas	5 días	lun 28/04/14	vie 02/05/14	13
17		Evaluación de métricas	5 días	lun 05/05/14	vie 09/05/14	16
18		Evaluación de viabilidad	5 días	lun 12/05/14	vie 16/05/14	17
19		Confeccion Pac3	5 días	lun 12/05/14	vie 16/05/14	
20		Entrega Pac3	1 día	lun 19/05/14	lun 19/05/14	
21		Confeccion Memoria	19 días?	mar 20/05/14	jue 12/06/14	
22		Entrega Pac4	1 día	vie 13/06/14	vie 13/06/14	
23		Confeccion Presentación	10 días?	lun 09/06/14	vie 20/06/14	
24		Entrega Presentacion	1 día	vie 20/06/14	vie 20/06/14	

Tabla 1. Lista de tareas y su planificación.

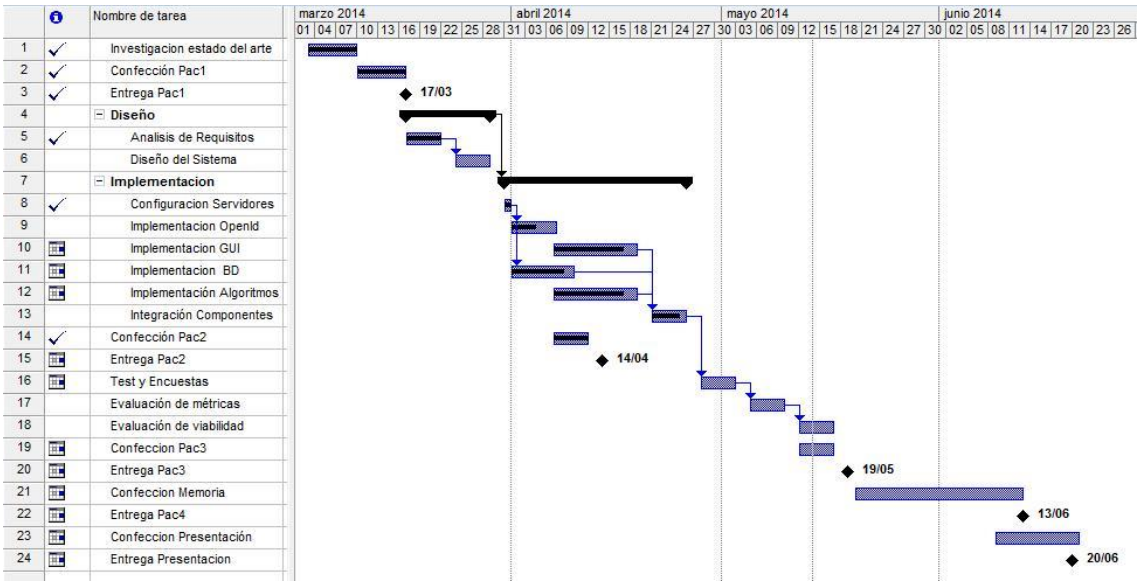


Ilustración 2. Diagrama de Gantt

1.5 Breve descripción de los productos obtenidos

El producto resultante de este trabajo, será un prototipo funcional de un servidor de autenticación gráfica externa junto a un estudio sobre el uso de contraseñas gráficas y viabilidad del sistema.

1.6 Breve descripción de otros capítulos de la memoria

En el siguiente capítulos se explicará cual es el estado del arte, tanto de las contraseñas gráficas como los sistemas de autenticación. En los capítulos 3 y 4 se explica cómo se ha llevado a cabo el diseño e implementación del prototipo. El capítulo 5 presenta cuales son las ventajas y desventajas de utilizar un sistema de contraseñas gráficas. Seguidamente en el capítulo 6, se da a conocer la experiencia de los usuarios después de probar el sistema. El capítulo 7 recoge un estudio de viabilidad desde el punto de vista económico según la magnitud del número de usuarios. Finalmente en el capítulo 8 se exponen las conclusiones recogidas a lo largo de todo el desarrollo del presente trabajo. Los anexos contienen información técnica del protocolo OpenId como también las instrucciones de uso del sistema.

2. Estado del Arte.

La utilización de contraseñas gráficas surge como una alternativa para intentar reducir la principal problemática existente con las contraseñas alfanuméricas que es el siguiente:

- Dificultad para el usuario de recordar contraseñas de elevado nivel de seguridad, y es por este mismo motivo que se escogen contraseñas simples, que son muy fáciles de adivinar o deducir si se conocen datos personales del usuario, o bien con ataques de diccionario o en su defecto por fuerza bruta.

Este problema se evidencia aún más cuando año tras año se dan a conocer los ranking de contraseñas que más han sido quebrantados, y las contraseñas siguen siendo prácticamente las mismas.
La probabilidad de acceder a un sistema de manera fraudulenta, utilizando alguna de las contraseñas listadas en el ranking mostrado en la tabla a continuación, es considerablemente alta.

Posición	***Contraseña***
1.	123456
2.	password
3.	12345678
4.	qwerty
5.	abc123
6.	123456789
7.	111111
8.	1234567
9.	iloveyou
10.	adobe123
11.	123123
12.	admin
13.	1234567890
14.	letmein
15.	photoshop
16.	1234
17.	monkey
18.	shadow
19.	sunshine

20.	12345
21.	password1
22.	princess
23.	azerty
24.	trustno1
25.	000000

Tabla 2. Lista de las peores contraseñas 2013

2.1 Clasificación de las contraseñas gráficas.

Las contraseñas gráficas se pueden dividir en las siguientes dos categorías [3]:

1. Contraseñas gráficas basadas en el recuerdo

El usuario debe reproducir algo que ha creado o seleccionado durante la fase de registro.
Algunas de ellas son las siguientes:

- **Draw-a-Secret (DAS):** El usuario ha de realizar un simple gráfico en un grid 2D. Las coordenadas ocupadas por el dibujo son almacenadas en el orden que se ha dibujado.

- **Firma:** Similar al anterior, pero solicitando al usuario que realice su firma.

- **PassPoint:** Seleccionar una secuencia de puntos sobre una imagen, previamente establecidos.

2. Basadas en el reconocimiento

El usuario debe seleccionar dentro de una serie de imágenes por medio del reconocimiento aquellas que ha seleccionado previamente en la fase de registro.
Algunas de ellas son las siguientes:

- **Dhamija and Perrig:** Seleccionar imágenes preseleccionadas entre muchas opciones.

- **PassFace:** Similar al anterior pero las imágenes son de rostros humanos.

● **Sobrado and Birget:** El usuario ha de seleccionar un punto dentro del área delimitada por una serie de objetos preseleccionados, dentro de una imagen llena de diversos objetos.

2.2 Contraseñas gráficas en la actualidad.

Actualmente, a nivel gráfico se está utilizando a la hora de desbloquear dispositivos móviles, el conocido "**lock pattern**"..

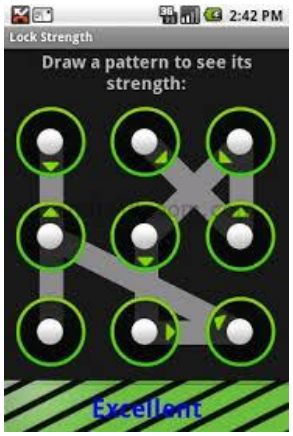


Ilustración 3. Lock Pattern 9 nodos

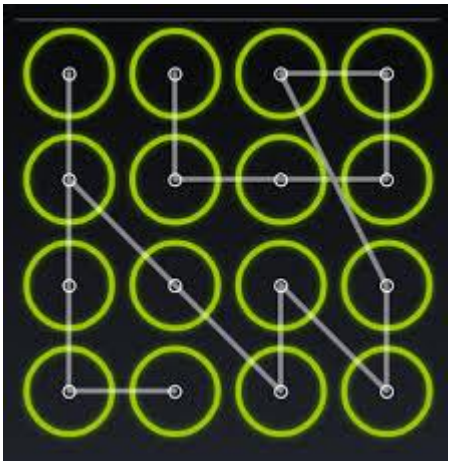


Ilustración 4. Lock Pattern 16 nodos

Es una manera muy fácil de recordar y muy rápida de introducir un pin y bastante efectiva y segura a **nivel de dispositivo personal,** siempre que limpiemos la pantalla para no dejar rastro de las trazas realizadas.

Pero cuando hablamos de autenticación frente a un sistema de acceso público, necesitamos generar contraseñas muy robustas y a medida que queramos hacerlo con un modelo similar a este, **incrementaremos notablemente la longitud del patrón, siendo imposible de recordar**.

Windows 8, propone un sistema para sus dispositivos móviles y portátiles basado en contraseñas gráficas el cual permite realizar tres gestos sobre una imagen.

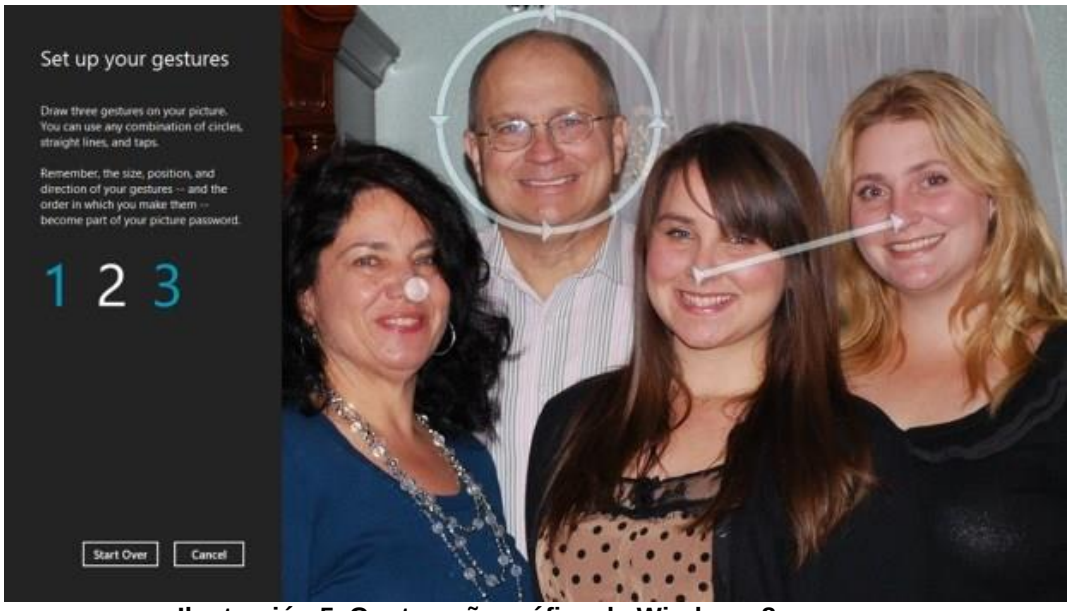


Ilustración 5. Contraseña gráfica de Windows 8

3. Análisis de requisitos y diseño del prototipo.

3.1 Requisitos del sistema

OpenGId es un prototipo de servidor de autenticación por tercera parte compatible con OpenId y que realiza el proceso de autenticación por medio de contraseñas gráficas.

3.1.1 Requisitos Funcionales

● OpenGId deberá permitir a un usuario loguearse en un sistema compatible con OpenId, por lo tanto deberá ofrecer un servicio que implemente el protocolo OpenId y que sea accesible desde la red en la que quiera ser utilizado (Internet, LAN, etc..).

● Para realizar la autenticación, OpenGId proporcionará una interface donde el usuario podrá introducir una contraseña de tipo gráfica. En este prototipo se utilizará el sistema PassPoint sobre una imagen predefinida por dicho usuario.
● El usuario podrá registrarse, cargar la imagen o escoger alguna de las disponibles y establecer los puntos que formarán la contraseña. Después de realizar el proceso de registro, OpenGId retornará al usuario un identificador que deberá utilizar cuando quiera autenticarse en algún sitio compatible con OpenId.

● El sistema deberá permitir al usuario consultar su historial de accesos, como también los intentos fallidos de uso de su contraseña.

● Se deberá permitir al usuario, cambiar su contraseña las veces que este lo considere necesario.

● Cada vez que se produzca un intento fallido de autenticación, el sistema enviará un email al usuario para informarle sobre lo ocurrido, recomendando al usuario cambiar su contraseña si los porcentajes de acierto han sido elevados.

3.1.2 Requisitos No Funcionales

Al ser un prototipo que será probado en la web, se ha decidido que el sistema se desarrollará usado las siguientes tecnologías:

Interfaces de Usuario : Todas las interfaces serán presentadas en un explorador de internet. Por lo tanto se utilizará HTML. Para ofrecer un aspecto agradable y compatible con todos los exploradores se utilizará el framework Tweeter Bootstrap.

Base de datos: MySQL.

Programación: PHP

La elección del lenguaje MySQL y PHP se debe a que es mucho más simple la instalación de paquetes abiertos como AppServer y XAMPP, donde ya vienen incluidos los siguientes componentes (Apache Server - MySQL - PHP - phpMyAdmin - FTP).

Las contraseñas serán almacenadas en la base de datos, después de aplicarle la función de resumen **sha256.** Las funciones de resumen tienen la característica de ser funciones unidireccionales, por lo tanto presentan una facilidad para pasar del texto claro al texto cifrado y una gran dificultad para pasar del texto cifrado al claro. Por lo tanto su uso es idóneo para el almacenamiento de contraseñas cifradas ya que no es necesario obtener el texto en claro. La validación de la contraseña se realiza mediante una comparación del texto cifrado generado por el resumen lo que garantiza su permanencia en secreto. El texto cifrado también tendrá la característica de tener siempre la misma cantidad de bits, en este caso 256, lo cual permite prever el espacio físico que utilizará su almacenamiento.
No obstante, el uso de la función de resumen genera un problema a la hora de trabajar con contraseñas gráficas las cuales requieren un cierto grado de tolerancia. Lo ideal sería trabajar con cifrados homomórficos, los cuales permitirían realizar alguna operación en el texto cifrado como la de obtener una distancia entre puntos, pero su complejidad excede el alcance de este trabajo.

Las soluciones propuestas a esta problemática son:

Particionar la imagen en pequeñas áreas y generar varias contraseñas según la proximidad de los puntos seleccionados a la fronteras de sus correspondientes áreas.
Cifrar la contraseña mediante un criptosistema simétrico y para luego descifrarla y realizar operaciones sobre la misma para calcular la desviación respecto a la introducida y evaluarla.

Ambas soluciones se explican detalladamente en el capítulo de implementación.

3.2 Casos de uso

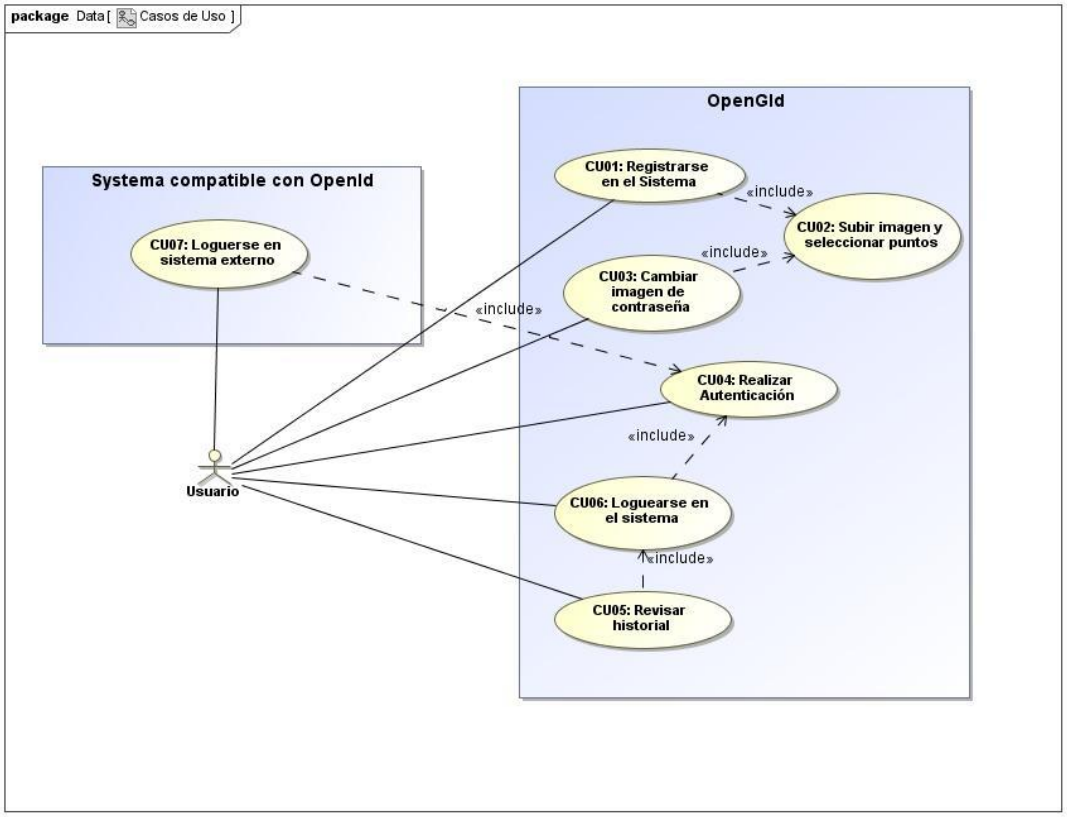


Ilustración 6. Diagrama de Casos de Uso

CU01	**Registrarse en el sistema**
Actores	Usuario
Descripción	Para poder hacer uso del sistema de autenticación, el usuario deberá registrarse en el sistema, cumplimentando un formulario de registro.
Interacción	1. Rellenar el formulario con los datos solicitados y pulsar en continuar. 2. Seleccionar una imagen de las recomendadas o subir una propia. 3. Seleccionar puntos de contraseña(CU02).
Precondición	Acceder al portal del sistema y seleccionar el link de registro.

Postcondición	El usuario ha quedado registrado en el sistema.

CU02	Seleccionar puntos en una imagen
Actores	Usuario
Descripción	Para introducir una contraseña tanto para registrarla como para autenticarse, el usuario ha de seleccionar una secuencia de puntos sobre una imagen previamente seleccionada.
Interacción	1. El sistema mostrará la imagen en pantalla. 2. El usuario introducirá una secuencia de puntos con el dispositivo señalador. Si la secuencia es la deseada se presiona continuar y el caso de uso acaba. Si la secuencia es incorrecta, se presiona reset y el caso de uso se reinicia.
Precondición	Estar registrado en el sistema, o realizando el proceso de registro..
Postcondición	El usuario es autenticado, o la contraseña queda guardada.

CU03	Cambiar imagen de contraseña
Actores	Usuario
Descripción	El usuario podrá cambiar la contraseña las veces que lo considere conveniente.
Interacción	1. Loguearse en sistema. 2. En el menú de opciones seleccionar cambiar contraseña. 3. Subir una imagen o seleccionar las recomendadas. 4. Seleccionar puntos de contraseña (CU02).
Precondición	Loguearse en el sistema. (CU06)
Postcondición	La contraseña ha sido cambiada.

CU04	Realizar Autenticación.
Actores	Usuario
Descripción	El usuario deberá demostrar al sistema que conoce cuales son los puntos escogidos cuando se ha registrado. (CU02)
Interacción	1. Seleccionar su imagen entre un grupo de imágenes mostradas. 2. Sobre la imagen seleccionada, seleccionar los puntos de

	contraseña y pulsar en continuar.. 3. Si la contraseña es correcta el usuario es autenticado y el caso de uso acaba. Si la contraseña es incorrecta, se va al punto 2.
Precondición	Introducir su identificador en un portal web OpenId compatible, o en el servidor OpenGid.
Postcondición	El usuario está autenticado. Se registra el resultado en la BD

CU05	Revisar Historial
Actores	Usuario
Descripción	El usuario podrá ver un historial de los autenticaciones exitosas y fallidas.
Interacción	1. Loguearse en sistema. 2. En el menú de opciones seleccionar ver historial. 3. El sistema mostrará una tabla con las autenticaciones realizadas y su resultado, mostrando fecha, hora y el portal que se ha solicitado la autenticación.
Precondición	Estar logueado en el sistema. (CU06)
Postcondición	El usuario obtiene la información solicitada.

CU06	Loguerse en el sistema.
Actores	Usuario
Descripción	El usuario podrá acceder a su cuenta personal, para poder realizar gestiones, como ver historial, modificar datos, o dar de baja su cuenta.
Interacción	1. Acceder al portal de OpenGId 2. Seleccionar Login. 3. Realizar el proceso de autenticación (CU04)
Precondición	Estar registrado en el sistema.
Postcondición	El usuario accede a su cuenta.

CU07	Loguearse en un sistema externo.

Actores	Usuario
Descripción	El usuario se podrá loguear en cualquier sistema compatible con OpenId.
Interacción	1. Introducir el identificador suministrado por OpenGId. 2. El usuario será redireccionado al servidor OpenGid, donde deberá realizar la autenticación. (CU04). 3. Si la contraseña es correcta, el usuario es autenticado y vuelve a ser redireccionado al servidor de origen.
Precondición	Estar registrado en el sistema.
Postcondición	El usuario es autenticado.

3.3 Estructura de la información

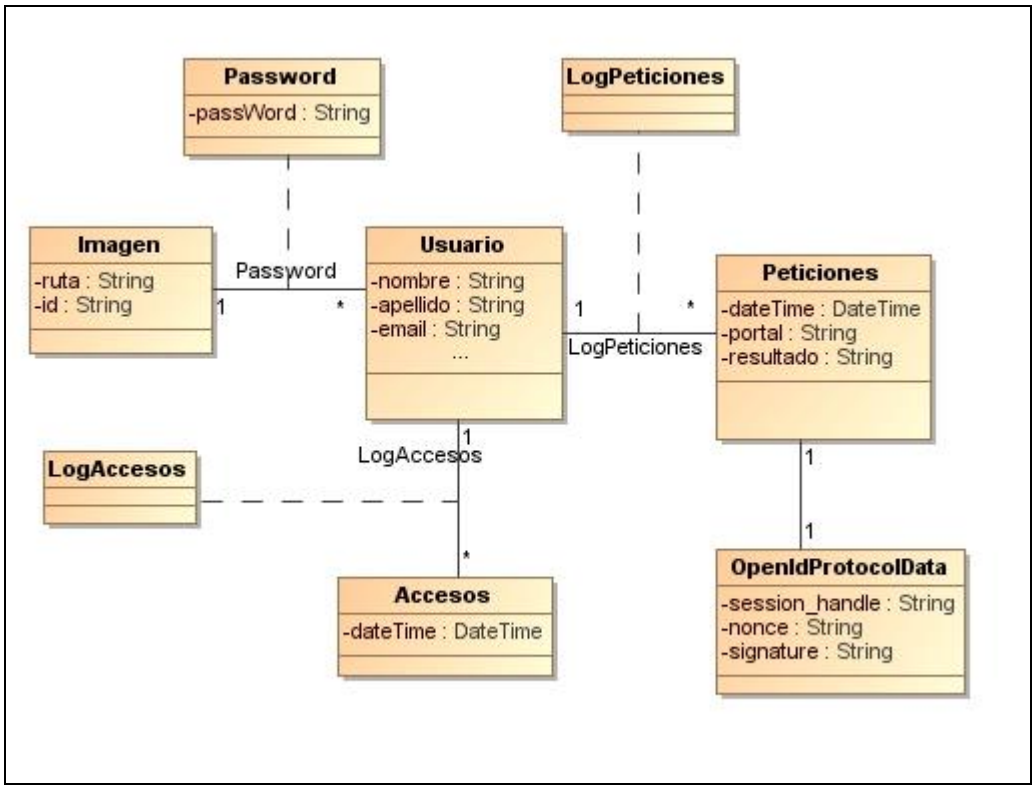


Ilustración 7. Diagrama de Clases UML

La clase “peticiones” contiene información requerida para el uso del protocolo OpenId. Cada interacción con servidores externos mediante OpenId, quedara almacenada en “LogPeticiones”.

La clase LogAccesos contiene información de los accesos realizados por el usuario al sistema, ya sea de manera local o mediante el proceso de autenticación externo.

La clase Imagen, guarda información de la ruta de la imagen y un identificador. En una futura implementación, el usuario podría disponer de varios pares imagen-contraseña para activarlas cuando lo considere necesario.

3.4 Diseño de la Base de Datos

A partir del diagrama de estructuras de datos, se realiza la transformación al modelo entidad relacional.

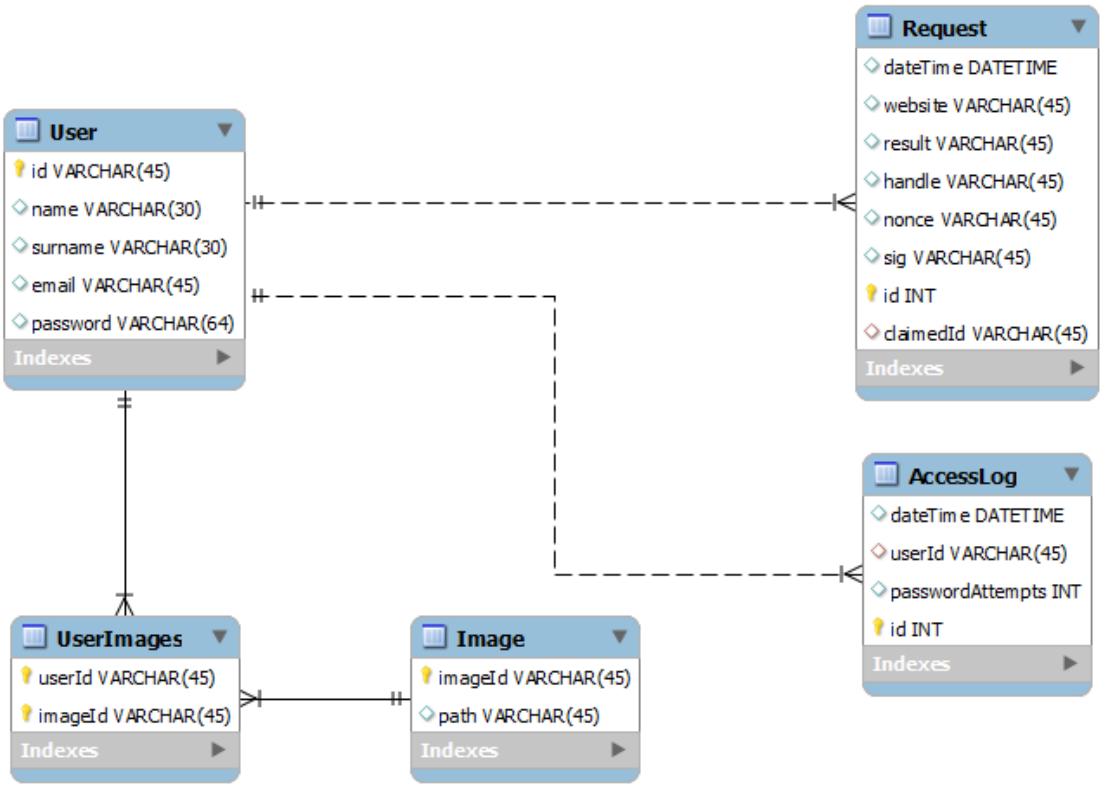


Ilustración 8. Diagrama ER de la base de datos.

3.5 Diseño de las interfaces gráficas

Creación de la contraseña:

La creación de la contraseña podrá realizarse sobre una imagen otorgada por el usuario o alguna escogida de las existentes en el banco de imágenes.

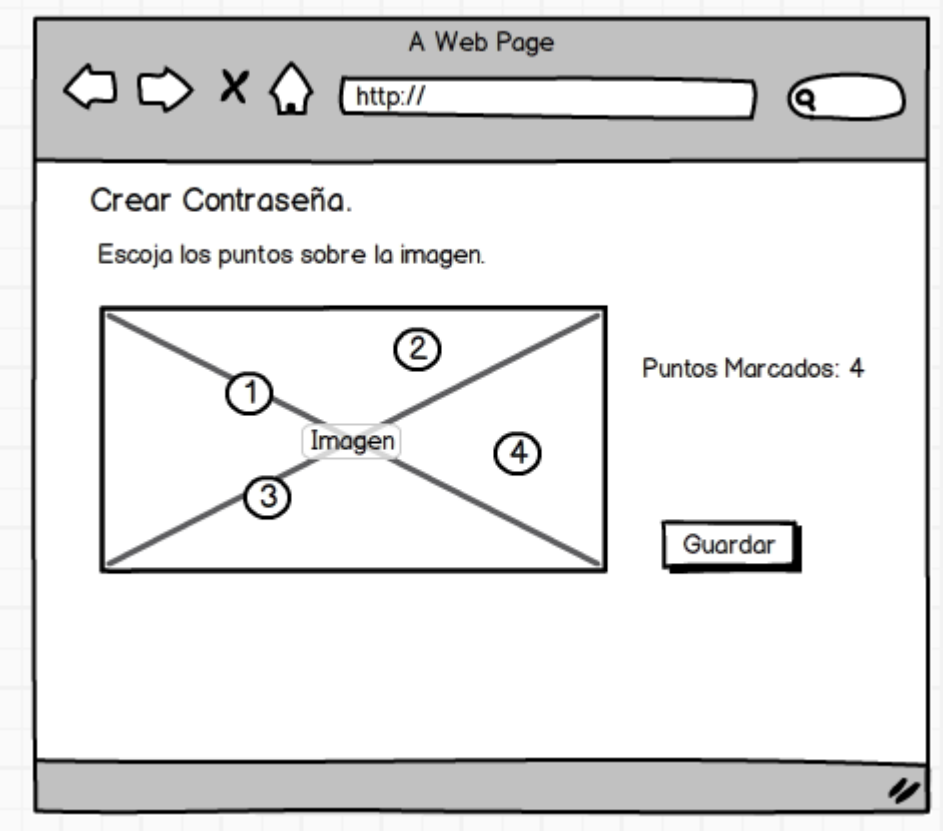


Ilustración 9. Prototipo de interface de creación de contraseña

Fase de autenticación:

- **Primer paso:**

Seleccionar su imagen entre una serie de imágenes presentadas

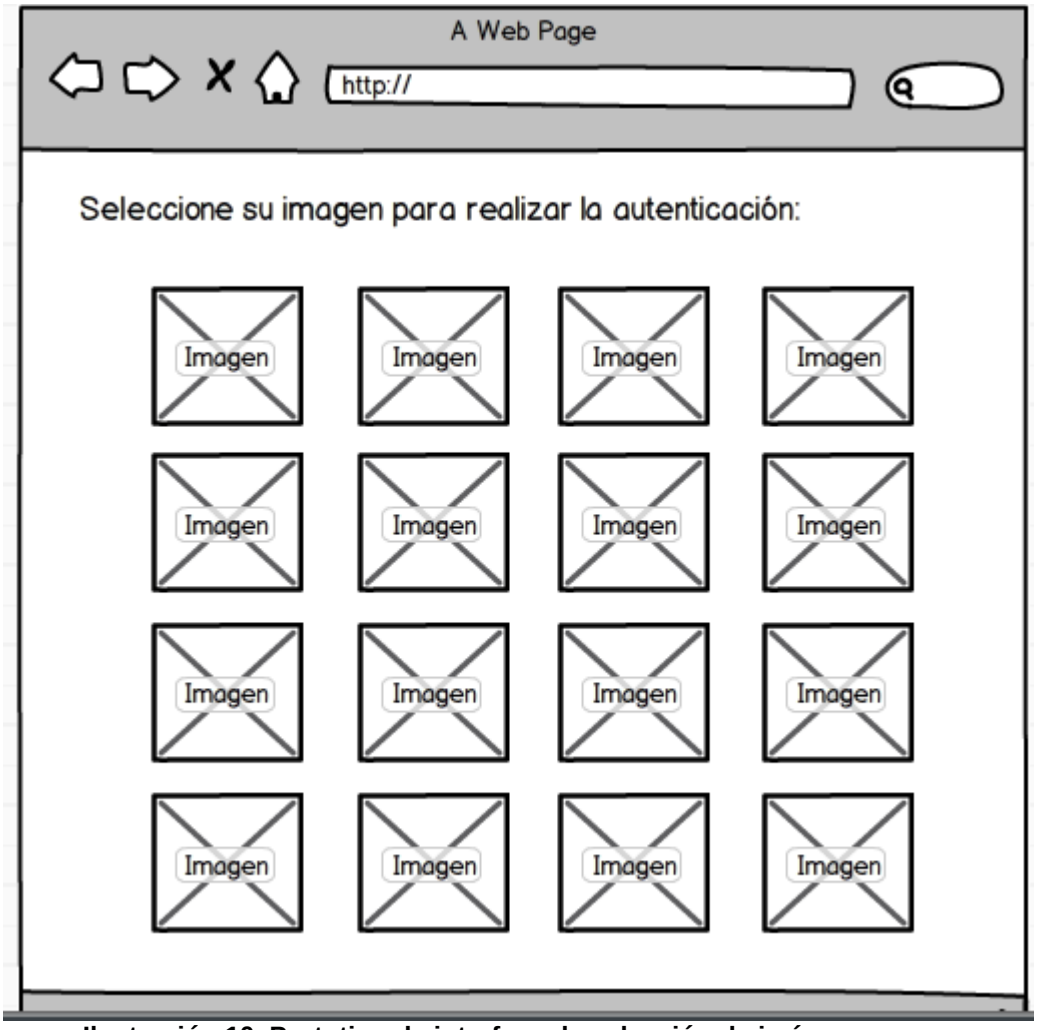


Ilustración 10. Prototipo de interface de selección de imágen.

- **Segundo Paso:**

  Seleccionar los puntos sobre la imagen seleccionada en el paso anterior.

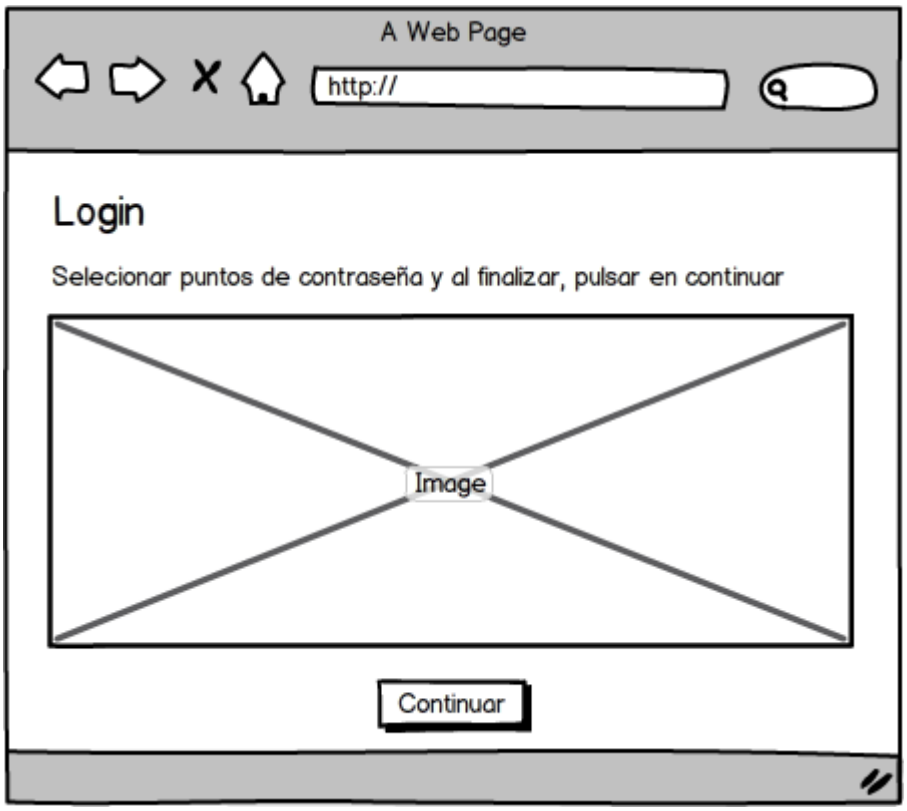


Ilustración 11. Prototipo de interface de selección de puntos.

4. Implementación del trabajo

4.1 Diagrama general de funcionamiento.

El siguiente diagrama muestra a gran escala la secuencia de funcionamiento del servidor.

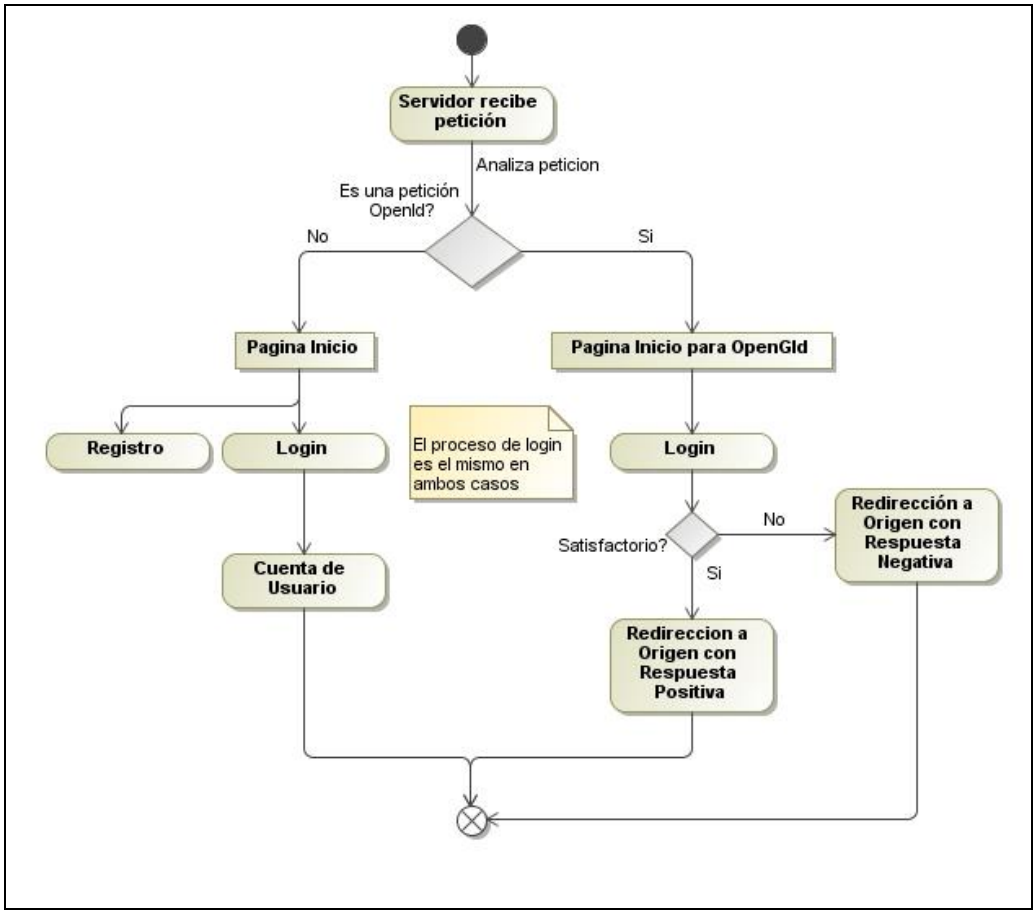


Ilustración 12. Diagrama general de funcionamiento.

Las peticiones son recibidas sobre la dirección *nombre_servidor/*opengid.php. El fichero opengid.php contiene el código que procesara las peticiones sobre la dirección mencionada y determinara si dichas peticiones corresponden al protocolo OpenId(en el caso de estar produciendo una autenticación) o bien si un usuario quiere acceder a su cuenta personal en el servidor.

A continuación se muestra el diagrama completo del flujo web en el servidor.

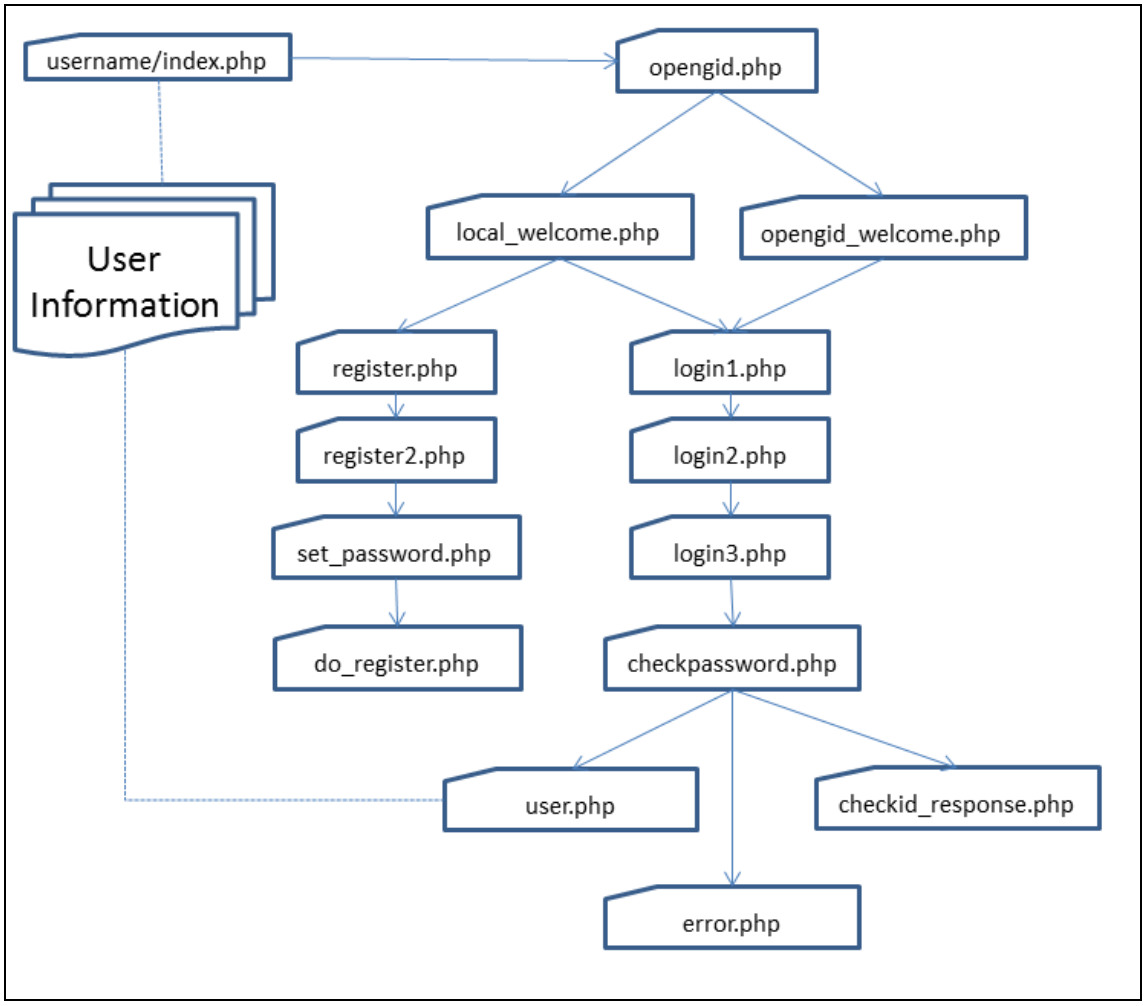


Ilustración 13. WebFlow del prototipo

4.2 Implementación de PassPoint.

Tal como se ha comentado en la introducción, existe una problemática al trabajar con este tipo de contraseña relacionada con la tolerancia. La probabilidad de acertar una contraseña de esta naturaleza puede ser tan pequeña que generaría un grave inconveniente al usuario, ya que dependiendo del dispositivo de entrada, podría ser incapaz de autenticarse, además de incrementar notablemente el tiempo de introducción de la contraseña por la precisión requerida.

Por lo anterior se proponen dos soluciones, la de dividir la imagen en áreas más grandes para crear esta tolerancia o la de realizar una comparación evaluando la proximidad de la contraseña guardada con la introducida.

Ambas soluciones tienen ventajas e inconvenientes. La primera tiene el inconveniente de lidiar con los puntos frontera, como se explicara a continuación, pero la ventaja de comparar contraseñas encriptadas por lo tanto de manera segura. La segunda, la comparación es mucho más sencilla pero la seguridad se ve reducida, al existir una bidireccionalidad en el cifrado de las contraseñas. Es decir, si un usuario malintencionado pudiera hacerse con la llave para desencriptar, podrá hacerse con todas las contraseñas almacenadas. Por este motivo, el presente prototipo utiliza el particionado de imagen.

4.2.1 Mediante particionado de la imagen.

La imagen se divide en áreas de un determinado tamaño según la tolerancia que se quiera dar a la contraseña. Cuanto más grande, mayor tolerancia, pero mayor reducción del espacio de contraseñas, por lo tanto más inseguras. Empíricamente, se ha comprobado que áreas de 20x20 píxeles para una imagen de 800x600 o 480x640 son un buen valor. A continuación se muestra un ejemplo de particionado de una imagen de 80x80 píxeles.

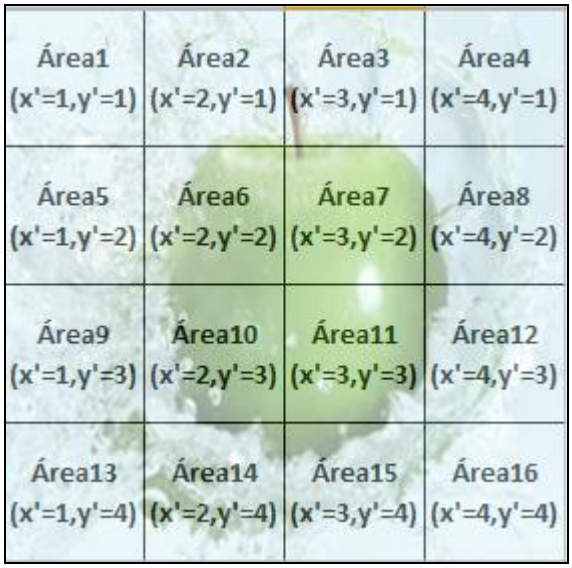


Ilustración 14. Ejemplo particionado de imágen de 80x80 píxeles.

La contraseña es una cadena de caracteres compuesta con el nombre de la imagen sucedida por las áreas seleccionadas después de particionar la imagen.

ej: img001,x1'.y1',x2',y2',x3'.y3'

Donde x'.y' no son las coordenadas reales del punto seleccionado, sino las coordenadas del área de 20x20 pixeles en donde se ha realizado el click.

Problema de las fronteras.

Supongamos que uno de los puntos escogidos se encuentra muy cerca de la frontera del área de particionado. La tolerancia en estos puntos se reduce. Por lo tanto, una contraseña con puntos próximos a la frontera sería difícil de acertar. Para incrementar la tolerancia en las fronteras, a la hora de introducir la contraseña, se agregarán hasta un máximo de 3 áreas por punto seleccionado según el criterio que se describe a continuación. Supongamos que se realiza un click dentro del Area10 representada en la siguiente figura con los correspondientes 20x20 píxeles:

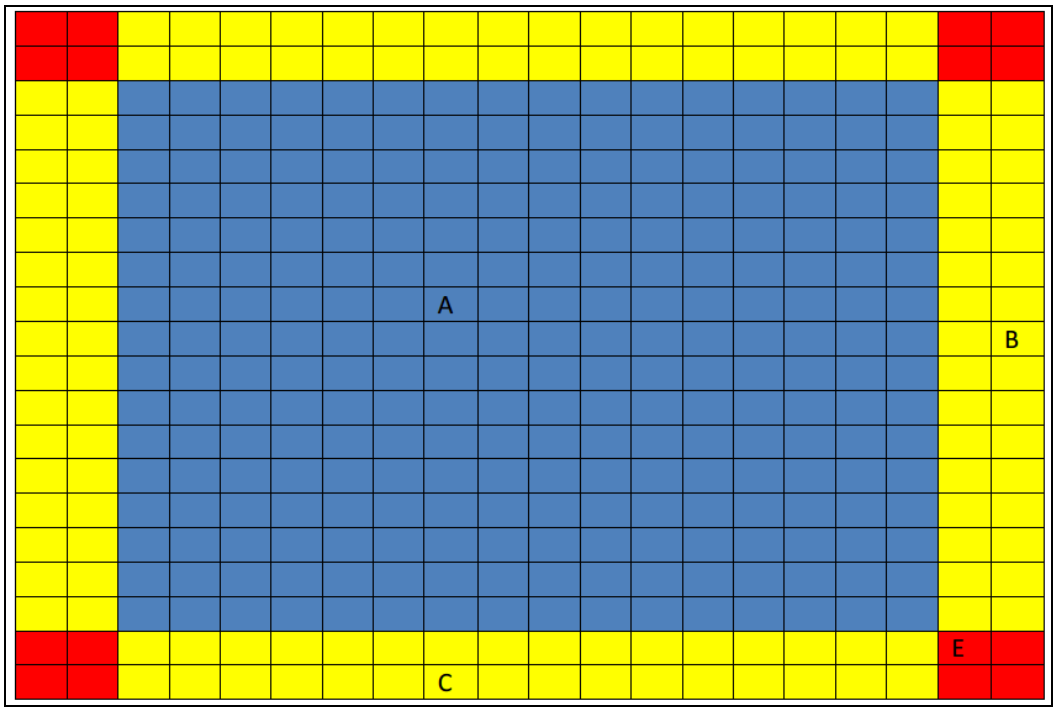


Ilustración 15. Representación de un área de partición de imágen.

- Para los puntos dentro de la zona azul como por ejemplo el punto A mostrado en la ilustración 15, el área final será el Área10.

- Para los puntos dentro de la zona amarilla, será Área10 y la adyacente. En el caso del punto B serán Área10 y Área11. En el caso del punto C serán Área10 y Área14.
- Para los puntos dentro de la zona roja, resultará en el punto Área10 más 3 áreas adicionales que serán las dos adyacentes más la diagonal. En el caso del punto E serán Área10, Área11, Área14 y Área15

A continuación, se generará una serie de contraseñas teniendo en cuenta las áreas extras añadidas para poder contemplar la tolerancia. La verificación se produce mediante una comparación encriptada de cada una de las citadas contraseñas generadas frente a la almacenada. Por ejemplo, supongamos que se introduce una contraseña de 3 puntos:

p1 (pertenece al Area2)
p2 (pertenece al Area5, y por tolerancia al área 6, 9 y 10)
p3 (pertenece al Area1)

Y el nombre de la imagen es img001, las contraseñas generadas serán:

Img001,A2,A5,A1 = img001,2.1,1.2,1.1
Img001,A2,A6,A1 = img001,2.1,2.2,1.1
Img001,A2,A9,A1 = img001,2.1,1.3,1.1
Img001,A2,A10,A1 = img001,2.1,2.3,1.1

Estas contraseñas se encriptarán y se compararán encriptadas con la contraseña almacenada.

4.2.2 Mediante evaluación de proximidad.

Este es un método mucho más simple de implementar, donde la contraseña almacenada es el nombre de la imagen concatenado con las coordenadas reales de los puntos seleccionados en la imagen y no de un área como en el caso anterior.

ej: img001,x1.y1,x2,y2,x3.y3

La verificación se realiza en texto en claro, por lo que se puede realizar una comparación del nombre de la imagen junto con una comparación punto a punto de la contraseña introducida con la almacenada, de la siguiente manera:

¿ Nombre imagen seleccionada = Nombre imagen almacenado ?

Para cada punto:

¿Distancia(punto introducido, punto almacenado) < Tolerancia ?

Si se cumplen las anteriores condiciones, se puede dar al usuario por autenticado.

4.3 Breve descripción del código implementado

Tal como se ha comentado en la sección de análisis de requisitos, la programación del servidor se ha desarrollado en PHP, por su facilidad de instalación y de interacción con la base de datos, como la de gestionar la información generada durante una sesión de usuario en el servidor.
A continuación se describe la funcionalidad de los ficheros más relevantes que dan la funcionalidad al servidor.

- opengid.php

Es la página principal del servidor. Es la encargada de detectar si la petición a esta página la realiza el propio usuario para acceder a su cuenta o un servidor que está solicitando la autenticación de un usuario mediante OpenId. Para ello, se analiza la cabecera de la petición en busca del valor **openid_mode.** El citado valor indicará en qué modo se ha de ejecutar el protocolo, por lo tanto se redirige el flujo de ejecución mediante condicionales tal como se muestra en el anterior extracto de código.

- local_welcome.php y opengid_welcome.php

Estos ficheros muestran la pantalla de bienvenida, según se acceda de manera local o redireccionada mediante OpenId. Opengid_welcome.php muestra información adicional sobre el lugar de origen desde donde se produjo el redireccionamiento, con el objetivo de que el usuario pueda comprobar que realmente el proceso de autenticación que va a realizar es el correcto. Esta comprobación es una medida de seguridad que permite al usuario abortar el proceso de autenticación si detectara alguna anomalía.

- register.php y register2.php

Estos ficheros generan un formulario estándar y muestran las imágenes por defecto y la posibilidad de agregar imágenes propias.

- set_password.php

Este fichero se encarga de recoger la contraseña introducida por el usuario, y convertirla al formato establecido para su almacenamiento.

- login1.php y login2.php

Estos ficheros recogen la imagen seleccionada por el usuario y los puntos seleccionados sobre dicha imagen en el proceso de login. La información recogida pasará a ser procesada en por login3.php

- login3.php

Aquí es donde se procesan los puntos seleccionados por el usuario y se convierten a áreas, donde según la posición del punto le corresponderán una o más áreas según se ha explicado en la sección de particionado de la imagen.

- check_password.php

Este fichero se encarga de comparar todas las contraseñas generadas por login3.php con la contraseña almacenada. En caso de coincidencia, si el acceso es local se redireccionará a user.php, de lo contrario, se generará una respuesta para ser enviada al servidor de origen con el resultado de la comprobación. Dicha respuesta se genera en checkid_response.php.

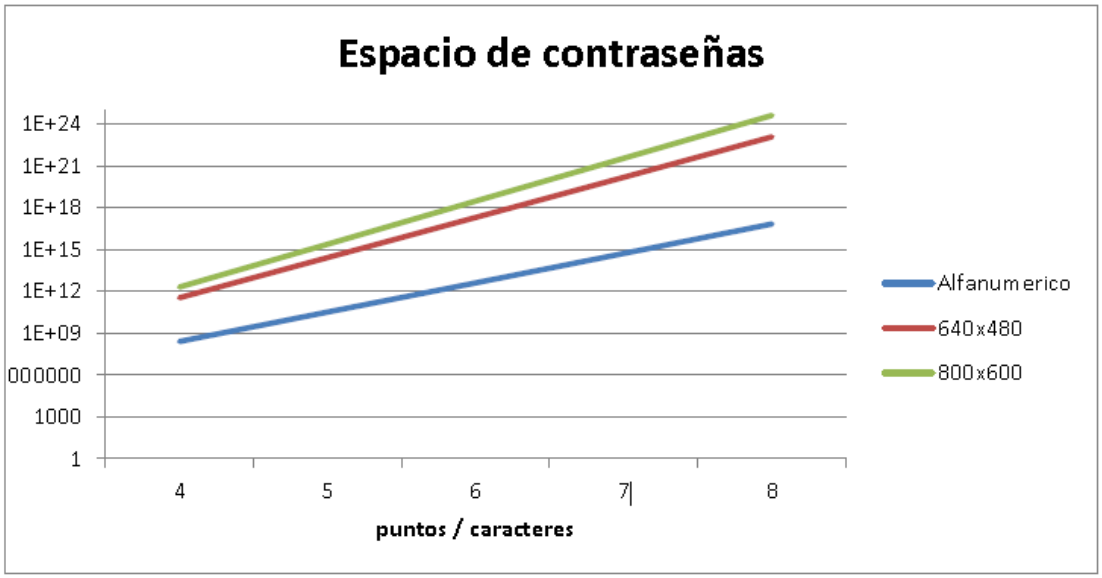
- checkid_response.php

Aquí se genera la respuesta que será retornada al servidor de origen tal como lo indica el protocolo OpenId explicado en el anexo.

5. Debilidades y fortalezas del método PassPoint.

5.1 Fortalezas

- **Robusteza:** Una de las fortalezas que tienen el método escogido, es la de poseer un mayor espacio de contraseñas. La cantidad de contraseñas posibles en notablemente mayor que aquellas generadas alfanuméricamente. Por lo tanto la dificultad de encontrar una contraseña desconocida por fuerza bruta es mayor.



- **Memorización**: El hecho de escoger puntos siguiendo algún patrón, o puntos que hayan llamado la atención o que tengan algún significado personal, los hacen fácilmente memorizables. No obstante, a mayor cantidad de puntos escogidos, se incrementa la dificultad de memorizarla.

5.2 Debilidades.

- **Reducción del espacio de contraseñas:** El hecho de escoger puntos en una imagen para luego recordarlos, hacen que los mismos tengan unas determinadas características, que los hace resaltar sobre los demás. Es decir, estadísticamente, los puntos escogidos son bordes, intersecciones, o con un alto contraste. Por lo tanto es

posible descartar puntos poco susceptibles de ser escogidos aplicando a la imagen algún tipo de procesamiento.

A continuación se muestra un ejemplo de cómo aplicando el filtro Sobel sobre una imagen se pueden eliminar puntos que se encuentran en zonas uniformes.

Ilustración 16. Imagen sin procesar.

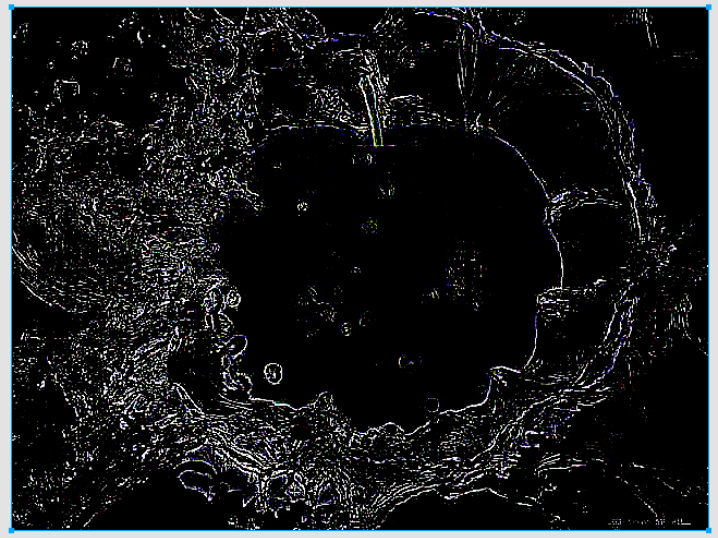


Ilustración 17. Imagen después de aplicarle el filtro de Sobel.

- **Shoulder Surfing:** Este tipo de contraseña gráfica no es resistente al ataque por observación. Un atacante estratégicamente situado, podría observar de manera visual los puntos introducidos por el usuario si no se toman las precauciones adecuadas. Este ataque podría resultar más complicado de llevarse a cabo si el usuario

accede desde un dispositivo móvil como teléfono o tableta debido al pequeño tamaño de la pantalla.

6. Evaluación de usabilidad

Para evaluar la usabilidad, se han considerado los siguientes parámetros, sobre una contraseña de 5 puntos en una muestra de 10 usuarios.

- Tiempo medio de introducción de la contraseña.

  La evaluación muestra que el tiempo medio de introducción de una contraseña de 5 puntos es de 13,59 segundos. Lo que equivale a casi 3 segundos por punto. Este resultado muestra una clara diferencia con una contraseña alfanumérica, donde la media por carácter es de menos de medio segundo en personas que utilizan el teclado frecuentemente.

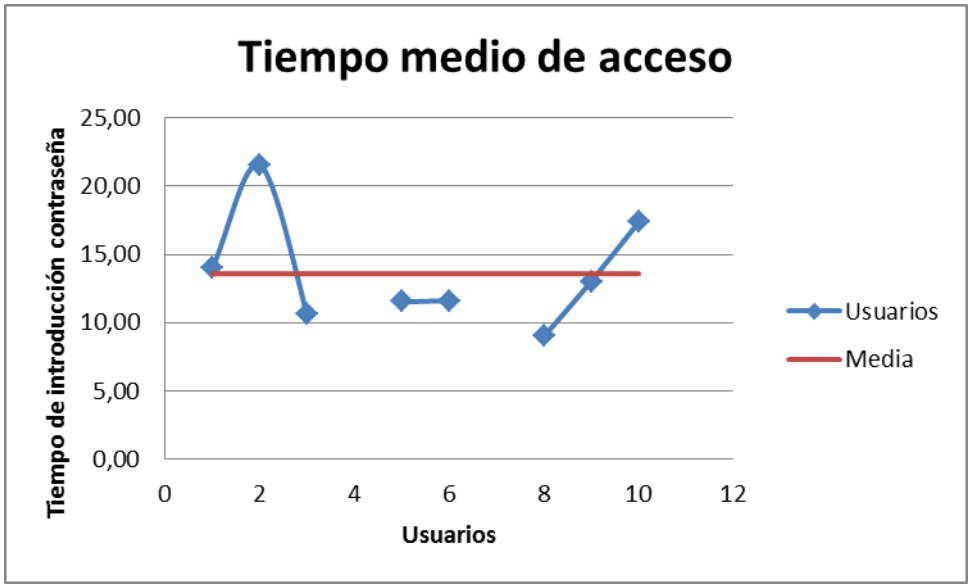


Ilustración 18. Gráfica de tiempo medio de acceso

- Tiempo medio de autenticación, el cual consiste en el redireccionamiento al servidor de autenticación, proceso de autenticación y retorno al origen.

  La evaluación muestra que este proceso se lleva a cabo en una media de 21,56 segundos. Este tiempo podría resultar excesivo si el proceso debiera realizarse repetidas veces.

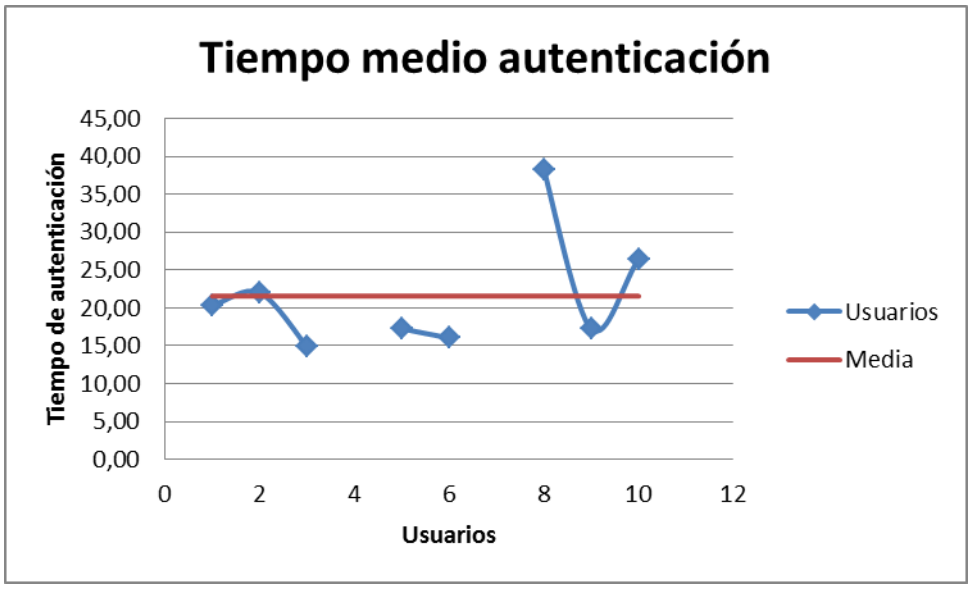


Ilustración 19. Gráfica de tiempo medio de autenticación.

- Tasa de aciertos.

  La media de aciertos ha resultado ser 63,16%. Cabe destacar en esta evaluación que los aciertos han ido disminuyendo a medida que la pantalla del dispositivo era más pequeña. Lo que indica que será necesario revisar el prototipo para adaptarlo a dispositivos móviles y mejorar la interacción.

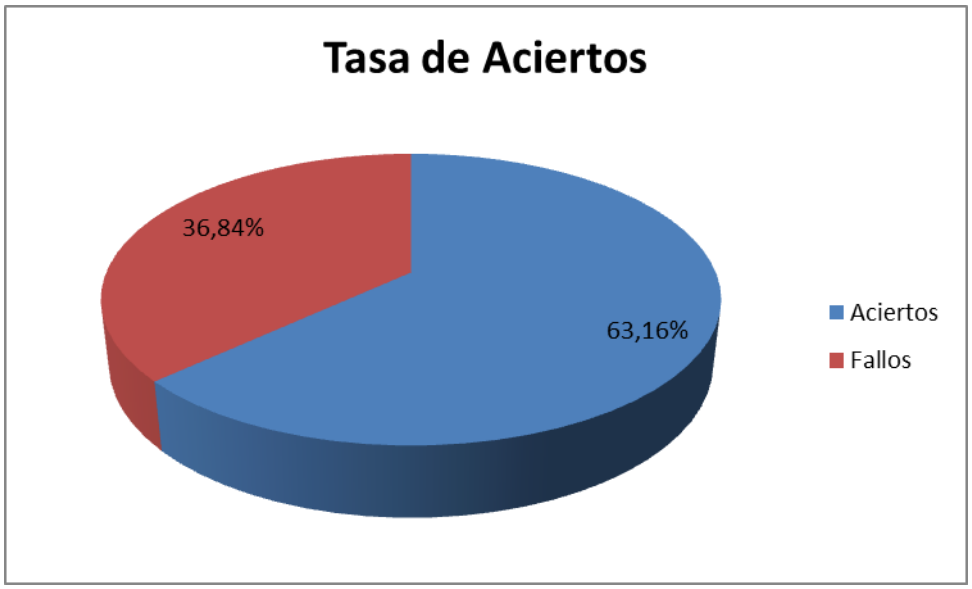


Ilustración 20. Gráfica de tasa de aciertos.

- Valoración del usuario sobre el proceso.

  Los usuarios que participaron en el estudio han valorado el sistema de manera positiva. Consideran que utilizando una imagen propia, la

memorización de la contraseña sea mayor y el la autorización centralizada la han valorado mejor que la distribuida.

7. Evaluación de viabilidad

El estudio de viabilidad se realizará desde un punto de vista de la aceptación del usuario como también de tipo económico.

7.1 Estudio de aceptación.

Para llevar a cabo dicho estudio se han formulado las siguientes preguntas a un grupo de diez personas que utilizan frecuentemente los servicios ofrecidos en Internet.

a) ¿Prefiere usted centralizar su proceso de autenticación a tener que autenticarse frente a cada servicio que utilice?

Existe desconfianza unánime del sistema presentado si la autenticación se realizara frente a entidades del tipo financieras o administrativas, lo cual es totalmente razonable. También es muy poco probable o imposible que dichas entidades delegaran el proceso de autenticación. También se ha recomendado mejorar la interface para tener una interacción más natural y veloz.

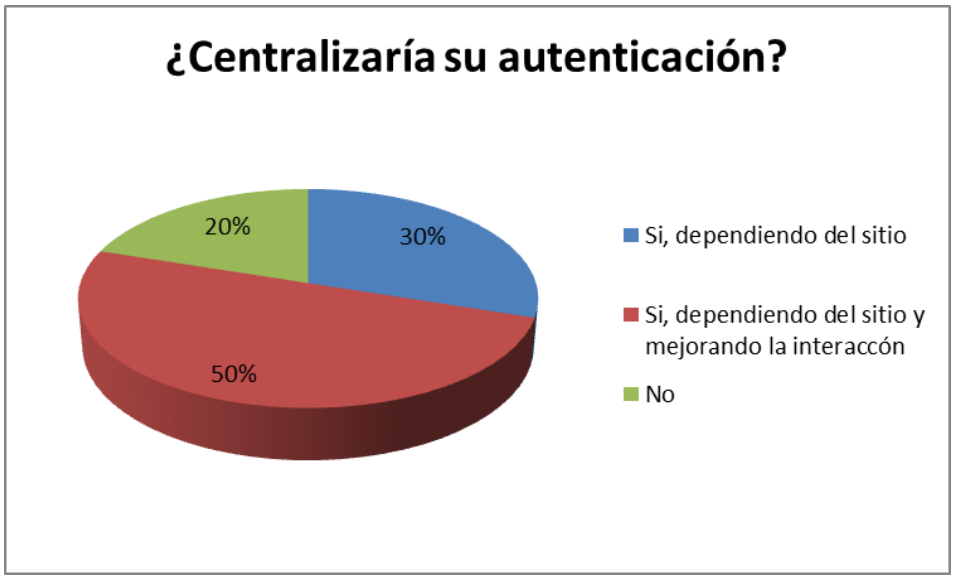


Ilustración 21. Gráfico estadístico pregunta A

b) ¿Considera más seguro tener sus datos personales y contraseña en un solo sitio de confianza a tenerlos diseminados por todos los servicios que utiliza?

En general se considera que la centralización de los datos aumenta la seguridad de los mismos. En mi opinión personal, no es el hecho de centralización si no que la probabilidad de que un conjunto de

servidores independientes implementen las medidas de seguridad correctamente es menor o igual que lo haga un único servidor.

Ilustración 22. Gráfico estadístico pregunta B

c) ¿Considera un aumento del tiempo consumido en el proceso de autenticación una incomodidad?

Aquellos usuarios que realizan muchos accesos autenticaciones consideran que un aumento del tiempo puede resultar tedioso, pero consideran que si se mejorara la interacción, posiblemente no resultara un problema.
Se debería estudiar la posibilidad de evitar repetir el proceso de autenticación.

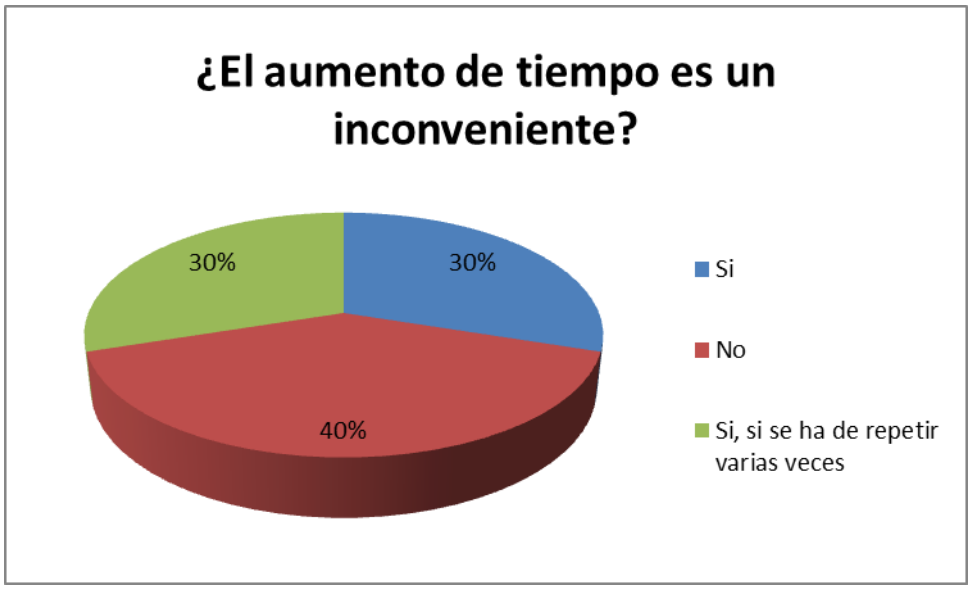


Ilustración 23. Gráfico estadístico pregunta C

d) ¿Considera una falta de privacidad que el servidor de autenticación centralizado conozca su historial de accesos?

Un importante número de usuarios considera que su privacidad ya se encuentra comprometida. Entienden que sus movimientos por Internet ya quedan registrados de manera transversal por medio de las llamadas cookies de terceros existentes en los sitios más concurridos.

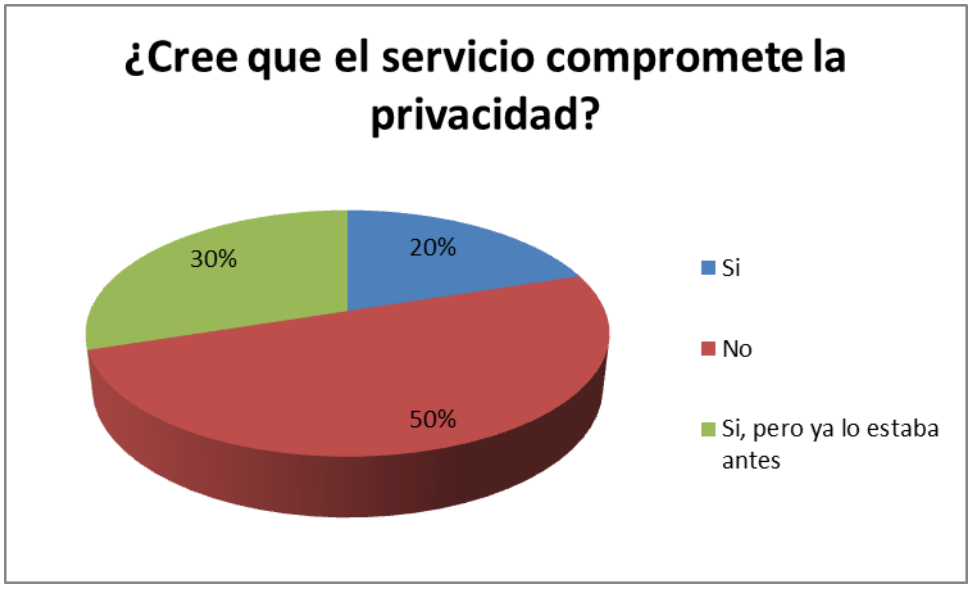


Ilustración 24. Gráfico estadístico pregunta D

e) ¿Pagaría por utilizar un servicio de autenticación externa?

Ninguno de los usuarios está dispuesto a pagar por este servicio. Entienden que quien debe asumir el gasto son los portales que requieren la autenticación.

Ilustración 25. Gráfico estadístico pregunta E

7.2 Estudio económico.

El producto final es un producto software, la viabilidad económica de su desarrollo dependerá de si existe un comprador que esté interesado en comprar y explotar el producto después de evaluar el prototipo. El producto final incluiría distintos métodos de autenticación gráfica para que el usuario pudiera elegir cual le es más conveniente.
El coste del producto se calcula basándose en las estimaciones detalladas en la siguiente tabla, según la cantidad de horas de trabajo consideradas.

Descripción	***Cantidad horas***	***Precio hora***	***Total***
Análisis de requisitos	10	20€	200€
Diseño	10	20€	200€
Prototipo funcional	50	20€	1000€
Producto Final	150	20€	3000€
Test de producto	20	20€	400€
		TOTAL	**4800€**

Tabla 3. Estimación del coste del proyecto.

Resultará interesante, para la empresa explotadora del producto, conocer la siguiente información para evaluar su viabilidad.

Para la utilización de OpenGid se necesitará:

- Una licencia que permita ofrecer el servicio o directamente comprar el desarrollo del producto.
- Un servidor con base de datos disponible las 24 horas, cuya capacidad variará dependiendo de la cantidad de usuarios registrados que tenga.
- Contrato de mantenimiento para corregir o actualizar el producto.

El uso del producto a nivel privado, como por ejemplo el acceso a distintos servicios dentro de una empresa, universidad u organización, el coste es muy reducido, ya que el sistema incluso podría ejecutarse en un servidor propio. No obstante, hoy en día existen servicios de hosting muy económicos incluso por menos de 10€ mensuales. Por lo tanto se considera que el producto es viable en este escenario.

Si se tratase de un uso público, con un uso masivo, entonces será necesario disponer de más servidores e implementar el sistema sobre una plataforma escalable. Esto es mucho más costoso y los usuarios no estarán dispuestos a pagar por utilizar este servicio según lo indican los resultados del estudio de aceptación. Por lo tanto en este caso, la viabilidad dependerá de la existencia de alguna empresa patrocinadora que quiera costear los gastos a cambio de la presencia de algún mensaje publicitario en el proceso de autenticación. Lamentablemente, hoy en día son muy pocos los sitios que están aceptando autenticación OpenId de nuevos proveedores, por lo tanto, no existe una motivación a usar este servicio públicamente de manera masiva. Consecuentemente, la viabilidad del producto es desfavorable en este escenario, y deberá ser evaluada nuevamente cuando la comunidad de Internet acepte la inclusión de nuevos proveedores de OpenId o algún otro sistema de autenticación abierto.

8. Conclusiones

Durante el transcurso del trabajo, se ha profundizado en la problemática existente en el uso de las contraseñas alfanuméricas. Se ha concluido que actualmente, la seguridad se ve afectada por la dificultad que encuentra una gran parte de los usuarios a la hora de recordar un determinado número de contraseñas largas y difíciles de adivinar por ataques de diccionario.
Las soluciones propuestas han sido recibidas con buena aceptación por parte de los usuarios que han puesto a prueba el sistema. De la realización de los ensayos se ha concluido lo siguiente:

- La autenticación por tercera parte es una solución viable y con aceptación para centralizar la autenticación en sitios donde la seguridad **no es crítica.** Es decir blogs, redes sociales, foros. Las encuestas no encuentran viable esta autenticación para ser utilizada en bancos, trámites administrativos, compras on-line, etc.
- El método de contraseña gráfica presentado ha tenido buena aceptación pero debería mejorarse la interface, para obtener una mejor interacción, con especial énfasis en los dispositivos móviles y de esta manera reducir el tiempo del proceso de autenticación.

Los objetivos del trabajo, que consistían en realizar un análisis de viabilidad y aceptación del esquema propuesto, evaluar las ventajas e inconvenientes de utilizar contraseñas gráficas como también de realizar una autenticación mediante una tercera parte se han cumplido.

Respecto a la planificación, se ha respetado durante todo el desarrollo del trabajo. No se ha aplazado ninguna de las fechas límites establecidas, intentando ir siempre un paso por delante para disponer de tiempo de maniobra frente a cualquier situación que no haya podido ser prevista.

En relación a la idea inicial, el diseño del sistema se ha ido modificando para poder adaptarlo a los recursos y conocimientos disponibles. Un ejemplo de esto ha sido el no poder realizar operaciones sobre información cifrada por la dificultad que supone la utilización de criptosistemas homomórficos. Considero este tipo de criptosistemas un área muy importante a investigar y desarrollar, dada la tendencia actual a utilizar servicios en la nube como por ejemplo Google Drive, donde la edición de documentos se realiza de manera remota, y no tenemos ninguna garantía de privacidad.

Otra de las situaciones detectadas durante la evaluación, es que debería considerarse implementar una autenticación persistente, para que el usuario no tenga que repetir el proceso de manera repetitiva cada vez que desee entrar a algún sitio. Considero que esta problemática debería estudiarse y resolverse en el producto final.

9. Glosario

10. Bibliografía

[1] Coulouris;Dollimore;Kindberg;Blair. Distributed Systems. Concepts and Design 5th Edition. Pearson 2012.

[2] "123456" lidera la lista de las peores contraseñas de internet de 2013.(2014). Consultado abril 2014. http://www.elpais.com.uy/vida-actual/lidera-lista-peores-contrasenas-internet.html

[3] Xiaoyuan Suo Ying Zhu G. Scott. Owen. *Graphical Passwords. A Survey*

[4] Smith. Authentication, From Passwords to Public Keys. Addison-Wesley 2002.

[5] OpenId Authentication 2.0 – Final. Consultado Mayo 2014 *http://openid.net/specs/openid-authentication-2_0.html*

11. Anexo I: Protocolo OpenId.

A continuación se explicará de manera detallada como se ha implementado el protocolo OpenId siguiendo las especificaciones del estándar tal y como se detallan en su página web [5].

11.1 Identificador

Un identificador OpenId es una URI, es decir, es una referencia a un recurso, que en este caso será una página de internet. En otras palabras, un identificador es una página web accesible desde internet.

Ej: http://nombre_recurso o https://nombre_recurso

En consecuencia, por cada usuario que se registre en el servidor de autenticación, se generará una página de internet, la cual sera accesible con el identificador. Lo que hace OpenGId es crear una carpeta con el nombre de usuario en el directorio raíz. Este recurso será identificador del usuario, en donde se guardarán las imágenes propias y contendrá el archivo index.php el cual será suministrado por el servidor automáticamente al acceder al recurso. El archivo index.php es donde se inicia el proceso de autenticación OpenId tal como se explica en el siguiente apartado.

11.2 Descubrimiento

El descubrimiento (**Discovery)** es el proceso mediante el cual, el servidor que solicita la autenticación, obtiene los datos necesarios para iniciarla a partir del identificador suministrado por el usuario. Para ello, se llevan a cabo los siguientes pasos:

- El servidor que requiere la autenticación accede al recurso suministrado: ej http://nombre_recurso
- En nuestro caso, el nombre_recurso es un directorio por lo tanto nuestro servidor devuelve el contenido generado por el fichero index.php que se encuentra en dicho directorio, tal y como está configurado en el servidor web. Dicho contenido incluye la siguiente información sobre donde se ejecuta el proveedor OpenId.

11.3 Redireccionamiento

Una vez descubierta la localización del proveedor, el servidor redirige al usuario a dicho proveedor, con el fin de que se produzca el proceso de autenticación, enviando los siguientes parámetros mediante el método POST.

Parámetro	***Valor***
openid.ns	Versión de OpenId
openid.mode	Para que se proceda a verificar identidad debe ser “checkid_setup”
openid.claimed_id	El identificador del usuario
openid.identity	El identificador local, suele ser el mismo que el claimed_id.
openid.assoc_handle	No utilizado en este prototipo
openid.return_to	Dirección a la que se ha de retornar
openid.realm	Identificador del servidor que requiere la autenticación

Tabla 4. Parámetros utilzados en petición OpenId

11.4 Retorno al origen

Una vez realizado el proceso de autenticación en el proveedor de la identificación, se retorna al usuario al servidor que requería la misma. En el redireccionamiento, también se envían por medio del método POST los parámetros detallados a continuación, donde entre otros se encuentra el resultado del proceso de autenticación.

Parámetro	***Valor***
openid.ns	Versión de OpenId
openid.mode	“id_res” en autenticación positiva “cancel” en caso negativo
openid.op_endpoint	El URL del proveedor
openid.claimed_id	El identificador del usuario
openid.identity	El identificador local, suele ser el mismo que el claimed_id.
openid.return_to	Copia exacta del parámetro recibido.
openid.response_nonce	Cadena de caracteres compuesta por la hora en el servidor en formato UTC concatenado con otros caracteres ASCII opcionales.
openid.signed	Lista de parámetros que serán firmados para comprobar autenticidad de la respuesta

openid.sig	Firma en Base64 de los parámetros indicados anteriormente.

Tabla 5. Parámetros utilizados en la respuesta de una petición OpenId.

11.5 Verificando la autenticación

Para comprobar que la respuesta recibida es válida, se realiza una comunicación entre el servidor y el proveedor ajena al usuario, donde el servidor envía una copia de respuesta recibida cambiando solamente el valor del parámetro openid.mode con el valor “check_authentication”.
En este caso, esta información no se envía por el método POST sino en el contenido bajo el formato clave:valor.

Recibido el mensaje por el proveedor, este comprueba que la firma sea correcta. El proveedor identificará la autenticación por medio del nonce.
La respuesta la hará en el mismo formato, pero con el valor “isvalid:true” si es correcto o “isvalid:false” en caso contrario.

12. Anexo II: Manual de Usuario

A continuación se describen los pasos básicos para poder utilizar una cuenta de OpenGId para poder autenticarse en aquellos sitios compatibles con OpenId.

12.1 Creación de una cuenta.

Lo primero es crear una cuenta. El prototipo dispone de la funcionalidad para hacerlo, a través de un formulario.

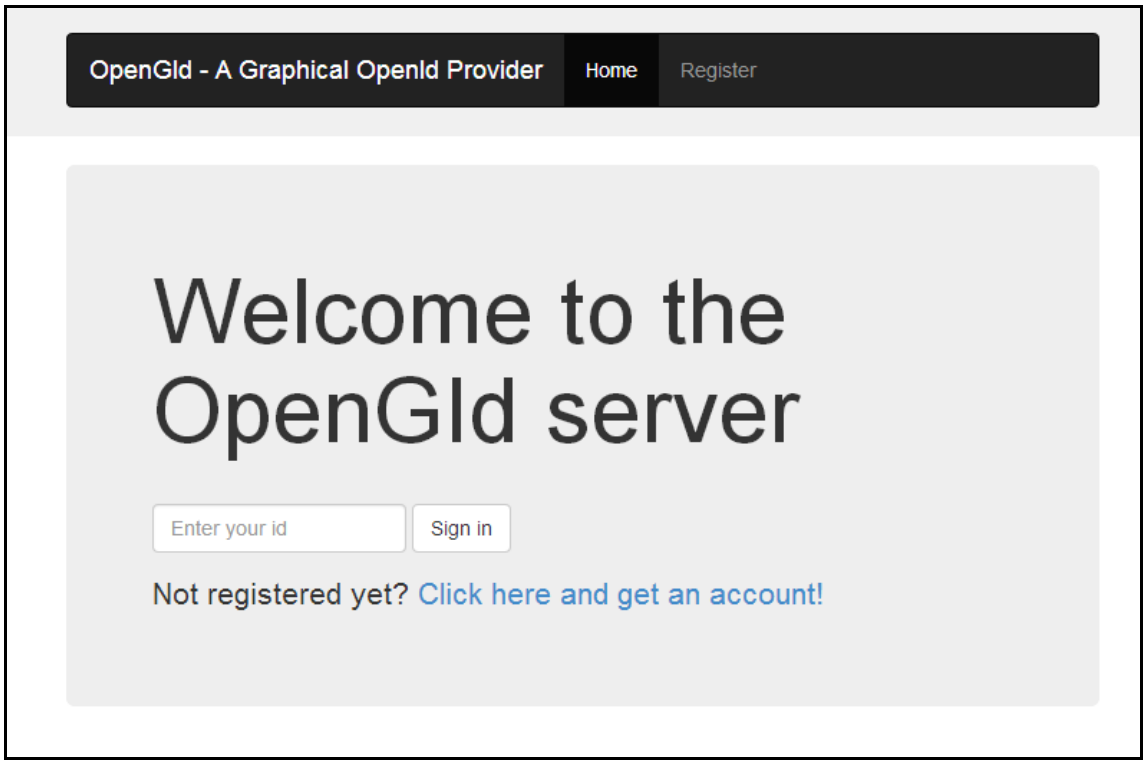


Ilustración 26. Página principal de OpenGid

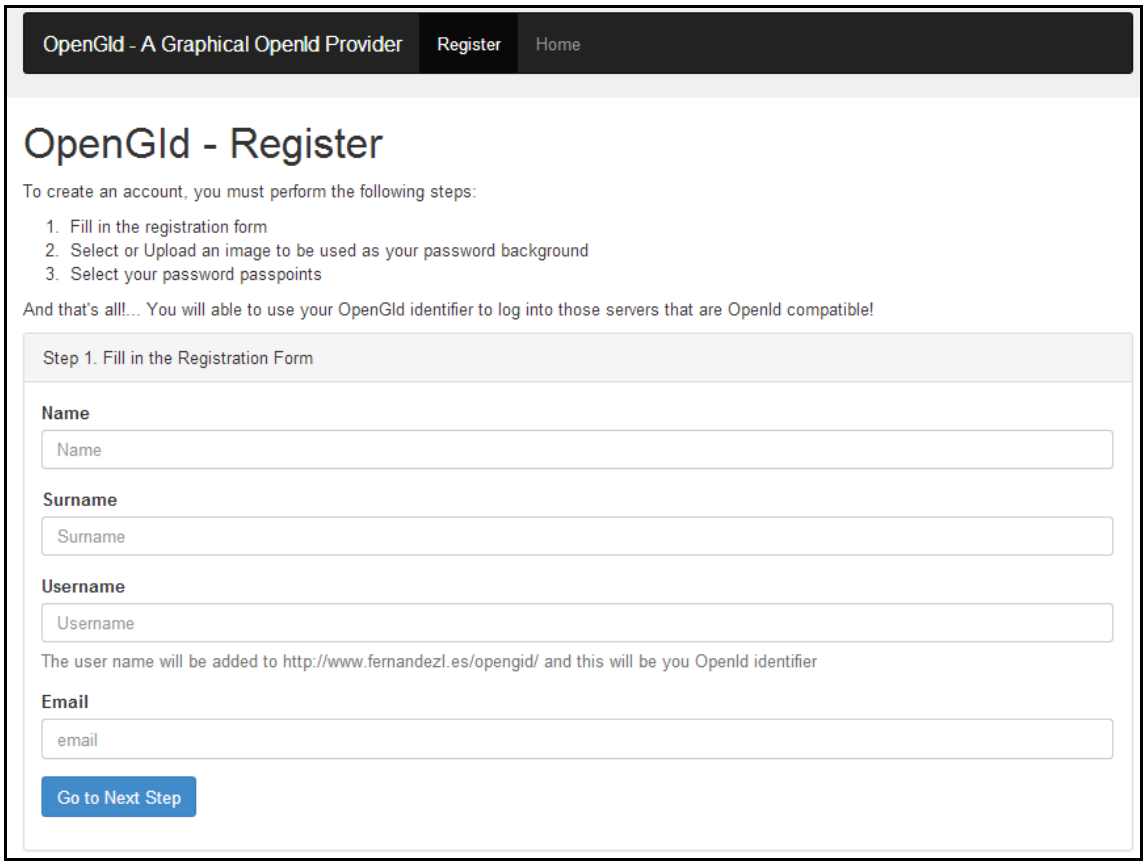


Ilustración 27. Formulario de registro.

Seguidamente, se habrá de crear la contraseña, para ello el prototipo propone una serie de imágenes por defecto, o permite que el usuario utilice una imagen propia.

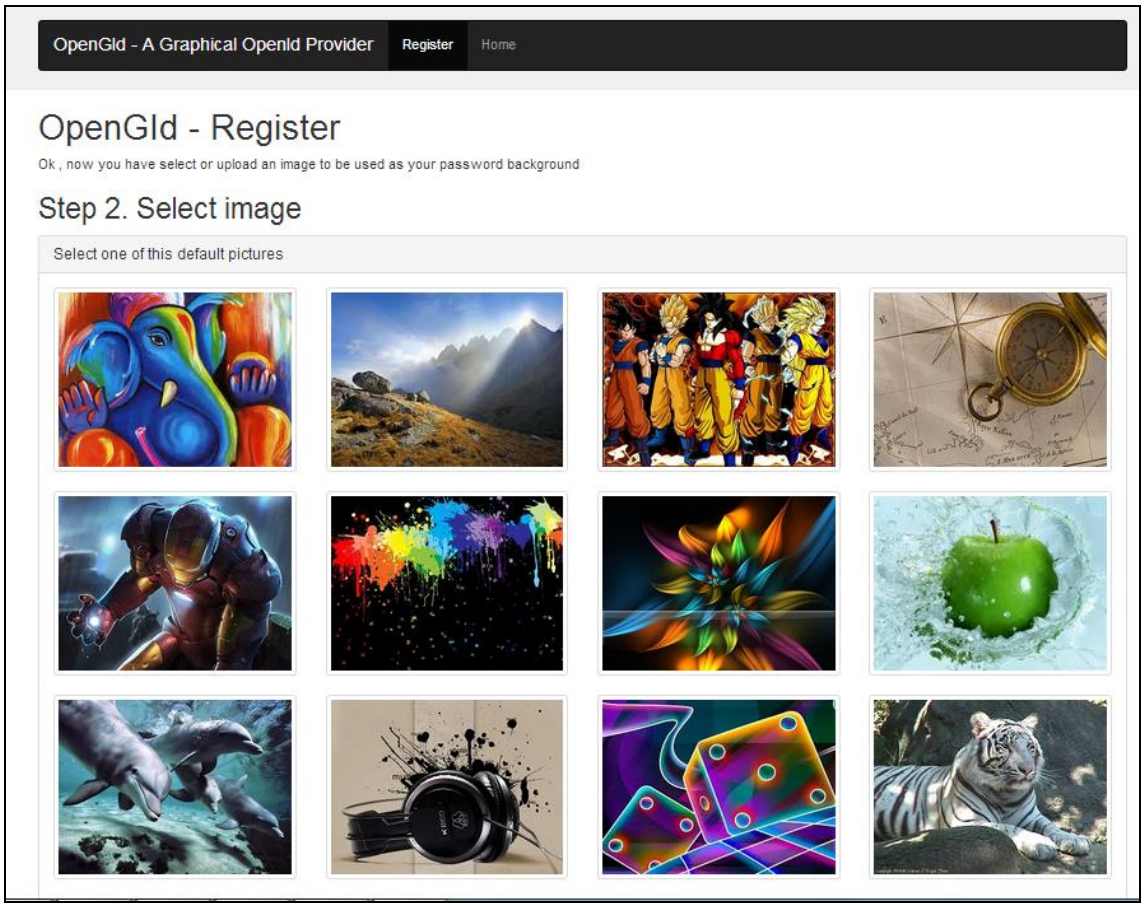


Ilustración 28. Elección de imágen de contraseña.

A continuación el usuario ha de seleccionar los puntos que representarán su contraseña sobre la imagen escogida.

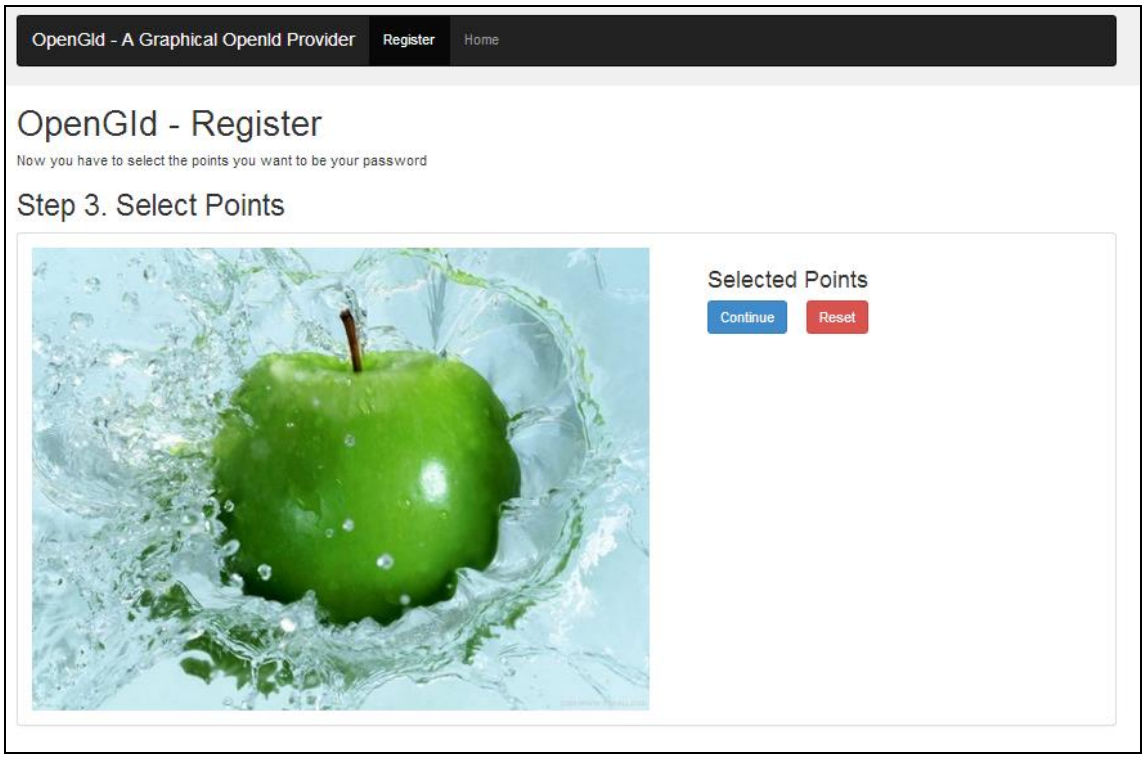


Ilustración 29. Selección de puntos sobre imagen.

12.2 Panel de Control.

El prototipo permite al usuario acceder a un panel de control donde encontrará un historial de autenticaciones como también información estadística sobre el uso de su contraseña. También dispone de la posibilidad de cambiarla.

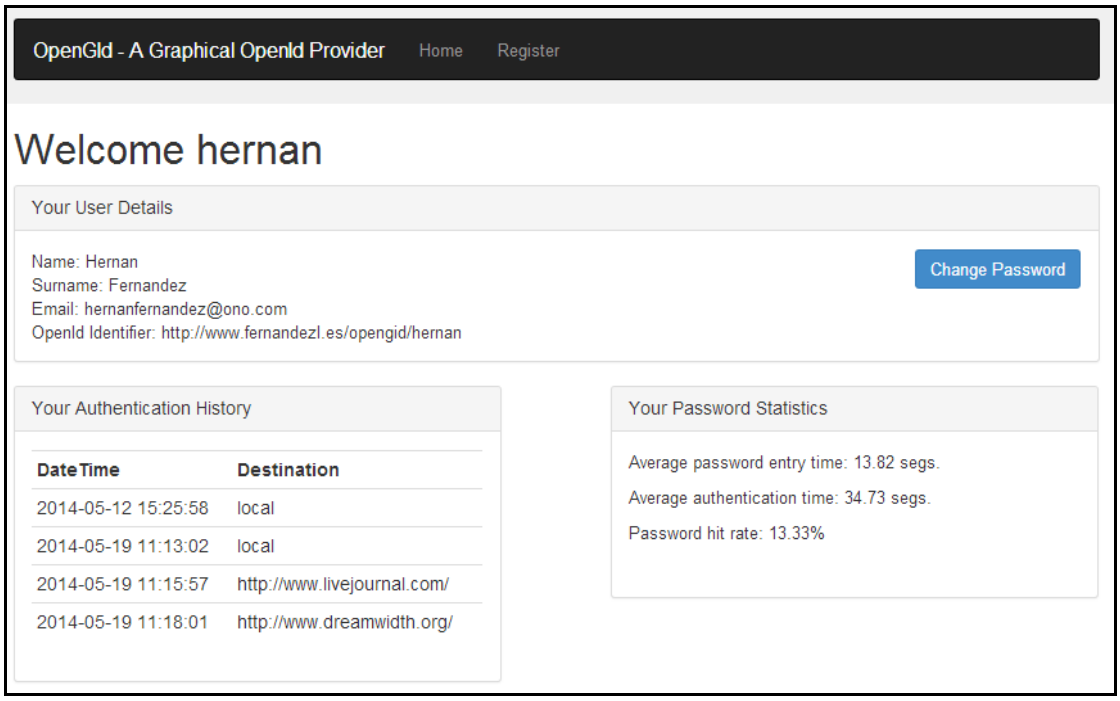


Ilustración 30. Panel de control.

12.3 Uso del identificador.

El identificador puede ser utilizado en todo sitio que soporte OpenId y permita identificadores otorgados por nuevos proveedores. Actualmente este sistema es muy utilizado pero la mayoría de sitos solo permiten proveedores como Facebook, Google, Yahoo. Un ejemplo donde se podría utilizar son los siguientes:

- www.dreamwith.org

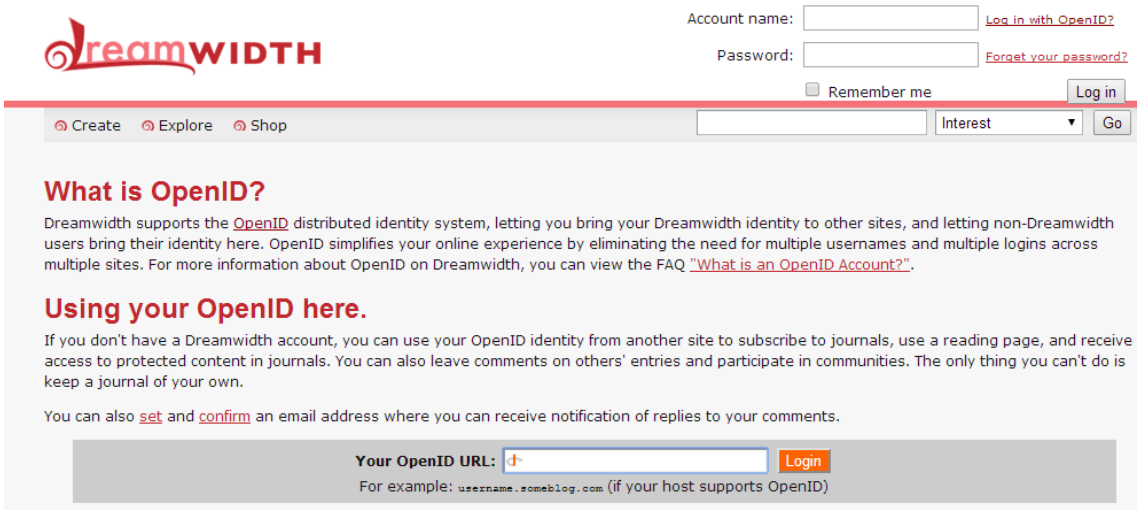


Ilustración 31. Login OpenId del portal DreamWith

- www.livejournal.com

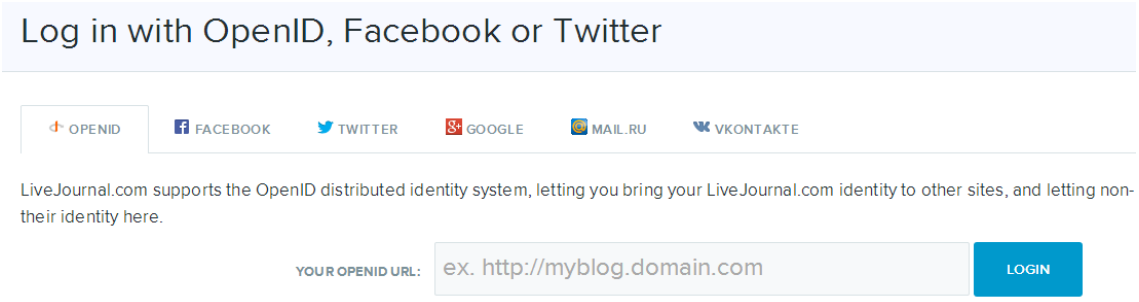


Ilustración 32. Login OpenId del portal LiveJournal