Sergio Arévalo Roa

Phishing y psicología: una plataforma de evaluación

TRABAJO DE FINAL DE GRADO

dirigido por Agustí Solanas y Edgar Batista

Grado de Ingeniería Informática

UNIVERSITAT ROVIRA I VIRGILI

Tarragona

2023

Resum.

Aquest projecte té com a objectiu realitzar una eina d'estudi que relacioni els trets psicològics d'un usuari i la probabilitat existent de caure en un atac de phishing.

El motiu d'aquest projecte és la necessitat d'estandardització de criteris al voltant dels aspectes psicològics en l'àmbit de la ciberseguretat i aportar a la comunitat una eina més personalitzada que l'educació i conscienciació de l'usuari de forma generalitzada.

S'implementarà una eina en forma web, permetent a l'usuari realitzar diferents test sobre la seva informació demogràfica, trets personals, intel·ligència i coneixement de phishings. Aquesta web comptarà amb un disseny responsiu i multi plataforma, i permetrà persistir la informació obtinguda perquè més tard serveixi com a font d'informació en els estudis que encara estan per realitzar-se.

Aquest projecte deixa oberta una porta a futures recerques en l'àmbit havent desenvolupat una eina que pugui ajudar a construir aquest pont entre tots dos mons.

Resumen.

En este proyecto se pretende realizar una herramienta de estudio que relacione los rasgos psicológicos de un usuario y la probabilidad existente de caer en un ataque de phishing.

El motivo de este proyecto es la necesidad de estandarización de criterios alrededor de los aspectos psicológicos en el ámbito de la ciberseguridad y aportar a la comunidad algo más personalizado que la educación y concienciación del usuario de forma generalizada.

Se implementará una herramienta en forma web, permitiendo al usuario realizar diferentes test sobre su información demográfica, rasgos personales, inteligencia y conocimiento de phishings. Esta web contará con un diseño responsivo y multiplataforma y permitirá persistir la información obtenida para que más tarde sirva como fuente de información en los estudios que todavía están por realizarse.

Este proyecto deja abierta una puerta a futuras investigaciones en el ámbito habiendo desarrollado una herramienta que pueda ayudar a construir este puente entre ambos mundos.

Abstract.

In this project, the aim is to develop a study tool that links a user's psychological traits to the existing probability of falling victim to a phishing attack.

The motivation behind this project is the need for standardized criteria regarding psychological aspects in the field of cybersecurity, and to provide the community with something more personalized than generalized user education and awareness.

A web-based tool will be implemented, allowing users to take various tests regarding their demographic information, personal traits, intelligence, and knowledge of phishing. This website will have a responsive and cross-platform design and will store the obtained information for later use as a data source in further studies yet to be conducted.

This project opens the door to future research in the field, having developed a tool that can help build a bridge between these two worlds.

Índice

1 INTRODUCCIÓN 4
1.1 PHISHING 4
1.2 LA RELACIÓN ENTRE EL USUARIO Y PHISHING 5
1.2.1 Operas 6
1.2.2 IQ 6
1.2.3 Test de Mail Phishing 7
1.3 MOTIVACIÓN 8
1.4 OBJETIVOS 9
2 REQUISITOS DE LA HERRAMIENTA 10
2.1 REQUISITOS FUNCIONALES 10
2.2 REQUISITOS NO FUNCIONALES 11
3 ANÁLISIS DE REQUISITOS 12
4 DISEÑO DE LA APLICACIÓN 14
4.1 ARQUITECTURA 14
4.2 DISEÑO DE LAS BASES DE DATOS 15
4.3 APLICACIÓN WEB 17
4.3.1 Modelo 17
4.3.2 Vista 18
4.3.3 Controlador 19
5 IMPLEMENTACIÓN 21
5.1 LENGUAJES 21
5.2 ESTRUCTURA DEL PROYECTO 22
5.3 ALGORITMOS ESPECÍFICOS 24
5.3.1 Operas 24
5.3.2 IQ 25
5.3.3 Mail Phishing 27
5.3.4 Envío de datos 28
6 JUEGO DE PRUEBAS 29
7 CONCLUSIONES 33
8 REFERENCIAS 35
ANNEXOS 36
ANNEXO 1. INSTALACIÓN Y PUESTA EN MARCHA 36
ANNEXO 2. MANUAL DE USUARIO 40

Índice de tablas

TABLA 1. INTERACCIÓN DE LOS COMPONENTES MVC 17
TABLA 2. JUEGOS DE PRUEBAS 32

Índice de figuras

FIGURA 1. EJEMPLO TEST IQ WAIS 7
FIGURA 2. EJEMPLO DEL TEST SOBRE MAIL PHISHING 8
FIGURA 3. DIAGRAMA DE CASOS DE USO 12
FIGURA 4. MODELO CLIENTE-SERVIDOR 15
FIGURA 5. MODELO RELACIONAL DE LA BASE DE DATOS 16
FIGURA 6. PATRÓN MVC 20
FIGURA 7. TEST SOBRE MAIL PHISHING EN MODO ENTRENAMIENTO 28
FIGURA 8. PUERTOS DE ESCUCHA MAMP 36
FIGURA 9. CONFIGURACIÓN RUTA DE LA APLICACIÓN 36
FIGURA 10. GESTOR BASE DE DATOS 37
FIGURA 11. CREACIÓN BASE DE DATOS 37
FIGURA 12. CREACIÓN BASE DE DATOS 2 38
FIGURA 13. IMPORTAR BASE DE DATOS 38
FIGURA 14. SELECCIÓN DE RUTA EN BASE DE DATOS 38
FIGURA 15. PAGINA HOME DE LA WEB 39
FIGURA 16. INICIO TEST EN MODO LINEAL 40
FIGURA 17. ACCESO TEST OPERAS INDIVIDUAL 40
FIGURA 18. INICIO TEST DE GOOGLE 41
FIGURA 19. ACCESO TEST IQ INDIVIDUAL 41
FIGURA 20. ACCESO TEST MAIL PHISHING DE GOOGLE INDIVIDUAL 41
FIGURA 21. INICIO TEST DE GOOGLE 41

1 Introducción

Los casos de ataques de phishing tienen una tendencia al alza en los últimos años. Solo en el último cuarto del año 2022 se han observado un total de 1.270.883 ataques de phishing [1], el récord en ataques jamás observado según la fundación APWG. Visto el dato anterior, se deben plantear soluciones que ayuden a combatir el aumento de ataques. Pero, ¿Cómo se pueden disminuir estos ataques? Para intentar plantear un abordaje de este problema, en este trabajo, se quiere encontrar la relación que existe entre los rasgos que definen a un usuario y su probabilidad de sufrir un ciberataque. Algunas de las cuestiones que se plantean son:

- ¿Existe relación entre los rasgos principales que definen la personalidad de una persona y la probabilidad de caer en un ataque de phishing?
- ¿Hay una relación entre la inteligencia del usuario y esta probabilidad?
- ¿Es el factor demográfico un factor que influya en el riesgo de sufrir uno de estos ataques?
- ¿Es capaz un usuario de reconocer un posible ataque de phishing?

Teniendo en cuenta lo anterior, el objetivo de este trabajo es encontrar la relación que existe entre los diferentes rasgos mencionados y la probabilidad de caer en un ataque de phishing. Para ello, se evaluarán los rasgos anteriores mediante cuestionarios. Los resultados de estos van a permitir crear un perfil de usuario para más tarde relacionarlo con la probabilidad mencionada. Esta evaluación se realizará mediante unos test implementados en una página web, más adelante en este documento detallarán estos. Una vez se hayan evaluado los rasgos del usuario, se le presentarán unos posibles casos de phishing. La presentación de estos casos pretende observar el grado de conciencia que tiene el usuario para detectar un posible ataque. Por último, se pretende que los resultados permitan, en un futuro, encontrar la relación sujeto-probabilidad.

1.1 Phishing

El termino phishing lleva conviviendo con nuestra sociedad desde 1996 gracias a que fue mencionado por primera vez en la publicación 2600: The Hacker Quarterly [2]. Derivado de la palabra ingles fishing nace el termino phishing. Este nombre hace referencia al modus operandi utilizado para realizar un ataque de estas características. La forma de actuar del atacante es la de lanzar el anzuelo y esperar a que las víctimas caigan en él. En la publicación de 2600 se menciona el phishing dando a conocer un ataque realizado a la compañía AOL. En este, un grupo de hackers intentaba robar cuentas a trabajadores la compañía. La forma de actuar de este atacante era la de suplantar la identidad de otro trabajador. De este modo, intentaba persuadir con mensajes a una potencial víctima. En estos mensajes se pretendía que el empleado atacado proporcionase información confidencial, como podrían ser contraseñas. Por tanto, podemos incluir el phishing como un ataque de ingeniería social. El termino ingeniería social hace referencia a la práctica de obtener información confidencial manipulando al usuario [3]. Además de lo comentado, en este tipo de ataques, en muchas ocasiones se buscan conseguir permisos o accesos a diferentes sistemas con la finalidad de comprometer la seguridad de la información obtenida. Este tipo de ataques se sustentan en que la parte débil de los sistemas siempre va a ser el usuario, por este motivo, lo utilizan como punto de acceso a la información.

Existen varias formas de llevar a cabo un ataque de esta característica, y aunque cada vez nacen nuevos métodos, a continuación, podemos ver cuáles son los principales según su forma de actuar o su vector de ataque [4]:

- **Mail Phishing o Spray and pray.** Es el tipo de phishing tradicional en el que el atacante realiza un envío masivo de correos electrónicos a diferentes usuarios. Este tipo de phishing basa su éxito en el hecho de suplantar a entidades para conseguir información del usuario.
- **Smishing.** Es un tipo de ataque phishing que se basa en suplantar la identidad de generalmente compañías con el fin de conseguir información por parte del usuario. Suele llevarse a cabo enviando un SMS a la posible victima haciéndose pasar por otra persona, generalmente compañías. En estos mensajes se pretende que el usuario acceda a un link e introduzca sus datos personales. Por tanto, el vector de ataque es un SMS.
- **Vishing.** Es similar a los dos anteriores con la diferencia de que el engaño se intenta realizar vía llamada telefónica.
- **Whaling.** La diferencia de este tipo de ataque con el resto es que este se dirige a personas importantes como podrían ser, por ejemplo, altos cargos de compañías. Es un tipo de phishing más personalizado que tiene como finalidad adquirir información confidencial.
- **Spear Phishing.** Aunque parecido al Whaling, este tipo de ataque phishing está orientado a engañar a una persona en concreto. Por este motivo, es un tipo de ataque phishing aún más personalizado que el anterior.
- **Pharming.** Es el tipo de phishing que consiste en redirigir a un usuario a una página web falsa mediante vulnerabilidades del servidor DNS.

Aunque la lista de tipos de ataques phishing pueda ampliarse con otros tipos de ataque, ya se han nombrado los más característicos. Algunos otros ejemplos de ataques podrían ser ataques como el Social Network Phishing, Evil Twin, Watering Hole Phishing, etc.

1.2 La relación entre el usuario y phishing

Como se ha podido observar en los apartados anteriores, los ataques de phishing se basan en las personas. Por este motivo, hace falta entender cómo funcionan los usuarios a nivel mental en situaciones que puedan suponer un riesgo para el en Internet. En este sentido, la personalidad es una característica que puede ser de gran importancia cuando una persona recibe un ataque de phishing. Tanto es así, que esta relación se plantea en varios estudios alrededor de cómo funciona el usuario delante de un posible ataque de phishing. En algunos de estos, se destaca la falta de una estandarización que establezca el papel de la psicología humana en el contexto del phishing [5].

Con el fin de crear una base para el estudio de esta relación, en este trabajo, se pretende analizar los principales rasgos de personalidad de una persona para así poder encontrar un punto de unión entre estos y el riesgo asociado a caer en un ataque de estas características.

Para ello se utilizarán diferentes herramientas, principalmente las siguientes:

- **Test de personalidad.** Se pretende utilizar este test para evaluar los cinco rasgos característicos que definen a las personas.

- **Test de coeficiente intelectual.** Se pretende medir el coeficiente intelectual de la persona para utilizar este dato como variable en el estudio.
- **Test de mail de susceptibilidad al phishing.** Se utilizará esta herramienta para evaluar y entrenar al usuario sobre diferentes phishings cuyo vector de ataque es el correo electrónico.

1.2.1 Operas

El test Operas (Overall Personality Assesment Scale) [6] implementado es un test que pretende analizar los Big Five. Los Big Five son cinco rasgos de la personalidad de una persona que sirven en investigaciones psicológicas para determinar los patrones generales de la conducta, la forma de pensar y el modo en el que las personas sienten. Los cinco rasgos en los que se basa el estudio son:

- **Extraversión.** Mide la tendencia que tienen las personas a sociabilizar con otras personas. En esta parte se evalúan comportamiento como la tendencia a ser hablador, a optar por situaciones sociales o por el aislamiento, etc.
- **Responsabilidad.** Sirve para determinar el grado en el que las personas se responsabilizan de su vida, evaluando partes como la planificación de objetivos, que tanto eficiente es, predisposición a la organización, etc.
- **Estabilidad Emocional.** Da la oportunidad de evaluar el equilibrio mental de una persona. De este modo podemos ver la tendencia de esa persona a mantener la calma, el grado de predisposición a las emociones negativas, etc.
- **Amabilidad.** Mide los rasgos que hacen que una persona sea propensa a tener en cuenta el bienestar de las personas a su alrededor, estudiando aspectos como la empatía, confianza en el resto de personas, etc.
- **Apertura a la experiencia.** Permite ver el grado de predisposición que tiene una persona a conocer más allá de su día a día i cambiar el modo de pensar. Se estudia a través de características como la imaginación, el interés por arte, etc.

La forma de evaluar en este test es mediante la presentación al usuario de preguntas cortas sobre situaciones cuotidianas. Estas preguntas no tienen una respuesta correcta, sino que se mide del uno al cinco dependiendo cuanto de acuerdo esté el usuario con esa pregunta.

1.2.2 IQ

Como una parte de las herramientas que se utilizarán en este proyecto para definir los rasgos característicos de las personas, se desarrollará el test de medición de coeficiente intelectual WAIS [7]. Aunque este test se compone de varios apartados para la medición de diferentes aspectos, en este proyecto se desarrollará una de las pruebas de razonamiento basada en matrices. Esta prueba, pretende mostrar al usuario una matriz que representa una serie de figuras incompletas. El usuario debe escoger la figura que falta para completar la serie.

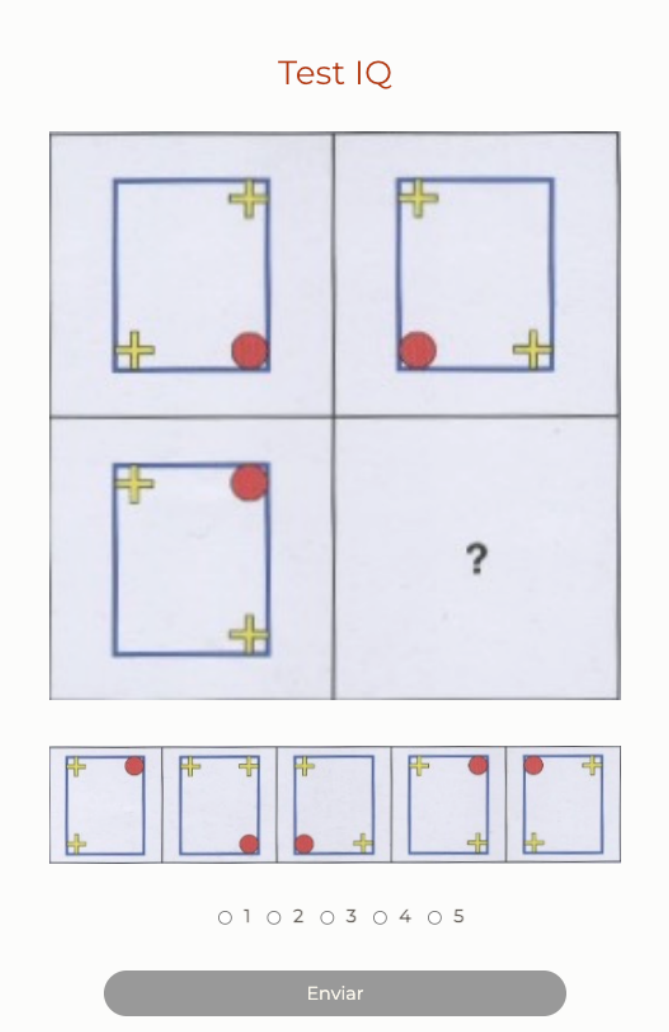

Figura 1. Ejemplo test IQ WAIS

1.2.3 Test de Mail Phishing

Como parte del estudio, una vez se han determinado las características principales de un sujeto, se pretende dar la oportunidad a este de autoevaluarse con casos de correos electrónicos reales. Estos correos se basan en la herramienta de Google [8] destinada a tal fin. Se ha decido utilizar esta herramienta como base de nuestro trabajo dada la gran popularidad de Google entre los usuarios, aunque cualquier tipo de mail sería aplicable para realizar el estudio. Por tanto, el objetivo de esta sección, es mostrar al usuario una serie de correos electrónicos y que este pueda determinar si se trata de un phishing.

Como parte de la funcionalidad que se le quiere dar a la plataforma, el usuario tendrá la posibilidad de realizar el test en modo entrenamiento y con ello poder observar los diferentes detalles que le ayudarán a detectar phishings, en este caso por el vector de correo electrónico.

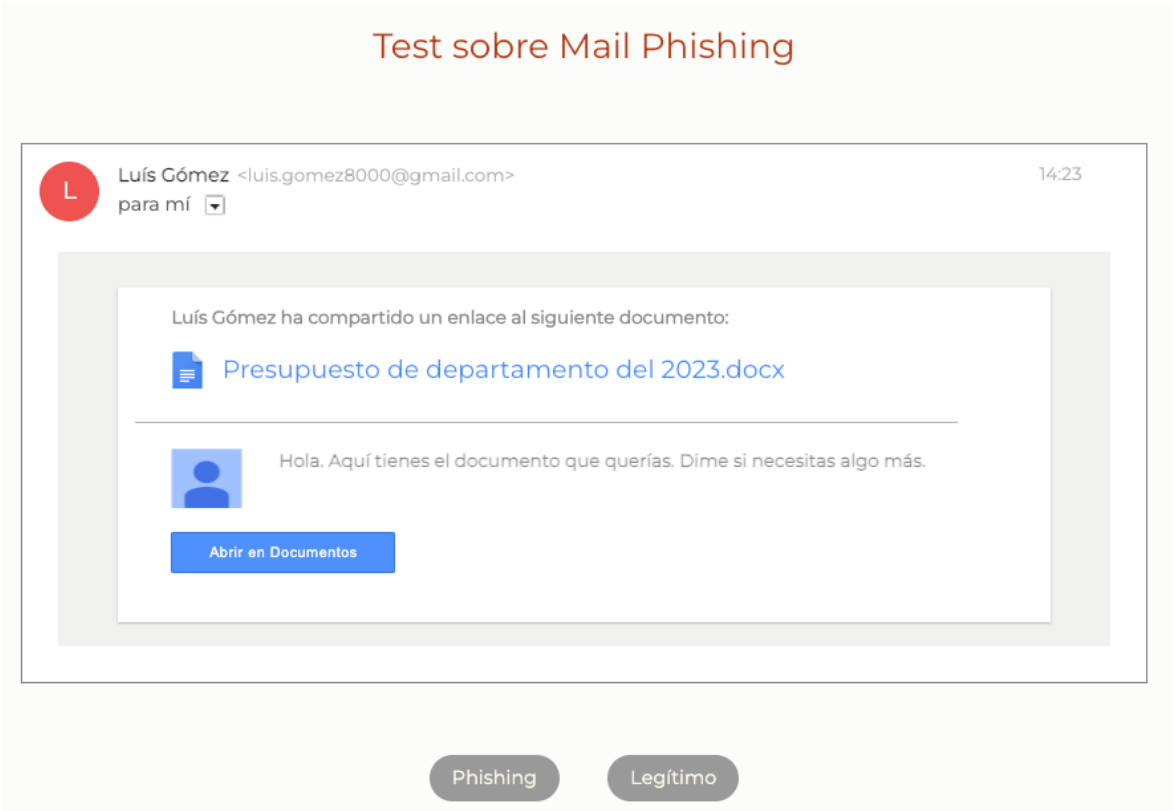

Figura 2. Ejemplo del test sobre Mail Phishing

1.3 Motivación

Los esfuerzos por la lucha anti phishing están creciendo junto con el número de ataques efectuados. Esto lleva a pensar que hay que sofisticar el grado de estudio de los factores externos que afectan al hecho de que un usuario caiga en uno de estos ataques. Actualmente, los recursos disponibles para estudiar la relación que existe entre los rasgos que definen a la persona y el riesgo de que caiga en un ataque de estas características son bastante reducidos. Quedando la mayoría de ellos reducidos a plataformas de concienciación, usualmente para empresas.

Desde hace un tiempo, parece que la estrategia para la lucha con el phishing ha sido de educación y concienciación, dejando totalmente de lado las diferencias existentes entre los diferentes usuarios.

En la actualidad, diferentes estudios empiezan a revelar que podría existir una relación entre los rasgos de personalidad y el riesgo mencionado. Un ejemplo es de cómo se relaciona el rasgo de extraversión con la posibilidad de ejecutar las acciones presentadas en un correo de phishing [9], apuntando a que un mayor grado de está característica en las personas influye de forma positiva en su tendencia a caer en estos ataques. También, podemos ver que se relacionan aspectos como la edad, el género y la situación demográfica del usuario con la posibilidad mencionada a caer en ataque de estas características [10]. Sirviendo de ejemplo el inicio de la relación de la extraversión con la susceptibilidad del phishing, se puede intuir que la investigación en este sentido tiene un futuro prometedor y puede ayudar a detectar patrones de la personalidad de las personas que afecten al riesgo de caer en un phishing.

1.4 Objetivos

Una vez hemos visto la problemática actual con el phishing y el vacío existente en el estudio de la relación entre la personalidad del usuario y la probabilidad de caer en un ataque de estas características, se definen los siguientes objetivos a desarrollar:

- Crear una herramienta que evalúe a los usuarios en base su situación demográfica, sus rasgos de personalidad y su coeficiente intelectual.
- Crear una herramienta que permita al usuario realizar un test de conocimiento sobre diferentes Mail Phishing.
- Plantear una plataforma que de pie a continuar estudiando la correlación entre los factores de estudio nombrados.
- Desarrollar una aplicación siguiendo buenas prácticas de programación, contando ésta con usabilidad, eficiencia y responsividad.

Por lo tanto, como objetivo más generalista, se pretende crear un punto de partida en el estudio de los diferentes rasgos de las personas en el contexto de la victimización del phishing utilizando herramientas de medición de rasgos característicos y autoevaluaciones del usuario.

2 Requisitos de la herramienta

2.1 Requisitos funcionales

La identificación de los requisitos funcionales de la aplicación web va a permitir definir con precisión cuales son las funciones que ésta debe implementar. A continuación, podemos ver un listado de los requisitos mínimos que debe cumplir:

- Una página principal que de acceso a cada test de forma individual
- La aplicación debe permitir realizar un test de personalidad mediante el test Operas, mediante lo siguiente:
    - Ha de mostrar todas las preguntas en el orden establecido y de forma independiente.
    - Al hacer click a enviar debe haber una persistencias de las respuestas en la base de datos.
    - Una vez guardada las respuesta, se debe mostrar la siguiente pregunta.
    - Al terminar el test por completo, se debe calcular la puntuación obtenida.
- La aplicación debe permitir realizar un test de cálculo del IQ, mediante lo siguiente:
    - Ha de mostrar todas las preguntas en el orden establecido y de forma independiente.
    - Las preguntas se mostrarán mediante imágenes.
    - Al hacer click en enviar debe haber una persistencia de las respuestas en la base de datos.
    - Una vez guardada la respuesta, se debe mostrar la siguiente pregunta.
    - Al final del test se debe calcular la nota final de este.
    - El test durará en base a los criterios de corrección.
- La aplicación debe permitir realizar un test sobre conocimiento de diferentes Phishings en formato mail, mediante lo siguiente:
    - Ha de mostrar todas las preguntas en el orden establecido y de forma independiente.
    - Se tendrá la oportunidad de realizar el test en modo entrenamiento o en modo experimento.
    - Si se realiza el test en modo entrenamiento, después de cada pregunta se mostrarán diferentes explicaciones de ese mail en concreto.
    - Las diferentes explicaciones se mostrarán mediante un símbolo rojo al lado de cada sección que necesita explicación.
    - Al hacer click en continuar se guardará la respuesta en la base de datos.
    - Una vez guardada la respuesta se debe mostrar la siguiente pregunta.
    - Al terminar el test por completo, se debe calcular la nota obtenida.
    - Al terminar el test aparecerá una pantalla informando de que se han finalizado todos los test.
- En el test de Mail Phishing se mostrarán diferentes mails con estilo similar a los de Gmail.
- Para los test sobre información demográfica, Operas y IQ habrá una pantalla principal que dará instrucciones e información sobre estos.

- Para el test de Mail Phishing habrá una pantalla principal que de instrucciones y información sobre estos y de la posibilidad de escoger el modo en el que se realiza el test.
- La interfaz de usuario debe ser intuitiva y responsive, de modo que se adapte a cualquier tamaño de pantalla.
- Los diferentes test deben poder realizarse tanto de forma lineal, todos los test seguidos, como de forma individual.
- Los resultados deben ser guardados para ser accesibles a posteriori.

2.2 Requisitos no funcionales

Por otro lado, se describen diferentes requisitos no funcionales que debe poseer el proyecto software. Estas son:

- Para la implementación se utilizará MAMP como entorno de desarrollo para MacOs.
- El backend se implementará utilizando PHP y MySQL.
- El frontend se implementará utilizando HTML, CSS y Javascript.
- Se utilizará a Bootstrap a modo de framework para construir una interfaz responsive.
- El índice se implementará utilizando PHP.
- Se utilizará una plantilla a modo de base para el frontend.
- El sistema debe ser desarrollado en base al patrón de diseño de software MVC.
- Se esconderán las URL y sus variables utilizando métodos POST mediante AJAX.
- Se realizará la comunicación entre las diferentes partes mediante el protocolo HTTPS.
- El sistema debe ser capaz de procesar todas las respuestas del usuario.
- El sistema debe ser capaz de mantener una sesión de usuario durante todo el tiempo que este utilice la web.
- La plataforma debe poder utilizarse en cualquier navegador web.
- Las respuestas de los usuarios deben ser persistidas en una base de datos para acceder a estas cuando se desee estudiar los resultados.
- Las preguntas y diferentes opciones de los test deben estar en la base de datos para su consulta y muestra al usuario.
- Se implementará un algoritmo de corrección de cada test.
- Se tratarán las inserciones y consultas en la base de datos para evitar ataques de inyección SQL.
- El tiempo de respuesta de cada uno de los test debe ser lo más rápido posible.
- Los datos de los usuarios deben ser almacenados de forma segura y no podrán compartirse con terceros.
- Una vez en el servidor, la aplicación debe estar disponible las 24 horas, los 7 días de la semana.
- Se deben realizar copias de seguridad para asegurar la no pérdida de datos.

3 Análisis de requisitos

Una vez se ha realizado el estudio de los requisitos que van a guiar el desarrollo de este proyecto se pretende realizar un análisis de estos para identificar las principales acciones a tomar y actividades a implementar. Para hacerlo se va a construir un diagrama de casos de uso de la aplicación que va a ayudar a entender el funcionamiento del proyecto, ver como se comunican las diferentes partes, validar de forma visual y, por último, servirá como referencia a la hora de realizar las diferentes pruebas de funcionamiento necesarias para validar el sistema.

A continuación, se puede ver el diagrama de casos de uso pensado para este proyecto de software en concreto.

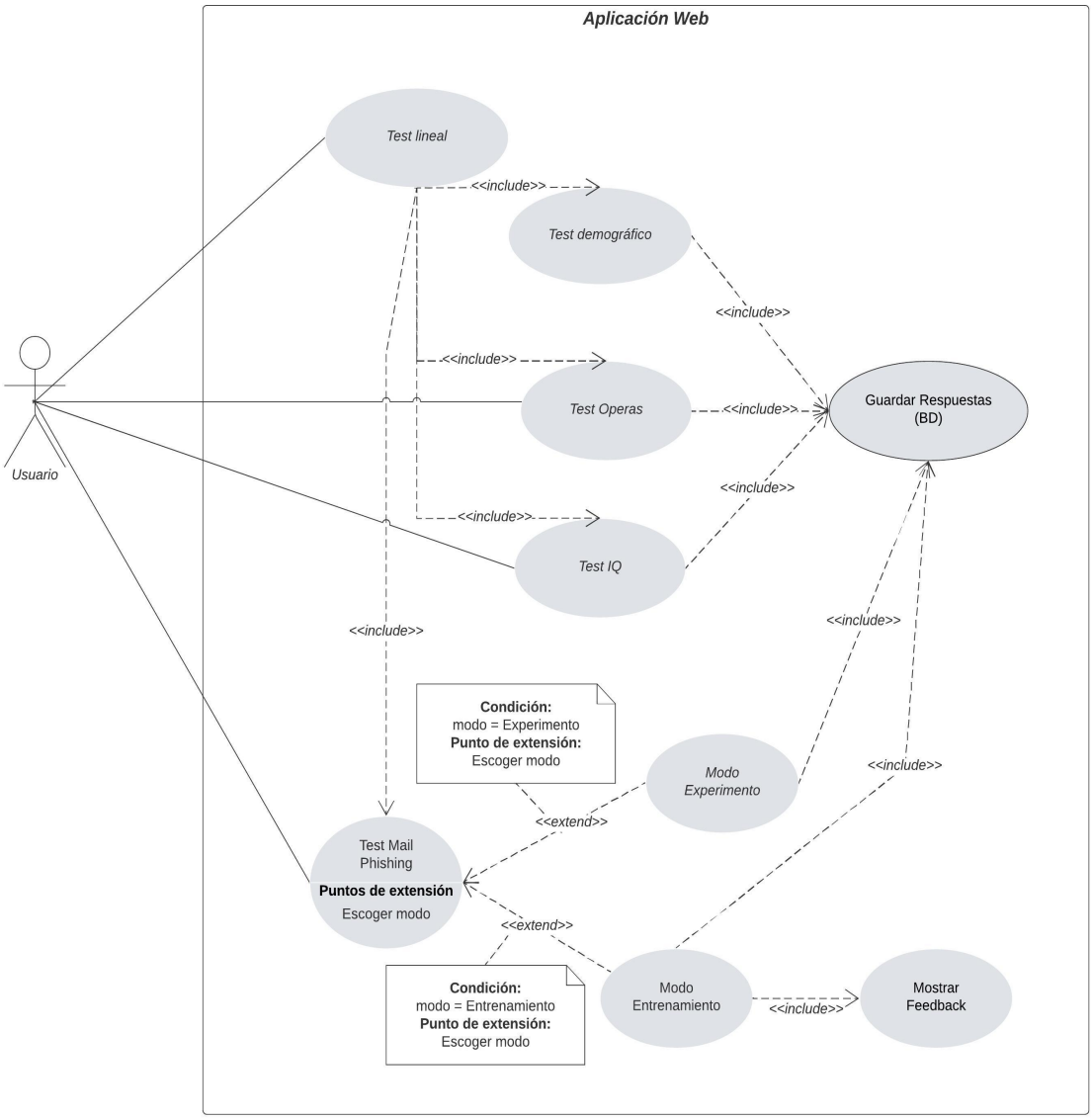

Figura 3. Diagrama de casos de uso

Como se puede observar, las acciones que va a poder realizar un usuario están bastante acotadas pudiendo quedar estas reducidas al hecho de realizar cada test de forma individual o de realizarlo de forma lineal.

4 Diseño de la aplicación

El diseño de una aplicación permite al desarrollador establecer una base sobre la que trabajar durante la implementación de esta. Este punto, pretende crear una imagen de cómo será la arquitectura de la aplicación web con el fin de dotarla de usabilidad, eficiencia, seguridad y que, a su vez, permita al investigador obtener conclusiones sobre el estudio realizado.

Mediante el análisis de los requisitos en el punto anterior, se pueden identificar varias tecnologías y métodos que se van a necesitar para implementar correctamente todas las funcionalidades.

En este apartado, hablaremos sobre cómo se va a construir la arquitectura del proyecto, se dará información sobre cómo se va a utilizar el Modelo-Vista-Controlador (MCV), se definirán las bases de comunicación entre los componentes y se definirá, también, el diseño de la base de datos que se va a utilizar en el backend.

4.1 Arquitectura

Para la implementación de este proyecto software se va a seguir una arquitectura cliente servidor. Este modelo es muy comúnmente utilizado en el desarrollo de páginas web. En este tipo de implementaciones, la aplicación se sustenta gracias a dos roles como son el servidor, o backend, y el cliente, o frontend. El backend del lado servidor, realizará las operaciones junto con la base de datos. Estas operaciones serán tanto de consulta, como inserción, actualización, etc. Por otro lado, en el lado cliente, mediate el frontend, el usuario podrá interaccionar con la página web. El modelo cliente servidor cuenta con diferentes características como pueden ser:

- **Separación de responsabilidades.** El servidor, por un lado, se encarga de realizar las operaciones junto con la base de datos mientras que el cliente se encarga de procesar la interfaz de usuario.
- **Comunicación** entre las diferentes partes mediante solicitudes realizadas por el cliente al servidor con el fin de pedir datos. El servidor al recibir estas solicitudes, las procesa y las envía al cliente. Estas solicitudes se realizan mediante HTTPS.
- **Escalabilidad.** Este modelo permite realizar un balance correcto de la carga de trabajo y distribuir este de forma equitativa entre las diferentes partes.
- **Mantenimiento.** Este modelo, al separar las diferentes funcionalidades, permite el mantenimiento y actualización de una parte sin afectar a la otra.

Por último, para ejemplificar gráficamente el funcionamiento de esta arquitectura, podemos ver el siguiente grafico:

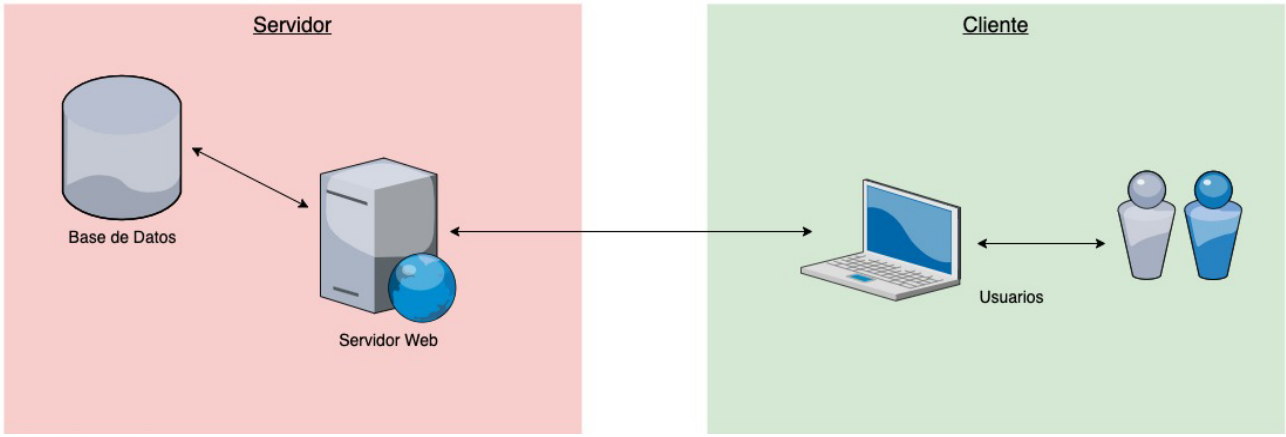

Figura 4. Modelo cliente-servidor

4.2 Diseño de las Bases de datos

En los puntos anteriores hemos comentado de la necesidad de persistir las datos recogiéndolos en una base de datos para su correcto tratamiento, con la finalidad de sacar conclusiones que puedan ayudar al estudio. El éxito de esta parte pasa por saber definir correctamente las diferentes entidades que entran en juego y sus relaciones entre ellas para que cada uno de los modelos tenga unos roles específicos. Con el fin de detallar el proceso de creación de la base de datos, se detallará este de forma incremental. Empezamos por la definición de las diferentes tablas de esta. El proyecto trabaja mediante la utilización de trece tablas, mostradas a continuación junto con su propósito:

- **Tabla users.** Recoge todos los datos introducidos por el usuario en el test de información demográfica.
- **Tabla questions_operas.** Guarda los enunciados de las preguntas del test Operas.
- **Tabla questions_iq.** Guarda las rutas de las imágenes que se mostrarán al usuario y de sus respuestas. Almacena también la respuesta correcta de cada pregunta.
- **Tabla questions_mail.** Guarda la respuesta correcta a cada pregunta del test sobre mail phishing.
- **Tabla answers_operas.** Guarda las respuestas introducidas por los usuarios mediante la realización del test Operas.
- **Tabla answers_iq.** Guarda las respuestas introducidas por los usuarios mediante la realización del test de IQ. Guarda también si esa pregunta es correcta.
- **Tabla answers_mail.** Guarda las respuestas introducidas por los usuarios mediante la realización del test sobre mail phishing.
- **Tabla puntuaciones.** Guarda las puntuaciones totales de cada uno de los test una vez han sido realizados por el usuario.
- **Tablas de apoyo al test demográfico.** Estas tablas han sido creadas con el fin de tener la posibilidad de aumentar el abanico de posibilidades en las diferentes preguntas en caso necesario.
    - **Tabla sexo.** Guarda los sexos a mostrar en el test, hombre o mujer.

- **Tabla color.** Guarda una lista de colores a mostrar durante la realización del test.
- **Tabla lugar_residencia.** Guarda diferentes lugares de residencia para mostrar durante la realización del test.
- **Tabla estudios.** Guarda diferentes opciones de estudios cursados que se muestran en el test.
- **Tabla ocupación.** Guarda diferentes opciones de ocupación para ser mostradas en el test.

Una vez hemos visto las diferentes tablas que se utilizan en el proyecto pasaremos a profundizar más en su implementación. De este modo, se podrá ver con más detalle las diferentes columnas que forman cada tabla, sus claves primarias y sus relaciones, mediante claves foráneas. En el siguiente esquema, se muestra el modelo relacional seguido en la implementación de la base de datos, donde cada línea que une las diferentes tablas representa la relación entre estas mediante el uso de claves primarias y foráneas.

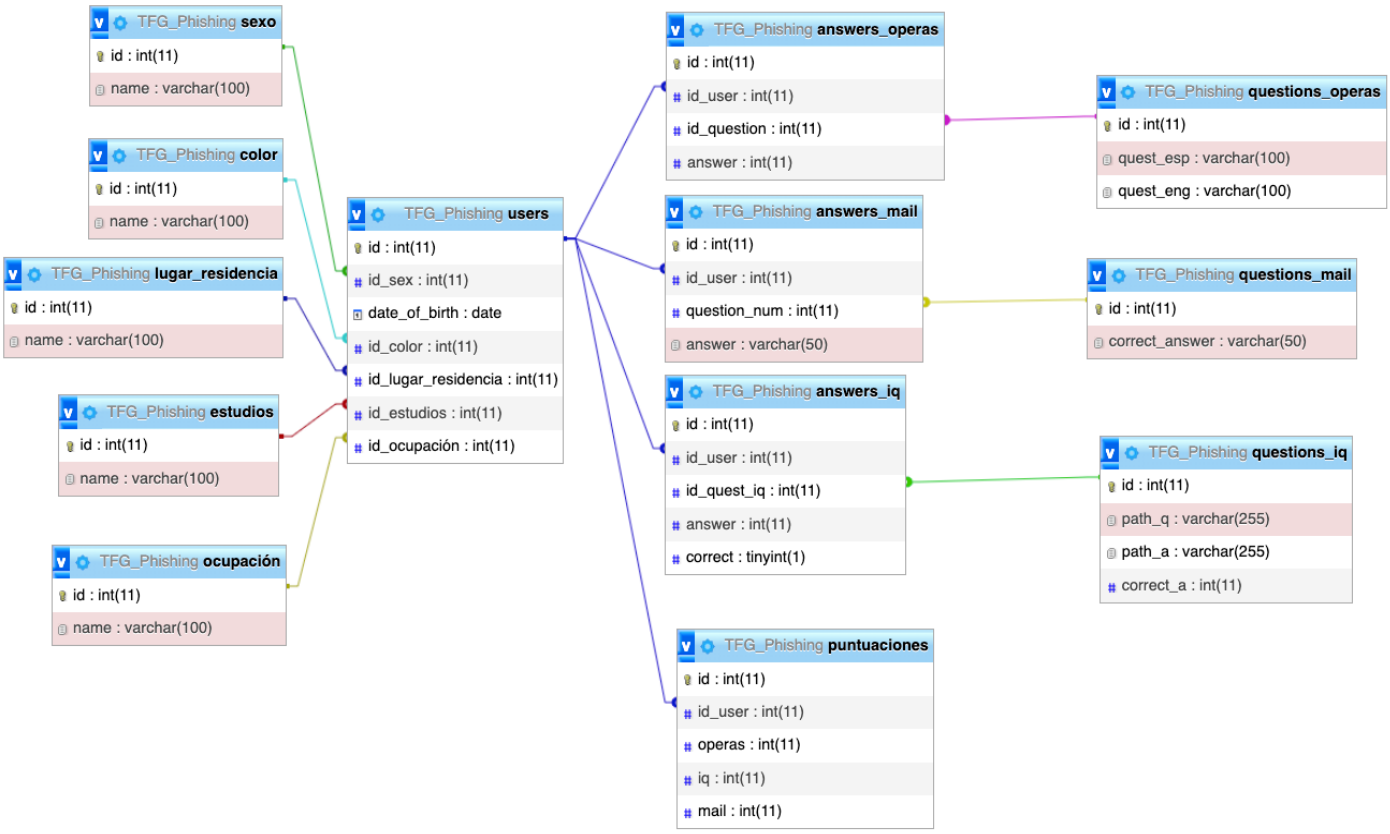

Figura 5. Modelo relacional de la base de datos

4.3 Aplicación Web

En los puntos anteriores, hemos visto que la arquitectura de la página web se basa principalmente en el modelo cliente-servidor. Para conseguir esta arquitectura se implementa el patrón de diseño Modelo-Vista-Controlador (MVC) [11] muy comúnmente utilizado en este tipo de aplicaciones. Este patrón de diseño basa su funcionamiento en tres partes importantes del proyecto web y que veremos con más detalle en los siguientes puntos: el Modelo, la Vista y el Controlador. Las funciones de cada uno de estos roles están correctamente definidas y todos en conjunto permiten aislar las diferentes responsabilidades para asegurar un correcto funcionamiento y mantenibilidad del software.

Modelo	Vista	Controlador
Interacción con base datos	Interacción con el usuario	Puente entre Modelo y Vista

Tabla 1. Interacción de los componentes MVC

Cabe resaltar que un punto muy importante del éxito de este patrón de diseño es como se implementa la comunicación entre los diferentes componentes, como se verá más adelante.

4.3.1 Modelo

El Modelo es la parte que se encarga del manejo de datos y la lógica de negocio. Es por ello que se encargará de realizar las consultas, actualizaciones e inserciones en las diferentes tablas de la base de datos. De este modo, será el único de los roles del MVC capaz de interacción con las diferentes bases de datos.

Como se ha observado en el apartado anterior, son varias las tablas utilizadas en la base datos para el correcto funcionamiento de la aplicación. Es por este motivo que se utilizarán varios modelos dependiendo de la tabla con la que se quiera interactuar. Los modelos utilizados son:

- **PersonalModel.** Este modelo está formado por una clase que tiene como finalidad consultar las diferentes tablas de la base de datos que contienen la información referente a las respuestas del test sobre información demográfica.
- **UserModel.** Es el modelo que se encargará de persistir en la base de datos un perfil de usuario en base a la información obtenida del test sobre información demográfica. Gracias a esto, podemos tener una tabla que guarda las puntuaciones del usuario en los diferentes test.
- **OperasModel.** Como su nombre indica, este modelo se encarga de todas las operaciones que relacionan el test OPERAS con las tablas de la base de datos correspondientes a este. Entre otras funcionalidades, este modelo leerá de la base datos las preguntas a mostrar al usuario para, más tarde, almacenar sus respuestas.

- **IqModel.** Este modelo gestiona las operaciones de consulta, inserción y actualización de la base de datos referentes al test sobre coeficiente intelectual. También se encarga de aplicar la lógica de corrección del test mediante varias funciones.
- **MailModel.** Es el modelo que gestiona las operaciones con la base de datos referentes al test sobre Mail Phishing implementado en la plataforma.

4.3.2 Vista

La vista es la parte del patrón de diseño MVC encargada de presentar al usuario los datos tratados por el modelo y de mostrarlos, mediante una interfaz de usuario, por el navegador. Como cabe esperar, se utilizan varias vistas para mostrar las diferentes páginas que forman la aplicación. Las vistas utilizadas son:

- **Home.** Es el punto de entrada a la aplicación vía un navegador web. Esta vista pretende presentar el proyecto al usuario y darle la opción de realizar los diferentes test. El usuario podrá realizar estos test tanto de modo lineal, como uno por uno.
- **Landing_test.** Es una vista pensada para presentar la información de cada uno de los test. Esta vista se presentará al usuario antes de iniciar cada uno de estos. En ella, se mostrarán los diferentes detalles de cada uno de los test.
- **User_data_view.** Esta vista se encarga de presentar el test para recoger la información demográfica del usuario.
- **Test_view.** Esta vista se encarga de presentar tanto el test OPERAS como el test de IQ al usuario
- **Test_phishing_view.** Esta vista presenta los diferentes emails con alta potencialidad de ser phishing para probar el grado de conocimiento del usuario.
- **Test_phishing_explanation.** Esta vista complementa a la vista anterior. En ella, se muestran los diferentes emails anteriores, pero con desplegables en los puntos susceptibles ofreciendo al usuario una explicación detallada. Esta vista es usada cuando el usuario realiza el test sobre mails en modo entrenamiento.
- **Finish_test_view.** Vista creada para indicar al usuario que ha finalizado los test en la plataforma. Es la encargada de llevar al usuario a la vista home de nuevo.
- **Auxiliares:**
    - **Head.** Archivo utilizado en las diferentes vistas a modo de head común. Se utiliza para los datos que utilizan en común todas las vistas.
    - **Header.** Archivo que añade una barra de opciones en la parte superior del navegador. Permite que el usuario pueda ir rápidamente a la home o a la página de landing de cada uno de los test.
    - **Footer.** Archivo que ofrece la posibilidad de insertar un pie de página común en todas las vistas. Se deja implementado por interés académico aun no siendo utilizado.

4.3.3 Controlador

Una vez revisado el funcionamiento de los dos extremos de este patrón de diseño, entendiendo así como se organiza la parte de comunicación con la base de datos y la parte que se muestra al usuario del navegador es el momento de hablar del Controlador.

Toda la comunicación entre el Modelo y la Vista no es posible sin un nexo entre éstas, el controlador. A modo de controlador, se utiliza un índice que se sitúa entre el servidor y el cliente y recoge las peticiones de ambos lados, las trata y actúa en consecuencia para mantener el flujo de trabajo en la aplicación.

Las Vistas necesitan los datos sobre las diferentes preguntas de los test, estos datos son recogidos por el modelo. Mientras que los Modelos necesitan las respuestas insertadas por el usuario mediante las vistas.

El Controlador realiza toda la lógica asociada a esta comunicación separando las peticiones por secciones, donde cada sección hace referencia al test que está ejecutando. Con el fin de ejemplificar lo mencionado, se presenta como sería la comunicación entre los componentes suponiendo que el usuario está realizando el test Operas:

1- El usuario accede a la sección del test Operas mediante el botón situado en el header destinado para ello.
2- Esa solicitud es recibida por el índice.
3- El índice solicita al modelo, en este caso OperasModel, la primera pregunta correspondiente al test.
4- El modelo realiza la consulta a la base de datos donde se encuentra almacenada la pregunta y la envía al índice.
5- El índice se comunica con la vista, en este caso Test_view, i le envía la correspondiente pregunta.
6- La vista, muestra la pregunta y las diferentes opciones de respuesta mediante un formulario.
7- Cuando el usuario responde, la vista envía la respuesta al índice.
8- El índice, mediante los diferentes parámetros, reconoce de que test se trata y comunica al modelo la respuesta a insertar en la base de datos.
9- El modelo realiza la inserción en la base de datos correspondiente.

Una vez visto el proceso de comunicación entre las diferentes partes, muy similar para todos los test, podemos comprender la necesidad de una correcta implementación de cada parte. Fruto de ello, la importancia de establecer correctamente las responsabilidades de cada parte.

En el siguiente grafico podemos ver como se organizan las diferentes partes de este patrón de diseño aplicado a la página web implementada.

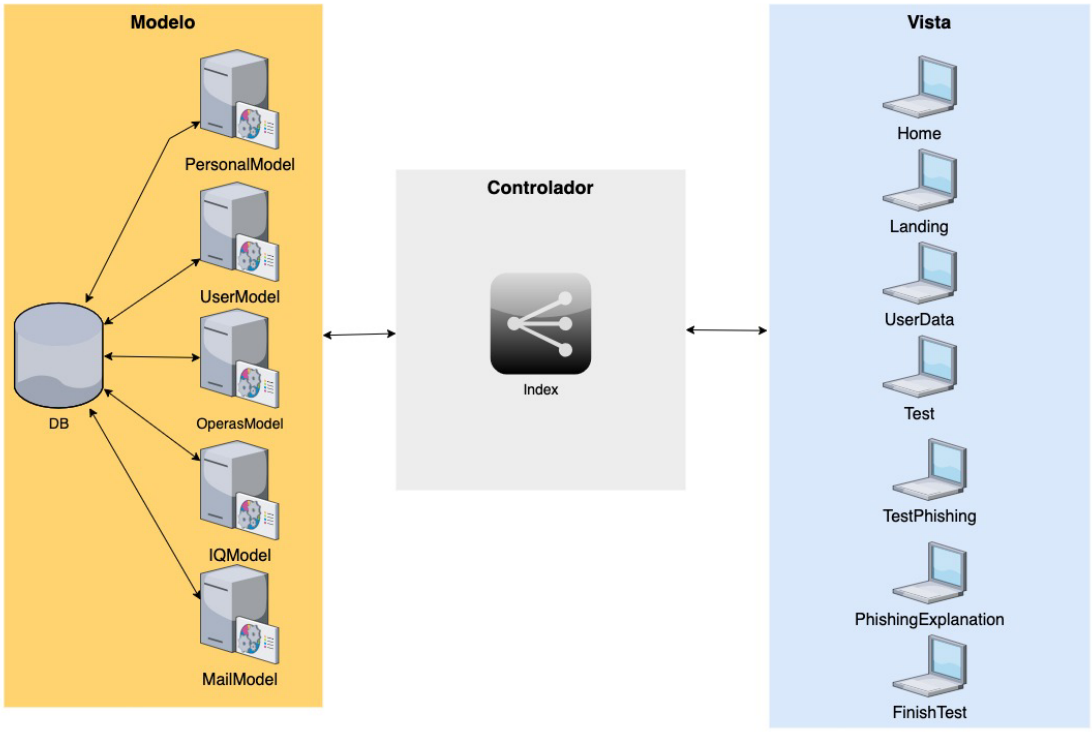

Figura 6. Patrón MVC

5 Implementación

5.1 Lenguajes

En este apartado se describirán las diferentes tecnologías utilizadas en el proyecto con el fin de entender cuál es el rol y funciones de cada una de estas dentro del software presentado. Como se ha visto en los requisitos de la aplicación, dependiendo de la parte del proyecto a implementar se utilizarán diferentes lenguajes y tecnologías.

Para la implementación del lado servidor se utilizará el lenguaje de programación **PHP**, de código abierto y ampliamente utilizado en el desarrollo de páginas web dinámicas. El servidor, como hemos visto, es la parte de nuestra arquitectura que se encargará de las operaciones con la base de datos. Estas operaciones se realizarán mediante **MySQL**, por lo que el lenguaje PHP ofrece soporte para interaccionar con la base de datos.

MySQL es un sistema de gestión de base de datos relacional, muy utilizado también en este tipo de proyectos. El uso de esta tecnología nos permite la creación de bases de datos con el fin de persistir en ellas toda la información necesaria para el correcto funcionamiento del proyecto. Este sistema de gestión de base de datos nos va a permitir definir mecanismos para la protección de la información mediante la definición de permisos para que solo los usuarios que hayan sido autorizados con anterioridad puedan interactuar con la base de datos.

Una vez vistas las tecnologías con las que se va a implementar el lado servidor, veremos las que nos van a ayudar con la implementación del lado cliente.Como es normal en el desarrollo web, la estructura de la página se construirá utilizando el lenguaje de marcado estandarizado HTML. Este lenguaje permite crear la estructura de las diferentes partes de la página web. Es decir, permite crear la estructura que verán los usuarios en su navegador. Por otro lado, el lenguaje de marcado HTML nos va a servir para implementar formularios de recogida de información proporcionada por el usuario. Fruto de esto, podremos enviar la información al backend de la aplicación para ser tratada.

Con el fin de dotar de diferentes estilos y hacer responsive a la página web, se utilizará el lenguaje **CSS** y el framework **Bootstrap**. El lenguaje CSS va a permitir definir los diferentes colores, alineaciones y estilos en general de las diferentes partes en HTML. Con CSS, podremos implementar diferentes estilos en una misma página web mediante el uso de clases e id’s. Por otro lado, se ha comentado que se utilizará el framework BootStrap. Este framework proporciona diferentes estilos mediante la utilización de las reglas CSS ya mencionadas. Aunque nos va a permitir utilizar estilos predefinidos, lo que se pretende con el uso de BootStrap es que la plataforma sea responsiva. Esto pasa por la estructura de BootStrap, que nos permite organizar el contenido a modo de rejillas o columnas que adaptan su tamaño al de la pantalla donde se muestra de forma automática.

Con el fin de añadir dinamismo a la página web y permitir que el usuario interaccione de una forma más provechosa con la plataforma se utilizará el lenguaje de programación **Javascript**. El uso de este lenguaje permite a la plataforma interaccionar con el DOM. Esto hecho va a permitir que se modifiquen elementos HTML en base a diferentes acciones, como podría ser el envío de un formulario. JavaScript va a permitir la creación de funciones incrustadas en el archivo HTML con la finalidad de llevar a cabo todas las operaciones para cumplir con los requisitos de la aplicación.

Relacionado con JavaScript, utilizaremos la técnica de programación **AJAX**. Esta técnica, nos va a permitir enviar y recibir datos desde el frontend al backend y actualizar diferentes partes del HTML sin necesidad de recargar la página por completo. Toda esta funcionalidad se consigue mediante el uso de JavaScript.

Con la finalidad de poder unir todas estas tecnologías para trabajar con ellas conjuntamente en un mismo proyecto se utilizará el entorno de desarrollo web **MAMP**. Es un entorno local, por lo que nos va a permitir no tener que subir la plataforma a un servidor de pago durante su desarrollo. Las siglas MAMP hacen referencia a Macintosh, Apache, MySQL y PHP que son las tecnologías y plataformas que se van a utilizar para la implementación del proyecto. Puntualizar que la versión de MAMP solo sirve para el sistema operativo MacOs dado que la plataforma se construirá utilizando este sistema operativo. El entorno incluye un servidor web Apache, tecnología muy conocida en el desarrollo web. Este servidor va a permitir recibir solicitudes mediante HTTPS desde el navegador web para ser procesadas y respondidas desde el backend. Como hemos podido ver con anterioridad, dispone del sistema de gestión de bases de dados MySQL y un intérprete para el lenguaje PHP. Cuenta con una interfaz de usuario amigable que nos va a permitir definir diferentes configuraciones para el proyecto, como podrían ser la versión PHP utilizada, los puertos a escuchar, entre otros. También cuenta con registros de log que van a permitir identificar correctamente los errores surgidos mediante la implementación con el fin de investigarlos y corregirlos.

A modo de conclusión, este entorno de desarrollo ha sido elegido por contar con todo lo necesario para desarrollar una web de estas características en un entorno local.

5.2 Estructura del proyecto

En los apartados anteriores hemos podido ver varias características del proyecto, desde los requerimientos a las diferentes tecnologías utilizadas. Una vez visto lo anterior, pasaremos a estudiar cómo es la estructura del proyecto en base a las funcionalidades que queremos implementar.

A continuación, se detalla cómo es la estructura de directorios del proyecto:

- app
    - models
        - iq_model.php
        - mail_model.php
        - operas_model.php
        - personal_model.php
        - user_model.php
    - views
        - finish_test_view.php
        - footer.html
        - head.html
        - header.html
        - home.html
        - landing_test.php
        - test_phishing_explanation.php
        - test_phishing_view.php

        - test_view.php
        - user_data_view.php
- assets
    - bootstrap3
        - css
            - bootstrap.css
        - fonts
        - js
    - css
        - ct-paper.css
    - icons
    - images
    - images_iq
    - js
        - send-form.js
    - sass
- config
    - config.php
    - database.php
- index.php

En el punto 4.3 vimos cómo se organizaban los modelos y las vistas que en este punto ya podemos ubicar en la estructura de directorios. Como ya se han explicado las diferentes funciones de estos, en este punto no se detallarán.

En la estructura mostrada anteriormente solo se han mostrado los ficheros más importantes con el fin de simplificar la explicación. Principalmente las funciones de cada uno de ellos son:

- **Bootstrap.css.** Este fichero declara las reglas CSS necesarias para la utilización del framework Bootstrap.
- **Ct-papper.css.** Es el fichero que declara las reglas CSS utilizadas por la plantilla escogida para el desarrollo del frontend.
- **Send-form.js.** Es el fichero con código Javascript que permite realizar envíos mediante el método POST HTTPS entre el frontend y el backend. Permite también no mostrar las variables necesarias para la comunicación en la URL del navegador.
- **Config.php.** Fichero utilizado para la definición de los paths más utilizados en el desarrollo del código. Permite aislar la declaración de estos y evitar así repetir código.
- **Database.php.** Es el fichero encargado de la conexión con la base de datos. Cada fichero que necesita interactuar con la base de datos pide a database.php la instancia de la conexión con esta.
- **Index.php.** Como se ha visto con anterioridad, es el Controlador del patrón de diseño MVC.

5.3 Algoritmos específicos

Durante la implementación del proyecto web han sido varios los algoritmos implementados. En este punto se intentará dar al lector una idea general del desarrollo de los algoritmos más importantes para el proyecto.

Los algoritmos que se mostrarán se realizarán mediante pseudocódigo y pretenden únicamente mostrar a vista de pájaro como se implementan las diferentes funcionalidades. No se mostrarán fragmentos de código en este apartado.

5.3.1 Operas

En esta sección se verá el comportamiento de la parte del servidor que permite al usuario realizar el test Operas, en el apartado 5.3.4 se podrá como se envían los datos desde el cliente, siendo esta implementación similar para todas las vistas:

- En index.php:

```
seccion = leerPost(seccion);
accion = leerPost(acción);
si (sección == 'operas'){
 modeloOperas = new OperasModel();
 switch (accion):
       caso 'inicia_test':
             pregunta = modeloOperas.getQuestion(1);
             //Función que envía la pregunta a mostrar a la vista
             muestraPregunta(pregunta, operas);

       caso 'inserta_pregunta':
             pregunta = leerPost(numero_pregunta);
             respuesta = leerPost(respuesta);
             modeloOperas.insertaRespuesta(pregunta, respuesta);

             //Actualizamos número de pregunta a la siguiente
             pregunta ++;
             preguntas_total = modeloOperas.getNumeroPreguntas();

             si (pregunta <= preguntas_total){
                   pregunta = modeloOperas.getQuestion(pregunta);
                   muestraPregunta(pregunta, operas);
             }
             sino{
                   modeloOperas.calculaPuntuacion();
                   muestraTestFinalizado();
             }
 }
```

- En OperasModel, funciones llamadas:

```
getQuestion(numero_pregunta){
       conexion = conectaBD();
       //Consulta el numero de pregunta en la base de datos
       pregunta = conexion.getPregunta(numero_pregunta);
       cierraConexion(conexion);
       retorna pregunta;
```

```
}

insertaRespuesta(pregunta, respuesta){
        conexion = conectaBD();
        //leemos el user_id de la variable de sesión
        id = leerSesion(id_usuario);
        si (conexión.inserta(pregunta, respuesta, id)){
              cierraConexion(conexion);
              retorna true;
        }
        sino{
              cierraConexion(conexion);
              retorna false;
        }
}

calculaPuntuacion(){
        conexion = conectaBD();
        //leemos el user_id de la variable de sesión
        id = leerSesion(id_usuario);

        puntuación = 0;
        para (i = 0; i < num_preguntas; i++){
              puntuación = puntuación + conexión.consultaPuntuacion(i,
id);
        }

        conexion.guardaPuntuacionTotal(puntuación, id);
        cierraConexion(conexion);
}
```

5.3.2 IQ

Para el test de IQ el modo de funcionamiento es idéntico a la operativa del test Operas a excepción de la corrección de los test. El test IQ cuenta con un algoritmo de corrección distinto en el que hay que seguir determinadas pautas para identificar los errores y las posibles salidas del test.

En el siguiente pseudocódigo podemos ver la implementación del método calculaPuntuacion en el IQModel:

```
calculaPuntuacion(pregunta, respuesta){
        conexion = conectaBD();

        si (pregunta >= 1 o pregunta <= 3){
              conexión.insertaIQ(pregunta,    respuesta,    esCorrecta
(pregunta, respuesta));
              si (ordenDescendente){
                    compruevaAnterior(pregunta);*
              }
              sino{
                    muestraSiquiente(pregunta++);
              }
        }
        sino{
              si (pregunta == 4 o Pregunta ==5){
```

```
                        conexión.insertaIQ(pregunta, respuesta, esCorrecta
(pregunta, respuesta));
                        si (!esCorrecta(pregunta, respuesta)){
                              pregunta--;
                              iniciaOrdenDescendente();*
                        }
                  }
                  sino {
                        //Preguntas con id superior a 5
                        conexión.insertaIQ(pregunta, respuesta, esCorrecta
(pregunta, respuesta));
                        seguir = compruevaCondiciones();*
                        si(seguir){
                              muestraSiquiente(pregunta++);
                        }
                        Sino{
                              finalizaTest();
                        }
                  }
            }

            cierraConexion(conexion);

      }
```

Este pseudocódigo ha sido desarrollado para ejemplificar el complejo algoritmo de corrección del test IQ. Con el fin de ayudar al lector a entender esta parte de la corrección, a continuación, vemos las condiciones de corrección:

- Se muestran en orden normal las preguntas 1, 2 y 3.
- Si se fallan las preguntas 4 o 5, se muestran las preguntas anteriores. Cuando se acierten dos seguidas se dan los números anteriores como correctos y se continua el test. Por ejemplo, el usuario falla la numero 5, se muestran las preguntas 4 y 3 y este las acierta. Se le darán por correctas las preguntas 2 y 1 y, acto seguido, se mostrará la pregunta 6.
- Si estamos mostrando en orden descendiente y no se aciertan un mínimo de dos preguntas seguidas, se acaba el test.
- Para las preguntas con índice superior a 5, las condiciones de finalización son las siguientes:
    - Cuatro fallos seguidos.
    - Cuatro fallos en cinco preguntas.

Por otro lado, en el pseudocódigo podemos ver que hay tres funciones marcadas con el símbolo *. A continuación, se detalla que hacen estas funciones:

- La función compruevaAnterior() se encarga de comprobar si la pregunta mostrada anteriormente a la actual (índice de pregunta ++) es correcta. En caso de serlo y que la respuesta actual también lo sea, muestra la pregunta siguiente a la fallada en primera instancia, que serían las preguntas 4 o 5.
- La función iniciaOrdenDescendete() se encarga de comenzar a mostrar las preguntas anteriores a la 4 o 5 en caso de haber fallado alguna de estas.
- La función compruevaCondiciones() se encarga de evaluar los resultados de las cuatro o cinco preguntas anteriores para detectar posibles puntos de finalización del test.

5.3.3 Mail Phishing

Del mismo modo que en los test Operas e IQ, el test de Mail Phishing realiza una comunicación con el servidor una vez que el usuario inserta su respuesta. El servidor almacena esta respuesta en la base de datos y envía al cliente la siguiente pregunta a mostrar.

Para este test, existe una pequeña diferencia que marca el comportamiento que tendrá. Esta diferencia es las opciones de realización que se proponen, modo experimento o modo entrenamiento.

Ya se ha podido ver que el modo entrenamiento, antes de mostrar la siguiente pregunta, enseña al usuario detalles que pueden ayudar a identificar un posible phishing. Esto se consigue gracias al siguiente fragmento de código ubicado en el fichero index.php:

```
sección = leerPost(seccion);
accion = leerPost(acción);
si (sección == 'mail'){
 modeloMail = new OperasMail();
 switch (accion){
       caso 'inserta_pregunta':
             pregunta = leerPost(numero_pregunta);
             respuesta = leerPost(respuesta);
             modo = leerPost(modo);

             modeloMail.insertaRespuesta(pregunta, respuesta);

             si (modo == 'entrenamiento'){
                   muestraExplicacion(pregunta);
             }
             sino{
                   //Actualizamos número de pregunta a la siguiente
             pregunta ++;
             preguntas_total = modeloOperas.getNumeroPreguntas();

             si(pregunta <= preguntas_total){
                    muestraPregunta(pregunta, mail);
             }
             sino {
                   modeloMail.calculaPuntuacion();
                   muestraTestFinalizado();
             }
 }
```

A modo de ejemplo, en la siguiente imagen podemos observar cómo se mostrarían las explicaciones cuando se realiza el test en modo entrenamiento.

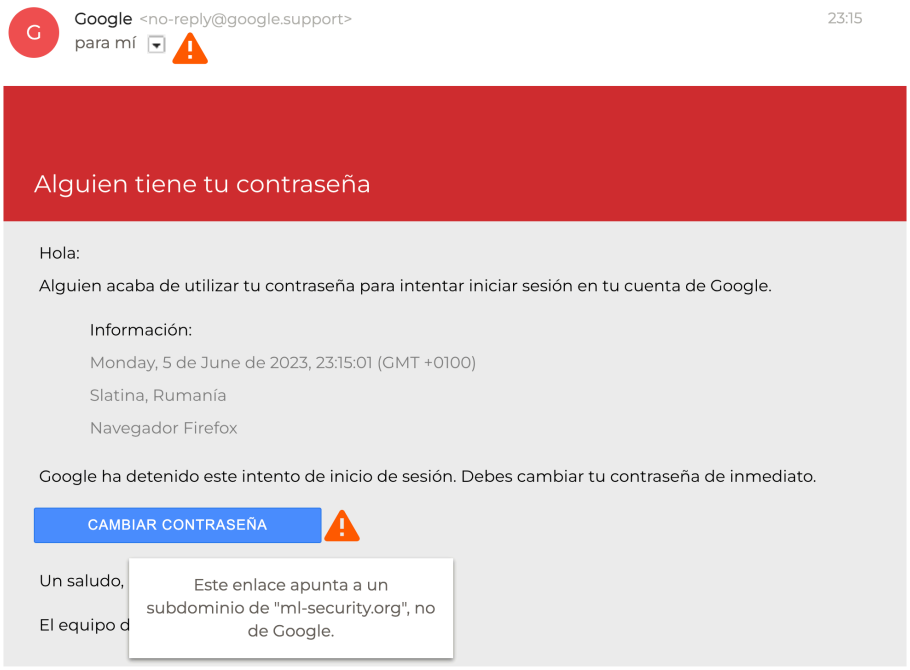

Figura 7. Test sobre mail phishing en modo entrenamiento

5.3.4 Envío de datos

Esta parte de la explicación de algoritmos hace referencia a la forma en la que se envían los datos desde el cliente al servidor. Esta función recoge los datos de un formulario al hacer submit, los trata y los envía mediante método POST al servidor.

Se ha decidido implementarlo de esta forma con el fin de aumentar la seguridad de la página web escondiendo las variables de la URL de modo que desde la barra de navegación no se puedan realizar accesos indeseados por parte del usuario.

El siguiente pseudocódigo muestra el código que se ejecuta cuando un usuario envía el formulario correspondiente al test que esté realizando:

```
datos = leerFormulario();
petición = creaPeticionHTTPS();
petición.inicializa(POST, 'index.php');
petición.enviar(datos);
```

Como se puede observar el proceso es sencillo, consiste básicamente en esperar un evento como un envío de formulario, leer los datos, crear una petición y realizar la comunicación con el servidor.

6 Juego de pruebas

Para asegurar el correcto funcionamiento de la plataforma y que se cumplen los requisitos propuestos al inicio del proyecto se han realizado diferentes pruebas reflejadas en la siguiente tabla:

Juego de pruebas	Resultado esperado	Resultado obtenido
En la home, seleccionar el botón Iniciar Test.	Se envía al usuario a la página de inicio del test sobre información demográfica.	Ok
En la home, seleccionar el botón del header Operas.	Se muestra la página de inicio del test Operas.	Ok
En la home, seleccionar el botón del header IQ.	Se muestra la página de inicio del test IQ.	Ok
En la home, seleccionar el botón del header Phishing.	Se muestra la página de inicio del test Phishing.	Ok
En cualquier otra vista, seleccionar el botón del header Inicio.	Se muestra la página de inicio.	Ok
En cualquier otra vista, seleccionar el botón del header home.	Se muestra la página de inicio.	Ok
En la vista de inicio del test de información demográfica, seleccionar botón iniciar.	Se muestra el test de información demográfica.	Ok
En la vista de inicio del test de Operas, seleccionar botón iniciar.	Se inicia el test de Operas.	Ok
En la vista de inicio del test de IQ, seleccionar botón iniciar.	Se inicia el test de cálculo del coeficiente intelectual IQ.	Ok
En la vista de inicio del test sobre Mail Phishing, seleccionar el Modo Entrenamiento.	Se inicia el test en Modo Entrenamiento.	Ok
En la vista de inicio del test sobre Mail Phishing, seleccionar el Modo Experimento.	Se inicia el test en Modo Experimento.	Ok
En el test de información demográfica se presiona el botón Enviar sin respuestas seleccionades.	La plataforma muestra un mensaje advirtiendo de que se debe seleccionar una respuesta.	Ok
En el test de información demográfica se presiona el botón Enviar con respuestas seleccionades.	Se muestra la página de inicio del test Operas.	Ok

En el test de información demográfica se presiona el botón Enviar con respuestas seleccionades.	Se guardan las respuestas del usuario en la tabla Users de la base de datos.	Ok
En el test de Operas se presiona el botón Enviar sin respuestas seleccionades.	La plataforma muestra un mensaje advirtiendo de que se debe seleccionar una respuesta.	Ok
En el test de Operas se presiona el botón Enviar con respuestas seleccionades sin estar terminado el test.	Se muestra la siguiente pregunta del test.	Ok
En el test de Operas se presiona el botón Enviar con respuestas seleccionades sin estar terminado el test.	Se guardan las respuestas en la tabla operas_answers.	Ok
En el test de Operas se presiona el botón Enviar con respuestas seleccionades siendo la última pregunta del test.	Se muestra la página de inicio del test IQ.	Ok
En el test de IQ se presiona el botón Enviar sin respuestas seleccionades.	La plataforma muestra un mensaje advirtiendo de que se debe seleccionar una respuesta.	Ok
En el test de IQ se presiona el botón Enviar con respuestas seleccionades sin estar terminado el test.	Se muestra la siguiente pregunta del test.	Ok
En el test de Operas se presiona el botón Enviar con respuestas seleccionades sin estar terminado el test.	Se guardan las respuestas en la tabla iq_answers.	Ok
En el test de IQ se presiona el botón Enviar con respuestas seleccionades siendo la última pregunta del test.	Se muestra la página de inicio del test Mail Phishing.	Ok
En el test de IQ se presiona el botón Enviar con respuestas seleccionades siendo la última pregunta del test.	Se realiza el cálculo de la puntuación total y se guarda en la tabla puntuaciones.	Ok
En el test de IQ se fallan las preguntas 1, 2 y 3 y se aciertan la 4 y 5.	En la tabla answers_iq se actualizan las respuestas 1, 2 y 3 a correctas.	Ok
En el test de IQ se falla la pregunta 4.	Se muestran las preguntas anteriores en orden descendente.	Ok
En el test de IQ se falla la pregunta 5.	Se muestran las preguntas anteriores en orden descendente.	Ok

En el test de IQ, cuando se muestran las preguntas en orden descendente se responden 2 correctamente.	Se actualizan las preguntas anteriores a correctas.	Ok
En el test de IQ, cuando se muestran las preguntas en orden descendente fallan todas.	Se termina el test.	Ok
En el test de IQ, a partir de la pregunta número 5, se fallan 4 preguntas seguidas.	Se termina el test.	Ok
En el test de IQ, a partir de la pregunta número 5, se fallan 4 preguntas sobre 5.	Se termina el test.	Ok
En el test de IQ, se termina el test.	Se muestra la pantalla de inicio del test Mail Phishing.	Ok
En la pantalla de inicio del test Mail Phishing se selecciona el botón Entrenamiento.	Inicia el test en Modo Entrenamiento.	Ok
En la pantalla de inicio del test Mail Phishing se selecciona el botón Experimento.	Inicia el test en Modo Experimento.	Ok
En el test Mail Phishing en modo Entrenamiento se selecciona una respuesta.	Se muestra la explicación de la pregunta tipo Mail Phishing Propuesta.	Ok
En el test Mail Phishing en modo Entrenamiento se selecciona una respuesta.	Se guarda la respuesta en la tabla de la base de datos mail_answers.	Ok
En el test Mail Phishing en modo Entrenamiento y en la sección de explicación, se pasa el ratón por encima del icono rojo.	Se abre un desplegable con la explicación.	Ok
En el test Mail Phishing en modo Experimento se selecciona una respuesta.	Se guarda la respuesta en la tabla de la base de datos mail_answers.	Ok
En el test Mail Phishing en modo Experimento se selecciona una respuesta.	Se muestra la siguiente pregunta.	Ok
En el test Mail Phishing en modo Entrenamiento se selecciona una respuesta siendo la anterior la última.	Se muestra una pantalla de finalización del test.	Ok
En el test Mail Phishing en modo Experimento se selecciona una respuesta siendo la anterior la última.	Se muestra una pantalla de finalización del test.	Ok

En el test Mail Phishing en modo Entrenamiento se selecciona una respuesta siendo la anterior la última.	Se guarda la puntuación total en la tabla puntuaciones.	Ok
En el test Mail Phishing en modo Experimento se selecciona una respuesta siendo la anterior la última.	Se guarda la puntuación total en la tabla puntuaciones.	Ok
En el test Operas, se intenta cambiar el número de pregunta mediante la URL.	No se puede efectuar dicha acción.	Ok
En el test IQ, se intenta cambiar el número de pregunta mediante la URL.	No se puede efectuar dicha acción.	Ok
En el test de Mail Phishing, se intenta cambiar el número de pregunta mediante la URL.	No se puede efectuar dicha acción.	Ok

Tabla 2. Juegos de pruebas

7 Conclusiones

Este trabajo pretendía crear una herramienta que permita relacionar los rasgos característicos de una persona y la probabilidad de caer en un ataque de phishing. Todo esto, se pretendía realizar desde un punto de partida en el que todavía no hay una amplia gama de información relacionada con el tema.

Por otro lado, y no menos importante, se pretendía que sirviese de consolidación de los aspectos técnicos adquiridos durante mis estudios de Ingeniería Informática.

De todos los objetivos detallados en la introducción, el que más relevancia puede tener desde el punto de vista de investigación y que deja una puerta abierta al futuro del estudio de la lucha contra el phishing es el de crear una herramienta de evaluación de los usuarios.

En este trabajo, se ha desarrollado dicha plataforma con éxito, teniendo ahora un punto de partida que pueda ayudar en la estandarización de criterios a la hora de relacionar personalidad y victimización por phishing.

Por otro lado, se ha integrado en este proyecto de web la posibilidad de realizar un entrenamiento al usuario sobre posibles phishing basados en casos reales y ofrecidos por la herramienta de Google.

A nivel técnico, como se ha comentado, se ha conseguido consolidar los conocimientos adquiridos previamente durante mis estudios. Por tanto, se podría decir que este proyecto ha servido también para consolidar las bases de la creación de páginas web tanto a nivel de backend como de frontend, utilizando herramientas comunes en el mercado y enfocando el diseño a webs responsivas que puedan ser ejecutadas en cualquier dispositivo. Cabe resaltar, que aunque lo ideal hubiese sido la implementación de pruebas unitarias de software, se han probado a conciencia todos los posibles casos de error en la plataforma, todos ellos superados con éxito.

La herramienta creada va a ser muy útil para futuros estudios que se están realizando en nuestra universidad y para demostrar las implicaciones de la psicología humana en este ámbito de la informática. El futuro de esta investigación está encaminado a seguir estudiando las características únicas de cada usuario y, una vez estudiadas, sacar conclusiones para poder ser aplicadas en el campo de la lucha contra el phishing. Por tanto, con el desarrollo de esta plataforma se ha avanzado un paso más en el estudio e investigación de la seguridad en internet, pudiendo servir esto tanto para usuarios rutinarios como para empleados de compañías.

Destacar, también, que con la investigación en este ámbito se han podido observar limitaciones que pueden llegar a entorpecer el estudio, como podría ser las diferencias que pueden presentar los rasgos personales de una persona dependiendo de si actúan en internet o en una situación cotidiana fuera de él. Por tanto, el futuro de está investigación también debería pasar por identificar este gap y actuar en base a ello.

Por último, como reflexión personal al trabajo realizado, se ha podido observar de primera mano el avance que se está realizando en el sector de la ciberseguridad y la protección de las personas en internet. Son muchas las personas que dedican su día a día a la investigación y que ofrecen estudios de calidad. Poco a poco, todo este trabajo culminará en una estandarización que ayudará de forma significativa al usuario.

Personalmente, he conseguido abrir el espectro en lo referente a la investigación. He podido observar de primera mano la importancia de ser rigurosos y actuar siempre en base a

estudios y el trabajo que ofrecen otras personas fruto de su investigación. Por lo que en el aspecto académico creo que he podido dar un paso más allá y ver la investigación como una herramienta más que necesaria en el mundo actual.

8 Referencias

[1] Anti Phishing Working Group, "Phishing Activity Trends Report 3rd Quarter 2022", p. 1-7, 2022.

[2] Wikipedia, "2600: The Hacker Quarterly", 2015, Recuperado de https://es.wikipedia.org/wiki/2600:_The_Hacker_Quarterly

[3] Edgar Jair Sandoval Castellanos, "Ingeniería Social: Corrompiendo la mente humana", Revista de Seguridad UNAM vol 10, 2011, Recuperado de https://revista.seguridad.unam.mx/category/revistas/numero-10

[4] Pablo Lopez-Aguilar and Agustí Solanas, "Human Susceptibility to Phishing Attacks Based on Personality Traits: The Role of Neuroticism", *2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC)*, Madrid, Spain, pp. 1363-1368, 2021.

[5] Rodolfo, "Ocho tipos de ataques phishing que ponen en riesgo tu seguridad", 2019, Recuperado de https://www.muyseguridad.net/2019/02/15/ataques-phishing-riesgo-seguridad/

[6] Andreu Vigil-Colet, "Development and validation of the overall personality assessment scale (OPERAS)". Psicothema, vol. 25, no 1, p. 100-106, 2013.

[7] Consejo General de la Psicología, "Evaluación del cuestionario WAIS-IV", p. 3 – 4, 2014.

[8] Jigsaw and Google, "Phishing Quiz", Recuperado de https://phishingquiz.withgoogle.com/

[9] Pablo Lopez- Aguilar, Constantinos Patsakis and Agustí Solanas, "The Role of Extraversion in Phishing Vitimisation: A Systematic Literature Review". In Proceedings of the Symposium on Electronic Crime Research (eCrime), p. 1-10, 2022.

[10] James L. Parrish, Janet L. Bailey and James F.Courtney, "A Personality Based Model for Determining Susceptibility to Phishing Attacks", Little Rock: UoA, pp. 285-296, 2009.

[11] Mdn web docs ,"MVC", 2022, Recuperado de https://developer.mozilla.org/es/docs/Glossary/MVC

Annexos

Annexo 1. Instalación y puesta en marcha

Este tutorial es para un entorno MacOs, ya que es donde se ha desarrollado la práctica. Para otros sistemas operativos, como Windows, los pasos a seguir son bastante similares.

Por tanto, con el fin de instalar todo el entorno necesario para la ejecución del software se recomienda seguir los siguientes pasos.:

1- Descarga MAMP de su sitio web oficial: https://www.mamp.info/en/downloads/
2- Ejecuta el instalador e instala la aplicación.
3- Abre MAMP y configura el entorno con las siguientes características:
   a. En Preferencias -> Puertos: Verificar que Apache escucha el puerto 8888 y que MySQL escucha el puerto 8889.

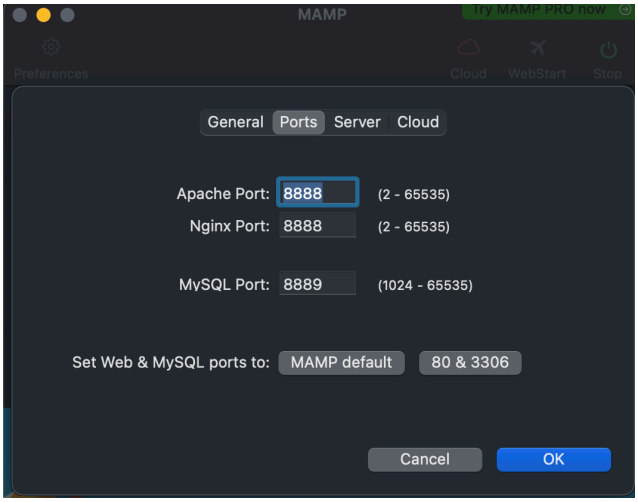

Figura 8. Puertos de escucha MAMP

   b. En Preferencias -> Servidor: Seleccionar la ruta /Applications/MAMP/htdocs como ruta para colocar el código de la página web.

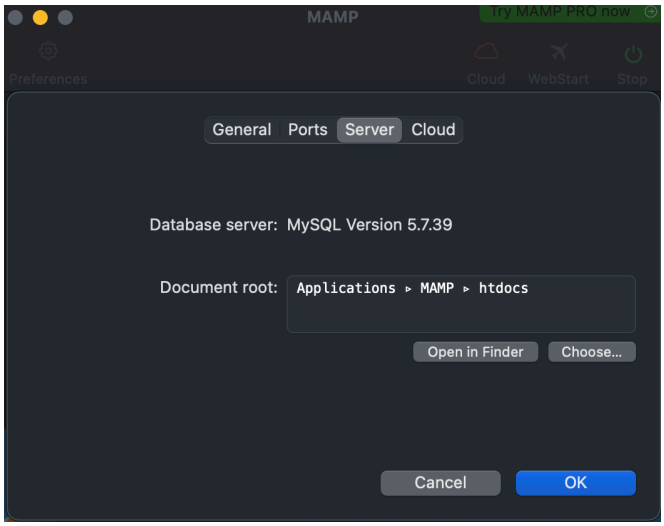

Figura 9. Configuración ruta de la aplicación

4- Descargar el código de la web del siguiente repositorio en GitHub: https://github.com/sergioarevro/TFG.git
5- Colocar la carpeta TFG en el directorio htdocs.
6- Colocar el fichero TFG_Phishing.sql en el directorio /Applications/MAMP/db.
7- Iniciar el servidor con el botón Start en la parte superior derecha de la ventana.
8- Cuando en tu navegador se abra la ventana principal, seleccionar en el apartado Tools la herramienta phpMyAdmin. Se abrirá el gestor de bases de datos.

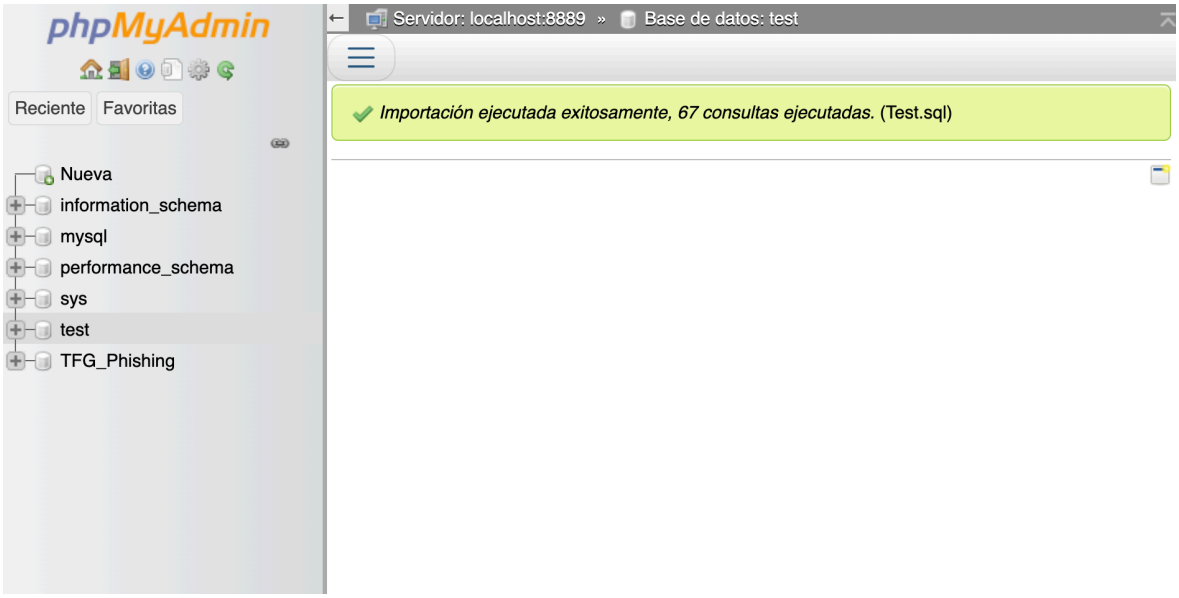

Figura 10. Gestor base de datos

9- En el gestor crear una nueva base de datos.

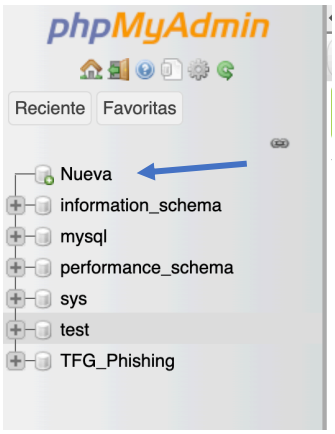

Figura 11. Creación base de datos

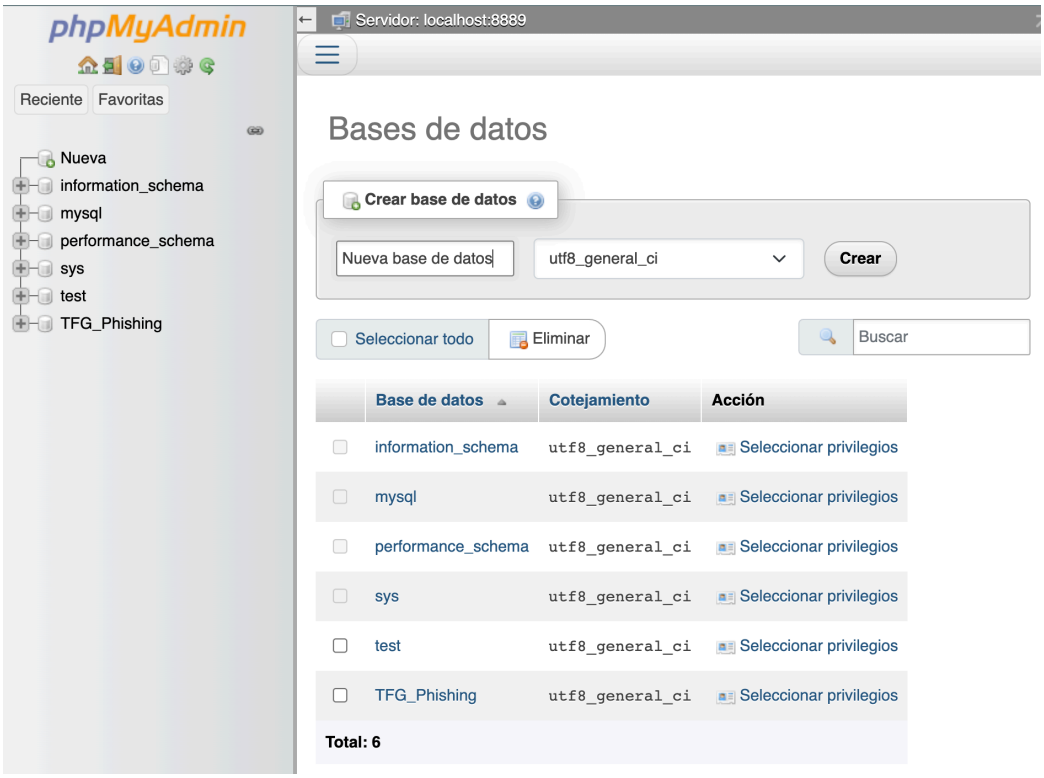

Figura 12. Creación base de datos 2

10- Importar la base de datos que hemos ubicado en /Applications/MAMP/db. Seleccionando la ruta y dejando el resto de opciones por defecto:

Figura 13. Importar base de datos

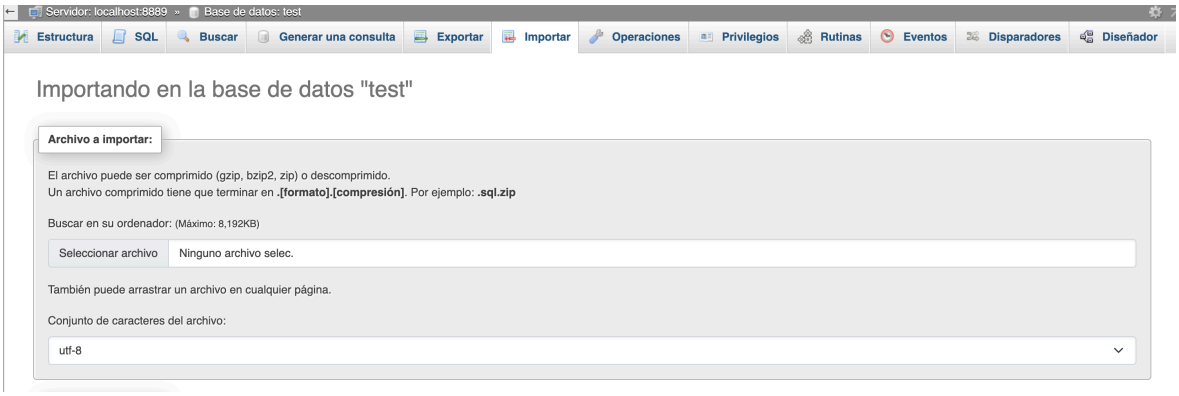

Figura 14. Selección de ruta en base de datos

11- Abrir una nueva ventana en el navegador e introducir la siguiente dirección para ejecutar la página web: http://localhost:8888/TFG/:

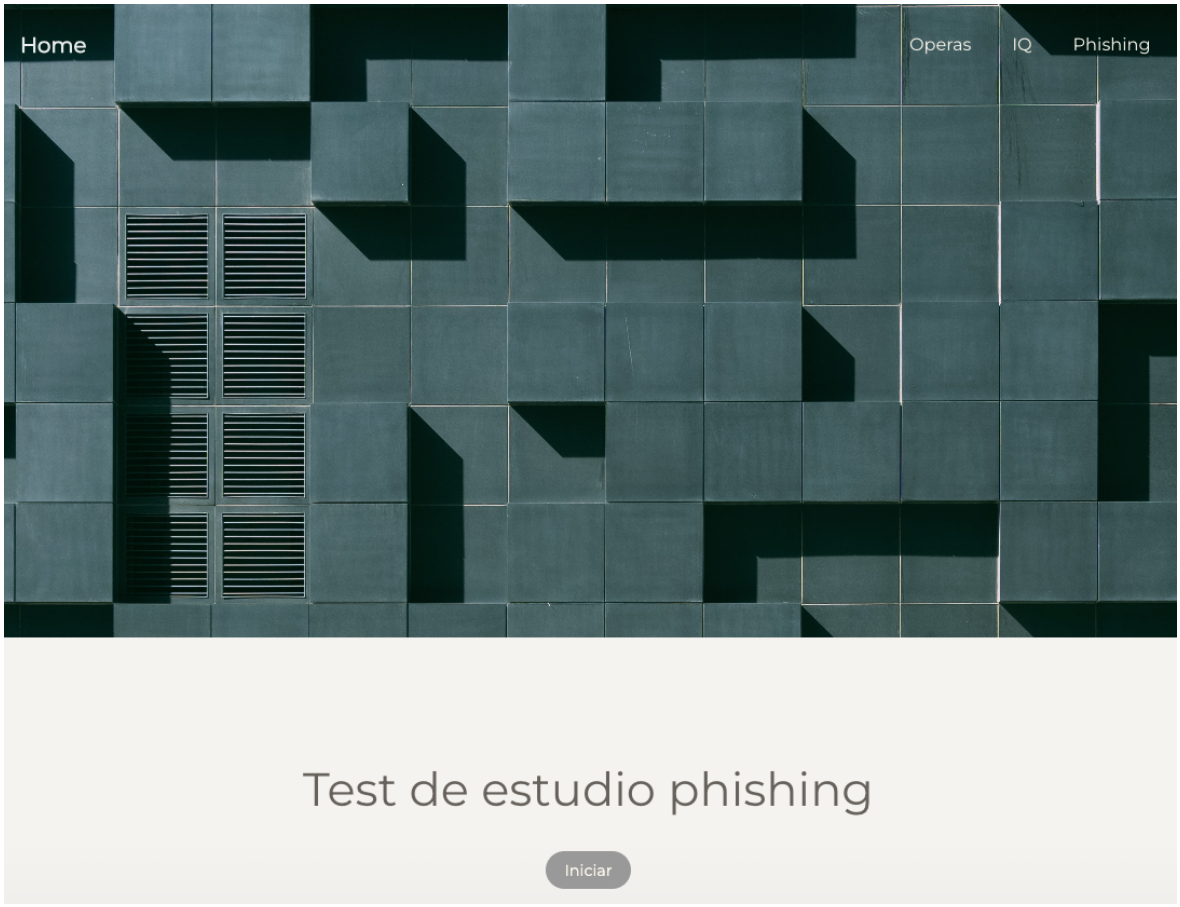

Figura 15. Pagina home de la web

Annexo 2. Manual de usuario

Este anexo pretende dar al usuario unas instrucciones básicas para la utilización de la plataforma.

Una vez se ha instalado todo el entorno y la plataforma está operativa, se pueden realizar las siguientes operaciones:

- **Realizar el test en modo lineal.**
    1. Acceder la home de la web y hacer click en el botón Iniciar del apartado Test de estudio phishing.

Figura 16. Inicio test en modo lineal

    2. Realizar los test, automáticamente al terminar cada uno de ellos se redirigirá a la página de inicio del siguiente.

- **Realizar el test demográfico.** Para realizar este test, iniciar el test en modo lineal, se mostrará este test primero.

- **Realizar el test Operas.**
    1. Acceder mediante la barra superior al test te Operas haciendo click en el botón Operas. Se redirigirá a la página de inicio de este test.

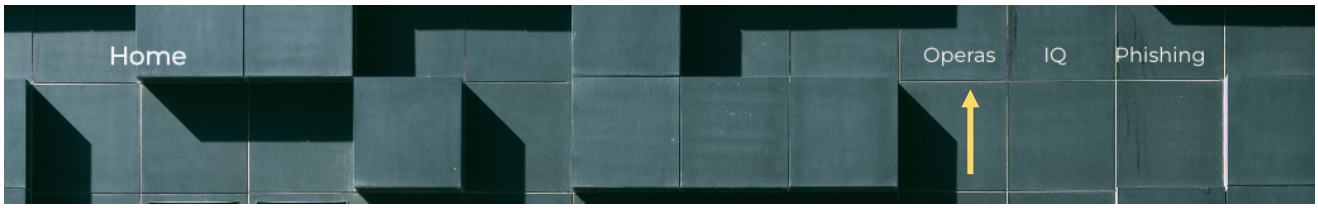

Figura 17. Acceso test Operas individual

    2. Acto seguido se mostrará la página de inicio de este Test. Hacer click en iniciar para comenzarlo.

- **Realizar el test IQ.**

    1. Acceder mediante la barra superior al test de IQ haciendo click en el botón IQ. Se redirigirá a la página de inicio de este test.

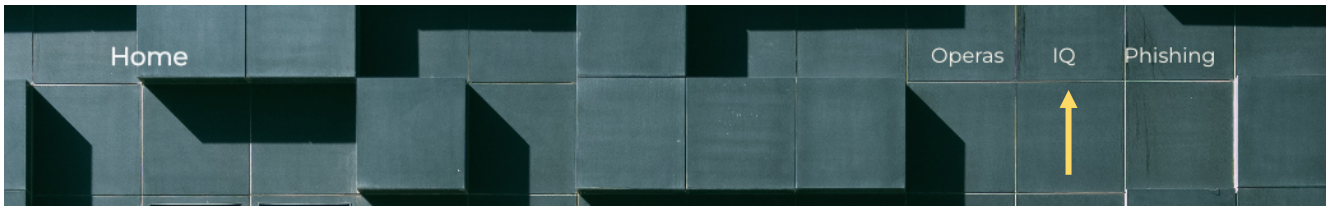

Figura 18. Inicio test de Google

Figura 19. Acceso test IQ individual

2. Acto seguido se mostrará la página de inicio de este Test. Haz click en iniciar para comenzarlo.

- **Realizar el test de Mail Phishing de Google.**

1. Acceder mediante la barra superior al test de mail Phishing haciendo click en el botón IQ. Se redirigirá a la página de inicio de este test.

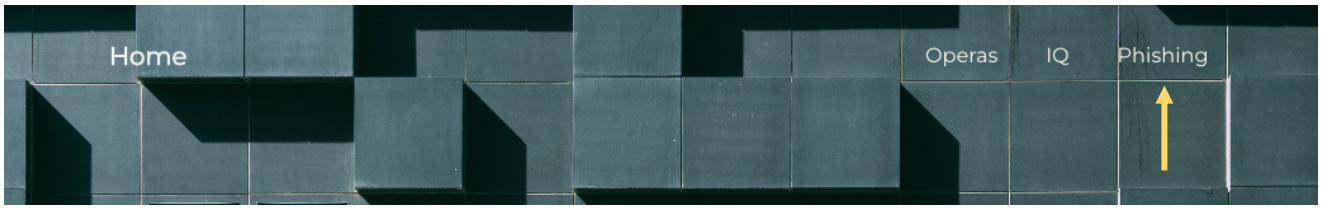

Figura 20. Acceso test Mail Phishing de Google individual

2. Escoger la opción modo entrenamiento para realizar el test obteniendo explicaciones, o de lo contrario, modo experimento para no obtener estas explicaciones.

Figura 21. Inicio test de Google