

UNIVERSIDAD DE LA EMPRESA

FACULTAD DE INGENIERÍA

**TÉCNICO EN INFORMÁTICA Y ANALISTA EN TECNOLOGÍA
INFORMÁTICA**

PROYECTO INTEGRADOR

**Herramienta de entrenamiento y capacitación para evitar fraudes a través de
Ingeniería Social.**

Integrantes:

Diego Balbis — C.I. 4.451.623-5

Dayron Muñoz — C.I. 6.295.606-3

Tutor:

Ing. Pablo Martres



CAPFIS
CIBERSEGURIDAD EFICIENTE

Introducción y Descripción del Problema

En la actualidad, los ciberataques basados en ingeniería social representan una de las amenazas más comunes y subestimadas dentro del ámbito digital.

A diferencia de los ataques puramente técnicos, que buscan vulnerar directamente los sistemas informáticos, este tipo de ataque se enfoca en el factor humano, explotando la confianza, la curiosidad o el miedo de las personas para obtener información sensible o acceso no autorizado.

Pese a los importantes recursos que las organizaciones destinan al fortalecimiento de la infraestructura tecnológica, la capacitación del usuario sigue siendo el punto más débil. Con frecuencia, los incidentes de seguridad ocurren no por fallas del sistema, sino por desconocimiento o falta de entrenamiento frente a técnicas de manipulación psicológica.

Ante este escenario, surge la necesidad de desarrollar una herramienta educativa que permita disminuir la vulnerabilidad humana. El proyecto CAPFIS (Capacitación en Fraudes a través de Ingeniería Social) responde a esa necesidad, ofreciendo un entorno web de aprendizaje interactivo donde los usuarios pueden adquirir conocimientos prácticos sobre las distintas formas de engaño digital y las medidas de prevención adecuadas.

Objetivo

De manera general el objetivo fue desarrollar una herramienta web interactiva de entrenamiento en ciberseguridad, orientada a la prevención de fraudes mediante ingeniería social. El sistema debe ofrecer contenidos educativos accesibles y actividades didácticas que promuevan el aprendizaje práctico sobre seguridad digital.

Los objetivos específicos que se cumplieron fueron los siguientes:

Contribuir a la formación de usuarios y organizaciones, fomentando la conciencia sobre la importancia de la seguridad digital.

Aplicar los conocimientos adquiridos durante la carrera para la concreción del proyecto final, logrando así la culminación de la formación como Técnico y Analista en Informática en la Universidad de la Empresa.

Alcance

La herramienta estará destinada a toda persona interesada en aprender sobre fraudes digitales, con especial énfasis en estudiantes, trabajadores y organizaciones que busquen reforzar sus conocimientos en esta temática.

El sistema se concentrará exclusivamente en los ataques de ingeniería social, por tanto, no se abordarán ataques puramente informáticos, sino aquellos en los que el atacante obtiene su beneficio mediante engaño o persuasión.

Especificación de Requerimientos

Este apartado tuvo el objetivo de documentar de manera completa y precisa las funcionalidades del sistema, garantizando que el equipo y las partes interesadas comprendan de forma unificada el alcance del proyecto.

Se definieron conjuntamente con el tutor académico tanto los requerimientos funcionales, que describen lo que el sistema debe hacer, como los no funcionales, que especifican las condiciones bajo las cuales debe operar.

El sistema CAPFIS contempla dos perfiles de usuario principales:

- Administrador: encargado de la gestión de usuarios, módulos y etapas de formación.
- Estudiante: quien accede al material educativo, realiza las actividades y visualiza su progreso.

Cada módulo de capacitación corresponde a un tipo de fraude específico, y el registro de avance permite monitorear la evolución del aprendizaje de cada estudiante.

Los requerimientos funcionales incluyen, entre otros:

- Inicio de sesión, registro y gestión de contraseñas.
- Visualización y edición del perfil.
- Acceso a los módulos educativos y seguimiento del progreso.
- Creación, modificación y eliminación de módulos y etapas por parte del administrador.
- Administración de usuarios y asignación de roles dentro del sistema.

En cuanto a los requerimientos no funcionales, el sistema se ejecuta en cualquier navegador moderno. Además las contraseñas se almacenan cifradas, permitiendo la seguridad de las mismas. La plataforma es escalable, permite actualizar los módulos sin tocar una línea de código. Cuenta con una interfaz intuitiva y adaptable a distintos dispositivos.

Estado del Arte

Con el fin de contextualizar el proyecto y conocer antecedentes similares, se realizó un análisis de distintas plataformas dedicadas a la formación en ciberseguridad. Entre las más destacadas se encuentran PhishLabs y KnowBe4.

PhishLabs ofrece soluciones corporativas orientadas a la detección y mitigación de ataques de phishing y otras amenazas digitales, mientras que KnowBe4 es una de las plataformas más reconocidas a nivel mundial en capacitación de usuarios y simulación de ataques de ingeniería social.

Sin embargo, ambas se orientan principalmente al ámbito empresarial y requieren licencias de pago, lo que limita su acceso a instituciones educativas o usuarios individuales.

En conclusión, mientras PhishLabs y KnowBe4 se consolidan como modelos de referencia para el ámbito empresarial, CAPFIS se presenta como una propuesta alternativa, sin licencia paga, totalmente gratuita, orientada al usuario final que quiera tener un primer contacto con el tema y no pretende en ningún momento competir con las mencionadas soluciones

Estudio de Factibilidad

El estudio de factibilidad evaluó la viabilidad del proyecto CAPFIS desde los aspectos técnico, legal, operativo, económico e institucional.

Tras el análisis, se determinó que el sistema podía implementarse exitosamente con los recursos disponibles y sin obstáculos legales o financieros.

Desde el punto de vista técnico, se eligió la plataforma .NET y SQL Server, por su robustez e integración nativa.

El equipo contaba con el equipamiento y las competencias necesarias, lo que garantizó un desarrollo estable y eficiente.

En el plano legal, el proyecto cumplió con la Ley N.º 18.331 de Protección de Datos Personales, asegurando el consentimiento del usuario y el uso exclusivo de recursos libres de derechos.

La factibilidad operativa y humana se sustentó en la experiencia del equipo y el acompañamiento del tutor académico, que aseguró el cumplimiento de los objetivos y la correcta ejecución del plan de trabajo.

Desde el punto de vista económico, el desarrollo no requirió inversión adicional, ya que se utilizaron herramientas gratuitas.

A nivel institucional, el anteproyecto fue avalado por la Facultad de Ingeniería de la UDE, garantizando su validez académica.

En conclusión, el análisis confirmó que CAPFIS fue técnica, legal y operativamente viable, sostenible y alineado con los objetivos de la carrera y de la Universidad.

Plan de Riesgos

El objetivo principal fue identificar los riesgos que pudieran surgir en las diferentes etapas del proyecto, evaluarlos en función de su impacto y probabilidad , y diseñar medidas de mitigación y contingencia para los más críticos.

Cada factor fue calificado en una escala del 1 al 5 tanto para impacto como para probabilidad, donde el valor más alto representa una mayor severidad o posibilidad de ocurrencia. La prioridad se determinó multiplicando ambos valores, lo que permitió jerarquizar los riesgos y concentrar los esfuerzos en aquellos más críticos.

Entre los riesgos más relevantes se encuentran la **definición incorrecta de requerimientos**, la **ausencia temporal de un integrante por fuerza mayor** y el **incumplimiento de plazos establecidos**. También se consideraron riesgos técnicos, como la **incompatibilidad de software**, el **limitado dominio de herramientas** y la **pérdida de documentación o código fuente**.

Para cada caso se definieron medidas preventivas y planes de acción:

- Ante errores en los requerimientos, se estableció una validación conjunta con el tutor y la documentación detallada de cada ajuste.
- En caso de ausencias o sobrecarga de trabajo, el otro integrante asumiría temporalmente las tareas, pudiendo extender la jornada o utilizar fines de semana.
- Para los retrasos, se definió un seguimiento semanal del cronograma y la posibilidad de redistribuir tareas.
- Respecto a los riesgos técnicos, se acordó investigar previamente las herramientas y mantener copias en Drive, GitHub y respaldos locales, garantizando la recuperación ante pérdidas.

Este enfoque permitió anticipar posibles incidentes y reducir su impacto en los plazos y la calidad del proyecto.

Plan de Proyecto

El plan de proyecto define la estructura de trabajo utilizada para desarrollar CAPFIS, incluyendo el modelo metodológico, las etapas, los roles y las políticas internas.

Se adoptó Scrum como metodología de desarrollo por su carácter ágil e iterativo, que permite trabajar en ciclos cortos, obtener retroalimentación constante del tutor y adaptar el plan ante cualquier cambio.

El proyecto se organizó en cinco grandes etapas:

Análisis: identificación de objetivos, requerimientos funcionales y no funcionales.

Diseño y planificación: elaboración de la arquitectura, diagramas y cronograma (Gantt).

Definición: documentación de los planes de riesgos, calidad y configuración.

Desarrollo e implementación: construcción de los módulos del sistema y ejecución de los sprints.

Testing y finalización: validación de funcionalidades, corrección de errores y auditorías.

Las políticas del proyecto establecieron normas de trabajo claras. Se definió un régimen de reuniones semanales entre los integrantes y quincenales con el tutor, además de la comunicación diaria por WhatsApp y Trello para ver las tareas activas.

Todo cambio relevante debía solicitarse, evaluarse y documentarse en acta, como ocurrió al migrar de MySQL a SQL Server por motivos de compatibilidad con Entity Framework Core 9. En cuanto a la calidad, se aplicaron las normas IEEE y formato APA, con revisiones cruzadas antes de cada entrega. El respaldo del trabajo se garantizó mediante GitHub y en Google Drive.

El equipo, conformado por Dayron Muñiz y Diego Balbis, asumió roles compartidos (Product Owner, Scrum Master, Desarrolladores y Testers), trabajando de manera colaborativa y equilibrada.

Las reuniones periódicas con el tutor permitieron mantener el control del avance, documentar actas y asegurar el cumplimiento de los plazos establecidos.

El diagrama de Gantt fue una herramienta fundamental para la planificación y el control del proyecto.

Cada actividad se organizó dentro de los cuatro sprints definidos por la metodología Scrum, reflejando el progreso desde la fase de análisis hasta el cierre y testing final.

Gracias al Gantt, el equipo pudo:

- Monitorear el cumplimiento de los plazos establecidos.
- Detectar desviaciones tempranas y reorganizar tareas cuando fue necesario.
- Mantener una visión clara del avance en cada etapa.
- Coordinar las actividades entre ambos integrantes y el tutor académico.

Arquitectura

La arquitectura de CAPFIS fue diseñada para garantizar modularidad, mantenibilidad y escalabilidad.

El sistema se desarrolló bajo el patrón MVVM (Modelo–Vista–VistaModelo), que permite separar la interfaz de usuario de la lógica de negocio y la capa de datos, facilitando el mantenimiento y la evolución del software.

Desde el punto de vista lógico, el sistema se organiza en tres capas:

- Presentación: interfaz web desarrollada con Razor Views y Bootstrap, que ofrece un diseño moderno y adaptable.
- Negocio: contiene las reglas del sistema implementadas en C# sobre .NET 9.
- Datos: maneja la persistencia mediante SQL Server 2022 y Entity Framework Core, garantizando integridad y coherencia.

En el plano físico, CAPFIS se basa en una arquitectura cliente-servidor, donde los usuarios acceden desde un navegador y el servidor de aplicaciones se comunica con el servidor de base de datos. Este esquema brinda seguridad, rendimiento y escalabilidad.

Las tecnologías empleadas incluyeron .NET 9.0.4, C# 13, Bootstrap 5.3.3, SQL Server 2022, Entity Framework Core 9, ASP.NET Core Identity, GitHub, Google Drive y GanttProject, combinando solidez, seguridad y facilidad de mantenimiento.

Plan de calidad

El Plan de Calidad tuvo aseguró que tanto el producto como el proceso de desarrollo mantuvieran un estándar técnico y documental durante todas las fases del proyecto. Desde el inicio se establecieron prácticas de control, revisión y mejora continua que garantizaron la coherencia del trabajo y el cumplimiento de los requerimientos definidos.

El plan abarcó todos los activos del proyecto —documentación, código fuente, archivos multimedia —, apoyándose en la autoevaluación constante del equipo y en la supervisión del tutor académico.

Para ello, se tomaron como referencia las normas IEEE 730 (Aseguramiento de la Calidad del Software) y IEEE 1058 (Gestión de Proyectos de Software), además del marco de trabajo Scrum y el patrón arquitectónico MVVM en el entorno ASP.NET Core 9.

En cuanto a los estándares de desarrollo, se aplicaron las convenciones de codificación de C#, tales como el uso de **PascalCase para clases y métodos**, **camelCase para variables** y **PascalCase para constantes**, con el fin de mantener un código legible y consistente.

El enfoque de calidad se basó en la prevención y la colaboración, priorizando las revisiones cruzadas entre los integrantes, la validación funcional de cada módulo y el uso de herramientas de control como Trello, GitHub y Google Drive.

Las principales actividades de aseguramiento incluyeron: revisión del código, verificación documental, ejecución de pruebas funcionales y retroalimentación continua del tutor.

Todo cambio fue registrado en GitHub y documentado en las actas de reunión, garantizando trazabilidad y control de versiones.

Finalmente, se establecieron criterios de aceptación para cada entregable: cumplimiento de los requerimientos funcionales y no funcionales, ausencia de errores críticos, integración correcta con el resto del sistema y validación del tutor.

Gracias a los lineamientos trazados en este apartado la aplicación alcanzó una calidad final adecuada para su implementación y presentación ante la Universidad.

Plan de Testing

El Plan de Testing se diseñó para comprobar que el sistema CAPFIS funcionara correctamente, cumpliendo con los requerimientos establecidos y garantizando estabilidad, seguridad y una buena experiencia de uso.

Las pruebas se integraron dentro del enfoque ágil de Scrum, de modo que en cada sprint se testearon los módulos desarrollados antes de pasar a la siguiente iteración.

El objetivo principal fue verificar el correcto comportamiento de cada módulo y etapa del sistema, asegurando la integridad de los datos y el cumplimiento de las funcionalidades. Las pruebas también incluyeron aspectos de rendimiento, seguridad y navegabilidad.

Gestión de usuario y perfil: edición y validación de datos personales.

Módulos educativos: navegación entre etapas, actividades interactivas (ahorcado, sopa de letras, encuentra la palabra) y seguimiento del progreso.

Gestión administrativa: creación y edición de módulos y usuarios, asignación de roles.

La estrategia de pruebas combinó cinco niveles complementarios:

Unitarias: para verificar el código individual en Visual Studio.

Integración: verificando la correcta comunicación de sus capas.

Funcionales: evaluando el cumplimiento de los requerimientos.

Regresión: asegurando estabilidad ante actualizaciones o modificaciones.

Aceptación: con la validación final del tutor y el equipo.

El entorno de pruebas incluyó Visual Studio 2022, SQL Server 2022, navegadores Edge y Chrome, Trello para seguimiento, GitHub para registrar versiones y Google Drive para documentación.

Los casos de prueba fueron documentados en el Anexo correspondiente.

El proceso de testing validó que CAPFIS cumple los objetivos propuestos y que las funcionalidades desarrolladas son estables, seguras y en concordancia con los requerimientos establecidos.

Las pruebas confirmaron la integridad general del sistema.

Plan de Métricas

El Plan de Métricas permitió medir objetivamente la calidad del producto y del proceso de desarrollo, combinando indicadores técnicos (mantenibilidad, complejidad, acoplamiento, herencia) con métricas de esfuerzo y tiempo.

Lo que se persiguió con el análisis de estas métricas fue tener durante todo el proceso de desarrollo valores que estuvieran en el mejor rango posible.

Métricas del producto

El análisis técnico se realizó mediante Visual Studio 2022 y el repositorio de GitHub, obteniendo resultados cuantificables en cada sprint.

El Índice de Mantenibilidad (MI) fue de 93,67, ubicando el sistema en el rango de excelente mantenibilidad.

La Complejidad Ciclomática promedio fue de 3,57, considerada baja. Su evolución fue controlada.

La Profundidad de Herencia obtuvo un valor de 2,91, lo que representa jerarquías cortas y fáciles de mantener.

El Acoplamiento de Clases fue de 3,42, indicando una arquitectura modular y con bajo nivel de dependencias.

Estos resultados confirman que el proyecto mantuvo una estructura limpia y estable.

Métricas del proceso

El análisis de proceso midió el esfuerzo real invertido en cada etapa. El total global fue de 508 horas, una cifra coherente con la planificación inicial que había sido estimada en 540 horas, teniendo en cuenta el trabajo de en promedio 10 horas semanales por cada integrante del equipo durante 27 semanas.

La distribución fue la siguiente:

Planificación: 11 h (2,2%)

Definición: 61 h (12%)

Análisis: 24 h (4,7%)

Diseño: 18 h (3,5%)

Desarrollo e implementación: 220 h (43,3%)

Testing y finalización: 174 h (34,3%)

Ambos integrantes —Dayron Muñiz (249 h) y Diego Balbis (259 h)— mantuvieron una participación equilibrada, demostrando un trabajo colaborativo constante.

Conclusiones Generales

El desarrollo de CAPFIS permitió integrar conocimientos técnicos, metodológicos y humanos en un proyecto con un claro valor educativo y social.

El sistema aborda una problemática actual: los fraudes digitales basados en ingeniería social, ofreciendo una herramienta web interactiva que combina teoría y práctica.

Desde el punto de vista técnico, el uso de .NET 9, MVVM y SQL Server 2022 garantizó una arquitectura modular, escalable y mantenible.

Las métricas obtenidas demostraron un código limpio, de alta mantenibilidad y baja complejidad, en línea con las buenas prácticas de desarrollo.

En el plano metodológico, Scrum permitió gestionar el trabajo de forma iterativa, con revisiones frecuentes, entregas parciales y capacidad de adaptación ante cambios. Esto facilitó la colaboración, la autogestión y el cumplimiento de los plazos.

En el aspecto formativo, el proyecto ofreció una experiencia integral, combinando análisis, diseño, desarrollo, testing y documentación académica.

CAPFIS se consolida así como un producto académico completo, capaz de demostrar competencias técnicas y de gestión.

Trabajos a Futuro

Aunque el sistema cumple plenamente los objetivos planteados, se identificaron posibles mejoras para versiones posteriores:

- Módulo de estadísticas avanzadas, con reportes automáticos de rendimiento y progreso.
- Optimización de accesibilidad, para garantizar el uso por personas con discapacidades.
- Aplicación móvil nativa, para Android e iOS.
- Integración de inteligencia artificial educativa, con tutorías automáticas y simulaciones adaptativas.

Estas líneas de trabajo permitirán que CAPFIS continúe evolucionando como una herramienta educativa moderna, viva y sostenible, capaz de adaptarse a nuevas formas de aprendizaje y a la evolución de los ciberataques basados en ingeniería social.

Agradecimientos

A la Universidad de la Empresa por brindarnos las herramientas académicas necesarias para concretar esta etapa.

A nuestro tutor, Ing. Pablo Martres, por su orientación técnica y acompañamiento constante durante el desarrollo del proyecto.

Y a nuestras familias y amigos, por su apoyo incondicional a lo largo de la carrera.

Muchas Gracias!!!