ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERÍA EN SISTEMAS INFORMÁTICOS

"BUENAS PRÁCTICAS DE SEGURIDAD EN EL SISTEMA DE ESTAFETAS DE LA DIRECCIÓN DE DESARROLLO ACADÉMICO EN LA ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO"

TRABAJO DE TITULACIÓN

TIPO: PROYECTO TÉCNICO

Presentado para optar al grado académico de:

INGENIERO EN SISTEMAS INFORMÁTICOS

AUTORES: SANDRA JOHANNA ORDOÑEZ GRANIZO

KEVIN DAVID CHIMBO ORTIZ

TUTOR: Ing. JORGE ARIEL MENÉNDEZ VERDECIA

Riobamba – Ecuador

2019

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERÍA EN SISTEMAS INFORMÁTICOS

El Tribunal del trabajo de titulación certifica que: El trabajo de investigación: Tipo: Proyecto técnico "BUENAS PRÁCTICAS DE SEGURIDAD EN EL SISTEMA DE ESTAFETAS DE LA DIRECCIÓN DE DESARROLLO ACADÉMICO EN LA ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO", de responsabilidad de los señores Sandra Johanna Ordoñez Granizo y Kevin David Chimbo Ortiz, ha sido minuciosamente revisado por los Miembros del Tribunal del trabajo de titulación, quedando autorizada su presentación.

	FIRMA	FECHA
Ing. Washington Luna **DECANO DE LA FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**	_____	_____
Ing. Patricio Moreno **DIRECTOR DE LA ESCUELA DE INGENIERÍA EN SISTEMAS**	_____	_____
Ing. Jorge Menéndez **DIRECTOR DEL TRABAJO DE TITULACIÓN**	_____	_____
Ing. Gloria Arcos **MIEMBRO DEL TRIBUNAL**	_____	_____

Nosotros, **Sandra Johanna Ordoñez Granizo** y **Kevin David Chimbo Ortiz** somos responsables de las ideas, doctrinas y resultados expuestos en este trabajo de titulación; y el patrimonio intelectual de la Tesis de Grado pertenece a la "ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO".

Sandra Johanna Ordoñez Granizo

Kevin David Chimbo Ortiz

DEDICATORIA

A Dios y la Virgen María por permitirme cumplir una de mis metas. A mis padres José y Alicia por haberme guiado por un buen camino con sus consejos, apoyo, confianza, amor y su esfuerzo diario para sacarnos adelante a mis hermanos y a mí por haberme dado la oportunidad de convertirme en una profesional. A mis hermanos Shubert, Karen y Meyson que de una u otra manera me han apoyado, ellos son mi inspiración para ser un buen ejemplo, a mi madrina Piedad Ordoñez y a mis primas Alexandra y Alina Cortez quienes me brindaron su apoyo siempre, a Kevin que me apoyo en los momentos más difíciles y siempre está presente, a mis amigos Bernabé, Washington, Cristina, Erika, Andrés, Evelyn, Jhon y Marisol por acompañarme en los buenos y malos momentos.

Sandra

A Dios, por todas las bendiciones que ha derramado sobre mí. A mi familia por estar siempre presente, en especial a mis padres, Orlando y Mónica por ser mi guía siempre y pilar fundamental lleno de amor y de valores en el cual me he reposado y me ha impulsado a seguir adelante. A mis hermanos y docentes, por esos buenos y malos momentos que tuvimos que soportar. A mis mejores amigos quienes han dejado una huella importante en mi vida y siempre están presentes. Finalmente, a una persona muy especial Sandra, quien ha marcado mi vida, está presente en todo momento y me ayuda a crecer como persona.

Kevin

AGRADECIMIENTO

Cuando apenas recordamos los instantes que empiezan todas las etapas, nos damos cuenta que cada una termina y de cada una se toman experiencias inolvidables. A este logro, pequeño o grande, no solo significamos nuestro esfuerzo y motivación sino el de grandiosas personas como son nuestros padres y docentes que aportaron a nuestro desarrollo personal. A Dios nuestro guía y fuerza para seguir adelante en todo momento, dándonos oportunidades para vivir y crecer, y dicha fuerza es la que nos permite ser mejores. A nuestra familia, eje incondicional en la vida, siendo nuestros padres ese timón por el cual hemos crecido a su imagen y semejanza. A nuestros amigos, aquellos grandes jóvenes que nos dieron la mano en cada instante de esta maravillosa etapa. A los ingenieros del DTIC por su participación y ayuda en el desarrollo del trabajo. Y por supuesto a los docentes de la Escuela de Ingeniería en Sistemas, especialmente al Dr. Julio Santillán, a los Ingenieros Jorge Menéndez, Gloria Arcos y Germania Veloz por su apoyo incondicional en este trabajo de titulación.

Sandra y Kevin

TABLA DE CONTENIDOS

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

ÍNDICE DE GRÁFICOS

ÍNDICE DE ANEXO

RESUMEN

En el presente trabajo de titulación, se aplicó buenas prácticas de seguridad en el mantenimiento del sistema de Estafetas de la Dirección de Desarrollo en la Escuela Superior Politécnica de Chimborazo para influir en el nivel de vulnerabilidad; en primera instancia se procedió con la evaluación de la seguridad del sistema mediante la técnica de Pentesting, de acuerdo con los indicadores CIA (Confidencialidad, integridad y disponibilidad) definidos por el estándar ISO 27001. Al verificar la necesidad de un sistema informático seguro, se procedió con el mantenimiento del mismo mediante el uso de la metodología ágil SCRUM combinada con el estándar IEEE 1219 realizando el correspondiente análisis preliminar, planificación y mantenimiento. Una vez realizado el mantenimiento y para determinar la influencia del mismo se aplicó la técnica Pentesting, del conjunto inicial de ataques se seleccionaron 6, los cuales fueron cross site scripting, SQL Injection, cross site request forgery, directorio transversal, buffer overflow e intercepción criptográfica y fueron aplicados a 23 funcionalidades que representan al sistema de estafetas y con la evaluación de los indicadores CIA. Los resultados obtenidos aplicando la técnica estadística chi-cuadrado con un 95 % de confianza y un error de 5% determinaron que se acepta la hipótesis alternativa, que establece que el uso de buenas prácticas de seguridad en el mantenimiento, influye en la vulnerabilidad del sistema de estafetas; con un valor de vulnerabilidad inicial de 54 % y un valor final de 20 %, se concluyen que se influyó positivamente en el nivel de vulnerabilidad del sistema de Estafetas ya que se mejoró en un 34% la seguridad del mismo. Se recomienda el uso del manual de buenas prácticas de seguridad en el desarrollo o mantenimiento de sistemas informáticos para resguardar los datos almacenados de mejor manera.

Palabras clave: <ANÁLISIS DE VULNERABILIDAD>, <PRUEBAS DE PENETRACIÓN>, <MANTENIMIENTO DE SOFTWARE>, <METODOLOGÍA DE DESARROLLO ÁGIL (SCRUM)>, <ESTÁNDAR IEEE 1219>, <ATAQUES INFORMÁTICOS>, <ISO 27001>, <INDICADORES CIA>.

ABSTRACT

In the present degree work, good security practices in the maintenance of the courier system, of the Direction of Development at Escuela Superior Politécnica de Chimborazo, were applied to influence in the level of vulnerability. In the first instance, the evaluation of the security of the system using the Pentesting technique was carried out, in accordance with the CIA (Confidentiality, integrity and availability) indicators defined by the ISO 27001. When verifying the need for a secure computer system, it was proceeded with its maintenance through the agile SCRUM methodology combined with the IEEE 1219 standard, performing the corresponding preliminary analysis, planning and maintenance. Once carried out the maintenance and to determine the influence of it, the Pentesting technique was applied, from the initial set of attacks, 6 were selected which were cross site scripting, SQL Injection, cross site request forgery, cross directory, buffer overflow and cryptographic interception and they were applied to 23 functionalities that represent the courier system and with the evaluation of the CIA indicators. The results obtained by applying the chi-square statistical technique with 95% confidence and an error of 5% determined that the alternative hypothesis is accepted, which states that the use of good security practices in maintenance influences the vulnerability of the system of couriers; with an initial vulnerability value of 54% and a final value of 20%, it is concluded that the level of vulnerability of the courier system was positively influenced, since its security was improved by 34%. It is recommended the use of the manual of good security practices in the development or maintenance of computer systems to safeguard the stored data in a better way.

Keywords: <VULNERABILITY ANALYSIS>, <PENETRATION TESTING>, <SOFTWARE MAINTENANCE>, <AGILE DEVELOPMENT METHODOLOGY (SCRUM)>, <IEEE 1219 STANDARD>, <COMPUTER ATTACKS>, <ISO 27001>, <CIA INDICATORS>.

INTRODUCCIÓN

Planteamiento del problema

Antecedentes

La Escuela Superior Politécnica de Chimborazo (ESPOCH, 2017), en el mes de julio del 2003 aprobó mediante resolución de Consejo Politécnico la reestructuración orgánica funcional de la institución, misma que involucró a las diferentes dependencias administrativas y académicas con la finalidad de lograr una administración moderna y eficiente en sus diferentes ámbitos. Este cambio determinó que las tareas académicas encargadas al Departamento de Cómputo y Sistemas se vinculen directamente a las diferentes facultades y las funciones técnicas de asesoría, desarrollo de soluciones tecnológicas en el área informática se integren en la Dirección de Tecnologías de la Información y Comunicación (DTIC) mismas que se encontraban divididas en el comité Informático y el Departamento de Cómputo y Sistemas. Una de las funciones del DTIC es desarrollar y mantener los sistemas informáticos administrativos, académicos y de la organización, uno de estos sistemas antes nombrados es el de ESTAFETAS, el mismo que gestiona el formato que se encuentra registrado en el **Reglamento para la distribución y cumplimiento de la jornada laboral del personal académico de la ESPOCH** (resolución 116.CP.2014), en su **Artículo 23**: *"El registro de la actividad o dedicación de las y los profesores e investigadores de la ESPOCH se efectuará en el formato de Distribución de Jornada de Trabajo Semanal del Docente…"*

En reuniones mantenidas con el personal del DTIC, manifestaron que se podría tener efectos en contra como pérdidas económicas por robo y destrucción de información además del incumplimiento de la **norma de contraloría No. 410-10 Seguridad de tecnología de información**, en su medida No. 5: *"Implementación y administración de seguridades a nivel de software y hardware, que se realizará con monitoreo de seguridad, pruebas periódicas y acciones correctivas sobre las vulnerabilidades o incidentes de seguridad identificados"*, así como el incumplimiento de la **ley orgánica de transparencia y acceso a la Información pública**, en su artículo 10 **Custodia de la información** y **artículo 13 Falta de claridad en la información**, los cuales hacen referencia a la confidencialidad, integridad y disponibilidad, principios de la seguridad informática, esto provocado por la

vulnerabilidad de los sistemas informáticos de la ESPOCH, siendo ese el problema principal de la falta de seguridad en dichos sistemas, esto debido a que los mismos requieren conexión a la red de internet, lo que sirve como causa para ser blanco de ataques informáticos, otro problema es no contar con el personal que realice pruebas de vulnerabilidad y que lleve un registro del comportamiento de los sistemas informáticos existentes, ante estas eventualidades y no se aplican buenas prácticas de seguridad en el desarrollo de los sistemas.

Por lo que en el presente trabajo de titulación se plantea aplicación de buenas prácticas de seguridad en el sistema de Estafetas de la Dirección de Desarrollo Académico (DDA).

Existen dos trabajos similares, los cuales son de (Monar Monar *et al.*, 2018), el mismo que propone un conjunto de técnicas de programación segura para reducir las vulnerabilidades en las aplicaciones web utilizando el entorno de desarrollo PHP y (Cevallos Muñoz y Siguenza Plaza, 2016), en el cual se ha emitido una propuesta de mejores prácticas de seguridad para la creación de la aplicaciones móviles. Para el mantenimiento del sistema existe el trabajo de (Alfonzo, Mariño, & Godoy, 2012), el mismo que servirá como base teórica para el mantenimiento del sistema ya que el trabajo se fundamenta en abordar la aplicación de las prácticas SCRUM en las actividades propuestas por el estándar IEEE 1219.

Formulación del problema

¿Aplicar buenas prácticas de seguridad en el mantenimiento del sistema de estafetas influirá en el nivel de vulnerabilidad que este posee?

Sistematización del problema

¿Cuál es el nivel de vulnerabilidad del sistema de estafetas actualmente?

¿Cuáles son las buenas prácticas de seguridad para una aplicación web?

¿Es posible aplicar buenas prácticas de seguridad en el mantenimiento del sistema de estafetas?

¿Cuál es el nivel de vulnerabilidad del sistema después de haber realizado el mantenimiento?

Justificación

Justificación teórica

Aplicar buenas prácticas de seguridad en los sistemas informáticos, traen ventajas a la organización como:

- **Reducir el riesgo**

Convertir la red en un sistema seguro con soluciones diseñadas para interoperar y brindar protección multicapa.

- **Obtener visibilidad total**

Usar datos en tiempo real para proteger el acceso seguro, proporcionar inteligencia y detectar actividades sospechosas incluso en tráfico cifrado.

- **Proteger contra amenazas**

Aplicar políticas y adoptar medidas para proteger la red de la organización de amenazas conocidas y desconocidas (Cisco Systems, 2018).

Aplicar buenas prácticas de seguridad en los sistemas informáticos, reduce en gran medida los problemas, porque permiten rastreabilidad de las transacciones e identificar en cualquier momento quién realizó cada operación, quién la autorizó, cuándo fue realizada, desde dónde y toda esta información está disponible para el personal de auditoría (Fajardo, 2018).

Las vulnerabilidades antes nombradas deben ser controladas al momento de construir un software, ya que excluyéndolas se logrará influir en la protección de la información.

Las buenas prácticas, aportarán elementos de seguridad a aspectos como la información almacenada dentro de la base de datos, permitiendo una mejor integración entre la fuente de datos, las aplicaciones

que dan soporte al sistema, todo esto evaluado de forma segura en un proceso de retroalimentación constante y siempre buscando la mayor cobertura en una visión integral del sistema (Acosta e Isaza, 2010, pp.53-54).

Este trabajo de titulación se realiza con el propósito de aportar conocimiento sobre buenas prácticas de seguridad informática, como instrumento para el desarrollo de software. Estas técnicas ayudarán a influenciar en cierto nivel en las vulnerabilidades de los sistemas informáticos. Los resultados de este trabajo podrán sistematizarse en una propuesta para ser incorporado como buenas prácticas de seguridad y se las podrán incluir en el desarrollo de aplicaciones en el DTIC.

Justificación aplicativa

La Dirección de Desarrollo Académico de la ESPOCH, entre otras funciones, lleva el control, seguimiento y generación de la Distribución de la jornada de trabajo semanal del docente, actualmente este proceso se lo realiza con uso excesivo de tiempo ya que existe un sistema que no se usa, porque el mismo se encuentra con un avance del 60% según el resumen ejecutivo Nro. **SDA.RE.003.2018** (Veloz, 2018). Sin el 40% del sistema no es posible que el mismo se publique en explotación ya que el sistema se complementa con las funcionalidades faltantes para su correcto funcionamiento.

El sistema de Estafetas con el avance del 60%, está compuesto por los módulos de:

Módulo de administrador

Las secciones de un documento son: Docencia, gestión e investigación

- **Secciones**: permite realizar la gestión de los ítems de una sección
  - **Ítems**: permite la gestión de los ítems de cada una de las secciones de un documento.
- **Tipos designación**: este módulo permite la gestión de los tipos de designación mismo que sirve para crear una comisión.
  - **Tipo función**: En este módulo permite la gestión de los tipos funciones, aquí se crean las funciones que se van a desempeñar dentro de una comisión.

- **Designación**: este módulo permite la gestión de las designaciones mismo, en el que se designa una comisión a una escuela.
  - **Función**: En este módulo permite la gestión de las funciones, es aquí en donde se designa a un docente a una designación.

Módulo de docente

- **Horario**: En este se presenta el horario del docente que viene desde el oasis y también permite ingresar un nuevo evento y en el mismo se pude visualizar.
- **Datos personales**: En este módulo se cargan los datos personales el docente de forma estática.
- **Estadística laboral**: En este módulo se presenta la estadística laboral del docente por días, el porcentaje del día que el docente trabaja.
- **Resumen de horas**. - En donde se muestran los ítems por cada sección con el respectivo número de horas. Además, en este módulo también está la hora de evidencia del acompañamiento a estudiantes.

Módulo de vicedecano

- **Reportes**. - en este módulo se pueden visualizar los reportes tanto de docentes que hayan enviado la estafeta como los que no la hayan enviado.

Los módulos en los cuales se trabajará en el presente trabajo de titulación serán:

Módulo general del sistema

- **Inicio de sesión:** integrar el sistema de autenticación institucional (CAS).

- **Cierre de sesión:** invalidar el uso del sistema con el usuario del docente, en caso de que este lo desee.

Módulo de docente

- **Horario:** permitir modificar y eliminar eventos del horario.
- **Resumen de Horas:** validaciones en el valor de las horas por ítems.
- **Observaciones:** observar el listado de observaciones realizadas a su estafeta.

Módulo de vicedecano

- **Tipos designación**: permite la gestión de los tipos de designación mismo que sirve para crear una comisión.
  - **Tipo función**: permite la gestión de los tipos funciones, aquí se crean las funciones que se van a desempeñar dentro de una comisión.
- **Designación**: permite la gestión de las designaciones mismo, en el que se designa una comisión a una escuela.
  - **Función**: permite la gestión de las funciones, es aquí en donde se designa a un docente a una designación.
- **Aprobación de estafeta**
  - **Horario:** permite solo ver el horario para determinar si se aprueba o se envía a corrección.
  - **Resumen de horas:** permite solo ver el número de horas por cada ítem filtrado por sección y el total de horas del cada uno.
  - **Aprobar:** si el número de horas es correcto aprueba la estafeta del docente.
  - **Desaprobar:** permite cambiar el estado de la estafeta en caso de errores o cambios.
  - **Observaciones:** insertar y listar observaciones realizadas a los horarios de cada docente.

Para optimizar el tiempo durante el seguimiento y generación de Distribución de la Jornada de Trabajo Semanal del Docente se proporcionará mantenimiento al sistema de estafetas, para permitir presentar resultados confiables y a la vez optimizar los recursos. Además, para influir en el nivel de vulnerabilidad del sistema se aplicarán buenas prácticas de seguridad mediante la siguiente metodología:

- Búsqueda bibliográfica de buenas prácticas de seguridad de aplicación web.
- Aplicar test de seguridad para medir el nivel de vulnerabilidad del sistema actual.
- Determinar las buenas prácticas que se van a usar para influir en las vulnerabilidades del sistema.
- Aplicar test de seguridad para medir el nivel de vulnerabilidad una vez realizado el mantenimiento.
- Comparar resultados obtenidos, para determinar si se influyó en el nivel de vulnerabilidad.

El sistema beneficiará a la institución en la minimización del tiempo en el proceso de planificación y presentación de documentos de evidencias de la estafeta en cuanto a proyectos de investigación, además de la generación de reportes, a nivel de Decano, para control y seguimiento de las estafetas de su facultad y director de escuela, para controlar la jornada del docente en la semana.

Según lo estipulado en la resolución 582. CP.2014-2018, el presente trabajo de titulación según el literal a que hace referencia al área de investigación científica del Ecuador recae en el numeral VIII , que trata de líneas y programas se encuentra en V Tecnologías de la información, comunicación, procesos industriales y biotecnológicos: literal c. Programa para el desarrollo de aplicaciones de software para procesos de gestión y administración pública y privada y d. Programa de desarrollo de la seguridad en la gestión de la información, dentro de las líneas de investigación de la Escuelas de Ingeniería en Sistemas (EIS) recaen la primera que se refiere al proceso de desarrollo de Software.

Objetivos

Objetivo general

Aplicar buenas prácticas de seguridad en el mantenimiento del sistema de Estafetas de la Dirección de Desarrollo en la Escuela Superior Politécnica de Chimborazo para influir en el nivel de vulnerabilidad.

Objetivos específicos

- Investigar acerca de las buenas prácticas de seguridad de aplicaciones web para crear un manual de buenas prácticas.
- Determinar el nivel de seguridad del sistema de Estafetas a través de pruebas de penetración para obtener el estado de vulnerabilidad actualmente.
- Aplicar buenas prácticas de seguridad en el mantenimiento del sistema de Estafetas mediante el uso del estándar IEEE 1219.
- Determinar el nivel de seguridad del sistema de Estafetas a través de pruebas de penetración para obtener el estado de vulnerabilidad una vez realizado el mantenimiento.

Estructura del documento

El presente trabajo de titulación está estructurado en tres capítulos, los mismo que se detallan a continuación.

Primer capítulo (Marco teórico referencial), está compuesto por el fundamento teórico para realizar el presente trabajo, se definen los ataques informáticos, la metodología para el desarrollo y las pruebas a realizar para comprobar el nivel de seguridad de la aplicación web.

Segundo capítulo (Marco metodológico), se detalla las acciones que se realizaran en cada fase de la metodología para el mantenimiento del software, las pruebas de intrusión, las técnicas empleadas y se establecerá como se va a demostrar la solución planteada en el inicio del trabajo de titulación.

Tercer capítulo (Marco de resultados y discusión), se exhibe y analiza los resultados de las pruebas de intrusión realizadas al sistema de estafetas, para determinar si el mantenimiento tubo efecto sobre el mismo.

Después de los capítulos descritos se describen las conclusiones obtenidas con el mantenimiento del sistema de estafetas y sus respectivas recomendaciones. Para finalizar, se detallan los anexos que están compuestos de información específica del desarrollo del trabajo de titulación.

CAPÍTULO I

1. MARCO TEÓRICO REFERENCIAL

1.1. Seguridad informática

La seguridad informática es un parámetro que deniega transacciones no autorizadas sobre un sistema o red de computadores, lo que produce modificación de la información y complica su confidencialidad, autenticidad o integridad; estos daños alteran el buen rendimiento de los equipos y restringen el acceso de personal autorizado al sistema (Gómez Vieites, 2011).

La seguridad informática es la técnica que se encarga de generar normas, procedimientos y métodos con el objetivo de crear un sistema informático seguro y confiable (Aguilera López, 2010).

La seguridad informática es la disciplina que se encarga de la conservación de la confidencialidad, integridad y disponibilidad de la información. Esta meta se consigue con la implementación de varios controles que circunscriben procedimientos, sistemas de hardware y software y políticas (Baluja García y Porvén Rubier, 2013).

Por lo tanto, el objetivo de la seguridad informática es resguardar la información, para que, esta no se modifique y así no se infrinjan las normas como confidencialidad, integridad y disponibilidad.

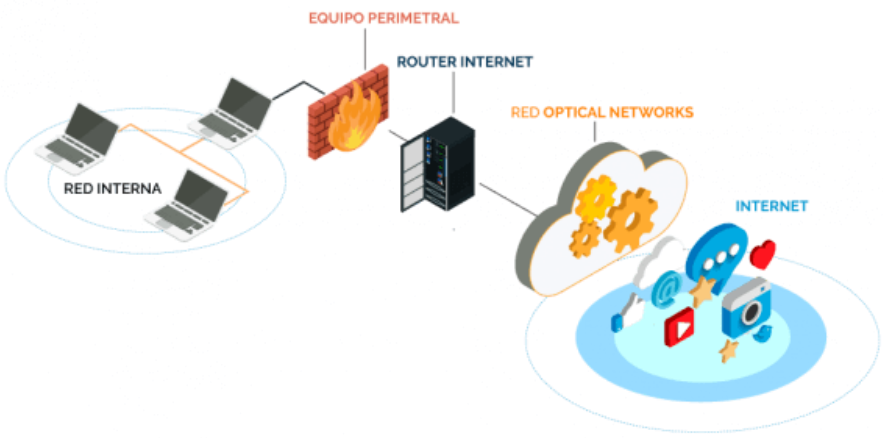


Figura 1-1: Función de la seguridad informática

Fuente: (Optical Networks, 2019)

1.2. ISO 270001: Gestión de la seguridad de la información

Es un estándar que permite el resguardo de la confidencialidad e integridad de la información y de los sistemas que la gestionan (ISOTools, 2019).

Mediante este estándar es posible determinar un sistema de gestión de seguridad y adecuarlo al negocio de manera rápida y sencilla (SGSI, 2017).

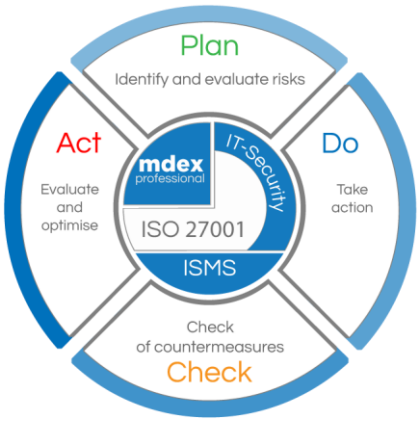


Figura 2-1: Estándar ISO 27001

Fuente: (Mdex, 2019)

Para el óptimo resguardo de la información, el estándar define como principio básico la CIA (Confidencialidad, integridad y disponibilidad) (SGSI, 2017).

- **Confidencialidad**: el conjunto de datos no debe ser habilitado ni develado a personal no autorizado.
- **Integridad**: las técnicas para procesar el conjunto de datos deben tener subsistencia de la precisión y totalidad.
- **Disponibilidad**: los actores, entes o técnicas que tengan autorización deben tener acceso y uso de los datos y aplicaciones de tratamiento cuando estos lo requieran.

Es importante tener en cuenta los principios antes citados, para no poner en peligro la información y obtener un producto software seguro bajo los parámetros que se considere en el desarrollo.

1.3. Fundación OWASP

El proyecto de seguridad de aplicaciones web abiertas (OWASP) es una organización mundial gratuita que tiene por objetivo disminuir las vulnerabilidades de los sistemas software. Además, emite herramientas de software y documentación basada en el conocimiento sobre seguridad en aplicaciones (OWASP Foundation, 2017).

1.4. Ataques informáticos

El ataque informático es el mecanismo por el cual se vulnera algún defecto en el hardware, software y personas que constituyen el ambiente informático, con el propósito de obtener beneficios, en su mayoría económicos, provocando fallos en la ejecución del sistema (Mieres, 2009).

El ataque informático es aquel proceso natural, técnico o humano que compromete la información de una entidad, ya que la misma puede llegar a ser modificada, destruida o interceptada (Parra y Porras, 2007).

El ataque informático es cualquier comportamiento delincuente en el cual está incluido un computador para modificar, interceptar o destruir información (Eliécer *et al.*, 2010).

El ataque informático, por lo tanto, es un procedimiento que tiene como principio fundamental comprometer la información almacenada en un ordenador, de forma tal, que la misma pueda ser obtenida, eliminada o modificada.

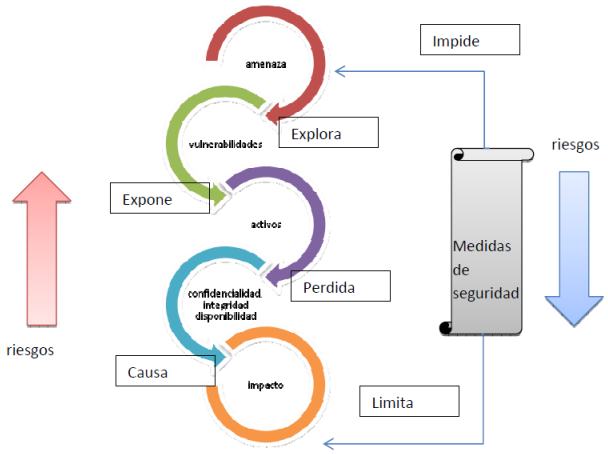


Figura 3-1: Funcionamiento de los ataques informáticos

Fuente: (Vélez, 2019)

Para el desarrollo del presente trabajo de titulación se definió 6 ataques informáticos, los cuales son más utilizados en aplicaciones web según la fundación OWASP y María Montes (OWASP Foundation, 2017) y (Montes, 2017); los mismos se detallan a continuación.

1.4.1. SQL injection

Es un ataque en contra de bases de datos como se observa en la **Figura 4-1**, el mismo que mediante parámetros de entrada o URL´s agrega código SQL (lenguaje de consulta estructurado) con el objetivo de acceder o manipular la base de datos (OWASP Foundation, 2017).

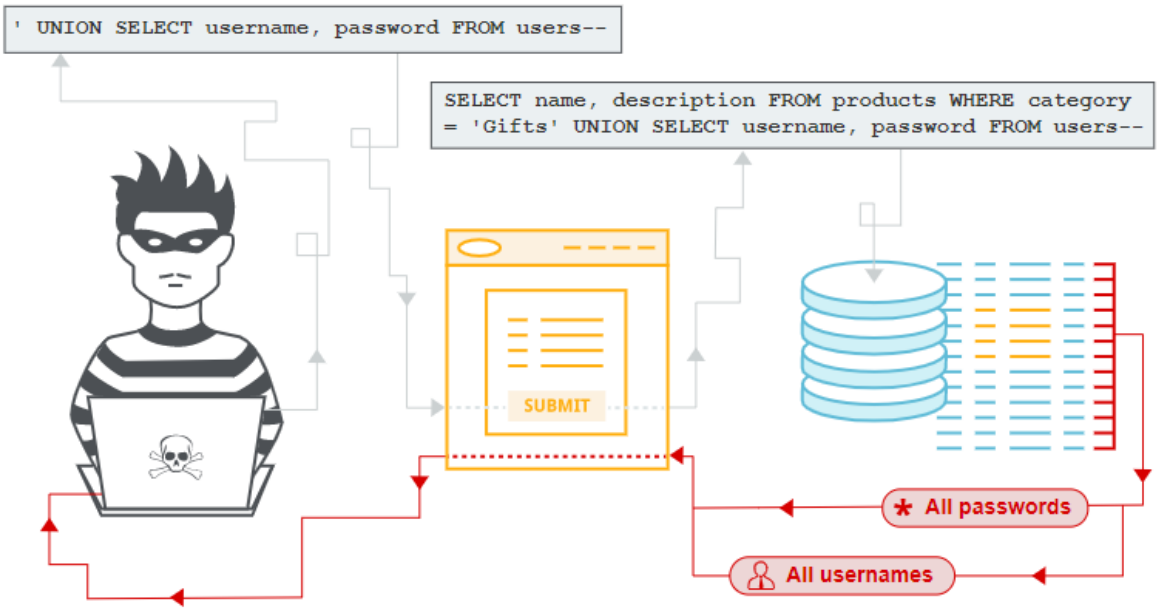


Figura 4-1 SQL Injection

Fuente: https://portswigger.net/web-security/images/sql-injection.svg

1.4.2. *Cross site scripting (XSS)*

Este ataque fue creado para vulnerar aplicaciones web, la misma que permite atacar sistemas mediante inserción de código JavaScript y como consecuencia el atacante podrá robar contraseñas, redirigir a otros sitios web falsos, entre otros (Soto, 2016).

Es posible explotar al máximo este ataque del lado del cliente cuando tenemos aplicaciones con campos de entrada que nos permita enviar información sin ninguna validación como se muestra en la **Figura 5-1**, aceptando scripts completos (Jimeno García, 2011, p.654).

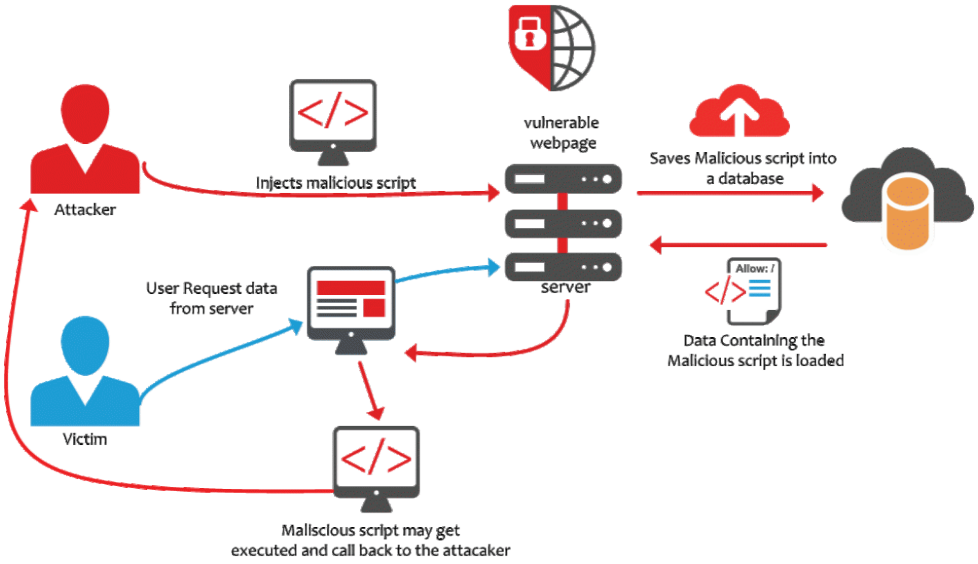


Figura 5-1 Ataque XSS

Fuente: https://cdn-images-1.medium.com/max/1600/1*rVA-dKOa7omxIQLYxRc1sw.gif

1.4.3. Buffer overflow (DOS)

Es un ataque informático orientado a cualquier sistema, en el cual se sobrepasa la capacidad máxima del búfer y se pierde acceso a los servicios o al sistema que se está consultando (Paguay, 2015), en las aplicaciones web, este ataque se realiza mediante la inserción de una extensa cantidad de caracteres en campos sin validaciones correspondientes.

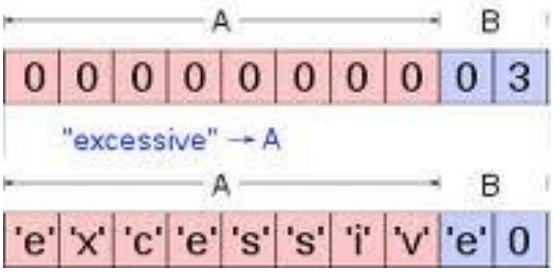


Figura 6-1 Ejemplo buffer overflow

Fuente:
https://www.ecured.cu/Desbordamiento_de_b%C3%BAfer#/media/File:Buffer_excesive.jpg

1.4.4. Directorio transversal

Este ataque permite tener acceso a los archivos de directorios superiores directamente del equipo que se está irrumpiendo, esto se genera cuando la aplicación no está asegurada adecuadamente para realizar validaciones en entradas de texto y los parámetros de las URL´s (fluidAttacks, 2018).

1.4.5. Cross site request forgery (CSRF)

La falsificación de solicitudes entre sitios es un tipo de ataque informático que duplica la sesión del usuario autenticado en la aplicación, y lo obliga a ejecutar acciones no autorizadas, son utilizadas al cambio de estados, pues el atacante no sabe cuál es el resultado ya que no puede ver la respuesta de la solicitud clonada (OWASP Foundation, 2018).

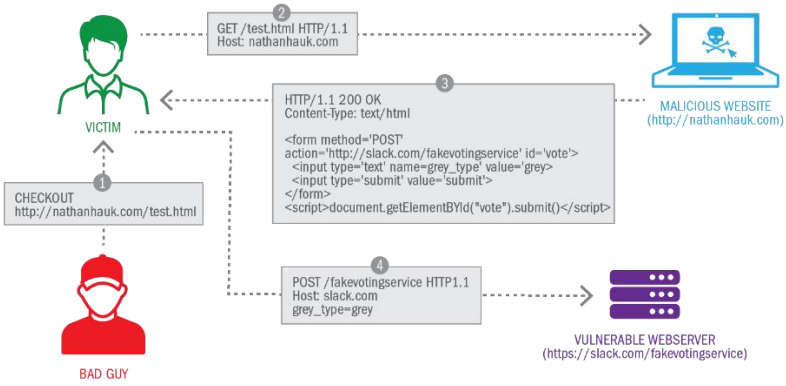


Figura 7-1: Ataque CSRF

Fuente: https://cdn-images-1.medium.com/max/1600/1*g_ISMQPCQpjw-6w3JRFUKQ.png

1.4.6. Intercepción criptográfica

Los ataques de intercepción criptográficos son orientados a obtener las claves utilizadas para cifrar información almacenada en un sistema (Gómez Vieites, 2014, p.6).

Esta técnica intenta capturar el tráfico que se genera por el uso de la aplicación y así obtener datos importantes que le permitan al atacante ingresar al sistema para extraer, modificar o insertar información (Paguay, 2015, p.15).

Con la captura del tráfico es posible obtener información cifrada y no cifrada, como se observa en la **Figura 8-1**, la diferencia es que en un canal de comunicación con seguridad va a ser casi imposible de acceder a la aplicación, caso que no sucede con el canal sin seguridad.

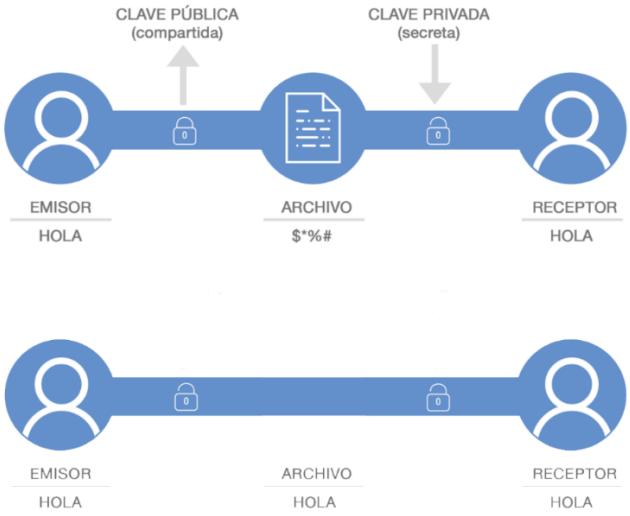


Figura 8-1 Canal de comunicación seguro e inseguro

Fuente: https://www.nextvision.com/wp-content/uploads/2017/08/Captura-de-pantalla-2017-08-24-a-las-12.26.29-p.m.png

1.5. Buenas prácticas

Las buenas prácticas son un conjunto de normas, probadas y aceptadas en el campo profesional, los mismos que sirven para aportar seguridad al negocio (Fontalvo, 2019, p.4).

Una buena práctica es una implementación de resultados favorables sobre uso de normas que han sido eficaces y útiles en determinados escenarios, contribuyendo a la mejora y solución de dificultades que se presentan a diario en el trabajo en diferentes contextos (Marco y Bermúdez Benítez, 2018, p.3).

Una buena práctica es una metodología que determina pautas para guiar a las organizaciones a conseguir objetivos, tener procedimientos adecuados y lograr productos de calidad; todo esto se da

gracias a experiencias obtenidas obteniendo resultados positivos, demostrando eficiencia y usabilidad en escenarios específicos (UNMSM, 2015, p.1).

Por lo tanto, el objetivo de las buenas prácticas es aportar con pasos para alcanzar un óptimo desenvolvimiento en la organización y la calidad que se aplicara en el proceso de producción para obtener bienes y servicios de calidad.

El presente trabajo de titulación está orientado a la seguridad en la programación de aplicaciones web, por tanto, a continuación, se numeran las buenas prácticas ante los ataques anteriormente citados.

Tabla 1-1 Buenas prácticas de seguridad en aplicaciones web

Ataques informáticos	Buenas prácticas
SQL Injection	• Usar PreparedStatrments en vez de Statements en la recuperación de consultas SQL. • Validar entradas en el lado del cliente con el archivo JavaScript validaciones.
Cross site scripting (XSS)	• Validación de entradas con la filtración de caracteres especiales en parámetros de formato establecido. • Filtración de etiquetas HTML, mediante el archivo JavaScript validaciones.
Buffer overflow (DOS)	• Filtrar la longitud del valor del parámetro para evitar la saturación de memoria del servidor.
Directorio transversal	• Validar entradas en el lado del cliente mediante el archivo JavaScript validaciones.
Cross site request forgery (CSRF)	• Mitigación basada en token • Implementar la librería CSRF Guard de OWASP Foundation (OWASP Foundation, 2018).
Intercepción criptográfica	• Encriptar las claves de autenticación • Instalar certificados SSL, con esto se encriptará el canal de comunicación cuando exista tráfico en la red.

Fuente: (Paguay, 2015)
Realizado por: Sandra Ordoñez y Kevin Chimbo, 2019

1.6. Vulnerabilidad de un sistema informático

Una vulnerabilidad informática es un defecto del sistema, la misma que permite que agentes no autorizados puedan obtener, eliminar o modificar información o procesos (Paguay, 2015, pp.9-7).

Una vulnerabilidad informática es una particularidad o propiedad de un activo del sistema informático que lo hace blanco fácil de ataques (Chicano, 2014).

Una vulnerabilidad informática es todo tipo de debilidad en los sistemas informáticos, es un conjunto de operaciones que en cualquier ámbito deje la seguridad de la información propensa a una amenaza (Techopedia, 2018).

De este modo, se define a una vulnerabilidad informática como aquella deficiencia de seguridad que un sistema posee, la cual permite que personal no autorizado realice transacciones que pueden afectar la integridad de la información. Además, se dice que mientras menos vulnerable, más segura es una aplicación informática.

La UNIR (Universidad de La Rioja, 2017) clasifica a las vulnerabilidades en tres tipos, los cuales se describen a continuación:

- **Fallos de implementación.** Se origina en la codificación de la fase previa correspondiente al diseño. Como ejemplos tenemos desbordamientos de búfer, formato, condiciones de carrera, path traversal, cross site scripting, inyección SQL, etc.
- **Fallos de diseño.** Se debe al incorrecto análisis de la fase previa la cual corresponde a requerimentación. Por ejemplo, TELNET no fue diseñado para su uso en entornos hostiles, para eso se implementó SSH.
- **Fallos de configuración**. Corresponde a la instalación de servicios innecesarios, los mismos que pueden estar realizados con escasa seguridad.

En el presente trabajo de titulación, para la detección de las vulnerabilidades que posee el sistema de estafetas se utilizará la técnica conocida como pruebas de penetración (Pentesting), la misma que se detalla a continuación.

1.6.1. Pruebas de penetración

Para comprender que son las pruebas de penetración, debemos conocer el significado de hacking ético, el mismo que SOLUTEC S.A. lo define como *"... una forma de referirse al acto de una persona usar sus conocimientos de informática y seguridad para realizar pruebas en redes y encontrar vulnerabilidades, para luego reportarlas y que se tomen medidas, sin hacer daño."* (SOLUTECSA, 2018).

El test de penetración es una técnica de hacking ético que, mediante un conjunto de procedimientos simula un ataque a un sistema con el objetivo de evaluar la seguridad de mismo y determinar posibles vulnerabilidades (Ramos, 2017, p.31).

Las pruebas de penetración son métodos que pretenden de diferente maneras vulnerar la seguridad de la red para extraer información sensible de la organización, para luego reportarlo y mejorar el nivel de seguridad (SOLUTECSA, 2018).

Una prueba de penetración es una evaluación, en la cual los sistemas informáticos son sometidos a ataques reales para alterar un determinado elemento de seguridad de la aplicación (López, 2017, p.13).

Por lo tanto, el objetivo de una prueba de penetración es simular experimentos que permiten vulnerar los sistemas informáticos y en base a los resultados, proteger las vulnerabilidades para que no exista pérdida o alteración de la información.

1.6.1.1. Tipos de pruebas de penetración

Existen tres tipos de pruebas de penetración, las mismas se detallan en la **Tabla 2-1**.

Tabla 2-1 Tipos de pruebas de penetración

Tipo	Descripción
Caja negra	La persona encargada de las pruebas no tiene idea alguna sobre los sistemas que va a intentar vulnerar.

Caja blanca	La persona encargada de las pruebas ha sido completamente informada sobre los sistemas que va a intentar vulnerar ya que se le ha proporcionado código fuente, información de la red, sistema operativo, etc.
Caja gris	La persona encargada de las pruebas ha sido parcialmente informada sobre los sistemas que va a intentar vulnerar ya que se le ha detalles internos de la aplicación.

Fuente: (López, 2017, p.15)
Realizado por: Sandra Ordoñez y Kevin Chimbo, 2019

En este trabajo de titulación se trabajará con pruebas de caja gris, ya que se tiene información como código fuente, direccionamiento IP, pero, no se tiene información acerca del servidor donde esta almacenado el sistema de estafetas.

Herramientas para pruebas de penetración

La principal herramienta que se utilizará para el desarrollo de las pruebas, es el sistema operativo Kali Linux, el mismo que se detalla a continuación.

- *Kali Linux*

Alonso Caballero lo define como *"... una distribución basada en GNU/Linux Debian, destinado a auditorias de seguridad y pruebas de penetración avanzadas ..."* (Caballero, 2018, pp.9-12).

- Características según (Caballero, 2018, pp.9-12)
  - Libre de código abierto.
  - Posee aproximadamente 600 herramientas para pruebas de penetración.
  - Desarrollado en entorno seguro
  - Núcleo único, con parches para inyección.
  - Multilenguaje
  - Multitarea
  - Cumple con el estándar de jerarquía de archivos (FHS)

Debido a que Kali Linux cuenta con varias herramientas para realizar pruebas de penetración, se seleccionó este para el desarrollo de la evaluación al sistema de estafetas, las mismas se detallan en la **Tabla 3-1**.

Tabla 3-1 Herramientas de Kali Linux para pruebas de penetración

Herramienta	Características	Ataque
BurpSuite	○ Intercepción de proxi ○ Rastreo ○ Detección de vulnerabilidades automáticamente ○ Herramienta de repetición ○ Permisos para escribir plugins propios.	CSRF SQL Injection
NMap	• Basada en la Web • Código libre • Automatiza detección de vulnerabilidades • Extrae información de la base de datos	SQL Injection
Wireshark	• Bajo licencia GLP • Robusto y no promiscuo • Captura datos de la red por protocolos. • Basado en pcap • Excelente capacidad de filtrado	Intercepción criptográfica CSRF SQL Injection
Mozilla Firefox	Navegador web Protección antiphishing Marcadores dinámicos Corrector ortográfico Búsqueda integrada	XSS CSRF SQL Injection Directorio transversal Buffer overflow Intercepción criptográfica

Fuente:(López, 2017, pp.17-18)
Realizado por: Sandra Ordoñez y Kevin Chimbo, 2019

1.7. Mantenimiento del software

El mantenimiento de software está determinado por un conjunto de procesos y directrices que se deben acatar para remediar o prevenir cualquier error en el sistema informático, así como agregar nuevas funcionalidades (Pérez Carvajal, 2014).

El mantenimiento de software es aquella corrección que se realiza a una solución de software para adecuarlo a un nuevo entorno, modificar errores o aumentar su utilidad (IEEE, 1998).

Se define al mantenimiento de software como un proceso continuo de mejoras, entre las que figuran: corrección de errores y adaptación a nuevas plataformas, dichas actividades serán planificadas para lograrlas en el tiempo estimado (Pressman, 2010).

Por tanto, el mantenimiento de software abarca todas las modificaciones, ya que las mejoras o adaptaciones que se realizan a un producto software se generan con el objetivo de optimizar la utilidad del mismo.

Tabla 4-1 Tipos de mantenimiento del software

Tipo		**Definición**
No panificable	Correctivo urgente	Modificar las potenciales fallas que impiden el flujo normal del software.
Panificable	Correctivo	Modificar las potenciales fallas que no impiden el flujo normal del software, pero son un problema para la organización.
	Perfectivo	Añadir al software valor agregado con requerimientos solicitados por el cliente.
	Adaptativo	Someter al software a una modificación, con el objetivo de adaptarlo a los procesos del entorno de trabajo.
	Preventivo	Someter al software a modificación, para mejorar sus características.

Fuente: (Ruiz, Polo y Alarcos, 2001)
Realizado por: Sandra Ordoñez y Kevin Chimbo, 2019

1.8. IEEE 1219 para el mantenimiento del Software

El estándar IEEE 1219 es un modelo que detalla los procedimientos para ejecutar mantenimiento de los sistemas software.

El IEEE clasifica al estándar en siete faces, las mismas que se detallan a continuación (IEEE, 1998).

1.8.1. Identificación, clasificación y priorización de problemas o modificaciones

En esta fase se da una clasificación de prioridad inicial al requerimiento.

- **Entrada**. Solicitud de modificación
- **Proceso.** Esta fase está determinada por los pasos que se encuentran a continuación:

a) Asignar un identificador
b) Determinar el tipo de mantenimiento
c) Analizar el cambio para determinar si se acepta, rechaza o sigue en evaluación
d) Realizar una modificación preliminar del cambio o modificación
e) Priorizar la modificación
f) Establecer una solicitud de modificación a un bloque de cambios programados para su implementación.

- **Control.** La solicitud de modificación se insertará en la pila de productos de manera única.
- **Salida.** Como resultado se obtiene la solicitud de modificación validada y la pila de productos definitiva, la misma que debe estar compuesta de los siguientes elementos:

a) Nueva modificación o requisito;
b) Evaluación de la nueva modificación o requerimiento
c) Clasificación del tipo de mantenimiento que se necesita
d) Prioridades
e) Información a verificar
f) Estimación de los recursos para el mantenimiento

1.8.2. Análisis

Empleará información del repositorio y la MR (Solicitud de modificación) certificado en el documento de modificación.

- **Entrada.** Contendrá los siguientes ítems:

a) MR debe estar aprobado;

b) Considerar inicialmente los recursos e información adicional del repositorio;

c) Documentación del proyecto y del sistema, si está disponible.

- **Proceso.** Debe tener por lo menos dos componentes:

a) Un estudio de factibilidad; y

b) Un estudio realizado minuciosamente.

Se recomienda realizar una ingeniería inversa en caso de no existir o no estar disponible la documentación y el código fuente sea la única herramienta confiable del sistema de software.

- **Análisis de viabilidad.** Se realiza el análisis para un MR además de la creación de un FR (reporte de factibilidad). El mismo que está compuesto por:

a) Pueden ser modificados

b) Debe tener un plan de contingencia para soluciones

c) Los requerimientos pueden ser cambiados

d) Incluye medidas de resguardo

e) Contiene elementos humanos

f) Los costos deben tener una prórroga corta y larga

g) Al actualizar se debe tener beneficio

- **Análisis detallado**. Compuesto por los siguientes ítems:

a) Especificar concretamente los requerimientos para actualizarlos

b) Buscar las funcionalidades que requieren ser actualizados

c) Buscar inconvenientes con respecto al resguardo

d) Tener un plan de contingencia con respecto a la evaluación

e) Realizar una estrategia para la ejecución del sistema

Cuando se localizan las funcionalidades que se van actualizar, las personas que van analizarlas inspeccionan todo en cuanto a software, base de datos, y la documentación realizada y que están involucradas. Cada uno de ellos deben tener una identificación y crearse de ser prioritario, detallando cada fragmento del sistema que se va actualizar, las interfaces que no cuentan con un buen funcionamiento, las modificaciones visibles por el cliente para realizar las actualizaciones, y el periodo necesario para realizar todas las actualizaciones.

La integración de pruebas está fundamentada en la entrada de la actividad anterior la cual identifica los elementos para actualizarlos. Requerimientos para no menos de tres niveles de pruebas, que integran pruebas de componentes unitarios, pruebas integrales y se especificaran pruebas de aceptación orientadas. Requerimientos de prueba que intenta descubrir errores vinculada con cada uno de estos niveles de prueba también serán detallados. Una técnica de implementación precedente deberá mostrar el diseño, la implementación, las pruebas y la entrega de la actualización se lo ejecutará con un impacto pequeño en los clientes actuales.

- **Control**. Incluye lo siguiente

a) Recuperar lo más relevante de la documentación del proyecto y del sistema del control de configuración función de la organización;

b) Estudiar los cambios planteados y análisis de ingeniería para calificar la viabilidad técnica y económica y calificar la corrección;

c) Reconocer las medidas de seguridad y resguardo;

d) Tomar en cuenta la integración de la modificación planteada dentro del software presente;

e) Comprobar que todo el análisis sea correcto y la documentación del proyecto estén renovados y adecuadamente estudiados;

f) Comprobar que la función de prueba de la institución provee un método para evaluar los cambios, y que el calendario de cambios pueda ayudar al método de prueba propuesta;

g) Investigar la evaluación y cronogramas de los elementos y controlar su precisión;

h) Evaluación técnica para escoger la documentación de problemas y las mejoras punteadas que se implementaran en la próxima presentación. La nómina de modificación tiene que tener documentación. Al finalizar la fase de análisis, se deberá hacer un análisis de riesgo. Después de obtener los resultados de la fase de análisis se revisará la evaluación previa de recursos, y se tomará una decisión incluyendo al cliente para decidir si se pasa a la fase de diseño.

- **Salida.** En la fase de análisis, la salida del análisis del proceso de mantenimiento incluirá lo siguiente:

a) FR para las solicitudes de modificación;

b) Documentación detallada de análisis;

c) Requerimientos renovados;

d) Nómina de cambios previos;

e) Técnica de prueba;

f) Método de implementación.

1.8.3. Diseño

En esta fase para el diseño de la modificación del proyecto se utilizará, toda la documentación actualizada tanto del sistema como del proyecto, la base de datos y el software y el resultado de la fase de análisis. Se debe recopilar e inspeccionar en tramos adecuados los factores y las medidas/métricas que están asociados para esta fase.

- **Entrada.** Para el proceso de mantenimiento a la fase de diseño se debe tomar en cuenta los siguientes ítems:

a) Salida de la fase de análisis

1. Estudio preciso
2. Información de requisitos actualizados
3. Listado de modificaciones realizadas con anterioridad
4. Plan de pruebas
5. Estrategia de prueba

b) Sistema y documentación del proyecto

c) Código fuente existente, comentarios y bases de datos

- **Proceso.** Incluye los siguientes pasos:

a) Determinar las funcionalidades fallidas del sistema;
b) Actualizar el manual técnico;
c) Crear pruebas de aceptación para las nuevas funcionalidades, incluyendo medidas de seguridad;
d) Determinar/ Establecer pruebas de software para detectar errores;
e) Determinar los requerimientos que se van a modificar en el manual técnico;
f) Renovar la lista de funcionalidades.

- **Control.** Se empleará la siguiente técnica de control para modificaciones en la fase de diseño

a) Realizar un estudio del diseño de software en concordancia con IEEE Std 1028-1997.
b) Comprobar que el actual diseño / requerimiento se encuentre en el manual técnico y tenga permisos de modificación del software.
c) Comprobar la inserción de los nuevos elementos de diseño, incluyendo medidas de seguridad.
d) Comprobar que se haya modificado el manual técnico con la prueba adecuada.
e) Terminar la representación de los requerimientos hasta el diseño.

- **Salida.** Determina lo siguiente:

a) Enumerar las actualizaciones analizadas;
b) Modificación de diseño formalmente revisadas;
c) Manual técnico reestablecido;
d) Análisis revisado minuciosamente;
e) Requerimientos analizados;
f) Tener revisado el plan de implementación;
g) Enumeración de limitaciones y riesgos documentados.

1.8.4. Implementación

Para el desarrollo de esta fase previamente se debe haber completado las fases de análisis y diseño, para guiar el esfuerzo de implementación, datos de medición y los elementos asociados que deben ser recopilados y revisados apropiadamente.

- **Entrada.** Tiene los siguientes puntos:

a) Conocer de la fase de diseño, los resultados obtenidos;
b) Líneas de código actuales, base de datos y construcción de lenguaje de programación;

c) Manual técnico y manual de usuario.

- **Proceso.** Contiene los siguientes subprocesos:

a) Comprobar mediante pruebas por funcionalidad, que la programación no tenga fallos;
b) Combinación de funcionalidades;
c) Pasos para identificar los activos informáticos, probabilidad de ocurrencia, impacto, amenazas y vulnerabilidades;
d) Verificación de la preparación para la evaluación.

Los elementos que intervienen deben ser evaluados en momentos adecuados.

- *Codificación y pruebas unitarias*. Al momento de modificar elementos se lo realizará en el código, pruebas unitarias y algunas normas adecuadas de verificación y validación (V&V), además incluirán procesos.
- *Interacción.* Luego de codificar los cambios y realizar las pruebas unitarias, o en pausas adecuadas durante el proceso de codificación, las pruebas integrales y el sistema se vincularán con el programa actualizado. Se deben realizar pruebas a los requerimientos no funcionales (seguridad, facilidad de uso, rendimiento) del mantenimiento realizado a la aplicación.

Todo resultado adverso deberá ser notificado y enviado a codificación y pruebas de aceptación.

- *Análisis y revisión de riesgos.* Al igual que en la fase de análisis y diseño la evaluación de los riesgos es más factible realizarla durante la implementación de la fase más que al finalizarla, además se deben utilizar medidas para ponderar la evaluación de los riesgos.
- *Revisión del estado de preparación.* Para analizar la forma en la que se desarrollara la prueba del sistema, se realizara una investigación de acuerdo con IEEE Std 1028-1997.

- **Control.** Incluye lo siguiente:

a) Indagar en el código del software tomando en cuenta a la IEEE Std 1028-1997.
b) Verificar que se ejecuten y se documenten las pruebas de unidad e integración.
c) Verificar que el manual técnico de la prueba este modificado o creado.
d) Determinar, realizar el manual técnico y solucionar cualquier riesgo que se identifique durante el desarrollo del software y la verificación del acondicionamiento para pruebas.
e) Comprobar que el actual software este dirigido por SCM.
f) Comprobar que el manual técnico y la capacitación hayan sido renovadas.
g) Comprobar la ruta del diseño hacia el código.

- **Salida.** Incluye los siguientes resultados:

a) El software esta renovado
b) El manual técnico de diseño esta renovado
c) El manual técnico de prueba esta renovado
d) El manual de usuario esta renovado
e) Los elementos de preparación están renovados
f) Una afirmación de impacto y riesgo para los clientes
g) Documentación de la verificación de la preparación para la evaluación

1.8.5. Pruebas del sistema

En el sistema actualizado se realizarán las pruebas de aceptación en base al estándar IEEE 610, la evaluación que se realice al sistema incluye las pruebas para buscar errores, las mismas que se deben realizar para aprobar que la codificación no incluya errores irreales antes de realizar el mantenimiento.

- **Entrada.** Incluye lo siguiente:

a) Documentación de verificación de estructuración para la evaluación
b) Documentos como:
1. Manual de usuario
2. Manual Técnico

c) Aplicación actualizada

- **Proceso.** Se debe tener un sistema integrado en su totalidad para realizar pruebas las mismas que incluyen los siguiente:

a) Prueba tipo caja negra (basada en la realización y estudio de las funcionalidades);
b) Pruebas de integración;
c) Pruebas de software que se ocupan de buscar errores;
d) Examinar la estructuración de la prueba para la evaluación de la estructuración de las pruebas de aceptación.

Nota: Las conclusiones de las pruebas que se obtuvieron no se deben ser incluidas en el informe de pruebas.

- **Control.** Se deben realizar las pruebas del sistema en concordancia con pruebas que proporcionan información objetiva e independiente o a su vez por el aseguramiento de la calidad del software. La prueba independiente será la encargada de dar a conocer las decisiones que encuentran en el plan de prueba y finalizar con éxito. El resumen será informado al personal encargado de las revisiones antes de realizar las pruebas de aceptación.
- **Salida.** Incluye lo siguiente:

a) Integración y realización de pruebas al sistema;
b) Pruebas con su documentación;
c) Documento de control de la estructuración para la evaluación.

1.8.6. Pruebas de aceptación

Estas pruebas se las realiza mediante un software que se encuentra bajo SCM en concordancia con el IEEE Std 828-1998 y el IEEE Std 730-1998, también se las debe realizar cuando el sistema se encuentre absolutamente completo, además de ser realizadas por un delegado del cliente, por el cliente o por el usuario.

- **Entrada.** Tiene los siguientes ítems:

a) Documento de estudio de estructuración para la evaluación
b) Aplicación totalmente completa
c) Técnicas para la evaluación de funcionalidades
d) Casos para la evaluación de funcionalidades
e) Pasos para la evaluación de funcionalidades

- **Proceso.** Procedimientos para evaluar funcionalidades

a) Ejecutar evaluaciones a cada funcionalidad
b) Evaluar la operatividad interna del sistema
c) Ejecutar pruebas para encontrar errores

- **Control.** Tiene los siguientes ítems:

a) Realizar pruebas antes de entregar el avance
b) Documentar los resultados de las pruebas para auditoria funcional
c) Generar la auditoria de los requerimientos
d) Determinar las funcionalidades por realizar

e) Ubicar los documentos de la evaluación de funcionalidades en concordancia con la gestión de la configuración del software.

- **Salida.** Tendrá los siguientes ítems:

a) Nuevos requerimientos para la aplicación
b) Documento de la auditoria de configuración funcional
c) Documento de evaluación de funcionalidades

Nota: Sera responsable de la documentación del informe de la verificación de las funcionalidades el cliente.

1.8.7. Entrega

Para entregar un proyecto actualizado se describen los requerimientos.

- **Entrada.** Las nuevas funcionalidades serán las que entren a mantenimiento con la aplicación probada correctamente
- **Proceso.** Debe incluir los siguientes pasos:

a) Verificar de forma técnica todos los elementos de configuración;
b) Realizar una alerta al grupo de clientes;
c) Realizar un respaldo del sistema;
d) Capacitar sobre el funcionamiento e instalar la aplicación en la localidad del usuario.

- **Control.** El control de entrega incluirá lo siguiente:

a) Ordenar y evidenciar una auditoria de configuración física;
b) Proveer las herramientas necesarias a los clientes para la utilización apropiada del sistema;
c) Actualizar el documento de que gestiona las versiones del sistema;
d) Modificar bases de datos completamente;
e) Controlar mediante la gestión de la configuración del software las funcionalidades a entregar.

- **Salida**. Incluye los siguientes ítems:

a) Documentación de la auditoria de la configuración física (IEEE Std 1028-1997);
b) Documento de descripción de la versión.

1.9. Metodología hibrida SCRUM e IEEE 1219

Para reducir el costo del mantenimiento de software se debe tener una metodología para gestionarlo. Por tanto, los autores seleccionaron SCRUM para gestionar la normativa presente en el estándar IEEE 1219 (Alfonzo, Mariño y Godoy, 2012).

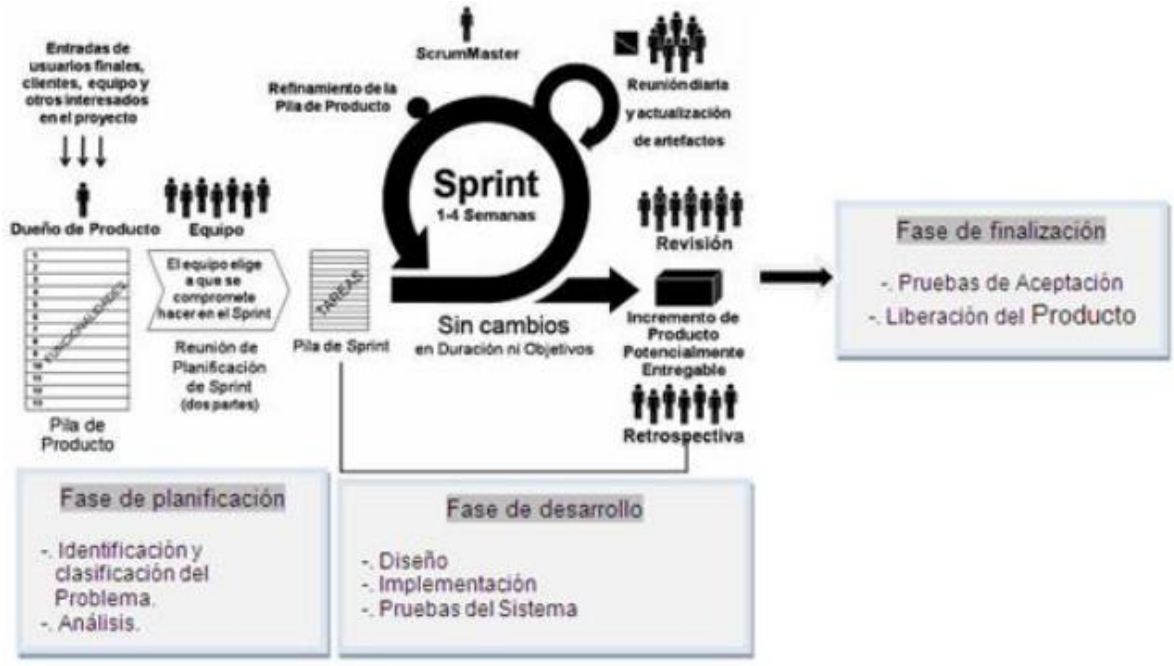


Figura 9-1 Funcionamiento de la metodología (SCRUM e IEEE 1219)

Fuente: (Alfonzo, Mariño y Godoy, 2012, p.5)

De manera general se refiere el proceso de mantenimiento, en la fase de planificación se define el equipo de trabajo, requerimientos de modificación del producto software, las mismas que son proyectadas en el Product Backlog y en función a los requisitos se escogen las solicitudes modificación que estarán presentes en el Sprint; además, se establece las estimaciones y el estudio de factibilidad, para determinar si el proyecto es viable. En la fase de desarrollo, se determinan los riesgos del proyecto; al terminar una petición, se realiza una reunión para validar y verificar el producto con el cliente, cada sprint tiene las fases de diseño, implementación y pruebas. Para la fase de finalización se realizará pruebas de aceptación y la liberación del producto (Alfonzo, Mariño y Godoy, 2012, p.6).

Tabla 5-1 Fases y actividades de la metodología SCRUM - IEEE 1219

Estándar IEEE 1219		**SCRUM**
Fases	**Tareas**	
Identificación y Clasificación del Problema o de la Modificación.	• Identificar el problema • Clasificar el problema por tipo de mantenimiento • Asignar prioridad • Obtener aprobación de la solicitud de modificación y las tareas a llevar a cabo. • Estimar inicialmente los recursos necesarios para modificar el sistema existente.	Fase de Planificación
Análisis.	• Evaluar el impacto • Evaluar los costos • Estudia la viabilidad y el alcance de las modificaciones • Desarrollar un plan preliminar de diseño, implementación, pruebas y liberación del software. • Desarrollar estrategia de pruebas	
Diseño	• Determinar objetos a modificar • Generar los casos de pruebas • Obtener lista de modificaciones revisada. • Generar guía básica del diseño actualizado. • Obtener planes de pruebas actualizados. • Obtener análisis detallado actualizado, requisitos verificados y plan de implementación revisado. • Generar lista de restricciones y riesgos documentados.	Fase de Desarrollo

Implementación	• Desarrollar y probar las modificaciones realizadas • Codificar y generar pruebas unitarias. • Integrar el software modificado con el sistema existente. • Analizar el riesgo. • Revisar la preparación para las pruebas.	
Pruebas del Sistema	• Realizar pruebas sobre el sistema modificado • Revisar integridad. • Obtener aprobación.	
Pruebas de Aceptación	• Realizar pruebas sobre el sistema completamente integrado	Fase de Finalización
Liberación del Producto	• Desarrollar un plan. • Notificar a los usuarios. • Realizar una copia de seguridad de la versión del sistema. • Realizar la instalación y capacitar a los usuarios.	

Fuente: (Alfonzo, Mariño y Godoy, 2012, pp.6-7)
Realizado por: Sandra Ordoñez y Kevin Chimbo, 2019

CAPÍTULO II

2. MARCO METODOLÓGICO

2.1. Tipo de estudio

El presente trabajo de titulación es una investigación aplicativa ya que se da mantenimiento a una herramienta que optimiza los procesos, por tanto, permite la solución de un problema, y su implantación beneficia a la institución involucrada.

2.2. Métodos y técnicas

En este estudio se utilizaron diferentes métodos y técnicas, las cuales se describen a continuación.

Investigación bibliográfica para definir las buenas prácticas de seguridad mediante artículos científicos, libros, revistas y páginas web confiables y seguras.

Para determinar el nivel de vulnerabilidad de la aplicación antes y después del mantenimiento se utilizó la técnica pentesting (pruebas de penetración) ayudándose del sistema operativo Kali Linux y las herramientas que este sistema operativo ofrece como: Mozilla Firefox, NMap, BurpSuit, y Wireshark. Además, para el desarrollo de las pruebas de penetración se tomó en cuenta la metodología propuesta en el trabajo de Delmys Pozo y otros (Pozo *et al.*, 2009), la misma que está compuesta por cuatro etapas.

En el mantenimiento de la aplicación se utilizó una metodología hibrida que combina a SCRUM como metodología ágil y a IEEE 1219 como estándar de mantenimiento del software, también se utilizaron las herramientas como Microsoft Planner para la gestión y repositorio de archivos del proyecto, Netbeas 8.0, el servidor Payara para el desarrollo y como base de datos PostgreSQL.

Para la presentación de resultados se aplicó la prueba estadística chi-cuadrado (x^2), con la ayuda de las herramientas Microsoft Excel y Minitab.

2.3. Población

La población en el presente trabajo se obtuvo a partir de 6 ataques propuestos por (OWASP Foundation, 2017) con (Montes, 2017) y las 23 funcionalidades que corresponden al sistema de estafetas, a cada ataque se asignan las todas las funcionalidades a las que es posible de la realización de la prueba, como se observa en la **Tabla 7-2**; obteniendo así, una población de 77 ataques a realizar.

2.4. Muestra

El tamaño de la muestra para el presente trabajo de titulación se determinó de acuerdo a la fórmula que se detalla a continuación.

$$n = \frac{z^2(p * q)}{e^2 + \frac{[z^2(p * q)]}{N}}$$

Donde,

n = tamaño de la muestra
z = nivel de confianza
p = porción de la población deseada (éxito)
q = porción de la población deseada (fracaso)
e = margen de error
N = tamaño de la población

Aplicando la ecuación se obtiene una muestra total de 65 ataques a realizar en el sistema de Estafetas.

2.5. Planteamiento de la hipótesis

H0: El uso de buenas prácticas de seguridad en el mantenimiento, NO influye en la vulnerabilidad del sistema de Estafetas.

H1: El uso de buenas prácticas de seguridad en el mantenimiento, influye en la vulnerabilidad del sistema de Estafetas.

2.5.1. Operacionalización de la hipótesis

En base a la hipótesis definida anteriormente se ha identificado dos variables, las mimas que se describen a continuación.

Independiente: Buenas prácticas de seguridad en el mantenimiento.

Dependiente: Vulnerabilidad del sistema de estafetas.

Tabla 6-2 Operacionalización de la hipótesis

Hipótesis	Variable	Indicadores	Instrumentos
El uso de buenas prácticas de seguridad en el mantenimiento, influyo en el nivel de vulnerabilidad del sistema de estafetas.	Buenas prácticas de seguridad en el mantenimiento	Técnicas de Aseguramiento aplicadas	Entorno de desarrollo integrado Código fuente
	Vulnerabilidad del sistema de estafetas	Confidencialidad	Software de monitoreo (NMAP, Burpsuite) Ejecución de algoritmos maliciosos
		Integridad	Software de monitoreo (NMAP, Burpsuite) Ejecución de algoritmos maliciosos
		Disponibilidad	Software de monitoreo (NMAP, Burpsuite) Ejecución de algoritmos maliciosos

Realizado por: Sandra Ordoñez y Kevin Chimbo, 2019

2.6. Procedimiento de pruebas de penetración

En la **Figura 10-2** se presenta el escenario inicial que se formulan para realizar las pruebas de seguridad, el cual está compuesto por un servidor donde se encuentra instalado el sistema sin buenas prácticas, un computador con el que se realizaran las pruebas de penetración, dispositivos para la conexión a la red MAN.

Figura 10-2 Escenario inicial para pruebas de penetración

Realizado por: Sandra Ordoñez y Kevin Chimbo, 2019

En la **Figura 10-2** se presenta el escenario final que se formulan para realizar las pruebas de seguridad, el cual está compuesto por un servidor donde se encuentra instalado el sistema con buenas prácticas, un computador con el que se realizaran las pruebas de penetración, dispositivos para la conexión a la red MAN.

Figura 11-2 Escenario final para pruebas de penetración

Realizado por: Sandra Ordoñez y Kevin Chimbo, 2019

2.6.1. Etapa 1: Planificación de las pruebas

En esta etapa se determinó los seis ataques que se van a ser aplicados en el sistema de estafetas, la categoría de la CIA (Confidencialidad, Integridad, Disponibilidad) a la que pertenecen y las funcionalidades a las que se va a aplicar dichos ataques tanto antes como después del mantenimiento, los datos antes mencionados se observan en la **Tabla 7-2**.

Tabla 7-2 Resumen de ataques al sistema de estafetas

Categoría	Ataque	Funcionalidad
Confidencialidad Integridad Disponibilidad	XSS	1. Insertar tipo designación 2. Insertar designación 3. Modificar tipo designación 4. Modificar designación 5. Insertar tipo función 6. Modificar tipo función 7. Insertar secciones 8. Modificar secciones 9. Insertar función 10. Modificar función 11. Insertar Ítems 12. Modificar Ítems
Confidencialidad Disponibilidad	Intercepción criptográfica	1. Autenticación local
Confidencialidad Integridad	SQL Injection	1. Insertar tipo designación 2. Insertar designación 3. Modificar tipo designación 4. Modificar designación 5. Insertar tipo función 6. Modificar tipo función 7. Ingresar secciones 8. Modificar secciones 9. Insertar función 10. Modificar función 11. Insertar Ítems 12. Modificar Ítems 13. Autenticación local 14. Listar eventos 15. Insertar eventos 16. Modificar eventos 17. Eliminar eventos 18. Eliminar secciones 19. Listar secciones 20. Listar Ítems 21. Listar designaciones

		22. Listar tipo designaciones 23. Eliminar designación
Confidencialidad Integridad Disponibilidad	CSRF	1. Insertar tipo designación 2. Insertar designación 3. Modificar tipo designación 4. Modificar designación 5. Insertar tipo función 6. Modificar tipo función 7. Ingresar secciones 8. Modificar secciones 9. Insertar función 10. Modificar función 11. Insertar Ítems 12. Modificar Ítems 13. Insertar eventos 14. Modificar eventos 15. Eliminar eventos 16. Eliminar secciones 17. Eliminar designación
Confidencialidad Disponibilidad Integridad	Buffer overflow (DOS)	1. Insertar tipo designación 2. Insertar designación 3. Insertar tipo función 4. Insertar Secciones 5. Insertar Ítems 6. Insertar función 7. Autenticación local
Integridad Disponibilidad	Directorio transversal	1. Insertar tipo designación 2. Insertar designación 3. Modificar tipo designación 4. Modificar designación 5. Insertar tipo función 6. Modificar tipo función 7. Ingresar secciones 8. Modificar secciones 9. Insertar función 10. Modificar función 11. Insertar Ítems 12. Modificar Ítems 13. Autenticación local 14. Visualizar horario del docente 15. Eliminar secciones 16. Listar secciones 17. Eliminar designación

Realizado por: Sandra Ordoñez y Kevin Chimbo, 2019

En la **Tabla 8-2** se muestra el plan para antes y después del mantenimiento en caso de la prueba XSS con sus respectivos elementos. En cuanto a la documentación de plan y demás pruebas de esta etapa se encuentra en la plantilla presente en el **Anexo A**.

Tabla 8-2 Planificación de la prueba XSS

NOMBRE	XSS (Cross-site scripting)	**IDENTIFICADOR**	PS01
ACTIVIDADES	Realizar ataques para vulnerar los requerimientos del sistema		
TIEMPO ESTIMADO	20 – 30 minutos por funcionalidad		
MÉTODOS O HERRAMIENTAS	Pentesting SO Kali Linux, Mozilla Firefox		
ENTREGABLES	Lista de chequeo del nivel de vulnerabilidad		

Realizado por: Sandra Ordoñez y Kevin Chimbo, 2019

2.6.2. Etapa 2: Diseño de pruebas

Para guiar los ataques que se van a realizar se determinó el diseño de las pruebas, la cual consta de ciertos campos que son necesarios para tener organización en cada prueba que se realice, en la **Tabla 9-2** se observa a detalle la información requerida para vulnerar la funcionalidad "Insertar tipo función" con XSS, las demás pruebas se encuentran documentadas en el **Anexo B**.

Tabla 9-2 Diseño de la prueba XSS

NOMBRE	XSS (Cross-site scripting)	**PRUEBAS**	P1
PROPÓSITO	Verificar si al aplicar el ataque XSS las funcionalidades del sistema son vulnerables.		
PRERREQUISITOS	• Haber solicitado realizar los ataques al servidor • Haber realizado un escenario de pruebas • Tener las funcionalidades previamente desarrolladas		
UBICACIÓN	Interfaz de usuario del sistema		
ENTRADA	Insertar código java script en los campos de las funcionalidades		
ORÁCULO	La interfaz reacciono de manera positiva a la prueba realizada.		
PASOS	1. Autenticarse en el sistema de estafetas 2. Seleccionar rol 3. Escoger en el menú la funcionalidad 4. Insertar código java script 5. Clic en guardar		

Realizado por: Sandra Ordoñez y Kevin Chimbo, 2019

2.6.3. Etapa 3: Ejecución de pruebas

En esta etapa se ejecutaron las pruebas haciendo uso de las herramientas y métodos definidos en las anteriores etapas, se inyecta código en la entrada de texto de una de las funcionalidades del sistema como se observa en la **Figura 12-2**.

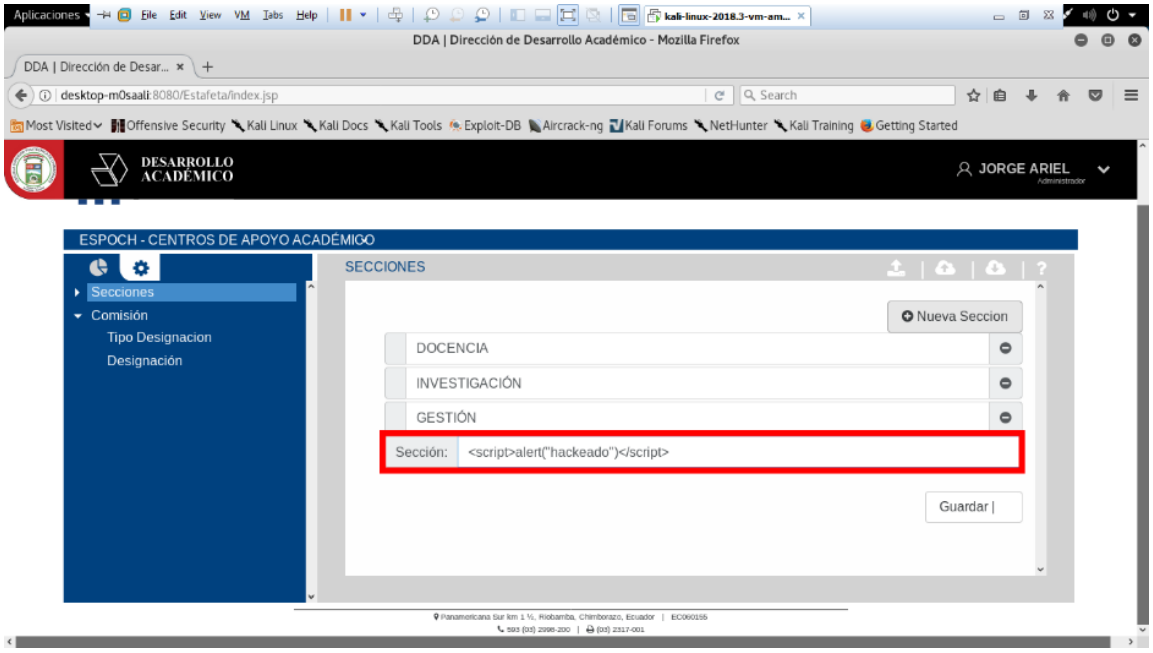


Figura 12-2 Ejecución de la prueba XSS

Realizado por: Sandra Ordoñez y Kevin Chimbo, 2019

En la **Figura 13-2**, se observa que el sistema reacciona ante el código inyectado en la funcionalidad de ingreso propuesta en la **Figura 12-2**.

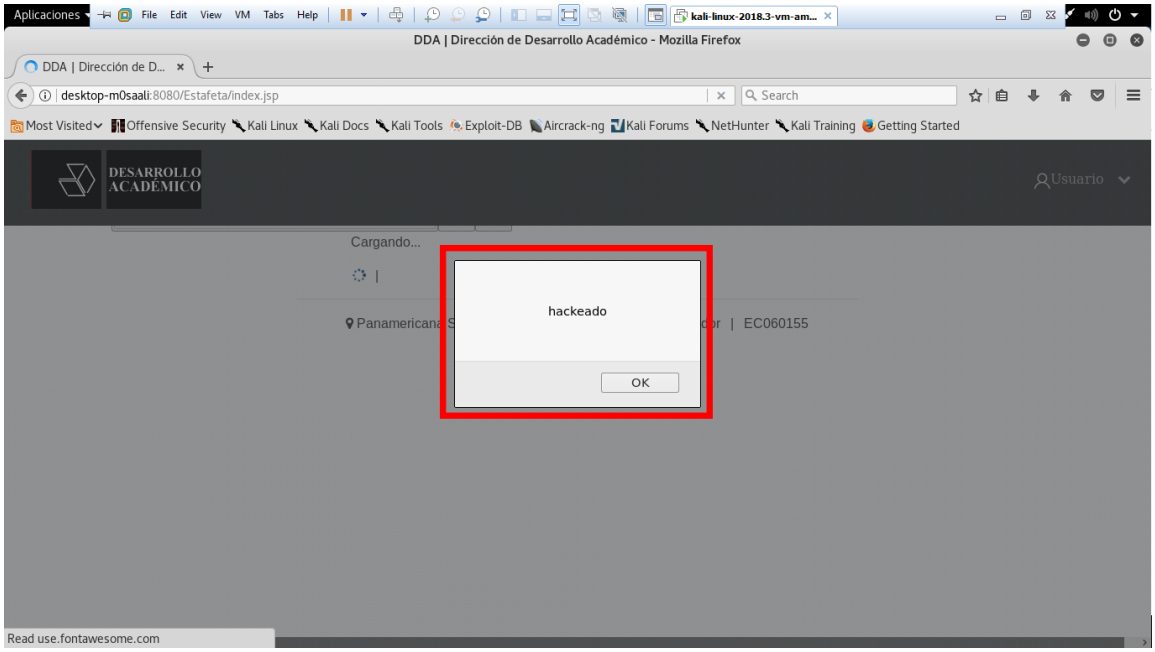


Figura 13-2 Reacción de la aplicación ante XSS

Realizado por: Sandra Ordoñez y Kevin Chimbo, 2019

En la **Tabla 10-2**, se observa el registro de la información obtenida a partir del ataque realizado a la funcionalidad propuesta antes de realizar el mantenimiento siendo esta, vulnerable. Todos los registros de las pruebas iniciales están en el **Anexo C**.

Tabla 10-2 Prueba XSS para insertar secciones sin buenas prácticas

Nombre	Insertar secciones (XSS)		**Identificador**	T01
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 100% vulnerable ante la prueba.			
Estado	**Vulnerable:**	X	**No Vulnerable:**	

Realizado por: Sandra Ordoñez y Kevin Chimbo, 2019

En la **Tabla 11-2**, se observa el registro de la información obtenida a partir del ataque realizado a la funcionalidad propuesta después de realizar el mantenimiento siendo esta, invulnerable. Todos los registros de las pruebas finales están en el **Anexo C**.

Tabla 11-2 Prueba XSS para insertar secciones con buenas prácticas

Nombre	Insertar tipo designación (XSS)		**Identificador**		T01
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 0% vulnerable ante la prueba.			
Estado		**Vulnerable:**		**No Vulnerable:**	X

Realizado por: Sandra Ordoñez y Kevin Chimbo, 2019

2.6.4. Etapa 4: Resultados

Los resultados serán calculados y presentados mediante los principios básicos del estándar ISO 27001, el cual define 3 indicadores los cuales son: confidencialidad, integridad y disponibilidad; además, se presentarán datos generales de los ataques realizados.

Para el cálculo de los porcentajes de vulnerabilidad de cada uno de los ataques que se aplicaron, se describe a continuación.

$$\%vulberabilidad_{(ataque)} = \frac{\sum_{I=1}^{n} x}{n}$$

Donde,

n = número de ataques realizados

x = valor de vulnerabilidad (0% = no vulnerable, 100% = vulnerable).

2.7. Metodología de mantenimiento de la aplicación

Existen varias metodologías de desarrollo de software tanto ágiles como tradicionales para el desarrollo de nuestro trabajo de titulación aplicaremos la metodología ágil SCRUM, aplicada al mantenimiento de software. Scrum permite trabajar colaborativamente y es resistente a cambios. Estas buenas prácticas nos permitirán llevar a cabo nuestro proyecto de manera eficiente con reuniones de entrega en las que el usuario podrá valorar lo trabajado de acuerdo a la **Tabla 5-1**. El principal problema del sistema, son las fallas en las funcionalidades que este posee.

2.7.1. Análisis

Estimaciones

Para realizar la estimación de cada tarea se utiliza el método de la talla de la camiseta o T-shirt. Las tallas o estimaciones del método son S, M, L y XL, como se lo presenta en la **Tabla 12-2**, descrita a continuación.

Tabla 12-2 Estimaciones por el método T-Shirt

Talla	Puntos estimados	Horas de Trabajo
S	30	30
M	60	60
L	120	120
XL	240	240

Realizado por: Sandra Ordoñez y Kevin Chimbo, 2019

Equipo de trabajo

En la **Tabla 13-2**, se determina el equipo de trabajo, el cual está definido por el rol que cada persona va a desempeñar en el mantenimiento del sistema.

Tabla 13-2 Equipo de trabajo

Persona	Rol
ESPOCH	Product Owner
Ing. Jorge Menéndez	Scrum Master
Sandra Ordoñez	Development Team
Kevin Chimbo	Development Team

Realizado por: Sandra Ordoñez y Kevin Chimbo, 2019

Estudio de factibilidad

Este estudio permite determinar si el desarrollo del sistema es factible con base en aspectos necesarios para la ejecución del proyecto. Se requiere de tres elementos para realizar correctamente este estudio los cuales están descritos a continuación.

- *Factibilidad técnica*

En este estudio se definen el hardware y software necesarios para el desarrollo y la instalación de la aplicación web. Como se menciona las herramientas de software son de licencia libre lo cual resulta un hecho beneficioso en la disminución de costos. **Anexo D**

- *Factibilidad operativa*

En la factibilidad operativa se determina el personal incluido en el mantenimiento del sistema informático y los usuarios del sistema antes mencionado. **Anexo E**

- *Factibilidad económica*

El análisis de la factibilidad económica está definido por todos los costos necesarios para el mantenimiento del sistema, los cuales son: costo de personal, instalación, operación, materiales y suministros, además de ciertos gastos adicionales que son necesarios para obtener un recurso económico real y acorde a las necesidades. **Anexo F**

En base a los estudios antes realizados se determina que la realización del proyecto es factible.

Riesgos

En este apartado se determinan los posibles riesgos que pueden presentarse en el desarrollo del manteniendo y que se puedan convertir en problemas, para evitar que esto suceda es necesario

documentar los riesgos mediante la identificación, análisis, priorización, y gestión para que sea posible dar seguimiento en el transcurso del desarrollo.

- *Identificación de riesgo*

En esta etapa se determinó el riesgo y se categorizo de acuerdo a su tipo, se obtuvo un total de 5 riesgos los cuales corresponden 3 riesgos de proyecto y 2 técnicos, la información detallada se encuentra en el **Anexo G**

- *Análisis de riesgos*

En esta etapa se determinó la probabilidad, el impacto, la exposición y la prioridad del riesgo, esta información se encuentra detallada a continuación.

Para establecer la probabilidad de que el riesgo se convierta en problema, se toma en cuenta los valores descritos a continuación: baja (1), 1% - 33%; media (2), 34% - 66% y alta (3), 67% - 99%. Si el riesgo se convierte en problema el proyecto no avanza, por lo que se parametriza para: bajo (1), retraso una semana; moderado (2), retraso 2 semanas; alto (3), retraso 1 mes y critico (4,5) retraso más de un mes.

Del valor de la multiplicación de probabilidad de ocurrencia y el impacto se calcula la exposición, en función del resultado se asigna la prioridad definida de la siguiente manera: alta (>5), prioridad 1; media (3,4), prioridad 2; baja (1,2) prioridad 3. **Anexo H**

- *Gestión de riesgos*

La gestión está determinada mediante los valores y consideraciones obtenidas en el análisis de los riesgos, causas, consecuencias y opciones para mitigar los riesgos antes que se conviertan en problema, para ello se define la reducción, supervisión, gestión y evaluación del estado en todas las etapas del mantenimiento del sistema, mediante una hoja de gestión. **Anexo I**

2.7.2. Fase de planificación

La planificación del proyecto es muy importante ya que permite tener una visión general del tiempo que se va a emplear en el mantenimiento, además proporciona un marco de trabajo en el cual se puede hacer estimaciones razonables de recursos y costos durante la ejecución del sistema. Se determinó la actualización y agregación de 24 requerimientos funcionalidades, además, se determinó a la seguridad como requerimiento no funcional.

Product backlog

El Product Backlog está compuesto por los requerimientos de acuerdo a una prioridad tomado en cuenta su complejidad, la descripción de requerimientos, los mismos que fueron transformados a historias de usuario y el tipo de mantenimiento como se observa en la **Tabla 14-2**.

Tabla 14-2 Product backlog

HU	Tarea	Tipo	Estimación
HU-19	Gestión Tipo designación para investigaciones	Perfectivo	60
HU-20	Gestión designación para investigaciones	Perfectivo	60
HU-21	Control de roles	Correctivo	30
HU-22	Aprobación del horario del docente	Perfectivo	30
HU-23	Resumen de horas docencia	Correctivo	30
HU-24	Resumen de horas gestión	Correctivo	30
HU-25	Resumen de horas investigación	Correctivo	30
HU-26	Control de horas de ÍTEMS de docencia	Correctivo	30
HU-27	Control de horas de ÍTEMS de investigación	Correctivo	30
HU-28	Control de horas de ÍTEMS de gestión	Correctivo	30
HU-29	Modificar eventos del horario	Perfectivo	30
HU-30	Eliminar evento del horario	Perfectivo	30
HU-31	Cerrar sesión	Perfectivo	30
HU-32	Listar docentes por facultad	Perfectivo	30
HU-33	Visualizar horario	Perfectivo	30
HU-34	Visualizar resumen de horas	Perfectivo	30
HU-35	Desaprobar estafeta	Perfectivo	30

HU-36	Autenticar con CAS	Correctivo	30
HT-09	Cambiar prepared statements del acceso a datos	Correctivo	30
HU-8	Insertar tipo designación	Correctivo	30
HU-4	Insertar tipo función	Correctivo	30
HU-10	Insertar función	Correctivo	30
HU-10	Insertar designación	Correctivo	30
HU-5	Insertar secciones	Correctivo	30
HU-5	Insertar ítems	Correctivo	30
HT-10	Modificar apertura y cierre de la conexión a la base de datos	Correctivo	30
HT-11	Seguridad en contra de los ataques CSRF	Correctivo	30
HT-12	Instalar el sistema en el servidor de pruebas institucional		30
HT-13	Revisión final del sistema		30
HT-14	Documentación final del sistema		30

Realizado por: Sandra Ordoñez y Kevin Chimbo, 2019

La historia de usuario determinada por un identificador que está compuesto por HU seguido de un número que varía secuencialmente, el usuario que requiere la funcionalidad, el sprint asignado, la prioridad en el negocio (alta, media o baja), la misma que es determinada por el solicitante en base a su necesidad.

Dependiendo del equipo de trabajo se determina el riesgo para el desarrollo (alto, medio o bajo), los puntos que se estimó y lo que tomó en realidad la funcionalidad, una descripción de lo que el cliente necesita que realice el sistema conjuntamente con el beneficio generado. Para finalmente describir las observaciones de acuerdo a detalles necesarios para el desarrollo. Un ejemplo de la historia de usuario se observa en la **Tabla 15-2**.

Se actualizaron y/o agregaron 24 historias de usuario, las mismas están detalladas en el **Anexo J**.

Tabla 15-2 Historia de usuario

Historia de Usuario	
Número: HU_22	**Nombre de la Historia:** Aprobación del horario del docente
Modificación de historia de usuario:	
Usuario: Vicedecano	**Sprint Asignada:** 2
Prioridad en el Negocio: Media	**Puntos Estimados:** 30
Riesgo en el Desarrollo: Medio	**Puntos Reales:** 30
Descripción: Como vicedecano quiero poder aprobar en horario del docente	

Observaciones:

Realizado por: Sandra Ordoñez y Kevin Chimbo, 2019

Sprint backlog

En el sprint backlog se organizó las historias técnicas y de usuario de acuerdo a la estimación obtenida en la **Tabla 12-2**, la misma que está compuesta por el responsable de actividad que se han planificado, fechas y esfuerzo como se muestra a continuación en la **Tabla 16-2**.

Tabla 16-2 Sprint backlog

ID	Fecha inicial	Fecha final	Esfuerzo	Responsable
SPRINT 1	**17/12/2018**	**03/01/2019**	**120**	
HU-19	17/12/2018	03/01/2019	60	Kevin Chimbo
HU-20	17/12/2018	03/01/2019	60	Sandra Ordoñez
SPRINT 2	**04/01/2019**	**17/01/2019**	**120**	
HU-21	04/01/2019	10/01/2019	30	Kevin Chimbo
HU-22	04/01/2019	10/01/2019	30	Sandra Ordoñez
HU-23	11/01/2019	17/01/2019	30	Kevin Chimbo
HU-24	11/01/2019	17/01/2019	30	Sandra Ordoñez
SPRINT 3	**18/01/2019**	**31/01/2019**	**120**	
HU-25	18/01/2019	24/01/2019	30	Kevin Chimbo
HU-26	18/01/2019	24/01/2019	30	Sandra Ordoñez
HU-27	25/01/2019	31/01/2019	30	Kevin Chimbo
HU-28	25/01/2019	31/01/2019	30	Sandra Ordoñez
SPRINT 4	**01/02/2019**	**14/02/2019**	**120**	
HU-29	01/02/2019	07/02/2019	30	Kevin Chimbo
HU-30	01/02/2019	07/02/2019	30	Sandra Ordoñez
HU-31	08/02/2019	14/02/2019	30	Kevin Chimbo
HU-32	08/02/2019	14/02/2019	30	Sandra Ordoñez
SPRINT 5	**15/02/2019**	**28/02/2019**	**120**	
HU-33	15/02/2019	21/02/2019	30	Kevin Chimbo
HU-34	15/02/2019	21/02/2019	30	Sandra Ordoñez
HU-35	22/02/2019	28/02/2019	30	Kevin Chimbo
HU-36	22/02/2019	28/02/2019	30	Sandra Ordoñez
SPRINT 6	**01/03/2019**	**18/03/2019**	**120**	
HT-09	01/03/2019	11/03/2019	30	Kevin Chimbo
HU-8	01/03/2019	11/03/2019	30	Sandra Ordoñez
HU-4	12/03/2019	18/03/2019	30	Kevin Chimbo

HU-10	12/03/2019	18/03/2019	30	Sandra Ordoñez
SPRINT 7	**19/03/2019**	**08/04/2019**	**120**	
HU-10	19/03/2019	25/03/2019	30	Kevin Chimbo
HU-5	19/03/2019	25/03/2019	30	Sandra Ordoñez
HU-5	26/03/2019	01/04/2019	30	Kevin Chimbo
HT-10	02/04/2019	08/04/2019	30	Sandra Ordoñez
SPRINT 8	**09/04/2019**	**06/05/2019**	**120**	
HT-11	09/04/2019	15/04/2019	30	Kevin Chimbo
HT-12	16/04/2019	22/04/2019	30	Sandra Ordoñez
HT-13	23/04/2019	29/04/2019	30	Kevin Chimbo
HT-14	30/04/2019	06/05/2019	30	Sandra Ordoñez

Realizado por: Sandra Ordoñez y Kevin Chimbo, 2019

2.7.3. Fase de desarrollo

El presente trabajo de titulación al tratarse del mantenimiento a un sistema, los estándares de codificación, el diseño de las interfaces y la base de datos ya se encuentran definidos, por lo cual se debe basar el perfeccionamiento de la aplicación en los aspectos ya propuestos por los anteriores desarrolladores.

Análisis de la base de datos

Para el diseño de las modificaciones a ejecutar, se realizó el análisis de la base de la base de datos, la cual está compuesta por .20 tablas, las mismas que fueron agregadas y/o actualizadas para cubrir las necesidades que generan las historias de usuario e historias técnicas del presente trabajo de titulación. El diagrama físico final del sistema se encuentra en el manual técnico registrado en el Planner.

Análisis de la arquitectura del sistema

Esta etapa se detalla mediante el diagrama de componentes el mismo que está compuesto por los módulos de las 3 capas (lógica de negocio, acceso a datos y base de datos) que se utilizó el proceso de desarrollo de software, además de los servicios web Unidos y UnidosAplicaciones, de los cuales

se obtiene información externa para el óptimo trabajo de las funcionalidades, esto se realizó para conocer las dependencias que conforman el sistema de Estafetas. El diagrama se lo realizó con el lenguaje de modelado unificado (UML) y se detalla en el **Anexo L**. Además, se analizó el diagrama de componentes propuesto en la anterior solución, el mismo que se mantiene por cuestiones de seguridad, este diagrama se encuentra en el manual técnico registrado en el Planner.

Implementación

En este diagrama se describe la organización de la aplicación, detallando las clases con los métodos y atributos que permiten la realización de las operaciones correspondientes, el mismo que está compuesto por 20 entidades, las cuales fueron actualizadas y/o agregadas para cubrir las necesidades de los requerimientos, este se detalla en el **Anexo K**.

En esta etapa, se desarrollaron y probaron las modificaciones realizadas en cada Sprint, las mismas que se documentaron mediante tareas de ingeniería de historias de usuario e historias técnicas, la información completa se la puede encontrar en el **Anexo J**.

Pruebas del sistema

La metodología utilizada en presente trabajo, plantea una comunicación continua con el cliente, para cumplir con ello, una vez por semana el grupo de desarrollo a cargo proyecto presenta avances con el objetivo de realizar pruebas sobre el sistema modificado, revisar integridad y obtener la aprobación del mismo mediante las pruebas de aceptación. Una vez finalizado el sprint se integró el software modificado de parte de los desarrolladores a cargo del proyecto; para la verificación se realizaron 45 pruebas de aceptación a nivel de las historias de usuario y 71 pruebas de aceptación a nivel de las tareas de ingeniería. Esta información se presenta en el **Anexo J**.

2.7.4. Fase de finalización

De acuerdo con la metodología aplicada se realizó pruebas sobre el sistema completamente integrado, para esto después de subir al sistema al servidor se evaluaron todas la funcionalidades de la aplicación

con el objetivo de comprobar el correcto funcionamiento de la misma; en cuanto a la liberación del producto se notificó al cliente el link para el acceso a la aplicación y se realizó una copia de seguridad en la plataforma Microsoft Planner, en la cual se encuentran todas las versiones del sistema, tareas realizadas, manual técnico y manual de usuario, esto para tener un registro de todas las versiones del proyecto realizado. Además, se debe realizar una capacitación de máximo 3 horas al personal que utilizara el sistema una vez este se encuentre en explotación para que los usuarios tengan una introducción al manejo de sistema, la misma que la complementarán con el manual de usuario presente en el Planner.

Seguimiento del riesgo

En el presente trabajo de titulación, se determinaron 5 riesgos, de los cuales el denominado "Funcionalidades mal desarrolladas" se convirtió en problema, todos los aspectos de reducción del mismo se planificaron correctamente y ayudaron a mitigar el riesgo que estaba valorado probabilidad, impacto y exposición alta; ubicándolo con prioridad de 1.

Reuniones

- Reuniones diarias, en las mismas se planifica las funcionalidades a desarrollar, se definieron las pruebas a realizar y en el último día del Sprint se planea la integración del software modificado.
- Reuniones de planificación, se realizarán una vez cada Sprint y se generó las actividades descritas a continuación.

  - Determinar objetos a modificar
  - Generar los casos de pruebas
  - Obtener lista de modificaciones.
  - Generar guía básica del diseño actualizado.
  - Obtener planes de pruebas actualizados.
  - Obtener análisis detallado actualizado de los requisitos verificados.
  - Generar lista de restricciones y riesgos documentados
- Reuniones de entrega, se las realiza una vez cada Sprint, en una de estas el cliente sugirió que se mejore el rendimiento de la aplicación ya que los datos se demoran en mostrarse en la interfaz. En cada reunión se generó las actividades descritas a continuación.

- Realizar pruebas sobre el sistema modificado
- Revisar integridad.
- Obtener aprobación.
- Liberar una nueva versión del producto.

- Reuniones de retroalimentación, se realizan después de cada Sprint y se generó las actividades tales como revisión de los errores generados durante las pruebas realizadas en las reuniones de entrega.

Las actividades anteriormente descritas en cada reunión se encuentran debidamente documentadas en el manual técnico y **Plan de pruebas** del sistema de Estafetas.

CAPÍTULO III

3. RESULTADOS Y DISCUSIÓN

La obtención de resultados se basa en la metodología definida en el capítulo anterior, en la cual se obtuvieron valores del comportamiento del sistema ante las pruebas realizadas.

En el cálculo del muestreo se determinó un total de 65 ataques a realizar, mediante un análisis de factibilidad presentado en una reunión del grupo de desarrollo se llegó al común acuerdo de realizar 77 ataques, con el objetivo de influir de manera positiva en el nivel de confianza del experimento y disminuir el margen de error.

3.1. Análisis de resultados antes de realizar mantenimiento

Los valores obtenidos en la ejecución de los ataques se presentan en la **Tabla 17-3**, descrita a continuación. Los valores detallados por funcionalidad se encuentran en el **Anexo C**.

Tabla 17-3 Resultado de los ataques antes del mantenimiento

Indicador	Ataque	No. Funcionalidades vulneradas	Porcentaje de vulnerabilidad
Confidencialidad	XSS	12	100,00%
	Intercepción criptográfica	1	0,00%
	SQL Injection	23	100,00%
	CSRF	17	100,00%
	Buffer overflow (DOS)	7	0,00%
Integridad	Directorio transversal	17	0,00%
	SQL Injection	23	100,00%
	CSRF	17	100,00%
	XSS	12	100,00%
	Buffer overflow (DOS)	7	0,00%
Disponibilidad	Directorio transversal	17	0,00%
	CSRF	17	100,00%
	Buffer overflow (DOS)	7	0,00%
	XSS	12	100,00%
	Intercepción criptográfica	1	0,00%

Realizado por: Sandra Ordoñez y Kevin Chimbo, 2019

Los resultados se encuentran en el **Gráfico 1-3**, los mismos que reflejan un 100% de vulnerabilidad en XSS, CSRF y SQL Injection, en cuanto a buffer overflow, intercepción criptográfica y directorio transversal se tiene un 0%.

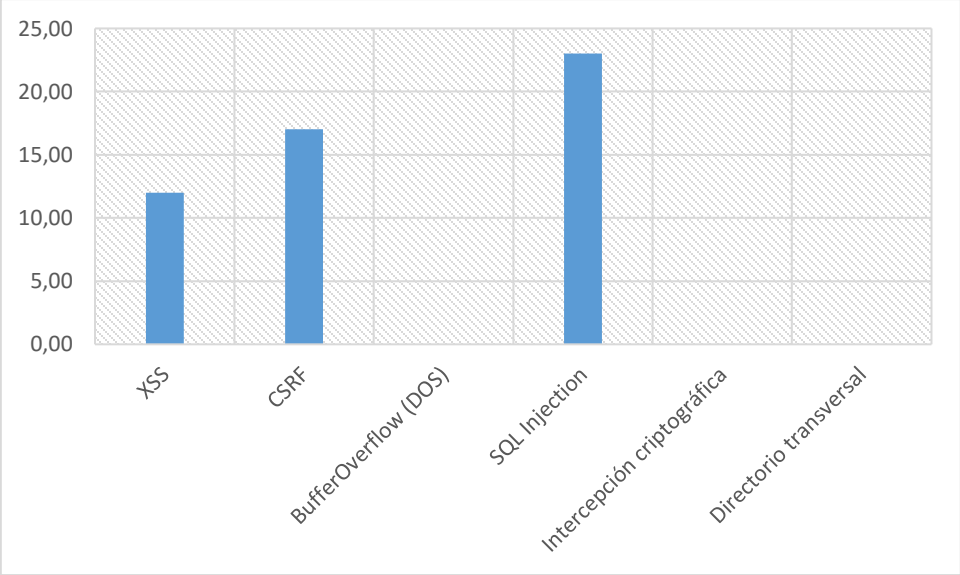


Gráfico 1-3 Ataques realizados al sistema antes del mantenimiento
Realizado por: Sandra Ordoñez y Kevin Chimbo, 2019

3.1.1. Datos estadísticos

Los resultados de los 6 ataques realizados se analizan mediante los parámetros de la estadística descriptiva para determinar la conducta ligado a la información obtenida.

Tabla 18-3 Estadística descriptiva del sistema sin buenas prácticas

Ataque	Funcionalidades vulnerables sin buenas prácticas (X)
XSS	12
CSRF	17
Buffer overflow (DOS)	0
SQL Injection	23
Intercepción criptográfica	0
Directorio transversal	0
Media	**8,67**

Varianza	**102,27**
Desviación estándar	**10,11**

Realizado por: Sandra Ordoñez y Kevin Chimbo, 2019

El resumen de la estadística descriptiva de los ataques realizados después de realizar el mantenimiento de la aplicación se presenta en la **Tabla 18-3**, dado que la desviación estándar de 10.11 tiene variación significativa con respecto a la media 8.67, los datos tienen muy poca concentración alrededor de la media.

3.2. Análisis de resultados después de realizar mantenimiento

Los valores obtenidos en la ejecución de los ataques se presentan en la **Tabla 19-3**, descrita a continuación. Los valores detallados por funcionalidad se encuentran en el **Anexo C**.

Tabla 19-3 Resultado de los ataques después del mantenimiento

Indicador	Ataque	No. Funcionalidades vulneradas	Porcentaje de vulnerabilidad
Confidencialidad	XSS	12	0,00%
	Intercepción criptográfica	1	0,00%
	SQL Injection	23	0,00%
	CSRF	17	100,00%
	Buffer overflow (DOS)	7	0,00%
Integridad	Directorio transversal	17	0,00%
	SQL Injection	23	0,00%
	CSRF	17	100,00%
	XSS	12	0,00%
	Buffer overflow (DOS)	7	0,00%
Disponibilidad	Directorio transversal	17	0,00%
	CSRF	17	100,00%
	Buffer overflow (DOS)	7	0,00%
	XSS	12	0,00%
	Intercepción criptográfica	1	0,00%

Realizado por: Sandra Ordoñez y Kevin Chimbo, 2019

Los resultados se encuentran en el **Gráfico 2-3**, los mismos que reflejan un 100% de vulnerabilidad en XSS, CSRF y SQL Injection, en cuanto a buffer overflow, intercepción criptográfica y directorio transversal se tiene un 0%.

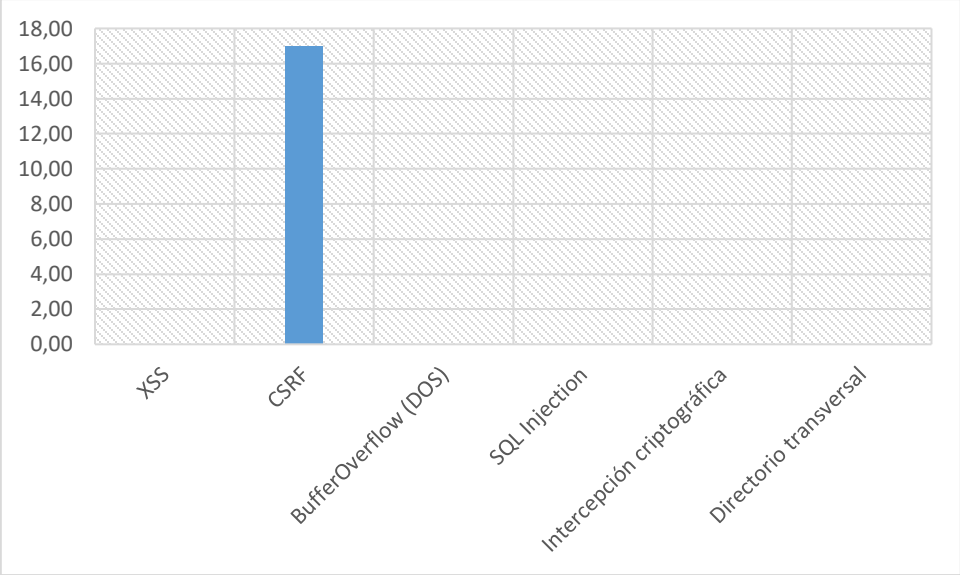


Gráfico 2-3 Ataques realizados al sistema de después del mantenimiento

Realizado por: Sandra Ordoñez y Kevin Chimbo, 2019

3.2.1. Datos estadísticos

Los resultados de los 6 ataques realizados se analizan mediante los parámetros de la estadística descriptiva para determinar la conducta ligado a la información obtenida.

Tabla 20-3 Estadística descriptiva del sistema scon buenas prácticas

Ataque	Funcionalidades vulnerables con buenas prácticas (Y)
XSS	0
CSRF	17
Buffer overflow (DOS)	0
SQL Injection	0
Intercepción criptográfica	0
Directorio transversal	0
Media	**2,83**

Varianza	**48,17**
Desviación estándar	**6,94**

Realizado por: Sandra Ordoñez y Kevin Chimbo, 2019

El resumen de la estadística descriptiva de los ataques realizados después de realizar el mantenimiento de la aplicación se presenta en la **Tabla 20-3**, dado que la desviación estándar de 6.94 tiene variación significativa con respecto a la media 2.83, los datos tienen muy poca concentración alrededor de la media.

3.3. Comparativo de resultados antes y después del mantenimiento

Confidencialidad e integridad

En el **Gráfico 3-3** se muestra la comparación entre la media del número de vulnerabilidades del sistema sin buenas prácticas de seguridad y con buenas prácticas de seguridad, donde se puede concluir que el mantenimiento mejoro disminuyo el número de vulnerabilidades en los indicadores de confidencialidad e integridad.

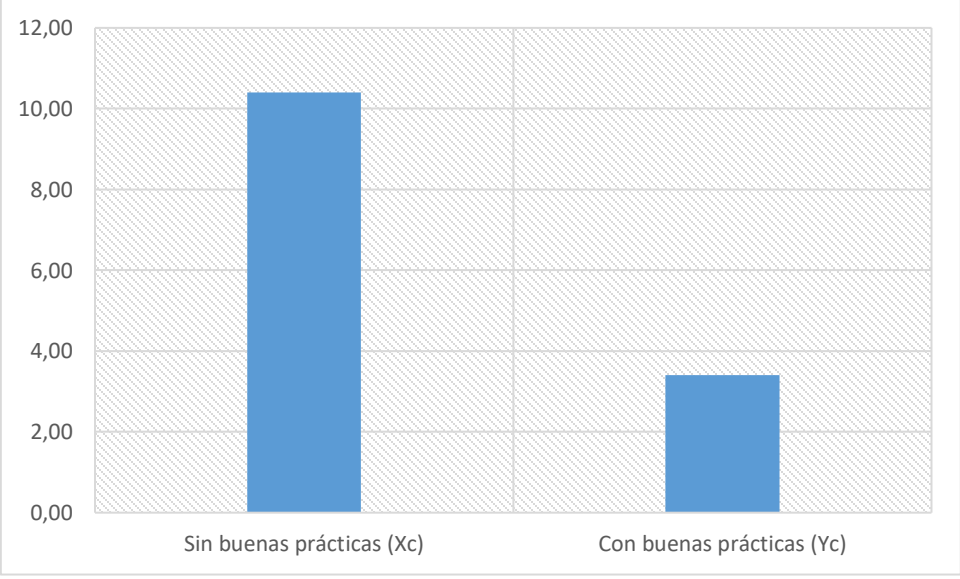


Gráfico 3-3 Confidencialidad e integridad antes y después del mantenimiento

Realizado por: Sandra Ordoñez y Kevin Chimbo, 2019

Disponibilidad

En el **Gráfico 4-3** se muestra la comparación entre la media del número de vulnerabilidades del sistema sin buenas prácticas de seguridad y con buenas prácticas de seguridad, donde se puede concluir que el mantenimiento disminuyo el número de vulnerabilidades en el indicador de disponibilidad.

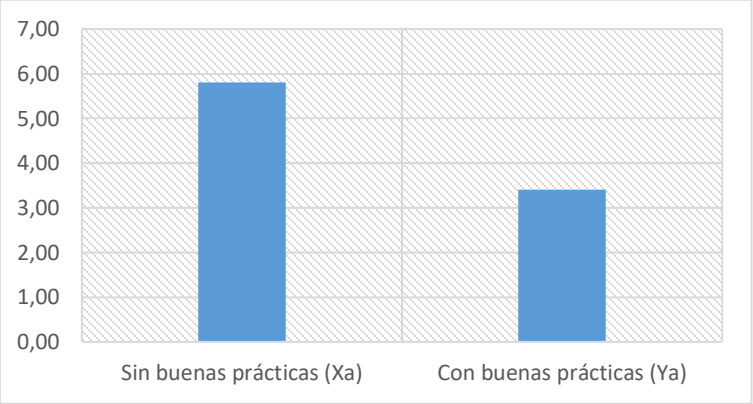


Gráfico 4-3 Disponibilidad antes y después del mantenimiento

Realizado por: Sandra Ordoñez y Kevin Chimbo, 2019

General

En el **Gráfico 5-3** se muestra la comparación entre la media del número de vulnerabilidades del sistema sin buenas prácticas de seguridad y con buenas prácticas de seguridad, donde se puede concluir que el mantenimiento influyó de manera positiva en los errores de seguridad del sistema.

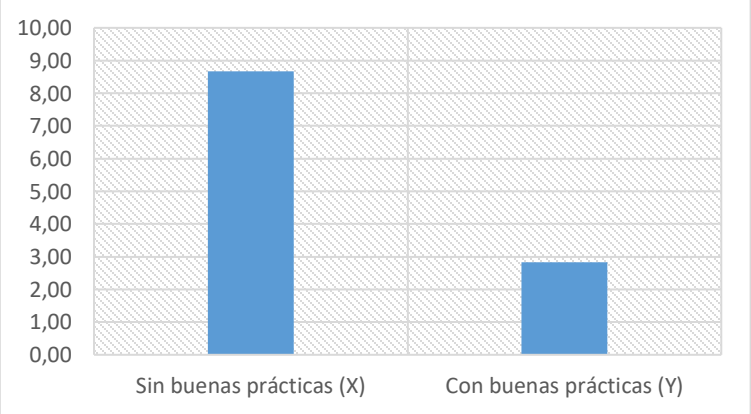


Gráfico 5-3 Resultados antes y después del mantenimiento

Realizado por: Sandra Ordoñez y Kevin Chimbo, 2019

En base a la definición de seguridad de la información según ISO 27001, es "*la preservación de la confidencialidad, integridad y disponibilidad de la información*". Entonces la seguridad será igual a la suma de la confidencialidad (C), Integridad (I) y la Disponibilidad (D), como lo determina (Paguay, 2015, pp.56-57).

En este trabajo, debido a su importancia, la amenaza a la integridad (I) representará el 40% de la seguridad, la amenaza a la confidencialidad (C) un 30% y la amenaza a la disponibilidad (D) un 30%, dando un total de la sumatoria de 100%.

$$Vulnerabilidad = C + I + D$$

Los resultados obtenidos de la aplicación de los ataques éticos sin aplicar el respectivo mantenimiento y con la aplicación del mismo, los valores se observan en la **Tabla 21-3** donde se concluye que el uso de buenas prácticas de seguridad disminuyó en un 34 % las vulnerabilidades del sistema de Estafetas.

Tabla 21-3 Resumen de resultados por indicador

Categoría		**Vulnerabilidad del sistema sin buenas prácticas (%)**		**Vulnerabilidad del sistema con buenas prácticas (%)**		**Disminución de vulnerabilidad**
Indicador	**Equivalencia**	**V. Real**	**V. Equivalente**	**V. Real**	**V. Equivalente**	
Confidencialidad	30%	60,00%	18,00%	20,00%	6,00%	40,00%
Integridad	40%	60,00%	24,00%	20,00%	8,00%	40,00%
Disponibilidad	30%	40,00%	12,00%	20,00%	6,00%	20,00%
TOTAL	**100%**		**54,00%**		**20,00%**	**34,00%**

Realizado por: Sandra Ordoñez y Kevin Chimbo, 2019

En el **Grafico 6-3**, se presenta los datos en un diagrama de barras en el cual se observa que el sistema de estafetas tiene una mejora significativa en los indicadores CIA, se percibe que existe una mejora del 40% para confidencialidad e integridad y del 20% para el indicador de disponibilidad.

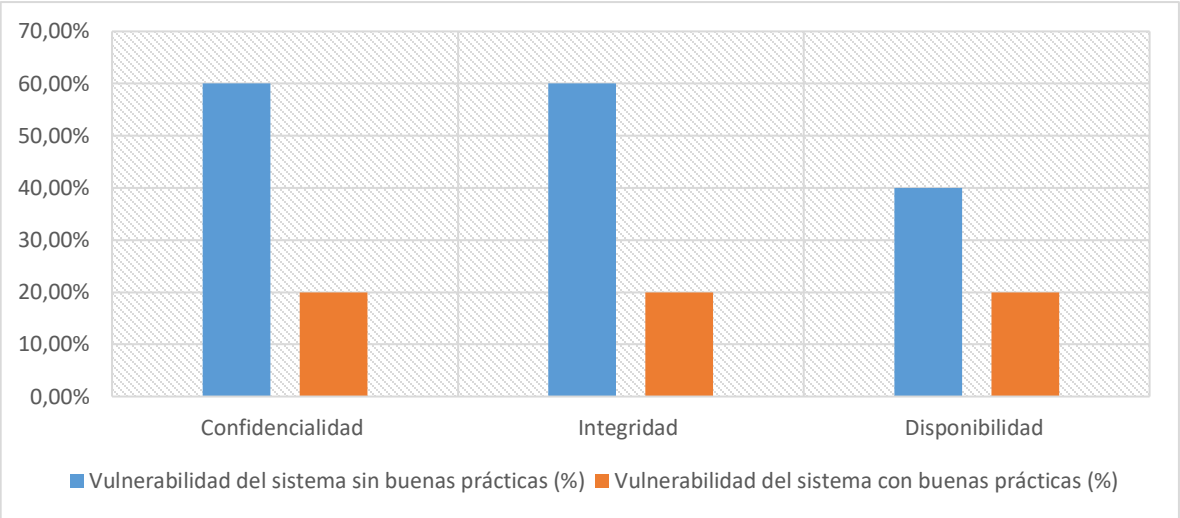


Gráfico 6-3 Resultados por indicador

Realizado por: Sandra Ordoñez y Kevin Chimbo, 2019

3.3.1. Prueba chi-cuadrado (x^2)

Para determinar si hay mejora en el sistema de manera general y para los indicadores confidencialidad e integridad se establece la tabla de contingencia para la prueba chi-cuadrado, la misma que se realizó mediante frecuencias representando la cantidad de funcionalidades vulnerables ante cierto ataque antes y después de realizar el correspondiente mantenimiento. Además, no se tomaron en cuenta los ataques que no vulneraban las funcionalidades porque su estudio no tiene incidencia en el cálculo del valor de la prueba estadística. El detalle de los datos se encuentra en la **Tabla 22-3**.

Tabla 22-3 Tabla de contingencia (Confidencialidad, integridad y general)

	Funcionalidades vulneradas sin buenas prácticas (X)	**Funcionalidades vulneradas con buenas prácticas (Y)**
XSS	12,00	0,00
CSRF	17,00	17,00
SQL Injection	23,00	0,00

Realizado por: Sandra Ordoñez y Kevin Chimbo, 2019

En la **Tabla 22-3**, se calculó el valor de chi-cuadrado es x^2(calculado) = 23.22 con 2 grados de libertad y un nivel de significancia del 5%, x^2(calculado) > 5.9915 por lo que se rechaza la H0 y se acepta la

H1, es decir que el uso de buenas prácticas de seguridad en el mantenimiento, influyen en el nivel de vulnerabilidad del sistema de estafetas.

Tabla 23-3 Prueba chi-cuadrado (Confidencialidad, integridad y general)

Prueba chi cuadrado (x^2)		
Parámetro	**X**	**Y**
Media	17,3333	5,6667
Varianza	30,3333	96,3333
Observaciones	3,0000	3,0000
Alfa	0,0500	
Grados de libertad	2,0000	
Estadístico x^2 (calculado)	23,2212	
Estadístico x^2 (tabla)	5,9915	

Realizado por: Sandra Ordoñez y Kevin Chimbo, 2019

En la **Figura 14-3**, se presenta el análisis de los resultados obtenidos en la prueba chi-cuadrado de manera gráfica, donde se reafirma que la aceptación de la hipótesis H1.

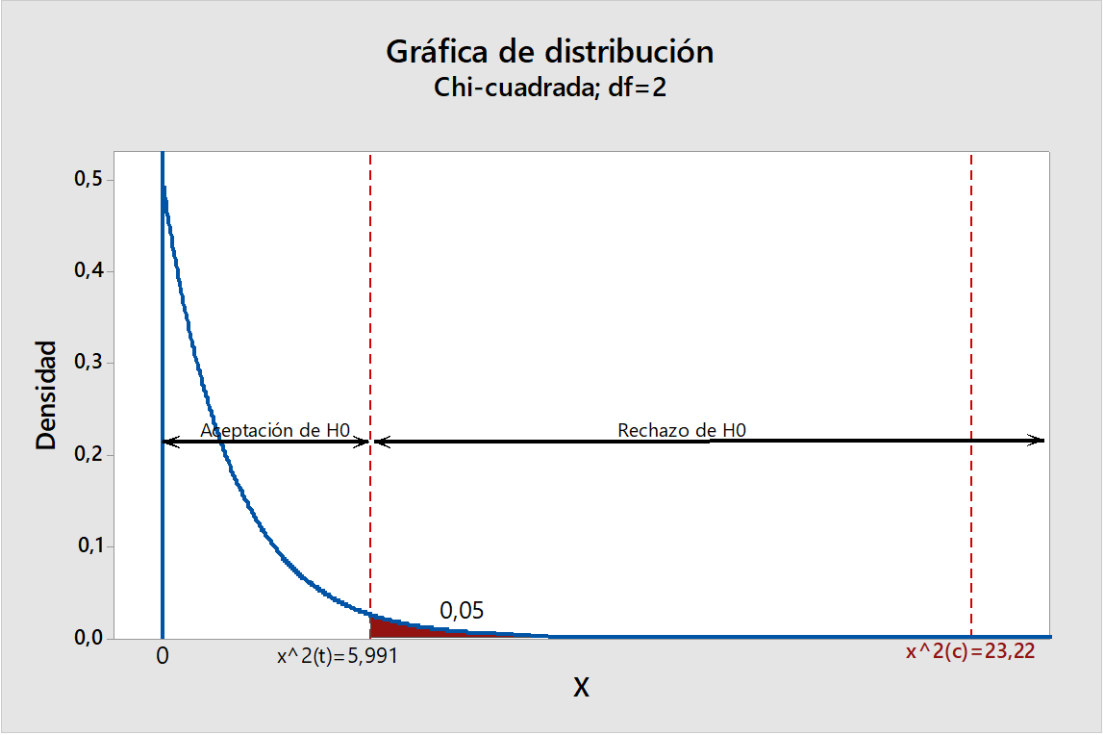


Figura 14-3 Distribución chi-cuadrado (Confidencialidad, integridad y general)

Realizado por: Sandra Ordoñez y Kevin Chimbo, 2019

Para determinar si hay mejora en el sistema para el indicador disponibilidad también se estableció la tabla de contingencia para la prueba chi-cuadrado, la misma que se realizó mediante frecuencias representando la cantidad de funcionalidades vulnerables ante cierto ataque antes de después de realizar el correspondiente mantenimiento. Además, no se tomaron en cuenta los ataques que no vulneraban las funcionalidades porque su estudio no tiene incidencia en el cálculo del valor de la prueba estadística. El detalle de los datos se encuentra en la **Tabla 24-3**.

Tabla 24-3 Tabla de contingencia (Disponibilidad)

	Funcionalidades vulnerables sin buenas prácticas (X)	**Funcionalidades vulnerables con buenas prácticas (Y)**
XSS	12,00	0,00
CSRF	17,00	17,00

Realizado por: Sandra Ordoñez y Kevin Chimbo, 2019

En la **Tabla 25-3**, se calculó el valor de chi-cuadrado es x^2(calculado) = 9.517 con 1 grado de libertad y un nivel de significancia del 5%, x^2(calculado) > 3.8415 por lo que se rechaza la H0 y se acepta la H1, es decir que el uso de buenas prácticas de seguridad en el mantenimiento, influyen en el nivel de vulnerabilidad del sistema de estafetas.

Tabla 25-3 Prueba chi-cuadrado (Disponibilidad)

Prueba chi cuadrado (x^2)		
Parámetro	**X**	**Y**
Media	14,5000	8,5000
Varianza	12,5000	144,5000
Observaciones	2,0000	2,0000
Alfa	0,0500	
Grados de libertad	1,0000	
Estadístico x^2 (calculado)	9,5172	
Estadístico x^2 (tabla)	3,8415	

Realizado por: Sandra Ordoñez y Kevin Chimbo, 2019

En la **Figura 15-3**, se presenta el análisis de los resultados obtenidos en la prueba chi-cuadrado de manera gráfica, donde se reafirma que la aceptación de la hipótesis H1.

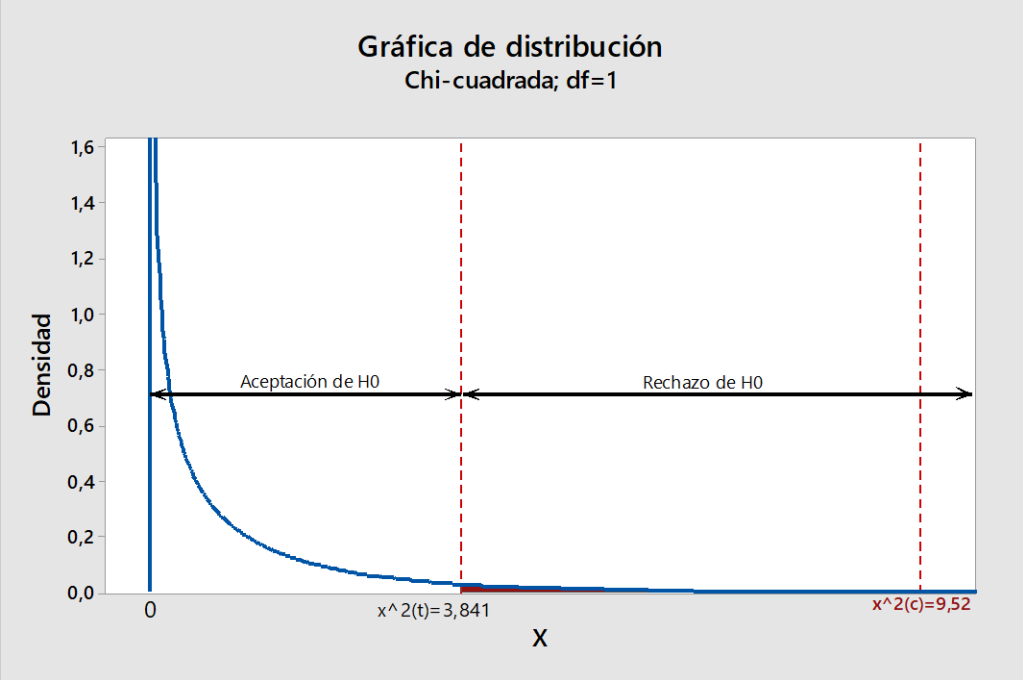


Figura 15-3 Distribución chi-cuadrado (Disponibilidad)

Realizado por: Sandra Ordoñez y Kevin Chimbo, 2019

CONCLUSIONES

- Se creó el manual de buenas prácticas en el cual están presentes las 10 técnicas de seguridad para 6 ataques analizados en aplicaciones web desarrolladas en Java, las mismas que permiten asegurar que la información no sea manipulada o extraída de la base de datos y así evitar problemas que se puedan suscitar por falta de datos o espionaje.
- Se determinó que el nivel inicial de vulnerabilidad del sistema de Estafetas es del 54%, mediante la aplicación de 6 ataques en las 23 funcionalidades que componen el sistema de Estafetas en función a los parámetros CIA definidos por el estándar ISO 27001.
- Se dio mantenimiento al sistema de estafetas, para esto se corrigieron y se aplicaron las buenas prácticas propuestas a las 23 funcionalidades ya desarrolladas y se agregaron 18; obteniendo un total de 41 requerimientos desarrollados.
- Se determinó que el nivel final de vulnerabilidad del sistema de Estafetas es de 20%, mediante la aplicación de 6 ataques en las 23 funcionalidades que formar parte del sistema de Estafetas en función a los parámetros CIA definidos por el estándar ISO 27001.
- Se redujo en un 12% el valor inicial de vulnerabilidad en el indicador confidencialidad.
- Se redujo en un 16% el valor inicial de vulnerabilidad en el indicador integridad
- Se redujo en un 6% el valor inicial de vulnerabilidad en el indicador disponibilidad
- Se influyó en el nivel de vulnerabilidad, ya que se disminuyó en un 34% las vulnerabilidades existentes; con un nivel de confianza del 95% y un marcen de error del 5% mediante el uso de la técnica estadística chi-cuadrado.

RECOMENDACIONES

- Se recomienda el uso del manual de buenas prácticas para el desarrollo de aplicaciones web desarrolladas en Java.
- Se sugiere investigar nuevos métodos de pruebas de penetración para revelar vulnerabilidades que no se hayan evaluado en este trabajo de titulación, puesto que a medida que avanza el tiempo se generan más amenazas a los sistemas informáticos.
- Se recomienda continuar con el mantenimiento del sistema de estafetas para agregar nuevas funcionalidades que el cliente requiera, utilizando las actividades asociadas al estándar IEEE 1219, ya que las mismas permiten realizar correctamente las tareas del proceso de mejora del sistema.
- Se sugiere investigar una técnica que permita contrarrestar los ataques CSRF, ya que la que se aplicó no ayudó a disminuir la vulnerabilidad del sistema en dicho ataque, puesto que el procedimiento aplicado no se encontraba actualizado porque su último registro fue en el año 2017.
- Actualizar el manual de buenas prácticas con nuevas técnicas que permitan influir en las diferentes vulnerabilidades existentes en las aplicaciones web con el objetivo de mejorar la seguridad en los sistemas informáticos.
- Investigar acerca del rendimiento del software para que pueda ser aplicado al sistema de Estafetas con el objetivo de mejorar los tiempos de respuestas en la aplicación.

BIBLIOGRAFÍA

ACOSTA, R. E. Y ISAZA, G. A. *Hacia un arquitectura de buenas prácticas de seguridad para sistemas ERP Towards an architecture of good practices for ERP safety systems*, 2010. Disponible en: http://vip.ucaldas.edu.co/vector/downloads/Vector5_6.pdf (Accedido: 6 de mayo de 2019).

AGUILERA LÓPEZ, P. *Seguridad informática*. Editex, 2010.

ALFONZO, P., MARIÑO, S. Y GODOY, M. «Propuesta de aplicación de SCRUM para gestionar el proceso de mantenimiento del software: estudio preliminar», en *20th IEEE International Conference on Software Maintenance. Proceedings, 2012*. Corrientes: IEEE, pp. 522-522. doi: 10.1109/ICSM.2004.1357864.

BALUJA GARCÍA, W. Y PORVÉN RUBIER, J. *Gestión automatizada e integrada de controles de seguridad informática, Ingeniería Electrónica, Automática y Comunicaciones*. Facultad de Ingeniería Eléctrica, Instituto Superior Politécnico José Antonio Echeverría, 2013. Disponible en: http://scielo.sld.cu/scielo.php?pid=S1815-59282013000100004&script=sci_arttext&tlng=pt (Accedido: 24 de octubre de 2018).

CABALLERO, A. *Hacking con Kali Linux, 2018*. Disponible en: http://www.reydes.com. (Accedido: 9 de mayo de 2019).

CEVALLOS MUÑOZ, F. D. Y SIGUENZA PLAZA, Á. J. «Propuesta de Mejores Prácticas de Seguridad para el Desarrollo de Aplicaciones Móviles.» Escuela Superior Politécnica de Chimborazo, 2016. Disponible en: http://dspace.espoch.edu.ec/handle/123456789/4762 (Accedido: 6 de mayo de 2019).

CHICANO, E. *Auditoría de seguridad informática*. ic editori, 2014. Disponible en: https://books.google.com.ec/books?hl=es&lr=&id=8a3KCQAAQBAJ&oi=fnd&pg=PT4&dq=auditoria+de+seguridad+informatica+ester+chicano&ots=ja1hJDQ1Qs&sig=I1CsAoGuoIUg73ueH_UlqnxvejY&redir_esc=y#v=onepage&q=vulnerabilidad&f=false (Accedido: 25 de octubre de 2018).

CISCO SYSTEMS *Soluciones de seguridad para redes empresariales y Cisco DNA - Cisco, 2018*. Disponible en: https://www.cisco.com/c/es_ec/solutions/enterprise-networks/enterprise-network-security/index.html?CCID=cc000009&DTID=pseggl000015&POSITION=SEM&COUNTRY_SITE=ec&CAMPAIGN=sc-00&CREATIVE=EC_SEM_SEC_Security-SPA_PM_NB-Seguridad-

&REFERRING_SITE=Google&KEYWORD= (Accedido: 6 de mayo de 2019).

ELIÉCER, J. *ET AL.* «Delitos Informaticos», *Delitos informaticos*, 11(28), pp. 41-66, 2010. Disponible en: http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0123-14722010000200003 (Accedido: 6 de mayo de 2019).

ESPOCH *Dirección de Tecnologías de la Información y Comunicación - Escuela Superior Politécnica de Chimborazo, 2017*. Disponible en: https://www.espoch.edu.ec/index.php/direccion-de-tecnologias-de-la-informacion-y-comunicacion.html (Accedido: 6 de mayo de 2019).

FAJARDO, J. *Estrategia de ciberseguridad, 2018*. Disponible en: https://www.b-secure.co/blog/estrategia-de-ciberseguridad (Accedido: 6 de mayo de 2019).

FLUIDATTACKS *Evitar Ataques de Directorio Transversal, 2018*. Disponible en: https://fluidattacks.com/web/es/defends/csharp/evitar-direct-transversal/ (Accedido: 6 de mayo de 2019).

FONTALVO, V. «Conceptos basicos en buenas practicas de TI y seguridad informatica», 2019. Disponible en: https://www.academia.edu/11494593/Conceptos_basicos_en_buenas_practicas_de_TI_y_seguridad_informatica?auto=download (Accedido: 7 de mayo de 2019).

GÓMEZ VIEITES, A. *Enciclopedia de la seguridad informática*. Ra-Ma, 2011.

GÓMEZ VIEITES, Á. *REDES INFORMÁTICAS, 2014*. Disponible en: www.keylogger.com (Accedido: 6 de mayo de 2019).

IEEE, S. *The IEEE standard for software maintenance, 1998, ACM SIGSOFT Software Engineering Notes*. doi: 10.1145/181610.181623.

ISOTOOLS *ISO 27001,* 2019. Disponible en: https://www.isotools.org/normas/riesgos-y-seguridad/iso-27001/ (Accedido: 22 de abril de 2019).

JIMENO GARCÍA, M. T. *La biblia del hacker edición 2012.* Anaya Multimedia, 2011.

LÓPEZ, R. (2017) «PRUEBAS DE PENETRACIÓN EN APLICACIONES WEB USANDO HACKEO ÉTICO», *REVISTA TECNOLÓGICA*, 10, pp. 13-19.

MARCO, L. Y BERMÚDEZ BENÍTEZ, A. *BUENAS PRÁCTICAS DE SEGURIDAD Y SALUD OCUPACIONAL Región Oriental de Salud Oriental,* 2018. Disponible en: https://www.salud.gob.sv/archivos/pdf/seguridad_ocupacional_2018_presentaciones/presentacion1

3092018/BUENAS-PRACTICAS-DE-SSO-REGION-ORIENTAL-DE-SALUD.pdf (Accedido: 7 de mayo de 2019).

MDEX *Seguridad informática ISO 27001*, 2019. Disponible en: https://www.mdex.de/en/solutions/iso-27001/ (Accedido: 2 de mayo de 2019).

MIERES, J. «Ataques informáticos», *Evil-Fingers*, p. 17, 2009. doi: 10.1016/j.geomorph.2005.07.005.

MONAR MONAR, J. S. *et al.* «Técnicas de programación segura para mitigar vulnerabilidades en aplicaciones web», *Congreso de Ciencia y Tecnología ESPE*, 13(1), pp. 129-132, 2018. doi: 10.24133/cctespe.v13i1.753.

MONTES, M. *Las principales vulnerabilidades web - Hacking Ético*,2017. Disponible en: https://hacking-etico.com/2017/04/04/las-principales-vulnerabilidades-web/ (Accedido: 5 de mayo de 2019).

OPTICAL NETWORKS *Seguridad informátoca*, 2019. Disponible en: https://www.optical.pe/wp-content/uploads/2018/09/SEGURIDAD-INFORMATICA-OPTICAL-e1537818737972.png (Accedido: 22 de abril de 2019).

OWASP FOUNDATION «OWASP Top 10 -2017», p. 25, 2017. Disponible en: https://www.owasp.org/images/5/5e/OWASP-Top-10-2017-es.pdf (Accedido: 4 de noviembre de 2018).

OWASP FOUNDATION *Falsificación de solicitudes entre sitios (CSRF) - OWASP*, 2018. Disponible en: https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF) (Accedido: 6 de mayo de 2019).

PAGUAY, P. *Propuesta de técnicas de aseguramiento de aplicaciones web desarrolladas en Java*, 2015. Escuela Superior Politécnica de Chimborazo. Disponible en: http://dspace.espoch.edu.ec/handle/123456789/4029.

PARRA, C. Y PORRAS, H. «LAS AMENAZAS INFORMÁTICAS:PELIGRO LATENTE PARA LAS ORGANIZACIONES ACTUALES», *Revista Gerencia Tecnológica Informática*, 6(16), pp. 85-97, 2017. Disponible en: https://revistas.uis.edu.co/index.php/revistagti/article/view/1259/1656 (Accedido: 25 de octubre de 2018).

PÉREZ CARVAJAL, R. J. *Mantenimiento del software (UF1894).* IC Editorial, 2014. Disponible en:

https://books.google.com.ec/books?id=UoPhCgAAQBAJ&printsec=frontcover&dq=mantenimiento+del+software&hl=es&sa=X&ved=0ahUKEwjp09mNrqPeAhXSmOAKHW57DtEQ6AEIJzAA#v=onepage&q=mantenimiento del software&f=false (Accedido: 26 de octubre de 2018).

POZO, D. *ET AL.* «Procedimiento para pruebas de intrusión en aplicaciones Web Procedure for intrusion testing for Web applications», *REICIS*, 5(2), pp. 31-38, 2009. Disponible en: https://www.redalyc.org/pdf/922/92217153010.pdf (Accedido: 5 de mayo de 2019).

PRESSMAN, R. S. *Ingenieria del Software. Un Enfoque Practico*, 2010. Disponible en: www.FreeLibros.me (Accedido: 29 de octubre de 2018).

RAMOS, L. *PRUEBAS DE PENETRACIÓN O PENT TEST*, 2017. Disponible en: http://tenable.com/ (Accedido: 5 de mayo de 2019).

RUIZ, F., POLO, M. Y ALARCOS, G. *Mantenimiento del Software*, 2001. Disponible en: http://alarcos.inf-cr.uclm.es/doc/mso/ (Accedido: 7 de mayo de 2019).

SGSI *Seguridad de la Información CIA (Confidencialidad, Integridad, Disponibilidad)*, 2017. Disponible en: https://www.pmg-ssi.com/2017/07/cia-confidencialidad-integridad-disponibilidad-seguridad-de-la-informacion/ (Accedido: 21 de enero de 2019).

SOLUTECSA *Hacking ético*, 2018. Disponible en: https://www.internetglosario.com/1131/Hackingetico.html (Accedido: 8 de mayo de 2019).

SOTO, M. *¿Qué es XSS?*, 2016. Disponible en: https://medium.com/@marvin.soto/qué-es-xss-b9330eedbc07 (Accedido: 5 de mayo de 2019).

TECHOPEDIA *What is Vulnerability?*, 2018. Disponible en: https://www.techopedia.com/definition/13484/vulnerability (Accedido: 26 de octubre de 2018).

UNIVERSIDAD DE LA RIOJA *Seguridad en el diseño de las aplicaciones web*, 2017. Disponible en: https://campusescueladeingenieria.unir.net/cursos/lecciones/ARCHIVOS_COMUNES/versiones_para_imprimir/ingsw12/tema3.pdf (Accedido: 4 de noviembre de 2018).

UNMSM, U. N. M. DE S. M. *Concepto de Buenas Prácticas*, 2015. Disponible en: http://gestionensalud.medicina.unmsm.edu.pe/wp-content/uploads/2015/08/MS_RB_08_Concepto_Buenas_Practicas.pdf (Accedido: 7 de mayo de 2019).

VÉLEZ, S. *Funcionamiento del ataque informático*, 2019. Disponible en: https://velezconde.files.wordpress.com/2010/10/pantallazo.png (Accedido: 2 de mayo de 2019).

VELOZ, G. *Resumen ejecutivo SDA*, 2018.

ANEXOS

Anexo A: Plan de pruebas de penetración

DESARROLLO ACADÉMICO

PLAN DE PRUEBAS Sistema de Estafetas

"Saber para ser"

Riobamba – Ecuador

2018

PLAN DE PRUEBAS DE SEGURIDAD
SISTEMA DE ESTAFETAS

HISTORIAL DE CAMBIOS

Fecha	Modificado por	Versión	Descripción
DD/MM/AA	(Nombre quien realizo la modificación)	0.0	(Especificar de forma clara y concisa el cambio en el Plan de Pruebas)
19/10/2018	Sandra Ordoñez, Kevin Chimbo	1.0	Organización del documento
19/10/2018	Sandra Ordoñez, Kevin Chimbo	1.1	Sección 1
24/10/2018	Sandra Ordoñez, Kevin Chimbo	1.2	Secciones 1 y 2
03/12/2018	Sandra Ordoñez, Kevin Chimbo	1.3	Secciones 3 y 4
17/04/2019	Sandra Ordoñez, Kevin Chimbo	1.4	Secciones restantes y Pruebas y Registro
10/05/2019	Sandra Ordoñez, Kevin Chimbo	1.5	Sección 6, Pruebas y Registro
13/05/2019	Sandra Ordoñez, Kevin Chimbo	1.6	Pruebas y Registro

Tabla 1: Historial de cambios

TABLA DE CONTENIDO

ÍNDICE DE TABLAS

PLAN DE PRUEBAS DE SEGURIDAD
SISTEMA DE ESTAFETAS

INTRODUCCIÓN

La seguridad de datos se refiere a medidas de protección de la privacidad digital que se aplican para evitar el acceso no autorizado a los datos**,** los cuales pueden encontrarse en ordenadores, bases de datos, sitios web, etc. La seguridad de datos también protege los datos de una posible corrupción.

Los hackers suelen analizar las redes de forma activa o pasiva en busca de agujeros y vulnerabilidades. Los analistas de seguridad de datos y los profesionales de la evaluación de vulnerabilidades son elementos clave en la identificación de posibles agujeros y en cerrarlos. El software de análisis de seguridad se utiliza para aprovechar cualquier vulnerabilidad de un ordenador, red o infraestructura de comunicaciones, priorizando y abordando cada uno de ellos con planes de seguridad de datos que protegen, detectan y reaccionan. [1]

Se documentará este plan de pruebas para identificar las vulnerabilidades en el sistema de estafetas y conocer el estado inicial de la seguridad de dicho sistema.

1.1. OBJETIVOS

El plan de pruebas de seguridad se elabora con el fin de especificar contra qué ataques es vulnerable el sistema de estafetas. Además, a través del plan de pruebas de seguridad se puede continuar identificando cuanto se han reducido las vulnerabilidades en el sistema.

Al desarrollar el plan de seguridad se realizan las correcciones pertinentes según el caso y se garantiza la seguridad del producto que se está entregando al cliente. El plan de seguridad se aplica sobre el producto, los resultados de las pruebas son registrados en un formato que se encuentra en el Anexo 1.

1.2. ESTRATEGIA DE PRUEBAS

A través de los documentos realizados, se pretende retomar la información relacionada con las pruebas, para garantizar la seguridad de estas y del producto. Además, le permite al responsable de las pruebas saber exactamente los criterios que se deben tener en cuenta para probar cada funcionalidad del sistema.

1.3. ALCANCE

Teniendo en cuenta los documentos hechos anteriormente, el grupo de trabajo pretende realizar las pruebas, de manera incremental, por tipo de vulnerabilidad. Para una mejor comprensión, ver la ilustración 1: Alcance del plan de pruebas, la cual muestra el alcance y el orden en que se realizaran.

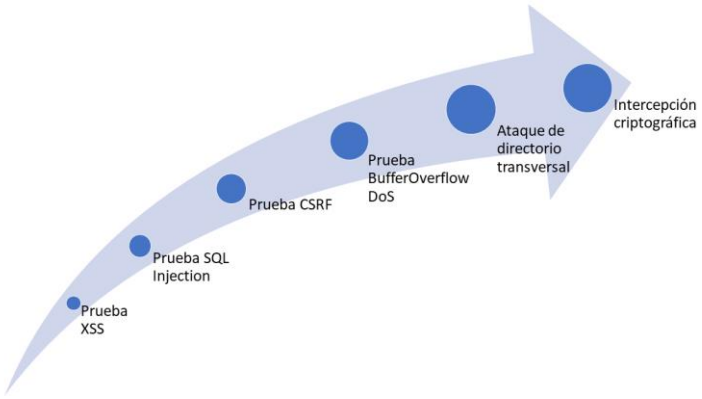


Ilustración 1: Alcance del plan de pruebas

Las pruebas de seguridad se realizan después de haber implementado la aplicación.

1.4. REFERENCIAS

[1] «PowerData,» [En línea]. Available: https://www.powerdata.es/seguridad-de-datos. [Último acceso: 4 Enero 2019].

[2] I. Pérez, «welivesecurity,» 5 Noviembre 2014. [En línea]. Available: https://www.welivesecurity.com/la-es/2014/11/05/como-funcionan-buffer-overflow/. [Último acceso: 4 Enero 2019].

[3] «EvadaliaHosting,» 16 Septiembre 2015. [En línea]. Available: https://blog.evidaliahost.com/cross-site-request-forgery-csrf/. [Último acceso: 4 Enero 2019].

[4] A. A. Domínguez, «UNAM,» 21 Agosto 2015. [En línea]. Available: https://www.seguridad.unam.mx/historico/documento/index.html-id=35. [Último acceso: 4 Enero 2019].

[5] P. C. Redondo, «OpenWebinars,» 12 Octubre 2017. [En línea]. Available: https://openwebinars.net/blog/que-es-sql-injection/. [Último acceso: 4 Enero 2019].

[6] «Behiqui Digital,» [En línea]. Available: La vulnerabilidad de Directorio Transversal (más conocida por Directory o Path Traversal), ocurre cuando no hay una gestión correcta (validación, autorización) de los parámetros provenientes del lado del cliente, específicamente aquellas relacionadas con . [Último acceso: 4 Enero 2019].

[7] «IEEE Computer Society,» [En línea]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=573169. [Último acceso: 4 Enero 2019].

[8] oscarcenteno, «Software Mantenible.com,» 6 Septiembre 2016. [En línea]. Available: https://softwaremantenible.com/2016/09/06/pruebas-unitarias-definicion-y-caracteristicas/. [Último acceso: 4 Enero 2019].

[9] «Mnuel.Cillero.es,» [En línea]. Available: https://manuel.cillero.es/doc/metrica-3/tecnicas/pruebas/integracion/. [Último acceso: 4 Enero 2019].

1.5. DEFINICIONES, ABREVIACIONES Y ACRÓNIMOS

CONCEPTO	DESCRIPCIÓN
XSS	Cross-Site Scripting
CSRF	Cross-Site Request Forgery
SO	Sistema Operativo

Tabla 2: Definición, abreviaciones y acrónimos

2. ARTEFACTOS DE PRUEBA

2.1. MÓDULOS DEL PROGRAMA

En esta sección se muestran los módulos que se pretenden probar, además de las especificaciones de las pruebas de seguridad a realizar en cada uno. Cabe notar, que cada módulo representa un componente del sistema.

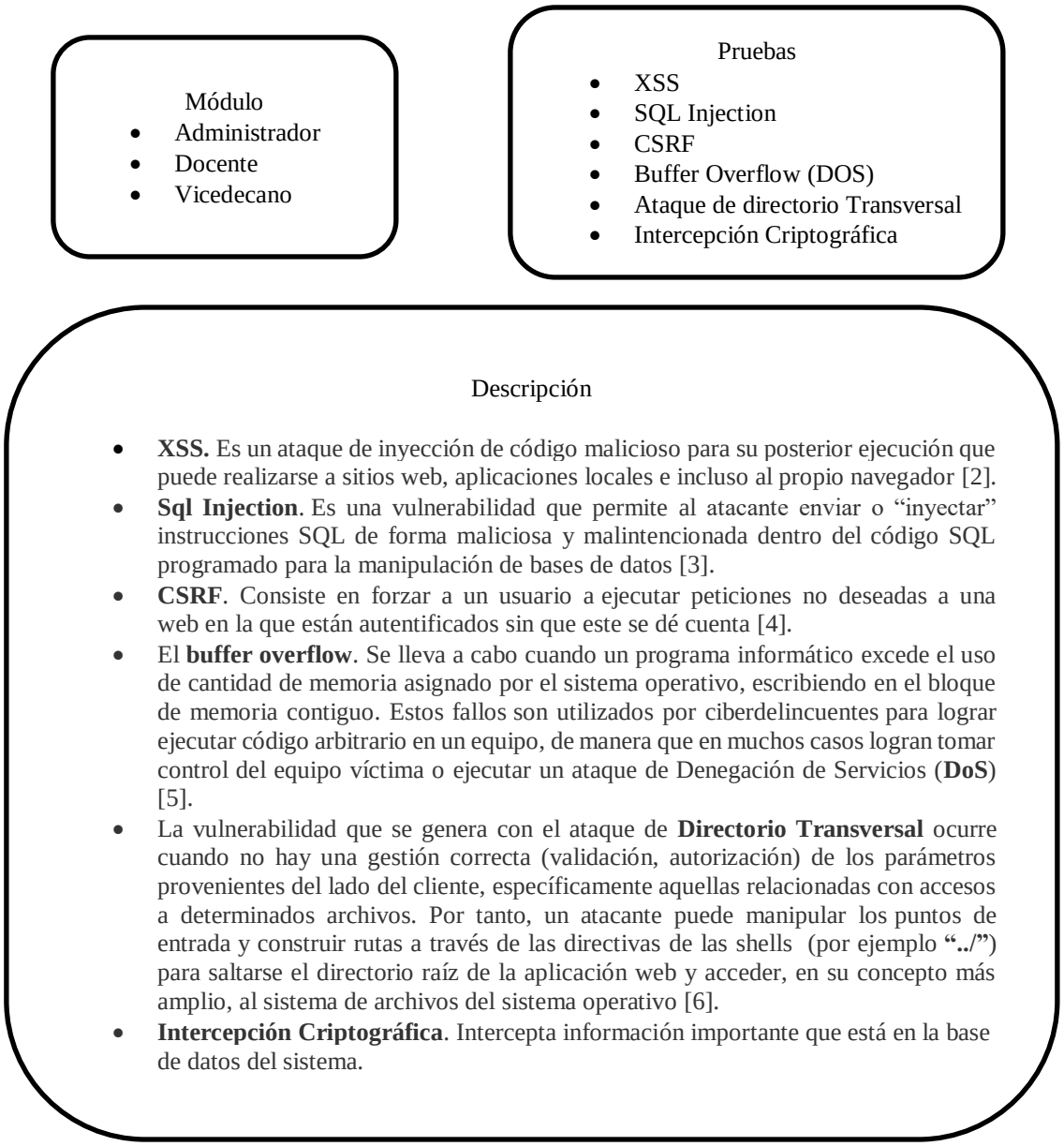


Ilustración 2: Módulos del programa

En esta sesión se describen los módulos a utilizar:

Modulo	Pruebas	Descripción
GUI	Seguridad	Se debe incorporar en el sistema seguridad para asegurar el desarrollo de cualquier tipo de funcionalidad
Lógica de Negocio	Funcionalidad	En el sistema se debe poder realizar todos los requerimientos establecidos con el cliente

Tabla 3: Módulos a probar en el sistema

2.2. PROCEDIMIENTOS DE USUARIO

Para utilizar el sistema informático de manera adecuada se necesitan guías manuales que sean claros, correctos, completos y coherentes, para que de esta manera el usuario pueda utilizar el sistema informático de manera correcta y pueda comprender los conceptos tras la funcionalidad. A continuación, se muestran los atributos de calidad de estos procedimientos [7].

- **Clara**: Las instrucciones proporcionadas en el documento, deben ser bien claras para que el usuario no tenga problemas al momento de manejar el sistema informático.
- **Correcta:** No existen errores semánticos, sintácticos, ortográficos ni de enlace dentro de la documentación proporcionada al usuario.
- **Completa:** La información debe estar completa desde la parte técnica hasta la parte funcional.
- **Coherente:** No existen ambigüedades, o congruencias que puedan confundir al usuario.

Los documentos a entregar con el sistema informático son:

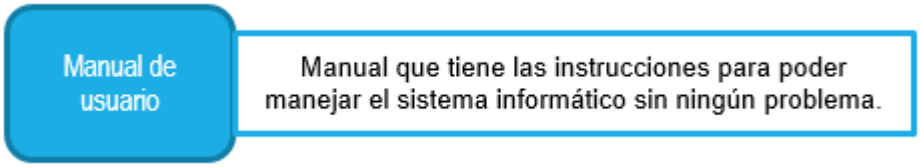


Ilustración 3: Documentos y Responsables

3. CARACTERÍSTICAS A SER PROBADAS

En esta sección se encuentran las características del sistema informático a ser probadas con un caso de estudio específico.

CARACTERÍSTICA	DESCRIPCIÓN	MODULO
Requerimientos Funcionales	Se debe tener en cuenta el criterio de aceptación y dependencias, para realizar pruebas en los módulos.	Los componentes donde se pueden probar son: • GUI • Lógica de Negocios • Acceso a Datos
Requerimientos No Funcionales	Seguridad	Todos los módulos del sistema

Tabla 4: Características a ser probadas.

4. CARACTERÍSTICAS A NO SER PROBADAS

En esta sección se encuentran las características del sistema informático que no serán probadas con un caso de estudio específico.

CARACTERÍSTICA	DESCRIPCIÓN
Pruebas Unitarias	Una prueba unitaria es un método que puede invocar al código que queremos probar y determina si el resultado obtenido es el esperado. Si es igual, entonces la prueba es exitosa, si no, falla [8].
Pruebas de Integración	En las pruebas de integración se examinan las interfaces entre grupos de componentes o subsistemas para asegurar que son llamados cuando es necesario y que los datos o mensajes que se transmiten son los requeridos [9].
Pruebas de Sistema	Las pruebas del sistema tienen como objetivo ejercitar profundamente el sistema comprobando la integración del sistema de información globalmente, verificando el funcionamiento correcto de las interfaces entre los distintos subsistemas que lo componen y con el resto de sistemas de información con los que se comunica [10].

Tabla 5: Características a no ser probadas.

5. APROXIMACIÓN

5.1. PRUEBAS DE SEGURIDAD

Las pruebas de seguridad son realizadas con el método Pentesting con el objetivo de determinar que requerimientos son vulnerables y cuáles no, ante los ataques realizados. La información antes mencionada se especifica en las tablas detalladas a continuación.

NOMBRE	**XSS (Cross-site scripting)**	**IDENTIFICADOR**	**PS01**
ACTIVIDADES	Ataques para vulnerar los requerimientos del sistema		
TIEMPO ESTIMADO	20 – 30 minutos por funcionalidad		
MÉTODOS O HERRAMIENTAS	Pentesting SO Kali Linux, Mozilla Firefox		
ENTREGABLES	Lista de chequeo del nivel de vulnerabilidad		

Tabla 6: Ataque XSS

NOMBRE	**SQL Injection**	**IDENTIFICADOR**	**PS02**
ACTIVIDADES	Ataques para vulnerar los requerimientos del sistema		
TIEMPO ESTIMADO	20 – 30 minutos por funcionalidad		
MÉTODOS O HERRAMIENTAS	Pentesting SO Kali Linux, Mozilla Firefox, NMap		
ENTREGABLES	Lista de chequeo del nivel de vulnerabilidad		

Tabla 7: Ataque SQL Injection

NOMBRE	**CSRF (Cross-site request forgery)**	**IDENTIFICADOR**	**PS03**
ACTIVIDADES	Ataques para vulnerar los requerimientos del sistema		

TIEMPO ESTIMADO	20 – 30 minutos por funcionalidad
MÉTODOS O HERRAMIENTAS	Pentesting SO Kali Linux, Mozilla Firefox, BurpSuite
ENTREGABLES	Lista de chequeo del nivel de vulnerabilidad

Tabla 8: Ataque CSRF.

NOMBRE	**Buffer Overflow**	**IDENTIFICADOR**	**PS04**
ACTIVIDADES	Ataques para vulnerar los requerimientos del sistema		
TIEMPO ESTIMADO	20 – 30 minutos por funcionalidad		
MÉTODOS O HERRAMIENTAS	Pentesting SO Kali Linux, Mozilla Firefox		
ENTREGABLES	Lista de chequeo del nivel de vulnerabilidad		

Tabla 9: Ataque buffer Overflow.

NOMBRE	**Ataque de directorio Transversal**	**IDENTIFICADOR**	**PS05**
ACTIVIDADES	Ataques para vulnerar los requerimientos del sistema		
TIEMPO ESTIMADO	20 – 30 minutos por funcionalidad		
MÉTODOS O HERRAMIENTAS	Pentesting SO Kali Linux, Mozilla Firefox		
ENTREGABLES	Lista de chequeo del nivel de vulnerabilidad		

Tabla 10: Ataque de directorio Transversal.

NOMBRE	**Intercepción Criptográfica**	**IDENTIFICADOR**	**PS06**
ACTIVIDADES	Ataques para vulnerar los requerimientos del sistema		
TIEMPO ESTIMADO	15 – 20 minutos por funcionalidad		
MÉTODOS O HERRAMIENTAS	Pentesting SO Kali Linux, Mozilla Firefox		
ENTREGABLES	Lista de chequeo del nivel de vulnerabilidad		

Tabla 11: Ataque de Intercepción Criptográfica.

6. PROCESO DE PRUEBAS

En esta sección se establecen los criterios de suspensión los cuales establece claramente bajo qué condiciones se detienen un conjunto de casos de pruebas, por ejemplo, en caso de existir defectos que impidan la ejecución de más casos de pruebas, cierto porcentaje de casos fallidos, o cualquier otro que se especifique. Los criterios de reanudación establecen bajo qué criterios se reanudarán las pruebas, por ejemplo: cuando nos entreguen una nueva versión de pruebas, cuando nos realicen las configuraciones necesarias, etc. Es importante considerar que para cada criterio de suspensión debe contar su criterio de reanudación. La ejecución y resultados se encuentra detallados en el trabajo de titulación.

Anexo B: Diseño de pruebas de penetración

Diseño de la prueba XSS

NOMBRE	XSS (Cross-site scripting)	**PRUEBAS**	P1
PROPÓSITO	Verificar si al aplicar el ataque XSS las funcionalidades del sistema son vulnerables.		
PRERREQUISITOS	• Haber solicitado realizar los ataques al servidor • Haber realizado un escenario de pruebas • Tener las funcionalidades previamente desarrolladas		
UBICACIÓN	Interfaz de usuario del sistema		
ENTRADA	Insertar código java script en los campos de las funcionalidades		
ORÁCULO	La interfaz reacciono de manera positiva a la prueba realizada.		
PASOS	1. Autenticarse en el sistema de estafetas 2. Seleccionar rol 3. Escoger en el menú la funcionalidad 4. Insertar código java script 5. Click en guardar		

Diseño de la prueba CSRF

NOMBRE	Cross site request forgery (CSRF)	**PRUEBAS**	P2
PROPÓSITO	Verificar si al aplicar el ataque CSRF las funcionalidades del sistema son vulnerables.		
PRERREQUISITOS	• Haber solicitado realizar los ataques al servidor • Haber realizado un escenario de pruebas • Tener las funcionalidades previamente desarrolladas		
UBICACIÓN	Burpsuite, Bloc de notas y Mozilla Firefox		
ENTRADA			
ORÁCULO	La interfaz reacciono de manera positiva a la prueba realizada.		
PASOS	1. Autenticarse en el sistema de estafetas 2. Ejecutar Burpsuite 3. Realizar gestión de datos en el sistema de estafetas 4. Capturar el tráfico con Burpsuite 5. Clic derecho sobre un elemento del trafico capturado		

	6. Seleccionar la opción generar documento CSRF 7. Copiar texto y guardarlo como html 8. Ejecutar en un navegador web

Diseño de la prueba buffer overflow

NOMBRE	Buffer overflow (DOS)	**PRUEBAS**	P3
PROPÓSITO	Verificar si al aplicar el ataque buffer overflow las funcionalidades del sistema son vulnerables.		
PRERREQUISITOS	• Haber solicitado realizar los ataques al servidor • Haber realizado un escenario de pruebas • Tener las funcionalidades previamente desarrolladas		
UBICACIÓN	Burpsuite, Bloc de notas y Mozilla Firefox		
ENTRADA	Una cadena de caracteres mayor a 500		
ORÁCULO	La interfaz reacciono de manera positiva a la prueba realizada.		
PASOS	1. Autenticarse en el sistema de estafetas 2. Seleccionar rol 3. Escoger en el menú la funcionalidad 4. Insertar cadena de caracteres 5. Click en guardar		

Diseño de la prueba SQL injection

NOMBRE	SQL Injection	**PRUEBAS**	P4
PROPÓSITO	Verificar si al aplicar el ataque SQL Injection las funcionalidades del sistema son vulnerables.		
PRERREQUISITOS	• Haber solicitado realizar los ataques al servidor • Haber realizado un escenario de pruebas • Tener la funcionalidad previamente desarrollada		
UBICACIÓN	Interfaz de NMap, java server pages (JSP)		
ENTRADA	Insertar una URL con petición para insertar un dato que contenga una sentencia SQL.		
ORÁCULO	La interfaz reacciono de manera positiva a la prueba realizada.		
PASOS	1. Autenticarse en el sistema de estafetas 2. Ejecutar los archivos JSP		

	3. Insertar una URL con sentencia SQL 4. Verificar resultados

Diseño de la intercepción criptográfica

NOMBRE	Intercepción criptográfica	**PRUEBAS**	P5
PROPÓSITO	Verificar si al aplicar el ataque CSRF las funcionalidades del sistema son vulnerables.		
PRERREQUISITOS	• Haber solicitado realizar los ataques al servidor • Haber realizado un escenario de pruebas • Tener las funcionalidades previamente desarrolladas		
UBICACIÓN	Wireshark y Mozilla Firefox		
ENTRADA			
ORÁCULO	La interfaz reacciono de manera positiva a la prueba realizada.		
PASOS	1. Ejecutar Wireshark 2. Autenticarse en el sistema de estafetas 3. Capturar el tráfico con Wireshark 4. Observar resultados 5. Interceptar claves		

Diseño de la prueba directorio transversal

NOMBRE	Directorio transversal	**PRUEBAS**	P6
PROPÓSITO	Verificar si al aplicar el ataque de directorio transversal las funcionalidades del sistema son vulnerables.		
PRERREQUISITOS	• Haber solicitado realizar los ataques al servidor • Haber realizado un escenario de pruebas • Tener la funcionalidad previamente desarrollada		
UBICACIÓN	Mozilla Firefox, java server pages (JSP)		
ENTRADA	Insertar una URL con petición para insertar un dato que contenga una paths de archivos del servidor.		
ORÁCULO	La interfaz reacciono de manera positiva a la prueba realizada.		
PASOS	1. Autenticarse en el sistema de estafetas 2. Ejecutar los archivos JSP 3. Insertar una URL con paths de archivos del servidor 4. Verificar resultados		

Anexo C: Ejecución de pruebas de penetración

Pruebas realizadas antes del mantenimiento

Pruebas Cross-site scripting (XSS)

Nombre	Insertar tipo designación (XSS)		**Identificador**		T01
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.			
Estado		**Vulnerable:**	X	**No Vulnerable:**	

Tabla 1: Entregable 01 de la Prueba XSS (Insertar tipo designación)

Nombre	Insertar designación (XSS)		**Identificador**		T02
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.			
Estado		**Vulnerable:**	X	**No Vulnerable:**	

Tabla 2: Entregable 02 de la Prueba XSS (Insertar designación)

Nombre	Modificar tipo designación (XSS)		**Identificador**		T03
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.			
Estado		**Vulnerable:**	X	**No Vulnerable:**	

Tabla 3: Entregable 03 de la Prueba XSS (Modificar tipo designación)

Nombre	Modificar designación (XSS)		**Identificador**		T04
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.			
Estado		**Vulnerable:**	X	**No Vulnerable:**	

Tabla 4: Entregable 04 de la Prueba XSS (Modificar designación)

Nombre	Insertar tipo función (XSS)		**Identificador**	T05
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.		
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.		
Estado		**Vulnerable:**	X	**No Vulnerable:**

Tabla 5: Entregable 05 de la Prueba XSS (Insertar tipo función)

Nombre	Modificar tipo función (XSS)		**Identificador**	T06
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.		
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.		
Estado		**Vulnerable:**	X	**No Vulnerable:**

Tabla 6: Entregable 06 de la Prueba XSS (Modificar tipo función)

Nombre	Ingresar secciones (XSS)		**Identificador**	T07
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.		
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.		
Estado		**Vulnerable:**	X	**No Vulnerable:**

Tabla 7: Entregable 07 de la Prueba XSS (Ingresar secciones)

Nombre	Modificar secciones (XSS)		**Identificador**	T08
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.		
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.		
Estado		**Vulnerable:**	X	**No Vulnerable:**

Tabla 8: Entregable 08 de la Prueba XSS (Modificar secciones)

Nombre	Insertar función (XSS)		**Identificador**	T09
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.		
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.		

Estado	**Vulnerable:**	X	**No Vulnerable:**	

Tabla 9: Entregable 09 de la Prueba XSS (Insertar función)

Nombre	Modificar función (XSS)		**Identificador**	T10
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 100% vulnerable ante la prueba.			
Estado	**Vulnerable:**	X	**No Vulnerable:**	

Tabla 10: Entregable 10 de la Prueba XSS (Modificar función)

Nombre	Insertar ítems (XSS)		**Identificador**	T11
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 100% vulnerable ante la prueba.			
Estado	**Vulnerable:**	X	**No Vulnerable:**	

Tabla 11: Entregable 11 de la Prueba XSS (Insertar ítems)

Nombre	Modificar ítems (XSS)		**Identificador**	T12
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 100% vulnerable ante la prueba.			
Estado	**Vulnerable:**	X	**No Vulnerable:**	

Tabla 12: Entregable 11 de la Prueba XSS (Modificar ítems)

ID	CASO DE USO	RESULTADO DE LA PRUEBA
HU-08	Insertar tipo designación	100% Vulnerable
HU-10	Insertar designación	100% Vulnerable
HU-03	Modificar tipo designación	100% Vulnerable
HU-03	Modificar designación	100% Vulnerable
HU-04	Insertar tipo función	100% Vulnerable
HU-04	Modificar tipo función	100% Vulnerable
HU-05	Insertar secciones	100% Vulnerable

HU-05	Modificar secciones	100% Vulnerable
HU-04	Insertar función	100% Vulnerable
HU-04	Modificar función	100% Vulnerable
HU-05	Insertar ítems	100% Vulnerable
HU-05	Modificar ítems	100% Vulnerable
	Promedio Total	100%

Tabla 13: Resumen de la Prueba XSS

Ataque de intercepción criptográfica

Nombre	Autenticación (Intercepción criptográfica)		**Identificador**	T13
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 14: Entregable 01 de la Prueba intercepción criptográfica (Autenticación)

ID	**CASO DE USO**	**RESULTADO DE LA PRUEBA**
HU-18	Autenticación	0% Vulnerable
	Promedio Total	0%

Tabla 15: Resumen de la Prueba de intercepción criptográfica

Pruebas de SQL Injection

Nombre	Insertar tipo designación (SQL Injection)		**Identificador**	T14
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 100% vulnerable ante la prueba.			
Estado	**Vulnerable:**	X	**No Vulnerable:**	

Tabla 16: Entregable 01 de la Prueba SQL Injection (Insertar tipo designación)

Nombre	Insertar designación (SQL Injection)		**Identificador**		T15
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.			
Estado		**Vulnerable:**	X	**No Vulnerable:**	

Tabla 16: Entregable 02 de la Prueba SQL Injection (Insertar designación)

Nombre	Modificar tipo designación (SQL Injection)		**Identificador**		T16
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.			
Estado		**Vulnerable:**	X	**No Vulnerable:**	

Tabla 17: Entregable 03 de la Prueba SQL Injection (Modificar tipo designación)

Nombre	Modificar designación (SQL Injection)		**Identificador**		T17
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.			
Estado		**Vulnerable:**	X	**No Vulnerable:**	

Tabla 18: Entregable 04 de la Prueba SQL Injection (Modificar designación)

Nombre	Insertar tipo función (SQL Injection)		**Identificador**		T18
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.			
Estado		**Vulnerable:**	X	**No Vulnerable:**	

Tabla 19: Entregable 05 de la Prueba SQL Injection (Insertar tipo función)

Nombre	Modificar tipo función (SQL Injection)		**Identificador**		T19
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.			
Estado		**Vulnerable:**	X	**No Vulnerable:**	

Tabla 20: Entregable 06 de la Prueba SQL Injection (Modificar tipo función)

Nombre	Ingresar secciones (SQL Injection)		**Identificador**		T20
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.			
Estado		**Vulnerable:**	X	**No Vulnerable:**	

Tabla 21: Entregable 07 de la Prueba SQL Injection (Ingresar secciones)

Nombre	Modificar secciones (SQL Injection)		**Identificador**		T21
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.			
Estado		**Vulnerable:**	X	**No Vulnerable:**	

Tabla 22: Entregable 08 de la Prueba SQL Injection (Modificar secciones)

Nombre	Insertar función (SQL Injection)		**Identificador**		T22
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.			
Estado		**Vulnerable:**	X	**No Vulnerable:**	

Tabla 23: Entregable 09 de la Prueba SQL Injection (Insertar función)

Nombre	Modificar función (SQL Injection1)		**Identificador**		T23
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.			
Estado		**Vulnerable:**	X	**No Vulnerable:**	

Tabla 24: Entregable 10 de la Prueba SQL Injection (Modificar función)

Nombre	Insertar ítems (SQL Injection)		**Identificador**		T24
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.			
Estado		**Vulnerable:**	X	**No Vulnerable:**	

Tabla 25: Entregable 11 de la Prueba SQL Injection (Insertar ítems)

Nombre	Modificar ítems (SQL Injection)		**Identificador**		T25
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.			
Estado		**Vulnerable:**	X	**No Vulnerable:**	

Tabla 26: Entregable 12 de la Prueba SQL Injection (Modificar ítems)

Nombre	Autenticación (SQL Injection)		**Identificador**		T26
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.			
Estado		**Vulnerable:**	X	**No Vulnerable:**	

Tabla 27: Entregable 13 de la Prueba SQL Injection (Autenticación)

Nombre	Listar eventos (SQL Injection)		**Identificador**		T27
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			

Resultados obtenidos	Funcionalidad 100% vulnerable ante la prueba.			
Estado	**Vulnerable:**	X	**No Vulnerable:**	

Tabla 28: Entregable 14 de la Prueba SQL Injection (Listar eventos)

Nombre	Insertar eventos (SQL Injection)		**Identificador**	T28
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 100% vulnerable ante la prueba.			
Estado	**Vulnerable:**	X	**No Vulnerable:**	

Tabla 29: Entregable 15 de la Prueba SQL Injection (Insertar eventos)

Nombre	Modificar eventos (SQL Injection)		**Identificador**	T29
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 100% vulnerable ante la prueba.			
Estado	**Vulnerable:**	X	**No Vulnerable:**	

Tabla 30: Entregable 16 de la Prueba SQL Injection (Modificar eventos)

Nombre	Eliminar eventos (SQL Injection)		**Identificador**	T30
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 100% vulnerable ante la prueba.			
Estado	**Vulnerable:**	X	**No Vulnerable:**	

Tabla 31: Entregable 17 de la Prueba SQL Injection **(**Eliminar eventos**)**

Nombre	Eliminar secciones (SQL Injection)		**Identificador**	T31
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 100% vulnerable ante la prueba.			
Estado	**Vulnerable:**	X	**No Vulnerable:**	

Tabla 32: Entregable 18 de la Prueba SQL Injection **(**Eliminar secciones**)**

Nombre	Listar secciones (SQL Injection)		**Identificador**		T32
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.			
Estado		**Vulnerable:**	X	**No Vulnerable:**	

Tabla 33: Entregable 19 de la Prueba SQL Injection **(**Listar secciones**)**

Nombre	Listar ítems (SQL Injection)		**Identificador**		T33
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.			
Estado		**Vulnerable:**	X	**No Vulnerable:**	

Tabla 34: Entregable 20 de la Prueba SQL Injection **(**Listar ítems**)**

Nombre	Listar designaciones (SQL Injection)		**Identificador**		T34
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.			
Estado		**Vulnerable:**	X	**No Vulnerable:**	

Tabla 35: Entregable 21 de la Prueba SQL Injection **(**Listar designaciones**)**

Nombre	Listar tipo designación (SQL Injection)		**Identificador**		T35
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.			
Estado		**Vulnerable:**	X	**No Vulnerable:**	

Tabla 36: Entregable 22 de la Prueba SQL Injection **(**Listar tipo designación**)**

Nombre	Eliminar designación (SQL Injection)	**Identificador**	T36

Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.			
Estado		**Vulnerable:**	X	**No Vulnerable:**	

Tabla 37: Entregable 23 de la Prueba SQL Injection **(**Eliminar designación**)**

ID	CASO DE USO	RESULTADO DE LA PRUEBA
HU-08	Insertar tipo designación	100% Vulnerable
HU-10	Insertar designación	100% Vulnerable
HU-03	Modificar tipo designación	100% Vulnerable
HU-03	Modificar designación	100% Vulnerable
HU-04	Insertar tipo función	100% Vulnerable
HU-04	Modificar tipo función	100% Vulnerable
HU-05	Insertar secciones	100% Vulnerable
HU-05	Modificar secciones	100% Vulnerable
HU-04	Insertar función	100% Vulnerable
HU-04	Modificar función	100% Vulnerable
HU-05	Insertar ítems	100% Vulnerable
HU-05	Modificar ítems	100% Vulnerable
HU-18	Autenticación	100% Vulnerable
HU-02	Listar eventos	100% Vulnerable
HU-02	Insertar eventos	100% Vulnerable
HU-02	Modificar eventos	100% Vulnerable
HU-02	Eliminar eventos	100% Vulnerable
HU-06	Eliminar secciones	100% Vulnerable
HU-07	Listar secciones	100% Vulnerable
HU-14	Listar ítems	100% Vulnerable
HU-10	Litar designaciones	100% Vulnerable
HU-08	Listar tipo designaciones	100% Vulnerable
HU-10	Eliminar designaciones	100% Vulnerable
	Promedio Total	100%

Tabla 38: Resumen de la Prueba SQL Injection

Pruebas de Cross-site request forgery (CSRF)

Nombre	Insertar tipo designación (CSRF)		**Identificador**	T37
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.		
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.		
Estado		**Vulnerable:**	X	**No Vulnerable:**

Tabla 39: Entregable 01 de la Prueba CSRF (Insertar tipo designación)

Nombre	Insertar designación (CSRF)		**Identificador**	T38
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.		
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.		
Estado		**Vulnerable:**	X	**No Vulnerable:**

Tabla 40: Entregable 02 de la Prueba CSRF (Insertar designación)

Nombre	Modificar tipo designación (CSRF)		**Identificador**	T39
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.		
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.		
Estado		**Vulnerable:**	X	**No Vulnerable:**

Tabla 41: Entregable 03 de la Prueba CSRF (Modificar tipo designación)

Nombre	Modificar designación (CSRF)		**Identificador**	T40
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.		
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.		
Estado		**Vulnerable:**	X	**No Vulnerable:**

Tabla 42: Entregable 04 de la Prueba CSRF (Modificar designación)

Nombre	Insertar tipo función (CSRF)	**Identificador**	T41

Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.			
Estado		**Vulnerable:**	X	**No Vulnerable:**	

Tabla 43: Entregable 05 de la Prueba CSRF (Insertar tipo función)

Nombre	Modificar tipo función (CSRF)		**Identificador**		T42
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.			
Estado		**Vulnerable:**	X	**No Vulnerable:**	

Tabla 44: Entregable 06 de la Prueba CSRF (Modificar tipo función)

Nombre	Ingresar secciones (CSRF)		**Identificador**		T43
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.			
Estado		**Vulnerable:**	X	**No Vulnerable:**	

Tabla 45: Entregable 07 de la Prueba CSRF (Ingresar secciones)

Nombre	Modificar secciones (CSRF)		**Identificador**		T44
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.			
Estado		**Vulnerable:**	X	**No Vulnerable:**	

Tabla 46: Entregable 08 de la Prueba CSRF (Modificar secciones)

Nombre	Insertar función (CSRF)		**Identificador**		T45
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.			
Estado		**Vulnerable:**	X	**No Vulnerable:**	

Tabla 47: Entregable 09 de la Prueba CSRF (Insertar función)

Nombre	Modificar función (CSRF)		**Identificador**		T46
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.			
Estado		**Vulnerable:**	X	**No Vulnerable:**	

Tabla 48: Entregable 10 de la Prueba CSRF (Modificar función)

Nombre	Insertar ítems (CSRF)		**Identificador**		T47
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.			
Estado		**Vulnerable:**	X	**No Vulnerable:**	

Tabla 49: Entregable 11 de la Prueba CSRF (Insertar ítems)

Nombre	Modificar ítems (CSRF)		**Identificador**		T48
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.			
Estado		**Vulnerable:**	X	**No Vulnerable:**	

Tabla 50: Entregable 12 de la Prueba CSRF (Modificar ítems)

Nombre	Insertar eventos (CSRF)		**Identificador**		T49
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.			
Estado		**Vulnerable:**	X	**No Vulnerable:**	

Tabla 51: Entregable 13 de la Prueba CSRF (Insertar eventos)

Nombre	Modificar eventos (CSRF)		**Identificador**	T50
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.		

Resultados obtenidos	Funcionalidad 100% vulnerable ante la prueba.			
Estado	**Vulnerable:**	X	**No Vulnerable:**	

Tabla 52: Entregable 14 de la Prueba CSRF (Modificar eventos)

Nombre	Eliminar eventos (CSRF)		**Identificador**	T51
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 100% vulnerable ante la prueba.			
Estado	**Vulnerable:**	X	**No Vulnerable:**	

Tabla 53: Entregable 15 de la Prueba CSRF **(**Eliminar eventos**)**

Nombre	Eliminar secciones (CSRF)		**Identificador**	T52
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 100% vulnerable ante la prueba.			
Estado	**Vulnerable:**	X	**No Vulnerable:**	

Tabla 54: Entregable 16 de la Prueba CSRF **(**Eliminar secciones**)**

Nombre	Eliminar designación (CSRF)		**Identificador**	T53
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 100% vulnerable ante la prueba.			
Estado	**Vulnerable:**	X	**No Vulnerable:**	

Tabla 55: Entregable 17 de la Prueba CSRF **(**Eliminar designación**)**

ID	CASO DE USO	RESULTADO DE LA PRUEBA
HU-08	Insertar tipo designación	100% Vulnerable
HU-10	Insertar designación	100% Vulnerable
HU-03	Modificar tipo designación	100% Vulnerable
HU-03	Modificar designación	100% Vulnerable
HU-04	Insertar tipo función	100% Vulnerable

HU-04	Modificar tipo función	100% Vulnerable
HU-05	Insertar secciones	100% Vulnerable
HU-05	Modificar secciones	100% Vulnerable
HU-04	Insertar función	100% Vulnerable
HU-04	Modificar función	100% Vulnerable
HU-05	Insertar ítems	100% Vulnerable
HU-05	Modificar ítems	100% Vulnerable
HU-02	Insertar eventos	100% Vulnerable
HU-02	Modificar eventos	100% Vulnerable
HU-02	Eliminar eventos	100% Vulnerable
HU-06	Eliminar secciones	100% Vulnerable
HU-10	Eliminar designaciones	100% Vulnerable
	Promedio Total	100%

Tabla 56: Resumen de la Prueba CSRF

Pruebas de Bufferoverflow (DOS)

Nombre	Insertar tipo designación (DOS)		**Identificador**	T54
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 57: Entregable 01 de la Prueba DOS (Insertar tipo designación)

Nombre	Insertar designación (DOS)		**Identificador**	T55
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 58: Entregable 02 de la Prueba DOS (Insertar designación)

Nombre	Insertar tipo función (DOS)		**Identificador**	T56
Valor máximo	100 %	**Valor mínimo**	0%	

Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 59: Entregable 03 de la Prueba DOS (Insertar tipo función)

Nombre	Insertar secciones (DOS)		**Identificador**	T57
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 60: Entregable 04 de la Prueba DOS (Insertar secciones)

Nombre	Insertar ítems (DOS)		**Identificador**	T58
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 61: Entregable 05 de la Prueba DOS (Insertar ítems)

Nombre	Insertar función (DOS)		**Identificador**	T59
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 62: Entregable 06 de la Prueba DOS (Insertar función)

Nombre	Autenticación (DOS)		**Identificador**	T60
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 63: Entregable 07 de la Prueba DOS (Insertar secciones)

ID	CASO DE USO	RESULTADO DE LA PRUEBA
HU-08	Insertar tipo designación	0% Vulnerable
HU-10	Insertar designación	0% Vulnerable
HU-04	Insertar tipo función	0% Vulnerable
HU-05	Insertar secciones	0% Vulnerable
HU-05	Insertar ítems	0% Vulnerable
HU-05	Insertar función	0% Vulnerable
HU-02	Insertar eventos	0% Vulnerable
	Promedio Total	0%

Tabla 64: Resumen de la Prueba Buffer overflow (DOS)

Pruebas de directorio transversal

Nombre	Insertar tipo designación (Directorio transversal)		**Identificador**		T61
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 0% vulnerable ante la prueba.			
Estado		**Vulnerable:**		**No Vulnerable:**	X

Tabla 65: Entregable 01 de la Prueba Directorio transversal (Insertar tipo designación)

Nombre	Insertar designación (Directorio transversal)		**Identificador**		T62
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 0% vulnerable ante la prueba.			
Estado		**Vulnerable:**		**No Vulnerable:**	X

Tabla 66: Entregable 02 de la Prueba Directorio transversal (Insertar designación)

Nombre	Modificar tipo designación (Directorio transversal)		**Identificador**	T63
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.		

Resultados obtenidos	Funcionalidad 100% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 67: Entregable 03 de la Prueba Directorio transversal (Modificar tipo designación)

Nombre	Modificar designación (Directorio transversal)		**Identificador**	T64
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 68: Entregable 04 de la Prueba Directorio transversal (Modificar designación)

Nombre	Insertar tipo función (Directorio transversal)		**Identificador**	T65
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 69: Entregable 05 de la Prueba Directorio transversal (Insertar tipo función)

Nombre	Modificar tipo función (Directorio transversal)		**Identificador**	T66
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 70: Entregable 06 de la Prueba Directorio transversal (Modificar tipo función)

Nombre	Ingresar secciones (Directorio transversal)		**Identificador**	T67
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			

Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 71: Entregable 07 de la Prueba Directorio transversal (Ingresar secciones)

Nombre	Modificar secciones (Directorio transversal)		**Identificador**	T68
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 72: Entregable 08 de la Prueba Directorio transversal (Modificar secciones)

Nombre	Insertar función (Directorio transversal)		**Identificador**	T69
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 73: Entregable 09 de la Prueba Directorio transversal (Insertar función)

Nombre	Modificar función (Directorio transversal)		**Identificador**	T70
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 74: Entregable 10 de la Prueba Directorio transversal (Modificar función)

Nombre	Insertar ítems (Directorio transversal)		**Identificador**	T71
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			

Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 75: Entregable 11 de la Prueba Directorio transversal (Insertar ítems)

Nombre	Modificar ítems (Directorio transversal)		**Identificador**	T72
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 76: Entregable 12 de la Prueba Directorio transversal (Modificar ítems)

Nombre	Autenticación (Directorio transversal)		**Identificador**	T73
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 77: Entregable 13 de la Prueba Directorio transversal (Insertar eventos)

Nombre	Visualizar horario del docente (Directorio transversal)		**Identificador**	T74
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 78: Entregable 14 de la Prueba Directorio transversal (Modificar eventos)

Nombre	Listar secciones (Directorio transversal)		**Identificador**	T75
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			

Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 79: Entregable 15 de la Prueba Directorio transversal **(**Eliminar eventos**)**

Nombre	Eliminar secciones (Directorio transversal)		**Identificador**	T76
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 80: Entregable 16 de la Prueba Directorio transversal **(**Eliminar secciones**)**

Nombre	Eliminar designación (Directorio transversal)		**Identificador**	T77
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 81: Entregable 17 de la Prueba Directorio transversal **(**Eliminar designación**)**

ID	CASO DE USO	RESULTADO DE LA PRUEBA
HU-08	Insertar tipo designación	0% Vulnerable
HU-10	Insertar designación	0% Vulnerable
HU-03	Modificar tipo designación	0% Vulnerable
HU-03	Modificar designación	0% Vulnerable
HU-04	Insertar tipo función	0% Vulnerable
HU-04	Modificar tipo función	0% Vulnerable
HU-05	Insertar secciones	0% Vulnerable
HU-05	Modificar secciones	0% Vulnerable
HU-04	Insertar función	0% Vulnerable
HU-04	Modificar función	0% Vulnerable
HU-05	Insertar ítems	0% Vulnerable
HU-05	Modificar ítems	0% Vulnerable

HU-02	Autenticación	0% Vulnerable
	Visualizar horario del docente	0% Vulnerable
HU-02	Eliminar secciones	0% Vulnerable
HU-06	Listar secciones	0% Vulnerable
HU-10	Eliminar designaciones	0% Vulnerable
	Promedio Total	0%

Tabla 82: Resumen de la Prueba Directorio transversal

Pruebas realizadas después del mantenimiento

Pruebas Cross-site scripting (XSS)

Nombre	Insertar tipo designación (XSS)		**Identificador**		T01
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 0% vulnerable ante la prueba.			
Estado		**Vulnerable:**		**No Vulnerable:**	X

Tabla 1: Entregable 01 de la Prueba XSS (Insertar tipo designación)

Nombre	Insertar designación (XSS)		**Identificador**		T02
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 0% vulnerable ante la prueba.			
Estado		**Vulnerable:**		**No Vulnerable:**	X

Tabla 2: Entregable 02 de la Prueba XSS (Insertar designación)

Nombre	Modificar tipo designación (XSS)		**Identificador**		T03
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 0% vulnerable ante la prueba.			
Estado		**Vulnerable:**		**No Vulnerable:**	X

Tabla 3: Entregable 03 de la Prueba XSS (Modificar tipo designación)

Nombre	Modificar designación (XSS)		**Identificador**		T04
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 0% vulnerable ante la prueba.			
Estado		**Vulnerable:**		**No Vulnerable:**	X

Tabla 4: Entregable 04 de la Prueba XSS (Modificar designación)

Nombre	Insertar tipo función (XSS)		**Identificador**		T05
Valor máximo	100 %	**Valor mínimo**		0%	

Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 5: Entregable 05 de la Prueba XSS (Insertar tipo función)

Nombre	Modificar tipo función (XSS)		**Identificador**	T06
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 6: Entregable 06 de la Prueba XSS (Modificar tipo función)

Nombre	Ingresar secciones (XSS)		**Identificador**	T07
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 7: Entregable 07 de la Prueba XSS (Ingresar secciones)

Nombre	Modificar secciones (XSS)		**Identificador**	T08
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 8: Entregable 08 de la Prueba XSS (Modificar secciones)

Nombre	Insertar función (XSS)		**Identificador**	T09
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 9: Entregable 09 de la Prueba XSS (Insertar función)

Nombre	Modificar función (XSS)		**Identificador**	T10
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.		
Resultados obtenidos		Funcionalidad 0% vulnerable ante la prueba.		
Estado		**Vulnerable:**	**No Vulnerable:**	X

Tabla 10: Entregable 10 de la Prueba XSS (Modificar función)

Nombre	Insertar ítems (XSS)		**Identificador**	T11
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.		
Resultados obtenidos		Funcionalidad 0% vulnerable ante la prueba.		
Estado		**Vulnerable:**	**No Vulnerable:**	X

Tabla 11: Entregable 11 de la Prueba XSS (Insertar ítems)

Nombre	Modificar ítems (XSS)		**Identificador**	T12
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.		
Resultados obtenidos		Funcionalidad 0% vulnerable ante la prueba.		
Estado		**Vulnerable:**	**No Vulnerable:**	X

Tabla 12: Entregable 11 de la Prueba XSS (Modificar ítems)

ID	**CASO DE USO**	**RESULTADO DE LA PRUEBA**
HU-08	Insertar tipo designación	0% Vulnerable
HU-10	Insertar designación	0% Vulnerable
HU-03	Modificar tipo designación	0% Vulnerable
HU-03	Modificar designación	0% Vulnerable
HU-04	Insertar tipo función	0% Vulnerable
HU-04	Modificar tipo función	0% Vulnerable
HU-05	Insertar secciones	0% Vulnerable
HU-05	Modificar secciones	0% Vulnerable
HU-04	Insertar función	0% Vulnerable
HU-04	Modificar función	0% Vulnerable
HU-05	Insertar ítems	0% Vulnerable

HU-05	Modificar ítems	0% Vulnerable
	Promedio Total	0%

Tabla 13: Resumen de la Prueba XSS

Ataque de intercepción criptográfica

Nombre	Autenticación (Intercepción criptográfica)		**Identificador**	T13
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 14: Entregable 01 de la Prueba intercepción criptográfica (Autenticación)

ID	**CASO DE USO**	**RESULTADO DE LA PRUEBA**
HU-18	Autenticación	0% Vulnerable
	Promedio Total	0%

Tabla 15: Resumen de la Prueba de intercepción criptográfica

Pruebas de SQL Injection

Nombre	Insertar tipo designación (SQL Injection)		**Identificador**	T14
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 16: Entregable 01 de la Prueba SQL Injection (Insertar tipo designación)

Nombre	Insertar designación (SQL Injection)		**Identificador**	T15
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			

Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 16: Entregable 02 de la Prueba SQL Injection (Insertar designación)

Nombre	Modificar tipo designación (SQL Injection)		**Identificador**	T16
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 17: Entregable 03 de la Prueba SQL Injection (Modificar tipo designación)

Nombre	Modificar designación (SQL Injection)		**Identificador**	T17
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 18: Entregable 04 de la Prueba SQL Injection (Modificar designación)

Nombre	Insertar tipo función (SQL Injection)		**Identificador**	T18
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 19: Entregable 05 de la Prueba SQL Injection (Insertar tipo función)

Nombre	Modificar tipo función (SQL Injection)		**Identificador**	T19
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			

Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 20: Entregable 06 de la Prueba SQL Injection (Modificar tipo función)

Nombre	Ingresar secciones (SQL Injection)		**Identificador**	T20
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 21: Entregable 07 de la Prueba SQL Injection (Ingresar secciones)

Nombre	Modificar secciones (SQL Injection)		**Identificador**	T21
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 22: Entregable 08 de la Prueba SQL Injection (Modificar secciones)

Nombre	Insertar función (SQL Injection)		**Identificador**	T22
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 23: Entregable 09 de la Prueba SQL Injection (Insertar función)

Nombre	Modificar función (SQL Injection1)		**Identificador**	T23
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 24: Entregable 10 de la Prueba SQL Injection (Modificar función)

Nombre	Insertar ítems (SQL Injection)		**Identificador**	T24
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.		
Resultados obtenidos		Funcionalidad 0% vulnerable ante la prueba.		
Estado		**Vulnerable:**	**No Vulnerable:**	X

Tabla 25: Entregable 11 de la Prueba SQL Injection (Insertar ítems)

Nombre	Modificar ítems (SQL Injection)		**Identificador**	T25
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.		
Resultados obtenidos		Funcionalidad 0% vulnerable ante la prueba.		
Estado		**Vulnerable:**	**No Vulnerable:**	X

Tabla 26: Entregable 12 de la Prueba SQL Injection (Modificar ítems)

Nombre	Autenticación (SQL Injection)		**Identificador**	T26
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.		
Resultados obtenidos		Funcionalidad 0% vulnerable ante la prueba.		
Estado		**Vulnerable:**	**No Vulnerable:**	X

Tabla 27: Entregable 13 de la Prueba SQL Injection (Autenticación)

Nombre	Listar eventos (SQL Injection)		**Identificador**	T27
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.		
Resultados obtenidos		Funcionalidad 0% vulnerable ante la prueba.		
Estado		**Vulnerable:**	**No Vulnerable:**	X

Tabla 28: Entregable 14 de la Prueba SQL Injection (Listar eventos)

Nombre	Insertar eventos (SQL Injection)		**Identificador**	T28
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.		

Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 29: Entregable 15 de la Prueba SQL Injection (Insertar eventos)

Nombre	Modificar eventos (SQL Injection)		**Identificador**	T29
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 30: Entregable 16 de la Prueba SQL Injection (Modificar eventos)

Nombre	Eliminar eventos (SQL Injection)		**Identificador**	T30
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 31: Entregable 17 de la Prueba SQL Injection **(**Eliminar eventos**)**

Nombre	Eliminar secciones (SQL Injection)		**Identificador**	T31
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 32: Entregable 18 de la Prueba SQL Injection **(**Eliminar secciones**)**

Nombre	Listar secciones (SQL Injection)		**Identificador**	T32
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 33: Entregable 19 de la Prueba SQL Injection **(**Listar secciones**)**

Nombre	Listar ítems (SQL Injection)		**Identificador**		T33
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 0% vulnerable ante la prueba.			
Estado		**Vulnerable:**		**No Vulnerable:**	X

Tabla 34: Entregable 20 de la Prueba SQL Injection **(**Listar ítems**)**

Nombre	Listar designaciones (SQL Injection)		**Identificador**		T34
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 0% vulnerable ante la prueba.			
Estado		**Vulnerable:**		**No Vulnerable:**	X

Tabla 35: Entregable 21 de la Prueba SQL Injection **(**Listar designaciones**)**

Nombre	Listar tipo designación (SQL Injection)		**Identificador**		T35
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 0% vulnerable ante la prueba.			
Estado		**Vulnerable:**		**No Vulnerable:**	X

Tabla 36: Entregable 22 de la Prueba SQL Injection **(**Listar tipo designación**)**

Nombre	Eliminar designación (SQL Injection)		**Identificador**		T36
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 0% vulnerable ante la prueba.			
Estado		**Vulnerable:**		**No Vulnerable:**	X

Tabla 37: Entregable 23 de la Prueba SQL Injection **(**Eliminar designación**)**

ID	CASO DE USO	RESULTADO DE LA PRUEBA
HU-08	Insertar tipo designación	0% Vulnerable
HU-10	Insertar designación	0% Vulnerable
HU-03	Modificar tipo designación	0% Vulnerable
HU-03	Modificar designación	0% Vulnerable
HU-04	Insertar tipo función	0% Vulnerable
HU-04	Modificar tipo función	0% Vulnerable
HU-05	Insertar secciones	0% Vulnerable
HU-05	Modificar secciones	0% Vulnerable
HU-04	Insertar función	0% Vulnerable
HU-04	Modificar función	0% Vulnerable
HU-05	Insertar ítems	0% Vulnerable
HU-05	Modificar ítems	0% Vulnerable
HU-18	Autenticación	0% Vulnerable
HU-02	Listar eventos	0% Vulnerable
HU-02	Insertar eventos	0% Vulnerable
HU-02	Modificar eventos	0% Vulnerable
HU-02	Eliminar eventos	0% Vulnerable
HU-06	Eliminar secciones	0% Vulnerable
HU-07	Listar secciones	0% Vulnerable
HU-14	Listar ítems	0% Vulnerable
HU-10	Litar designaciones	0% Vulnerable
HU-08	Listar tipo designaciones	0% Vulnerable
HU-10	Eliminar designaciones	0% Vulnerable
	Promedio Total	0%

Tabla 38: Resumen de la Prueba SQL Injection

Pruebas de Cross-site request forgery (CSRF)

Nombre	Insertar tipo designación (CSRF)	**Identificador**	T37
Valor máximo	100 %	**Valor mínimo**	0%
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.		
Resultados obtenidos	Funcionalidad 100% vulnerable ante la prueba.		

Estado	**Vulnerable:**	X	**No Vulnerable:**	

Tabla 39: Entregable 01 de la Prueba CSRF (Insertar tipo designación)

Nombre	Insertar designación (CSRF)		**Identificador**	T38
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 100% vulnerable ante la prueba.			
Estado	**Vulnerable:**	X	**No Vulnerable:**	

Tabla 40: Entregable 02 de la Prueba CSRF (Insertar designación)

Nombre	Modificar tipo designación (CSRF)		**Identificador**	T39
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 100% vulnerable ante la prueba.			
Estado	**Vulnerable:**	X	**No Vulnerable:**	

Tabla 41: Entregable 03 de la Prueba CSRF (Modificar tipo designación)

Nombre	Modificar designación (CSRF)		**Identificador**	T40
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 100% vulnerable ante la prueba.			
Estado	**Vulnerable:**	X	**No Vulnerable:**	

Tabla 42: Entregable 04 de la Prueba CSRF (Modificar designación)

Nombre	Insertar tipo función (CSRF)		**Identificador**	T41
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 100% vulnerable ante la prueba.			
Estado	**Vulnerable:**	X	**No Vulnerable:**	

Tabla 43: Entregable 05 de la Prueba CSRF (Insertar tipo función)

Nombre	Modificar tipo función (CSRF)		**Identificador**	T42
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.		
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.		
Estado		**Vulnerable:**	X	**No Vulnerable:**

Tabla 44: Entregable 06 de la Prueba CSRF (Modificar tipo función)

Nombre	Ingresar secciones (CSRF)		**Identificador**	T43
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.		
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.		
Estado		**Vulnerable:**	X	**No Vulnerable:**

Tabla 45: Entregable 07 de la Prueba CSRF (Ingresar secciones)

Nombre	Modificar secciones (CSRF)		**Identificador**	T44
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.		
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.		
Estado		**Vulnerable:**	X	**No Vulnerable:**

Tabla 46: Entregable 08 de la Prueba CSRF (Modificar secciones)

Nombre	Insertar función (CSRF)		**Identificador**	T45
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.		
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.		
Estado		**Vulnerable:**	X	**No Vulnerable:**

Tabla 47: Entregable 09 de la Prueba CSRF (Insertar función)

Nombre	Modificar función (CSRF)		**Identificador**	T46
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.		
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.		

Estado	**Vulnerable:**	X	**No Vulnerable:**	

Tabla 48: Entregable 10 de la Prueba CSRF (Modificar función)

Nombre	Insertar ítems (CSRF)		**Identificador**	T47
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 100% vulnerable ante la prueba.			
Estado	**Vulnerable:**	X	**No Vulnerable:**	

Tabla 49: Entregable 11 de la Prueba CSRF (Insertar ítems)

Nombre	Modificar ítems (CSRF)		**Identificador**	T48
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 100% vulnerable ante la prueba.			
Estado	**Vulnerable:**	X	**No Vulnerable:**	

Tabla 50: Entregable 12 de la Prueba CSRF (Modificar ítems)

Nombre	Insertar eventos (CSRF)		**Identificador**	T49
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 100% vulnerable ante la prueba.			
Estado	**Vulnerable:**	X	**No Vulnerable:**	

Tabla 51: Entregable 13 de la Prueba CSRF (Insertar eventos)

Nombre	Modificar eventos (CSRF)		**Identificador**	T50
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 100% vulnerable ante la prueba.			
Estado	**Vulnerable:**	X	**No Vulnerable:**	

Tabla 52: Entregable 14 de la Prueba CSRF (Modificar eventos)

Nombre	Eliminar eventos (CSRF)	**Identificador**	T51

Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.			
Estado		**Vulnerable:**	X	**No Vulnerable:**	

Tabla 53: Entregable 15 de la Prueba CSRF **(**Eliminar eventos**)**

Nombre	Eliminar secciones (CSRF)		**Identificador**		T52
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.			
Estado		**Vulnerable:**	X	**No Vulnerable:**	

Tabla 54: Entregable 16 de la Prueba CSRF **(**Eliminar secciones**)**

Nombre	Eliminar designación (CSRF)		**Identificador**		T53
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 100% vulnerable ante la prueba.			
Estado		**Vulnerable:**	X	**No Vulnerable:**	

Tabla 55: Entregable 17 de la Prueba CSRF **(**Eliminar designación**)**

ID	CASO DE USO	RESULTADO DE LA PRUEBA
HU-08	Insertar tipo designación	100% Vulnerable
HU-10	Insertar designación	100% Vulnerable
HU-03	Modificar tipo designación	100% Vulnerable
HU-03	Modificar designación	100% Vulnerable
HU-04	Insertar tipo función	100% Vulnerable
HU-04	Modificar tipo función	100% Vulnerable
HU-05	Insertar secciones	100% Vulnerable
HU-05	Modificar secciones	100% Vulnerable
HU-04	Insertar función	100% Vulnerable
HU-04	Modificar función	100% Vulnerable
HU-05	Insertar ítems	100% Vulnerable

HU-05	Modificar ítems	100% Vulnerable
HU-02	Insertar eventos	100% Vulnerable
HU-02	Modificar eventos	100% Vulnerable
HU-02	Eliminar eventos	100% Vulnerable
HU-06	Eliminar secciones	100% Vulnerable
HU-10	Eliminar designaciones	100% Vulnerable
	Promedio Total	100%

Tabla 56: Resumen de la Prueba CSRF

Pruebas de Bufferoverflow (DOS)

Nombre	Insertar tipo designación (DOS)		**Identificador**	T54
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 57: Entregable 01 de la Prueba DOS (Insertar tipo designación)

Nombre	Insertar designación (DOS)		**Identificador**	T55
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 58: Entregable 02 de la Prueba DOS (Insertar designación)

Nombre	Insertar tipo función (DOS)		**Identificador**	T56
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 59: Entregable 03 de la Prueba DOS (Insertar tipo función)

Nombre	Insertar secciones (DOS)	**Identificador**	T57

Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 0% vulnerable ante la prueba.			
Estado		**Vulnerable:**		**No Vulnerable:**	X

Tabla 60: Entregable 04 de la Prueba DOS (Insertar secciones)

Nombre	Insertar ítems (DOS)		**Identificador**		T58
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 0% vulnerable ante la prueba.			
Estado		**Vulnerable:**		**No Vulnerable:**	X

Tabla 61: Entregable 05 de la Prueba DOS (Insertar ítems)

Nombre	Insertar función (DOS)		**Identificador**		T59
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 0% vulnerable ante la prueba.			
Estado		**Vulnerable:**		**No Vulnerable:**	X

Tabla 62: Entregable 06 de la Prueba DOS (Insertar función)

Nombre	Autenticación (DOS)		**Identificador**		T60
Valor máximo	100 %	**Valor mínimo**		0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos		Funcionalidad 0% vulnerable ante la prueba.			
Estado		**Vulnerable:**		**No Vulnerable:**	X

Tabla 63: Entregable 07 de la Prueba DOS (Insertar secciones)

ID	CASO DE USO	RESULTADO DE LA PRUEBA
HU-08	Insertar tipo designación	0% Vulnerable
HU-10	Insertar designación	0% Vulnerable
HU-04	Insertar tipo función	0% Vulnerable
HU-05	Insertar secciones	0% Vulnerable

HU-05	Insertar ítems	0% Vulnerable
HU-05	Insertar función	0% Vulnerable
HU-02	Insertar eventos	0% Vulnerable
	Promedio Total	0%

Tabla 64: Resumen de la Prueba Buffer overflow (DOS)

Pruebas de directorio transversal

Nombre	Insertar tipo designación (Directorio transversal)		**Identificador**	T61
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 65: Entregable 01 de la Prueba Directorio transversal (Insertar tipo designación)

Nombre	Insertar designación (Directorio transversal)		**Identificador**	T62
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 66: Entregable 02 de la Prueba Directorio transversal (Insertar designación)

Nombre	Modificar tipo designación (Directorio transversal)		**Identificador**	T63
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 100% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 67: Entregable 03 de la Prueba Directorio transversal (Modificar tipo designación)

Nombre	Modificar designación (Directorio transversal)		**Identificador**	T64
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.		
Resultados obtenidos		Funcionalidad 0% vulnerable ante la prueba.		
Estado		**Vulnerable:**	**No Vulnerable:**	X

Tabla 68: Entregable 04 de la Prueba Directorio transversal (Modificar designación)

Nombre	Insertar tipo función (Directorio transversal)		**Identificador**	T65
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.		
Resultados obtenidos		Funcionalidad 0% vulnerable ante la prueba.		
Estado		**Vulnerable:**	**No Vulnerable:**	X

Tabla 69: Entregable 05 de la Prueba Directorio transversal (Insertar tipo función)

Nombre	Modificar tipo función (Directorio transversal)		**Identificador**	T66
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.		
Resultados obtenidos		Funcionalidad 0% vulnerable ante la prueba.		
Estado		**Vulnerable:**	**No Vulnerable:**	X

Tabla 70: Entregable 06 de la Prueba Directorio transversal (Modificar tipo función)

Nombre	Ingresar secciones (Directorio transversal)		**Identificador**	T67
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.		
Resultados obtenidos		Funcionalidad 0% vulnerable ante la prueba.		
Estado		**Vulnerable:**	**No Vulnerable:**	X

Tabla 71: Entregable 07 de la Prueba Directorio transversal (Ingresar secciones)

Nombre	Modificar secciones (Directorio transversal)		**Identificador**	T68
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.		
Resultados obtenidos		Funcionalidad 0% vulnerable ante la prueba.		
Estado		**Vulnerable:**	**No Vulnerable:**	X

Tabla 72: Entregable 08 de la Prueba Directorio transversal (Modificar secciones)

Nombre	Insertar función (Directorio transversal)		**Identificador**	T69
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.		
Resultados obtenidos		Funcionalidad 0% vulnerable ante la prueba.		
Estado		**Vulnerable:**	**No Vulnerable:**	X

Tabla 73: Entregable 09 de la Prueba Directorio transversal (Insertar función)

Nombre	Modificar función (Directorio transversal)		**Identificador**	T70
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.		
Resultados obtenidos		Funcionalidad 0% vulnerable ante la prueba.		
Estado		**Vulnerable:**	**No Vulnerable:**	X

Tabla 74: Entregable 10 de la Prueba Directorio transversal (Modificar función)

Nombre	Insertar ítems (Directorio transversal)		**Identificador**	T71
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.		
Resultados obtenidos		Funcionalidad 0% vulnerable ante la prueba.		
Estado		**Vulnerable:**	**No Vulnerable:**	X

Tabla 75: Entregable 11 de la Prueba Directorio transversal (Insertar ítems)

Nombre	Modificar ítems (Directorio transversal)		**Identificador**	T72
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.		
Resultados obtenidos		Funcionalidad 0% vulnerable ante la prueba.		
Estado		**Vulnerable:**	**No Vulnerable:**	X

Tabla 76: Entregable 12 de la Prueba Directorio transversal (Modificar ítems)

Nombre	Autenticación (Directorio transversal)		**Identificador**	T73
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.		
Resultados obtenidos		Funcionalidad 0% vulnerable ante la prueba.		
Estado		**Vulnerable:**	**No Vulnerable:**	X

Tabla 77: Entregable 13 de la Prueba Directorio transversal (Insertar eventos)

Nombre	Visualizar horario del docente (Directorio transversal)		**Identificador**	T74
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.		
Resultados obtenidos		Funcionalidad 0% vulnerable ante la prueba.		
Estado		**Vulnerable:**	**No Vulnerable:**	X

Tabla 78: Entregable 14 de la Prueba Directorio transversal (Modificar eventos)

Nombre	Listar secciones (Directorio transversal)		**Identificador**	T75
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado		Funcionalidad 0% vulnerable ante la prueba.		
Resultados obtenidos		Funcionalidad 0% vulnerable ante la prueba.		
Estado		**Vulnerable:**	**No Vulnerable:**	X

Tabla 79: Entregable 15 de la Prueba Directorio transversal **(**Eliminar eventos**)**

Nombre	Eliminar secciones (Directorio transversal)		**Identificador**	T76
Valor máximo	100 %	**Valor mínimo**	0%	

Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 80: Entregable 16 de la Prueba Directorio transversal **(**Eliminar secciones**)**

Nombre	Eliminar designación (Directorio transversal)		**Identificador**	T77
Valor máximo	100 %	**Valor mínimo**	0%	
Resultado esperado	Funcionalidad 0% vulnerable ante la prueba.			
Resultados obtenidos	Funcionalidad 0% vulnerable ante la prueba.			
Estado	**Vulnerable:**		**No Vulnerable:**	X

Tabla 81: Entregable 17 de la Prueba Directorio transversal **(**Eliminar designación**)**

ID	CASO DE USO	RESULTADO DE LA PRUEBA
HU-08	Insertar tipo designación	0% Vulnerable
HU-10	Insertar designación	0% Vulnerable
HU-03	Modificar tipo designación	0% Vulnerable
HU-03	Modificar designación	0% Vulnerable
HU-04	Insertar tipo función	0% Vulnerable
HU-04	Modificar tipo función	0% Vulnerable
HU-05	Insertar secciones	0% Vulnerable
HU-05	Modificar secciones	0% Vulnerable
HU-04	Insertar función	0% Vulnerable
HU-04	Modificar función	0% Vulnerable
HU-05	Insertar ítems	0% Vulnerable
HU-05	Modificar ítems	0% Vulnerable
HU-02	Autenticación	0% Vulnerable
	Visualizar horario del docente	0% Vulnerable
HU-02	Eliminar secciones	0% Vulnerable
HU-06	Listar secciones	0% Vulnerable
HU-10	Eliminar designaciones	0% Vulnerable
	Promedio Total	0%

Tabla 82: Resumen de la Prueba Directorio transversal

Anexo D: Factibilidad técnica

Hardware existente para el desarrollo

HARDWARE	CARACTERÍSTICAS
LAPTOP ASUS	Intel Core i7, Memoria RAM de 8 GB, Disco duro de 1 TB, Sistema operativo Windows 10 de 64 bits.
LAPTOP DELL	Intel Core i7, Memoria RAM de 16 GB, Disco duro de 1 TB, Sistema operativo Windows 10 de 64 bits.
Impresora EPSON	L200 Multifunción, Color y Blanco Negro.

Hardware existente para la implementación

CANT.	DESCRIPCIÓN	ESTADO
1	Servidor Core i5, memoria RAM de 8 GB procesador i5 3.5 GHZ, disco duro 3 TB, memoria 8GB	Bueno
1	Servidor Core i5, memoria RAM de 8 GB procesador i5 3.5 GHZ, disco duro 3 TB, memoria 8GB	Bueno

Software existente para la implantación

NOMBRE	DESCRIPCIÓN	LICENCIA
CentOS 7	Sistema operativo de 64 bits	Libre
Payara Server	Servidor Web	Libre
PostgreSQL	Gestor de base de datos	Libre

Anexo E: Factibilidad operativa

CANT.	CARGO	FUNCIÓN	ACTIVIDAD
1	Administrador	Gestionar todos los módulos	Listado, ingreso, modificación y eliminación de reportes
7	Vicedecano	Crear tipo designación y tipo función Designar docentes Evaluar de la jornada laboral semanal del docente	
	Docente	Consulta de resumen de horas por ítem Gestión del horario Visualización de datos personales	

Anexo F: Factibilidad económica

Costo de personal

PERSONAL			
PERSONAL	**DESCRIPCIÓN**	**SUELDO**	**V. TOTAL**
Sandra Ordoñez	Autor de tesis	$ 400,00	$ 2.400,00
Kevin Chimbo	Autor de tesis	$ 400,00	$ 2.400,00
		TOTAL	**$ 4.800,00**

Costo de instalación

CANT.	DESCRIPCIÓN	V. UNITARIO	V. TOTAL
1	Costo de personal durante la instalación	$ 100,00	$ 100,00
		TOTAL	**$ 100,00**

Costo de operación

CANT.	DESCRIPCIÓN	V. UNITARIO	V. TOTAL
1	Costo de capacitación al personal que operará el sistema	$ 500,00	$ 500,00
		TOTAL	**$ 500,00**

Costo de materiales y suministros

MATERIALES Y SUMINISTROS			
CANT.	**DESCRIPCIÓN**	**V. UNITARIO**	**V. TOTAL**
3,00	Resma de papel	$ 3,00	$ 9,00
2,00	Tinta negra	$ 9,00	$ 18,00
6,00	Tinta color	$ 11,00	$ 66,00
4,00	CD	$ 0,50	$ 2,00
2,00	Pendrive	$ 8,00	$ 16,00
2,00	Carpeta	$ 1,00	$ 2,00
		TOTAL	**$ 113,00**

Otros costos

OTROS	
DESCRIPCIÓN	**V. TOTAL**
Transporte	$ 240,00

Alimentación	$ 150,00
TOTAL	**$ 390,00**

Anexo G: Identificación de riesgos.

ID	Descripción del riesgo	Tipo	Consecuencias
R01	Cambio de requerimientos del sistema.	Proyecto	Retraso del proyecto
R02	Descoordinación en el equipo de trabajo.	Proyecto	Retraso del proyecto
R03	Funcionalidades desarrolladas incorrectamente	Proyecto	Retraso del proyecto
R04	Servicios web externos no funcionales	Técnico	Dificultad de implementación
R05	Acoplarse a las herramientas y técnicas del sistema desarrollado.	Técnico	Dificultad en el desarrollo del sistema

Anexo H: Análisis de riesgos

- Determinación de la Probabilidad

Rango de Probabilidades	Descripción	Valor
1% - 33%	Baja	1
34% - 67%	Media	2
68% - 99%	Alta	3

- Determinación del Impacto

Impacto	Retraso	Impacto Técnico	Valor
Bajo	1 semana	Ligero efecto	1
Moderado	2 semanas	Moderado efecto	2
Alto	1 mes	Severo efecto	3
Crítico	Más de un mes	Proyecto no puede ser culminado	4

- Determinación de la Exposición al Riesgo

Exposición al Riesgo	Valor	Color
Baja	1 o 2	1
Media	3 o 4	2
Alta	Mayor a 6	3

Impacto Prob.	Bajo=1	Moderado=2	Alto=3	Critico=4
Alta=3	3	6	9	12
Media=2	2	4	6	8
Baja=1	1	2	3	4

- Determinación de la Prioridad del Riesgo

ID	Probabilidad			Impacto		Exposición	
	Porcentaje	Probabilidad	Valor	Impacto	Valor	Exposición	Valor
R01	70%	Alta	3	Alto	3	Alta	9
R02	33%	Baja	1	Bajo	1	Baja	1
R03	70%	Alta	3	Alto	3	Alta	9
R04	30%	Baja	1	Moderado	2	Baja	2
R05	35%	Media	2	Moderado	2	Media	4

ID	PRIORIDAD	VALOR EXPOSICIÓN
R03	1	9
R01	1	9
R05	2	4
R04	3	2
R02	4	1

Anexo I: Gestión de Riesgos

<table>
<tr><th colspan="4">HOJA DE GESTIÓN DE RIESGO</th></tr>
<tr><td colspan="2">ID. DEL RIESGO: R01</td><td colspan="2">FECHA: 10/11/2018</td></tr>
<tr><td>Probabilidad: Alta
Valor: 3</td><td>Impacto: Alto
Valor: 3</td><td>Exposición: Alto
Valor: 9</td><td>Prioridad: 1</td></tr>
<tr><td colspan="4">DESCRIPCIÓN: Cambio de requerimientos del sistema.</td></tr>
<tr><td colspan="4">REFINAMIENTO:
Causas:
• Falta de comunicación.
• Falta de especificación de los requisitos del sistema.
Consecuencia:
• Demora en la implantación del sistema
• Retraso en los tiempos de entrega del proyecto.
• Desperdicio de recursos y tiempo.</td></tr>
<tr><td colspan="4">REDUCCIÓN:
• Falta de comunicación.
• Malos entendidos.</td></tr>
<tr><td colspan="4">SUPERVISIÓN:
• Verificar las fuentes de información
• Presentar un entregable/documento luego del análisis de requisitos.</td></tr>
<tr><td colspan="4">GESTIÓN:
• Definir las cláusulas del proyecto.
• Información correcta.
• Reorganizar los cambios presentados en el proyecto.</td></tr>
<tr><td colspan="4">ESTADO ACTUAL:
Fase de Reducción Iniciada ■
Fase de Supervisión Iniciada ■
Gestionando el Riesgo ☐</td></tr>
<tr><td colspan="4">RESPONSABLE:
- Sandra Ordoñez
- Kevin Chimbo</td></tr>
</table>

HOJA DE GESTIÓN DE RIESGO			
ID. DEL RIESGO: R02		**FECHA:** 10/11/2018	
Probabilidad: Baja **Valor:** 1	**Impacto:** Bajo **Valor:** 1	**Exposición:** Bajo **Valor:** 1	**Prioridad:** 4
DESCRIPCIÓN: Descoordinación en el equipo de trabajo.			
REFINAMIENTO: **Causas:** • El ambiente del personal no es sociable entre ellos. • Falta de comunicación. • Desacuerdo en la toma de decisiones dentro del grupo. **Consecuencia:** • Retraso en los tiempos de entrega del proyecto. • Suspensión de personal. • Proyecto incompleto.			
REDUCCIÓN: • Entenderse de mejor manera con las ideas del grupo. • Apoyo y resolución de problemas entre el grupo.			
SUPERVISIÓN: • Comprobar el desempeño del equipo de trabajo. • Relaciones interpersonales con el equipo de trabajo.			
GESTIÓN: • Reorganizar el equipo de trabajo. • Charlas con el equipo de trabajo.			
ESTADO ACTUAL: Fase de Reducción Iniciada ■ Fase de Supervisión Iniciada ■ Gestionando el Riesgo ☐			
RESPONSABLE: - Sandra Ordoñez - Kevin Chimbo			

HOJA DE GESTIÓN DE RIESGO			
ID. DEL RIESGO: R03		**FECHA:** 10/11/2018	
Probabilidad: Alta **Valor:** 3	**Impacto:** Alto **Valor:** 3	**Exposición:** Alta **Valor:** 9	**Prioridad:**1
DESCRIPCIÓN: Funcionalidades desarrolladas incorrectamente.			
REFINAMIENTO: **Causas:** • Los anteriores desarrolladores desconocían el proceso. • Falla en la recaudación de requerimientos. • Falta de comunicación. **Consecuencia:** • Retraso del proyecto. • Suspensión temporal o definitiva del proyecto.			
REDUCCIÓN: • Reuniones con el nivel estratégico para plantear una nueva planificación. • Solucionar el error en el momento.			
SUPERVISIÓN: • Verificar el correcto cumplimiento del reglamento.			
GESTIÓN: • Implicar a los encargados del proyecto con entregables funcionales que lo hagan participe al proyecto.			
ESTADO ACTUAL: - Fase de reducción Iniciada [redacted] - Fase de supervisión Iniciada [redacted] - Gestionado el Riesgo [redacted]			
RESPONSABLE: - Sandra Ordoñez - Kevin Chimbo			

HOJA DE GESTIÓN DE RIESGO			
ID. DEL RIESGO: R04		**FECHA:** 10/11/2018	
Probabilidad: Baja **Valor:** 1	**Impacto:** Moderado **Valor:** 2	**Exposición:** Baja **Valor:** 2	**Prioridad:** 3
DESCRIPCIÓN: Servicios web externos no funcionales.			
REFINAMIENTO: **Causas:** • Mal desarrollo de los servicios web. • Servidores no funcionales. **Consecuencia:** • Retraso del proyecto.			
REDUCCIÓN: • Involucrar a los encargados del proyecto las pruebas del sistema con los entregables para tratar de encontrar una solución temprana.			
SUPERVISIÓN: • Documentación de las evaluaciones para poder aplicar mejoras.			
GESTIÓN: • Mejorar el proceso en el que se cometió el error. • Documentación de las evaluaciones para poder aplicar mejoras.			
ESTADO ACTUAL: - Fase de reducción Iniciada ■ - Fase de supervisión Iniciada ■ - Gestionado el Riesgo ☐			
RESPONSABLE: - Sandra Ordoñez - Kevin Chimbo			

HOJA DE GESTIÓN DE RIESGO			
ID. DEL RIESGO: R05		**FECHA:** 10/11/2018	
Probabilidad: Media **Valor:** 2	**Impacto:** Moderado **Valor:** 2	**Exposición:** Moderados **Valor:** 4	**Prioridad:** 2
DESCRIPCIÓN: Acoplarse a las herramientas y técnicas del sistema desarrollado			

REFINAMIENTO:

Causas:

- Falta de experiencia.

Consecuencia:

- Retraso del proyecto

REDUCCIÓN:

- Involucrar a los encargados del proyecto las pruebas del sistema con los entregables para tratar de encontrar una solución temprana.

SUPERVISIÓN:

- Documentación de las evaluaciones para poder aplicar mejoras.

GESTIÓN:

- Documentación de las evaluaciones para poder aplicar mejoras.
- Integrar un diseñador gráfico al equipo de proyecto.
- Mejorar el proceso en el que se cometió el error.

ESTADO ACTUAL:

- Fase de reducción Iniciada ■
- Fase de supervisión Iniciada ■
- Gestionado el Riesgo ☐

RESPONSABLE:

- Sandra Ordoñez
- Kevin Chimbo

Anexo J: Historias de Usuario, técnicas, tareas de ingeniería y pruebas de aceptación

HU-19 Gestión tipo designación para investigaciones

Historia de Usuario	
Número: HU_19	**Nombre de la Historia:** Gestión tipo designación para investigaciones.
Modificación de historia de usuario:	
Usuario: Vicedecano	**Sprint Asignada:** 1
Prioridad en el Negocio: Media	**Puntos Estimados:** 60
Riesgo en el Desarrollo: Medio	**Puntos Reales:** 60
Descripción: Como vicedecano quiero poder crear un tipo designación para investigaciones.	
Observaciones:	

Historia de Usuario (Reverso) Pruebas de Aceptación
• Verificar que se ingrese, modifique, elimine y visualice el tipo designación para investigación. • Verificar que los datos ingresados, modificados o eliminados se guarden correctamente.

TAREA DE INGENIERÍA	
Historia de Usuario: Gestión tipo designación para investigaciones.	
Número de Tarea: TI_01	**Nombre de Tarea:** Crear las clases para el ingreso, modificaciones, eliminación y visualización del tipo designación en las capas interfaz de usuario y en el acceso a datos
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 30
Fecha Inicio: 17/12/2018	**Fecha Fin:** 21/12/2018
Programador Responsable: Kevin Chimbo	
Descripción: Como desarrollador necesito crear las clases necesarias para la gestión de tipos designación para un posterior uso.	
PRUEBAS DE ACEPTACIÓN	
• Comprobar que se hayan creado correctamente las clases. • Comprobar que cada clase tenga los atributos y métodos necesarios.	

TAREA DE INGENIERÍA	
Historia de Usuario: Gestión tipo designación para investigaciones	
Número de Tarea: TI_02	**Nombre de la Tarea:** Crear la interfaz de usuarios para el ingreso de cada tipo designación.
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 30
Fecha Inicio: 26/12/2018	**Fecha Fin:** 03/01/2019
Programador Responsable: Kevin Chimbo	
Descripción: Como desarrollador necesito crear la interfaz de usuario para el ingreso de tipo designación.	
PRUEBAS DE ACEPTACIÓN	
• Verificar si la interfaz de usuario correspondiente cumple con el estándar establecido. • Verificar que la interfaz tenga los campos necesarios requeridos por la funcionalidad.	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-HU_19	**Nombre de la Historia**: Gestión tipo designación para investigaciones.
Nombre: Verificar que se ingrese, modifique, elimine y visualice el tipo designación para investigación.	
Responsable: Sandra Ordoñez	**Fecha:** 03/01/2019
Descripción: Verificar que los tipos designación se pueda gestionar es decir guardar, eliminar, actualizar y visualizar cada uno de ellos.	
Condiciones de Ejecución: • Que ya este creado el acceso a datos y la interfaz de usuario.	
Pasos de Ejecución: • Para realizar un ingreso debe dirigirse a la opción tipo designación en el cual debe seleccionar una de las secciones presentadas. • Seleccionar uno de los ítems en el cual va a crear el tipo designación • Click en nuevo tipo designación al seleccionar el tipo designación podrá visualizar el nombre del ítem en el que va a crear al dar clic en el nombre podrá visualizar el campo a llenar con el nombre del tipo designación. • Para modificar debe seleccionar la sección, así como el ítem dar clic en el nombre del ítem y podrá modificar el nombre del tipo designación seleccionado • Para poder eliminar este se lo realizara siempre y cuando no posea tipo función • Para poder visualizar debe dar clic en guardar datos	

Resultado Esperado: Que se puedan ingresar, modificar, eliminar y visualizar la estructura de la designación.
Evaluación de la Prueba: Exitosa

PRUEBA DE ACEPTACIÓN	
Código: PA_02-HU_19	**Nombre de la Historia**: Gestión tipo designación para investigaciones.
Nombre: Verificar que los datos ingresados, modificados y eliminados se guarden correctamente.	
Responsable: Sandra Ordoñez	**Fecha:** 03/01/2019
Descripción: Comprobar que los datos ingresados, modificados y eliminados se guarden correctamente.	
Condiciones de Ejecución: • Que ya este creado el acceso a datos y la interfaz de usuario.	
Pasos de Ejecución: • Modificar, eliminar o ingresar los datos • Ingresar a la base de datos • Verificar que los datos se hayan alterado correctamente en base a la acción realizada.	
Resultado Esperado: Que la información sea guardada correctamente después de modificar eliminar e ingresar datos.	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_01	**Tarea Ingeniería:** TI_01 Crear las clases para el ingreso, modificaciones, eliminación y visualización del tipo designación en las capas interfaz de usuario y en el acceso a datos.
Nombre: Comprobar que se hayan creado correctamente las clases	
Responsable: Sandra Ordoñez	**Fecha:** 19/12/2018
Descripción: Se verificará que las clases sean creadas correctamente para ingresar, modificar, liminar y visualizar los tipos de designación en las capas de interfaz de usuario y accesos a datos.	
Condiciones de Ejecución: • Que ya este creado el acceso a datos y la interfaz de usuario.	
Pasos de Ejecución: • Revisar que las clases existan en la capa de acceso	

• Debe contar con todas las clases y cada una de ellas se encuentra heredadas de las clases de clase común.
Resultado Esperado: Que las clases se hayan creado correctamente.
Evaluación de la Prueba: Exitosa

PRUEBA DE ACEPTACIÓN	
Código: PA_02-TI_01	**Tarea Ingeniería:** TI_01 Crear las clases para el ingreso, modificaciones, eliminación y visualización de tipo designación en la capa de acceso.
Nombre: Comprobar que cada clase tenga los atributos y métodos necesarios.	
Responsable: Sandra Ordoñez	**Fecha:** 21/12/2018
Descripción: Se verificará que cada clase creada tenga los atributos y métodos necesarios.	
Condiciones de Ejecución: • Que ya este creado el acceso a datos y la interfaz de usuario. • Que ya este creada la clase común.	
Pasos de Ejecución: • Compara que todos los atributos estén acordes con la base de datos • Verificar que todos los métodos estén creados en base a la funcionalidad requerida y a los atributos antes definidos.	
Resultado Esperado: Que se haya comprobado que cada clase creada cuente con los atributos y métodos necesarios.	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_02	**Tarea Ingeniería:** TI_02 Crear la interfaz de usuarios para el ingreso de cada tipo designación.
Nombre: Verificar si la interfaz de usuario correspondiente cumple con el estándar establecido.	
Responsable: Sandra Ordoñez	**Fecha:** 28/12/2018
Descripción: Verificar que la interfaz realizada cumpla con los parámetros del estándar establecido.	
Condiciones de Ejecución • Controlador debe estar creado. • La vista debe estar creado.	
Pasos de Ejecución: • Escoger el rol de vicedecano	

• Ir al menú • Dar clic en gestión tipo designación • Verificar interfaz
Resultado Esperado: La interfaz cumple con el estándar establecido
Evaluación de la Prueba: Exitosa

PRUEBA DE ACEPTACIÓN	
Código: PA_02-TI_02	**Tarea Ingeniería:** TI_02 Crear la interfaz de usuarios para el ingreso de cada tipo designación.
Nombre: Verificar que la interfaz tenga los campos necesarios requeridos por la funcionalidad.	
Responsable: Sandra Ordoñez	**Fecha:** 03/01/2019
Descripción: Verificar que la interfaz de usuario tenga los campos necesarios que requiere la funcionalidad.	
Condiciones de Ejecución • Que ya este creado el acceso a datos y la interfaz de usuario. • Que ya este creada la clase común.	
Pasos de Ejecución: • Loguearse en el CAS • Dar click en tipo designación • Compara los datos que requiere la funcionalidad con la interfaz de usuario	
Resultado Esperado: La interfaz cumple con el estándar establecido	
Evaluación de la Prueba: Exitosa	

HU-20 Gestión designación para investigaciones

Historia de Usuario	
Número: HU_20	**Nombre de la Historia:** Gestión designación para investigaciones.
Modificación de historia de usuario:	
Usuario: Vicedecano	**Sprint Asignada:** 1
Prioridad en el Negocio: Media	**Puntos Estimados:** 60
Riesgo en el Desarrollo: Medio	**Puntos Reales:** 60
Descripción: Como vicedecano quiero poder designar al docente un evento.	
Observaciones:	

Historia de Usuario (Reverso) Pruebas de Aceptación
• Verificar que se ingrese, modifique, elimine y visualice la designación para investigación. • Verificar que los datos ingresados, modificados o eliminados se guarden correctamente.

TAREA DE INGENIERÍA	
Historia de Usuario: Gestión designación para investigaciones.	
Número de Tarea: TI_01	**Nombre de Tarea:** Crear las clases para el ingreso, modificaciones, eliminación y visualización de la designación en las capas interfaz de usuario y en el acceso a datos
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 30
Fecha Inicio: 17/12/2018	**Fecha Fin:** 21/12/2018
Programador Responsable: Sandra Ordoñez	
Descripción: Como desarrollador necesito crear las clases necesarias para la gestión de designación para un posterior uso.	
PRUEBAS DE ACEPTACIÓN	
• Comprobar que se hayan creado correctamente las clases. • Comprobar que cada clase tenga los atributos y métodos necesarios.	

TAREA DE INGENIERÍA	
Historia de Usuario: Gestión designación para investigaciones	
Número de Tarea: TI_02	**Nombre de la Tarea:** Crear la interfaz de usuarios para el ingreso de cada designación.
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 30
Fecha Inicio: 26/12/2018	**Fecha Fin:** 03/01/2019
Programador Responsable: Sandra Ordoñez	
Descripción: Como desarrollador necesito crear la interfaz de usuario para el ingreso de la designación.	
PRUEBAS DE ACEPTACIÓN	
• Verificar si la interfaz de usuario correspondiente cumple con el estándar establecido. • Verificar que la interfaz tenga los campos necesarios requeridos por la funcionalidad.	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-HU_20	**Nombre de la Historia**: Gestión designación para investigaciones.
Nombre: Verificar que se ingrese, modifique, elimine y visualice la designación para investigación.	
Responsable: Kevin Chimbo	**Fecha:** 03/01/2019
Descripción: Verificar que la designación se pueda gestionar es decir guardar, eliminar, actualizar y visualizar cada uno de ellos.	
Condiciones de Ejecución: • Que ya este creado el acceso a datos y la interfaz de usuario.	
Pasos de Ejecución: • Para realizar un ingreso debe dirigirse a la opción designación en el cual debe seleccionar una de las secciones presentadas. • Seleccionar uno de los ítems en el cual va asignar la designación • Click en nueva designación al seleccionar la designación podrá visualizar el nombre del ítem en el que va a designar al dar clic en el nombre podrá visualizar el campo a llenar con el nombre de la designación. • Para modificar debe seleccionar la sección, así como el ítem dar clic en el nombre del ítem y podrá modificar el nombre de la designación seleccionado • Para poder eliminar este se lo realizara siempre y cuando no posea tipo función • Para poder visualizar debe dar clic en guardar datos	
Resultado Esperado: Que se puedan ingresar, modificar, eliminar y visualizar la estructura de la designación.	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_02-HU_20	**Nombre de la Historia**: Gestión designación para investigaciones.
Nombre: Verificar que los datos ingresados, modificados y eliminados se guarden correctamente.	
Responsable: Kevin Chimbo	**Fecha:** 03/01/2019
Descripción: Comprobar que los datos ingresados, modificados y eliminados se guarden correctamente.	
Condiciones de Ejecución: • Que ya este creado el acceso a datos y la interfaz de usuario.	
Pasos de Ejecución:	

• Modificar, eliminar o ingresar los datos • Ingresar a la base de datos • Verificar que los datos se hayan alterado correctamente en base a la acción realizada.
Resultado Esperado: Que la información sea guardada correctamente después de modificar eliminar e ingresar datos.
Evaluación de la Prueba: Exitosa

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_01	**Tarea Ingeniería:** TI_01 Crear las clases para el ingreso, modificaciones, eliminación y visualización de la designación en las capas interfaz de usuario y en el acceso a datos.
Nombre: Comprobar que se hayan creado correctamente las clases	
Responsable: Kevin Chimbo	**Fecha:** 19/12/2018
Descripción: Se verificará que las clases sean creadas correctamente para ingresar, modificar, liminar y visualizar la designación en las capas de interfaz de usuario y accesos a datos.	
Condiciones de Ejecución: • Que ya este creado el acceso a datos y la interfaz de usuario.	
Pasos de Ejecución: • Revisar que las clases existan en la capa de acceso • Debe contar con todas las clases y cada una de ellas se encuentra heredadas de las clases de clase común.	
Resultado Esperado: Que las clases se hayan creado correctamente.	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_02-TI_01	**Tarea Ingeniería:** TI_01 Crear las clases para el ingreso, modificaciones, eliminación y visualización de la designación en la capa de acceso.
Nombre: Comprobar que cada clase tenga los atributos y métodos necesarios.	
Responsable: Kevin Chimbo	**Fecha:** 21/12/2018
Descripción: Se verificará que cada clase creada tenga los atributos y métodos necesarios.	
Condiciones de Ejecución: • Que ya este creado el acceso a datos y la interfaz de usuario. • Que ya este creada la clase común.	

Pasos de Ejecución: • Compara que todos los atributos estén acordes con la base de datos • Verificar que todos los métodos estén creados en base a la funcionalidad requerida y a los atributos antes definidos.
Resultado Esperado: Que se haya comprobado que cada clase creada cuente con los atributos y métodos necesarios.
Evaluación de la Prueba: Exitosa

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_02	**Tarea Ingeniería:** TI_02 Crear la interfaz de usuarios para el ingreso de cada designación.
Nombre: Verificar si la interfaz de usuario correspondiente cumple con el estándar establecido.	
Responsable: Kevin Chimbo	**Fecha:** 28/12/2018
Descripción: Verificar que la interfaz realizada cumpla con los parámetros del estándar establecido.	
Condiciones de Ejecución • Controlador debe estar creado. • La vista debe estar creado.	
Pasos de Ejecución: • Escoger el rol de vicedecano • Ir al menú • Dar clic en gestión designación • Verificar interfaz	
Resultado Esperado: La interfaz cumple con el estándar establecido	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_02-TI_02	**Tarea Ingeniería:** TI_02 Crear la interfaz de usuarios para el ingreso de cada designación.
Nombre: Verificar que la interfaz tenga los campos necesarios requeridos por la funcionalidad.	
Responsable: Kevin Chimbo	**Fecha:** 03/01/2019
Descripción: Verificar que la interfaz de usuario tenga los campos necesarios que requiere la funcionalidad.	
Condiciones de Ejecución	

• Que ya este creado el acceso a datos y la interfaz de usuario. • Que ya este creada la clase común.
Pasos de Ejecución: • Loguearse en el CAS • Dar click en tipo designación • Compara los datos que requiere la funcionalidad con la interfaz de usuario
Resultado Esperado: La interfaz cumple con el estándar establecido
Evaluación de la Prueba: Exitosa

HU-21 Control de roles

Historia de Usuario	
Número: HU_21	**Nombre de la Historia:** Control de roles
Modificación de historia de usuario:	
Usuario: Administrador	**Sprint Asignada:** 2
Prioridad en el Negocio: Media	**Puntos Estimados:** 30
Riesgo en el Desarrollo: Medio	**Puntos Reales:** 30
Descripción: Como administrador quiero poder realizar el control de roles	
Observaciones:	

Historia de Usuario (Reverso) Pruebas de Aceptación
• Verificar que devuelva el rol o los roles al ingresar un número de cédula • Verificar si los roles de la persona autenticada son correctos

TAREA DE INGENIERÍA	
Historia de Usuario: Control de roles	
Número de Tarea: TI_01	**Nombre de Tarea:** Crear la interfaz de usuario
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 15
Fecha Inicio: 04/12/2018	**Fecha Fin:** 08/12/2018
Programador Responsable: Kevin Chimbo	
Descripción: Como desarrollador necesito crear la interfaz de usuario para el control de roles.	
PRUEBAS DE ACEPTACIÓN	
• Verificar si la interfaz de usuario correspondiente cumple con el estándar establecido.	

TAREA DE INGENIERÍA	
Historia de Usuario: Control de roles	
Número de Tarea: TI_02	**Nombre de Tarea:** Crear el método AutenticarUsuario en el servicio web
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 15
Fecha Inicio: 09/01/2019	**Fecha Fin:** 10/01/2019
Programador Responsable: Kevin Chimbo	
Descripción: Como desarrollador necesito crear el método AutenticarUsuario en el servicio web.	
PRUEBAS DE ACEPTACIÓN	
• Comprobar que se hayan creado correctamente las clases. • Comprobar que cada clase tenga los atributos y métodos necesarios.	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-HU_21	**Nombre de la Historia**: Control de roles
Nombre: Verificar que devuelva el rol o los roles al ingresar un número de cédula	
Responsable: Sandra Ordoñez	**Fecha:** 10/01/2019
Descripción: Verificar que se devuelva el rol o los roles al ingresar el número de cedula de un docente.	
Condiciones de Ejecución: • Que ya este creado el acceso a datos y la interfaz de usuario.	
Pasos de Ejecución: • Ingresar al sistema de estafetas • Loguearse en el CAS	
Resultado Esperado: Que devuelva el rol o los roles al ingresar un número de cédula	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_02-HU_21	**Nombre de la Historia**: Control de roles
Nombre: Verificar si los roles de la persona autenticada son correctos	
Responsable: Sandra Ordoñez	**Fecha:** 10/01/2019
Descripción: Comprobar que los roles asignados al docente sean los correctos	
Condiciones de Ejecución: • Que ya este creado el acceso a datos y la interfaz de usuario.	

Pasos de Ejecución: • Ejecutar el método que devuelve el decano, vicedecano, docentes • Compara con los roles obtenidos en la interfaz de usuario.
Resultado Esperado: Que los roles sean los correctos después de loguearse
Evaluación de la Prueba: Exitosa

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_01	**Tarea Ingeniería:** TI_01 Crear la interfaz de usuario
Nombre: Verificar si la interfaz de usuario correspondiente cumple con el estándar establecido.	
Responsable: Sandra Ordoñez	**Fecha:** 08/01/2019
Descripción: Verificar que la interfaz realizada cumpla con los parámetros del estándar establecido.	
Condiciones de Ejecución • Controlador debe estar creado. • La vista debe estar creado.	
Pasos de Ejecución: • Escoger el rol de administrador • Ir al menú • Dar clic en control de roles • Verificar interfaz	
Resultado Esperado: La interfaz cumple con el estándar establecido	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_02	**Tarea Ingeniería:** TI_02 Crear el método AutenticarUsuario en el servicio web
Nombre: Comprobar que se hayan creado correctamente las clases	
Responsable: Sandra Ordoñez	**Fecha:** 09/01/2019
Descripción: Se verificará que las clases sean creadas correctamente en las capas de interfaz de usuario y accesos a datos.	
Condiciones de Ejecución: • Que ya este creado el acceso a datos y la interfaz de usuario.	
Pasos de Ejecución: • Revisar que las clases existan en la capa de acceso	

• Debe contar con todas las clases y cada una de ellas se encuentra heredadas de las clases de clase común.
Resultado Esperado: Que las clases se hayan creado correctamente.
Evaluación de la Prueba: Exitosa

PRUEBA DE ACEPTACIÓN	
Código: PA_02-TI_02	**Tarea Ingeniería:** TI_02 Crear el método AutenticarUsuario en el servicio web
Nombre: Comprobar que cada clase tenga los atributos y métodos necesarios.	
Responsable: Sandra Ordoñez	**Fecha:** 010/01/2019
Descripción: Se verificará que cada clase creada tenga los atributos y métodos necesarios.	
Condiciones de Ejecución: • Que ya este creado el acceso a datos y la interfaz de usuario. • Que ya este creada la clase común.	
Pasos de Ejecución: • Compara que todos los atributos estén acordes con la base de datos • Verificar que todos los métodos estén creados en base a la funcionalidad requerida y a los atributos antes definidos.	
Resultado Esperado: Que se haya comprobado que cada clase creada cuente con los atributos y métodos necesarios.	
Evaluación de la Prueba: Exitosa	

HU-22 Aprobación del horario del docente

Historia de Usuario	
Número: HU_22	**Nombre de la Historia:** Aprobación del horario del docente
Modificación de historia de usuario:	
Usuario: Vicedecano	**Sprint Asignada:** 2
Prioridad en el Negocio: Media	**Puntos Estimados:** 30
Riesgo en el Desarrollo: Medio	**Puntos Reales:** 30
Descripción: Como vicedecano quiero poder aprobar en horario del docente	
Observaciones:	

Historia de Usuario (Reverso) Pruebas de Aceptación
• Verificar que se apruebe correctamente el horario del docente

TAREA DE INGENIERÍA	
Historia de Usuario: Aprobación del horario del docente	
Número de Tarea: TI_01	**Nombre de Tarea:** Crear las clases en las capas interfaz de usuario y en el acceso a datos
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 15
Fecha Inicio: 04/01/2019	**Fecha Fin:** 08/01/2019
Programador Responsable: Sandra Ordoñez	
Descripción: Como desarrollador necesito crear las clases necesarias para la aprobación del horario del docente	
PRUEBAS DE ACEPTACIÓN	
• Comprobar que se hayan creado correctamente las clases. • Comprobar que cada clase tenga los atributos y métodos necesarios.	

TAREA DE INGENIERÍA	
Historia de Usuario: Aprobación del horario del docente	
Número de Tarea: TI_02	**Nombre de la Tarea:** Crear la interfaz de usuario
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 15
Fecha Inicio: 09/01/2019	**Fecha Fin:** 10/01/2019
Programador Responsable: Sandra Ordoñez	
Descripción: Como desarrollador necesito crear la interfaz de usuario para la aprobación del horario del docente	
PRUEBAS DE ACEPTACIÓN	
• Verificar si la interfaz de usuario correspondiente cumple con el estándar establecido. • Verificar que la interfaz tenga los campos necesarios requeridos por la funcionalidad.	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-HU_22	**Nombre de la Historia**: Aprobación del horario del docente
Nombre: Verificar que se apruebe correctamente el horario del docente	
Responsable: Kevin Chimbo	**Fecha:** 10/01/2019
Descripción: Verificar que el horario del docente se apruebe correctamente	
Condiciones de Ejecución:	

• Que ya este creado el acceso a datos y la interfaz de usuario.
Pasos de Ejecución: • Ingresar al sistema de estafetas • Loguearse en el CAS • Seleccionar el rol de vicedecano • Escoger la opción Aprobar estafeta • Seleccionar el docente • Click en el botón aprobar • Observar en la interfaz en la fila del docente seleccionado se muestra el estado en aprobado.
Resultado Esperado: Que se apruebe la estafeta de forma correcta
Evaluación de la Prueba: Exitosa

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_01	**Tarea Ingeniería:** TI_01 Crear las clases en las capas interfaz de usuario y en el acceso a datos
Nombre: Comprobar que se hayan creado correctamente las clases.	
Responsable: Kevin Chimbo	**Fecha:** 07/01/2019
Descripción: Se verificará que las clases sean creadas correctamente en las capas de interfaz de usuario y accesos a datos.	
Condiciones de Ejecución: • Que ya este creado el acceso a datos	
Pasos de Ejecución: • Revisar que las clases existan en la capa de acceso • Revisar que se tenga todas las clases y cada una de ellas se encuentra heredadas de las clases de clase común.	
Resultado Esperado: Que las clases se hayan creado correctamente.	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_02-TI_01	**Tarea Ingeniería:** TI_01 Crear las clases en las capas interfaz de usuario y en el acceso a datos
Nombre: Comprobar que cada clase tenga los atributos y métodos necesarios.	
Responsable: Kevin Chimbo	**Fecha:** 08/01/2019
Descripción: Se verificará que cada clase creada tenga los atributos y métodos necesarios.	
Condiciones de Ejecución: • Que ya este creado el acceso a datos y la interfaz de usuario. • Que ya este creada la clase común.	
Pasos de Ejecución: • Compara que todos los atributos estén acordes con la base de datos • Verificar que todos los métodos estén creados en base a la funcionalidad requerida y a los atributos antes definidos.	
Resultado Esperado: Que se haya comprobado que cada clase creada cuente con los atributos y métodos necesarios.	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_02	**Tarea Ingeniería:** TI_02 Crear la interfaz de usuario
Nombre: Verificar si la interfaz de usuario correspondiente cumple con el estándar establecido.	
Responsable: Kevin Chimbo	**Fecha:** 09/01/2019
Descripción: Verificar que la interfaz realizada cumpla con los parámetros del estándar establecido.	
Condiciones de Ejecución • Controlador debe estar creado. • La vista debe estar creado.	
Pasos de Ejecución: • Escoger el rol de vicedecano • Ir al menú • Dar clic en aprobación del horario del docente • Verificar interfaz	
Resultado Esperado: La interfaz cumple con el estándar establecido	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_02-TI_02	**Tarea Ingeniería:** TI_02 Crear la interfaz de usuario
Nombre: Verificar que la interfaz tenga los campos necesarios requeridos por la funcionalidad.	
Responsable: Kevin Chimbo	**Fecha:** 10/01/2019
Descripción: Verificar que la interfaz de usuario tenga los campos necesarios que requiere la funcionalidad.	
Condiciones de Ejecución • Que ya este creado el acceso a datos y la interfaz de usuario. • Que ya este creada la clase común.	
Pasos de Ejecución: • Loguearse en el CAS • Dar click en aprobar horario docente • Compara los datos que requiere la funcionalidad con la interfaz de usuario	
Resultado Esperado: La interfaz cumple con el estándar establecido	
Evaluación de la Prueba: Exitosa	

HU-23 Resumen de horas docencia

Historia de Usuario	
Número: HU_23	**Nombre de la Historia:** Resumen de horas docencia
Modificación de historia de usuario:	
Usuario: Docente	**Sprint Asignada:** 2
Prioridad en el Negocio: Media	**Puntos Estimados:** 30
Riesgo en el Desarrollo: Medio	**Puntos Reales:** 30
Descripción: Como docente quiero poder observar el resumen de horas	
Observaciones:	

Historia de Usuario (Reverso) Pruebas de Aceptación
• Verificar que el listado pertenezca al docente seleccionado

TAREA DE INGENIERÍA	
Historia de Usuario: Resumen de horas docencia	
Número de Tarea: TI_01	**Nombre de Tarea:** Crear las clases en las capas interfaz de usuario y en el acceso a datos
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 15
Fecha Inicio: 11/01/2019	**Fecha Fin:** 14/01/2019
Programador Responsable: Kevin Chimbo	
Descripción: Como desarrollador necesito crear las clases necesarias para el resumen de horas para la docencia.	
PRUEBAS DE ACEPTACIÓN	
• Comprobar que se hayan creado correctamente las clases. • Comprobar que cada clase tenga los atributos y métodos necesarios.	

TAREA DE INGENIERÍA	
Historia de Usuario: Resumen de horas docencia	
Número de Tarea: TI_02	**Nombre de la Tarea:** Crear la interfaz de usuario
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 15
Fecha Inicio: 15/01/2019	**Fecha Fin:** 17/01/2019
Programador Responsable: Kevin Chimbo	
Descripción: Como desarrollador necesito crear la interfaz de usuario para obtener el resumen de horas para la docencia	
PRUEBAS DE ACEPTACIÓN	
• Verificar si la interfaz de usuario correspondiente cumple con el estándar establecido • Verificar que la interfaz muestre los campos necesarios requeridos por la funcionalidad	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-HU_23	**Nombre de la Historia**: Resumen de horas docencia
Nombre: Verificar que el listado pertenezca al docente seleccionado	
Responsable: Sandra Ordoñez	**Fecha:** 10/01/2019
Descripción: Verificar q se muestre el resumen de horas docencia	
Condiciones de Ejecución: • Que ya este creado el acceso a datos y la interfaz de usuario.	
Pasos de Ejecución:	

• Ingresar al sistema de estafetas • Loguearse en el CAS • Seleccionar el rol de docente • Dar clic en el menú • Seleccionar el docente • Click en el botón en la columna resumen de horas del docente que se desea observar • Observar en la interfaz el resumen de horas del docente
Resultado Esperado: Que se observe el resumen de horas del docente
Evaluación de la Prueba: Exitosa

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_01	**Tarea Ingeniería:** TI_01 Crear las clases en las capas interfaz de usuario y en el acceso a datos
Nombre: Comprobar que se hayan creado correctamente las clases.	
Responsable: Sandra Ordoñez	**Fecha:** 11/01/2019
Descripción: Se verificará que las clases sean creadas correctamente en las capas de interfaz de usuario y accesos a datos.	
Condiciones de Ejecución: • Que ya este creado el acceso a datos	
Pasos de Ejecución: • Revisar que las clases existan en la capa de acceso • Revisar que se tenga todas las clases y cada una de ellas se encuentra heredadas de las clases de clase común.	
Resultado Esperado: Que las clases se hayan creado correctamente.	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_02-TI_01	**Tarea Ingeniería:** TI_01 Crear las clases en las capas interfaz de usuario y en el acceso a datos
Nombre: Comprobar que cada clase tenga los atributos y métodos necesarios.	
Responsable: Sandra Ordoñez	**Fecha:** 14/01/2019
Descripción: Se verificará que cada clase creada tenga los atributos y métodos necesarios.	
Condiciones de Ejecución: • Que ya este creado el acceso a datos y la interfaz de usuario. • Que ya este creada la clase común.	
Pasos de Ejecución: • Compara que todos los atributos estén acordes con la base de datos • Verificar que todos los métodos estén creados en base a la funcionalidad requerida y a los atributos antes definidos.	
Resultado Esperado: Que se haya comprobado que cada clase creada cuente con los atributos y métodos necesarios.	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_02	**Tarea Ingeniería:** TI_02 Crear la interfaz de usuario
Nombre: Verificar si la interfaz de usuario correspondiente cumple con el estándar establecido.	
Responsable: Sandra Ordoñez	**Fecha:** 16/01/2019
Descripción: Verificar que la interfaz realizada cumpla con los parámetros del estándar establecido.	
Condiciones de Ejecución • Controlador debe estar creado. • La vista debe estar creado.	
Pasos de Ejecución: • Escoger el rol de docente • Ir al menú • Dar clic en resumen de horas docencia • Verificar interfaz	
Resultado Esperado: La interfaz cumple con el estándar establecido	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_02-TI_02	**Tarea Ingeniería:** TI_02 Crear la interfaz de usuario
Nombre: Verificar que la interfaz tenga los campos necesarios requeridos por la funcionalidad.	
Responsable: Sandra Ordoñez	**Fecha:** 17/01/2019
Descripción: Verificar que la interfaz de usuario tenga los campos necesarios que requiere la funcionalidad.	
Condiciones de Ejecución • Que ya este creado el acceso a datos y la interfaz de usuario. • Que ya este creada la clase común.	
Pasos de Ejecución: • Loguearse en el CAS • Dar click en resumen de horas docencia • Compara los datos que requiere la funcionalidad con la interfaz de usuario	
Resultado Esperado: La interfaz cumple con el estándar establecido	
Evaluación de la Prueba: Exitosa	

HU-24 Resumen de horas gestión

Historia de Usuario	
Número: HU_24	**Nombre de la Historia:** Resumen de horas gestión
Modificación de historia de usuario:	
Usuario: Docente	**Sprint Asignada:** 2
Prioridad en el Negocio: Media	**Puntos Estimados:** 30
Riesgo en el Desarrollo: Medio	**Puntos Reales:** 30
Descripción: Como docente quiero poder observar el resumen de horas	
Observaciones:	

Historia de Usuario (Reverso) Pruebas de Aceptación
• Verificar que el listado pertenezca al docente seleccionado

TAREA DE INGENIERÍA	
Historia de Usuario: Resumen de horas gestión	
Número de Tarea: TI_01	**Nombre de Tarea:** Crear las clases en las capas interfaz de usuario y en el acceso a datos
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 15
Fecha Inicio: 11/01/2019	**Fecha Fin:** 14/01/2019
Programador Responsable: Sandra Ordoñez	
Descripción: Como desarrollador necesito crear las clases necesarias para el resumen de horas para la gestión.	
PRUEBAS DE ACEPTACIÓN	
• Comprobar que se hayan creado correctamente las clases. • Comprobar que cada clase tenga los atributos y métodos necesarios.	

TAREA DE INGENIERÍA	
Historia de Usuario: Resumen de horas gestión	
Número de Tarea: TI_02	**Nombre de la Tarea:** Crear la interfaz de usuario
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 15
Fecha Inicio: 15/01/2019	**Fecha Fin:** 17/01/2019
Programador Responsable: Sandra Ordoñez	
Descripción: Como desarrollador necesito crear la interfaz de usuario para obtener el resumen de horas para la gestión	
PRUEBAS DE ACEPTACIÓN	
• Verificar si la interfaz de usuario correspondiente cumple con el estándar establecido • Verificar que la interfaz muestre los campos necesarios requeridos por la funcionalidad	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-HU_24	**Nombre de la Historia**: Resumen de horas gestión
Nombre: Verificar que el listado pertenezca al docente seleccionado	
Responsable: Kevin Chimbo	**Fecha:** 10/01/2019
Descripción: Verificar q se muestre el resumen de horas de gestión	
Condiciones de Ejecución: • Que ya este creado el acceso a datos y la interfaz de usuario.	
Pasos de Ejecución:	

• Ingresar al sistema de estafetas • Loguearse en el CAS • Seleccionar el rol de docente • Dar clic en el menú • Seleccionar el docente • Click en el botón en la columna resumen de horas de la gestión que se desea observar • Observar en la interfaz el resumen de horas de la gestión
Resultado Esperado: Que se observe el resumen de horas de la gestión
Evaluación de la Prueba: Exitosa

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_01	**Tarea Ingeniería:** TI_01 Crear las clases en las capas interfaz de usuario y en el acceso a datos
Nombre: Comprobar que se hayan creado correctamente las clases.	
Responsable: Kevin Chimbo	**Fecha:** 11/01/2019
Descripción: Se verificará que las clases sean creadas correctamente en las capas de interfaz de usuario y accesos a datos.	
Condiciones de Ejecución: • Que ya este creado el acceso a datos	
Pasos de Ejecución: • Revisar que las clases existan en la capa de acceso • Revisar que se tenga todas las clases y cada una de ellas se encuentra heredadas de las clases de clase común.	
Resultado Esperado: Que las clases se hayan creado correctamente.	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_02-TI_01	**Tarea Ingeniería:** TI_01 Crear las clases en las capas interfaz de usuario y en el acceso a datos
Nombre: Comprobar que cada clase tenga los atributos y métodos necesarios.	
Responsable: Kevin Chimbo	**Fecha:** 14/01/2019
Descripción: Se verificará que cada clase creada tenga los atributos y métodos necesarios.	
Condiciones de Ejecución: • Que ya este creado el acceso a datos y la interfaz de usuario. • Que ya este creada la clase común.	
Pasos de Ejecución: • Compara que todos los atributos estén acordes con la base de datos • Verificar que todos los métodos estén creados en base a la funcionalidad requerida y a los atributos antes definidos.	
Resultado Esperado: Que se haya comprobado que cada clase creada cuente con los atributos y métodos necesarios.	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_02	**Tarea Ingeniería:** TI_02 Crear la interfaz de usuario
Nombre: Verificar si la interfaz de usuario correspondiente cumple con el estándar establecido.	
Responsable: Kevin Chimbo	**Fecha:** 16/01/2019
Descripción: Verificar que la interfaz realizada cumpla con los parámetros del estándar establecido.	
Condiciones de Ejecución • Controlador debe estar creado. • La vista debe estar creado.	
Pasos de Ejecución: • Escoger el rol de docente • Ir al menú • Dar clic en resumen de horas gestión • Verificar interfaz	
Resultado Esperado: La interfaz cumple con el estándar establecido	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_02-TI_02	**Tarea Ingeniería:** TI_02 Crear la interfaz de usuario
Nombre: Verificar que la interfaz tenga los campos necesarios requeridos por la funcionalidad.	
Responsable: Kevin Chimbo	**Fecha:** 17/01/2019
Descripción: Verificar que la interfaz de usuario tenga los campos necesarios que requiere la funcionalidad.	
Condiciones de Ejecución • Que ya este creado el acceso a datos y la interfaz de usuario. • Que ya este creada la clase común.	
Pasos de Ejecución: • Loguearse en el CAS • Dar click en resumen de horas gestión • Compara los datos que requiere la funcionalidad con la interfaz de usuario	
Resultado Esperado: La interfaz cumple con el estándar establecido	
Evaluación de la Prueba: Exitosa	

HU-25 Resumen de horas investigación

Historia de Usuario	
Número: HU_25	**Nombre de la Historia:** Resumen de horas investigación
Modificación de historia de usuario:	
Usuario: Docente	**Sprint Asignada:** 3
Prioridad en el Negocio: Media	**Puntos Estimados:** 30
Riesgo en el Desarrollo: Medio	**Puntos Reales:** 30
Descripción: Como docente quiero poder observar el resumen de horas	
Observaciones:	

Historia de Usuario (Reverso) Pruebas de Aceptación
• Verificar que el listado pertenezca al docente seleccionado

TAREA DE INGENIERÍA	
Historia de Usuario: Resumen de horas investigación	
Número de Tarea: TI_01	**Nombre de Tarea:** Crear las clases en las capas interfaz de usuario y en el acceso a datos
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 15
Fecha Inicio: 18/01/2019	**Fecha Fin:** 21/01/2019
Programador Responsable: Kevin Chimbo	
Descripción: Como desarrollador necesito crear las clases necesarias para el resumen de horas para investigación.	
PRUEBAS DE ACEPTACIÓN	
• Comprobar que se hayan creado correctamente las clases. • Comprobar que cada clase tenga los atributos y métodos necesarios.	

TAREA DE INGENIERÍA	
Historia de Usuario: Resumen de horas investigación	
Número de Tarea: TI_02	**Nombre de la Tarea:** Crear la interfaz de usuario
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 15
Fecha Inicio: 22/01/2019	**Fecha Fin:** 24/01/2019
Programador Responsable: Kevin Chimbo	
Descripción: Como desarrollador necesito crear la interfaz de usuario para obtener el resumen de horas para investigación	
PRUEBAS DE ACEPTACIÓN	
• Verificar si la interfaz de usuario correspondiente cumple con el estándar establecido • Verificar que la interfaz muestre los campos necesarios requeridos por la funcionalidad	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-HU_25	**Nombre de la Historia**: Resumen de horas investigación
Nombre: Verificar que el listado pertenezca al docente seleccionado	
Responsable: Sandra Ordoñez	**Fecha:** 24/01/2019
Descripción: Verificar q se muestre el resumen de horas de investigación	
Condiciones de Ejecución: • Que ya este creado el acceso a datos y la interfaz de usuario.	

Pasos de Ejecución:

- Ingresar al sistema de estafetas
- Loguearse en el CAS
- Seleccionar el rol de docente
- Dar clic en el menú
- Seleccionar el docente
- Click en el botón en la columna resumen de horas de investigación que se desea observar
- Observar en la interfaz el resumen de horas de investigación

Resultado Esperado: Que se observe el resumen de horas de investigación

Evaluación de la Prueba: Exitosa

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_01	**Tarea Ingeniería:** TI_01 Crear las clases en las capas interfaz de usuario y en el acceso a datos
Nombre: Comprobar que se hayan creado correctamente las clases.	
Responsable: Sandra Ordoñez	**Fecha:** 18/01/2019
Descripción: Se verificará que las clases sean creadas correctamente en las capas de interfaz de usuario y accesos a datos.	
Condiciones de Ejecución: • Que ya este creado el acceso a datos	
Pasos de Ejecución: • Revisar que las clases existan en la capa de acceso • Revisar que se tenga todas las clases y cada una de ellas se encuentra heredadas de las clases de clase común.	
Resultado Esperado: Que las clases se hayan creado correctamente.	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_02-TI_01	**Tarea Ingeniería:** TI_01 Crear las clases en las capas interfaz de usuario y en el acceso a datos
Nombre: Comprobar que cada clase tenga los atributos y métodos necesarios.	
Responsable: Sandra Ordoñez	**Fecha:** 21/01/2019
Descripción: Se verificará que cada clase creada tenga los atributos y métodos necesarios.	
Condiciones de Ejecución: • Que ya este creado el acceso a datos y la interfaz de usuario. • Que ya este creada la clase común.	
Pasos de Ejecución: • Compara que todos los atributos estén acordes con la base de datos • Verificar que todos los métodos estén creados en base a la funcionalidad requerida y a los atributos antes definidos.	
Resultado Esperado: Que se haya comprobado que cada clase creada cuente con los atributos y métodos necesarios.	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_02	**Tarea Ingeniería:** TI_02 Crear la interfaz de usuario
Nombre: Verificar si la interfaz de usuario correspondiente cumple con el estándar establecido.	
Responsable: Sandra Ordoñez	**Fecha:** 23/01/2019
Descripción: Verificar que la interfaz realizada cumpla con los parámetros del estándar establecido.	
Condiciones de Ejecución • Controlador debe estar creado. • La vista debe estar creado.	
Pasos de Ejecución: • Escoger el rol de docente • Ir al menú • Dar clic en resumen de horas investigación • Verificar interfaz	
Resultado Esperado: La interfaz cumple con el estándar establecido	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_02-TI_02	**Tarea Ingeniería:** TI_02 Crear la interfaz de usuario
Nombre: Verificar que la interfaz tenga los campos necesarios requeridos por la funcionalidad.	
Responsable: Sandra Ordoñez	**Fecha:** 24/01/2019
Descripción: Verificar que la interfaz de usuario tenga los campos necesarios que requiere la funcionalidad.	
Condiciones de Ejecución • Que ya este creado el acceso a datos y la interfaz de usuario. • Que ya este creada la clase común.	
Pasos de Ejecución: • Loguearse en el CAS • Dar click en resumen de horas investigación • Compara los datos que requiere la funcionalidad con la interfaz de usuario	
Resultado Esperado: La interfaz cumple con el estándar establecido	
Evaluación de la Prueba: Exitosa	

HU-26 Control de horas de ÍTEMS de docencia

Historia de Usuario	
Número: HU_26	**Nombre de la Historia:** Control de horas de ÍTEMS de docencia
Modificación de historia de usuario:	
Usuario: Docente	**Sprint Asignada:** 3
Prioridad en el Negocio: Media	**Puntos Estimados:** 30
Riesgo en el Desarrollo: Medio	**Puntos Reales:** 30
Descripción: Como docente quiero poder controlar las horas de los ítems de docencia	
Observaciones:	

Historia de Usuario (Reverso) Pruebas de Aceptación
• Verificar si las horas de docencia ingresadas por el docente son correctas en base a los máximos y mínimos de cada ítem según el reglamento.

TAREA DE INGENIERÍA

Historia de Usuario: Control de horas de ÍTEMS de docencia	
Número de Tarea: TI_01	**Nombre de Tarea:** Modificar la clase itemsDIU para controlar las horas de los ítems
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 30
Fecha Inicio: 18/01/2019	**Fecha Fin:** 24/01/2019
Programador Responsable: Sandra Ordoñez	
Descripción: Como desarrollador necesito modificar la clase evidenciaIU para el control de ítems de docencia	
PRUEBAS DE ACEPTACIÓN	
• Verificar las horas de docencia en la clase itemsDIU • Verificar que la interfaz muestre el estado correcto en base al número de horas	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-HU_26	**Nombre de la Historia**: Control de horas de ÍTEMS de docencia
Nombre: Verificar si las horas de docencia ingresadas por el docente son correctas en base a los máximos y mínimos de cada ítem según el reglamento	
Responsable: Kevin Chimbo	**Fecha:** 24/01/2019
Descripción: Verificar que las horas de docencia ingresadas por el docente sean correctas basándose en los máximos y en los mínimos de acuerdo al reglamento.	
Condiciones de Ejecución: • Que ya este creado el acceso a datos y la interfaz de usuario.	
Pasos de Ejecución: • Ingresar al sistema de estafetas • Loguearse en el CAS • Seleccionar el rol de docente • Dar clic en el menú • Click en el botón resumen de horas • Observar en la interfaz el resumen de horas de docencia del docente	
Resultado Esperado: Que las horas ingresadas sean correctas en base a los máximos y mininos de acuerdo al reglamento.	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN

Código: PA_01-TI_01	**Tarea Ingeniería:** TI_01 Modificar la clase itemsDIU para controlar las horas de los ítems
Nombre: Verificar las horas de docencia en la clase itemsDIU	
Responsable: Kevin Chimbo	**Fecha:** 22/01/2019
Descripción: Se verificará que las horas de docencia se encuentren en la clase itemsDIU	
Condiciones de Ejecución: • Que ya este creado el acceso a datos	
Pasos de Ejecución: • Abrir la clase itemsDIU • Verificar que cada ítem tenga su validación en base al reglamento	
Resultado Esperado: Que las horas se encuentren en la clase itemsDIU	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_02-TI_01	**Tarea Ingeniería:** TI_01 Modificar la clase itemsDIU para controlar las horas de los ítems
Nombre: Verificar que la interfaz muestre el estado correcto en base al número de horas	
Responsable: Kevin Chimbo	**Fecha:** 24/01/2019
Descripción: Se verificará que la interfaz muestre el estado correcto en base al número de horas	
Condiciones de Ejecución: • Que ya este creado el acceso a datos • Que ya este creada la clase común.	
Pasos de Ejecución: • Comparar el número de horas de docencia que el docente ingreso con el reglamento del distributivo de la jornada laboral semanal del docente • Verificar que sea correcto el número de horas de docencia ingresado por el docente	
Resultado Esperado: Que la interfaz haya mostrado el estado correcto en base al número de horas	
Evaluación de la Prueba: Exitosa	

HU-27 Control de horas de ÍTEMS de investigación

Historia de Usuario	
Número: HU_27	**Nombre de la Historia:** Control de horas de ÍTEMS de investigación
Modificación de historia de usuario:	
Usuario: Docente	**Sprint Asignada:** 3
Prioridad en el Negocio: Media	**Puntos Estimados:** 30
Riesgo en el Desarrollo: Medio	**Puntos Reales:** 30
Descripción: Como docente quiero poder controlar las horas de los ítems de investigación	
Observaciones:	

Historia de Usuario (Reverso) Pruebas de Aceptación
• Verificar si las horas de investigación ingresadas por el docente son correctas en base a los máximos y mínimos de cada ítem según el reglamento.

TAREA DE INGENIERÍA	
Historia de Usuario: Control de horas de ÍTEMS de investigación	
Número de Tarea: TI_01	**Nombre de Tarea:** Modificar la clase itemsDIU para controlar las horas de los ítems
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 30
Fecha Inicio: 25/01/2019	**Fecha Fin:** 31/01/2019
Programador Responsable: Kevin Chimbo	
Descripción: Como desarrollador necesito modificar la clase evidenciaIU para el control de horas de ítems de investigación	
PRUEBAS DE ACEPTACIÓN	
• Verificar las horas de investigación en la clase itemsDIU • Verificar que la interfaz muestre el estado correcto en base al número de horas	

PRUEBA DE ACEPTACIÓN

Código: PA_01-HU_27	**Nombre de la Historia**: Control de horas de ÍTEMS de investigación
Nombre: Verificar si las horas de investigación ingresadas por el docente son correctas en base a los máximos y mínimos de cada ítem según el reglamento	
Responsable: Sandra Ordoñez	**Fecha:** 31/01/2019
Descripción: Verificar que las horas de investigación ingresadas por el docente sean correctas basándose en los máximos y en los mínimos de acuerdo al reglamento.	
Condiciones de Ejecución: • Que ya este creado el acceso a datos y la interfaz de usuario.	
Pasos de Ejecución: • Ingresar al sistema de estafetas • Loguearse en el CAS • Seleccionar el rol de docente • Dar clic en el menú • Click en el botón resumen de horas • Observar en la interfaz el resumen de horas de investigación del docente	
Resultado Esperado: Que las horas ingresadas sean correctas en base a los máximos y mininos de acuerdo al reglamento.	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_01	**Tarea Ingeniería:** TI_01 Modificar la clase itemsDIU para controlar las horas de los ítems
Nombre: Verificar las horas de investigación en la clase itemsDIU	
Responsable: Sandra Ordoñez	**Fecha:** 29/01/2019
Descripción: Se verificará que las horas de investigación se encuentren en la clase itemsDIU	
Condiciones de Ejecución: • Que ya este creado el acceso a datos	
Pasos de Ejecución: • Abrir la clase itemsDIU • Verificar que cada ítem tenga su validación en base al reglamento	
Resultado Esperado: Que las horas se encuentren en la clase itemsDIU	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_02-TI_01	**Tarea Ingeniería:** TI_01 Modificar la clase itemsDIU para controlar las horas de los ítems
Nombre: Verificar que la interfaz muestre el estado correcto en base al número de horas	
Responsable: Sandra Ordoñez	**Fecha:** 31/01/2019
Descripción: Se verificará que la interfaz muestre el estado correcto en base al número de horas	
Condiciones de Ejecución: • Que ya este creado el acceso a datos • Que ya este creada la clase común.	
Pasos de Ejecución: • Comparar el número de horas de investigación que el docente ingreso con el reglamento del distributivo de la jornada laboral semanal del docente • Verificar que sea correcto el número de horas de investigación ingresado por el docente	
Resultado Esperado: Que la interfaz haya mostrado el estado correcto en base al número de horas	
Evaluación de la Prueba: Exitosa	

HU-28 Control de horas de ÍTEMS de gestión

Historia de Usuario	
Número: HU_28	**Nombre de la Historia:** Control de horas de ÍTEMS de gestión
Modificación de historia de usuario:	
Usuario: Docente	**Sprint Asignada:** 3
Prioridad en el Negocio: Media	**Puntos Estimados:** 30
Riesgo en el Desarrollo: Medio	**Puntos Reales:** 30
Descripción: Como docente quiero poder controlar las horas de los ítems de gestión	
Observaciones:	

Historia de Usuario (Reverso) Pruebas de Aceptación
• Verificar si las horas de gestión ingresadas por el docente son correctas en base a los máximos y mínimos de cada ítem según el reglamento.

TAREA DE INGENIERÍA	
Historia de Usuario: Control de horas de ÍTEMS de gestión	
Número de Tarea: TI_01	**Nombre de Tarea:** Modificar la clase itemsDIU para controlar las horas de los ítems
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 30
Fecha Inicio: 25/01/2019	**Fecha Fin:** 31/01/2019
Programador Responsable: Sandra Ordoñez	
Descripción: Como desarrollador necesito modificar la clase evidenciaIU para el control de horas de ítems de gestión	
PRUEBAS DE ACEPTACIÓN	
• Verificar las horas de gestión en la clase itemsDIU • Verificar que la interfaz muestre el estado correcto en base al número de horas	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-HU_28	**Nombre de la Historia**: Control de horas de ÍTEMS de gestión
Nombre: Verificar si las horas de gestión ingresadas por el docente son correctas en base a los máximos y mínimos de cada ítem según el reglamento.	
Responsable: Kevin Chimbo	**Fecha:** 31/01/2019
Descripción: Verificar que las horas de gestión ingresadas por el docente sean correctas basándose en los máximos y en los mínimos de acuerdo al reglamento.	
Condiciones de Ejecución: • Que ya este creado el acceso a datos y la interfaz de usuario.	
Pasos de Ejecución: • Ingresar al sistema de estafetas • Loguearse en el CAS • Seleccionar el rol de docente • Dar clic en el menú • Click en el botón resumen de horas • Observar en la interfaz el resumen de horas de gestión del docente	
Resultado Esperado: Que las horas ingresadas sean correctas en base a los máximos y mininos de acuerdo al reglamento.	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_01	**Tarea Ingeniería:** TI_01 Modificar la clase itemsDIU para controlar las horas de los ítems
Nombre: Verificar las horas de gestión en la clase itemsDIU	
Responsable: Kevin Chimo	**Fecha:** 29/01/2019
Descripción: Se verificará que las horas de gestión se encuentren en la clase itemsDIU	
Condiciones de Ejecución: • Que ya este creado el acceso a datos	
Pasos de Ejecución: • Abrir la clase itemsDIU • Verificar que cada ítem tenga su validación en base al reglamento	
Resultado Esperado: Que las horas se encuentren en la clase itemsDIU	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_02-TI_01	**Tarea Ingeniería:** TI_01 Modificar la clase itemsDIU para controlar las horas de los ítems
Nombre: Verificar que la interfaz muestre el estado correcto en base al número de horas	
Responsable: Kevin Chimbo	**Fecha:** 31/01/2019
Descripción: Se verificará que la interfaz muestre el estado correcto en base al número de horas	
Condiciones de Ejecución: • Que ya este creado el acceso a datos • Que ya este creada la clase común.	
Pasos de Ejecución: • Comparar el número de horas de gestión que el docente ingreso con el reglamento del distributivo de la jornada laboral semanal del docente • Verificar que sea correcto el número de horas de gestión ingresado por el docente	
Resultado Esperado: Que la interfaz haya mostrado el estado correcto en base al número de horas	
Evaluación de la Prueba: Exitosa	

HU-29 Modificar eventos del horario

Historia de Usuario	
Número: HU_29	**Nombre de la Historia:** Modificar eventos del horario
Modificación de historia de usuario:	
Usuario: Docente	**Sprint Asignada:** 4
Prioridad en el Negocio: Media	**Puntos Estimados:**30
Riesgo en el Desarrollo: Medio	**Puntos Reales:** 30
Descripción: Como docente quiero poder modificar eventos del horario.	
Observaciones:	

Historia de Usuario (Reverso) Pruebas de Aceptación
• Verificar que los datos de los eventos del horario se modifiquen correctamente • Verificar que los datos modificados se hayan guardado correctamente

TAREA DE INGENIERÍA	
Historia de Usuario: Modificar eventos del horario	
Número de Tarea: TI_01	**Nombre de Tarea:** Crear el método actualizarHorarioDocente en el servicio web del acceso a datos
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 15
Fecha Inicio: 01/02/2019	**Fecha Fin:** 04/02/2019
Programador Responsable: Kevin Chimbo	
Descripción: Como desarrollador quiero crear el método actualizarHorarioDocente en el servicio web del Acceso a Datos para poder obtener los datos y poder modificarlos.	
PRUEBAS DE ACEPTACIÓN	
• Verificar que el método creado modifique correctamente la fecha y hora del evento	

TAREA DE INGENIERÍA

Historia de Usuario: Modificar eventos del horario	
Número de Tarea: TI_02	**Nombre de Tarea:** T1_02 Crear la interfaz de usuario para modificar eventos del horario
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 15
Fecha Inicio: 05/02/2019	**Fecha Fin:** 07/02/2019
Programador Responsable: Kevin Chimbo	
Descripción: Creación la interfaz de usuario para modificar eventos del horario	
PRUEBAS DE ACEPTACIÓN	
• Verificar que la interfaz de usuario permita arrastrar un evento para modificarlo • Verificar si la interfaz de usuario correspondiente cumple con el estándar establecido	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-HU_29	**Nombre de la Historia**: Modificar eventos del horario
Nombre: Verificar que los datos de los eventos del horario se modifiquen correctamente	
Responsable: Sandra Ordoñez	**Fecha:** 07/02/2019
Descripción: Se obtendrán los datos de los eventos del horario para poder modificarlos	
Condiciones de Ejecución: • Que este creada la base de datos • Que los servicios funcionen correctamente • Que exista la interfaz de usuario creada	
Pasos de Ejecución: • Escoger el rol de docente • Ir al menú • Seleccionar horario del docente • Seleccionar y arrastrar el evento a modificar	
Resultado Esperado: Que se puedan modificar los datos de los eventos del horario correctamente.	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_02-HU_29	**Nombre de la Historia**: Modificar eventos del horario
Nombre: Verificar que los datos modificados se hayan guardado correctamente	
Responsable: Sandra Ordoñez	**Fecha:** 07/02/2019
Descripción: Se verificará que los datos modificados se hayan guardado correctamente	
Condiciones de Ejecución: • Que este creada la base de datos • Que los servicios funcionen correctamente • Que exista la interfaz de usuario creada	

Pasos de Ejecución: • Escoger el rol de docente • Ir al menú • Seleccionar horario del docente • Observar si el evento anterior fue reemplazado por el evento actual
Resultado Esperado: Que se haya guardado correctamente el evento modificado
Evaluación de la Prueba: Exitosa

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_01	**Tarea Ingeniería:** TI_01 Crear el método actualizarHorarioDocente en el servicio web del acceso a datos
Nombre: Verificar que el método creado modifique correctamente la fecha y hora del evento	
Responsable: Sandra Ordoñez	**Fecha:** 04/02/2019
Descripción: Se verificará que el método creado modifique correctamente la fecha y hora del evento	
Condiciones de Ejecución: • Conexión a la base de datos • Método actualizarHorarioDocente debe estar creado	
Pasos de Ejecución: • Ejecutar el servicio web del acceso a datos • Insertar datos en el método actualizarHorarioDocente • Click sobre el botón para ejecutar el método actualizarHorarioDocente	
Resultado Esperado: Que se modifiquen correctamente la fecha y hora del evento	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_02	**Tarea Ingeniería:** TI_02 Crear la interfaz de usuario para modificar eventos del horario
Nombre: Verificar que la interfaz de usuario permita arrastrar un evento para modificarlo	
Responsable: Sandra Ordoñez	**Fecha:** 06/02/2019
Descripción: Verificar que la interfaz realizada permita arrastrar un evento para modificarlo	
Condiciones de Ejecución: • Controlador debe estar creado • La vista debe estar creado	
Pasos de Ejecución: • Escoger el rol de docente • Ir al menú	

• Seleccionar horario del docente • Seleccionar y arrastrar el evento a modificar
Resultado Esperado: Que la interfaz permita arrastrar un evento para modificarlo
Evaluación de la Prueba: Exitosa

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_02	**Tarea Ingeniería:** TI_02 Crear la interfaz de usuario para modificar eventos del horario
Nombre: Verificar si la interfaz de usuario correspondiente cumple con el estándar establecido.	
Responsable: Sandra Ordoñez	**Fecha:** 07/02/2019
Descripción: Verificar que la interfaz realizada cumpla con los parámetros del estándar establecido.	
Condiciones de Ejecución: • Controlador debe estar creado. • La vista debe estar creado.	
Pasos de Ejecución: • Escoger el rol de docente • Ir al menú • Dar clic en horario del docente • Verificar interfaz	
Resultado Esperado: La interfaz cumple con el estándar establecido	
Evaluación de la Prueba: Exitosa	

HU-30 Eliminar eventos del horario

Historia de Usuario	
Número: HU_30	**Nombre de la Historia:** Eliminar eventos del horario
Modificación de historia de usuario:	
Usuario: Docente	**Sprint Asignada:** 4
Prioridad en el Negocio: Media	**Puntos Estimados:**30
Riesgo en el Desarrollo: Medio	**Puntos Reales:** 30
Descripción: Como docente quiero poder eliminar eventos del horario.	
Observaciones:	

Historia de Usuario (Reverso) Pruebas de Aceptación
• Verificar que los datos de los eventos del horario se eliminen correctamente • Verificar que al eliminar el evento los cambios se hayan guardado correctamente

TAREA DE INGENIERÍA	
Historia de Usuario: Eliminar eventos del horario	
Número de Tarea: TI_01	**Nombre de Tarea:** Crear el método eliminarHorarioDocente en el servicio web del acceso a datos
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 15
Fecha Inicio: 01/02/2019	**Fecha Fin:** 04/02/2019
Programador Responsable: Sandra Ordoñez	
Descripción: Como desarrollador quiero crear el método eliminarHorarioDocente en el servicio web del Acceso a Datos para poder obtener el evento y poder eliminarlos.	
PRUEBAS DE ACEPTACIÓN	
• Verificar que el método creado elimine correctamente el evento	

TAREA DE INGENIERÍA	
Historia de Usuario: Eliminar eventos del horario	
Número de Tarea: TI_02	**Nombre de Tarea:** T1_02 Crear la interfaz de usuario para eliminar eventos del horario
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 15
Fecha Inicio: 05/02/2019	**Fecha Fin:** 07/02/2019
Programador Responsable: Sandra Ordoñez	
Descripción: Creación de la interfaz de usuario para eliminar los eventos del horario	
PRUEBAS DE ACEPTACIÓN	
• Verificar que la interfaz de usuario muestre la pantalla modal para eliminar el evento • Verificar si la interfaz de usuario correspondiente cumple con el estándar establecido	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-HU_30	**Nombre de la Historia**: Eliminar eventos del horario
Nombre: Verificar que los datos de los eventos del horario se eliminen correctamente	
Responsable: Kevin Chimbo	**Fecha:** 07/02/2019

Descripción: Se obtendrán los datos de los eventos del horario para poder eliminarlos
Condiciones de Ejecución: • Que este creada la base de datos • Que los servicios funcionen correctamente • Que exista la interfaz de usuario creada
Pasos de Ejecución: • Escoger el rol de docente • Ir al menú • Seleccionar horario del docente • Dar click sobre el evento a eliminar • Observar una pantalla modal con el evento para eliminarlo • Eliminar evento
Resultado Esperado: Que se puedan eliminar los datos de los eventos del horario correctamente.
Evaluación de la Prueba: Exitosa

PRUEBA DE ACEPTACIÓN	
Código: PA_02-HU_30	**Nombre de la Historia**: Eliminar eventos del horario
Nombre: Verificar que al eliminar el evento los cambios se hayan guardado correctamente	
Responsable: Kevin Chimbo	**Fecha:** 07/02/2019
Descripción: Se verificará que al eliminar el evento los cambios se hayan guardado correctamente	
Condiciones de Ejecución: • Que este creada la base de datos • Que los servicios funcionen correctamente • Que exista la interfaz de usuario creada	
Pasos de Ejecución: • Escoger el rol de docente • Ir al menú • Seleccionar horario del docente • Observar si el evento ya no existe en el horario del docente	
Resultado Esperado: Que al eliminar el evento los cambios se hayan guardado correctamente	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_01	**Tarea Ingeniería:** TI_01 Crear el método eliminarHorarioDocente en el servicio web del acceso a datos

Nombre: Verificar que el método creado elimine correctamente el evento	
Responsable: Kevin Chimbo	**Fecha:** 04/02/2019
Descripción: Se verificará que el método creado elimine correctamente el evento	
Condiciones de Ejecución • Conexión a la base de datos • Método eliminarHorarioDocente debe estar creado	
Pasos de Ejecución • Ejecutar el servicio web del acceso a datos • Insertar datos en el método eliminarHorarioDocente • Click sobre el botón para ejecutar el método eliminarHorarioDocente	
Resultado Esperado: Que se eliminen los datos de los eventos del horario correctamente	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_02	**Tarea Ingeniería:** TI_02 Crear la interfaz de usuario para eliminar eventos del horario
Nombre: Verificar que la interfaz de usuario muestre la pantalla modal para eliminar el evento	
Responsable: Kevin Chimbo	**Fecha:** 06/02/2019
Descripción: Verificar que la interfaz realizada muestre una pantalla modal para poder eliminar el evento del horario	
Condiciones de Ejecución: • Controlador debe estar creado. • La vista debe estar creado.	
Pasos de Ejecución: • Escoger el rol de docente • Ir al menú • Dar clic sobre el evento a ser eliminado • Observar que se muestre una pantalla modal para poder eliminar el evento	
Resultado Esperado: Que la interfaz cumpla con el estándar establecido	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_02	**Tarea Ingeniería:** TI_02 Crear la interfaz de usuario para eliminar eventos del horario
Nombre: Verificar si la interfaz de usuario correspondiente cumple con el estándar establecido.	

Responsable: Kevin Chimbo	**Fecha:** 07/02/2019
Descripción: Verificar que la interfaz realizada cumpla con los parámetros del estándar establecido.	
Condiciones de Ejecución: • Controlador debe estar creado. • La vista debe estar creado.	
Pasos de Ejecución: • Escoger el rol de docente • Ir al menú • Dar clic en horario del docente • Verificar interfaz	
Resultado Esperado: La interfaz cumple con el estándar establecido	
Evaluación de la Prueba: Exitosa	

HU-31 Cerrar sesión

Historia de Usuario	
Número: HU_31	**Nombre de la Historia:** Cerrar sesión
Modificación de historia de usuario:	
Usuario: Docente	**Sprint Asignada:** 4
Prioridad en el Negocio: Alta	**Puntos Estimados:** 30
Riesgo en el Desarrollo: Alta	**Puntos Reales:** 30
Descripción: Como docente quiero cerrar sesión del sistema de autenticación institucional (CAS).	
Observaciones:	

Historia de Usuario (Reverso) Pruebas de Aceptación
• Verificar que la sesión se destruya.

TAREA DE INGENIERÍA	
Historia de Usuario: Cerrar sesión	
Número de Tarea: TI_01	**Nombre de Tarea:** Crear el método logout en el controlador.

Tipo de Tarea: Desarrollo	**Puntos Estimados:** 30
Fecha Inicio: 08/02/2019	**Fecha Fin:** 14/02/2019
Programador Responsable: Kevin Chimbo	
Descripción: Como desarrollador quiero cerrar la sesión (CAS).	
PRUEBAS DE ACEPTACIÓN	
• Verificar que se redirija la página cerrar sesión de forma correcta.	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-HU_31	**Nombre de la Historia**: Cerrar sesión
Nombre: Verificar que la sesión se destruya.	
Responsable: Sandra Ordoñez	**Fecha:** 14/02/2019
Descripción: Se verificará que se destruya la sesión	
Condiciones de Ejecución: • Que este creada la base de datos • Que el sistema este ejecutado • Que se haya realizado una autenticación previamente	
Pasos de Ejecución: • Autenticarse en el sistema • Dar click en el botón que está ubicado alado de la persona que esta autenticada • Dar click en cerrar sesión	
Resultado Esperado: Que se pueda cerrar la sesión del sistema.	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_01	**Tarea Ingeniería:** Crear el método logout en el controlador.
Nombre: Verificar que se redirija la página cerrar sesión de forma correcta**.**	
Responsable: Sandra Ordoñez	**Fecha:** 14/02/2019
Descripción: Se verificará que se redirija a la página cerrar sesión de forma correcta.	
Condiciones de Ejecución: • Que este creada la base de datos • Que el sistema este ejecutado • Que se haya realizado una autenticación previamente	
Pasos de Ejecución: • Autenticarse en el sistema • Dar click en el botón que está ubicado alado de la persona que esta autenticada • Dar click en cerrar sesión	
Resultado Esperado: Que se muestre la página de cierre de sesión del CAS.	
Evaluación de la Prueba: Exitosa	

HU-32 Listar docentes por facultad

Historia de Usuario	
Número: HU_32	**Nombre de la Historia:** Listar docentes por facultad
Modificación de historia de usuario:	
Usuario: Vicedecano	**Sprint Asignada:** 4
Prioridad en el Negocio: Media	**Puntos Estimados:**30
Riesgo en el Desarrollo: Medio	**Puntos Reales:** 30
Descripción: Como vicedecano quiero visualizar el Listado de los docentes por facultad.	
Observaciones:	

Historia de Usuario (Reverso) Pruebas de Aceptación
• Verificar que se visualice la lista de docentes por facultad • Verificar que se visualice que la lista de docentes esté ordenada según las horas de su horario

TAREA DE INGENIERÍA	
Historia de Usuario: Listar docentes por facultad	
Número de Tarea: TI_01	**Nombre de Tarea:** Crear el método listaDocentesFacultad en el swGestion del acceso a datos
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 15
Fecha Inicio: 08/02/2019	**Fecha Fin:** 11/02/2019
Programador Responsable: Sandra Ordoñez	
Descripción: Crear el método listaDocentesFacultad en el swGestion del acceso a datos para poder obtener los datos correctos.	
PRUEBAS DE ACEPTACIÓN	
• Verificar que se muestren la lista de los docentes por facultad correctamente • Verificar que la codificación se encuentre acorde al estándar de codificación establecido	

TAREA DE INGENIERÍA	
Historia de Usuario: Listar docentes por facultad	
Número de Tarea: TI_02	**Nombre de Tarea:** T1_02 Crear la interfaz

Tipo de Tarea: Desarrollo	**Puntos Estimados:** 15
Fecha Inicio: 12/02/2019	**Fecha Fin:** 14/02/2019
Programador Responsable: Sandra Ordoñez	
Descripción: Crear la interfaz de usuario para poder visualizar la lista de docentes por facultad	
PRUEBAS DE ACEPTACIÓN	
• Verificar que la interfaz tenga los campos necesarios requeridos por la funcionalidad • Verificar si la interfaz de usuario correspondiente cumple con el estándar establecido.	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-HU_32	**Nombre de la Historia**: Listar docentes por facultad
Nombre: Verificar que se visualice la lista de docentes por facultad	
Responsable: Kevin Chimbo	**Fecha:** 14/02/2019
Descripción: Se visualiza el listado de los docentes por facultad	
Condiciones de Ejecución: • Que este creada la base de datos • Que los servicios funcionen correctamente • Que exista la interfaz de usuario creada	
Pasos de Ejecución: • Ingresar al sistema de estafetas • Escoger el rol de vicedecano • Ir al menú • Seleccionar aprobar estafeta • Visualizar la lista de los docentes por facultad	
Resultado Esperado: Se visualiza el listado de los docentes por facultad	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_02-HU_32	**Nombre de la Historia**: Listar docentes por facultad
Nombre: Verificar que se visualice la lista de docentes esté ordenada según las horas de su horario	
Responsable: Kevin Chimbo	**Fecha:** 14/02/2019
Descripción: Se visualiza el listado de los docentes este ordenado según las horas de su horario	
Condiciones de Ejecución: • Que este creada la base de datos • Que los servicios funcionen correctamente • Que exista la interfaz de usuario creada	
Pasos de Ejecución: • Ingresar al sistema de estafetas • Escoger el rol de vicedecano • Ir al menú • Seleccionar aprobar estafeta	

• Visualizar que los docentes estén categorizados según su número de horas en las diferentes etiquetas
Resultado Esperado: Se visualiza el listado de los docentes por facultad de acuerdo al número de hora de su horario
Evaluación de la Prueba: Exitosa

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_01	**Tarea Ingeniería:** TI_01 Crear el método listaDocentesFacultad en el swGestion del acceso a datos
Nombre: Verificar que se muestren la lista de los docentes por facultad correctamente	
Responsable: Kevin Chimbo	**Fecha:** 11/02/2019
Descripción: Se verificará que se muestren la lista de los docentes por facultad correctamente	
Condiciones de Ejecución: • Conexión a la base de datos • Método listaDocentesFacultad debe estar creado	
Pasos de Ejecución: • Crear la función en la base de datos. • Invocar al método listaDocentesFacultad	
Resultado Esperado: Que se muestren los datos de los docentes según la facultad correctamente.	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_02-TI_01	**Tarea Ingeniería:** TI_01 Crear el método listaDocentesFacultad en el swGestion del acceso a datos
Nombre: Verificar que la codificación se encuentre acorde al estándar de codificación establecido	
Responsable: Kevin Chimbo	**Fecha:** 11/02/2019
Descripción: Se verificará que la codificación se encuentre acorde al estándar de codificación establecido	
Condiciones de Ejecución: • Controlador debe estar creado • La vista debe estar creado	
Pasos de Ejecución: • Limpiar el proyecto • Construir el proyecto • Desplegar el proyecto • Verificar si la codificación del listar docentes por facultad está acorde a estándar de codificación	
Resultado Esperado: Que la codificación este acorde con el estándar de codificación	

Evaluación de la Prueba: Exitosa

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_02	**Tarea Ingeniería:** TI_02 Crear la interfaz
Nombre: Verificar que la interfaz tenga los campos necesarios requeridos por la funcionalidad.	
Responsable: Kevin Chimbo	**Fecha:** 13/02/2019
Descripción: Verificar que la interfaz de usuario tenga los campos necesarios que requiere la funcionalidad (Id, nombres, número de horas, reportes)	
Condiciones de Ejecución: • Que ya este creado el acceso a datos • Que ya este creada la clase común	
Pasos de Ejecución: • Loguearse en el CAS • Escoger el rol de vicedecano • Ir al menú • Seleccionar aprobar estafeta • Compara los datos que requiere la funcionalidad con la interfaz de usuario	
Resultado Esperado: Que la interfaz de usuario tenga los campos requeridos por la funcionalidad (Id, nombres, número de horas, reportes)	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_02-TI_02	**Tarea Ingeniería:** TI_02 Crear la interfaz
Nombre: Verificar si la interfaz de usuario correspondiente cumple con el estándar establecido	
Responsable: Kevin Chimbo	**Fecha:** 14/02/2019
Descripción: Verificar que la interfaz realizada cumpla con los parámetros del estándar establecido	
Condiciones de Ejecución • Controlador debe estar creado • La vista debe estar creada	
Pasos de Ejecución: • Loguearse en el CAS • Escoger el rol de vicedecano • Ir al menú • Seleccionar aprobar estafeta • Verificar interfaz	
Resultado Esperado: La interfaz cumple con el estándar establecido	

Evaluación de la Prueba: Exitosa

HU-33 Visualizar horario

Historia de Usuario	
Número: HU_33	**Nombre de la Historia:** Visualizar horario
Modificación de historia de usuario:	
Usuario: Vicedecano	**Sprint Asignada:** 5
Prioridad en el Negocio: Media	**Puntos Estimados:**30
Riesgo en el Desarrollo: Medio	**Puntos Reales:** 30
Descripción: Como vicedecano quiero visualizar el horario de cada docente por facultad	
Observaciones:	

Historia de Usuario (Reverso) Pruebas de Aceptación
• Verificar que se visualice el horario del docente • Verificar que el horario sea del docente seleccionado

TAREA DE INGENIERÍA	
Historia de Usuario: Visualizar horario	
Número de Tarea: TI_01	**Nombre de Tarea:** T1_01 Crear la interfaz
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 30
Fecha Inicio: 15/02/2019	**Fecha Fin:** 21/02/2019
Programador Responsable: Kevin Chimbo	
Descripción: Crear la interfaz de usuario para poder visualizar el horario del docente por facultad	
PRUEBAS DE ACEPTACIÓN	
• Verificar que la interfaz tenga los campos necesarios requeridos por la funcionalidad • Verificar si la interfaz de usuario correspondiente cumple con el estándar establecido	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-HU_33	**Nombre de la Historia**: Visualizar horario
Nombre: Verificar que se visualice el horario del docente	
Responsable: Sandra Ordoñez	**Fecha:** 21/02/2019
Descripción: Se visualiza el horario del docente por facultad	
Condiciones de Ejecución: • Que este creada la base de datos	

• Que los servicios funcionen correctamente • Que este creado el método que devuelva el usuario
Pasos de Ejecución: • Ingresar al sistema de estafetas • Escoger el rol de vicedecano • Ir al menú • Escoger aprobar estafeta • Seleccionar docente • Seleccionar botón para ver horario
Resultado Esperado: Se visualiza el listado de los docentes por facultad
Evaluación de la Prueba: Exitosa

PRUEBA DE ACEPTACIÓN	
Código: PA_02-HU_33	**Nombre de la Historia**: Visualizar horario
Nombre: Verificar que el horario sea del docente seleccionado	
Responsable: Sandra Ordoñez	**Fecha:** 21/02/2019
Descripción: Se verificará que el horario sea del docente seleccionado	
Condiciones de Ejecución: • Que este creada la base de datos • Que los servicios funcionen correctamente • Que este creado el método que devuelva el usuario	
Pasos de Ejecución: • Ingresar al sistema de estafetas • Escoger el rol de vicedecano • Ir al menú • Seleccionar aprobar estafeta • Seleccionar docente • Seleccionar botón para ver horario • Observar el nombre y el horario del docente	
Resultado Esperado: Que el horario sea del docente seleccionado	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_01	**Tarea Ingeniería:** TI_01 Crear la interfaz
Nombre: Verificar que la interfaz tenga los campos necesarios requeridos por la funcionalidad.	
Responsable: Sandra Ordoñez	**Fecha:** 18/02/2019
Descripción: Verificar que la interfaz de usuario tenga los campos necesarios que requiere la funcionalidad (Días, horas, eventos)	
Condiciones de Ejecución:	

• Que ya este creado el acceso a datos • Que ya este creada la clase común
Pasos de Ejecución: • Loguearse en el CAS • Escoger el rol de vicedecano • Ir al menú • Seleccionar aprobar estafeta • Compara los datos que requiere la funcionalidad con la interfaz de usuario
Resultado Esperado: Que la interfaz de usuario tenga los campos requeridos por la funcionalidad (Días, horas, eventos)
Evaluación de la Prueba: Exitosa

PRUEBA DE ACEPTACIÓN	
Código: PA_02-TI_01	**Tarea Ingeniería:** TI_01 Crear la interfaz
Nombre: Verificar si la interfaz de usuario correspondiente cumple con el estándar establecido	
Responsable: Sandra Ordoñez	**Fecha:** 21/02/2019
Descripción: Verificar que la interfaz realizada cumpla con los parámetros del estándar establecido	
Condiciones de Ejecución: • Controlador debe estar creado • La vista debe estar creada	
Pasos de Ejecución: • Loguearse en el CAS • Escoger el rol de vicedecano • Ir al menú • Seleccionar aprobar estafeta • Verificar interfaz	
Resultado Esperado: La interfaz cumple con el estándar establecido	
Evaluación de la Prueba: Exitosa	

HU-34 Visualizar resumen de horas

Historia de Usuario	
Número: HU_34	**Nombre de la Historia:** Visualizar resumen de horas
Modificación de historia de usuario:	
Usuario: Vicedecano	**Sprint Asignada:** 5
Prioridad en el Negocio: Media	**Puntos Estimados:**30

Riesgo en el Desarrollo: Medio	**Puntos Reales:** 30
Descripción: Como vicedecano quiero visualizar el resumen de horas	
Observaciones:	

Historia de Usuario (Reverso) Pruebas de Aceptación
• Verificar que se visualice el resumen de horas • Verificar que el resumen de horas pertenezca al docente

TAREA DE INGENIERÍA	
Historia de Usuario: Visualizar resumen de horas	
Número de Tarea: TI_01	**Nombre de Tarea:** T1_01 Crear la interfaz
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 30
Fecha Inicio: 15/02/2019	**Fecha Fin:** 21/02/2019
Programador Responsable: Sandra Ordoñez	
Descripción: Crear la interfaz de usuario para poder visualizar el resumen de horas	
PRUEBAS DE ACEPTACIÓN	
• Verificar que la interfaz tenga los campos necesarios requeridos por la funcionalidad • Verificar si la interfaz de usuario correspondiente cumple con el estándar establecido	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-HU_34	**Nombre de la Historia**: Visualizar resumen de horas
Nombre: Verificar que se visualice el resumen de horas	
Responsable: Kevin Chimbo	**Fecha:** 21/02/2019
Descripción: Se visualiza el resumen de horas	
Condiciones de Ejecución: • Que este creada la base de datos • Que los servicios funcionen correctamente • Que este creado el método que devuelva el usuario	
Pasos de Ejecución: • Ingresar al sistema de estafetas • Escoger el rol de vicedecano • Ir al menú • Escoger aprobar estafeta • Seleccionar docente • Seleccionar botón para ver el resumen de horas	
Resultado Esperado: Se visualiza el listado de los docentes por facultad	

Evaluación de la Prueba: Exitosa

PRUEBA DE ACEPTACIÓN	
Código: PA_02-HU_34	**Nombre de la Historia**: Visualizar resumen de horas
Nombre: Verificar que el resumen de horas pertenezca al docente	
Responsable: Kevin Chimbo	**Fecha:** 21/02/2019
Descripción: Se verificará que el resumen de horas pertenezca al docente	
Condiciones de Ejecución: • Que este creada la base de datos • Que los servicios funcionen correctamente • Que este creado el método que devuelva el usuario	
Pasos de Ejecución: • Ingresar al sistema de estafetas • Escoger el rol de vicedecano • Ir al menú • Seleccionar aprobar estafeta • Seleccionar docente • Seleccionar botón para ver el resumen de horas • Observar el nombre y el resumen de horas del docente	
Resultado Esperado: Que el resumen de horas pertenezca al docente	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_01	**Tarea Ingeniería:** TI_01 Crear la interfaz
Nombre: Verificar que la interfaz tenga los campos necesarios requeridos por la funcionalidad	
Responsable: Kevin Chimbo	**Fecha:** 18/02/2019
Descripción: Verificar que la interfaz de usuario tenga los campos necesarios que requiere la funcionalidad (Id, ítem, número de horas por ítem, número de horas totales)	
Condiciones de Ejecución: • Que ya este creado el acceso a datos • Que ya este creada la clase común	
Pasos de Ejecución: • Loguearse en el CAS • Escoger el rol de vicedecano • Ir al menú • Seleccionar aprobar estafeta • Compara los datos que requiere la funcionalidad con la interfaz de usuario	

Resultado Esperado: Que la interfaz de usuario tenga los campos requeridos por la funcionalidad (Id, ítem, número de horas por ítem, número de horas totales)
Evaluación de la Prueba: Exitosa

PRUEBA DE ACEPTACIÓN	
Código: PA_02-TI_01	**Tarea Ingeniería:** TI_01 Crear la interfaz
Nombre: Verificar si la interfaz de usuario correspondiente cumple con el estándar establecido	
Responsable: Kevin Chimbo	**Fecha:** 21/02/2019
Descripción: Verificar que la interfaz realizada cumpla con los parámetros del estándar establecido	
Condiciones de Ejecución: • Controlador debe estar creado • La vista debe estar creada	
Pasos de Ejecución: • Loguearse en el CAS • Escoger el rol de vicedecano • Ir al menú • Seleccionar aprobar estafeta • Verificar interfaz	
Resultado Esperado: La interfaz cumple con el estándar establecido	
Evaluación de la Prueba: Exitosa	

HU-35 Desaprobar estafeta

Historia de Usuario	
Número: HU_35	**Nombre de la Historia:** Desaprobar estafeta
Modificación de historia de usuario:	
Usuario: Vicedecano	**Sprint Asignada:** 5
Prioridad en el Negocio: Media	**Puntos Estimados:** 30
Riesgo en el Desarrollo: Medio	**Puntos Reales:** 30
Descripción: Como vicedecano quiero poder desaprobar la estafeta	
Observaciones:	

Historia de Usuario (Reverso) Pruebas de Aceptación
• Verificar que la estafeta cambie a estado no aprobado • Verificar que se haya guardado correctamente el cambio de estado de la estafeta

TAREA DE INGENIERÍA	
Historia de Usuario: Desaprobar estafeta	
Número de Tarea: TI_01	**Nombre de Tarea:** Crear el acceso a datos
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 15
Fecha Inicio: 22/02/2019	**Fecha Fin:** 25/02/2019
Programador Responsable: Kevin Chimbo	
Descripción: Como desarrollador quiero crear el acceso a datos	
PRUEBAS DE ACEPTACIÓN	
• Verificar que el método desaprobacionEstafeta del acceso a datos modifique el estado de la estafeta • Verificar que el método cumpla con el estándar de codificación establecido	

TAREA DE INGENIERÍA	
Historia de Usuario: Desaprobar estafeta	
Número de Tarea: TI_02	**Nombre de la Tarea:** Crear el botón en la interfaz de usuario
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 15
Fecha Inicio: 26/02/2019	**Fecha Fin:** 28/02/2019
Programador Responsable: Kevin Chimbo	
Descripción: Como desarrollador necesito crear la interfaz de usuario para la desaprobación de la estafeta del docente	
PRUEBAS DE ACEPTACIÓN	
• Verificar si la interfaz de usuario correspondiente cumple con el estándar establecido • Verificar que el botón realice la acción desaprobar estafeta correctamente	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-HU_35	**Nombre de la Historia**: Desaprobar estafeta
Nombre: Verificar que la estafeta cambie a estado no aprobado	
Responsable: Sandra Ordoñez	**Fecha:** 28/02/2019

Descripción: Verificar que la estafeta cambie su estado a no aprobado
Condiciones de Ejecución: • Que ya este creado el acceso a datos y la interfaz de usuario.
Pasos de Ejecución: • Ingresar al sistema de estafetas • Loguearse en el CAS • Seleccionar el rol de vicedecano • Escoger la opción Aprobar estafeta • Seleccionar el docente • Click en el botón desaprobar • Observar en la interfaz en la fila del docente seleccionado se muestra el estado en no aprobado
Resultado Esperado: Que la estafeta cambie su estado a no aprobado
Evaluación de la Prueba: Exitosa

PRUEBA DE ACEPTACIÓN	
Código: PA_01-HU_35	**Nombre de la Historia**: Desaprobar estafeta
Nombre: Verificar que se haya guardado correctamente el cambio de estado de la estafeta	
Responsable: Sandra Ordoñez	**Fecha:** 28/02/2019
Descripción: Verificar que la estafeta se haya guardado correctamente con el estado no aprobado	
Condiciones de Ejecución: • Que ya este creado el acceso a datos y la interfaz de usuario	
Pasos de Ejecución: • Ingresar al sistema de estafetas • Loguearse en el CAS • Seleccionar el rol de vicedecano • Escoger la opción Aprobar estafeta • Seleccionar el docente • Click en el botón desaprobar • Observar en la interfaz en la fila del docente seleccionado se muestra el estado en no aprobado	
Resultado Esperado: Que la estafeta cambie su estado a no aprobado	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_01	**Tarea Ingeniería:** TI_01 Crear el acceso a datos

Nombre: Verificar que el método desaprobacionEstafeta del acceso a datos modifique el estado de la estafeta	
Responsable: Sandra Ordoñez	**Fecha:** 22/02/2019
Descripción: Se verificará que el método desaprobacionEstafeta del acceso a datos modifique el estado de la estafeta	
Condiciones de Ejecución: • Que ya este creado el acceso a datos	
Pasos de Ejecución: • Revisar que las clases existan en la capa de acceso • Revisar que se tenga todas las clases y cada una de ellas se encuentra heredadas de las clases de clase común	
Resultado Esperado: Que el método desaprobacionEstafeta modifique correctamente el estado de la estafeta	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_02-TI_01	**Tarea Ingeniería:** TI_01 Crear el acceso a datos
Nombre: Verificar que el método cumpla con el estándar de codificación establecido	
Responsable: Sandra Ordoñez	**Fecha:** 25/02/2019
Descripción: Se verificará que la codificación se encuentre acorde al estándar de codificación establecido	
Condiciones de Ejecución: • Controlador debe estar creado • La vista debe estar creado	
Pasos de Ejecución: • Limpiar el proyecto • Construir el proyecto • Desplegar el proyecto • Verificar si la codificación del método desaprobacionEstafeta esté acorde a estándar de codificación	
Resultado Esperado: Que la codificación este acorde con el estándar de codificación	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_02	**Tarea Ingeniería:** TI_02 Crear el botón en la interfaz

Nombre: Verificar si la interfaz de usuario correspondiente cumple con el estándar establecido	
Responsable: Sandra Ordoñez	**Fecha:** 27/02/2019
Descripción: Verificar que la interfaz realizada cumpla con los parámetros del estándar establecido	
Condiciones de Ejecución: • Controlador debe estar creado • La vista debe estar creado	
Pasos de Ejecución: • Escoger el rol de vicedecano • Ir al menú • Dar clic en desaprobación del horario del docente • Verificar interfaz	
Resultado Esperado: La interfaz cumple con el estándar establecido	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_02-TI_02	**Tarea Ingeniería:** TI_02 Crear el botón en la interfaz
Nombre: Verificar que el botón realice la acción desaprobar estafeta correctamente	
Responsable: Sandra Ordoñez	**Fecha:** 28/02/2019
Descripción: Verificar que el botón desapruebe la estafeta del docente correctamente	
Condiciones de Ejecución: • Que este creado el botón en la interfaz de usuario	
Pasos de Ejecución: • Loguearse en el CAS • Seleccionar el rol de vicedecano • Ir al menú • Escoger aprobar la estafeta • Seleccionar un docente con la estafeta aprobada • Click en el botón desaprobar • Observar que el archivo js emita una alerta donde la estafeta este desaprobada	
Resultado Esperado: Que el botón desapruebe la estafeta correctamente	
Evaluación de la Prueba: Exitosa	

HU-36 Autenticar con CAS

Historia de Usuario	
Número: HU_36	**Nombre de la Historia:** Autenticar con CAS
Modificación de historia de usuario:	
Usuario: Docente	**Sprint Asignada:** 5
Prioridad en el Negocio: Alta	**Puntos Estimados:**30
Riesgo en el Desarrollo: Alta	**Puntos Reales:** 30
Descripción: Como docente quiero ingresar al sistema mediante el sistema de autenticación institucional (CAS).	
Observaciones:	

Historia de Usuario (Reverso) Pruebas de Aceptación
• Verificar que después de autenticarse en el CAS se ingrese al sistema de estafetas correctamente.

TAREA DE INGENIERÍA	
Historia de Usuario: Autenticar con CAS	
Número de Tarea: TI_01	**Nombre de Tarea:** Modificar el archivo web.xml
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 30
Fecha Inicio: 22/02/2019	**Fecha Fin:** 28/02/2019
Programador Responsable: Sandra Ordoñez	
Descripción: Se modificará el archivo web.xml para añadir las direcciones del CAS.	
PRUEBAS DE ACEPTACIÓN	
• Verificar que al abrir el proyecto se redirija a la página de autenticación del CAS.	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-HU_36	**Nombre de la Historia**: Autenticar con CAS

Nombre: Verificar que después de autenticarse en el CAS se ingrese al sistema de estafetas correctamente.	
Responsable: Kevin Chimbo	**Fecha:** 28/02/2019
Descripción: Se verificará que después de haberse autentificado en el CAS se ingrese al sistema de estafetas.	
Condiciones de Ejecución: • Que este creada la base de datos • Que se modifique el filtro con la dirección del CAS	
Pasos de Ejecución: • Desplegar la aplicación • Ejecutar la aplicación • Ingresar el usuario y contraseña	
Resultado Esperado: Que se pueda ingresar al sistema de estafetas por medio del sistema de autenticación institucional CAS	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01_TI_01	**Historia Técnica:** TI_01 Modificar el filtro para el CAS y obtener Token de usuario.
Nombre: Verificar que el usuario se pueda autentificarse correctamente.	
Responsable: Kevin Chimbo	**Fecha:** 28/02/2019
Descripción: Se verificará que el usuario pueda autentificarse correctamente para que pueda manipular el sistema.	
Condiciones de Ejecución: • Encender el entorno de desarrollo NetBeans	
Pasos de ejecución: • Desplegar el sistema • Ejecutar el sistema • Ingresar usuario y contraseña • Verificar que el usuario se autentique en el sistema	
Resultado esperado: Que se pueda modificar el filtro para el CAS y obtener el Token de usuario.	
Evaluación de la prueba: Exitosa	

HT-09 Cambiar preparet statements del acceso a datos

HISTORIA TÉCNICA DEL SISTEMA	
Número: HT_09	**Nombre de la Historia:** Cambiar preparet statements del acceso a datos
Modificación de historia de usuario:	
Usuario: Desarrollador	**Sprint Asignada:** 6
Prioridad en el Negocio: Alta	**Puntos Estimados:**30
Riesgo en el Desarrollo: Alta	**Puntos Reales:** 30
Descripción: Como desarrollador quiero cambiar el preparet statements del acceso a datos	
Observaciones: La base de datos esta realizadas con funciones y procedimientos almacenados.	

Historia de Usuario (Reverso) Pruebas de Aceptación
• Verificar si el cambio realizado obtiene correctamente la información de la base de datos

TAREA DE INGENIERÍA	
Historia Técnica: HT_09 Cambiar preparet statements del acceso a datos	
Número de Tarea: TI_01	**Nombre de Tarea:** Modificar el acceso a datos para cambiar los statements a preparet statements
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 30
Fecha Inicio: 01/03/2019	**Fecha Fin:** 11/03/2019
Programador Responsable: Kevin Chimbo	
Descripción: Se modificará el acceso a datos para cambiar el preparet statements del acceso a datos.	
Pruebas de Aceptación • Verificar que los cambios se hayan hecho de acuerdo al estándar de codificación	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-HT_09	**Nombre de la Historia:** Cambiar preparet statements del acceso a datos
Nombre: Verificar si el cambio realizado obtiene correctamente la información de la base de datos	
Responsable: Sandra Ordoñez	**Fecha:** 11/03/2019
Descripción: Se verificará si el cambio que se realizó obtiene correctamente la información de la base de datos	
Condiciones de Ejecución: • Que este creado el acceso a datos	
Pasos de Ejecución: • Desplegar la aplicación • Ejecutar el acceso a datos • Ejecutar un método del acceso a datos • Verificar la información obtenida con la que se encuentra en la base a datos	
Resultado Esperado: Que el cambio realizado obtenga correctamente la información de la base de datos	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01_TI_01	**Historia Técnica:** TI_01 Modificar el acceso a datos para cambiar los statements a preparet statements
Nombre: Verificar que los cambios se hayan hecho de acuerdo al estándar de codificación	
Responsable: Sandra Ordoñez	**Fecha:** 11/03/2019
Descripción: Se verificará que los cambios realizados se hayan hecho de acuerdo al estándar de codificación	
Condiciones de Ejecución: • Que este creado el acceso a datos	
Pasos de ejecución: • Limpiar y construir • Desplegar el acceso a datos • Realizar peticiones a la base de datos • Observar si existen datos	
Resultado esperado: Que los cambios realizados estén realizados de acuerdo al estándar de codificación	
Evaluación de la prueba: Exitosa	

HU-08 Insertar tipo designación

Historia de Usuario	
Número: HU_08	**Nombre:** Insertar tipo designación
Modificación de metáfora del sistema:	
Usuario: Vicedecano	**Sprint Asignada:** 6
Prioridad en Negocio: Medio	**Puntos Estimados:** 30
Riesgo en desarrollo: Medio	**Puntos Reales:** 30
Descripción: Se desarrollará para dar mantenimiento a los campos de tipo designación	
Observaciones:	
(Reverso) Pruebas de Aceptación • Verificar que el usuario no ingrese caracteres especiales en los campos • Verificar la conexión a la base de datos	

TAREA DE INGENIERÍA	
Historia de Usuario: Insertar tipo designación	
Número de Tarea: TI_01	**Nombre de Tarea:** T1_01 Implementar el método soloLetras para validar el js
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 30
Fecha Inicio: 01/03/2019	**Fecha Fin:** 11/03/2019
Programador Responsable: Sandra Ordoñez	
Descripción: Implementar el método soloLetras para validar el js	
PRUEBAS DE ACEPTACIÓN	
• Verificar que el método acepte vocales con tilde y espacios • Verificar que el método se implemente correctamente en la interfaz de usuario	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-HU_08	**Nombre de la Historia**: Insertar tipo designación
Nombre: Verificar que el usuario no ingrese caracteres especiales en los campos	
Responsable: Kevin Chimbo	**Fecha:** 11/03/2019
Descripción: Se verificará que el usuario no ingrese caracteres especiales en los campos	
Condiciones de Ejecución: • Que este creada la base de datos	

• Que los servicios funcionen correctamente • Que este creado el método que devuelva el usuario
Pasos de Ejecución: • Ingresar al sistema de estafetas • Escoger el rol de vicedecano • Ir al menú • Escoger tipo designación • Verificar que no se ingresen caracteres especiales
Resultado Esperado: Que el usuario no ingrese caracteres especiales en los campos
Evaluación de la Prueba: Exitosa

PRUEBA DE ACEPTACIÓN	
Código: PA_02-HU_08	**Nombre de la Historia**: Insertar tipo designación
Nombre: Verificar la conexión a la base de datos	
Responsable: Kevin Chimbo	**Fecha:** 11/03/2019
Descripción: Se verificará la conexión a la base de datos	
Condiciones de Ejecución: • Tener la codificación realizada para la conexión a la base de datos	
Condiciones de Ejecución: • Limpiar y construir el proyecto. • Cargar el proyecto. • Verificar si la codificación implementada permite la conexión a la base de datos	
Resultado Esperado: Que el usuario no ingrese caracteres especiales en los campos	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_01	**Tarea Ingeniería:** TI_01 Implementar el método soloLetras para validar el js
Nombre: Verificar que el método acepte vocales con tilde y espacios	
Responsable: Kevin Chimbo	**Fecha:** 06/03/2019
Descripción: Verificar que el método acepte vocales con tilde y espacios	
Condiciones de Ejecución: • Que ya este creado el acceso a datos • Que ya este creada la clase común	
Pasos de Ejecución: • Loguearse en el CAS • Escoger el rol de vicedecano	

• Ir al menú • Seleccionar tipo designación • Insertar datos • Verificar que se cree un nuevo tipo designación
Resultado Esperado: Que el método acepte vocales con tilde y espacios
Evaluación de la Prueba: Exitosa

PRUEBA DE ACEPTACIÓN	
Código: PA_02-TI_01	**Tarea Ingeniería:** TI_01 Implementar el método soloLetras para validar el js
Nombre: Verificar que el método se implemente correctamente en la interfaz de usuario	
Responsable: Kevin Chimbo	**Fecha:** 11/03/2019
Descripción: Verificar que el método se implemente correctamente en la interfaz de usuario	
Condiciones de Ejecución: • Controlador debe estar creado • La vista debe estar creada	
Pasos de Ejecución: • Loguearse en el CAS • Escoger el rol de vicedecano • Ir al menú • Seleccionar tipo designación • Verificar que no se ingrese caracteres especiales	
Resultado Esperado: Que el método se implemente correctamente en la interfaz de usuario	
Evaluación de la Prueba: Exitosa	

HU-04 Insertar tipo función

Historia de Usuario	
Número: HU_04	**Nombre:** Insertar tipo función
Modificación de metáfora del sistema:	
Usuario: Vicedecano	**Sprint Asignada:** 6
Prioridad en Negocio: Medio	**Puntos Estimados:** 30
Riesgo en desarrollo: Medio	**Puntos Reales:** 30
Descripción: Se desarrollará para dar mantenimiento a los campos de tipo función	
Observaciones:	
(Reverso) Pruebas de Aceptación	

• Verificar que el usuario no ingrese caracteres especiales en los campos • Verificar la conexión a la base de datos

TAREA DE INGENIERÍA	
Historia de Usuario: Insertar tipo función	
Número de Tarea: TI_01	**Nombre de Tarea:** T1_01 Implementar el método soloLetras para validar el js
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 30
Fecha Inicio: 12/03/2019	**Fecha Fin:** 18/03/2019
Programador Responsable: Kevin Chimbo	
Descripción: Implementar el método soloLetras para validar el js	
PRUEBAS DE ACEPTACIÓN	
• Verificar que el método acepte vocales con tilde y espacios • Verificar que el método se implemente correctamente en la interfaz de usuario	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-HU_04	**Nombre de la Historia**: Insertar tipo función
Nombre: Verificar que el usuario no ingrese caracteres especiales en los campos	
Responsable: Sandra Ordoñez	**Fecha:** 18/03/2019
Descripción: Se verificará que el usuario no ingrese caracteres especiales en los campos	
Condiciones de Ejecución: • Que este creada la base de datos • Que los servicios funcionen correctamente • Que este creado el método que devuelva el usuario	
Pasos de Ejecución: • Ingresar al sistema de estafetas • Escoger el rol de vicedecano • Ir al menú • Escoger tipo función • Verificar que no se ingresen caracteres especiales	
Resultado Esperado: Que el usuario no ingrese caracteres especiales en los campos	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_02-HU_04	**Nombre de la Historia**: Insertar tipo función
Nombre: Verificar la conexión a la base de datos	
Responsable: Sandra Ordoñez	**Fecha:** 18/03/2019
Descripción: Se verificará la conexión a la base de datos	
Condiciones de Ejecución: • Tener la codificación realizada para la conexión a la base de datos	
Condiciones de Ejecución: • Limpiar y construir el proyecto. • Cargar el proyecto. • Verificar si la codificación implementada permite la conexión a la base de datos	
Resultado Esperado: Que el usuario no ingrese caracteres especiales en los campos	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_01	**Tarea Ingeniería:** TI_01 Implementar el método soloLetras para validar el js
Nombre: Verificar que el método acepte vocales con tilde y espacios	
Responsable: Sandra Ordoñez	**Fecha:** 14/03/2019
Descripción: Verificar que el método acepte vocales con tilde y espacios	
Condiciones de Ejecución: • Que ya este creado el acceso a datos • Que ya este creada la clase común	
Pasos de Ejecución: • Loguearse en el CAS • Escoger el rol de vicedecano • Ir al menú • Seleccionar tipo función • Insertar datos • Verificar que se cree un nuevo tipo función	
Resultado Esperado: Que el método acepte vocales con tilde y espacios	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_02-TI_01	**Tarea Ingeniería:** TI_01 Implementar el método soloLetras para validar el js
Nombre: Verificar que el método se implemente correctamente en la interfaz de usuario	
Responsable: Sandra Ordoñez	**Fecha:** 18/03/2019
Descripción: Verificar que el método se implemente correctamente en la interfaz de usuario	
Condiciones de Ejecución: • Controlador debe estar creado • La vista debe estar creada	
Pasos de Ejecución: • Loguearse en el CAS • Escoger el rol de vicedecano • Ir al menú • Seleccionar tipo función • Verificar que no se ingrese caracteres especiales	
Resultado Esperado: Que el método se implemente correctamente en la interfaz de usuario	
Evaluación de la Prueba: Exitosa	

HU-10 Insertar función

Historia de Usuario	
Número: HU_10	**Nombre:** Insertar función
Modificación de metáfora del sistema:	
Usuario: Vicedecano	**Sprint Asignada:** 6
Prioridad en Negocio: Medio	**Puntos Estimados:** 30
Riesgo en desarrollo: Medio	**Puntos Reales:** 30
Descripción: Se desarrollará para dar mantenimiento a los campos de función	
Observaciones:	
(Reverso) Pruebas de Aceptación • Verificar que el usuario no ingrese caracteres especiales en los campos • Verificar la conexión a la base de datos	

TAREA DE INGENIERÍA	
Historia de Usuario: Insertar función	
Número de Tarea: TI_01	**Nombre de Tarea:** T1_01 Implementar el método soloLetras para validar el js
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 30
Fecha Inicio: 12/03/2019	**Fecha Fin:** 18/03/2019
Programador Responsable: Sandra Ordoñez	
Descripción: Implementar el método soloLetras para validar el js	
PRUEBAS DE ACEPTACIÓN	
• Verificar que el método acepte vocales con tilde y espacios • Verificar que el método se implemente correctamente en la interfaz de usuario	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-HU_10	**Nombre de la Historia**: Insertar función
Nombre: Verificar que el usuario no ingrese caracteres especiales en los campos	
Responsable: Kevin Chimbo	**Fecha:** 18/03/2019
Descripción: Se verificará que el usuario no ingrese caracteres especiales en los campos	
Condiciones de Ejecución: • Que este creada la base de datos • Que los servicios funcionen correctamente • Que este creado el método que devuelva el usuario	
Pasos de Ejecución: • Ingresar al sistema de estafetas • Escoger el rol de vicedecano • Ir al menú • Escoger función • Verificar que no se ingresen caracteres especiales	
Resultado Esperado: Que el usuario no ingrese caracteres especiales en los campos	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_02-HU_10	**Nombre de la Historia**: Insertar función
Nombre: Verificar la conexión a la base de datos	
Responsable: Kevin Chimbo	**Fecha:** 18/03/2019
Descripción: Se verificará la conexión a la base de datos	
Condiciones de Ejecución: • Tener la codificación realizada para la conexión a la base de datos	
Condiciones de Ejecución: • Limpiar y construir el proyecto.	

• Cargar el proyecto. • Verificar si la codificación implementada permite la conexión a la base de datos
Resultado Esperado: Que el usuario no ingrese caracteres especiales en los campos
Evaluación de la Prueba: Exitosa

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_01	**Tarea Ingeniería:** TI_01 Implementar el método soloLetras para validar el js
Nombre: Verificar que el método acepte vocales con tilde y espacios	
Responsable: Kevin Chimbo	**Fecha:** 14/03/2019
Descripción: Verificar que el método acepte vocales con tilde y espacios	
Condiciones de Ejecución: • Que ya este creado el acceso a datos • Que ya este creada la clase común	
Pasos de Ejecución: • Loguearse en el CAS • Escoger el rol de vicedecano • Ir al menú • Seleccionar función • Insertar datos • Verificar que se cree una nueva función	
Resultado Esperado: Que el método acepte vocales con tilde y espacios	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_02-TI_01	**Tarea Ingeniería:** TI_01 Implementar el método soloLetras para validar el js
Nombre: Verificar que el método se implemente correctamente en la interfaz de usuario	
Responsable: Kevin Chimbo	**Fecha:** 18/03/2019
Descripción: Verificar que el método se implemente correctamente en la interfaz de usuario	
Condiciones de Ejecución: • Controlador debe estar creado • La vista debe estar creada	
Pasos de Ejecución: • Loguearse en el CAS • Escoger el rol de vicedecano • Ir al menú • Seleccionar función	

• Verificar que no se ingrese caracteres especiales
Resultado Esperado: Que el método se implemente correctamente en la interfaz de usuario
Evaluación de la Prueba: Exitosa

HU-10 Insertar designación

Historia de Usuario	
Número: HU_10	**Nombre:** Insertar designación
Modificación de metáfora del sistema:	
Usuario: Vicedecano	**Sprint Asignada:** 7
Prioridad en Negocio: Medio	**Puntos Estimados:** 30
Riesgo en desarrollo: Medio	**Puntos Reales:** 30
Descripción: Se desarrollará para dar mantenimiento a los campos de designación	
Observaciones:	
(Reverso) Pruebas de Aceptación • Verificar que el usuario no ingrese caracteres especiales en los campos • Verificar la conexión a la base de datos	

TAREA DE INGENIERÍA	
Historia de Usuario: Insertar designación	
Número de Tarea: TI_01	**Nombre de Tarea:** T1_01 Implementar el método soloLetras para validar el js
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 30
Fecha Inicio: 19/03/2019	**Fecha Fin:** 25/03/2019
Programador Responsable: Kevin Chimbo	
Descripción: Implementar el método soloLetras para validar el js	
PRUEBAS DE ACEPTACIÓN	
• Verificar que el método acepte vocales con tilde y espacios • Verificar que el método se implemente correctamente en la interfaz de usuario	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-HU_10	**Nombre de la Historia**: Insertar designación
Nombre: Verificar que el usuario no ingrese caracteres especiales en los campos	
Responsable: Sandra Ordoñez	**Fecha:** 25/03/2019
Descripción: Se verificará que el usuario no ingrese caracteres especiales en los campos	
Condiciones de Ejecución: • Que este creada la base de datos • Que los servicios funcionen correctamente • Que este creado el método que devuelva el usuario	
Pasos de Ejecución: • Ingresar al sistema de estafetas • Escoger el rol de vicedecano • Ir al menú • Escoger designación • Verificar que no se ingresen caracteres especiales	
Resultado Esperado: Que el usuario no ingrese caracteres especiales en los campos	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_02-HU_10	**Nombre de la Historia**: Insertar designación
Nombre: Verificar la conexión a la base de datos	
Responsable: Sandra Ordoñez	**Fecha:** 25/03/2019
Descripción: Se verificará la conexión a la base de datos	
Condiciones de Ejecución: • Tener la codificación realizada para la conexión a la base de datos	
Condiciones de Ejecución: • Limpiar y construir el proyecto • Cargar el proyecto • Verificar si la codificación implementada permite la conexión a la base de datos	
Resultado Esperado: Que el usuario no ingrese caracteres especiales en los campos	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_01	**Tarea Ingeniería:** TI_01 Implementar el método soloLetras para validar el js
Nombre: Verificar que el método acepte vocales con tilde y espacios	
Responsable: Sandra Ordoñez	**Fecha:** 21/03/2019
Descripción: Verificar que el método acepte vocales con tilde y espacios	

Condiciones de Ejecución: • Que ya este creado el acceso a datos • Que ya este creada la clase común
Pasos de Ejecución: • Loguearse en el CAS • Escoger el rol de vicedecano • Ir al menú • Seleccionar designación • Insertar datos • Verificar que se cree una nueva designación
Resultado Esperado: Que el método acepte vocales con tilde y espacios
Evaluación de la Prueba: Exitosa

PRUEBA DE ACEPTACIÓN	
Código: PA_02-TI_01	**Tarea Ingeniería:** TI_01 Implementar el método soloLetras para validar el js
Nombre: Verificar que el método se implemente correctamente en la interfaz de usuario	
Responsable: Sandra Ordoñez	**Fecha:** 25/03/2019
Descripción: Verificar que el método se implemente correctamente en la interfaz de usuario	
Condiciones de Ejecución: • Controlador debe estar creado • La vista debe estar creada	
Pasos de Ejecución: • Loguearse en el CAS • Escoger el rol de vicedecano • Ir al menú • Seleccionar designación • Verificar que no se ingrese caracteres especiales	
Resultado Esperado: Que el método se implemente correctamente en la interfaz de usuario	
Evaluación de la Prueba: Exitosa	

HU-05 Insertar secciones

Historia de Usuario	
Número: HU_05	**Nombre:** Insertar secciones
Modificación de metáfora del sistema:	
Usuario: Vicedecano	**Sprint Asignada:** 7
Prioridad en Negocio: Medio	**Puntos Estimados:** 30
Riesgo en desarrollo: Medio	**Puntos Reales:** 30
Descripción: Se desarrollará para dar mantenimiento a los campos de secciones	
Observaciones:	
(Reverso) Pruebas de Aceptación • Verificar que el usuario no ingrese caracteres especiales en los campos • Verificar la conexión a la base de datos	

TAREA DE INGENIERÍA	
Historia de Usuario: Insertar secciones	
Número de Tarea: TI_01	**Nombre de Tarea:** T1_01 Implementar el método soloLetras para validar el js
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 30
Fecha Inicio: 19/03/2019	**Fecha Fin:** 25/03/2019
Programador Responsable: Sandra Ordoñez	
Descripción: Implementar el método soloLetras para validar el js	
PRUEBAS DE ACEPTACIÓN	
• Verificar que el método acepte vocales con tilde y espacios • Verificar que el método se implemente correctamente en la interfaz de usuario	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-HU_05	**Nombre de la Historia**: Insertar secciones
Nombre: Verificar que el usuario no ingrese caracteres especiales en los campos	
Responsable: Kevin Chimbo	**Fecha:** 25/03/2019
Descripción: Se verificará que el usuario no ingrese caracteres especiales en los campos	
Condiciones de Ejecución: • Que este creada la base de datos • Que los servicios funcionen correctamente • Que este creado el método que devuelva el usuario	
Pasos de Ejecución: • Ingresar al sistema de estafetas • Escoger el rol de vicedecano • Ir al menú	

• Escoger secciones • Verificar que no se ingresen caracteres especiales
Resultado Esperado: Que el usuario no ingrese caracteres especiales en los campos
Evaluación de la Prueba: Exitosa

PRUEBA DE ACEPTACIÓN	
Código: PA_02-HU_05	**Nombre de la Historia**: Insertar secciones
Nombre: Verificar la conexión a la base de datos	
Responsable: Kevin Chimbo	**Fecha:** 25/03/2019
Descripción: Se verificará la conexión a la base de datos	
Condiciones de Ejecución: • Tener la codificación realizada para la conexión a la base de datos	
Condiciones de Ejecución: • Limpiar y construir el proyecto • Cargar el proyecto • Verificar si la codificación implementada permite la conexión a la base de datos	
Resultado Esperado: Que el usuario no ingrese caracteres especiales en los campos	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_01	**Tarea Ingeniería:** TI_01 Implementar el método soloLetras para validar el js
Nombre: Verificar que el método acepte vocales con tilde y espacios	
Responsable: Kevin Chimbo	**Fecha:** 21/03/2019
Descripción: Verificar que el método acepte vocales con tilde y espacios	
Condiciones de Ejecución: • Que ya este creado el acceso a datos • Que ya este creada la clase común	
Pasos de Ejecución: • Loguearse en el CAS • Escoger el rol de vicedecano • Ir al menú • Seleccionar secciones • Insertar datos • Verificar que se cree una nueva sección	
Resultado Esperado: Que el método acepte vocales con tilde y espacios	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_02-TI_01	**Tarea Ingeniería:** TI_01 Implementar el método soloLetras para validar el js
Nombre: Verificar que el método se implemente correctamente en la interfaz de usuario	
Responsable: Kevin Chimbo	**Fecha:** 25/03/2019
Descripción: Verificar que el método se implemente correctamente en la interfaz de usuario	
Condiciones de Ejecución: • Controlador debe estar creado • La vista debe estar creada	
Pasos de Ejecución: • Loguearse en el CAS • Escoger el rol de vicedecano • Ir al menú • Seleccionar secciones • Verificar que no se ingrese caracteres especiales	
Resultado Esperado: Que el método se implemente correctamente en la interfaz de usuario	
Evaluación de la Prueba: Exitosa	

HU-05 Insertar ítems

Historia de Usuario	
Número: HU_05	**Nombre:** Insertar ítems
Modificación de metáfora del sistema:	
Usuario: Vicedecano	**Sprint Asignada:** 7
Prioridad en Negocio: Medio	**Puntos Estimados:** 30
Riesgo en desarrollo: Medio	**Puntos Reales:** 30
Descripción: Se desarrollará para dar mantenimiento a los campos de secciones	
Observaciones:	
(Reverso) Pruebas de Aceptación • Verificar que el usuario no ingrese caracteres especiales en los campos • Verificar la conexión a la base de datos	

TAREA DE INGENIERÍA

Historia de Usuario: Insertar ítems	
Número de Tarea: TI_01	**Nombre de Tarea:** T1_01 Implementar el método soloLetras para validar el js
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 30
Fecha Inicio: 26/03/2019	**Fecha Fin:** 01/04/2019
Programador Responsable: Kevin Chimbo	
Descripción: Implementar el método soloLetras para validar el js	
PRUEBAS DE ACEPTACIÓN	
• Verificar que el método acepte vocales con tilde y espacios • Verificar que el método se implemente correctamente en la interfaz de usuario	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-HU_05	**Nombre de la Historia**: Insertar ítems
Nombre: Verificar que el usuario no ingrese caracteres especiales en los campos	
Responsable: Sandra Ordoñez	**Fecha:** 01/04/2019
Descripción: Se verificará que el usuario no ingrese caracteres especiales en los campos	
Condiciones de Ejecución: • Que este creada la base de datos • Que los servicios funcionen correctamente • Que este creado el método que devuelva el usuario	
Pasos de Ejecución: • Ingresar al sistema de estafetas • Escoger el rol de vicedecano • Ir al menú • Escoger ítems • Verificar que no se ingresen caracteres especiales	
Resultado Esperado: Que el usuario no ingrese caracteres especiales en los campos	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_02-HU_05	**Nombre de la Historia**: Insertar ítems
Nombre: Verificar la conexión a la base de datos	
Responsable: Sandra Ordoñez	**Fecha:** 01/04/2019
Descripción: Se verificará la conexión a la base de datos	
Condiciones de Ejecución: • Tener la codificación realizada para la conexión a la base de datos	
Condiciones de Ejecución: • Limpiar y construir el proyecto	

• Cargar el proyecto • Verificar si la codificación implementada permite la conexión a la base de datos
Resultado Esperado: Que el usuario no ingrese caracteres especiales en los campos
Evaluación de la Prueba: Exitosa

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_01	**Tarea Ingeniería:** TI_01 Implementar el método soloLetras para validar el js
Nombre: Verificar que el método acepte vocales con tilde y espacios	
Responsable: Sandra Ordoñez	**Fecha:** 28/03/2019
Descripción: Verificar que el método acepte vocales con tilde y espacios	
Condiciones de Ejecución: • Que ya este creado el acceso a datos • Que ya este creada la clase común	
Pasos de Ejecución: • Loguearse en el CAS • Escoger el rol de vicedecano • Ir al menú • Seleccionar ítems • Insertar datos • Verificar que se cree un nuevo ítem	
Resultado Esperado: Que el método acepte vocales con tilde y espacios	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_02-TI_01	**Tarea Ingeniería:** TI_01 Implementar el método soloLetras para validar el js
Nombre: Verificar que el método se implemente correctamente en la interfaz de usuario	
Responsable: Sandra Ordoñez	**Fecha:** 01/04/2019
Descripción: Verificar que el método se implemente correctamente en la interfaz de usuario	
Condiciones de Ejecución: • Controlador debe estar creado • La vista debe estar creada	
Pasos de Ejecución: • Loguearse en el CAS • Escoger el rol de vicedecano	

• Ir al menú • Seleccionar ítems • Verificar que no se ingrese caracteres especiales
Resultado Esperado: Que el método se implemente correctamente en la interfaz de usuario
Evaluación de la Prueba: Exitosa

HT-10 Modificar apertura y cierre de la conexión a la base de datos

HISTORIA TÉCNICA DEL SISTEMA		
Número: HT_10	**Nombre de la Historia:** Modificar apertura y cierre de la conexión a la base de datos	
Modificación de historia de usuario:		
Usuario: Desarrollador		**Sprint Asignada:** 7
Prioridad en el Negocio: Alta		**Puntos Estimados:**30
Riesgo en el Desarrollo: Alta		**Puntos Reales:** 30
Descripción: Como desarrollador quiero modificar la apertura y cierre de la conexión a la base de datos		
Observaciones: La base de datos esta realizadas con funciones y procedimientos almacenados.		

Historia de Usuario (Reverso) Pruebas de Aceptación
• Verificar si al hacer el cambio la base de datos ya no se sature

TAREA DE INGENIERÍA		
Historia Técnica: HT_10 Modificar apertura y cierre de la conexión a la base de datos		
Número de Tarea: TI_01	**Nombre de Tarea:** Modificar el acceso a datos en todas las funcionalidades	
Tipo de Tarea: Desarrollo		**Puntos Estimados:** 30
Fecha Inicio: 02/04/2019		**Fecha Fin:** 08/04/2019
Programador Responsable: Sandra Ordoñez		
Descripción: Se modificará el acceso a datos en todas las funcionalidades		
Pruebas de Aceptación • Verificar que los cambios se hayan hecho de acuerdo al estándar de codificación		

PRUEBA DE ACEPTACIÓN

Código: PA_01-HT_10	Nombre de la Historia: **Modificar apertura y cierre de la conexión a la base de datos**
Nombre: Verificar si al hacer el cambio la base de datos ya no se sature	
Responsable: Kevin Chimbo	**Fecha:** 08/04/2019
Descripción: Se verificará si al hacer el cambio la base de datos ya no se satura	
Condiciones de Ejecución: • Que este creado el acceso a datos	
Pasos de Ejecución: • Desplegar la aplicación • Ejecutar el acceso a datos • Ejecutar un método del acceso a datos • Verificar si el acceso a datos trae información de la base de datos	
Resultado Esperado: Que al realizar el cambio la base de datos no se sature	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01_TI_01	**Historia Técnica:** TI_01 Modificar el acceso a datos en todas las funcionalidades
Nombre: Verificar que los cambios se hayan hecho de acuerdo al estándar de codificación	
Responsable: Kevin Chimbo	**Fecha:** 08/04/2019
Descripción: Se verificará que los cambios realizados se hayan hecho de acuerdo al estándar de codificación	
Condiciones de Ejecución: • Que este creado el acceso a datos	
Pasos de ejecución: • Limpiar y construir • Desplegar el acceso a datos • Realizar peticiones a la base de datos • Observar si existen datos	
Resultado esperado: Que los cambios realizados estén realizados de acuerdo al estándar de codificación	
Evaluación de la prueba: Exitosa	

HT-11 Seguridad en contra de los ataques CSRF

HISTORIA TÉCNICA DEL SISTEMA	
Número: HT_11	**Nombre de la Historia:** Seguridad en contra de los ataques CSRF
Modificación de historia de usuario:	

Usuario: Desarrollador	**Sprint Asignada:** 8
Prioridad en el Negocio: Alta	**Puntos Estimados:**30
Riesgo en el Desarrollo: Alta	**Puntos Reales:** 30
Descripción: Como desarrollador quiero prevenir los ataques CSRF	
Observaciones: La base de datos esta realizadas con funciones y procedimientos almacenados.	

Historia de Usuario (Reverso) Pruebas de Aceptación
• Verificar si la aplicación es vulnerable ante el ataque CSRF

TAREA DE INGENIERÍA	
Historia Técnica: HT_11 Seguridad en contra de los ataques CSRF	
Número de Tarea: TI_01	**Nombre de Tarea:** Instalar la librería CsrfGuard
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 30
Fecha Inicio: 09/04/2019	**Fecha Fin:** 15/04/2019
Programador Responsable: Kevin Chimbo	
Descripción: Se instalará la librería CsrfGuard para prevenir los ataques CSRF	
Pruebas de Aceptación • Verificar que el uso de la librería mitigue la vulnerabilidad CSRF	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-HT_11	Nombre de la Historia: **Seguridad en contra de los ataques CSRF**
Nombre: Verificar si la aplicación es vulnerable ante el ataque CSRF	
Responsable: Sandra Ordoñez	**Fecha:** 15/04/2019
Descripción: Se verificará si la aplicación es vulnerable ante el ataque CSRF	
Condiciones de Ejecución: • Que este creada la interfaz de usuario	
Pasos de Ejecución: • Iniciar la interfaz de usuario • Loguearse en el CAS • Realizar alguna gestión en el sistema • Realizar ataque CSRF	
Resultado Esperado: Tener un resultado ante el ataque CSRF	

Evaluación de la Prueba: Exitosa

PRUEBA DE ACEPTACIÓN	
Código: PA_01_TI_01	**Historia Técnica:** TI_01 Instalar la librería CsrfGuard
Nombre: Verificar que el uso de la librería mitigue la vulnerabilidad CSRF	
Responsable: Sandra Ordoñez	**Fecha:** 15/04/2019
Descripción: Se verificará que la librería instalada mitigue la vulnerabilidad CSRF	
Condiciones de Ejecución: • Obtener la librería CsrfGuard • Que este creada la interfaz de usuario	
Pasos de ejecución: • Iniciar la interfaz de usuario • Loguearse en el CAS • Realizar alguna gestión en el sistema • Realizar ataque CSRF	
Resultado esperado: Que la aplicación no sea vulnerable ante el ataque CSRF	
Evaluación de la prueba: Fallida	

HT-12 Instalar el sistema en el servidor de pruebas institucional

HISTORIA TÉCNICA DEL SISTEMA	
Número: HT_12	**Nombre de la Historia:** Instalar el sistema en el servidor de pruebas institucional
Modificación de historia de usuario:	
Usuario: Desarrollador	**Sprint Asignada:** 8
Prioridad en el Negocio: Alta	**Puntos Estimados:**30
Riesgo en el Desarrollo: Alta	**Puntos Reales:** 30
Descripción: Como desarrollador quiero instalar el sistema en el servidor de pruebas institucional	
Observaciones: La base de datos esta realizadas con funciones y procedimientos almacenados.	

Historia de Usuario (Reverso) Pruebas de Aceptación
• Verificar si se ejecuta correctamente el sistema en el servidor

TAREA DE INGENIERÍA	
Historia Técnica: HT_12 Instalar el sistema en el servidor de pruebas institucional	
Número de Tarea: TI_01	**Nombre de Tarea:** Instalar la interfaz de usuario, el acceso a datos y la base de datos en el servidor
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 30
Fecha Inicio: 16/04/2019	**Fecha Fin:** 22/04/2019
Programador Responsable: Sandra Ordoñez	
Descripción: Se instalará la interfaz de usuario, el acceso a datos y la base de datos en el servidor	
Pruebas de Aceptación • Verificar la comunicación entre la interfaz de usuario, el acceso a datos y el servidor	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-HT_12	**Nombre de la Historia:** Instalar el sistema en el servidor de pruebas institucional
Nombre: Verificar si se ejecuta correctamente el sistema en el servidor	
Responsable: Kevin Chimbo	**Fecha:** 22/04/2019
Descripción: Se verificará si se ejecuta correctamente el sistema en el servidor	
Condiciones de Ejecución: • Que este creado el acceso a datos • Que este creada la interfaz de usuario • Que este creada la base a datos	
Pasos de Ejecución: • Insertar la url de la aplicación en el navegador • Loguearse en el CAS • Realizar gestiones en la aplicación • Verificar que se obtenga datos en la interfaz de usuario	
Resultado Esperado: Que se haya ejecutado el sistema correctamente	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01_TI_01	**Historia Técnica:** TI_01 Instalar la interfaz de usuario, el acceso a datos y la base de datos en el servidor
Nombre: Verificar la comunicación entre la interfaz de usuario, el acceso a datos y el servidor	
Responsable: Kevin Chimbo	**Fecha:** 22/04/2019
Descripción: Se verificará que se	
Condiciones de Ejecución: • Que este creado el acceso a datos • Que este creada la interfaz de usuario	

• Que este creada la base a datos
Pasos de ejecución: • Ejecutar el acceso a datos **y** la interfaz de usuario • Ejecutar un método del acceso a datos **y** la interfaz de usuario • Verificar si el acceso a datos trae información de la base de datos
Resultado esperado: Que exista comunicación entre la interfaz de usuario, el acceso a datos y el servidor
Evaluación de la prueba: Exitosa

HT-13 Revisión final del sistema

HISTORIA TÉCNICA DEL SISTEMA	
Número: HT_13	**Nombre de la Historia:** Revisión final del sistema
Modificación de historia de usuario:	
Usuario: Desarrollador	**Sprint Asignada:** 8
Prioridad en el Negocio: Alta	**Puntos Estimados:**30
Riesgo en el Desarrollo: Alta	**Puntos Reales:** 30
Descripción: Como desarrollador quiero instalar el sistema en el servidor de pruebas institucional	
Observaciones: La base de datos esta realizadas con funciones y procedimientos almacenados.	

Historia de Usuario (Reverso) Pruebas de Aceptación
• Verificar con el cliente el funcionamiento de la aplicación

TAREA DE INGENIERÍA	
Historia Técnica: HT_13 Revisión final del sistema	
Número de Tarea: TI_01	**Nombre de Tarea:** Reunirse con el cliente para revisar todas las funcionalidades que componen el sistema
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 30
Fecha Inicio: 23/04/2019	**Fecha Fin:** 29/04/2019
Programador Responsable: Kevin Chimbo	
Descripción: Se realizarán reuniones con el cliente para revisar todas las funcionalidades que componen el sistema	
Pruebas de Aceptación • Verificar que las funcionalidades sean las requeridas por el cliente	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-HT_13	**Nombre de la Historia:** Revisión final del sistema
Nombre: Verificar con el cliente el funcionamiento de la aplicación	
Responsable: Sandra Ordoñez	**Fecha:** 29/04/2019
Descripción: Se verificará con el cliente el funcionamiento de la aplicación	
Condiciones de Ejecución: • Que la aplicación este instalada en el servidor	
Pasos de Ejecución: • Inicializar la aplicación • Loguearse en el CAS • Realizar todas las gestiones que componen el sistema de estafetas	
Resultado Esperado: Completar la revisión del sistema	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01_TI_01	**Historia Técnica:** TI_01 Reunirse con el cliente para revisar todas las funcionalidades que componen el sistema
Nombre: Verificar que las funcionalidades sean las requeridas por el cliente	
Responsable: Sandra Ordoñez	**Fecha:** 29/04/2019
Descripción: Se verificará que las funcionalidades sean las requeridas por el cliente	
Condiciones de Ejecución: • Que este instalada la aplicación en el servidor	
Pasos de ejecución: • Inicializar la aplicación • Loguearse en el CAS • Realizar todas las gestiones que componen el sistema de estafetas	
Resultado esperado: Que todas las funcionalidades estén correctamente desarrolladas en función a los requerimientos del cliente	
Evaluación de la prueba: Exitosa	

HT-14 Documentación final del sistema

HISTORIA TÉCNICA DEL SISTEMA	
Número: HT_14	**Nombre de la Historia:** Documentación final del sistema
Modificación de historia de usuario:	
Usuario: Desarrollador	**Sprint Asignada:** 8
Prioridad en el Negocio: Alta	**Puntos Estimados:**30

Riesgo en el Desarrollo: Alta	**Puntos Reales:** 30
Descripción: Como desarrollador quiero instalar el sistema en el servidor de pruebas institucional	
Observaciones: La base de datos esta realizadas con funciones y procedimientos almacenados.	

Historia de Usuario (Reverso) Pruebas de Aceptación
• Verificar que todas las funcionalidades se hayan documentado correctamente

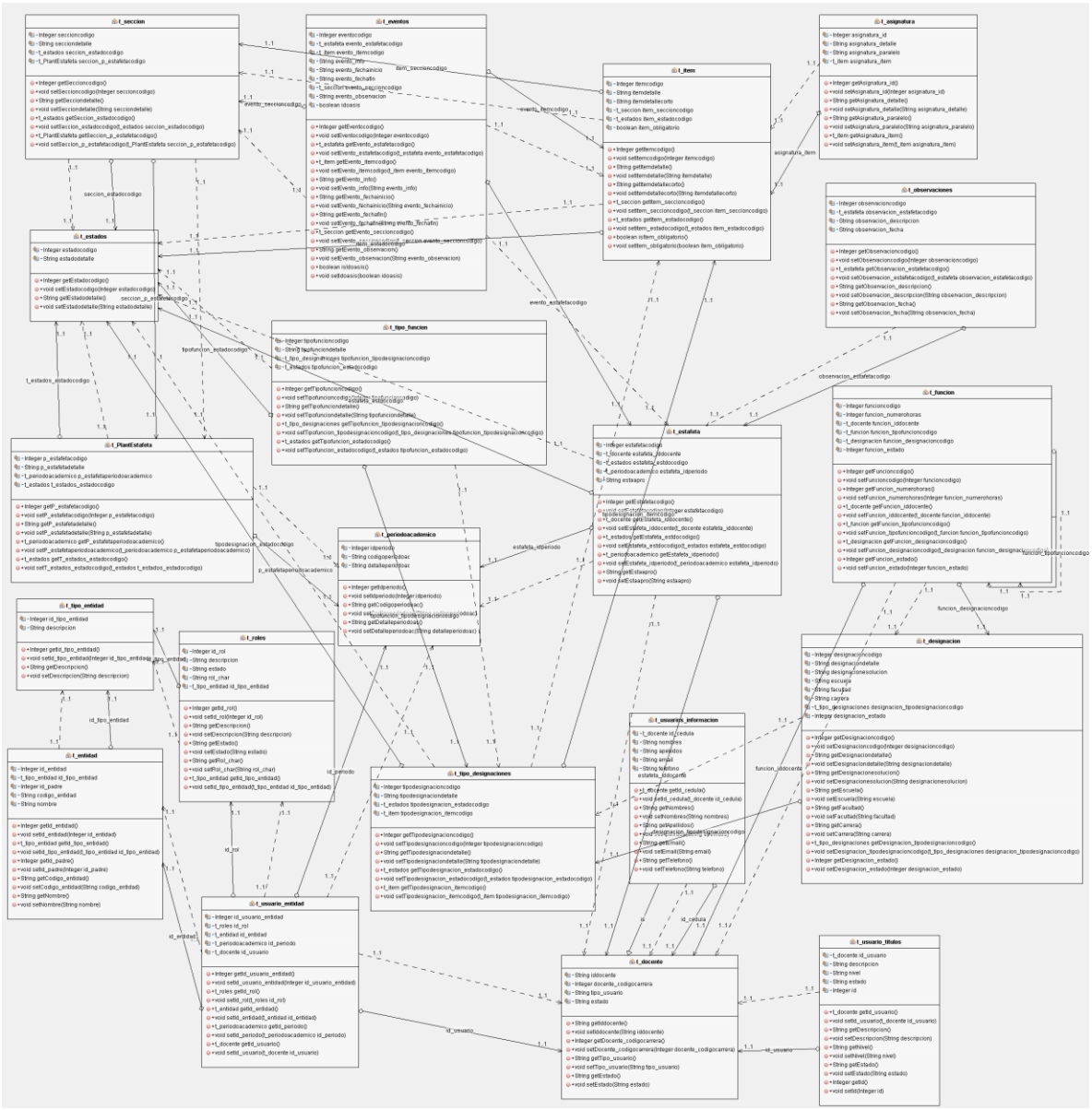
TAREA DE INGENIERÍA	
Historia Técnica: HT_14 Documentación final del sistema	
Número de Tarea: TI_01	**Nombre de Tarea:** Documentar las tareas de ingeniería y las pruebas de aceptación
Tipo de Tarea: Desarrollo	**Puntos Estimados:** 30
Fecha Inicio: 30/04/2019	**Fecha Fin:** 06/05/2019
Programador Responsable: Sandra Ordoñez	
Descripción: Se realizarán las tareas de ingeniería y las pruebas de aceptación para cada funcionalidad	
Pruebas de Aceptación • Verificar que las tareas de ingeniería y las pruebas de aceptación correspondan a la funcionalidad requerida	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-HT_14	**Nombre de la Historia:** Documentación final del sistema
Nombre: Verificar que todas las funcionalidades se hayan documentado correctamente	
Responsable: Kevin Chimbo	**Fecha:** 06/05/2019
Descripción: Se verificará que todas las funcionalidades se hayan documentado correctamente	
Condiciones de Ejecución: • Que se haya documentado todas las funcionalidades	
Pasos de Ejecución: • Verificar la documentación de cada funcionalidad	
Resultado Esperado: Que todas las funcionalidades se hayan documentado correctamente	
Evaluación de la Prueba: Exitosa	

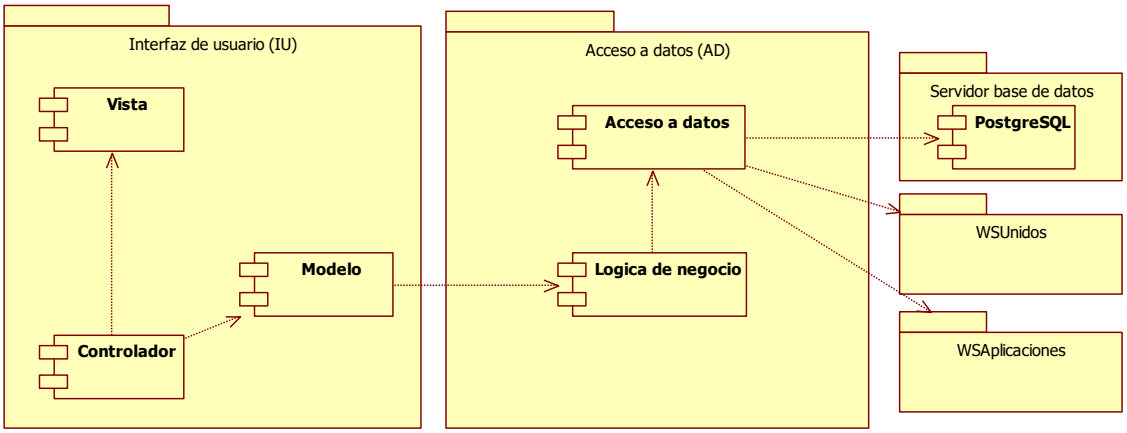
PRUEBA DE ACEPTACIÓN

Código: PA_01_TI_01	**Historia Técnica:** TI_01 Documentar las tareas de ingeniería y las pruebas de aceptación
Nombre: Verificar que las tareas de ingeniería y las pruebas de aceptación correspondan a la funcionalidad requerida	
Responsable: Kevin Chimbo	**Fecha:** 06/05/2019
Descripción: Se verificará que las tareas de ingeniería y las pruebas de aceptación correspondan a la funcionalidad requerida	
Condiciones de Ejecución: • Que las tareas de ingeniería y las pruebas de aceptación estén documentadas	
Pasos de ejecución: • Verificar cada historia de usuario con sus tareas de ingeniería y pruebas de aceptación	
Resultado esperado: Que las tareas de ingeniería y las pruebas de aceptación correspondan a la funcionalidad requerida	
Evaluación de la prueba: Exitosa	

Anexo K: Diagrama de clases



Anexo L: Diagrama de componentes



Anexo M: Manual de buenas prácticas de seguridad

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERÍA EN SISTEMAS

MANUAL DE BUENAS PRÁCTICAS

APLICACIONES WEB DESARROLLADAS EN JAVA

AUTORES

SANDRA JOHANNA ORDOÑEZ GRANIZO

KEVIN DAVID CHIMBO ORTIZ

Riobamba – Ecuador

2019

INTRODUCCIÓN

El propósito de este manual de buenas prácticas es facilitar al desarrollador técnicas que permiten mitigar 6 vulnerabilidades que los sistemas informáticos poseen, la aplicación de las técnicas se realizó en el sistema del distributivo de jornada de trabajo semanal del docente.

CONTENIDO

1. Implementación del sistema

a) Requerimientos hardware

Contar con:

- Computador personal.
- Conexión a Internet.

b) REQUERIMIENTOS SOFTWARE

Contar con:

- Sistema operativo Windows.
- Navegador (Internet Explorer, Google Chrome, Mozilla Firefox, etc.).
- Entorno de desarrollo integrado (Netbeans, Eclipse, etc.)

2. Buenas prácticas de seguridad

2.1. SQL Injection

Es un ataque en contra de bases de datos como se observa en la Figura 4-1, el mismo que mediante parámetros de entrada o URL´s agrega código SQL (lenguaje de consulta estructurado) con el objetivo de acceder o manipular la base de datos (OWASP Foundation, 2017).

2.1.1. Técnica para prevenir SQL Injection

Usar PreparedStatrments en lugar de Statements en la recuperación de consultas SQL.

```java
public Integer idPeriodoInsert(AccesoDatos ad, String codPeriodo) {
    int id = 0;
    try {
        String SQL = "INSERT INTO estafeta.t_periodoacademico(\n"
                + "            codigoperiodoac, detalleperiodoac)\n"
                + "    VALUES ('" + codPeriodo + "', '');";
        //Statement ps = ad.getCon().createStatement();
        PreparedStatement ps = (PreparedStatement) ad.getCon().createStatement();
        ps.executeUpdate(SQL);
        id = idPeriodoSelect(ad, codPeriodo);
    } catch (SQLException e) {
        id = -2;
    } finally {ad.Desconectar();}
    return id;
}
```

Ilustración 1: PreparedStatement en Java

Validar entradas en el lado del cliente mediante el archivo JavaScript validaciones

```javascript
function soloLetras(e) {
    key = e.keyCode || e.which;
    tecla = String.fromCharCode(key).toLowerCase();
    letras = " áéíóúabcdefghijklmnñopqrstuvwxyz";
    especiales = [8, 39, 44, 46, 59];

    tecla_especial = false
    for (var i in especiales) {
        if (key == especiales[i]) {
            tecla_especial = true;
            break;
        }
    }

    if (letras.indexOf(tecla) == -1 && !tecla_especial)
        return false;
}
```

Ilustración 2: Archivo validaciones.js

```
<div class='input-group'>
    <input type='hidden' id='tipo' value='Regis'>
    <span style='width: 20%'class='input-group-addon' >Detalle:</span>
    <input type='text' class='form-control' id='detalle'  placeholder="CARRERA"
           onkeypress="return soloLetras(event)" onblur="limpia()" required="required" maxlength="100"/>
</div>

<div class='input-group'>
    <input type='hidden' id='tipo' value='Regis'>
    <span style='width: 20%' class='input-group-addon' >Resolución:</span>
    <input  type='text' class='form-control' id='resolucion'  placeholder="EF-001"
           onkeypress="return soloResolucion(event)" onblur="limpia()" required="required" maxlength="50"/>
    <span class='input-group-addon'><input id ='oficio' name='oficio' type="file"><a class='nav-link text-black fa fa-cloud-upload'  data-toggle='tooltip'
</div>
```

Ilustración 3: Uso del archivo validaciones.js y sus métodos

2.2. Cross site scripting (XSS)

Este ataque fue creado para vulnerar aplicaciones web, la misma que permite vulnerar el sistema mediante inserción de código JavaScript y como consecuencia el atacante podrá robar contraseñas, redirigir a otros sitios web falsos, entre otros (Soto, 2016).

2.2.1. Técnica para prevenir Cross site scripting (XSS)

Validación de entradas con la filtración de caracteres especiales en parámetros de formato establecido y filtración de etiquetas HTML, con el JavaScript validaciones.

```
function soloLetras(e) {
    key = e.keyCode || e.which;
    tecla = String.fromCharCode(key).toLowerCase();
    letras = " áéíóúabcdefghijklmnñopqrstuvwxyz";
    especiales = [8, 39, 44, 46, 59];

    tecla_especial = false
    for (var i in especiales) {
        if (key == especiales[i]) {
            tecla_especial = true;
            break;
        }
    }

    if (letras.indexOf(tecla) == -1 && !tecla_especial)
        return false;
}
```

Ilustración 4: Archivo validaciones.js

```
<div class='input-group'>
    <input type='hidden' id='tipo' value='Regis'>
    <span style='width: 20%'class='input-group-addon' >Detalle:</span>
    <input type='text' class='form-control' id='detalle'  placeholder="CARRERA"
           onkeypress="return soloLetras(event)" onblur="limpia()" required="required" maxlength="100"/>
</div>

<div class='input-group'>
    <input type='hidden' id='tipo' value='Regis'>
    <span style='width: 20%' class='input-group-addon' >Resolución:</span>
    <input  type='text' class='form-control' id='resolucion'  placeholder="EF-001"
           onkeypress="return soloResolucion(event)" onblur="limpia()" required="required" maxlength="50"/>
    <span class='input-group-addon'><input id ='oficio' name='oficio' type="file"><a class='nav-link text-black fa fa-cloud-upload'  data-toggle='tooltip'
</div>
```

Ilustración 5: Uso del archivo validaciones.js y sus métodos

2.3. Buffer overflow (DOS)

Es un ataque informático orientado a cualquier sistema, en el cual se sobrepasa la capacidad máxima del búfer y se pierde acceso a los servicios o al sistema que se está consultando (Paguay, 2015), en

las aplicaciones web, este ataque se realiza mediante la inserción de una extensa cantidad de caracteres en campos sin validaciones correspondientes.

2.3.1. Técnica para prevenir Buffer overflow (DOS)

Filtrar la longitud del valor del parámetro para evitar la saturación de memoria del servidor

```
<div class='input-group'>
    <input type='hidden' id='tipo' value='Regis'>
    <span style='width: 20%'class='input-group-addon' >Detalle:</span>
    <input type='text' class='form-control' id='detalle'  placeholder="CARRERA"
        onkeypress="return soloLetras(event)" onblur="limpia()" required="required" maxlength="100"/>
</div>

<div class='input-group'>
    <input type='hidden' id='tipo' value='Regis'>
    <span style='width: 20%' class='input-group-addon' >Resolución:</span>
    <input  type='text' class='form-control' id='resolucion'  placeholder="EF-001"
        onkeypress="return soloResolucion(event)" onblur="limpia()" required="required" maxlength="50"/>
    <span class='input-group-addon'><input id ='oficio' name='oficio' type="file"><a class='nav-link text-black fa fa-cloud-upload'  data-toggle='tooltip'
</div>
```

Ilustración 6: Filtrado de la longitud de las variables

2.4. Directorio transversal

Este ataque permite tener acceso a archivos de directorios superiores directamente del equipo que se está irrumpiendo, esto se genera cuando la aplicación no está asegurada adecuadamente para realizar validaciones en entradas de texto y los parámetros de las URL´s (fluidAttacks, 2018).

2.4.1. Técnica para prevenir el ataque de directorio transversal

Validar entradas en el lado del cliente con el JavaScript validaciones

```
<div class='input-group'>
    <input type='hidden' id='tipo' value='Regis'>
    <span style='width: 20%'class='input-group-addon' >Detalle:</span>
    <input type='text' class='form-control' id='detalle'  placeholder="CARRERA"
        onkeypress="return soloLetras(event)" onblur="limpia()" required="required" maxlength="100"/>
</div>

<div class='input-group'>
    <input type='hidden' id='tipo' value='Regis'>
    <span style='width: 20%' class='input-group-addon' >Resolución:</span>
    <input  type='text' class='form-control' id='resolucion'  placeholder="EF-001"
        onkeypress="return soloResolucion(event)" onblur="limpia()" required="required" maxlength="50"/>
    <span class='input-group-addon'><input id ='oficio' name='oficio' type="file"><a class='nav-link text-black fa fa-cloud-upload'  data-toggle='tooltip'
</div>
```

Ilustración 7: Uso del archivo validaciones.js y sus métodos

2.5. Cross site request forgery (CSRF)

La falsificación de solicitudes entre sitios es un tipo de ataque informático que duplica la sesión del usuario autenticado en la aplicación, y lo obliga a ejecutar acciones no autorizadas, son utilizadas al cambio de estados, pues el atacante no sabe cuál es el resultado ya que no puede ver la respuesta de la solicitud clonada (OWASP Foundation, 2018)

2.5.1. Técnica para prevenir Cross site request forgery (CSRF)

Mitigación basada en token, en la Ilustración 7, se observa que se obtiene un TOKEN del servidor de autenticación CAS para así poder abrir las páginas que el usuario legalmente autenticado requiere.

```
if (request.getUserPrincipal() != null) {
   AttributePrincipal principal = (AttributePrincipal) request.getUserPrincipal();
    final Map attributes = principal.getAttributes();
   if (attributes != null) {
       Iterator attributeNames = attributes.keySet().iterator();
     out.println("<b>Attributes:</b>");
       if (attributeNames.hasNext()) {
           out.println("<hr><table border='3pt' width='100%'>");
           out.println("<th colspan='2'>Attributes</th>");
           out.println("<tr><td><b>Key</b></td><td><b>Value</b></td></tr>");
           for (; attributeNames.hasNext(); ) {
               out.println("<tr><td>");
               String attributeName = (String) attributeNames.next();
               out.println(attributeName);
               out.println("</td><td>");
               final Object attributeValue = attributes.get(attributeName);
               if (attributeValue instanceof List) {
                   final List values = (List) attributeValue;
                   out.println("<strong>Multi-valued attribute: " + values.size() + "</strong>");
                   out.println("<ul>");
                   for (Object value : values) {
                       out.println("<li>" + value + "</li>");
                   }
                   out.println("</ul>");
               } else {
                   out.println(attributeValue);
               }
               out.println("</td></tr>");
           }
           out.println("</table>");

   } else {
       out.println("<pre>The attribute map is empty. Review your CAS filter configurations.</pre>");
   }
```

Ilustración 8: Mitigación basada en token mediante CAS Server

Implementar la librería CSRF Guard de OWASP Foundation

Los usuarios pueden descargar la última versión de OWASP CSRFGuard mediante uno de los siguientes enlaces:

- Descargue y genere la fuente en - https://github.com/aramrami/OWASP-CSRFGuard

La instalación de OWASP CSRFGuard 3 es muy sencilla y requiere tres pasos simples:

1. Copie el archivo Owasp.CsrfGuard.jar a la ruta de clase de su aplicación

2. Declare CsrfGuard en el descriptor de implementación de su aplicación (web.xml)

```
<listener>
            <listener-class>
org.owasp.csrfguard.CsrfGuardServletContextListener </listener-
class>
       </listener>
       <listener>
            <listener-class>
org.owasp.csrfguard.CsrfGuardHttpSessionListener </listener-class>
       </listener>
       <context-param>
             <param-name> Owasp.CsrfGuard.Config </param-name>
             <param-value> WEB-INF / Owasp.CsrfGuard.properties
</param-value>
       </context-param>
       <context-param>
             <param-name> Owasp.CsrfGuard.Config.Print </param-
name>
             <param-value> true </param-value>
       </context-param>
<filter>
    <filter-name> CSRFGuard </filter-name>
    <filter-class> org.owasp.csrfguard.CsrfGuardFilter </filter-
class>
</filter>
<filter-mapping>
    <filter-name> CSRFGuard </filter-name>
    <url-pattern> / * </url-pattern>
</filter-mapping>
```

3. Configure el archivo Owasp.CsrfGuard.properties como mejor le parezca

Los ajustes de configuración mínimos que los usuarios deben revisar incluyen:

- Página de inicio de token nueva predeterminada (org.owasp.csrfguard.NewTokenLandingPage)

```
org.owasp.csrfguard.Logger = org.owasp.csrfguard.log.ConsoleLogger
org.owasp.csrfguard.NewTokenLandingPage = / Owasp.CsrfGuard.Test /
index.jsp
org.owasp.csrfguard.TokenPerPage = true
org.owasp.csrfguard.Rotate = false
```

- Soporte para Ajax y XMLHttpRequest (org.owasp.csrfguard.Ajax)

```
org.owasp.csrfguard.Ajax = true
```

- Recursos URI que no deberían estar protegidos (org.owasp.csrfguard.unprotected. *)

```
org.owasp.csrfguard.unprotected.Tag = / Owasp.CsrfGuard.Test /
index.jsp
org.owasp.csrfguard.unprotected.JavaScriptServlet = /
Owasp.CsrfGuard.Test / JavaScriptServlet
org.owasp.csrfguard.unprotected.Html = *. jsp
org.owasp.csrfguard.unprotected.Public = / MySite / Public / *
```

- Acciones ejecutadas cuando se detecta un ataque (org.owasp.csrfguard.action. *)

```
sintaxis: org.owasp.csrfguard.action. [actionName] = [className]
ejemplo: org.owasp.csrfguard.action.class.Redirect =
org.owasp.csrfguard.actions.Redirect
```

Los usuarios de OWASP CSRFGuard pueden insertar tokens de prevención en HTML utilizando dos estrategias:

- Manipulación de DOM de JavaScript: proceso en gran parte automatizado que requiere un esfuerzo mínimo y es ideal para la mayoría de las aplicaciones web
- Biblioteca de etiquetas JSP: proporciona una estrategia detallada ideal para situaciones donde la manipulación automatizada de DOM es insuficiente.

2.6. Intercepción criptográfica

Los ataques de intercepción criptográficos son orientados a obtener las claves utilizadas para cifrar información almacenada en un sistema (Gómez Vieites, 2014, p.6).

Según (Paguay, 2015, p.15) esta técnica intenta capturar el tráfico+ que se genera por el uso de la aplicación y así obtener datos importantes que le permitan al atacante ingresar al sistema para extraer, modificar o insertar información.

2.6.1. Técnica para prevenir Intercepción criptográfica

Encriptar las claves de autenticación, como se muestra en la Ilustración 8.

```
<form action="procesaseguro.jsp" name="form" id="form" onsubmit="encriptar()" method="post" >
        INICIO DE SESI&Oacute;N DEL USUARIO <br />
        C&eacute;dula:<input type="text" maxlength="20" name="txtCedula" id="txtCedula"><br />
        Clave:<input type="password" name="txtClave" id="txtClave"><br />
        <input type="submit" name="btnIngresar" id="btnIngresar" value="Ingresar">
</form>
```

Ilustración 9: Encriptación de la clave de usuario

Instalar certificados SSL, con esto se encriptará el canal de comunicación cuando exista tráfico en la red.

BIBLIOGRAFÍA

FLUIDATTACKS (2018) *Evitar Ataques de Directorio Transversal*. Disponible en: https://fluidattacks.com/web/es/defends/csharp/evitar-direct-transversal/ (Accedido: 6 de mayo de 2019).

GÓMEZ VIEITES, Á. (2014) *REDES INFORMÁTICAS*. Disponible en: www.keylogger.com (Accedido: 6 de mayo de 2019).

OWASP FOUNDATION (2017) «OWASP Top 10 -2017», p. 25. Disponible en: https://www.owasp.org/images/5/5e/OWASP-Top-10-2017-es.pdf (Accedido: 4 de noviembre de 2018).

OWASP FOUNDATION (2018) *Falsificación de solicitudes entre sitios (CSRF) - OWASP*. Disponible en: https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF) (Accedido: 6 de mayo de 2019).

PAGUAY, P. (2015) *Propuesta de técnicas de aseguramiento de aplicaciones web desarrolladas en Java*. Escuela Superior Politécnica de Chimborazo. Disponible en: http://dspace.espoch.edu.ec/handle/123456789/4029.

SOTO, M. (2016) *¿Qué es XSS?* Disponible en: https://medium.com/@marvin.soto/qué-es-xss-b9330eedbc07 (Accedido: 5 de mayo de 2019).