AUDITORIA AL CUMPLIMIENTO DE UNA POLITICA DE DESARROLLO SEGURO BASADA EN LA ISO 27001

CASO DE ESTUDIO: ESCUELA COLOMBIANA DE INGENIERIA JULIO GARAVITO

NICOLAS ALMANZAR ESPITIA

JUAN DAVID VANZINA SOLIS

UNIVERSIDAD CATÓLICA DE COLOMBIA

FACULTAD DE INGENIERÍA

PROGRAMA DE ESPECIALIZACIÓN EN AUDITORIA EN SISTEMAS DE INFORMACION

BOGOTÁ D.C 2018

AUDITORIA AL CUMPLIMIENTO DE UNA POLITICA DE DESARROLLO SEGURO BASADA EN LA ISO 27001

CASO DE ESTUDIO: ESCUELA COLOMBIANA DE INGENIERIA JULIO GARAVITO

NICOLAS ALMANZAR ESPITIA

JUAN DAVID VANZINA SOLIS

Trabajo de grado para obtener el título de especialista en Auditoría de Sistemas de Información

Asesor: PhD. ALEXANDRA MARÍA LÓPEZ SEVILLANO

UNIVERSIDAD CATÓLICA DE COLOMBIA

FACULTAD DE INGENIERÍA

PROGRAMA DE ESPECIALIZACIÓN EN AUDITORIA EN SISTEMAS DE INFORMACION

BOGOTÁ D.C 2018

TABLA DE CONTENIDO

LISTA DE ILUSTRACIONES

LISTA DE TABLAS

Pág.

RESUMEN

La Escuela Colombiana de Ingeniería Julio Garavito cuenta con un área de desarrollo de software, la cual se encarga de responder a las necesidades de la comunidad con la construcción de nuevos sistemas de información desarrollados a la medida.

El presente proyecto tiene como finalidad auditar el proceso de desarrollo de software de La Escuela Colombiana de Ingeniería Julio Garavito, de acuerdo con los lineamientos de desarrollo seguro definidos por la institución basados en la norma ISO 27001, con el fin de hallar oportunidades de mejora. Para cumplir con el objetivo del proyecto, inicialmente se realizará la identificación del estado actual del proceso de desarrollo de software por medio de instrumentos de recolección de datos como encuestas, entrevistas y observación directa, con la información recopilada se desarrollará una guía de auditoria que permitirá evaluar el cumplimiento de la política de desarrollo seguro.

Posteriormente se ejecutarán las pruebas de auditoria y con base a los resultados obtenidos se documentarán los hallazgos y las recomendaciones. Las recomendaciones emitidas se priorizarán de acuerdo al nivel de riesgo de cada etapa del proceso de desarrollo de software, con el fin de identificar las recomendaciones que se deberían ejecutar con mayor prioridad.

Palabras clave: Desarrollo de software, sistemas de información, auditar, desarrollo seguro, ISO 27001, hallazgos, oportunidades de mejora, instrumentos, encuestas, entrevistas, observación directa, guía de auditoria, política, pruebas de auditoria, recomendaciones.

ABSTRACT

The Colombian School of Engineering Julio Garavito has a software development area, which is responsible for responding to the needs of the community with the construction of new information systems developed to measure.

The purpose of this project is to audit the software development process of the Colombian School of Engineering Julio Garavito, in accordance with the safe development guidelines defined by the institution based on the ISO 27001 standard, in order to find opportunities for improvement. To fulfill the objective of the project, initially the identification of the current state of the software development process will be carried out by means of data collection instruments such as surveys, interviews and direct observation, with the collected information an audit guide will be developed that will allow assess compliance with the safe development policy.

Subsequently, the audit tests will be carried out and, based on the results obtained, the findings and recommendations will be documented. The recommendations issued will be prioritized according to the level of risk of each stage of the software development process, in order to identify the recommendations that should be executed with the highest priority.

Keywords: Software development, information systems, auditing, secure development, ISO 27001, findings, improvement opportunities, instruments, surveys, interviews, direct observation, audit guide, policy, audit tests, recommendations

INTRODUCCIÓN

Actualmente las organizaciones han adoptado como solución a sus necesidades tecnológicas incluir dentro de su de área de TI (Tecnologías de la información) una coordinación dedicada al desarrollo de software, con el fin de diseñar y construir sus propias soluciones a medida en menor tiempo e incluso a un menor precio.

Esta iniciativa si bien optimiza recursos es también fuente de preocupación al hablar de riesgos, y es que si no se lleva a cabo un desarrollo bien definido y estructurado siguiendo metodologías y buenas prácticas pueden llevar a la organización a ver la materialización de amenazas en disponibilidad, confidencialidad e integridad en sus activos de información.

Por otra parte, las normas y políticas en las organizaciones son resultado de la necesidad de controlar y estandarizar procedimientos que perfectamente aplican al desarrollo de software, como es el caso de la Escuela Colombiana de Ingeniería Julio Garavito, en donde existe una política de seguridad de la información basada en la norma ISO27001 y una sección dedicada exclusivamente al desarrollo seguro.

La Escuela Colombiana de Ingeniería es una entidad colombiana, dedicada a ofrecer programas de educación superior que adopta un modelo mixto dado que cuenta con sus propios desarrolladores, pero también contrata soluciones de software y aplicaciones.

Dada la incertidumbre del cumplimiento o no de la política de seguridad y específicamente la sección de desarrollo seguro, en este trabajo se pretende evaluar el cumplimiento de las normas por parte de los desarrolladores de la institución.

GENERALIDADES

LÍNEA DE INVESTIGACIÓN

Software Inteligente y Convergencia Tecnológica puesto que al realizar la auditoria se evalúa el uso de buenas prácticas, el cumplimiento de controles de seguridad y procedimientos que se tienen establecidos en la institución para el desarrollo de software.

Con la auditoria se busca encontrar oportunidades de mejora para el proceso de desarrollo de software bajo el cumplimiento de la política de seguridad de la Institución la cual está regida por la norma ISO 27001.

PLANTEAMIENTO DEL PROBLEMA

Los gobiernos, las instituciones financieras, los hospitales y las organizaciones privadas a nivel mundial tienen información confidencial sobre sus empleados, productos, investigaciones, clientes, entre otros. Esta información es recolectada, procesada, almacenada y puesta a disposición de las personas que tengan el permiso de consultarla. Si se da el caso de que información confidencial de una organización llegue a personas equivocadas, esta se hará pública de una forma no autorizada y esto puede tener graves consecuencias, debido a que se perderá credibilidad de los clientes, se perderán posibles negocios, se puede enfrentar a demandas e incluso puede causar la quiebra de organizaciones. (ISOTools Excellence, 2015).

En el marco normativo de los estándares relacionados con la seguridad informática, está incluida la familia de estándares ISO/IEC 27000 e ISM3, que son normas específicas para la gestión de seguridad de la información y pueden ser aplicables a cualquier organización, independientemente de su tamaño o actividad. Otros estándares relacionados son los de calidad ISO 9001, medio ambientales como ISO 1 4000, de TI como el estándar CobIT y de entrega de servicios ITIL. (Metodología de análisis y evaluación de riesgos aplicados a la seguridad informática y de información bajo la norma ISO/IEC 27001)

La familia de estándares de la ISO/IEC 27000 se encuentra la norma ISO 27001, la cual se refiere a la confidencialidad, la integridad y la disponibilidad de la información y los datos importantes para la organización, independientemente del formato que tengan, estos pueden ser: electrónicos, en papel, audio o vídeo. Mediante la especificación de las secciones y controles que son necesarios para declarar la conformidad en la implementación de un sistema de gestión de la seguridad de la información en función de las buenas prácticas.

En la Escuela Colombiana de Ingeniería Julio Garavito se definió una política de seguridad de la información basada en la norma ISO 27001 en la que se definen los controles para los procesos de la institución. Específicamente en la sección de desarrollo seguro se detallan las normas que encaminan a que el desarrollo interno o externo de los sistemas de información institucionales cumplan con los requerimientos de seguridad esperados, así como con metodologías para la realización de pruebas de aceptación y seguridad al software desarrollado. Sin embargo, para la política de desarrollo seguro no existen procedimientos que evalúen el cumplimiento de estos lineamientos alineados al ciclo PHVA.

Antecedentes Del Problema.

La Seguridad de la Información, según ISO27001, se refiere a la confidencialidad, la integridad y la disponibilidad de la información y los datos importantes para la organización, independientemente del formato que tengan, estos pueden ser: electrónicos, en papel, audio o vídeo.

Los gobiernos, las instituciones financieras, los hospitales y las organizaciones privadas tienen información confidencial sobre sus empleados, productos, investigaciones, clientes, entre otros. Esta información es recolectada, procesada, almacenada y puesta a disposición de las personas que tengan el permiso de consultarla. Si se da el caso de que información confidencial de la organización caen en manos de la competencia, esta se hará pública de una forma no autorizada y esto puede tener graves consecuencias, debido a que se perderá credibilidad de los clientes, se perderán posible negocios, se puede enfrentar a demandas e incluso puede causar la quiebra de la organización. (ISOTools Excellence, 2015).

Es por esto que surge la necesidad “proteger la información, ya que es un requisito del negocio, y se convierte en algo ético y una obligación legal” (ISOTools Excellence, 2015) .

Actualmente la Escuela Colombiana de Ingeniería Julio Garavito cuenta con una política de seguridad de la información, en donde se encuentra un conjunto de normas que describen los lineamientos de seguridad de la información.

La política de desarrollo seguro definida en el manual de políticas se encamina a:

> Que el desarrollo interno o externo de los sistemas de información cumpla con los requerimientos de seguridad esperados, con las buenas prácticas para desarrollo seguro de aplicativos, así como con metodologías para la realización de pruebas de aceptación y seguridad al software desarrollado. Además, se asegurará que todo software desarrollado o adquirido, interna o externamente cuenta con el nivel de soporte requerido por la institución.(OSIRIS (Abril.2018).

Sin embargo, para la política de desarrollo seguro no existen procedimientos que evalúen el cumplimiento de estas normas estipuladas, y es allí donde se requiere una manera eficaz que permita realizar esta evaluación.

Pregunta De Investigación.

¿La Auditoría al cumplimiento de la política de desarrollo seguro basada en la ISO 27001 mejora la construcción de software en la escuela colombiana de ingeniería Julio Gravito?

Variables Del Problema.

Variables dependientes:

- Calidad: El software desarrollado no cuentan con un nivel de calidad adecuado, el cual se ve reflejado en la no conformidad de los usuarios.
- Seguridad: los sistemas desarrollados no garantizan la confidencialidad e integridad de la información.
- Software: programas desarrollados de acuerdo a las necesidades de las áreas de la institución.
- Buenas prácticas: Es un conjunto de mejores prácticas que aportan a la calidad del software desarrollado, las cuales no se aplican en el proceso.
- Especificación de requerimientos: Falta de claridad en cómo debería funcionar el sistema a desarrollar.

Variables Independientes:

- Incumplimiento: La metodología de desarrollo utilizada, no está siendo efectiva puesto que hay demoras en las entregas de software o requerimientos nuevos.
- Recomendaciones: Opiniones dadas por los usuarios para el desarrollo del software.
- Claridad del usuario: Falta de claridad de la necesidad por parte de los usuarios.

Alcance y Limitaciones.

ALCANCE

El presente proyecto tiene como objetivo realizar una auditoría a la política de seguridad basada en la norma ISO 27001 de la Escuela Colombiana de Ingeniería Julio Garavito, teniendo como propósito la sección de desarrollo seguro para determinar el cumplimiento de los sistemas desarrollados internamente.

LIMITACIONES

- Para la ejecución de este proyecto contamos con un tiempo estimado de cuatro meses, lo cual impide la evaluación de toda la política.
- Disposición de tiempo de los empleados para brindar la información.
- Acceso a información de uso restringido por parte de la organización.

JUSTIFICACIÓN

Los avances tecnológicos y su impacto en la sociedad han exigido a las organizaciones, la implementación de sistemas de información que permitan transaccionalidad instantánea con altos estándares de seguridad, por esta razón, es importante para los involucrados en dichas transacciones la protección de los activos mediante "los tres principios de seguridad de la información (integridad, disponibilidad y confidencialidad)" (Erick A. Lamilla Rubio, 2009). Y estos principios deben estar sustentados sobre "los tres pilares de la seguridad de software (administración del riesgo, aplicación de prácticas especificas en etapas del ciclo de vida de desarrollo y el conocimiento)" (Tovar, E., Carrillo, J., Vega, V., Gasca., G. 2006).

Según un artículo presentado en enero de 2018 por Forrester Research en el que se evalúan las violaciones de acceso exitoso a empresas para el año 2017, se observó que los tres principales objetivos son: En primer lugar, las vulnerabilidades del software, En segundo lugar: la aplicación de ataques web (mediante aplicación de SQL, aplicación de scripts o inclusión remota de archivos) y, en tercer lugar: El robo de credenciales (DeMartine, 2018).

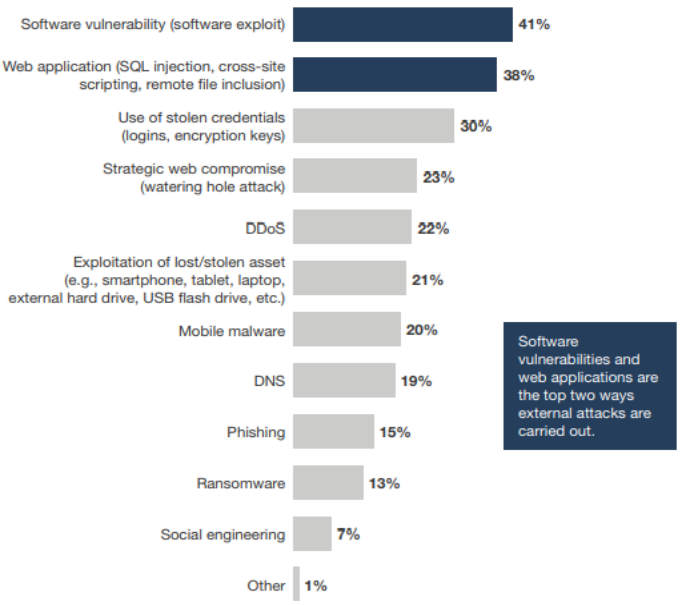


Ilustración 1. Objetivos principales en violaciones exitosas a software y aplicaciones en 2017

Dada la falta de implementación de buenas prácticas para la realización de software seguro y el alto costo generado a las empresas, Surge la necesidad de evaluar los lineamientos establecidos por cada empresa para buscar elementos que permitan la mejora y actualización continua tanto de sus políticas como del software resultante de ello.

El Instituto Nacional de Estándares y Tecnología (NIST, por sus siglas en inglés) estima que las correcciones de código realizadas después de la puesta en producción pueden dar como resultado 30 veces el costo de las reparaciones realizadas durante la fase de diseño. Como se muestra en la **¡Error! No se encuentra el origen de la referencia.**, el costo de corregir las vulnerabilidades es más alto después de que se ha implementado una aplicación. (Wayne, 2012)

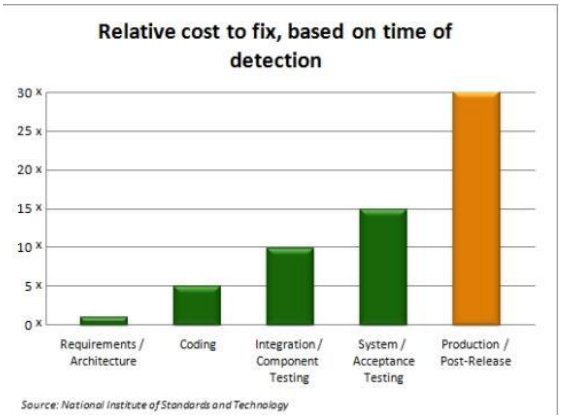


Ilustración 2 Costo relativo de reparación, basado en el tiempo de detección

Es por ello que las organizaciones establecen un conjunto de directrices, normas, procedimientos e instrucciones que deben ser adoptados por los empleados de los entes empresariales y que son estandarizados en un documento llamado Política de Seguridad, este documento estandariza y normaliza las actividades tanto humanas como tecnológicas de tal forma que se cumpla al máximo con los principios de seguridad de la información (Dussan.2006)

En el caso de estudio: La Escuela Colombiana de Ingeniería Julio Garavito, se tiene una política de seguridad de la información y dentro de ella, unos controles basados en la norma ISO 27001 para el desarrollo seguro, sin embargo, más allá de la existencia del documento no se tiene conocimiento de si se cumple o por lo menos en que porcentaje y tampoco se tiene conocimiento si los controles son suficientes o que buenas prácticas se pueden implementar.

OBJETIVOS

Objetivo General.

Auditar el proceso de desarrollo de software de la Escuela Colombiana de Ingeniería Julio Garavito bajo los lineamientos de la política de seguridad de la institución basada en la norma ISO 27001 para recomendar oportunidades de mejora en el proceso.

Objetivos Específicos.

1. Identificar la situación actual de la institución para obtener una familiarización con el proceso de desarrollo.

2. Establecer una guía de auditoria para alinear el desarrollo seguro con las mejores prácticas institucionales.

3. Aplicar la guía de auditoria seleccionada, con el fin de evaluar el cumplimiento de la política de desarrollo seguro.

MARCOS DE REFERENCIA

MARCO CONCEPTUAL

Política De Seguridad.

Las políticas de seguridad de la tecnología de la Información identifican las reglas y procedimientos para cada usuario que accede o usa los recursos tecnológicos de una organización.

Una política de seguridad efectiva es un modelo de la cultura de la organización en donde las reglas y los procedimientos son guiados por el acercamiento de los usuarios a la información y el trabajo, que tanto valoran la información y la perspectiva a la tolerancia a fallos. (Eugenio Duarte, 2014)

Sistema de información.

"Es un conjunto de elementos orientados al tratamiento y administración de datos e información, organizados y listos para su uso posterior, generados para cubrir una necesidad o un objetivo" (M.Cornejo Velázquez).

Buenas Prácticas.

Se refiere a toda experiencia que se guía por principios, objetivos y procedimientos apropiados o pautas aconsejables que se adecuan a una determinada perspectiva normativa o a un parámetro consensuado, así como también toda experiencia que ha arrojado resultados positivos, demostrando su eficacia y utilidad en un contexto concreto. (Caracola Consultores, s.f.)

Código Fuente.

"En el contexto de la informática, el código fuente se define como el conjunto de líneas de

textos, que son las directrices que debe seguir la computadora para realizar dicho programa; por lo que es en el código fuente, donde se encuentra escrito el funcionamiento de la computadora" (Definista, 2016).

El código fuente de un programa está escrito en un lenguaje de programación determinado, sin embargo, este tipo de "lenguaje no puede ser ejecutado directamente por el computador, sino que debe ser traducido a otro lenguaje que el ordenador pueda ejecutar más fácilmente. Para esta traducción se emplean los llamados compiladores, ensambladores o intérpretes" (Definista, 2016).

Datos

Los datos son la mínima unidad semántica, y se corresponden con elementos primarios de información que por sí solos son irrelevantes como apoyo a la toma de decisiones. También se pueden ver como un conjunto discreto de valores, que no dicen nada sobre el porqué de las cosas y no son orientativos para la acción. (Prusak, 1999)

Información.

La información se puede definir como un conjunto de datos procesados y que tienen un significado (relevancia, propósito y contexto), y que por lo tanto son de utilidad para quién debe tomar decisiones, al disminuir su incertidumbre (Prusak, 1999)

Confidencialidad.

La confidencialidad es la "garantía de que la información personal será protegida para que no sea divulgada sin consentimiento de la persona. Dicha garantía se lleva a cabo por medio de un grupo de reglas que limitan el acceso a ésta información" (José Alberto Ávila Funes, 2013).

Integridad.

"La integridad hace referencia a la cualidad de la información para ser correcta y no haber

sido modificada, manteniendo sus datos exactamente tal cual fueron generados, sin manipulaciones ni alteraciones por parte de terceros" (Firma-e, 2014).

Evaluación.

Consiste en analizar mediante pruebas la calidad y cumplimiento de funciones, actividades y procedimientos que se realizan en una organización o área. "Las evaluaciones se utilizan para valorar registros, planes, presupuestos, programas, controles y otros aspectos que afectan la administración y control de una organización o las áreas que la integran" (Solarte, 2011).

La evaluación se aplica para "investigar algún hecho, comprobar alguna cosa, verificar la forma de realizar un proceso, evaluar la aplicación de técnicas, métodos o procedimientos de trabajo, verificar el resultado de una transacción, comprobar la operación correcta de un sistema software entre otros muchos aspectos" (Solarte, 2011).

Riesgo.

"La incertidumbre de que ocurra un evento y pueda tener un impacto en el logro de los objetivos. El riesgo se mide en términos de impacto y probabilidad" (Instituto de Auditores Internos, 2012).

CLASIFICACIÓN DE LOS RIESGOS:

Riesgo Estratégico.

El riesgo estratégico se define como el "impacto actual y futuro en los ingresos y el capital que podría surgir de las decisiones adversas de negocios, la aplicación indebida de las decisiones, o la falta de capacidad de respuesta a los cambios de la industria". (J.P. Morgan).

Riesgo Reputacional.

Es un riesgo de pérdida resultante de daños a la reputación de una empresa, de pérdida de

ingresos; aumento de los costos operativos, de capital o regulatorios; o la destrucción del valor para el accionista, como consecuencia de un evento adverso o potencialmente delictivo, incluso si la empresa no es declarada culpable. Los eventos adversos típicamente asociados con el riesgo de reputación incluyen la ética, la seguridad, la sostenibilidad, la calidad y la innovación.

Riesgo Leal.

Es la posibilidad de pérdida en que incurre una empresa o entidad al ser sancionada, multada u obligada a indemnizar daños como resultado del incumplimiento de normas o regulaciones y obligaciones contractuales.

El riesgo legal surge también como consecuencia de fallas en los contratos y transacciones, derivadas de actuaciones malintencionadas, negligencia o actos involuntarios que afectan la formalización o ejecución de contratos o transacciones.

Riesgo Operativo.

Riesgo producido por fallos de los procesos básicos de la organización. Es la posibilidad de incurrir en pérdidas por deficiencias, fallas o inadecuaciones, en el recurso humano, los procesos, la tecnología, la infraestructura o por la ocurrencia de acontecimientos externos.

Riesgos De Contagio.

Es la posibilidad de pérdida que una empresa o entidad puede sufrir, directa o indirectamente, por una acción o experiencia de un relacionado o asociado.

Riesgos Tecnológicos.

Se define como la pérdida potencial por daños, interrupción, alteración o fallas derivadas del uso o dependencia en el hardware, software, aplicaciones, redes, y cualquier otro canal de distribución de información en la prestación de servicios con los clientes internos o externos de la empresa.

Riesgos financieros.

Es la posibilidad de que el desempeño financiero de la organización sea diferente de lo planeado asociada al cumplimiento de compromisos por parte de sus contrapartes (riesgo de crédito), al cambio en los precios o valores de referencia de los activos (riesgo de mercado) o a la capacidad de obtener recursos en efectivo o cumplir con sus obligaciones líquidas (riesgo de liquidez).

Amenaza.

"Es toda acción que aprovecha una vulnerabilidad para atentar contra la seguridad de un sistema de información. Es decir, que podría tener un potencial efecto negativo sobre algún elemento de nuestros sistemas" (INCIBE, 2017).

Vulnerabilidad.

Es una debilidad o fallo en un "sistema de información que pone en riesgo la seguridad de la información pudiendo permitir que un atacante pueda comprometer la integridad, disponibilidad o confidencialidad de la misma, por lo que es necesario encontrarlas y eliminarlas lo antes posible" (INCIBE, 2017).

Control.

Es el conjunto de normas, técnicas, acciones y procedimientos que interrelacionados e interactuando entre sí con los sistemas y subsistemas organizacionales y administrativos, permite evaluar, comparar y corregir aquellas actividades que se desarrollan en las organizaciones, garantizando la ejecución de los objetivos y el logro de las metas institucionales. (Solarte, 2011).

Desarrollo de software

El Desarrollo de Software es una disciplina que estudia los componentes necesarios para la

creación, gestión, mantenimiento y testeo de software computacional. “El software puede entenderse como la programación lógica que todo sistema computacional necesita para funcionar apropiadamente y permitir al usuario disfrutar de aspectos como una interfaz amigable y las funciones que el programa realice” (universidades.cr, 2018).

Ciclo de vida del desarrollo de Software:

“Es una secuencia estructurada y bien definida de las etapas en Ingeniería de software para desarrollar el producto software deseado” (TutorialsPoint, 2017).

Oportunidades De Mejora:

Consiste en el análisis de los procesos "Tal como está" y la identificación de temas problemáticos y mejoras potenciales. “Factores como la necesidad de adaptarse a cambios reglamentarios, asuntos de seguridad, y el esfuerzo para garantizar que la información llegue correctamente, pueden impulsar los cambios necesarios”. (Guía de implementación de la facilitación del comercio, 2012).

PHVA:

“El nombre del Ciclo PDCA (o Ciclo PHVA) viene de las siglas Planificar, Hacer, Verificar y Actuar, en inglés “Plan, Do, Check, Act” (Bernal., 2017). También es conocido como Ciclo de mejora continua o Círculo de Deming, por ser Edwards Deming su autor.

Esta metodología describe los cuatro pasos esenciales que se deben llevar a cabo de forma sistemática para lograr la mejora continua, entendiendo como tal al mejoramiento continuado de la calidad (disminución de fallos, aumento de la eficacia y eficiencia, solución de problemas, previsión y eliminación de riesgos potenciales…). (Bernal., 2017).

El círculo de Deming lo componen 4 etapas cíclicas, de forma que una vez acabada la etapa

final se debe volver a la primera y repetir el ciclo de nuevo, “de forma que las actividades son reevaluadas periódicamente para incorporar nuevas mejoras. La aplicación de esta metodología está enfocada principalmente para para ser usada en empresas y organizaciones” (Bernal., 2017).

Auditoria Interna.

“La auditoría interna es un sistema de control interno de la empresa y consiste en el conjunto de medidas, políticas y procedimientos establecidos en una organización concreta para proteger su activo, minimizar riesgos, incrementar la eficacia de los procesos operativos y optimizar y rentabilizar, en definitiva, el negocio”. (emprendepyme).

Informe De Auditoría:

Expresión escrita por el auditor respecto a los resultados de las verificaciones realizadas durante la ejecución de la auditoría, manifestando sus criterios y comentarios respecto a los estados financieros y otros hechos económicos. (Definista, 2016)

Sistema:

“Hace referencia a un todo organizado y complejo; un conjunto o combinación de cosas o partes que forman un todo complejo o unitario. Es un conjunto de objetos unidos por alguna forma de interacción o interdependencia” (Caliche, 2011).

Probabilidad:

Es la oportunidad de que algo suceda, “esté o no definido, medido o determinado objetiva o subjetivamente, cualitativa o cuantitativamente, y descrito utilizando términos generales o

matemáticos (como la probabilidad numérica o la frecuencia en un periodo de tiempo determinado).(ISO 31000:2018 - Guía ISO 73:2009)" (INCONTEC.2011).

Calidad:

Grado en el que un conjunto de características cumple con los requisitos.(ISO 9001:2015) tipificación de la calidad

Auditoría TI:

"Es la revisión y evaluación de procesos implementados en la empresa con respecto a sus equipos de cómputo, como se están utilizando y su eficiencia. Estas auditorías son necesarias también para lograr una utilización más eficiente y segura de la información" (Moyano, 2017).

MARCO TEÓRICO

En esta sección se hace referencia a los términos y metodologías o normas que serán tratadas a lo largo de desarrollo del proyecto y que deben ser tenidas en cuenta para obtener una mejor contextualización de lo aplicado.

ISO 27001.

ISO 27001 es una norma internacional emitida por la Organización Internacional de Normalización (ISO) y describe cómo gestionar la seguridad de la información en una empresa. La revisión más reciente de esta norma fue publicada en 2013 y ahora su nombre completo es ISO/IEC 27001:2013. La primera revisión se publicó en 2005 y fue desarrollada en base a la norma británica BS 7799-2.

ISO 27001 puede ser implementada en cualquier tipo de organización, con o sin fines de lucro, privada o pública, pequeña o grande. Está redactada por los mejores especialistas del mundo en el tema y proporciona una metodología para implementar la gestión de la seguridad de la información en una organización. También permite que una empresa sea certificada; esto significa que una entidad de certificación independiente confirma que la seguridad de la información ha sido implementada en esa organización en cumplimiento con la norma ISO 27001. (Segovia, advisera)

¿POR QUÉ ISO 27001 ES IMPORTANTE PARA LA EMPRESA?.

Hay 4 ventajas comerciales esenciales que una empresa puede obtener con la implementación de esta norma para la seguridad de la información (Segovia, 27001academy, 2018) :

- Cumplir con los requerimientos legales – cada vez hay más y más leyes, normativas y requerimientos contractuales relacionados con la seguridad de la información. La buena noticia es que la mayoría de ellos se pueden resolver implementando ISO 27001 ya que esta norma le proporciona una metodología perfecta para cumplir con todos ellos.

- Obtener una ventaja comercial – si su empresa obtiene la certificación y sus competidores no, es posible que usted obtenga una ventaja sobre ellos ante los ojos de los clientes a los que les interesa mantener en forma segura su información.

- Menores costos – la filosofía principal de ISO 27001 es evitar que se produzcan incidentes de seguridad, y cada incidente, ya sea grande o pequeño, cuesta dinero; por lo tanto, evitándolos su empresa va a ahorrar mucho dinero. Y lo mejor de todo es que la inversión en ISO 27001 es mucho menor que el ahorro que obtendrá.

- Una mejor organización – en general, las empresas de rápido crecimiento no tienen tiempo para hacer una pausa y definir sus procesos y procedimientos; como consecuencia, muchas veces los empleados no saben qué hay que hacer, cuándo y quién debe hacerlo. La implementación de ISO 27001 ayuda a resolver este tipo de situaciones ya

que alienta a las empresas a escribir sus principales procesos (incluso los que no están relacionados con la seguridad), lo que les permite reducir el tiempo perdido de sus empleados.

ISO 33000.

La familia de normas ISO/IEC 33000 define un marco de trabajo para la evaluación de procesos software.

El propósito de la serie de estándares ISO/IEC 33000 es proporcionar un enfoque estructurado para la evaluación de procesos, permitiendo a las organizaciones:

2. Comprender el estado de sus propios procesos buscando la mejora de los mismos.
3. Determinar la idoneidad de sus propios procesos para un requerimiento en particular o para un conjunto de requerimientos.
4. Determinar la idoneidad de los procesos de otra organización para un contrato específico o para un conjunto de contratos.

Esta norma, está constituida por los siguientes elementos:

33001 Conceptos y terminología.

33002 Requisitos para realizar la evaluación del proceso.

33003 Requisitos para los marcos de medición del proceso.

33004 Requisitos para los modelos d

e proceso.

OWASP.

El proyecto abierto de seguridad en aplicaciones Web (OWASP por sus siglas en inglés) es una comunidad abierta dedicada a habilitar a las organizaciones para desarrollar, comprar y mantener aplicaciones confiables. Todas las herramientas, documentos, foros y capítulos de OWASP son gratuitos y abierto a cualquiera interesado en mejorar la seguridad de aplicaciones. Abogamos por resolver la seguridad de aplicaciones como un problema de gente, procesos y tecnología porque las soluciones más efectivas incluyen mejoras en todas estas áreas. (Creative Commons Attribution-ShareAlike, 2014)

Niveles de verificación de seguridad de aplicaciones:

El Estándar de Verificación de Seguridad en Aplicaciones define tres niveles de verificación de seguridad, incrementando la profundidad con cada nivel.

1. ASVS nivel 1 se encuentra dirigido a todo tipo de software.
2. ASVS nivel 2 es para aplicaciones que contienen datos sensibles, que requieren protección.
3. ASVS nivel 3 es para las aplicaciones más críticas - aplicaciones que realizan transacciones de alto valor, contienen datos médicos confidenciales, o cualquier aplicación que requiera el más alto nivel de confianza.

Cada nivel ASVS contiene una lista de requerimientos de seguridad. Cada uno de estos requisitos puede también corresponderse a funcionalidades específicas de seguridad y

capacidades que deben construirse por los desarrolladores de software. (OWASP, 2017)

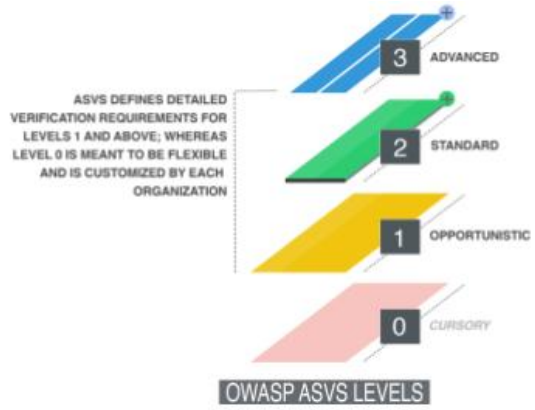


Figura 1 Niveles del Estándar de Verificación de Seguridad en Aplicaciones de OWASP

REQUERIMIENTOS DE SEGURIDAD OWASP:

"Los siguientes temas deben ser considerados durante las actividades de entendimiento de riesgo y definición de requerimientos. Este esfuerzo debe producir un conjunto de requerimientos ajustados, específicos y probables". (OWASP, 2015)

Validación De Entradas Y Codificación:

Los requerimientos deben especificar las reglas para validación y codificación de cada dato de entrada a la aplicación, ya sea de usuarios, sistemas de archivos, bases de datos o sistemas externos. La regla predeterminada debe ser que todas las entradas sean validadas a menos que cumplan con una especificación detallada de que está permitido. "Además, los requerimientos deben especificar las acciones a tomar cuando se reciben entradas no válidas. Específicamente, la aplicación no debe ser susceptible a inyecciones, desbordamientos, manipulación y otros ataques con entradas de usuario corruptas".(OWASP, 2015).

Autentificación Y Manejo De Sesiones:

Los requerimientos deben especificar como se protegerán las credenciales para autentificación y los identificadores de sesión a través del ciclo de desarrollo de software. “Los requerimientos para todas las funciones relacionadas deben ser agregados incluyendo recuperar contraseñas, cambio de contraseñas, recordar contraseñas, desconexión y conexión múltiple” (OWASP, 2015).

Control De Acceso:

Los requerimientos deben incluir una descripción detallada de todos los roles (grupos, privilegios, autorizaciones) usadas en la aplicación. Los requerimientos deben indicar todos los activos y funciones que provee la aplicación. “Los requerimientos deben especificar detallada y exactamente los derechos de acceso para cualquier activo y función de cada rol. Se sugiere utilizar un formato de matriz de control de acceso para documentar estas reglas” (OWASP, 2015) .

Manejo de Errores:

“Los requerimientos deben detallar como se van a manejar los errores que ocurran dentro del procesamiento. Algunas aplicaciones deberían hacer lo mejor posible en caso de un error, mientras que otras deberían terminar su procesamiento inmediatamente” (OWASP, 2015).

Historial:

Los requerimientos deben especificar que eventos son relevantes para la seguridad y necesitan ser registrados, como ataques detectados, intentos de conexión fallidos e intentos de exceder la autorización. “Los requerimientos deben especificar también que información registrar con cada evento, incluyendo hora y fecha, descripción del evento, detalles de aplicación, y otra información útil en esfuerzos forenses” (OWASP, 2015).

Conexiones A Sistemas Externos:

Los requerimientos deben especificar como la autentificación y cifrado será manejado para todos los sistemas externos, tales como bases de datos, directorios y servicios Web. “Todas las credenciales requeridas para la comunicación con sistemas externos deben almacenarse cifradas y fuera del código dentro de archivos de configuración” (OWASP, 2015).

Cifrado:

Los requerimientos “deben especificar qué datos deben ser cifrados, como serán cifrados y como todos los certificados y otras credenciales deben ser manejados. Las aplicaciones deben usar algoritmos estándar implementados en una librería de cifrado que hayan sido usadas y probadas ampliamente” (OWASP, 2015).

Disponibilidad:

“Los requerimientos deben especificar como protegerse de ataques de negación de servicio.

Todos los posibles ataques en la aplicación deben ser considerados, incluyendo bloqueos de autentificación, agotamiento de conexiones y otros ataques de agotamiento de recursos" (OWASP, 2015).

Configuración Segura:

Los requerimientos deben especificar que los valores predeterminados para todas las configuraciones relacionadas a seguridad deben ser seguras. "Para propósitos de auditoría, el software debería ser capaz de producir un reporte sencillo de leer que muestre los detalles de todas las configuraciones relacionadas con seguridad" (OWASP, 2015).

MARCO JURÍDICO

Para el presente trabajo se realizó una búsqueda de las principales leyes y normas aplicables a las políticas de información con el fin de tener conocimiento jurídico que aportan para el desarrollo del proyecto.

1. **Ley Estatutaria 15 81 de 2012 y Reglamentada Parcialmente por el Decreto Nacional 1377 de 2013:** Por la cual se dictan disposiciones generales para la protección de datos personales.

   Se toma esta norma para el proceso de levantamiento de información, con el fin de identificar su aplicabilidad en el desarrollo de software.

2. **Decreto 2578 de 2012:** Por medio del cual se reglamenta el Sistema Nacional de Archivos. Incluye "*El deber de entregar inventario de los documentos de archivo a cargo del servidor público, se circunscribe tanto a los documentos físicos en archivos tradicionales, como a los documentos electrónicos que se encuentren en equipos de cómputo, sistemas de información, medios portátiles*" … entre otras disposiciones.

   Se usa con el fin de conocer que tipos de archivos deben ser inventariados en el área de desarrollo de software y donde se encuentran ubicados los archivos que almacena cada sistema de información.

3. **Ley 1273 DE 2009:** Por medio de la cual se modifica el Código Penal, se crea un nuevo bien tutelado denominado “de la protección de la información y los datos” y se preservan integralmente los sistemas que utilicen las tecnologías de la información y las comunicaciones, entre otras disposiciones.

Se usa como referencia con el fin de conocer cómo se debe cumplir con la confidencialidad de la información manejada en los sistemas de información dela institución.

4. ISO 27002:2005: Esta norma proporciona recomendaciones de las mejores prácticas en la Gestión de la Seguridad de la Información a todos los interesados y responsables en iniciar e implantar o mantener sistemas de gestión de la seguridad de la información.

Se utiliza como referencia para comprender los controles que pueden utilizar en la seguridad de la información.

5. **ISO/IEC 27001:2005:** Es la evolución certificable del código de buenas prácticas ISO 17799. Define cómo organizar la seguridad de la información en cualquier tipo de organización, con o sin fines de lucro, privada o pública, pequeña o grande. Es posible afirmar que esta norma constituye la base para la gestión de la seguridad de la información.

Se usa con el fin de analizar e identificar el cumplimiento de esta norma en el ciclo de desarrollo de software y como esta influye en cada una de sus etapas. El manual de políticas de seguridad de la institución se basa en esta norma y es importante tener conocimiento de sus lineamientos.

6. **ISO/IEC TR 18044:2004:** Ofrece asesoramiento y orientación sobre la Seguridad de la Información de Gestión de incidencias para los administradores de seguridad de la información y de los administradores de sistemas de información.

Se usa como referencia para comprender el manejo adecuado de la información en los sistemas de información.

7. **Ley Estatutaria 1266 de 2008:** Por la cual se dictan las disposiciones generales de hábeas data y se regula el manejo de la información contenida en bases de datos personales, en especial la financiera, crediticia, comercial, de servicios y la proveniente de terceros países y se dictan otras disposiciones.

Es importante conocer cómo se regula el manejo de la información contenida en las bases de datos y el derecho que tiene los estudiantes en conocer actualizar y rectificar la información que se hayan recogido sobre ellos.

8. **Ley 1581 de 2012:** la cual se dictan disposiciones generales para la Protección de Datos Personales.

Se usa como referente frente al adecuado tratamiento de esa información y datos personales que se leen, insertan, modifican o eliminan en los sistemas de información.

MARCO GEOGRÁFICO.

El proyecto se lleva a cabo en la Escuela Colombiana de Ingeniería Julio Garavito, localizada en Bogotá-Colombia en la AK.45 No.205-59 (Autopista Norte) localidad de Usaquén.

Ilustración 3Ubicación Institución (Google Maps, 2018)

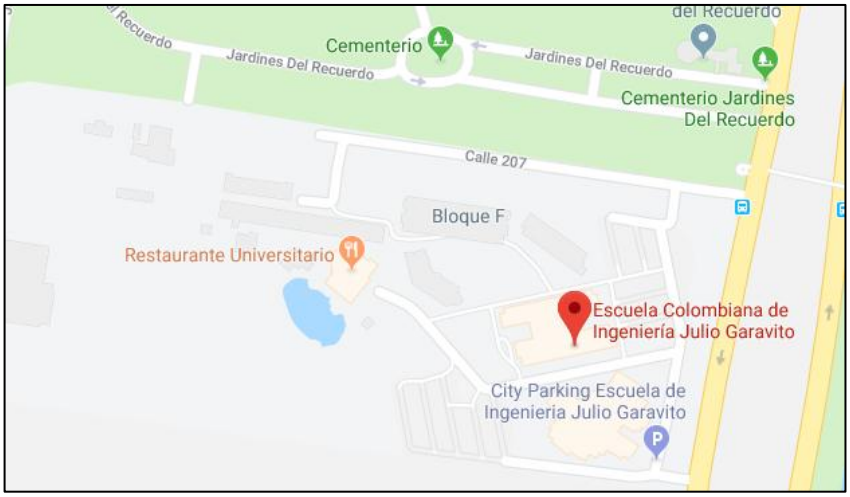


Ilustración 4 Ubicación Institución (Google Maps, 2018)

MARCO DEMOGRÁFICO.

La Escuela Colombiana de Ingeniería Julio Garavito es una institución universitaria de carácter privado con única sede en Bogotá D.C – Colombia, que tiene como misión la formación de la persona, fundamentada en una alta preparación científica y tecnológica, tiene aproximadamente 935 empleados y 4464 estudiantes activos, la población con la que se va trabajar el proyecto son 4 empleados de la coordinación de desarrollo de software, los cuales 3 cuentan con el título de Ingeniero de sistemas y uno se encuentra en proceso de finalización de estudios de ingeniería de sistemas. La institución pertenece al estrato socio económico 4.

ESTADO DEL ARTE

Dada la competencia, la globalización y las nuevas formas de comercio, las organizaciones deben estar preparadas para asegurar sus activos y en especial los de información. Sin importar el tamaño de la organización ese activo representa un capital importante para el funcionamiento en sí de cada institución, pero también para el apoyo a la toma decisiones.

Es por ello que en la actualidad, existen varias normas internacionales que estandarizan las condiciones y buenas prácticas en el manejo seguro de información y se evidencia la importancia de aplicar un SGSI (Sistema de Gestión de Seguridad de la información) en las organizaciones. Las principales y más usadas son las que pertenecen a la familia de la ISO/ IEC 27000.

“Este estándar internacional ha sido desarrollado para proporcionar un modelo para establecer, implementar, operar, monitorear, revisar, mantener y mejorar un SGSI” (Espinoza A. 2013). Y Cada uno de los puntos exigidos en la norma pertenece a una etapa de un proceso: Plan – Do – Check – Act (Planificar-Hacer-Verificar-Actuar), que se aplica para estructurar todos los

procesos del SGSI.

No es obligatorio en Colombia aún que una organización obtenga certificación en la norma ISO 27001 a diferencia de otros países latinoamericanos, pero es necesario implementar buenas prácticas que permitan establecer controles para proteger las características de la seguridad de la información. (Ladino, M., Villa, P. Lopez, A. (Abril. 2011).

Según la Organización Internacional de Estandarización (ISO), La certificación en ISO 27001 en "Sudamérica ha llevado una progresión creciente, en el año 2006 sólo existían 18 certificados, en 2010 ya eran 117 certificados y en el año 2016 la cifra ascendió a 564 certificados. Esto supuso un incremento del 1,7% en 10 años" (SGSI, 2017).

Los países más representativos, en cuanto a número de certificaciones, en Sudamérica son: Argentina con 88 certificaciones en ISO 27001, Brasil cuenta con 117 certificados, Chile tiene 49 certificados, Colombia con 163, Costa Rica cuenta con 21 certificados, Ecuador tiene 11 certificados en ISO 27001, México supera a la Colombia ya que tiene 221 certificados, Perú cuenta con algunos menos certificados en este caso 32 certificados en ISO 27001". (SGSI. 2017).

METODOLOGÍA

FASES DEL TRABAJO DE GRADO

El trabajo de grado se realizará en 5 etapas, las cuales permitirán alcanzar con el objetivo del proyecto:

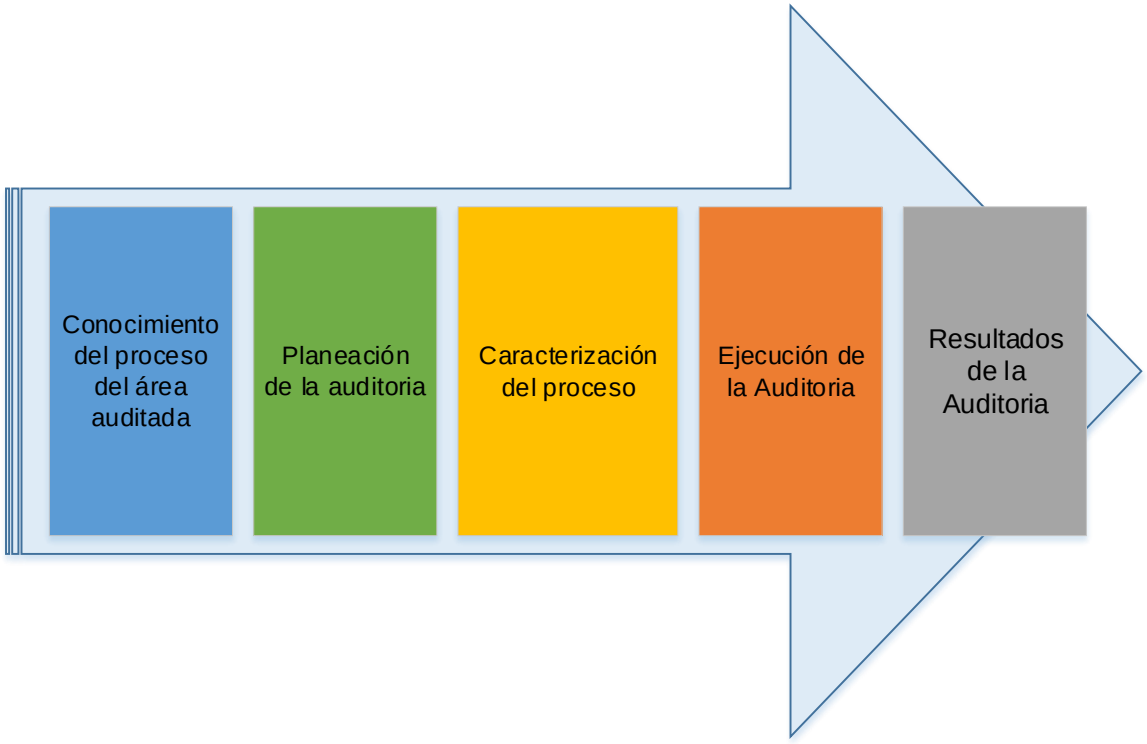


Ilustración 5 Metodología implementada en el proyecto - (Fuente propia, 2018)

ETAPA 1: Conocimiento Del Proceso Del Área Auditada.

Esta etapa tiene como fin la familiarización e identificación del funcionamiento actual del proceso de desarrollo de software de la institución con el fin de tener claridad y conocimiento de

las características del proceso a auditar.

Ilustración 6 Etapa 1 Conocimiento del proceso (Fuente propia, 2018)

ETAPA 2: Planeación De La Auditoria.

Consiste en definir los objetivos, alcance y limitación de la auditoria. Se selecciona los estándares que se van a utilizar para cumplir con el objetivo del proyecto, además se debe identificar y seleccionar las herramientas, instrumentos y procedimientos necesarios para la Auditoria. Se debe determinar los puntos de proceso que serán evaluados.

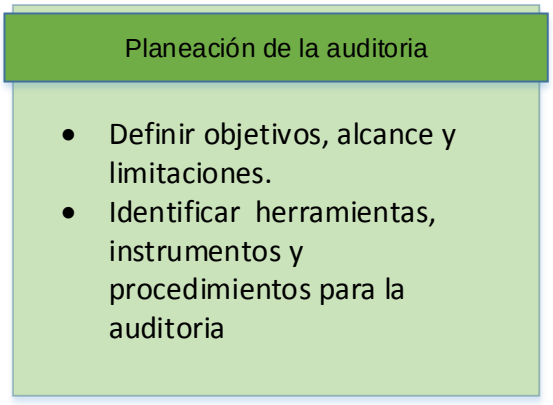


Ilustración 7 Etapa 2 Planeación de la auditoria (Fuente propia, 2018)

ETAPA 3: Caracterización Del Proceso

De acuerdo a la información recolectada en las etapas anteriores, se procede al diseño y elaboración de una guía de auditoria para aplicarla en la institución.

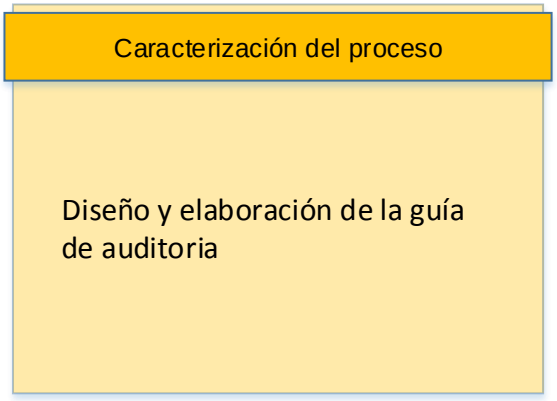


Ilustración 8 Etapa 3 Caracterización del proceso (Fuente propia, 2018)

ETAPA 4: Ejecución De La Auditoria

En esta etapa se procede a realizar la auditoria, aplicando la guía desarrollada en la etapa anterior la cual está determinada por las características y puntos del proceso los cuales serán evaluados.

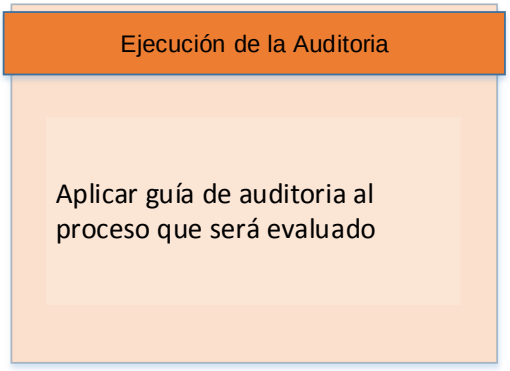


Ilustración 9 Etapa 4 Ejecución de la auditoria (Fuente propia, 2018)

ETAPA 5: Dictamen De La Auditoria

Finalmente se debe realizar un informe con las oportunidades de mejora encontrados en el proceso, además se debe emitir las recomendaciones que se deben tener en cuenta para mejorar el proceso auditado.

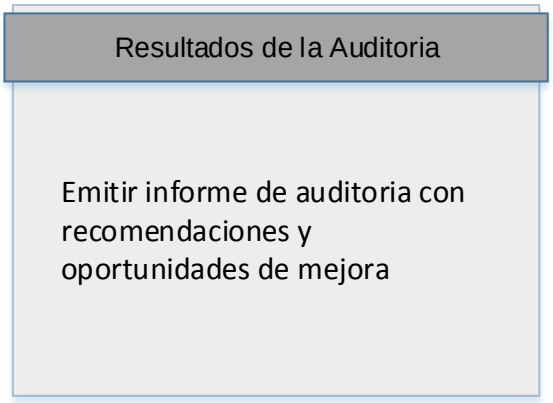


Ilustración 10 Resultados de la auditoria (Fuente propia, 2018)

METODO DE INVESTIGACION

El método de elaboración utilizado para la investigación es el cualitativo-descriptivo, el cual se basa en la inducción a partir de observación y entrevistas no estructuradas. Describir implica observar sistemáticamente el objeto de estudio y catalogar la información que se observa para que pueda ser utilizada y replicada por otros. (Deisy Yanez)

El principal método de investigación descriptiva son las encuestas aplicadas en un caso de estudio. Este proyecto está alineado a evidenciar de manera documental, el cumplimiento de la política de desarrollo seguro en el proceso actual de desarrollo de software.

ETAPAS DEL MÉTODO CUALITATIVO-DESCRIPTIVO

Identificación y delimitación del problema.

Es el primer paso de la investigación. "Se trata del momento en el que se decide lo que se va a investigar y el tipo de preguntas a las que se les buscará respuesta" (Deisy Yanez).

Elaboración y construcción de los instrumentos

De acuerdo con lo que se pretenda investigar, se deben seleccionar los instrumentos para la recogida de datos.

"Esta fase del proceso debe realizarse con cierta anticipación, para asegurarse de que los instrumentos serán los adecuados para obtener la información deseada" (Deisy Yanez).

Observación y registro de datos.

"Es un momento crucial dentro del proceso, puesto que implica estar atento a la realidad observada para poder tomar nota de la mayor cantidad de detalles posible" (Deisy Yanez).

"Lo ideal es que esta observación no altere las condiciones naturales en las que se da el fenómeno o la situación a estudiar" (Deisy Yanez).

Decodificación y categorización de la información.

En este momento del proceso, los datos percibidos se transcriben en algún formato y se organizan según su importancia o su significado.

De este modo, "será más fácil procesar la información cuando se trata de cantidades grandes o de categorías distintas que podrían confundirse" (Deisy Yanez).

Análisis.

"Una vez que los datos han sido catalogados, será el momento de su interpretación y análisis con referencia al objeto de estudio" (Deisy Yanez).

Ese análisis no debe establecer relaciones causales, puesto que la naturaleza del método no lo permite.

Propuestas.

Este es el "momento de proceso en el que se sugieren los siguientes pasos de la investigación del objeto de estudio dado" (Deisy Yanez).

INSTRUMENTOS O HERRAMIENTAS UTILIZADAS

Para la elaboración del proyecto se va hacer uso de la guía de auditoria la cual será

desarrollada y adaptada de acuerdo a las características del proceso de desarrollo de software que se lleva actualmente en la institución, las técnicas para recolectar información son las siguientes:

Documentos.

Un documento, es un soporte en papel o un elemento electrónico donde, existe una serie de datos incluidos en un orden determinado, sobre un tema específico, estos datos tienen un sentido simbólico, el cual puede ser extraído de ellos. Los documentos pueden tener un contenido expresivo (lo que dice) y un contenido instrumental (lo que motiva). (EA, 2015).

Revisar la información escrita del proceso de desarrollo de software como documentación, manuales o flujo gramas, con el fin de comprender las acciones realizadas en el proceso.

Cuestionario.

Conjunto de preguntas las cuales permiten recolectar información para identificar las características del proceso de desarrollo de software.

El cuestionario será contestado por los 4 ingenieros de la coordinación de desarrollo, con el fin de identificar las principales características del proceso.

(Ver anexo A).

Entrevistas.

Sostener una conversación con cada uno de los integrantes de la coordinación de desarrollo de

software con el fin conocer qué actividades desempeñan y como las realizan, permitiendo tener un punto de vista más real del proceso de desarrollo de software. Para está labor se utilizará un formato que contiene las preguntas a realizar (Ver anexo B).

Observación directa.

Permite recolectar datos por medio de la observación al objeto de estudio dentro del proceso de desarrollo de software.

Para aplicar esta técnica, la observación debe efectuarse de tal manera que las personas observadas son conscientes de estar siendo objetos de la técnica. La observación consiste en mirar el proceso de desarrollo, de tal forma que el auditor verifique si los procedimientos se están llevando a cabo correctamente. Para llevar a cabo el control de esta observación se debe diligenciar un formato que contenga las características de la observación (Ver anexo C).

POBLACIÓN Y MUESTRA

La población es un conjunto finito o infinito de elementos, a los que se les va a realizar alguna observación (Universidad e la Punta, s.f.), la población para este proyecto será la Escuela Colombiana de Ingeniera Julio Garavito.

La muestra es un subconjunto finito de una población (Universidad e la Punta, s.f.), para el desarrollo del presente proyecto se tomará como muestra el grupo de trabajo de la coordinación de desarrollo de software de la institución, compuesta por 4 personas.

Población.

Segmentación de la población

La población se encuentra conformada por un coordinador de desarrollo, un arquitecto de software y 2 ingenieros de desarrollo.

Muestra.

Método de la significancia.

Para determinar el tamaño adecuado de la muestra se hace uso del método probabilístico de significancia este método se usa cuando se conoce el tamaño de la población, aplicando la siguiente formula:

$$n = \frac{Z^2 * p * q * N}{e^2(N-1) + Z^2 * p * q}$$

En donde:

N = tamaño de la población

Z = nivel de confianza

p = probabilidad de éxito, o proporción esperada

q = probabilidad de fracaso (1-p)

e = Margen de error

Solución:

Tamaño de la población (N)= 4

Nivel de confianza = 95%

Margen de error e =1%

P = 0.5 por defecto

Q= 1-0.5 = 0.5

$$n = \frac{95^2 \times 0.5 \times 0.5 \times 4}{1^2\,(4-1) + 95^2 \times 0.5 \times 0.5}$$

n = 3.9946 ≈ 4

El tamaño de la muestra da como resultado 4

DIAGNÓSTICO DE LA MUESTRA

Se debe aplicar los instrumentos de recolección de información de las 4 personas que conforman la coordinación de desarrollo de software de la institución.

DESARROLLO DE LA PROPUESTA

DISEÑO.

Para el desarrollo de la propuesta se tienen contempladas las siguientes actividades para cada objetivo:

OBJETIVO 1.

- Contextualización de la Organización.
- Identificación de la plataforma tecnológica
- Descripción del proceso de desarrollo de software

OBJETIVO 2.

- Diseñar la guía de auditoria.

OBJETIVO 3.

- Documentar hallazgos.
- Documentar recomendaciones.

CONTEXTUALIZACIÓN DE LA ORGANIZACIÓN

Organización.

La Escuela Colombiana de Ingeniería Julio Garavito es una institución universitaria de carácter privado de Colombia, "sujeta a inspección y vigilancia por medio de la Ley 1740 de 2014 y la ley 30 de 1992 del Ministerio de Educación de Colombia. Fundada el 20 de octubre de 1972 y ubicada al Norte de Bogotá" (Wikipedia, 2018).

OBJETIVOS INSTITUCIONALES

La Escuela Colombiana de Ingeniería asume como Objetivos Institucionales:

"Contribuir al progreso personal, social y del conocimiento" (Escuela Colombiana de Ingeniería, 2008), a través de:

a) la formación integral de la persona, caracterizada por la alta preparación científica, tecnológica, técnica, ética, social y humanística;

b) la construcción y desarrollo de conocimiento, especialmente científico y tecnológico;

c) la interacción dinámica, real y permanente con el entorno.

Fortalecer la vivencia de los valores que a través de su historia se han hecho evidentes en todos los órdenes de la vida institucional y en sus egresados, en un ambiente propicio para el logro de su Misión.

Fortalecer una cultura académica, enmarcada en la excelencia, la creatividad y la

innovación.

Contextualizar la actividad académica en las necesidades del entorno y en los propósitos y oportunidades nacionales de desarrollo. (Escuela Colombiana de Ingeniería, 2008)

Naturaleza.

"La Escuela Colombiana de Ingeniería Julio Garavito. Es una universidad de carácter tecnológico" (Escuela Colombiana de Ingeniería, 2008).

Valores Institucionales.

"La Escuela Colombiana de Ingeniería practica valores como el respeto, el cumplimiento, la transparencia, la tolerancia, la honradez y la solidaridad, entre otros, que tienen su raíz en la Declaración de Principios" (Escuela Colombiana de Ingeniería, 2008).

"Como aspectos orientadores de su práctica académica y administrativa, la Escuela Colombiana de Ingeniería valora" (Escuela Colombiana de Ingeniería, 2008):

- El perfeccionamiento del ser humano.
- La búsqueda de la excelencia institucional.
- La autonomía con responsabilidad.
- La creatividad y la innovación orientadas especialmente al desarrollo del país.
- La vocación de servicio.
- La confianza mutua.
- La participación con compromiso y entusiasmo.

- Impulsa la vivencia de los valores en las personas y en los estamentos de la comunidad universitaria.

Misión.

La Escuela tiene como misión la formación de la persona, fundamentada en una alta preparación científica y tecnológica, armonizada con un profundo sentido de solidaridad social y un compromiso ético por parte de todos los miembros de la comunidad académica, para que su ejemplo constituya una lección de comportamiento ciudadano transmitida a la sociedad. La formación que se brinda alienta el espíritu de creatividad e innovación y se enmarca en el contexto de la realidad colombiana para que los egresados estén en capacidad de plantear soluciones autóctonas a los problemas nacionales e igualmente puedan desempeñarse con eficiencia en un mundo competitivo y globalizado. (Escuela Colombiana de Ingeniería, Mision)

Dentro del espíritu que inspiró a sus fundadores, desarrolla las funciones de docencia, investigación y proyección social en concordancia con las normas legales y de acuerdo con la evolución del conocimiento, el progreso científico y los avances en el campo de la educación. La Escuela es un escenario abierto a las diversas corrientes de pensamiento y mantiene independencia frente a todo credo político, racial, económico o religioso y, en consecuencia, es ajena a todo interés partidista surgido de tales credos. Para alcanzar sus objetivos, la Escuela cuenta con docentes de alto nivel académico cuya labor se refleja en la excelencia de los programas y sus egresados. (Escuela Colombiana de Ingeniería, Mision)

Como condición esencial para la convivencia ciudadana y la armonía con la naturaleza,

la Escuela propicia la formación integral de la persona y fomenta en ella una actitud de respeto por la dignidad humana y por su entorno, con la convicción de que los elementos de la biosfera hacen parte de una totalidad universal cuyo equilibrio es necesario para la conservación de los ecosistemas y de la vida sobre la Tierra. (Escuela Colombiana de Ingeniería, Mision).

Visión

La Escuela en su empeño por realizar el sueño de una sociedad mejor, cumplirá su misión con excelencia y alentará en forma permanente la participación activa de la comunidad académica en el estudio de la realidad colombiana, de tal manera que tenga un efecto multiplicador y contribuya a solucionar las necesidades básicas del país. (Escuela Colombiana de Ingeniería, 2008).

Los estudiantes de la Escuela serán el centro del proceso educativo y los docentes, sus guías y consultores. La formación científica y tecnológica estará complementada con una adecuada preparación humanística y un sólido conocimiento del entorno, lo cual les permitirá un mejor desempeño en los ámbitos nacional e internacional. (Escuela Colombiana de Ingeniería, 2008).

Así mismo, la Escuela contará con unidades de investigación especializada que se constituirán en centros de generación y difusión del conocimiento, y se transformará en una universidad con nuevos campos de acción, en respuesta a las necesidades de formación del país. (Escuela Colombiana de Ingeniería, 2008).

Estrategias

La Escuela Colombiana de Ingeniería Julio Garavito es una institución universitaria privada, organizada como corporación sin ánimo de lucro, de conformidad con la legislación colombiana, dedicada a la enseñanza de la ingeniería, la economía, la administración de empresas y las matemáticas; a la investigación y a la relación con el entorno a partir de actividades de extensión. (Escuela Colombiana de Ingeniería, 2008).

La institución en su calidad definió un plan de desarrollo institucional como herramienta de gestión y de cohesión que, a partir de propósitos comunes, permite plasmar los hitos de desarrollo de la Escuela en un tiempo establecido, lo cual lo convierte en la carta de navegación y principal instrumento de planeación en el que se expresan los ejes estratégicos, sus objetivos, programas y proyectos, de acuerdo con sus principios filosóficos.

El Plan de Desarrollo Institucional 2016-2025 contempla 7 (siete) ejes estratégicos, que funcionan como pilares fundamentales y que establecen la Escuela para el desarrollo institucional (denominados objetivos generales en el Plan de Desarrollo 2010-2020), presentados en la siguiente figura:

Articulación de ejes del plan de desarrollo

Fuente: Oficina de Desarrollo Institucional – Julio de 2016

A continuación, se presentan los líderes (por cargo) asociados a cada Eje estratégico.

N°	Eje estratégico	Líderes
1.	Formación de excelencia.	Vicerrector Académico
2.	Desarrollo de la investigación.	Director de I+i
3.	Fortalecimiento de la relación con el entorno.	Director de la Unidad de Gestión Externa
4.	Aseguramiento de la calidad.	Director de la Oficina de Desarrollo Institucional
5.	Desarrollo de la comunidad universitaria.	Director de Bienestar Universitario Director de Recursos Humanos Coordinador de Desarrollo Profesoral
6.	Eficiencia y sostenibilidad institucional.	Vicerrector Administrativo
7.	Infraestructura sostenible.	Vicerrector Administrativo *Director de Planta Física

Valores institucionales

"La Escuela Colombiana de Ingeniería práctica valores como el respeto, el cumplimiento, la transparencia, la tolerancia, la honradez y la solidaridad, entre otros, que tienen su raíz en la

Declaración de Principios" (Escuela Colombiana de Ingeniería, 2008).

Como aspectos orientadores de su práctica académica y administrativa, la Escuela Colombiana de Ingeniería valora:

- El perfeccionamiento del ser humano
- La búsqueda de la excelencia institucional
- La autonomía con responsabilidad
- La creatividad y la innovación orientadas especialmente al desarrollo del país
- La vocación de servicio
- La confianza mutua
- La participación con compromiso y entusiasmo
- Impulsa la vivencia de los valores en las personas y en los estamentos de la comunidad universitaria.

Procesos de la empresa.

La Escuela Colombiana de Ingeniería asume como Objetivos Institucionales:

- Contribuir al progreso personal, social y del conocimiento (Escuela Colombiana de Ingeniería, 2008), a través de:

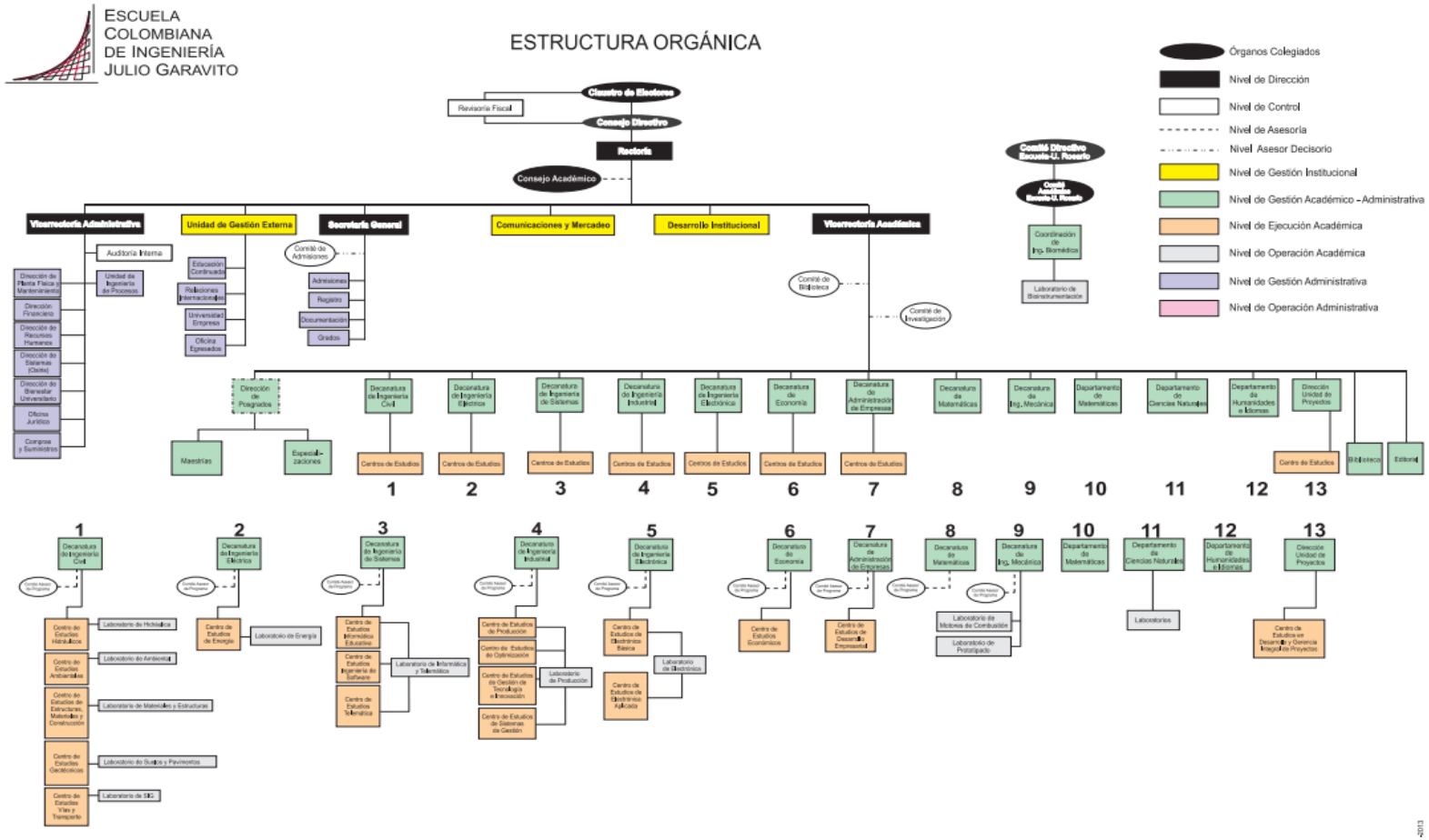
  a) La formación integral de la persona, caracterizada por la alta preparación científica, tecnológica, técnica, ética, social y humanística; b) la construcción y desarrollo de conocimiento, especialmente científico y tecnológico; y c) la interacción dinámica, real y permanente con el entorno.

- Fortalecer la vivencia de los valores que a través de su historia se han hecho evidentes en

todos los órdenes de la vida institucional y en sus egresados, en un ambiente propicio para el logro de su Misión.

- Fortalecer la vivencia de los valores que a través de su historia se han hecho evidentes en todos los órdenes de la vida institucional y en sus egresados, en un ambiente propicio para el logro de su Misión.
- Fortalecer la vivencia de los valores que a través de su historia se han hecho evidentes en todos los órdenes de la vida institucional y en sus egresados, en un ambiente propicio para el logro de su Misión.

Organigrama



IDENTIFICACIÓN DE LA PLATAFORMA TECNOLÓGICA

DESCRIPCIÓN DE LA ARQUITECTURA.

Hardware: Servidores

- Procesador Intel Xeon 5620
- Memoria RAM 32 Gb
- Almacenamiento en 2 HD de 2 Tb en RAID 1 (espejo)

Software:

- Sistema Operativo Linux (Fedora 17).
- Servidor de aplicaciones Glassfish 3.1.2
- DBMS SQL Server (Microsoft).
- Aplicativo web desarrollado en Java (Netbeans JDK 1.7).

Redes:

- Firewall check point
- Antivirus ESET
- Uso de VPN
- Sistema operativo de la red Windows server 2012 y Linux
- Protección conexión a internet por medio de Firewall, router y filtro de Paquetes

DIAGRAMA DE COMPONENTES

A continuación, se relaciona la topología de la red de la organización:

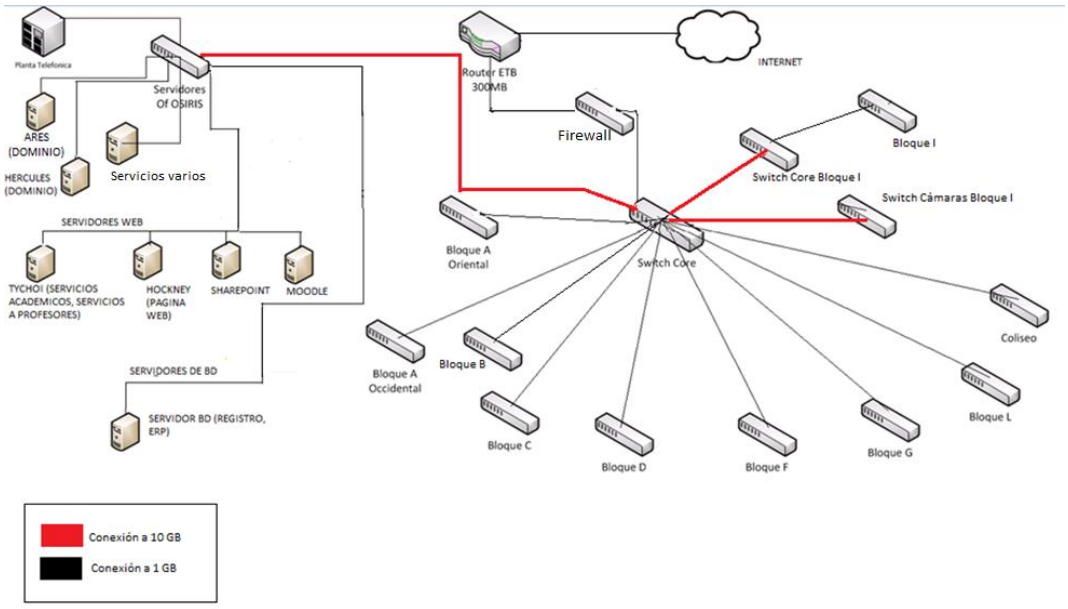


Ilustración 11 Topología de la red.

DESCRIPCIÓN DEL PROCESO DE DESARROLLO

Dentro de los objetivos de este trabajo se estableció definir el proceso de desarrollo de software llevado a cabo en la Escuela Colombiana de Ingeniería.

La metodología que se utilizó fue entrevistas personales con el grupo de desarrollo de software, los cuales conocen el funcionamiento actual del proceso. A través de estas entrevistas se definió el proceso de desarrollo de software el cual se detalla 6 etapas, descritas a continuación:

Diagrama general del proceso

El área de desarrollo de software de la Escuela Colombiana de Ingeniería Julio Garavito utiliza la metodología de desarrollo en cascada, la cual ordena las etapas de desarrollo de tal forma que el inicio de cada etapa debe esperar a la finalización de la etapa anterior. En la siguiente imagen se muestra el diagrama de desarrollo en cascada:

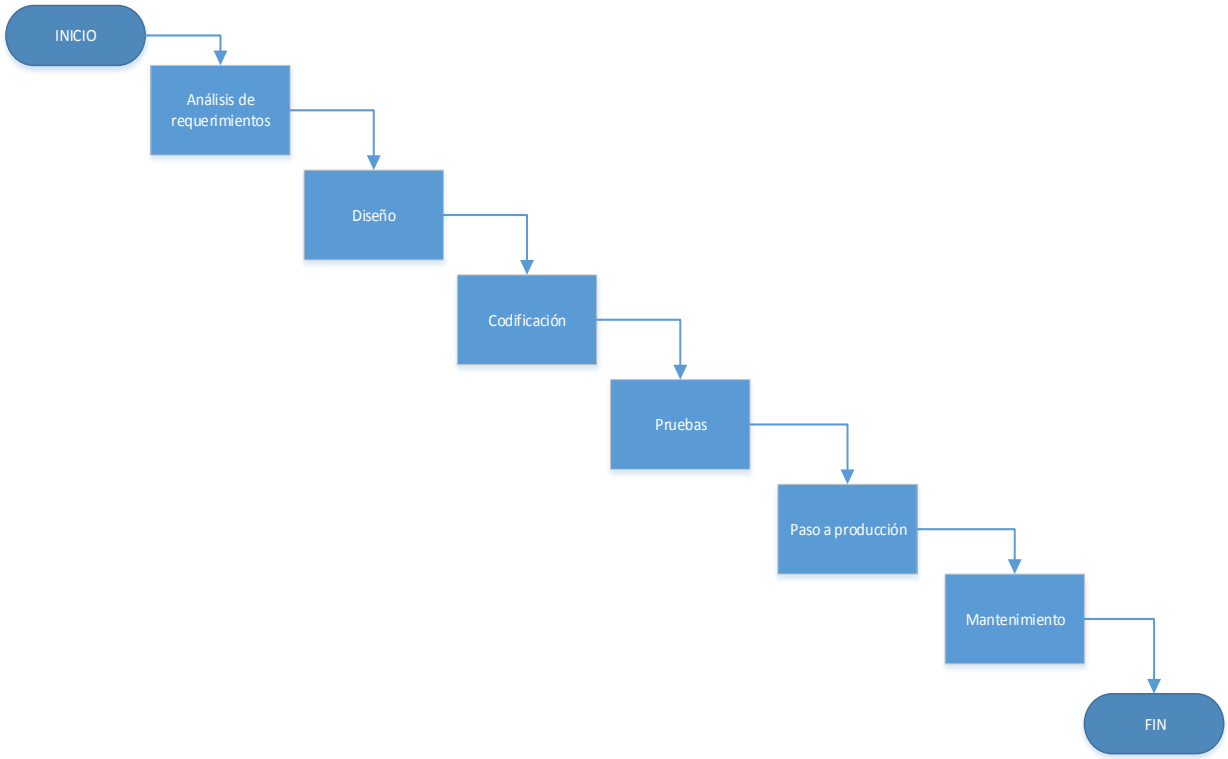


Ilustración 12 Proceso de desarrollo cascada

A continuación, se detalla el modelo actual del proceso de desarrollo de software:

Análisis de Requerimientos.

En esta actividad se analizan las necesidades del cliente referente al proyecto de tal forma que son verificadas y validadas por el cliente.

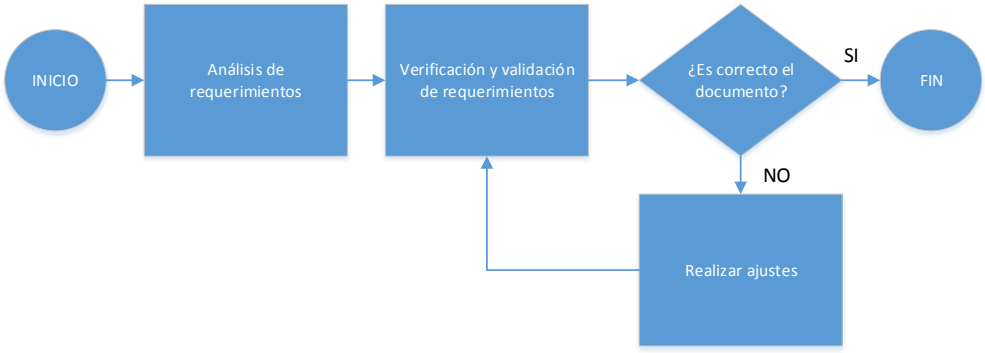


Ilustración 13 Etapa de análisis de requerimientos.

Diseño.

En esta etapa se genera, verifica y valida el diseño de la solución propuesta. De esta actividad se genera un documento que contiene características técnicas y es la guía de ejecución en el desarrollo.

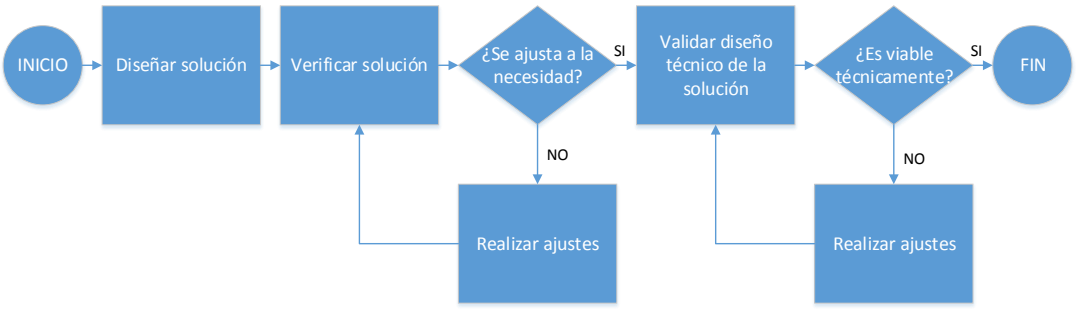


Ilustración 14 Etapa de diseño.

Codificación.

En esta actividad se lleva a cabo el desarrollo como tal de la solución e incluye varias tareas para validar el ambiente de desarrollo, coordinar tareas con los involucrados y realizar la

codificación. También se ejecutan casos de pruebas unitarias y se analizan posibles errores.

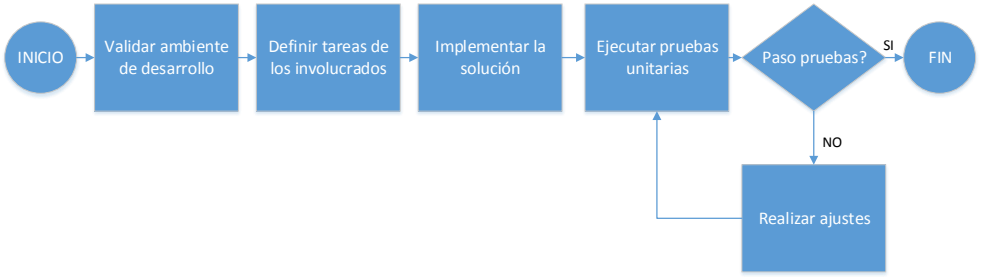


Ilustración 15 Etapa de codificación.

Pruebas.

En esta etapa se realizan pruebas con el usuario final, se planifican y ejecutan los casos de pruebas del software desarrollado.

Una vez ejecutada las pruebas, se verifica si el software pasa las pruebas, si no las pasa se deben corregir los errores y posteriormente volver a ejecutar las pruebas.

Si el software cumple con éxito las pruebas planificadas se finaliza la etapa de pruebas.

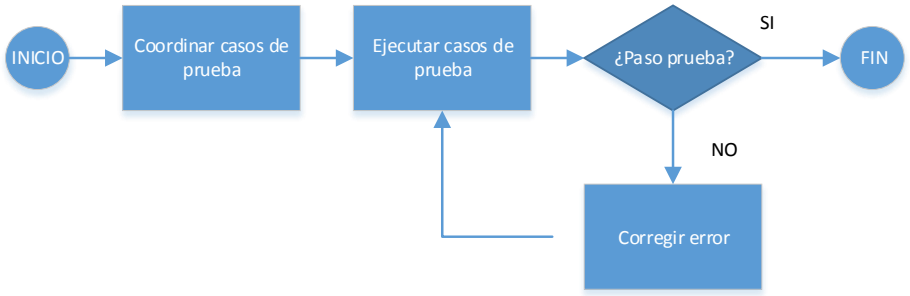


Ilustración 16 Etapa de pruebas

Paso a producción.

Una vez que el desarrollo está finalizado y el software completó satisfactoriamente el plan de pruebas, se realiza la planeación a paso producción del software, en donde se alista la base de

datos de producción con las nuevas tablas, y posteriormente se despliega en el servidor el ejecutable del aplicativo desarrollado de acuerdo a la fecha de salida a producción con el usuario funcional.

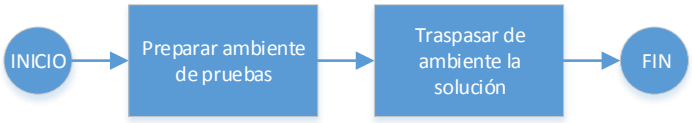


Ilustración 17 Etapa paso a producción

Mantenimiento.

En la última etapa del proceso de desarrollo de software se reciben los soportes enviados por los usuarios finales por medio de correo electrónico, se priorizan las solicitudes y se solucionan, finalmente se verifica que la solicitud se haya solucionado correctamente.

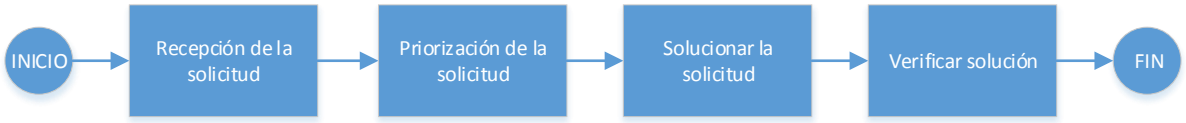
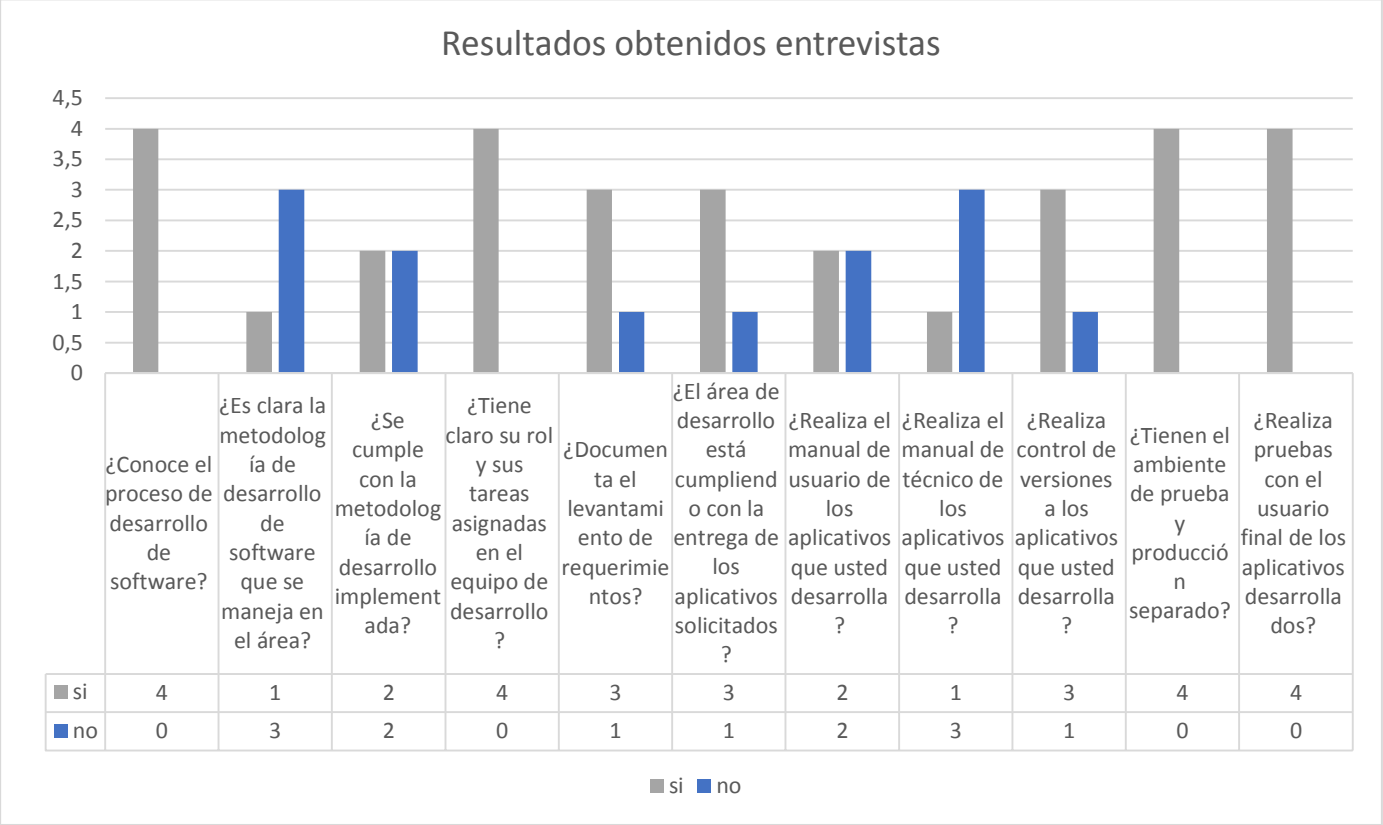


Ilustración 18 Etapa mantenimiento

Análisis De Resultados

Encuestas

La siguiente grafica muestra el resultado de las encuesta contestada por el equipo de desarrollo, la cual tiene como finalidad conocer el proceso de desarrollo de software para entender su funcionamiento actual. (ver anexo A)



	¿Conoce el proceso de desarrollo de software?	¿Es clara la metodología de desarrollo de software que se maneja en el área?	¿Se cumple con la metodología de desarrollo implementada?	¿Tiene claro su rol y sus tareas asignadas en el equipo de desarrollo?	¿Documenta el levantamiento de requerimientos?	¿El área de desarrollo está cumpliendo con la entrega de los aplicativos solicitados?	¿Realiza el manual de usuario de los aplicativos que usted desarrolla?	¿Realiza el manual de técnico de los aplicativos que usted desarrolla?	¿Realiza control de versiones a los aplicativos que usted desarrolla?	¿Tienen el ambiente de prueba y producción separado?	¿Realiza pruebas con el usuario final de los aplicativos desarrollados?
si	4	1	2	4	3	3	2	1	3	4	4
no	0	3	2	0	1	1	2	3	1	0	0

Ilustración 19 Encuesta al equipo de desarrollo

De la gráfica anterior se puede observar que el equipo de desarrollo difiere en un 64% frente a las siguientes preguntas:

- ¿Es clara la metodología de desarrollo de software que se maneja en el área?
- ¿Se cumple con la metodología de desarrollo implementada?
- ¿Documenta el levantamiento de requerimientos?
- ¿El área de desarrollo está cumpliendo con la entrega de los aplicativos solicitados?
- ¿Realiza el manual de usuario de los aplicativos que usted desarrolla?
- ¿Realiza el manual de técnico de los aplicativos que usted desarrolla?
- ¿Realiza control de versiones a los aplicativos que usted desarrolla?

El 36% restante tienen la misma respuestas, esto indica que el equipo de desarrollo no tiene clara la metodología utilizada en el desarrollo de software, no hay definido una guía para realizar la documentación de requerimientos, no hay lineamiento para la documentación de manuales técnicos y de usuario, finalmente no hay un buen control del manejo de versiones de los aplicativos desarrollados. Por lo cual se puede concluir que no hay claridad en las etapas de requerimientos, planeación y documentación del proceso de desarrollo de software.

A continuación, se analiza los resultados obtenidos de cada pregunta de la encuesta aplicada.

De la primera pregunta el 100% del equipo de desarrollo coincidieron con las respuestas, indicando que tienen claro el proceso de desarrollo de software, como se muestra en la siguiente gráfica:

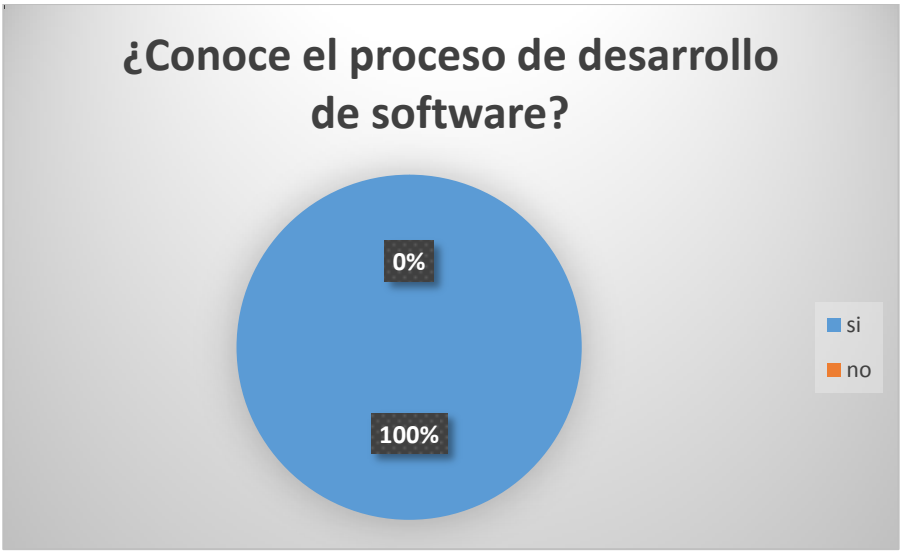


Ilustración 20 Resultados obtenidos encuesta – (Fuente propia, 2018)

El 75% del equipo de desarrollo de software no tiene claro la metodología de desarrollo que se maneja en el área, solo el 25% tiene claridad de la metodología, como se muestra en la siguiente gráfica:

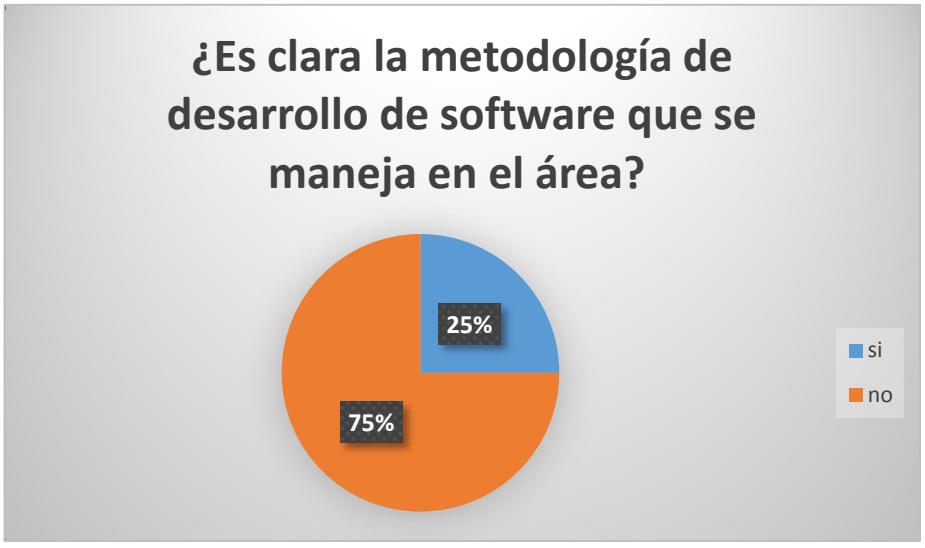


Ilustración 21 Resultados obtenidos encuesta – (Fuente propia, 2018)

El 50% del equipo de desarrollo de software no cumple con la metodología de desarrollo que maneja en el área, el otro 50% cumple con la metodología de desarrollo implementada, como se muestra en la siguiente gráfica:

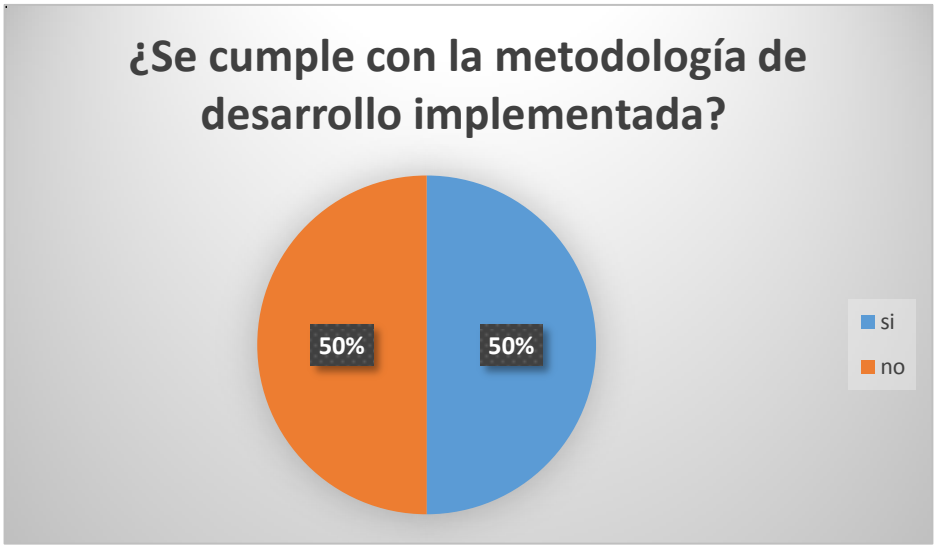


Ilustración 22 Resultados obtenidos encuesta – (Fuente propia, 2018)

El 100% del equipo de desarrollo de software tiene claro su rol y tareas asignadas para el desarrollo de software, como se muestra en la siguiente gráfica:

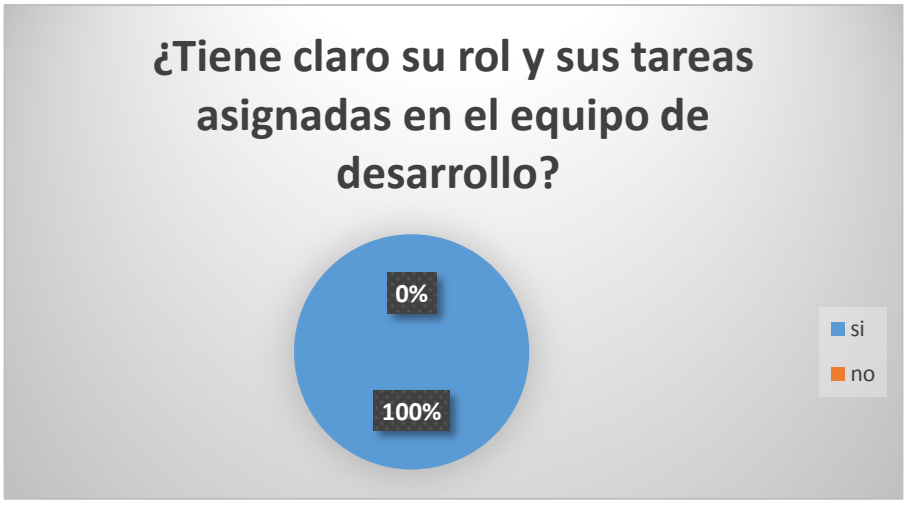


Ilustración 23 Resultados obtenidos encuesta – (Fuente propia, 2018)

El 75% del equipo de desarrollo de software documentan los requerimientos y solo el 25% no realiza la documentación, como se muestra en la siguiente gráfica:

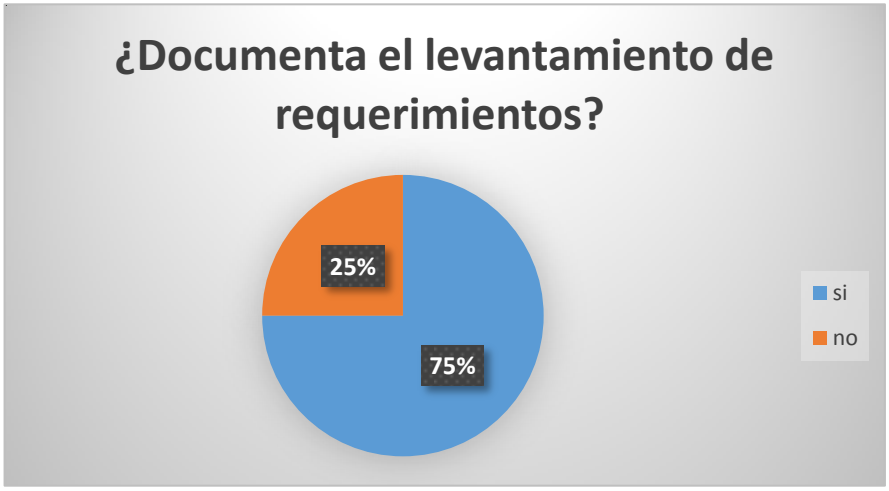


Ilustración 24 Resultados obtenidos encuesta – (Fuente propia, 2018)

El 75% del equipo de desarrollo de software cumple con la entrega de los aplicativos solicitados, solo el 25% está incumpliendo con las fechas de las entregas, como se muestra en la siguiente gráfica:

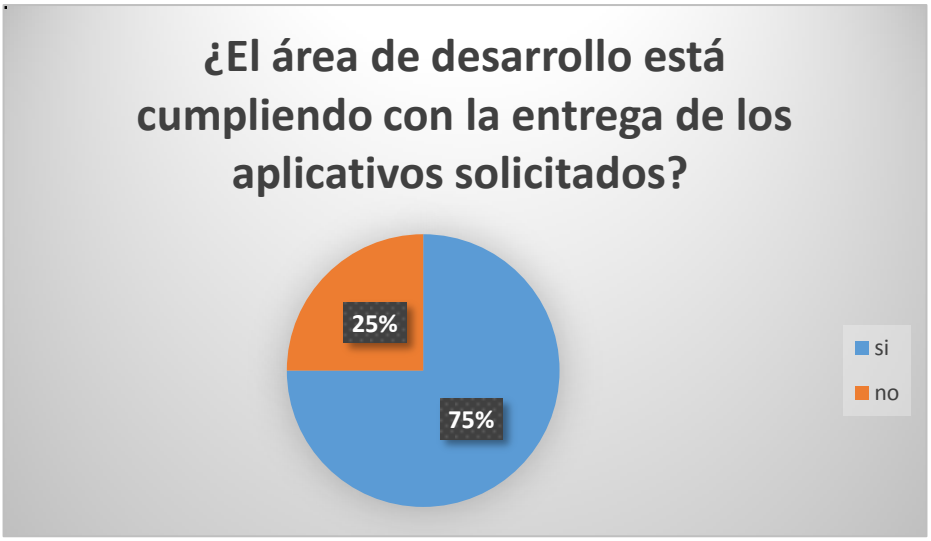


Ilustración 25 Resultados obtenidos encuesta – (Fuente propia, 2018)

El 50% del equipo de desarrollo de software no realizan el manual de usuario de los aplicativos desarrollados, el otro 50% realiza la documentación del manual de usuario, como se muestra en la siguiente gráfica:

Ilustración 26 Resultados obtenidos encuesta – (Fuente propia, 2018)

El 75% del equipo de desarrollo de software no realizan el manual técnico de los aplicativos desarrollados, el 25% realiza la documentación del manual técnico, como se muestra en la siguiente gráfica:

Ilustración 27 Resultados obtenidos encuesta – (Fuente propia, 2018)

El 25% del equipo de desarrollo de software no realizan control de versiones de los aplicativos desarrollados, el otro 75% si realiza el control de versiones de los aplicativos, como se muestra en la siguiente gráfica:

Ilustración 28 Resultados obtenidos encuesta – (Fuente propia, 2018)

El equipo de desarrollo de software tiene el ambiente de pruebas y de producción separados, como se muestra en la siguiente gráfica:

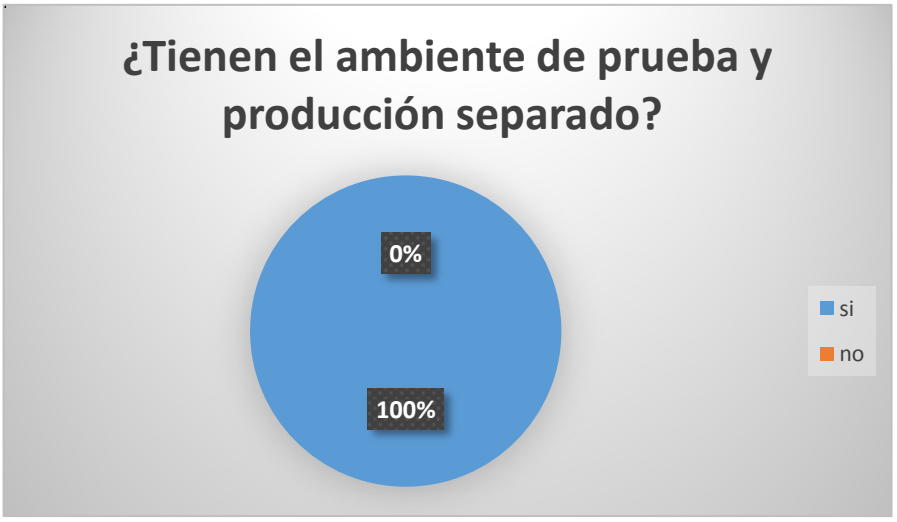


Ilustración 29 Resultados obtenidos encuesta – (Fuente propia, 2018)

El 100% del equipo de desarrollo de software realizan pruebas con el usuario final de los aplicativos desarrollados antes de salir a producción, como se muestra en la siguiente gráfica:

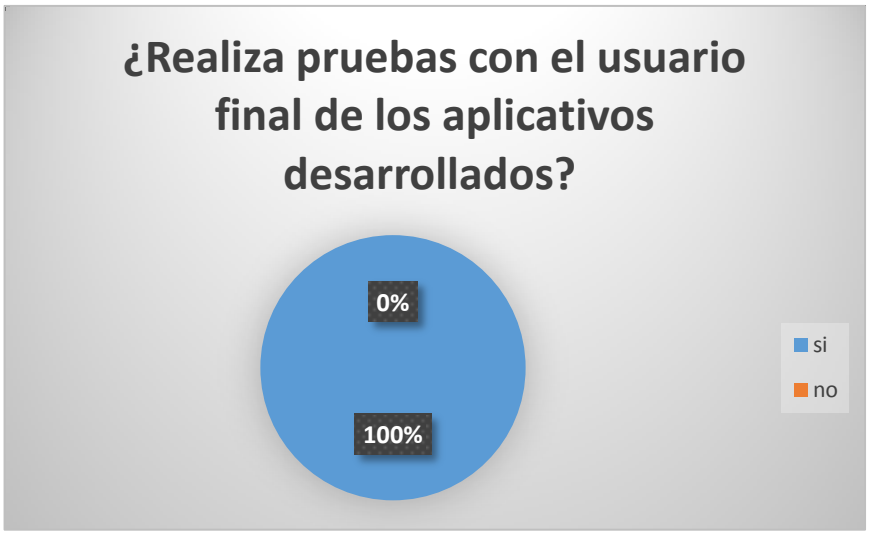


Ilustración 30 Resultados obtenidos encuesta – (Fuente propia, 2018)

GUÍA DE AUDITORIA

CUADRO DESCRIPTIVO DE LA NORMA ISO 27001

El siguiente cuadro muestra las características de la norma ISO 27001 aplicadas en la política de desarrollo seguro de la institución:

Tabla 1 Características ISO 27001

Objetivo	Finalidad	Ventajas	Desventajas
Proteger la confidencialidad, la integridad y la disponibilidad de la información en una organización.	Definir controles para garantizar la integridad y disponibilidad de la información	Reduce el riesgo de que se produzcan pérdidas de información en las organizaciones. Hace una revisión continua de los riesgos a los que están expuestos los clientes. Competitividad Calidad a la seguridad Concienciación y compromiso Normas y Estándares Visión externa y metódica del sistema Supervivencia de mercado	-No tiene retorno: Una vez que se ha empezado el camino de implementación de la norma ISO-27001, tenemos la opción de certificar o no. Sea cual fuere la elección, el cúmulo de actividades realizadas exige un mantenimiento y mejora continua, sino deja de ser un SGSI, y ello salta a la vista en el muy corto plazo. - Requiere esfuerzo continuo: Independientemente de las tareas periódicas que implica una vez lanzado el SGSI para los administradores del mismo, el mantenimiento del nivel alcanzado requerirá inexorablemente un esfuerzo continuado de toda la organización al completo.

Borghello, Cristian. (2018).

Realizando un análisis del cuadro anterior se deduce que la norma ISO 27001 permite a la institución dar seguridad a la información protegiendo su integridad, confidencialidad y disponibilidad por medio de controles definidos.

Los sistemas de información desarrollados por la institución deben contemplar requerimientos de seguridad, con las buenas prácticas para desarrollo seguro de aplicativos, así como con metodologías para la realizar pruebas de aceptación y seguridad al software desarrollado, reduciendo el riesgo de pérdida de información según lo definido en la norma ISO 27001.

El siguiente cuadro muestra los controles establecidos en la política de desarrollo seguro, indicando si el control aplica y por qué, al proceso de desarrollo:

Tabla 2Política de desarrollo seguro

CONTROL DEFINIDO EN EL MANUAL DE POLITICAS	¿APLICA?	JUSTIFICACIÓN
• La Oficina de Sistemas y Recursos Informáticos (Osiris) en conjunto con los propietarios de los aplicativos deben realizar las pruebas necesarias para asegurar que los sistemas de información desarrollados "cumplen con los requerimientos de seguridad establecidos antes del paso a producción" OSIRIS. (Abril.2018) .	SI	Este control afecta directamente las fases de desarrollo de pruebas y puesta en producción por lo que es de vital importancia verificar su cumplimiento.
• La Oficina de Sistemas y Recursos Informáticos (Osiris) en conjunto con los propietarios de los aplicativos "deben realizar las pruebas de los sistemas de información utilizando metodologías establecidas para este fin, documentado las pruebas realizadas y aprobando los pasos a producción" OSIRIS. (Abril.2018).	SI	El equipo de desarrollo es actualmente quien tiene la responsabilidad de implementar en una de las fases de desarrollo el uso de metodologías para pruebas, por lo que hace parte de nuestra auditoria.
• La Oficina de Sistemas y Recursos Informáticos (Osiris) en conjunto con los propietarios de los aplicativos deben realizar las pruebas por entrega de funcionalidades nuevas, por ajustes de funcionalidad o por cambios sobre la plataforma tecnológica en la cual funcionan	SI	La entrega de funcionalidades a usuarios hace parte de las funciones del equipo de desarrollo y es necesario evidenciar la forma en que se hace y si existe un control para su cumplimiento.

los aplicativos.		
• La Oficina de Sistemas y Recursos Informáticos (Osiris) en conjunto con los propietarios de los aplicativos "deben aprobar las migraciones entre los ambientes de desarrollo, pruebas y producción de sistemas de información nuevos y de cambios o nuevas funcionalidades" OSIRIS. (Abril.2018).	SI	Se requiere verificar el procedimiento de aprobación para el paso en los diferentes ambientes.
• La Oficina de Sistemas y Recursos Informáticos (Osiris) "debe implantar los controles necesarios para asegurar que las migraciones entre los ambientes de desarrollo, pruebas y producción han sido aprobadas, de acuerdo con el procedimiento de control de cambios" OSIRIS. (Abril.2018).	SI	Esta labor es afectada por el área de desarrollo en sus labores de paso a producción.
• La Oficina de Sistemas y Recursos Informáticos (Osiris) "debe contar con sistemas de control de versiones para administrar los cambios de los sistemas de información" OSIRIS. (Abril.2018).	SI	En el proceso de desarrollo de software es de gran impacto el contar con control de versiones por lo que aplica totalmente a la auditoria.
• La Oficina de Sistemas y Recursos Informáticos (Osiris) "debe asegurase que los sistemas de información adquiridos o desarrollados por terceros cuenten con un acuerdo de licenciamiento el cual debe especificar las condiciones de uso del software y los derechos de propiedad intelectual" OSIRIS. (Abril.2018).	NO	Dentro del alcance del proyecto no se tienen contemplados los controles a terceros.
• La Oficina de Sistemas y Recursos Informáticos (Osiris) "debe generar metodologías para la realización de pruebas al software desarrollado, que contengan pautas para la selección de escenarios, niveles, tipos, datos de pruebas y sugerencias de documentación" OSIRIS. (Abril.2018).	SI	De acuerdo a la metodología de desarrollo seleccionada se definen este tipo de controles por lo que este punto es importante analizarlo para identificar si se cubre total o parcialmente.
• La Oficina de Sistemas y Recursos Informáticos (Osiris) "debe asegurar que la plataforma tecnológica, las herramientas de desarrollo y los componentes de cada sistema de información estén actualizados	NO	Esta responsabilidad hace parte de la oficina de TI, sin embargo, no es responsabilidad del área de desarrollo.

con todos los parches generados para las versiones en uso y que estén ejecutando la última versión aprobada del sistema" OSIRIS. (Abril.2018).		
• La Oficina de Sistemas y Recursos Informáticos (Osiris) "debe incluir dentro del procedimiento y los controles de gestión de cambios el manejo de los cambios en el software aplicativo y los sistemas de información de la institución" OSIRIS. (Abril.2018).	NO	A pesar de ser parte importante en el proceso, se menciona para toda lo oficina de TI y dentro del listado de controles a evaluar existe uno que hace esta labor específica para desarrollo y será verificada.
• "Los desarrolladores de los sistemas de información deben considerar las buenas prácticas y lineamientos de desarrollo seguro durante el ciclo de vida de los mismos, pasando desde el diseño hasta la puesta en marcha" OSIRIS. (Abril.2018).	SI	Se debe comprobar el uso de buenas prácticas dentro de la metodología de desarrollo utilizada en el área.
• Los "desarrolladores deben proporcionar un nivel adecuado de soporte para solucionar los problemas que se presenten en el aplicativo; dicho soporte debe contemplar tiempos de respuesta aceptables" OSIRIS. (Abril.2018).	SI	Es necesario brindar asistencia a los usuarios si se tiene algún problema con los aplicativos.
• "Los desarrolladores deben construir los aplicativos de tal manera que efectúen las validaciones de datos de entrada y la generación de los datos de salida de manera confiable, utilizando rutinas de validación centralizadas y estandarizadas" OSIRIS. (Abril.2018).	SI	Se debe verificar que la información procesada y generada en el aplicativo sea correcta, con el fin de mostrar información precisa y confiable para la toma de decisiones.
• "Los desarrolladores deben asegurar que los sistemas de información construidos validen la información suministrada por los usuarios antes de procesarla" OSIRIS. (Abril.2018)., teniendo en cuenta aspectos como: – Tipos de datos – Rangos válidos – Longitud – Listas de caracteres aceptados – Caracteres considerados peligrosos – Caracteres de alteración de rutas.	SI	Es necesario verificar que los datos ingresados por el usuario al sistema sean válidos y cumplan con los tipos de datos especificados en el levantamiento de requerimientos, con el fin de que la información sea integra y confiable.
• "Los desarrolladores deben suministrar opciones de desconexión o cierre de sesión	SI	Se debe cerrar las sesiones de los usuarios conectados en el

de los aplicativos (logout) que permitan terminar completamente con la sesión o conexión asociada, las cuales deben encontrarse disponibles en todas las páginas protegidas por autenticación" OSIRIS. (Abril.2018).		sistema, con el fin de evitar suplantaciones y accesos no autorizados.
• "Los desarrolladores deben asegurar el manejo de operaciones sensibles o críticas en los aplicativos desarrollados permitiendo el uso de dispositivos adicionales como parámetros adicionales de verificación" OSIRIS (Abril.2018).	SI	Los aplicativos deben manejar de forma segura los procesos sensibles de la organización.
• "Los desarrolladores deben asegurar que los aplicativos proporcionen la mínima información de la sesión establecida, almacenada en cookies y complementos" OSIRIS (Abril.2018).	SI	Los aplicativos desarrollados no deben almacenar y recuperar información sensible de los usuarios.
• "Los desarrolladores deben garantizar que no se divulgue información confidencial en respuestas de error, incluyendo detalles del sistema, identificadores de sesión o información de las cuentas de usuarios; así mismo, deben implementar mensajes de error genéricos" OSIRIS (Abril.2018).	SI	Es necesario que cuando hayan Bug's en el sistema, el usuario no vea información sensible que maneje el sistema del servidor, podría afectar la disponibilidad y confidencialidad del mismo.
• Los "desarrolladores deben prevenir la revelación de la estructura de directorios de los sistemas de información construidos" OSIRIS (Abril.2018).	SI	El usuario que maneje el sistema desarrollado no debe saber la ruta del servidor donde se almacenan los archivos de descarga, se vería afectado la confidencialidad de la información.
• Los "desarrolladores deben remover información innecesaria en los encabezados de respuesta que se refieran a los sistemas operativos y versiones del software utilizado" OSIRIS (Abril.2018).	SI	Lo sistemas desarrollados no deben mostrar a los usuarios información de las características del servidor de aplicaciones, se vería afectado la confidencialidad de la información.
• "Los desarrolladores deben evitar incluir las cadenas de conexión a las bases de datos en el código de los aplicativos. Dichas cadenas de conexión deben estar en archivos de configuración independientes, los cuales se recomienda que estén	SI	Las credenciales de autenticación a la base de datos que usen los aplicativos no deben estar quemadas por código, puesto que serían vulnerables se vería afectado la

cifrados" OSIRIS (Abril.2018).		confidencialidad, integridad y disponibilidad de la información.
• Los desarrolladores deben certificar el cierre de la conexión a las bases de datos desde los aplicativos tan pronto como estas no sean requeridas. – Eliminar privilegios de ejecución a los archivos transferidos – Asegurar que dichos archivos sólo tengan privilegios de lectura.	SI	Si el aplicativo desarrollado realiza algún query a la base de datos, una vez finalizada la transacción se debe cerrar la conexión, con el fin de evitar demoras de procesamiento o inyección en la base de datos, se vería afectado la confidencialidad de la información.
• "Los desarrolladores deben proteger el código fuente de las aplicaciones construidas, de tal forma de que no pueda ser descargado ni modificado por los usuarios" OSIRIS (Abril.2018).	SI	A los ambientes de desarrollo solo deben tener acceso solo los programadores, se hace necesario que el código de los aplicativos esté de almacenado de forma segura, se vería afectado la confidencialidad e integridad de la información.

La siguiente guía permitirá verificar el cumplimiento de la política de desarrollo seguro establecida en el manual de políticas de información de la institución, la cual contiene la dependencia a la cual se le va aplica la guía, el proceso a auditar, el objetivo de la guía y las normas aplicables se utilizará una nomenclatura las para las pruebas de auditoria la cual es PA que significa Prueba de Auditoria seguido del número consecutivo de la prueba, ejemplo PA1:

GUIA DE AUDITORIA		
DEPENDENCIA: Oficina de sistemas y recursos informáticos.	**FECHA:**	
PROCESO: Desarrollo de software	**ELABORAD**	
PROCEDIMIENTO DE AUDITORÍA	**REF. P/T**	**POR**

OBJETIVOS. Evaluar el proceso de desarrollo de la Escuela Colombiana de Ingeniería Julio Garavito bajo los lineamientos de la política de seguridad de la institución basada en la norma ISO 27001.		
NORMATIVA APLICABLE. • Manual de políticas de seguridad de la información basada en la norma ISO 27001.		
1. La Oficina de Sistemas y Recursos Informáticos (Osiris) en conjunto con los propietarios de los aplicativos deben realizar las pruebas necesarias para asegurar que los sistemas de información desarrollados "cumplen con los requerimientos de seguridad establecidos antes del paso a producción" OSIRIS (Abril.2018).		
1.1 Seleccionar un aplicativo que se encuentre en desarrollo y comprobar mediante observación directa el proceso de pruebas realizadas por los desarrolladores antes del paso a producción.	PA1	
2. La Oficina de Sistemas y Recursos Informáticos (Osiris) en conjunto con los propietarios de los aplicativos "deben realizar las pruebas de los sistemas de información utilizando metodologías establecidas para este fin, documentado las pruebas realizadas y aprobando los pasos a producción" OSIRIS (Abril.2018)**..**		
2.1 Realizar encuesta a dos desarrolladores, donde puedan describir el proceso de pruebas realizado con los propietarios de los aplicativos.	PA2	
3. La Oficina de Sistemas y Recursos Informáticos (Osiris) en conjunto con los propietarios de los aplicativos deben realizar las pruebas "por entrega de funcionalidades nuevas, por ajustes de funcionalidad o por cambios sobre la plataforma tecnológica en la cual funcionan los aplicativos" OSIRIS (Abril.2018).		
3.1 Seleccionar un aplicativo con solicitud de cambios y hacer el acompañamiento para que mediante la técnica de observación directa sea documentado en los papeles de trabajo el proceso realizado por parte del desarrollador con el propietario del aplicativo.	PA3	
4. La Oficina de Sistemas y Recursos Informáticos (Osiris) en conjunto con los propietarios de los aplicativos "deben aprobar las migraciones entre los ambientes de desarrollo, pruebas y producción de sistemas de información nuevos y de cambios o nuevas funcionalidades" OSIRIS (Abril.2018).		
4.1 Mediante observación directa, identificar los procedimientos de aprobación para migración de ambiente de desarrollo, pruebas y producción.	PA4	
5. La Oficina de Sistemas y Recursos Informáticos (Osiris) "debe implantar los controles		

necesarios para asegurar que las migraciones entre los ambientes de desarrollo, pruebas y producción han sido aprobadas, de acuerdo con el procedimiento de control de cambios" OSIRIS (Abril.2018).		
5.1 Con un aplicativo que requiera modificaciones, realizar el seguimiento al de por la organización mediante Observación directa de la fase de pruebas y paso a producción.	PA5	
6. La Oficina de Sistemas y Recursos Informáticos (Osiris) "debe contar con sistemas de control de versiones para administrar los cambios de los sistemas de información" OSIRIS (Abril.2018)**.**		
6.1 Realizar una entrevista a un desarrollador, para verificar cuales son los controles implementados por el área en la migración entre ambientes de desarrollo, pruebas y producción.	PA6	
7. La Oficina de Sistemas y Recursos Informáticos (Osiris) "debe generar metodologías para la realización de pruebas al software desarrollado, que contengan pautas para la selección de escenarios, niveles, tipos, datos de pruebas y sugerencias de documentación" OSIRIS (Abril.2018).		
7.1 Seleccionar un aplicativo en desarrollo y comprobar las fases de desarrollo respecto a la metodológica utilizada por el área mediante la técnica de observación directa.	PA7	
8. "Los desarrolladores de los sistemas de información deben considerar las buenas prácticas y lineamientos de desarrollo seguro durante el ciclo de vida de los mismos, pasando desde el diseño hasta la puesta en marcha" OSIRIS (Abril.2018).		
8.1 Realizar una encuesta a dos desarrolladores donde se verifiquen buenas prácticas y lineamientos de la organización en cada fase de desarrollo.	PA8	
9. "Los desarrolladores deben proporcionar un nivel adecuado de soporte para solucionar los problemas que se presenten en el aplicativo; dicho soporte debe contemplar tiempos de respuesta aceptables" OSIRIS (Abril.2018).		
9.1 Identificar los acuerdos de nivel de servicio del área de desarrollo y comprobar si están documentados.	PA9	
10. "Los desarrolladores deben proporcionar un nivel adecuado de soporte para solucionar los problemas que se presenten en el aplicativo; dicho soporte debe contemplar tiempos de respuesta aceptables" OSIRIS (Abril.2018).		
10.1 Verificar tiempos de respuesta para dos soportes que lleguen al momento de realizar la prueba.	PA10	
11. "**Los desarrolladores deben construir los aplicativos de tal manera que efectúen las validaciones de datos de entrada y la generación de los datos de salida de manera**		

confiable, utilizando rutinas de validación centralizadas y estandarizadas" OSIRIS (Abril.2018).		
11.1 Seleccionar un aplicativo desarrollado por el área y comprobar en un servidor de pruebas la validación de formularios en campos de texto, números, fecha y correo electrónico, para luego confirmar que sean acordes al modelo de datos de la organización.	PA11	
12. "**Los desarrolladores deben asegurar que los sistemas de información construidos validen la información suministrada por los usuarios antes de procesarla"** OSIRIS (Abril.2018)**.**		
12.1 Seleccionar un aplicativo desarrollado por el área y comprobar que el dato que se ingresó en el aplicativo corresponda al mismo tipo de datos que esté definido en el modelo de base de datos.	PA12	
13. "Los desarrolladores deben suministrar opciones de desconexión o cierre de sesión de los aplicativos (logout) que permitan terminar completamente con la sesión o conexión asociada, las cuales deben encontrarse disponibles en todas las páginas protegidas por autenticación" OSIRIS (Abril.2018).		
13.1 ingresar con un usuario de prueba a una aplicación desarrollada por el área, posteriormente duplicar en otra página del navegador el mismo aplicativo con la misma sesión con la que se autentico. Una vez estén las dos páginas con el mismo usuario autenticado, cerrar sesión en una de las páginas y verificar en la otra página que se halla cerrado sesión también.	PA13	
14. "Los desarrolladores deben asegurar el manejo de operaciones sensibles o críticas en los aplicativos desarrollados permitiendo el uso de dispositivos adicionales como parámetros adicionales de verificación" OSIRIS (Abril.2018).		
14.1 Seleccionar un aplicativo desarrollado por el área y verificar la autenticación con dispositivos de seguridad adicionales que utilice la institución.	PA14	
15. "Los desarrolladores deben asegurar que los aplicativos proporcionen la mínima información de la sesión establecida, almacenada en cookies y complementos" OSIRIS (Abril.2018).		
15.1 abrir sesión con un usuario de prueba en uno de los aplicativos desarrollados por el área y verificar si el aplicativo solicita de forma clara al usuario su consentimiento al uso de cookies si esta lo requiere.	PA15	
16. "Los desarrolladores deben garantizar que no se divulgue información confidencial en respuestas de error, incluyendo detalles del sistema, identificadores de sesión o información de las cuentas de usuarios; así mismo, deben implementar mensajes de error genéricos" OSIRIS (Abril.2018).		
16.1 Solicitar acceso a una aplicación en un ambiente de pruebas controlado y con ayuda de un desarrollador simular un error en el	PA16	

sistema. Posteriormente verificar si la información que muestra el error es sensible y confidencial para la institución.		
17. "Los desarrolladores deben prevenir la revelación de la estructura de directorios de los sistemas de información construidos" OSIRIS (Abril.2018).		
17.1 Seleccionar un aplicativo desarrollado por el área que permita descargar archivos, después descargar algún documento y comprobar que en el proceso de descarga no se muestra información del directorio del servidor donde se aloja el documento descargado.	PA17	
18. "Los desarrolladores deben remover información innecesaria en los encabezados de respuesta que se refieran a los sistemas operativos y versiones del software utilizado" OSIRIS (Abril.2018).		
18.1 seleccionar un aplicativo desarrollado por el área abrir sesión y por medio del navegador verificar que los encabezados de respuesta no muestren información sensible como el sistema operativo y versiones de software usado.	PA18	
19. "Los desarrolladores deben evitar incluir las cadenas de conexión a las bases de datos en el código de los aplicativos. Dichas cadenas de conexión deben estar en archivos de configuración independientes, los cuales se recomienda que estén cifrados" OSIRIS (Abril.2018).		
19.1 Verificar donde se encuentran las credenciales de conexión con la base de datos de los aplicativos, las cuales no deben estar quemadas en el código fuente de las aplicaciones y deben estar cifradas.	PA19	
20. "Los desarrolladores deben certificar el cierre de la conexión a las bases de datos desde los aplicativos tan pronto como estas no sean requeridas" OSIRIS (Abril.2018).		
20.1 Seleccionar un aplicativo desarrollado por el área y con acompañamiento de un desarrollador verificar en el código fuente de la aplicación que las conexiones de la base de datos se cierren por cada transacción.	PA20	
21. "Los desarrolladores deben proteger el código fuente de las aplicaciones construidas, de tal forma de que no pueda ser descargado ni modificado por los usuarios" OSIRIS (Abril.2018).		
21.1 verificar en que equipos se encuentran alojados el código fuente de las aplicaciones desarrolladas por el área, y comprobar que usuarios tienen acceso a esos equipos.	PA21	

APLICAR GUÍA DE AUDITORIA

Prueba De Auditoria PA1

Formato Observación Directa

Fecha:	01/11/2018
Ubicación:	Oficina de sistemas y recursos informáticos
Que se observa:	Proceso de pruebas unitarias
Situación observada y contexto:	Proceso de entrega de desarrollos
Tiempo de observación:	1 Horas
Observador:	Juan David Vanzina Solis
Observado:	Leonardo Salazar Bogotá

Hora	Descripción de la actividad	Interpretación
3.00 pm 3.20 pm	El desarrollador procede solicitar al DBA el ajuste a su ambiente de pruebas. Allí solicita generar un Backup de la base de datos de producción para que sea cargado en pruebas. Ahora procese a alistar su equipo local para que el aplicativo apunte a la base de datos de pruebas.	El proceso de generación y carga al ambiente de pruebas es relativamente rápido, no tarda más de 15 minutos. El desarrollador cambia la IP donde se encuentra la base de datos, pero no se cuenta con una infraestructura

	Leonardo procede a diseñar casos de prueba, según el requerimiento del propietario Procede a ejecutar cada caso preparado Termina pruebas exitosamente.	similar a la de producción. Las pruebas no son exhaustiva.

Prueba De Auditoria PA2.

Para esta prueba se decidió realizar una encuesta a dos desarrolladores, donde describen el proceso realizado en la etapa de pruebas con los propietarios de los aplicativos. A continuación se listan las preguntas realizadas:

1. ¿Realiza una planeación de pruebas?
2. ¿Hace el diseño las pruebas en colaboración con los propietarios de los sistemas?
3. ¿Implementa las pruebas de forma automatizada o manual?
4. ¿Establece criterios necesarios para evaluar la salida del sistema?
5. ¿Documenta las pruebas realizadas?
6. ¿Mediante actas hace entrega formal del proceso?

Prueba De Auditoria PA3.

Formato Observación Directa

Fecha:	01/11/2018
Ubicación:	Oficina de sistemas y recursos informáticos
Que se observa:	Proceso de pruebas y paso a producción
Situación observada y contexto:	Proceso de entrega de desarrollos
Tiempo de observación:	30 minutos
Observador:	Juan David Vanzina Solis
Observado:	Andrés Felipe Rojas

Hora	Descripción de la actividad	Interpretación
2.30 pm	El Usuario se acerca la oficina del desarrollador para la reunión de entrega de requerimientos solicitados.	El proceso de entrega no se comunica formalmente, se hace por teléfono.
2.40 pm	El desarrollador procede a presentar las nuevas funcionalidades al usuario.	
3.00 pm	El usuario acepta verbalmente que se cumple con los requisitos luego de ejecutar unas pruebas puntuales del propietario del software.	No se cuenta con un acta de entrega, en donde se diligencie cada aceptación por parte del propietario del software.

Prueba De Auditoria PA4

Para la verificación del proceso se evidencia que no existe documentación de migración entre ambientes llevado a cabo por el área de desarrollo, por lo que se decidió realizar observación directa en el proceso de pruebas y paso a producción y se identificó lo siguiente:

Formato Observación Directa

Fecha:	29/10/2018
Ubicación:	Oficina de sistemas y recursos informáticos
Que se observa:	Proceso de pruebas y paso a producción
Situación observada y contexto:	Realización de pruebas unitarias y paso a producción
Tiempo de observación:	2 horas
Observador:	Juan David Vanzina
Observado:	Andrés Felipe Rojas

Hora	Descripción de la actividad	Interpretación

9 am	El desarrollador procede solicitar al DBA el ajuste a su ambiente de pruebas. Allí solicita generar un Backup de la base de datos de producción para que sea cargado en pruebas.	El proceso de generación y carga al ambiente de pruebas es relativamente rápido, no tarda más de 15 minutos.
9.35 am	Posteriormente alista su equipo local para que el aplicativo apunte a la base de datos de pruebas.	El desarrollador cambia la IP donde se encuentra la base de datos, pero no se cuenta con una infraestructura similar a la de producción.
10 am	Ahora el desarrollador plantea casos de uso en donde identifica las variables de entrada y de salida.	Los casos pueden abarcar gran parte de la necesidad, pero no se tiene apoyo del usuario funciona para confirmar esol.
10:20 am	Procede a ejecutar cada caso preparado	El desarrollador no hace ningún tipo de documentación donde se verifique el proceso realizado.
10:40 am	El desarrollador se da cuenta que en uno de	

	los casos de uso debe hacer modificaciones porque no le funciono.	A pesar de realizar cambios en el código, él continúa sus pruebas en vez de iniciar nuevamente todo el proceso.
10:45 am	Realiza las modificaciones y continúa con los demás casos propuestos.	
10:55 am		
11:02 am	Termina pruebas.	La seguridad en el paso a producción está limitada únicamente a un usuario y contraseña del servidor.
	Realiza el paso a producción mediante una aplicación de FTP	Tampoco se cuenta con usuarios separados, es un único usuario para el área de desarrollo.

Prueba De Auditoria PA5

Para la verificación del proceso se evidencia que no existe documentación de migración entre ambientes llevado a cabo por el área de desarrollo, por lo que se decidió realizar observación directa en el proceso de pruebas y paso a producción y se identificó lo siguiente:

Formato Observación Directa

Fecha:	28/10/2018
Ubicación:	Oficina de sistemas y recursos informáticos
Que se observa:	Proceso de pruebas y paso a producción
Situación observada y contexto:	Realización de pruebas unitarias y paso a producción
Tiempo de observación:	2 horas
Observador:	Juan David Vanzina
Observado:	Andrés Felipe Rojas

Hora	Descripción de la actividad	Interpretación
9 am 9.35 am	El desarrollador procede solicitar al DBA el ajuste a su ambiente de pruebas. Allí solicita generar un Backup de la base de datos de producción para que sea cargado en pruebas. Posteriormente alista su equipo local para que el aplicativo apunte a la base de datos de pruebas.	El proceso de generación y carga al ambiente de pruebas es relativamente rápido, no tarda más de 15 minutos. El desarrollador cambia la IP donde se encuentra la base de datos, pero no se cuenta con una infraestructura similar a la de producción. Los casos pueden abarcar gran parte de la necesidad, pero no se tiene apoyo del

		usuario funciona para confirmar esol.
10 am	Ahora el desarrollador plantea casos de uso en donde identifica las variables de entrada y de salida.	El desarrollador no hace ningún tipo de documentación donde se verifique el proceso realizado.
10:20 am	Procede a ejecutar cada caso preparado	
10:40 am		
10:45 am	El desarrollador se da cuenta que en uno de los casos de uso debe hacer modificaciones porque no le funciono.	A pesar de realizar cambios en el código, él continúa sus pruebas en vez de iniciar nuevamente todo el proceso.
	Realiza las modificaciones y continúa con los demás casos propuestos.	
10:55 am		La seguridad en el paso a producción está limitada únicamente a un usuario y contraseña del servidor.
11:02 am	Termina pruebas.	
	Realiza el paso a producción mediante una	Tampoco se cuenta con usuarios separados, es un

	aplicación de FTP	único usuario para el área de desarrollo.

Prueba De Auditoria PA6

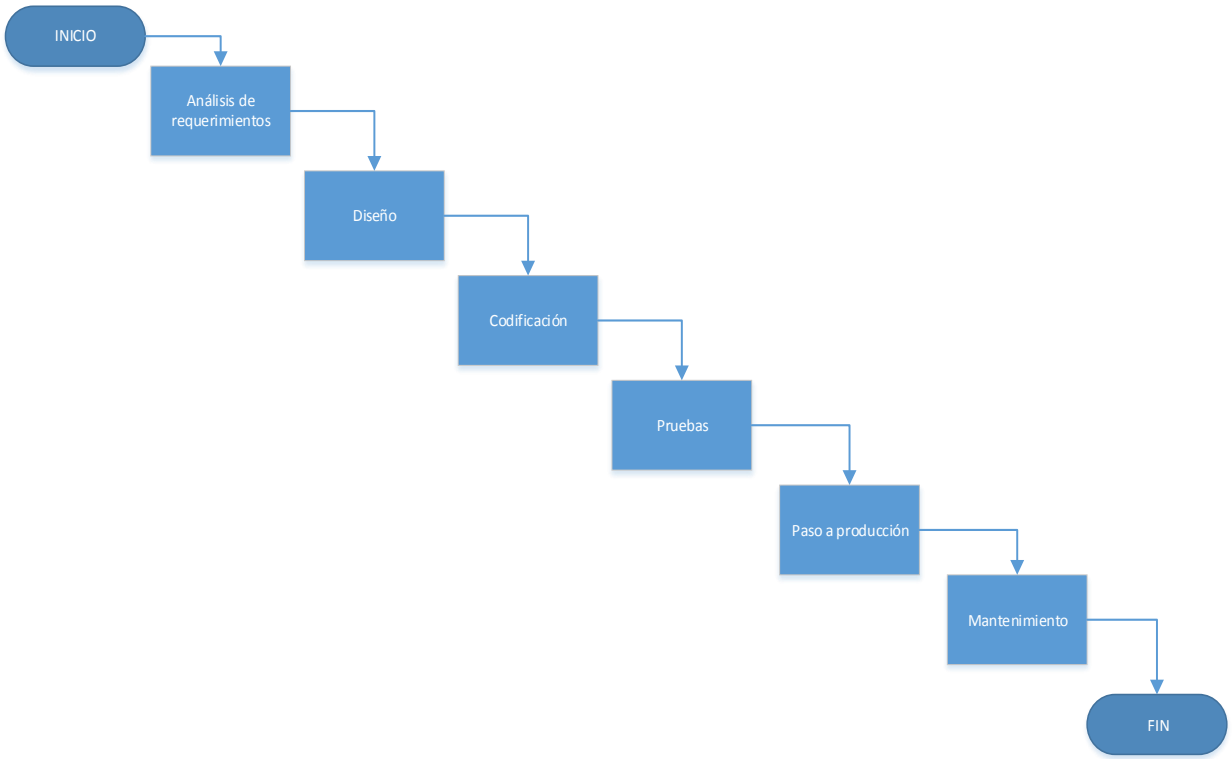
Se realiza una entrevista al desarrollador, quien nos indica que no existe un proceso de control de cambios formalizado y que solo cuenta con una carpeta compartida en donde semanalmente realizan el Backup de los contenidos generados. Adicional a esto él indica que esa información es respaldada semanalmente por el área de infraestructura y que la información no se borra en ningún momento. Finalmente comenta que el acceso a dicha carpeta es restringido bajo la administración del área de infraestructura.

A continuación, se listan las preguntas de la entrevista:

7. ¿De qué se hace copia de seguridad?
8. ¿Cada cuánto hace las copias de seguridad?
9. ¿Qué tipo de copias de seguridad hace (completa, diferencial o incremental)?
10. ¿Por cuánto tiempo conserva las copias de seguridad?
11. ¿Comprueba que pueden ser restauradas periódicamente?
12. ¿Las copias que realiza, tienen algún tipo de cifrado?
13. ¿En cuántos y cuales medios almacena sus copias de seguridad?
14. ¿Dónde almacena los medios?
15. ¿Hace copias de seguridad en la nube? ¿Qué precauciones toma?
16. ¿Documenta el proceso donde deja evidencia de la tarea realizada?

Prueba De Auditoria PA7

La institución no cuenta con documentación sobre la metodología de desarrollo implementada y los procesos tienden a ser muy empíricos, sin embargo, se realiza el levantamiento de información que resulta en las siguientes fases de desarrollo.



Este proceso contiene mayor detalle en la sección anterior de familiarización. Durante el proceso de levantamiento de información se evidencian falencias en segregación de funciones lo que genera alta criticidad en fase de pruebas y en fase paso a producción.

Prueba De Auditoria PA8.

En este punto se evidencio que no existe un documento con lineamientos para el desarrollo de

software diferente a la política de seguridad, por lo que se optó por realizar una encuesta para verificar la adopción de buenas prácticas en desarrollo que se muestra en el anexo b.

Prueba De Auditoria PA9.

Para esta prueba se videncia que no se tienen documentado los ANS (acuerdos de nivel de servicio), por lo que se decidió identificar los tiempos de respuesta a incidentes mediante una entrevista con los desarrolladores, quienes indican lo siguiente:

Disponibilidad de servicio

Se establece una disponibilidad en días hábiles de la semana por ocho horas cada día en horario de 8 am a 5 pm. Los usuarios deben reportar a través de los medios dispuestos cualquier evento que consideren que les impida el acceso a los sistemas de información ofrecidos por el área de desarrollo; Esto es validado directamente por el desarrollador responsable para determinar la prioridad. En el caso en que se requieran cambios sobre los procesos en cualquiera de los sistemas de información la solicitud debe ser enviada por algún integrante de la oficina propietaria y será tramitada por aparte.

Canales de atención.

- Atención presencial, la oficina de desarrollo está disponible para atender a los usuarios

en la única sede de la institución ubicada en Bogotá - Colombia en la AK.45 No.205-59 (Autopista Norte) bajo la disponibilidad de servicio mencionada en el punto anterior.

- Atención telefónica, la institución cuenta con la línea telefónica 6683600 en Bogotá - Colombia y los desarrolladores cuentan con dos extensiones telefónicas para recibir incidencias dentro de la institución o fuera de ella, las extensiones son la 272 y la 568.
- Correo electrónico, cada desarrollador tiene su propia cuenta de correo electrónico y desde allí gestiona las solicitudes.

Tiempos de respuesta

Según la frecuencia e impacto del incidente reportado, los desarrolladores disponen de los siguientes tiempos de respuesta y prioridad de la solicitud como se muestra en las siguientes tablas:

		FRECUENCIA			
		1 (Poco Frecuente)	2 (Frecuencia normal)	3 (Frecuente)	4 (Muy Frecuente)
IMPACTO	5 (Extremo)	0- 24 Horas	0 - 16 Horas	0 - 8 Horas	0 - 8 Horas
	4 (Mayor)	0 - 48 Horas	0 - 24 Horas	0 - 16 Horas	0 - 8 Horas
	3 (Moderado)	0 - 48 Horas	0 - 24 Horas	0 - 16 Horas	0 - 16 Horas
	2 (Menor)	0 - 72 Horas	0 - 48 Horas	0 - 24 Horas	0 - 24 Horas
	1 (Insignificante)	0 - 72 Horas	0 - 72 Horas	0 - 72 Horas	0 - 72 Horas

Dadas estas condiciones se definen las prioridades y tiempos de cada prioridad:

Prioridad	
Baja	0 - 72 Horas
Media	0- 24 Horas
Alta	0 - 8 Horas

Se aclara que los tiempos para solución a incidencias están sujetos a la disponibilidad de servicio definida anteriormente.

Prueba De Auditoria PA10

Para esta prueba se realiza la evaluación a dos solicitudes de soporte vía correo electrónico, en donde se evidencian los tiempos de respuesta y la valoración dada por el desarrollador como se muestra a continuación:

- El soporte 1 fue clasificado por el desarrollador con prioridad alta. Fue solicitado el día 26 de octubre de 2018 a las 10:10 am y fue tramitado el día 26 de octubre de 2018 a las 11:12 am lo cual cumple con los tiempos mencionados por el equipo de desarrollo.

- El soporte 2 fue clasificado por el desarrollador con prioridad media. Fue solicitado el día viernes, 26 de octubre de 2018 a las 4:23 pm y fue tramitado el día lunes, 29 de octubre de 2018 a las 3:46 pm lo cual se ajusta muy bien a los tiempos estipulados.

A continuación, se presenta en imágenes la evidencia de la prueba:

Soporte 1

De: PAOLA ANDREA PERDOMO MORENO
Enviado el: viernes, 26 de octubre de 2018 10:10 a.m.
Para: NICOLAS ALMANZAR ESPITIA <nicolas.almanzar@escuelaing.edu.co>
CC: JIMMY IGNACIO RUIZ VILLATE <jimmy.ruiz@escuelaing.edu.co>
Asunto: Solicitud de adición Id_sal a informe

Apreciado Nicolás, cordial saludo.

Te pido por favor tu colaboración adicionando al informe de la librería INF_GEN – horario de asignaturas el Id_sal de los salones asignados en los horarios de clase.

Quedo pendiente de tus comentarios, muchas gracias Nicolas.

Cordialmente,

Eco. Paola Andrea Perdomo Moreno. MSc
Líder en programación académica
Equipo para horarios y recursos
Proyecto Enlace
paola.perdomo@escuelaing.edu.co
Escuela Colombiana de Ingeniería Julio Garavito
Autopista Norte AK 45 No. 205-59
PBX: (57-1) 6683600 Ext.
Bogotá, D.C., Colombia
www.escuelaing.edu.co

Ilustración 31(Escuela Ing Julio Garavito. 2018. evidencia soporte)

De: NICOLAS ALMANZAR ESPITIA
Enviado el: viernes, 26 de octubre de 2018 11:12 a. m.
Para: PAOLA ANDREA PERDOMO MORENO <paola.perdomo@escuelaing.edu.co>
CC: JIMMY IGNACIO RUIZ VILLATE <jimmy.ruiz@escuelaing.edu.co>
Asunto: RE: Solicitud de adición Id_sal a informe

Buen día Paola,

Ya se adiciono el campo id_sal al informe solicitado, por favor verificar.

Cordialmente,

ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

Ing. Nicolás Almanzar Espitia
Osiris
Oficina de Sistemas y Recursos Informáticos
nicolas.almanzar@escuelaing.edu.co
Escuela Colombiana de Ingeniería Julio Garavito
Autopista Norte AK 45 No. 205-59
Bogotá, D.C., Colombia
www.escuelaing.edu.co

Ilustración 32 (Escuela Ing Julio Garavito. 2018. evidencia soporte)

Soporte 2.

-----Mensaje original-----
De: RODRIGUEZ MORA LINA MARIA
Enviado el: viernes, 26 de octubre de 2018 4:23 p. m.
Para: ANDRES FELIPE ROJAS ORTIZ <andres.rojas@escuelaing.edu.co>
Asunto: Eciciencia

Buenas tardes, quería comentarte sobre un problema que tuve a la hora de inscribir mi grupo de eciciencia, con otro compañero estamos inscritos en el concurso de el tornillo de arquímedes, pero cuando nos inscribimos se crearon dos grupos, la cuestión es que me acerque a la oficina de eciciencia donde me dijeron que ellos no podían solucionar nada, pues que la base de datos ya no la podían modificar, me dijeron que fuera a donde el decano de Ingeniería civil, el cual fui y el me dijo que le enviara una carta, la realice pero cuando fui a entregarla no estaba, así que decidí dejarla con la secretaria, le comenté a ella la razón pero me dijo que el no podría hacer nada que enviara mi queja a este correo, andres.rojas@mail.escuelaibg.edu.co, así que quiero comunicarle que mi grupo se llama "el equipo del tío de conejo" conformado por Javier Ernesto Rojas Corso ID 2104433 y Lina María Rodríguez Mora ID 2109369, la inscripción a este concurso ya está cancelada por mi compañero Javier, espero usted pueda solucionarme este problema ya que tenemos que presentar este trabajo el día lunes 29 de octubre.
Espero una pronta respuesta.

Gracias por su atención

Lina Rodríguez M

Ilustración 33 (Escuela Ing Julio Garavito. 2018.evidencia soporte)

-------- Mensaje original --------
Asunto: RE: Eciciencia
De: ANDRES FELIPE ROJAS ORTIZ
Para: RODRIGUEZ MORA LINA MARIA
CC:

Buena tarde,

Ya se asignó al estudiante Javier Ernesto al equipo " el equipo del tío de conejo " por favor revisar el número de carnet del estudiante ya que no es el correcto. Como la orden tenía fecha extemporánea de pago para el 26 de octubre, ya no puede realizar el pago de la misma.

Cordialmente,

Andrés Felipe Rojas
Ingeniero de Desarrollo
Oficina de sistemas y recursos informáticos.
andres.rojas@escuelaing.edu.co
Escuela Colombiana de Ingeniería Julio Garavito
Autopista Norte AK 45 No. 205-59
PBX: (57-1) 6683600 Ext. 272
Bogotá, D.C., Colombia
www.escuelaing.edu.co

Ilustración 34 (Escuela Ing Julio Garavito. 2018.evidencia soporte)

Prueba De Auditoria PA11.

Para la realización de la prueba se seleccionó el sistema ‘Encuentro de ingeniería mecánica’ dado que es uno de los más expuestos a personas externas a la institución según lo informado por los desarrolladores. Este sistema nos fue entregado sobre un ambiente de pruebas y su página de inicio es la que se muestra a continuación:

Ilustración 35 (Escuela Ing. Julio Garavito.2018. Página Encuentro de ingeniería mecánica)

Se procede a hacer la evaluación en la pantalla de registro de usuarios que se muestra a continuación:

Ilustración 36 (Escuela Ing. Julio Garavito.2018. Página Encuentro de ingeniería mecánica)

En este formulario se procede a verificar que cada campo tenga las validaciones necesarias para que la información que se va a almacenar en la base de datos sea confiable. Allí se evidencio lo siguiente:

- Los campos Nombre y Apellido aceptan caracteres numéricos lo cual puede representar información poco confiable.
- El campo tipo de documento está limitado por una lista desplegable lo cual proporciona mayor seguridad de que la información seleccionada es correcta.
- El campo documento de identidad tiene bloqueada la opción de ingresar letras lo cual aplica perfectamente.
- Se tienen validaciones para el formato de fecha y se presenta en el label el formato que se debe usar lo cual se ajusta al correcto funcionamiento.
- El campo institución se selecciona de un desplegable lo que evidencia una buena práctica.
- El campo teléfono de contacto no tiene validaciones y acepta cualquier carácter lo cual evidencia falencias en el proceso de almacenamiento de datos, además no está particionado para identificar indicativo y/o código postal.
- El campo dirección acepta cualquier tipo de carácter lo cual aplica perfectamente al tipo de información solicitada.
- Se valida el campo de correo electrónico perfectamente.
- Se hace validación de ingreso obligatorio de contraseña y el tipo de rol, lo cual asegura la inserción completa y confiable de datos.

A continuación, se presentan las imágenes de la prueba realizada:

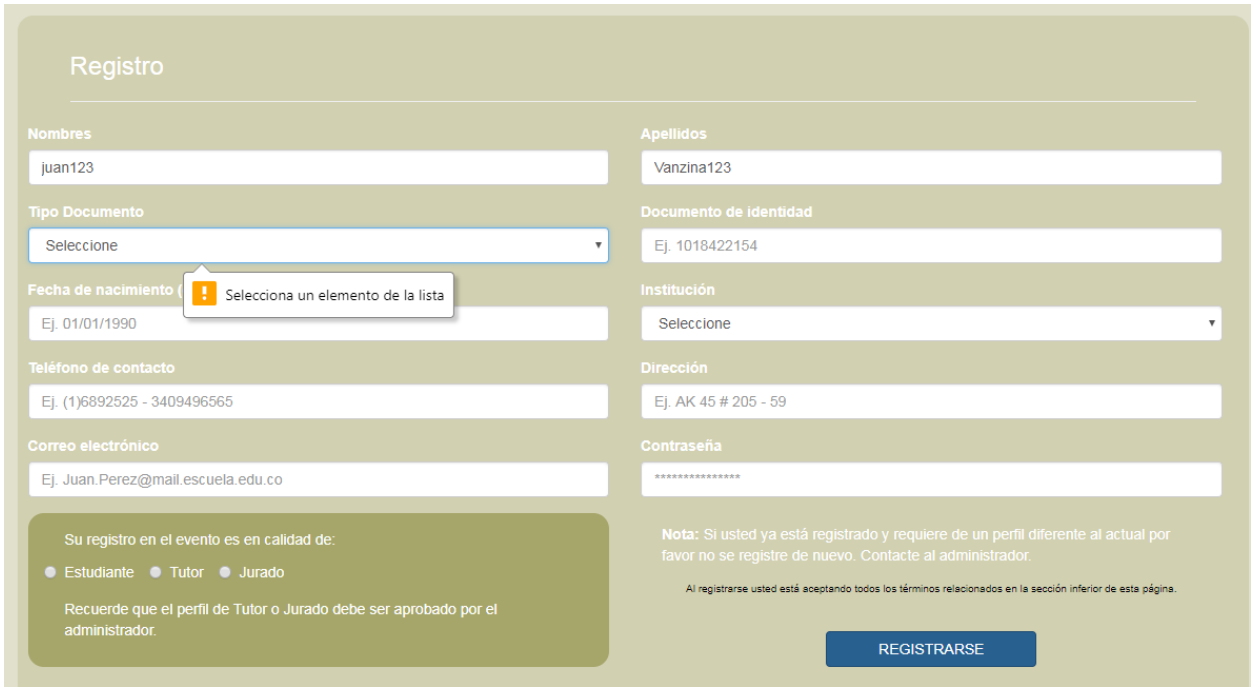


Ilustración 37 (Escuela Ing. Julio Garavito.2018. Página Encuentro de ingeniería mecánica)

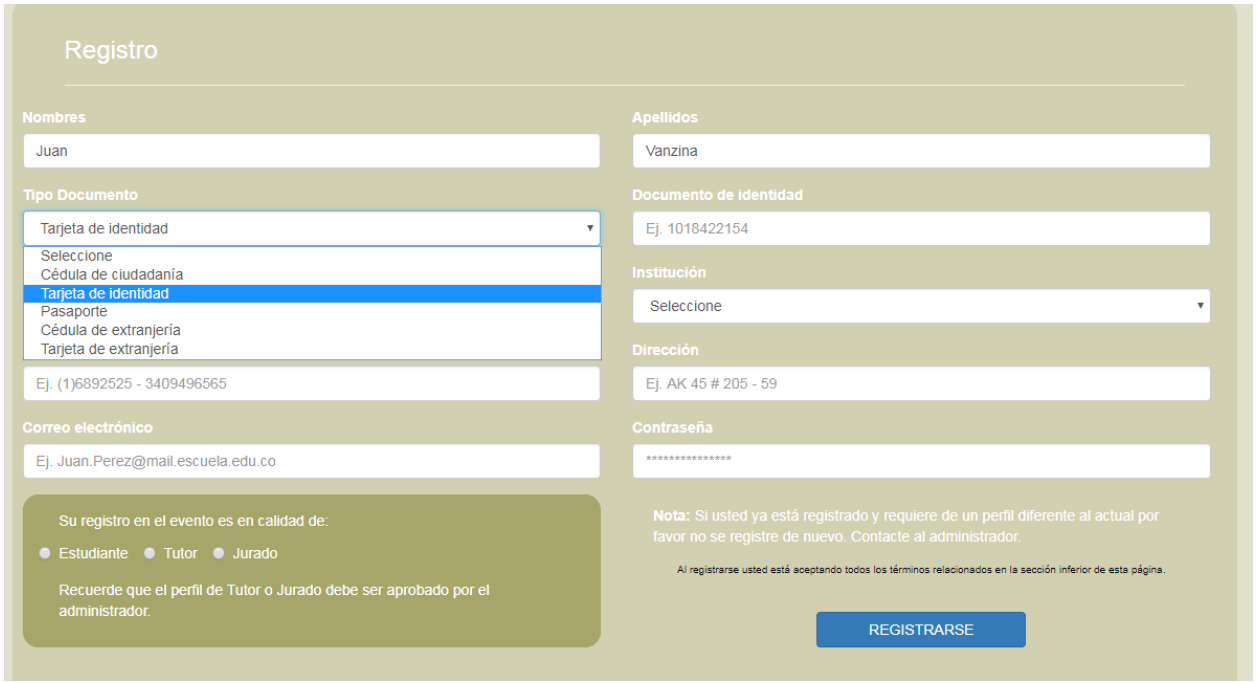


Ilustración 38 (Escuela Ing. Julio Garavito.2018. Página Encuentro de ingeniería mecánica)

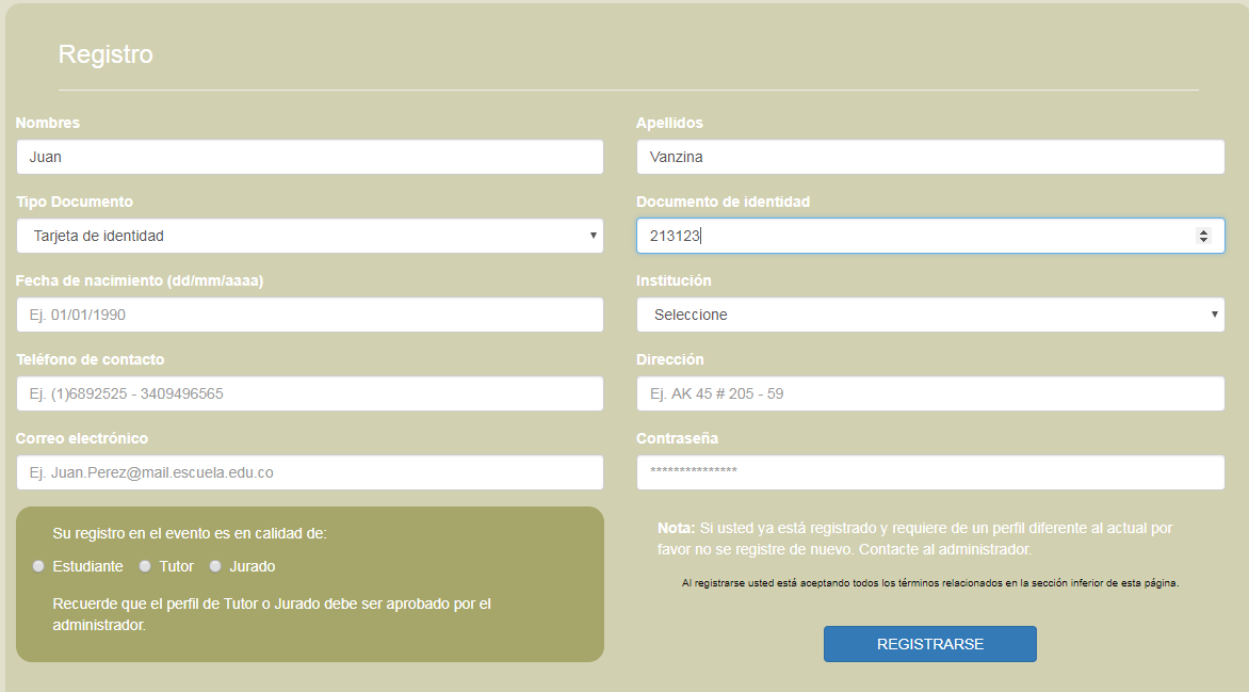


Ilustración 39 (Escuela Ing. Julio Garavito.2018. Página Encuentro de ingeniería mecánica)

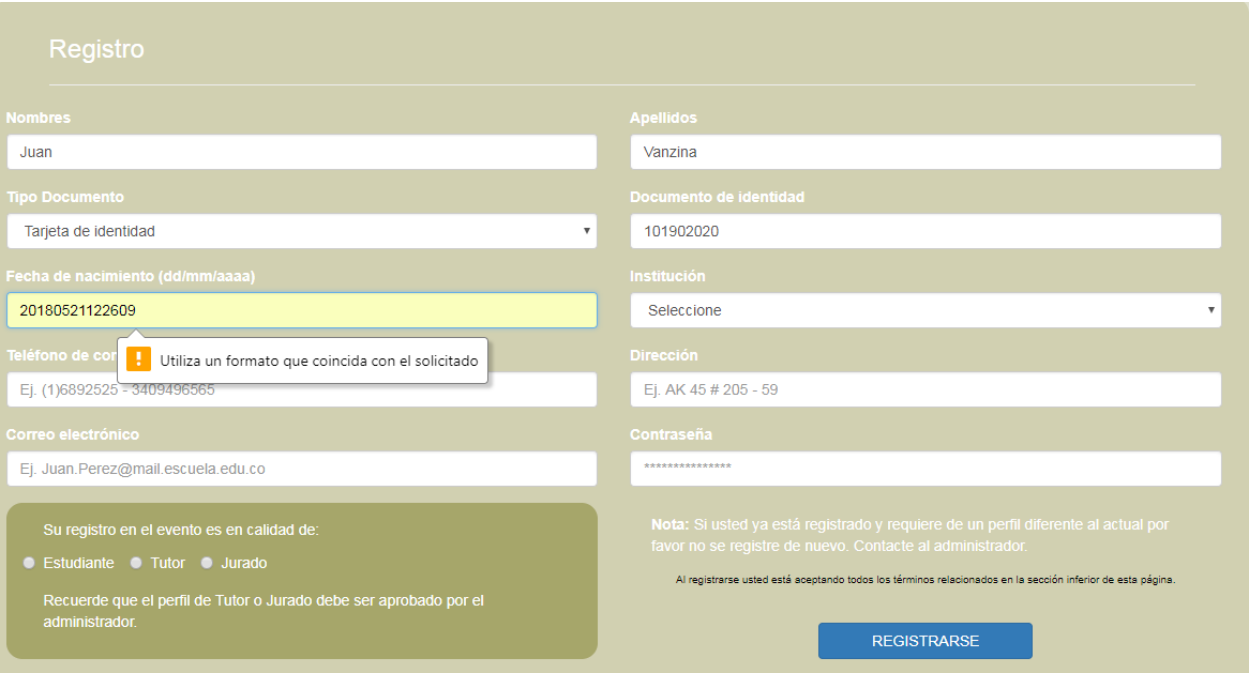


Ilustración 40 (Escuela Ing. Julio Garavito.2018. Página Encuentro de ingeniería mecánica)

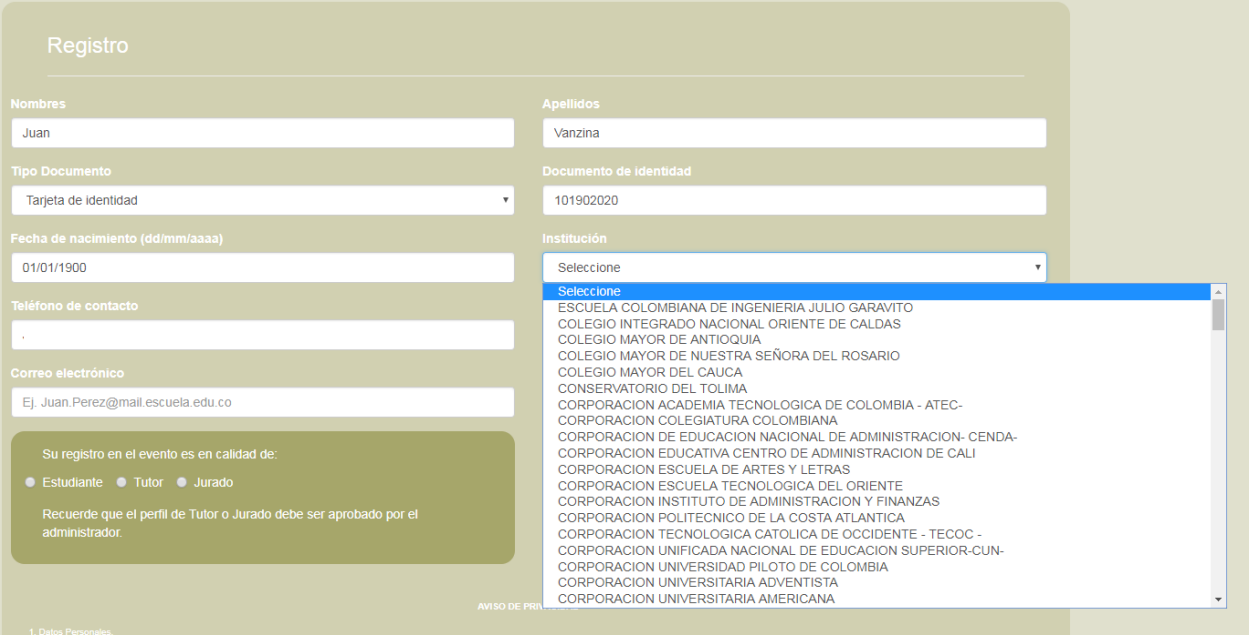


Ilustración 41 (Escuela Ing. Julio Garavito.2018. Página Encuentro de ingeniería mecánica)

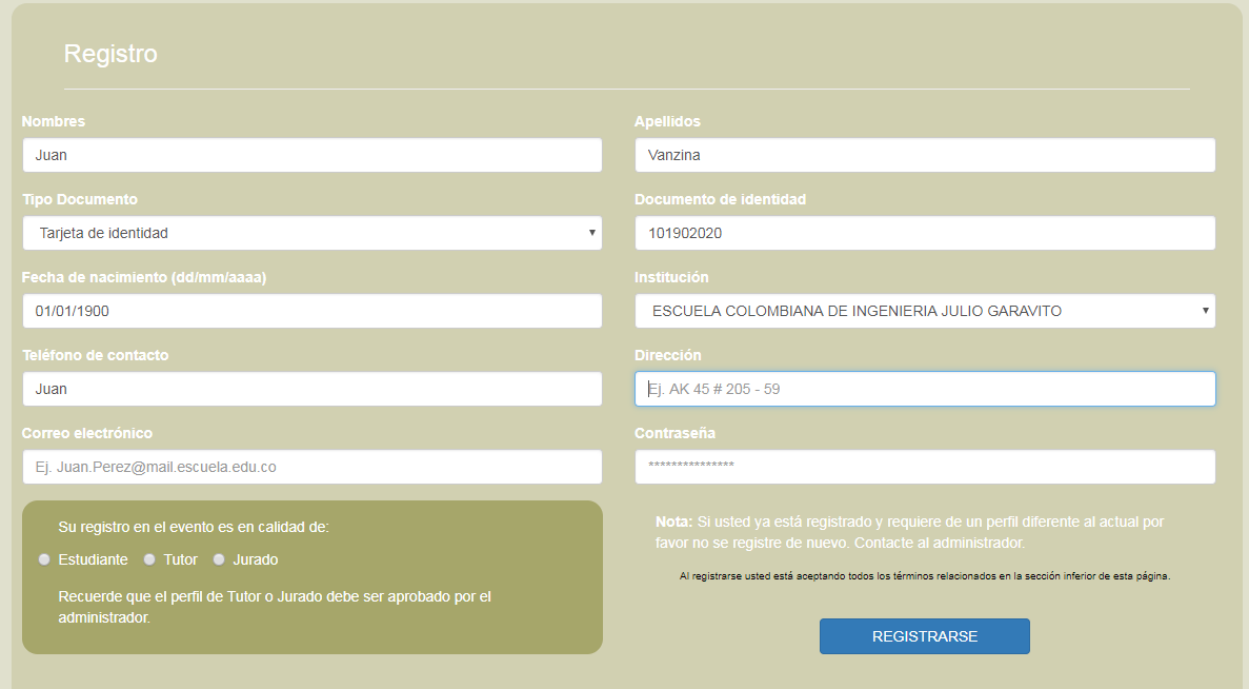


Ilustración 42 (Escuela Ing. Julio Garavito.2018. Página Encuentro de ingeniería mecánica)

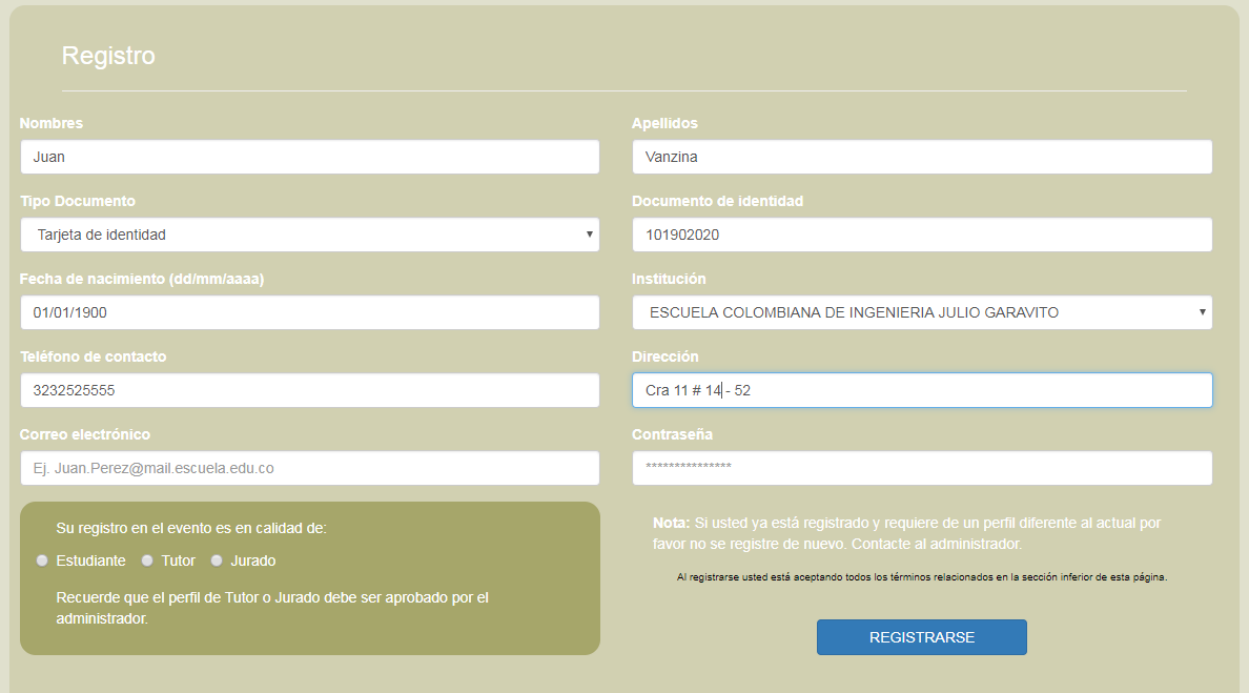


Ilustración 43 (Escuela Ing. Julio Garavito.2018. Página Encuentro de ingeniería mecánica)

Ilustración 44 (Escuela Ing. Julio Garavito.2018. Página Encuentro de ingeniería mecánica)

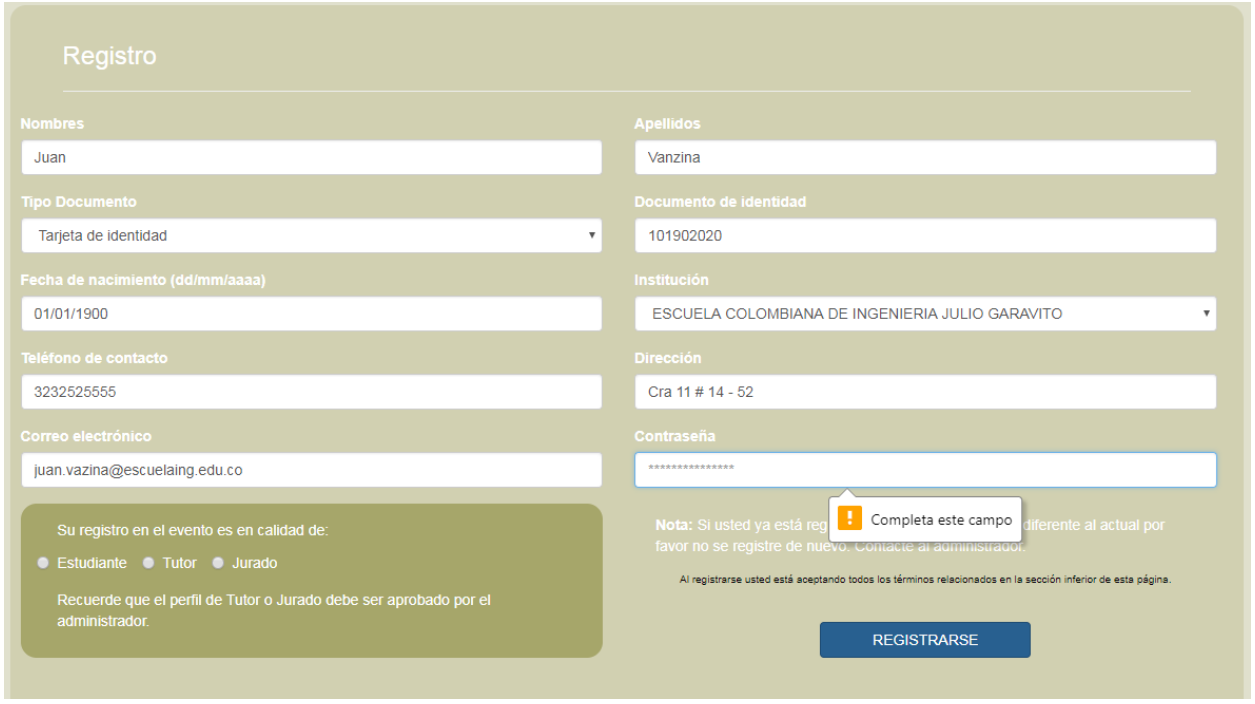


Ilustración 45 (Escuela Ing. Julio Garavito.2018. Página Encuentro de ingeniería mecánica)

Ilustración 46 (Escuela Ing. Julio Garavito.2018. Página Encuentro de ingeniería mecánica)

Prueba De Auditoria PA12

Para la validación de la prueba, el área de desarrollo puso a disposición el sistema “Plataforma de seguimiento a la planeación” desarrollado por ellos en un ambiente de pruebas, proporcionando un usuario y una contraseña de pruebas.

Con acompañamiento e inducción de un desarrollador sobre la funcionalidad del sistema, se procede a realizar la creación de un proyecto de prueba con el fin de verificar que se valide la información ingresada por los usuarios, para ello se ingresa en el campo fecha un número y posteriormente se da clic en el botón “Guardar Datos Básicos”, inmediatamente sale un mensaje informando que se debe utilizar el formato de fecha para poder procesar la información, como se muestra en la siguiente imagen.

Adicionalmente se intentó ingresar letras en el campo de fecha y se evidencio que el campo no permite la escritura de estos caracteres.

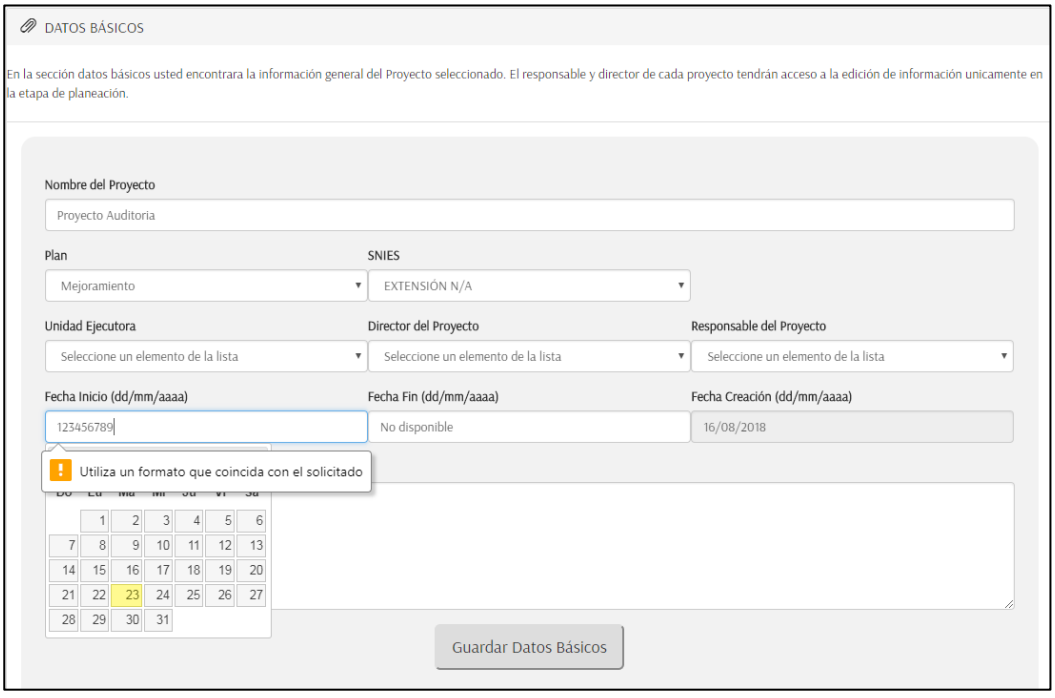


Ilustración 47 (Escuela Ing. Julio Garavito.2018. Página Plataforma de seguimiento a la planeación)

Se realiza una validación de rango de un campo, para esto se procedió a crear una actividad para en el proyecto. En el campo de porcentaje para el proyecto se ingresa un número mayor a 100% y se da clic en el botón “Crear Actividad”, inmediatamente sale un mensaje indicando que el valor del campo porcentaje debe ser inferior o igual a 100, como se muestra a continuación:

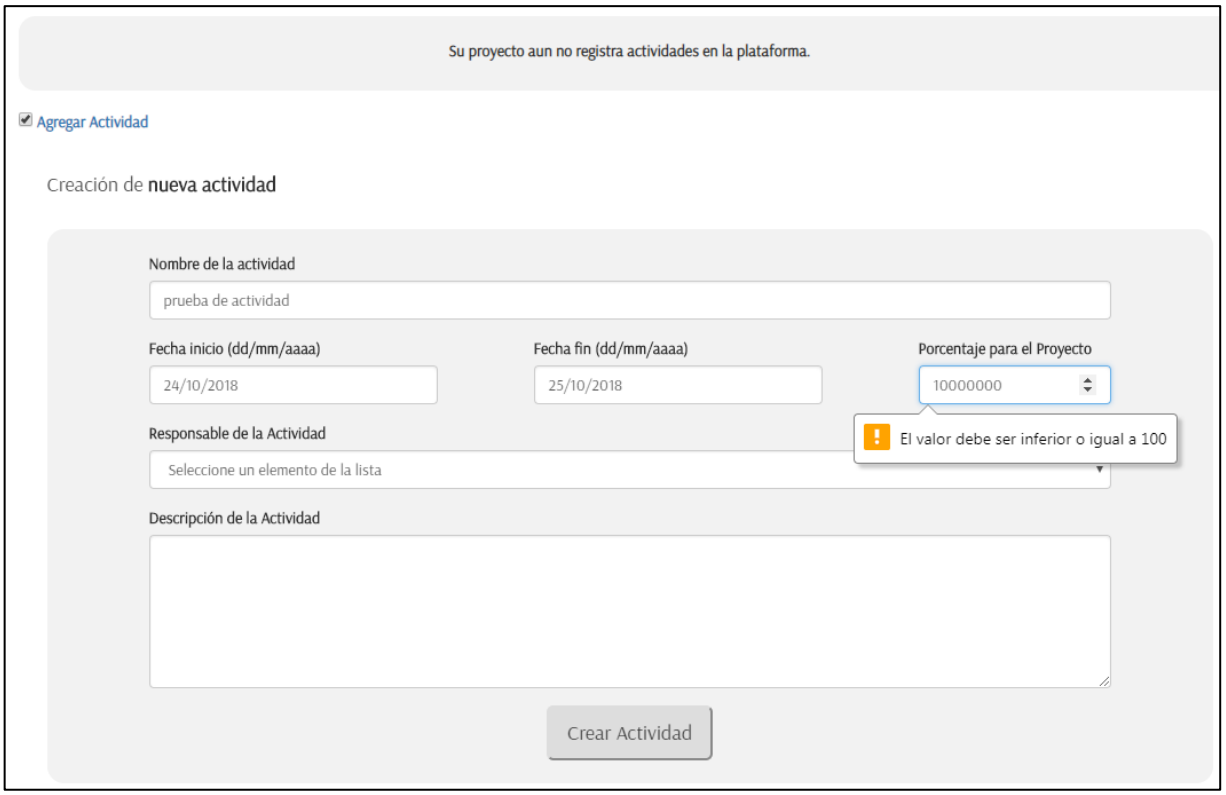


Ilustración 48 (Escuela Ing. Julio Garavito.2018. Página Plataforma de seguimiento a la planeación)

Prueba De Auditoria PA13.

Para la siguiente prueba se abren dos páginas del explorador y se accede al mismo sistema “Plataforma de seguimiento a la planeación” con las mismas credenciales de autenticación, como se muestra a continuación:

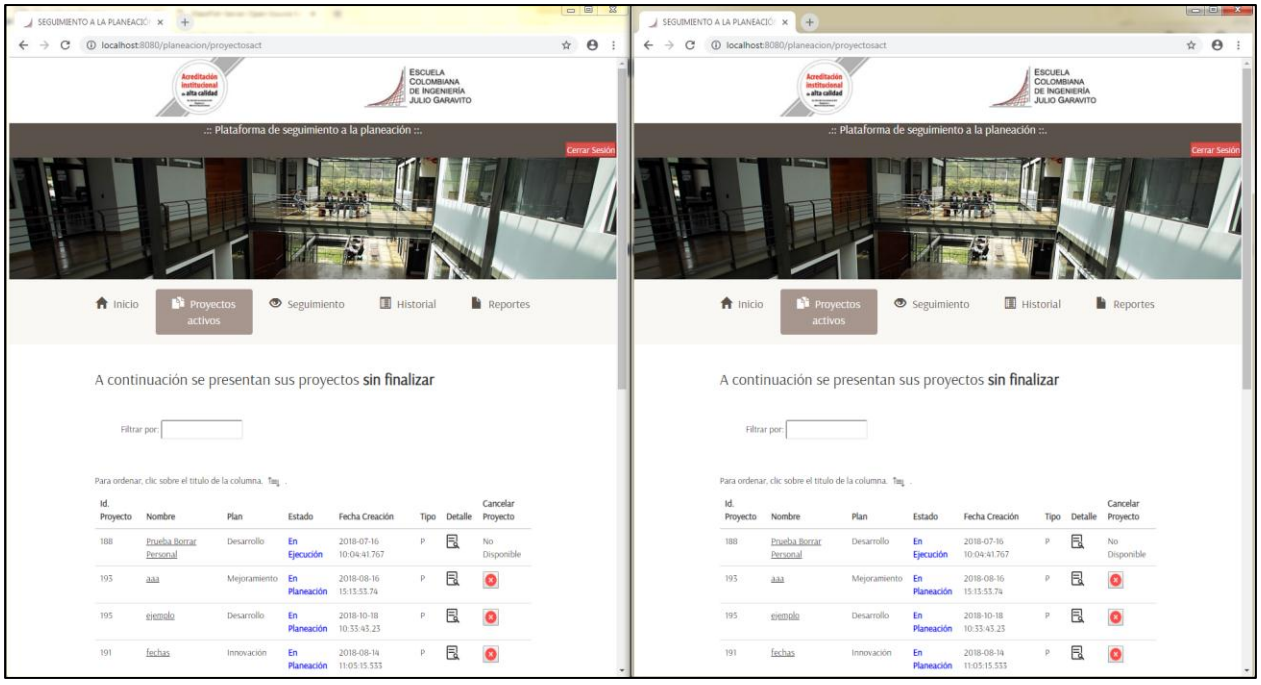


Ilustración 49 (Escuela Ing. Julio Garavito.2018. Página Plataforma de seguimiento a la planeación)

Posteriormente se cierra sesión en una de las páginas y verificar que se halla cerrado correctamente como se muestra en la pantalla del lado izquierdo de la siguiente imagen.

A continuación, la página del lado derecho se actualiza y se comprueba que también se haya cerrado la sesión de la cuenta con la que se autentico, como se evidencia a continuación:

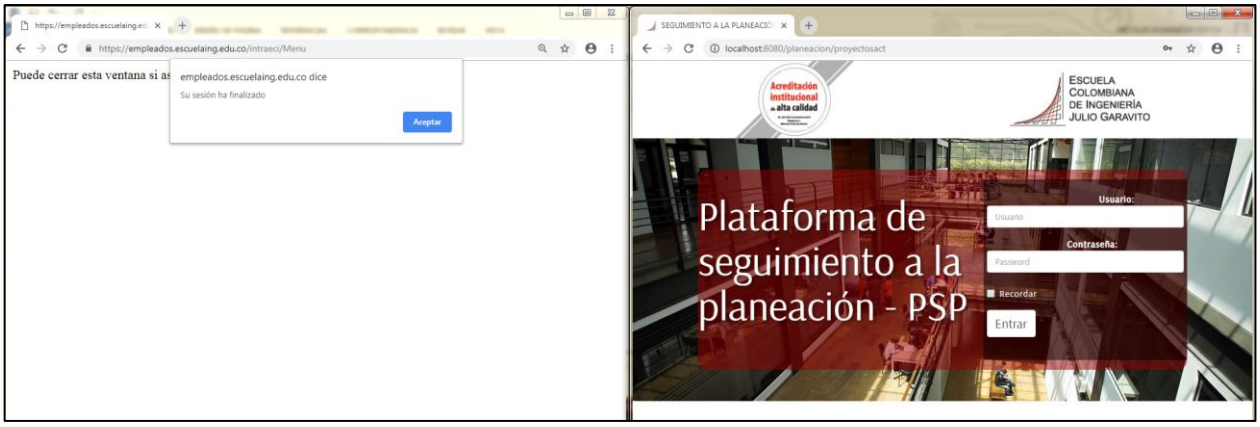


Ilustración 50 (Escuela Ing. Julio Garavito.2018. Página Plataforma de seguimiento a la planeación)

Prueba De Auditoria PA14.

Actualmente no se hace uso de dispositivos de autenticación para las aplicaciones en las que ejecutan procesos sensibles y críticos de la institución, solo se autentican son el usuario y correo institucional.

Prueba De Auditoria PA15.

Se procede a verificar las cookies que utiliza el sistema "Plataforma de seguimiento a la planeación", donde se evidencia que las cookies que se usan al ingresar al sistema es la cookie JSESSIONID la cual se crea o envía cuando se inicia la sesión, como se muestra a continuación:

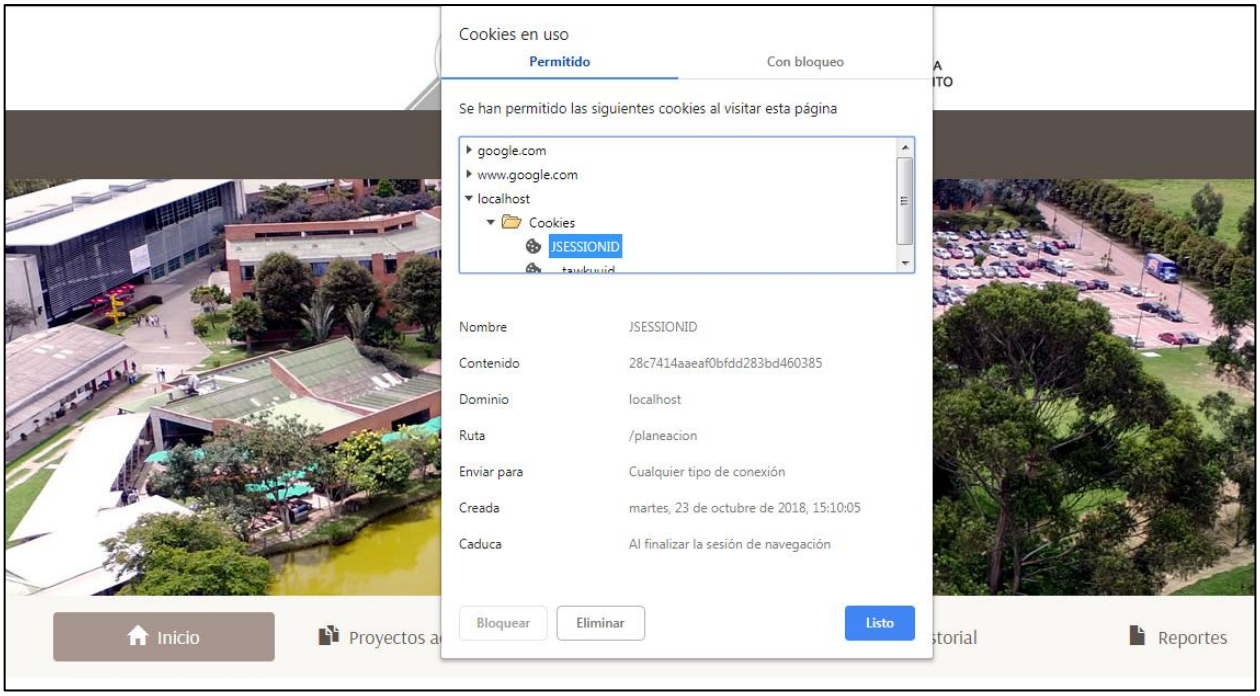


Ilustración 51 (Escuela Ing. Julio Garavito.2018. Página Plataforma de seguimiento a la planeación)

La plataforma también usa la cookie __tawkuuid de forma transparente para la Web, identifica el usuario que ha iniciado sesión, como se muestra a continuación:

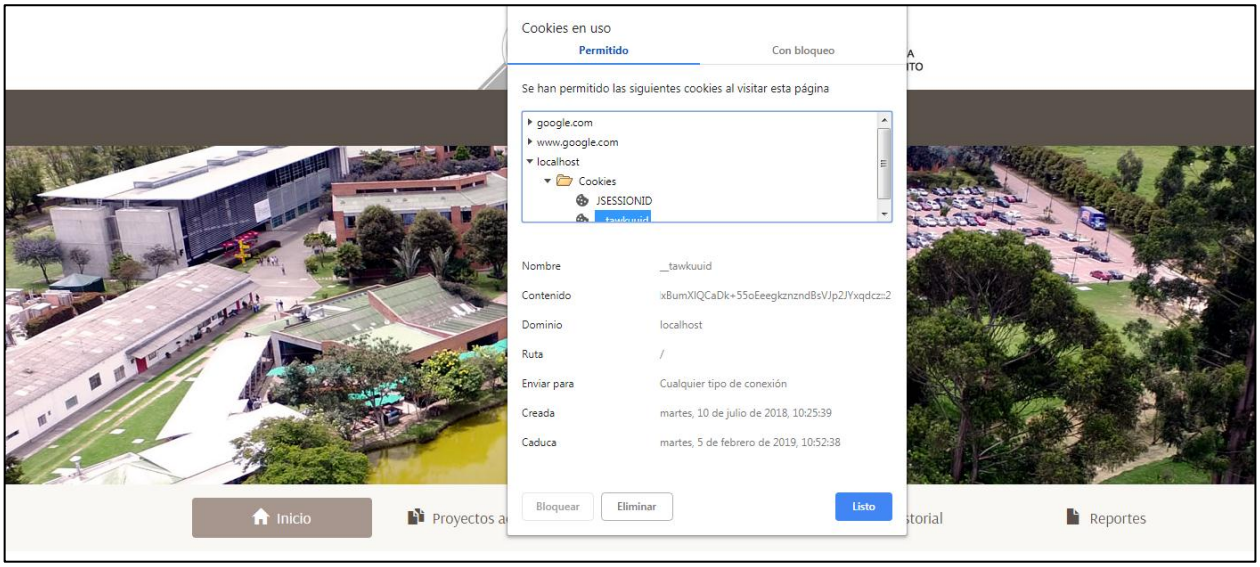


Ilustración 52 (Escuela Ing. Julio Garavito.2018. Página Plataforma de seguimiento a la planeación)

Prueba De Auditoria PA16

Con ayuda de un desarrollador se simulo un error en el sistema “Plataforma de seguimiento a la planeación”, en donde se puede evidenciar como primera instancia que no existe un mensaje de error genérico. Aunque el error simulado no muestra información de cuentas de usuario, si se puede observar la versión del servidor de aplicaciones utilizado, en este caso es Glassfish 3.1.2.2

Ilustración 53 (Escuela Ing. Julio Garavito.2018. Página de error plataforma de seguimiento a la planeación)

Prueba De Auditoria PA17.

Para esta prueba se ingresa a la opción archivos asociados al proyecto, se pude observar que el proyecto tiene un archivo adjunto, como se muestra a continuación:

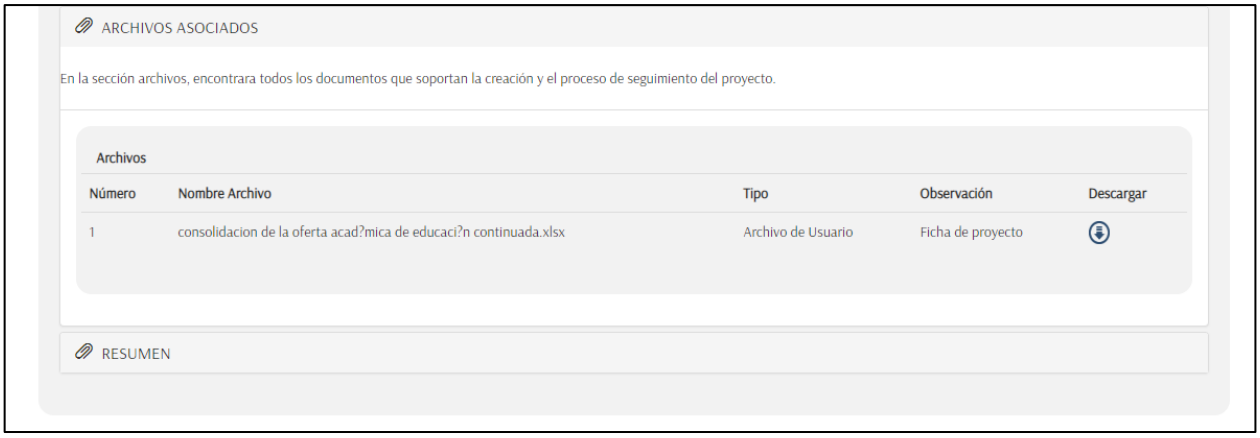


Ilustración 54 (Escuela Ing. Julio Garavito.2018. Página Plataforma de seguimiento a la planeación)

Posteriormente se procede a descargar el archivo y se verifica en las descargar del navegador si muestra alguna estructura de directorios del sistema de información, como se evidencia en la siguiente imagen no se revela directorios del servidor donde está alojado los archivos.

Ilustración 55 (Escuela Ing. Julio Garavito.2018. Página Plataforma de seguimiento a la planeación)

Prueba De Auditoria PA18.

Para verificar la información de los encabezados de respuesta del sistema “Plataforma de seguimiento a la planeación”, se hace uso de la del complemento de Chrome Espía HTTP, en donde se observa que la única información relevante es el nombre del servidor de aplicaciones Glassfish versión 3.1.2.2, como se muestra en la siguiente imagen.

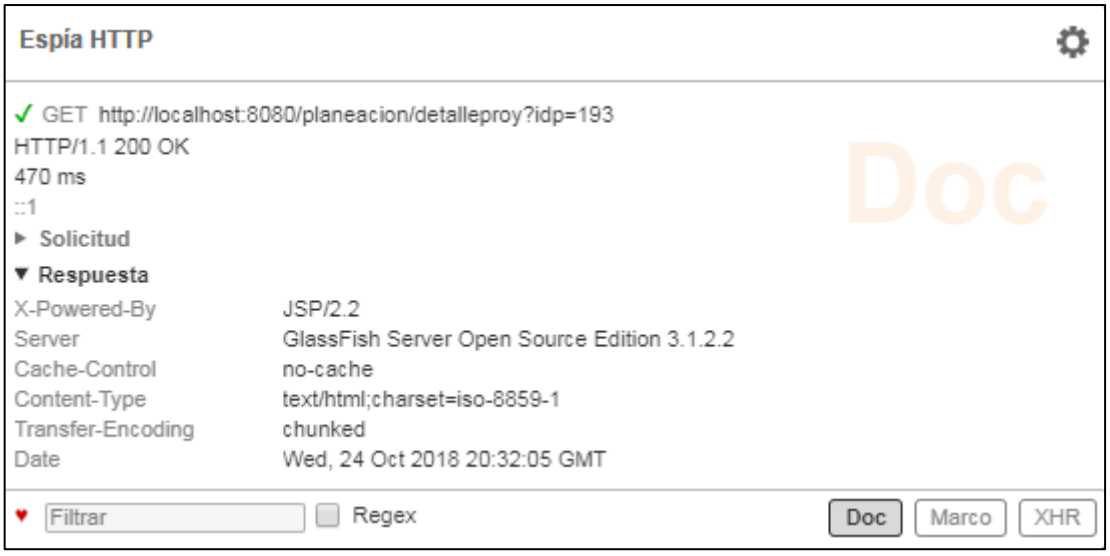


Ilustración 56(Escuela Ing. Julio Garavito.2018. Encabezados de respuestas Página plataforma de seguimiento a la planeación)

Prueba De Auditoria PA19.

Se verifico en compañía de un desarrollador y nos comenta que los aplicativos desarrollados no tienen quemado por código las cadenas de conexión de la base de datos. Los aplicativos tienen las credenciales de usuario, contraseña e IP de la base de datos en el pool de conexión de Glassfish. No se pueden tomar evidencias por confidencialidad de la información.

Prueba De Auditoria PA20.

Se verifico el código fuente del aplicativo "Plataforma de seguimiento a la planeación" en compañía de un desarrollador y se pudo constatar que por cada query realizado a la base de datos se realiza la desconexión a la base de datos. No se pueden tomar evidencias por confidencialidad de la información.

Prueba De Auditoria PA21.

El código fuente de las aplicaciones se encuentran alojadas en los equipos de cada desarrollador. A estos equipos puede ingresar cualquier empleado de la institución con su cuenta de dominio y no tendrá acceso a la información ni códigos fuentes de las aplicaciones de los

desarrolladores. Los backups de los códigos fuentes de las aplicaciones Web se encuentran en un servidor del área de TI.

RECOMENDACIONES Y RESULTADOS

Prueba de Auditoria	HALLAZGOS	RECOMENDACIONES	
PA1	Se encuentra que los desarrolladores hacen pruebas funcionales y de seguridad en inicio de sesión, sin embargo, no se evidencian pruebas que evalúen otras vulnerabilidades de seguridad del aplicativo.	RC1	• Se recomienda hacer uso de herramientas que hagan pruebas automáticas de vulnerabilidades sobre los aplicativos cuando salen a producción. • Verificar que existe una política explícita para el manejo de las claves criptográficas (por ejemplo, generadas, distribuidas, revocadas y vencidas). • Establecer lineamientos enfocados a la seguridad en credenciales de acceso a todos lo sistemas.
PA2	En la encuesta realizada se encuentra que los desarrolladores no dejan constancia de las reuniones ni tampoco documentan la aceptación de los requerimientos por parte del propietario del sistema. También se encuentra que la recepción de funcionalidades es algunas veces delegada a otros funcionarios.	RC2	• Se recomienda hacer uso de actas que contengan la información de las pruebas realizadas con el usuario para controlar la aceptación de entrega de sistemas o funcionalidades nuevas. • Establecer medidas que aseguren que propietario del software es el único autorizado para dar aprobación a nuevos requerimientos o sistemas nuevos antes de la salida a producción. • Verificar que todos los componentes que no son parte de la aplicación pero que son necesarios para su funcionamiento, estén alineados a la estrategia de negocio.
PA3	Durante el proceso de observación se evidencia que para las funcionalidades nuevas se realiza un proceso de pruebas unitarias insuficientes para determinar el paso a producción.	RC3	• Se recomienda optimizar la fase de pruebas, implementando el uso de metodologías y buenas prácticas que disminuyan el riesgo que implica pasar aplicativos a producción.
PA4	El proceso de migración entre ambientes no está segregado a un único responsable, por lo que cualquiera de los desarrolladores pues migrar	RC4	• Se recomienda limitar el acceso a cada ambiente según el rol del desarrollador en la institución. • Delegar credenciales específicas tanto para los desarrolladores como para los aplicativos que

	código de aplicativos o sentencias en las bases de datos sin contar con restricciones o pistas de auditoria ya que utilizan el mismo usuario.		salen a producción y que hacen conexión a la base de datos.
PA5	Se encuentra que el paso entre ambientes no cuenta con controles que garanticen la seguridad en integridad, confidencialidad y disponibilidad.	RC5	• Asignar usuarios y contraseñas para el paso entre ambientes que permita dejar pistas de auditoria sobre los movimientos efectuados por el personal.
PA6	Se evidencia la ausencia de un control de versiones formalizado.	RC6	• Se recomienda estandarizar y documentar el proceso de control de versiones y paso entre ambientes. .
PA7	Al realizar la observación del proceso se encuentran falencias en el proceso de documentación en cada fase sin embargo, se cumple con la totalidad de las fases.	RC7	• Se recomienda la segregación de funciones para dar adecuado tratamiento a cada fase de desarrollo en especial en las fases de pruebas, mantenimiento y paso a producción. • Hacer inspecciones periódicas de los procesos realizados en cada fase de desarrollo.
PA8	Se evidencia que no existe un documento con lineamientos de la institución para el desarrollo seguro, por lo que se procede a evaluar respecto a las buenas prácticas de OWASP.	RC8	• Se recomienda documentar las buenas prácticas utilizadas por la institución y adoptar una metodología de desarrollo ágil. • De igual forma se recomienda hacer uso de buenas prácticas basados como mínimo en el top 10 de riesgos críticos en aplicaciones web, emitido por OWASP anualmente.
PA9	No se encuentran documentados los acuerdos de nivel de servicio (ANS), sin embargo, se evidencia la unanimidad de las condiciones entre los desarrolladores al realizar el levantamiento de información.	RC9	• Es indispensable la documentación de los ANS para los servicios que provee el área de desarrollo. • Se recomienda clasificar la prioridad de atención de las solicitudes para la institución respecto al nivel de riesgo, es decir, Probabilidad v Impacto y no por frecuencia como especifican los desarrolladores.
PA10	.Durante la prueba se realiza el seguimiento a dos soportes de días anteriores y se comprueba que los tiempos de respuesta corresponden a los mencionados en la prueba 9.	RC10	• Se recomienda implementar mecanismos que permitan realizar el seguimiento a las solicitudes de soporte para evaluar posibles comportamientos inusuales.
PA11	En la prueba realizada se encuentra que el aplicativo cuenta con las validaciones necesarias para asegurar que los	RC11	• Se recomienda utilizar otras técnicas de validación que no puedan ser manipuladas por los usuarios en los navegadores. • Las contraseñas definidas por los usuarios

	datos de entrada y generación de datos de salida sean confiables, sin embargo, se evidencia que los scripts de validación son vulnerables ante algunos navegadores y ante la manipulación del código de la página.		deben tener mecanismos de encripción al realizar el almacenamiento que requiera un factor de trabajo lo suficientemente alto para evitar un ataque de fuerza bruta. • Verificar que al usuario se le sugiera la opción de terminar todas las otras sesiones activas después de un proceso de cambio de contraseña exitoso
PA12	Se evidencia la validación del tipo de datos, rangos y longitud de la información ingresada por el usuario para el aplicativo de prueba, sin embargo, no se pudo realizar la prueba para todos los aplicativos Web. No se encuentra documentación de los estándares de validación de los datos que ingresan a los aplicativos, las validaciones evidenciadas son realizadas según el criterio de cada desarrollador.	RC12	• Se recomienda realizar una validación del tipo de información capturada en los formularios de todos los aplicativos. • Los aplicativos webs deben especificar las reglas para validación y codificación de cada dato de entrada para todas las aplicaciones desarrolladas. • Los aplicativos web deben tener codificados y validados las acciones a tomar cuando se reciben entradas no válidas.
PA13	La aplicación de prueba permite abrir con la misma sesión múltiples pestañas y páginas del navegador de la misma aplicación. Sin embargo, cuando se realiza el cierre de sesión en una de las páginas, simultáneamente se cierra sesión en las demás páginas que tenga abierta el mismo sistema.	RC13	• Las aplicaciones Web no deben permitir que múltiples pestañas o ventanas de navegador compartan la misma sesión de usuario. • Si un usuario a iniciado sesión se debe solicitar al usuario volver a autenticarse si abre una nueva pestaña o ventana del navegador web de la misma aplicación web. • La aplicación no debería permitir compartir la misma ID de sesión simultáneamente en dos páginas o pestañas diferentes. • Las sesiones deben ser invalidadas cuando ya no son necesarias y el tiempo es limitado durante los períodos de inactividad. • Se recomienda aplicar las directrices generales para el manejo de sesiones definidos por OWASP.
PA14	Para los procesos sensibles de la institución realizados en aplicativos desarrollados in-house, no se realiza una autenticación con dispositivos adicionales de verificación, se hace la autenticación únicamente con usuario y	RC14	• Para los procesos sensibles de la institución que sean ejecutados en alguno de los aplicativos Web, se debe tener un dispositivo de autenticación como un token, que permita demostrar que la identidad de la persona es correcta.

	contraseña.		
PA15	En el aplicativo de prueba se evidencia el uso de las siguientes cookies almacenados: la cookie JSESSIONID el cual se crea o envía cuando se inicia la sesión en la aplicación y la cookie __tawkuuid el cual permite identificar al usuario que ha iniciado sesión en el aplicativo.	RC15	• Aplicar el atributo de cookie "Seguro" indicando a los navegadores web que solo envíen la cookie a través de una conexión HTTPS (SSL / TLS) cifrada. • Forzar a la aplicación web a usar solo HTTPS para su comunicación. • Aplicar el atributo de cookie "HttpOnly" indica a los navegadores web que no permitan que los scripts tengan la capacidad de acceder a las cookies a través del objeto DOM document.cookie. Esta protección de ID de sesión es obligatoria para evitar el robo de ID de sesión a través de ataques XSS.
PA16	Se evidencio que al generar un error en el sistema Web, no existe un mensaje de error genérico, aunque el error simulado no muestra información de cuentas de usuario, si se puede observar la versión del servidor de aplicaciones utilizado, en este caso es Glassfish 3.1.2.2.	RC16	• Implementar mensajes de error genéricos que sean activados cuando ocurra una excepción en el sistema, como por ejemplo "Error del sistema: inténtelo de nuevo más tarde". • No exponer información confidencial en mensajes de excepción, cualquier información interna del sistema debe estar oculta al usuario. • No ponga nombres de personas ni ninguna información de contacto interna en los mensajes de error. • No coloque ninguna información "personal", lo que llevaría a un nivel de familiaridad y una explotación de ingeniería social.
PA17	No se evidencia la revelación de directorios del servidor donde está alojado los archivos al momento de descargarlos del aplicativo web.	RC17	• Prevenir la revelación de la estructura de directorios en el archivo robots.txt colocando directorios que estén disponibles para el índice público en un directorio raíz aislado.

PA18	En el encabezado web de la aplicación de prueba se observa que la única información relevante es el nombre del servidor de aplicaciones el cual es Glassfish versión 3.1.2.2.	RC18	• Hacer uso de las directivas de control de los encabezados de respuesta enviada por el servidor de memoria rápida que pueden ser configuradas mediante código. Estas directivas controlan los contenidos que el navegador del cliente guarda en memoria rápida. Las directivas a configurar son cache-control: no-cache ó cache-control: no-store.
PA19	Se evidencia por medio de la observación directa en el código fuente del aplicativo, que no se tiene quemado por código las credenciales de conexión a la base de datos, el desarrollador indica que la conexión se hace a través del pool de conexiones de Glassfish	RC19	• Las credenciales requeridas para la comunicación con la base de datos deben almacenarse cifradas y fuera del código dentro de archivos de configuración.
PA20	Por observación directa sobre el código fuente del aplicativo de prueba, se evidencia que, por cada insert, update, delete o select realizado a la base de datos se realiza la desconexión a la base de datos.	RC20	• Verificar el cierre de conexiones de la de a base de datos de todos los insert, update, delete o select que se realicen en todas las aplicaciones web, la desconexión se debe hacer tanto si se ha ejecutado con éxito la operación o no.
PA21	Se evidencia que todas las fuentes de los aplicativos web desarrollados por el área, se encuentran alojados en cada uno de los equipos de los desarrolladores, además los backups de los códigos fuentes están en una carpeta en un servidor del área de TI al cual tiene acceso los desarrolladores, la coordinadora de seguridad informática, el coordinador de infraestructura, el DBA y el director de TI.	RC21	• Llevar un registro actualizado de todos los programas fuente en uso, Indicando nombre del programa, versión, fecha de última modificación y fecha / hora de compilación. • Administrar las distintas versiones de una aplicación. • Establecer que todo ejecutable en producción tenga un único programa fuente asociado que garantice su origen. • A los backups de los códigos fuentes de las aplicaciones solo debe tener acceso los desarrolladores, se debe restringir el acceso del resto de personal.

RC (Recomendación). R (1)-R (21).

PRIORIZACIÓN DE LAS RECOMENDACIONES

Haciendo uso de una matriz de prioridad para los procesos de TI, se realizará la matriz

ponderada para los subprocesos del desarrollo de software, ordenándolos en términos de riesgo e impactos, con el fin de realizar una clasificación priorizada de las recomendaciones emitidas. Las recomendaciones se agruparán según al subproceso de desarrollo que correspondan.

A continuación, se muestra las escalas de probabilidad e impacto utilizadas para la valoración de riesgo:

Tabla 3 Escala de probabilidad

Probabilidad	
1	baja
2	media
3	alta

Tabla 4 Escala de valoración de impacto

Escala de Valoración de Impacto			
Valor	Confidencialidad	Disponibilidad	Integridad
1	bajo	bajo	bajo
2	medio	medio	medio
3	alto	alto	alto

A continuación se realiza en análisis de riesgo para los subprocesos: Análisis de requerimientos, Diseño, Codificación, Pruebas, Paso a producción y Mantenimiento del proceso de desarrollo de software de la Escuela colombiana de ingeniería Julio Garavito, bajo los criterios de confidencialidad, disponibilidad e integridad a los cuales se les asignara un peso del 1 al 3 de acuerdo a la importancia que tienen para la institución, en donde la confidencialidad tiene un peso de 2, la disponibilidad tiene un peso de 1 y la integridad tiene un peso de 3, siendo 3 el peso más alto, 2 el medio y 1 el bajo.

A cada subproceso se le asignó el valor correspondiente de la probabilidad y el impacto, el

resultado del producto de la probabilidad por el impacto por el peso del criterio da como resultado el nivel de riesgo de cada subproceso de acuerdo con el criterio de evaluación. Con el fin de obtener la sumatoria final de las calificaciones de cada subproceso se realiza la suma de la calificación obtenida en cada uno de los criterios de evaluación de riesgo.

Tabla 5 Nivel de riesgos subproceso de desarrollo

		Criterios de evaluación de riesgos									
		Confidencialidad			Disponibilidad			Integridad			
		Peso	2		Peso	1		Peso	3		
	Sub Procesos TI	Probabilidad	Impacto	Calificación	Probabilidad	Impacto	Calificación	Probabilidad	Impacto	Calificación	Sumatoria de Calificaciones
Procesos de TI	Mínimo	1	1	2	1	1	1	1	1	3	6
	Máximo	3	3	18	3	3	9	3	3	27	54
Desarrollo de Software	Análisis de requerimientos	2	2	8	1	3	3	2	3	18	29
	Diseño	1	3	6	2	2	4	1	3	9	19
	Codificación	3	3	18	2	3	6	2	3	18	42
	Pruebas	1	3	6	1	2	2	2	3	18	26
	Paso a producción	2	3	12	1	3	3	2	3	18	33
	Mantenimiento	1	3	6	1	3	3	1	3	9	18

La siguiente escala permite clasificar la valoración obtenida de los subprocesos de desarrollo de software:

Tabla 6 Escala de valoración de subprocesos

Escala de valoración de subprocesos		Descripción
Alto	38-54	Afecta gravemente a la institución
Medio	22-37	Afecta levemente a la institución
Bajo	6-21	No se ve afectada la institución

Posteriormente, se obtiene la priorización de los subprocesos del proceso de desarrollo de software, en donde se evidencia que la etapa de desarrollo tiene un nivel de prioridad alto, seguido de las etapas de análisis de requerimientos y pruebas con un nivel de prioridad medio y finalmente se tiene las etapas de diseño y arquitectura, documentación y mantenimiento en el nivel de prioridad bajo.

Tabla 7 Priorización de los subprocesos de desarrollo

Resultados de los Subprocesos Ordenados		
Sub Proceso	Calificación de Riesgo	Prioridad
Desarrollo	42	alto
Paso a producción	33	medio
Análisis de requerimientos	29	medio
Pruebas	26	bajo
Diseño	19	bajo
Mantenimiento	18	bajo

A continuación, se muestra la tabla priorizada las recomendaciones según la clasificación de los riesgos para los subprocesos de desarrollo de software, en donde la marca RC seguido de un número consecutivo significa el número de la Recomendación.

Tabla 6 Priorización de recomendaciones

Resultados de los Subprocesos Ordenados			
Sub Proceso	Recomendación	Calificación de Riesgo	Prioridad
Desarrollo	RC15	42	Alto
Desarrollo	RC16	42	Alto
Desarrollo	RC17	42	Alto
Desarrollo	RC18	42	Alto
Desarrollo	RC20	42	Alto
Desarrollo	RC11	42	Alto
Paso a producción	RC4	33	Medio
Paso a producción	RC5	33	Medio
Análisis de requerimientos	RC12	28	Medio
Análisis de requerimientos	RC14	28	Medio
Pruebas	RC1	28	Medio
Pruebas	RC2	28	Medio
Pruebas	RC3	28	Medio
Diseño y arquitectura	RC13	19	Bajo
Diseño y arquitectura	RC19	19	Bajo
Mantenimiento	RC6	19	Bajo
Mantenimiento	RC7	19	Bajo
Mantenimiento	RC8	19	Bajo
Mantenimiento	RC9	19	Bajo
Mantenimiento	RC10	19	Bajo
Mantenimiento	RC21	18	Bajo

RC (Recomendación). R (1)-R (21).

En la tabla anterior se evidencian los resultados de priorización de las recomendaciones agrupadas, de acuerdo con el subproceso de desarrollo correspondiente. Por consiguiente, las recomendaciones con nivel de prioridad alto deben ser las primeras en implementarse, y en secuencia la prioridad medio y bajo.

PRESUPUESTO

Tabla 8 Presupuesto global de la propuesta por fuentes de financiación (en miles de $).

RUBROS	VALOR UNITARIO	VALOR TOTAL
PERSONAL		**$ 340.908**
EQUIPOS		**$ 6.399.857**
SOFTWARE		**$ 143.992**
MATERIALES		**$ 0**
SALIDAS DE CAMPO		**$ 809.600**
MATERIAL BIBLIOGRÁFICO		**$ 0**
PUBLICACIONES Y PATENTES		**$ 0**
SERVICIOS TÉCNICOS		**$ 0**
VIAJES		**$ 0**
CONSTRUCCIONES		**$ 0**
MANTENIMIENTO		**$ 0**
ADMINISTRACION		**$ 0**
TOTAL		**$ 7.694.357**

Tabla 9 Descripción de los gastos de personal (en miles de $).

INVESTIGADOR / EXPERTO/ AUXILIAR	FORMACIÓN ACADÉMICA	FUNCIÓN DENTRO DEL PROYECTO	DEDICACIÓN Horas/semana	VALOR
Experto	Ingeniero de sistemas	Auditor de sistemas de información	12	$ 170.454
Experto	Ingeniero de sistemas	Auditor de sistemas de información	12	$ 170.454
TOTAL				**$ 340.908**

Tabla 10 Descripción y cuantificación de los equipos de uso propio (en miles de $)

EQUIPO	VALOR TOTAL
CPU Procesador: Intel(R) Core(TM) i7-6700 CPU 3.40GHz RAM: 8Gb OS: Windows 7 **Pantalla:** HP 22 Pulgadas.	$3.399.733
CPU Procesador: Intel(R) Core(TM) i7-3770 CPU 3.40GHz RAM: 8Gb OS: Windows 7 **Pantalla:** HP 22 Pulgadas.	$3.000.124

TOTAL	**$ 6.399.857**

Tabla 11 Descripción del software que se planea adquirir (en miles de $).

SOFTWARE	JUSTIFICACIÓN	VALOR TOTAL
Office 365	Se requiere para la documentación.	$ 71.996
Office 365	Se requiere para la documentación.	$ 71.996
TOTAL		**$ 143.992**

Tabla 12 Valoración de las salidas de campo (en miles de $).

ITEM	COSTO UNITARIO	#	TOTAL
Transporte al sitio	4.600	88	$ 404.800
Transporte al sitio	4.600	88	$ 404.800
TOTAL			**$ 809.600**

CONCLUSIONES, RECOMENDACIONES, APORTES Y TRABAJOS FUTUROS

PRODUCTOS PARA ENTREGAR

Los resultados obtenidos para cada objetivo planteado en el presente trabajo se describen a continuación, inicialmente se realizó el análisis del funcionamiento actual del proceso de desarrollo de software, posteriormente se realiza la guía de auditoria bajo la norma ISO 27001 con el fin de verificar el cumplimiento de la política de desarrollo seguro.

Finalmente se aplicó la guía de auditoria al proceso de desarrollo la cual da como resultado los hallazgos y recomendaciones encontrados con el fin de mejorar la calidad del proceso de desarrollos de software, además de priorizan de las recomendaciones obtenidas basadas en el nivel de riesgo de los subproceso de desarrollo de software (Tabla 6 Priorización de recomendaciones).

Tabla 13 Productos a entregar

Objetivo	Entregable
1. Identificar la situación actual de la institución para obtener una familiarización con el proceso de desarrollo.	• Análisis del funcionamiento actual del proceso de desarrollo de software.
2. Establecer una guía de auditoria para alinear el desarrollo seguro con las mejores prácticas institucionales.	• Guía de auditoria
3. Aplicar la guía de auditoria seleccionada, con el fin de evaluar el cumplimiento de la política de desarrollo seguro.	• Informe de auditoría • Matriz de priorización de recomendaciones

CONCLUSIONES

- Se identificó el estado actual del proceso de desarrollo de software, obteniendo un conocimiento de la ejecución de cada una de las etapas del proceso.
- Se debe realizar una mejor apropiación del equipo de desarrollo sobre el proceso y la metodología utilizados en el desarrollo de software.
- Los desarrolladores no documentan ninguno de los procesos realizados en cada etapa de la metodología de desarrollo.
- Se definido una guía de auditoria, con el fin de facilitar la verificación del cumplimiento de la política de desarrollo seguro de la institución para todos sus sistemas de información internos.
- Se realizaron pruebas de auditoria a dos sistemas de información internos, permitiendo obtener como resultado hallazgos y recomendaciones del cumplimento de la política de desarrollo seguro.
- Se realizó el análisis de riesgo de las etapas del proceso de desarrollo de software de acuerdo su probabilidad e impacto, bajo los criterios de disponibilidad, integridad y confidencialidad, obteniendo como resultado la priorización de las etapas del proceso de desarrollo de software de acuerdo a su nivel de riesgo.
- Con los resultados obtenidos de la auditoria, se evidencia que al proceso de desarrollo de software de la institución se le debe aplicar puntos de mejora, principalmente en la etapa de desarrollo puesto que es la etapa que obtuvo el mayor nivel de riesgo en el proceso.

RECOMENDACIONES

Una vez concluido el trabajo de grado, se recomienda:

- Atender las recomendaciones de la auditoría realizada, en base a la calificación del riesgo.
- Establecer el apetito de riesgo de la organización.
- Documentar la totalidad de los riesgos y controles en la institución para mediante mapas de calor evidenciar el riesgo inherente y residual.
- Dar cumplimiento a todas las políticas de desarrollo seguro en el desarrollo de nuevos sistemas de información y los sistemas existentes.
- Definir la totalidad de actividades propias de un SGSI, siguiendo estrictamente el ciclo PHVA.

APORTES

- La guía de auditoria entregada es aplicable a la organización cuando se requiera una nueva evaluación.
- El proceso de pruebas de auditoria sirvió para que los desarrolladores interiorizaran las falencias en el proceso de desarrollo.
- Las recomendaciones emitidas pueden mejorar considerablemente la situación actual de la organización.

TRABAJOS FUTUROS

Continuar con la ejecución de la auditoría al cumplimiento de la política de desarrollo seguro con todos los sistemas de información desarrollados in-house, incluyendo los que se encuentran en desarrollo y en producción.

Realizar la auditoria a los sistemas de información desarrollados por terceros puesto que estos sistemas también deben cumplir con las políticas de seguridad de información de la Escuela Colombiana de ingeniería julio Garavito.

BIBLIOGRAFÍA

Creative Commons Attribution-ShareAlike. (11 noviembre de 2014). OWASP. Recuperado de OWASP: https://www.owasp.org/index.php/Sobre_OWASP

Bernal., J. J. (2017). pdcahome. Recuperado de pdcahome: https://www.pdcahome.com/5202/ciclo-pdca/

Caracola Consultores. (s.f.). ¿Qué son las buenas prácticas?. Recuperado de http://www.planandino.org/bancoBP/node/3#sdfootnote1sym

Dussan Clavijo, C. A. (01 enero de 2006). redalyc.org. Politicas de Seguridad Informatica. Vol 2. Recuperado de redalyc.org: http://www.redalyc.org/html/2654/265420388008/

Definista. (01 junio de 2016). Concepto definicion. Codigo de fuente. Recuperado de: http://conceptodefinicion.de/codigo-fuente/

Deisy Yanez. (s.f.). lifeder. Método descriptivo: características, etapas y ejemplos Recuperado de lifeder: https://www.lifeder.com/metodo-descriptivo/

DeMartine, Amy. (23 enero de 2018). The State Of Application Security, 2018. forrester, 15.

Tovar, E., Carrillo, J., Vega, V., Gasca., G. (2 septiembre de 2006). DESARROLLO DE PRODUCTOS DE SOFTWARE SEGUROS EN SINTONÍA CON LOS MODELOS SSE-CMM, COBIT E ITIL . Vol 3. Universidad Católica del Norte , 8.

Emprendepyme. (s.f.). (2016). emprendepyme. La Auditoria Interna. Recuperado de :https://www.emprendepyme.net/auditoria-interna.html

Erick A. Lamilla Rubio, J. R. (2009). Desarrollo de Políticas de Seguridad Informática e Implementación de Cuatro Dominios en Base a la Norma 27002 para el Área de Hardware en la Empresa Uniplex Systems S.A. en Guayaquil. Escuela Superior Politécnica del

Litoral, 9

Eugenio Duarte. (17 marzo de 2014). capacityacademy. Recuperado de: http://blog.capacityacademy.com/2014/03/17/son-las-politicas-de-seguridad-de-la-tecnologia-de-la-informacion/

Firma-e. (14 octubre de 2014). Obtenido de https://www.firma-e.com/blog/pilares-de-la-seguridad-de-la-informacion-confidencialidad-integridad-y-disponibilidad/

García, L. (07 Marzo de 2010). METODOLOGÍA OSSTMM. Recuperado de http://www.securitybydefault.com/2010/03/metodologia-osstmm.html

Guía de implementación de la facilitación del comercio. (2012). Identifique Oportunidades de Mejora. Recuperado de Identifique Oportunidades de Mejora: http://tfig.unece.org/SP/contents/identify-opportunities-improvement.htm

INCIBE. (20 marzo de 2017). INCIBE. Recuperado de https://www.incibe.es/protege-tu-empresa/blog/amenaza-vs-vulnerabilidad-sabes-se-diferencian

ISOTools Excellence. (21 de 05 de 2015). ISO 27001: ¿Qué significa la Seguridad de la Información?. Recuperado de https://www.pmg-ssi.com/2015/05/iso-27001-que-significa-la-seguridad-de-la-informacion/

José Alberto Ávila Funes. (25 de Abril de 2013). Confidencialidad de la información. Recuperado de http://www.innsz.mx/opencms/contenido/investigacion/comiteEtica/confidencialidadInformacion.html

OWASP. (14 de Abril de 2015). Anexo para Contrato de Software Seguro de OWASP. Recuperado de Anexo para Contrato de Software Seguro de OWASP: https://www.owasp.org/index.php/Anexo_para_Contrato_de_Software_Seguro_de_OWAS

P

OWASP. (04 de 2017). Estándar de Verificación de Seguridad en Aplicaciones.

Provencio, F. L. (15 de enero de 2015). Metodologías para el desarrollo de software seguro. Recuperado de https://upcommons.upc.edu/bitstream/handle/2099.1/24902/103275.pdf

Provencio, F. L. (15 de enero de 2015). Metodologías para el desarrollo de software seguro. Recuperado de Metodologías para el desarrollo de software seguro: https://upcommons.upc.edu/bitstream/handle/2099.1/24902/103275.pdf?sequence=1&isAllowed=y

Prusak, D. y. (1999). Sinnexus. Recuperado de http://www.sinnexus.com/business_intelligence/piramide_negocio.aspx

Segovia, A. J. (2018). 27001academy. ¿Qué es norma ISO 27001?. Recuperado de 27001academy: https://advisera.com/27001academy/es/que-es-iso-27001/

Solarte, f. n. (30 de noviembre de 2011). Auditoría Informática Y De Sistemas. Técnicas E Instrumentos Para Realizar Auditoria Informática Y De Sistemas. Recuperado de http://auditordesistemas.blogspot.com.co/2011/11/tecnicas-e-instrumentos-para-realizar.html

SOLARTE, F. N. (30 de noviembre de 2011). METODOLOGÍA PARA REALIZAR AUDITORÍA. Recuperado de http://auditordesistemas.blogspot.com.co/2011/11/metodologia-para-realizar-auditoria.html.

Universidad de la Punta. (s.f.). Población y Muestra. Recuperado de Población y Muestra: http://contenidosdigitales.ulp.edu.ar/exe/matematica3/poblacin_y_muestra.html

universidades.cr. (2018). Desarrollo de Software. Recuperado de Desarrollo de Software:

https://universidades.cr/carreras/desarrollo-de-software

M.Cornejo Velázquez, I.M. González Cerón, M. N. Guerrero Rubio. (s.f.). Seguridad en Sistemas de Información Transaccionales. Universidad Auntonoma de Hidalgo. Recuperado de https://www.uaeh.edu.mx/scige/boletin/tlahuelilpan/n5/p3.html.

Wayne, R. R. (Octubre de 2012). Threat Models in Social Networks. Recuperado de https://www.researchgate.net/publication/301789336_HACKMI2_Threat_Models_in_Social_NetworksHACKMI2_Threat_Models_in_Social_Networks

Espinoza A, Hans. (Octubre 2013). Lima. Análisis y diseño de un sistema de gestión de seguridad de información basado en la norma ISO/IEC 27001:2005 para una empresa de producción y comercialización de productos de consumo masivo. Pontificia Universidad Católica de Peru. Recuperado de: http://tesis.pucp.edu.pe/repositorio/bitstream/handle/123456789/4957/ESPINOZA_HANS_ANALISIS_SISTEMA_GESTION_SEGURIDAD_INFORMACION_ISO_IEC%202700 1_2005_COMERCIALIZACION_PRODUCTOS_CONSUMO_MASIVO.pdf?sequence= 1

Ladino, M., Villa, P. Lopez, A. (Abril. 2011). FUNDAMENTOS DE ISO 27001 Y SU APLICACIÓN EN LAS EMPRESAS. Universidad Tecnológica de Pereira. ISSN 0122-1701 334. Recuperado de: http://revistas.utp.edu.co/index.php/revistaciencia/article/view/1177/669.

SGSI. (28 de septiembre 2017). .¿Cuál es la situación de la norma ISO 27001 en Sudamérica?. Blog especializado en Sistemas de Gestión de Seguridad de la Información. Recuperado de: https://www.pmg-ssi.com/2017/09/situacion-norma-iso-27001-sudamerica/

OSIRIS. (Abril.2018). Oficina De Sistemas Y Recursos Informaticos. Adquisicion, desarrollo y mantenimiento de sistemas. Manual de Politicas Seguridad y privacidad de la informacion.. (pp.62-66). Bogotà: Escuela Colombiana De Ingenieria Julio Garavito. Recuperado de: https://www.escuelaing.edu.co/escuela/importantDoc/Manual-politica-seguridad-dela-Informacion.pdf

OSIRIS (Abril.2018). Oficina De Sistemas Y Recursos Informaticos. Adquisicion, desarrollo y mantenimiento de sistemas. Manual de Politicas Seguridad y privacidad de la informacion.. (pp. 62). Bogotà: Escuela Colombiana De Ingenieria Julio Garavito. Recuperado de: https://www.escuelaing.edu.co/escuela/importantDoc/Manual-politica-seguridad-dela-Informacion.pdf

J.P. Morgan, Chase Bank N.A, Sucursal Buenos Aires. Recuperado de: https://www.jpmorgan.com/jpmpdf/1320734842796.pdf

INCONTEC. (16 JUNIO DE 2011). Norma Técnica Colombiana. NTC-ISO 31000. GESTION DEL RIESGO PRINCIPIOS Y DIRECTRICES. Recuperado de : https://sitios.ces.edu.co/Documentos/NTC-ISO31000_Gestion_del_riesgo.pdf

Wikipedia. (1 noviembre 2018). Enciclopedia libre. Escuela Colombiana de Ingenieria. Recuperado de: https://es.wikipedia.org/wiki/Escuela_Colombiana_de_Ingenier%C3%ADa

Escuela Colombiana de Ingeniería, (25 de agosto 2008). FILOSOFIA INSTITUCIONAL. Recuperado de : https://www.escuelaing.edu.co/es/conozcanos/filosofia

Escuela Colombiana de Ingeniería, Misión de la escuela. Recuperado de : https://www.escuelaing.edu.co/es/interna/mision/3535

Borghello, Cristian. (2018). Ventajas y desventajas de ISO 27001 en las Pymes. SEGU.INFO. Noticias sobre la seguridad de la información. Recuperado de : https://blog.segu-info.com.ar/2008/03/ventajas-y-desventajas-de-iso27001-en.html

ANEXOS

ANEXO A. Encuestas al equipo de desarrollo

ENCUESTA

Nombre de la Empresa:	Escuela Colombiana de Ingeniería Julio Garavito
Área:	Desarrollo de software
Objetivo:	Conocer el proceso de desarrollo de software con el fin de entender su funcionamiento actual.
Cargo:	Ing. de Desarrollo

Responda a las siguientes preguntas (Marcar solo una alternativa SI o NO)

No.	PREGUNTAS	SI	NO
1	¿Conoce el proceso de desarrollo de software?	X	
2	¿Es clara la metodología de desarrollo de software que se maneja en el área?		X
3	¿Se cumple con la metodología de desarrollo implementada?		X
4	¿Tiene claro su rol y sus tareas asignadas en el equipo de desarrollo?	X	
5	¿Documenta el levantamiento de requerimientos?		X
6	¿El área de desarrollo está cumpliendo con la entrega de los aplicativos solicitados?		X
7	¿Realiza el manual de usuario de los aplicativos que usted desarrolla?	X	
8	¿Realiza el manual de técnico de los aplicativos que usted desarrolla?		X
9	¿Realiza control de versiones a los aplicativos que usted desarrolla?	X	
10	¿Tienen el ambiente de prueba y producción separado?	X	
11	¿Realiza pruebas con el usuario final de los aplicativos desarrollados?	X	

Ilustración 57 Encuesta aplicada

ENCUESTA

Nombre de la Empresa:	Escuela Colombiana de Ingeniería Julio Garavito
Área:	Desarrollo de software
Objetivo:	Conocer el proceso de desarrollo de software con el fin de entender su funcionamiento actual.
Cargo:	Desarrollo

Responda a las siguientes preguntas (Marcar solo una alternativa SI o NO)

No.	PREGUNTAS	SI	NO
1	¿Conoce el proceso de desarrollo de software?	X	
2	¿Es clara la metodología de desarrollo de software que se maneja en el área?		X
3	¿Se cumple con la metodología de desarrollo implementada?		X
4	¿Tiene claro su rol y sus tareas asignadas en el equipo de desarrollo?	X	
5	¿Documenta el levantamiento de requerimientos?	X	
6	¿El área de desarrollo está cumpliendo con la entrega de los aplicativos solicitados?	X	
7	¿Realiza el manual de usuario de los aplicativos que usted desarrolla?		X
8	¿Realiza el manual de técnico de los aplicativos que usted desarrolla?		X
9	¿Realiza control de versiones a los aplicativos que usted desarrolla?		X
10	¿Tienen el ambiente de prueba y producción separado?	X	
11	¿Realiza pruebas con el usuario final de los aplicativos desarrollados?	X	

Ilustración 58 Encuesta aplicada

ENCUESTA

Nombre de la Empresa:	Escuela Colombiana de Ingeniería Julio Garavito
Área:	Desarrollo de software
Objetivo:	Conocer el proceso de desarrollo de software con el fin de entender su funcionamiento actual.
Cargo:	

Responda a las siguientes preguntas (Marcar solo una alternativa SI o NO)

No.	PREGUNTAS	SI	NO
1	¿Conoce el proceso de desarrollo de software?	X	
2	¿Es clara la metodología de desarrollo de software que se maneja en el área?		X
3	¿Se cumple con la metodología de desarrollo implementada?	X	
4	¿Tiene claro su rol y sus tareas asignadas en el equipo de desarrollo?	X	
5	¿Documenta el levantamiento de requerimientos?	X	
6	¿El área de desarrollo está cumpliendo con la entrega de los aplicativos solicitados?	X	
7	¿Realiza el manual de usuario de los aplicativos que usted desarrolla?	X	
8	¿Realiza el manual de técnico de los aplicativos que usted desarrolla?	X	
9	¿Realiza control de versiones a los aplicativos que usted desarrolla?	X	
10	¿Tienen el ambiente de prueba y producción separado?	X	
11	¿Realiza pruebas con el usuario final de los aplicativos desarrollados?	X	

Ilustración 59 Encuesta aplicada

ENCUESTA

Nombre de la Empresa:	Escuela Colombiana de Ingeniería Julio Garavito
Área:	Desarrollo de software
Objetivo:	Conocer el proceso de desarrollo de software con el fin de entender su funcionamiento actual.
Cargo:	

Responda a las siguientes preguntas (Marcar solo una alternativa SI o NO)

No.	PREGUNTAS	SI	NO
1	¿Conoce el proceso de desarrollo de software?	X	
2	¿Es clara la metodología de desarrollo de software que se maneja en el área?	X	
3	¿Se cumple con la metodología de desarrollo implementada?	X	
4	¿Tiene claro su rol y sus tareas asignadas en el equipo de desarrollo?	X	
5	¿Documenta el levantamiento de requerimientos?		X
6	¿El área de desarrollo está cumpliendo con la entrega de los aplicativos solicitados?	X	
7	¿Realiza el manual de usuario de los aplicativos que usted desarrolla?		X
8	¿Realiza el manual de técnico de los aplicativos que usted desarrolla?		X
9	¿Realiza control de versiones a los aplicativos que usted desarrolla?	X	
10	¿Tienen el ambiente de prueba y producción separado?	X	
11	¿Realiza pruebas con el usuario final de los aplicativos desarrollados?	X	

Ilustración 60 Encuesta aplicada

ANEXO B. Encuesta uso de buenas prácticas de desarrollo

Encuesta Desarrolladores

Para las siguientes preguntas, por favor responda con una marca **SI** realiza o **NO** las actividades mencionadas en cada ítem, en caso de no aplicar por favor marque la casilla **NA**

Arquitectura, diseño y modelado de amenazas (v1)

ACTIVIDAD	SI	NO	NA
Verificar que todos los componentes de la aplicación se encuentran identificados y asegurar que son necesarios. SI_ NO		X	
Verificar todos los componentes, tales como bibliotecas, módulos y sistemas externos, que no son parte de la aplicación pero que la misma los necesita para funcionar se han identificado.	X		
Verificar que se ha definido una arquitectura de alto nivel para la aplicación.		X	
Verificar que todos los controles de seguridad (incluyendo las bibliotecas que llaman a servicios de seguridad externos) tienen una implementación centralizada.	X		
Verificar que los componentes están separados unos de otros mediante controles de seguridad, tales como segmentación de la red, reglas de firewall, o grupos de seguridad basados en la nube.	X		
Verificar que la aplicación tiene una clara separación entre la capa de datos, la capa de control y la capa de presentación, tal que las decisiones de seguridad pueden aplicarse en sistemas confiables.	X		
Verificar que no hay ninguna lógica de negocio sensible, claves secretas u otra información propietaria en el código del lado del cliente.	X		

Requisitos de verificación de autenticación (v2)

ACTIVIDAD	SI	NO	NA
Verificar que todas las páginas y recursos requieran autenticación excepto aquellos que sean específicamente destinados a ser públicos (Principio de mediación completa).	X		
Verificar que todos los campos de credenciales no reflejen las contraseñas del usuario. Cargar la credencial por parte de la aplicación implica que la misma fue almacenada de forma reversible o en texto plano, lo que se encuentra explícitamente prohibido.	X		
Verificar que todos los controles de autenticación se realicen del lado del servidor.	X		
Verificar que los controles de autenticación fallan de forma segura para evitar que los atacantes no puedan iniciar sesión.	X		
Verificar que los campos de contraseñas permiten o fomentan el uso de frases como contraseñas (passphrases) y no impiden el uso de gestores de contraseñas, contraseñas largas o altamente complejas.	X		

Verificar que la funcionalidad de cambio de contraseña solicite la contraseña anterior, la nueva contraseña y una confirmación de la contraseña.	X		
Verificar que existen mecanismos de anti-automatización que previenen la verificación de credenciales obtenidas de forma masiva, ataques de fuerza bruta y ataques de bloqueos de cuentas	X		

Requisitos de verificación de gestión de sesiones (v3)

ACTIVIDAD	SI	NO	NA
Verificar que no se utiliza un gestor de sesiones personalizado, o que, si el gestor de sesiones es personalizado, éste sea resistente contra los ataques más comunes.	X		
Verificar que las sesiones se invalidan cuando el usuario cierra la sesión.	X		
Verificar que las sesiones se invalidan luego de un período determinado de inactividad.	X		
Verificar que todas las páginas que requieren autenticación poseen acceso fácil y visible a la funcionalidad de cierre de sesión.	X		
Verificar que el identificador de sesión nunca se revele en URLs, mensajes de error o registros de bitácora. Esto incluye verificar que la aplicación no es compatible con la re-escritura de URL incluyendo el identificador de sesión.	X		
Verificar que toda autenticación exitosa y reautenticaciones generen un nuevo identificador de sesión.	X		
Verificar que los identificadores de sesión son suficientemente largos, aleatorios y únicos para las sesiones activas.	X		

Requisitos de verificación del Control de acceso (v4)

ACTIVIDAD	SI	NO	NA
Verificar que existe el principio de privilegio mínimo - los usuarios sólo deben ser capaces de acceder a las funciones, archivos de datos, URL, controladores, servicios y otros recursos, para los cuales poseen una autorización específica. Esto implica protección contra suplantación de identidad y elevación de privilegios.		X	
Verificar que el acceso a registros sensibles esté protegido, tal que sólo objetos autorizados o datos sean accesibles por cada usuario (por ejemplo, proteger contra la posible manipulación hecha por usuarios sobre un parámetro para ver o modificar la cuenta de otro usuario).	X		
Verificar que la navegación del directorio esté deshabilitada a menos que esto sea deliberadamente deseado. Además, las aplicaciones no deben permitir el descubrimiento o divulgación de metadatos de archivos o directorios, como carpetas que contengan Thumbs.db, DS_Store, o directorios .git o SVN.	X		
Verificar que los controles de acceso fallen de forma segura.	X		
Verificar que las mismas reglas de control de acceso implícitas en la capa de presentación son aplicadas en el servidor.	X		

Verificar que la aplicación aplique correctamente la autorización contextual para no permitir la manipulación de parámetros de la URL.	X		

Requisitos de verificación para Manejo de entrada de datos maliciosos (v5)

ACTIVIDAD	SI	NO	NA
Verificar que el entorno de ejecución no es susceptible a desbordamientos de búfer, o que los controles de seguridad previenen desbordamientos de búfer.	X		
Verificar que las fallas de validación de entradas de datos del lado del servidor sean rechazadas y registradas.	X		
Verificar que se aplican las rutinas de validación de entradas de datos del lado del servidor.	X		
Verificar que todas las consultas de SQL, HQL, OSQL, NOSQL y procedimientos almacenados, llamadas de procedimientos almacenados están protegidos por la utilización de declaraciones preparadas o parametrización de consultas, y por lo tanto no sean susceptibles a la inyección de SQL	X		
Verificar que la aplicación no es susceptible a la inyección de comandos del sistema operativo, o que los controles de seguridad previenen la inyección de comandos del sistema operativo	X		
Verificar que la aplicación no es susceptible a ataques comunes de XML, como manipulación de consultas XPath, ataques de entidad externa XML, y ataques de inyección XML.	X		
Asegurar que todas las variables string utilizadas dentro de HTML u otro lenguaje web interpretado en cliente se encuentra apropiadamente codificada manualmente o se utiliza plantillas que automáticamente codifican contextualmente para asegurar que la aplicación no sea susceptible a ataques DOM Cross-Site Scripting (XSS).	X		
Verificar que HTML no confiable proveniente de editores WYSIWYG o similares sean debidamente sanitizados con un sanitizador de HTML y se manejen apropiadamente según la validación de entrada y codificación.	X		

Requisitos de verificación para la criptografía en el almacenamiento (v7)

ACTIVIDAD	SI	NO	NA
Verificar que todos los módulos criptográficos fallen de forma segura, y que los errores sean manejados de tal manera que no permitan ataques Oracle padding		X	
Verificar que los algoritmos criptográficos utilizados por la aplicación hayan sido validados contra FIPS 140-2 o un estándar equivalente.		X	
Verificar que los módulos criptográficos operen en su modo aprobado según sus políticas de seguridad publicadas.	X		
La información de identificación personal debe almacenarse de forma cifrada y verificar que la comunicación se lleve a cabo utilizando de canales protegidos.	X		
Verificar que contraseñas y claves criptográficas sean sobreescritas con ceros en memoria tan pronto no sean necesarias, con el fin de mitigar ataques de volcado de memoria.		X	

Requisitos de verificación de gestión y registro de errores (v8)

ACTIVIDAD	SI	NO	NA

Verificar que la aplicación no emita mensajes de error o rastros de pilas que contengan datos sensibles que podrían ayudar a un atacante, incluyendo el identificador de sesión, versiones de software/entorno y datos personales.		X	
Verificar que un registro de auditoría o similar permita la no repudiación de transacciones claves.	X		
Verificar que la lógica de manejo de errores en controles de seguridad niegue el acceso por defecto.	X		
Verificar que los controles del registro de seguridad proporcionen la capacidad para registrar los eventos de éxito y sobre todo los eventos de falla que son identificados como relevantes para la seguridad.	X		
Verificar que cada registro de evento incluya la información necesaria para permitir una eventual investigación y correlación con otros eventos.	X		

Requisitos de Verificación de Protección de Datos (v9)

ACTIVIDAD	SI	NO	NA
Verificar que todos los formularios que contengan información sensible se les haya desactivado el almacenamiento de caché en el cliente, incluyendo funciones de autocompletar.		X	
Verificar que toda información sensible es enviada al servidor en el cuerpo o cabeceras del mensaje HTTP (por ejemplo, los parámetros de la URL nunca se deben utilizar para enviar datos sensibles).	X		
Verificar que la aplicación establece encabezados anticaché adecuados según el riesgo de la aplicación, tales como las siguientes: Expires: Tue, 03 Jul 2001 06:00:00 GMTT Last-Modified: {now} GMT Cache-Control: no-store, no-cache, mustrevalidate, max-age=0 Cache-Control: post-check = 0, pre-check = 0 Pragma: no-cache	X		
Verificar que datos almacenados en el cliente (como almacenamiento local de HTML5, almacenamiento de la sesión, IndexedDB, cookies normales o las cookies de Flash) no contengan información sensible o información personal identificable.	X		
Verificar que, en el servidor, todas las copias almacenadas en caché o temporales de datos sensibles estén protegidos de accesos no autorizados o son purgados/invalidados después del acceso por parte del usuario autorizado.	X		
Verificar que la aplicación reduce al mínimo el número de parámetros en una solicitud, como campos ocultos, variables de Ajax, cookies y valores en encabezados.	X		
Verificar que el acceso a datos sensibles es registrado en bitácora, los datos son registrados acorde a las directivas de protección de datos o cuando el registro de los accesos es requerido.	X		
Verificar que existe un mecanismo para eliminar de la aplicación todo tipo de dato sensible luego de transcurrido el tiempo definido por la política de retención.		X	

Requisitos de Verificación de Seguridad de las Comunicaciones (v10)

ACTIVIDAD	SI	NO	NA
Verificar que puede construirse la cadena de confianza desde una CA (Autoridad de Certificación) para cada certificado TLS (Transport Layer Security) del servidor, y que cada certificado del servidor sea válido.		X	

Verificar que se utiliza TLS para todas las conexiones (incluyendo conexiones back-end y externas) autenticadas o que involucran funciones o información sensible, y no recaigan en protocolos inseguros o sin cifrado. Asegúrese de que la alternativa más fuerte es el algoritmo preferido.		X	
Verificar que los encabezados HTTP Strict Transport Security sean incluidos en todas las peticiones y para todos los subdominios, como Strict-Transport-Security: max-age = 15724800; includeSubdomains		X	
Verificar que una adecuada revocación de certificados, tal como el protocolo de estatus de certificado en línea (OSCP), está habilitado y configurado para determinar el estado de vigencia del certificado.		X	
Verificar que se utilicen únicamente algoritmos, cifradores y protocolos fuertes, a través de toda la cadena de confianza, incluyendo certificados raíz y certificados intermediarios de la autoridad certificadora seleccionada.		X	
Verificar que la configuración de TLS esté en línea con las mejores prácticas actuales, particularmente debido a que configuraciones comunes se convierten en inseguras a medida que transcurre el tiempo.		X	

Requisitos de verificación de configuración de seguridad HTTP (v11)

ACTIVIDAD	SI	NO	NA
Verificar que la aplicación acepte solo un conjunto definido de métodos de solicitud HTTP y que son necesarios, como GET y POST, y métodos no utilizados (por ejemplo: TRACE, PUT y DELETE) se encuentran explícitamente bloqueados.		X	
Verificar que cada respuesta HTTP contenga una cabecera content-type en la que se especifique un conjunto utilizando un conjunto de caracteres seguros (Ejemplo: UTF-8, ISO 8859-1).	X		
Verificar que los encabezados HTTP o cualquier parte de la respuesta HTTP no expongan información detallada de la versión de los componentes del sistema.		X	
Verificar que todas las respuestas del API contienen opciones X-Content-Type: nosniff y ContentDisposition: attachment; filename="api.json" (u otro nombre de archivo apropiado para el tipo de contenido).		X	
Verificar que la política de seguridad de contenido (CSPv2) está en uso de tal manera que ayude a mitigar vulnerabilidades de inyección comunes de DOM, XSS, JSON y Javascript		X	
Verificar que el encabezado "X-XSS-Protection: 1; mode=block" esté presente para habilitar a los navegadores a filtrar XSS reflejados		X	

Requisitos de verificación para Controles Malicioso (v13)

ACTIVIDAD	SI	NO	NA
Verificar que toda actividad maliciosa sea adecuadamente aislada o encajonada para retrasar y disuadir a los atacantes de atacar a otras aplicaciones		X	
Verificar que el código fuente de la aplicación y tantas bibliotecas de terceros como sean posibles, no poseen puertas traseras, huevos de pascua, o fallas de lógica en la autenticación, control de acceso, validaciones de entrada y lógica de negocio en transacciones de alto valor.		X	

Requisitos de verificación para lógica de negocios (v15)

ACTIVIDAD	SI	NO	NA
Verificar que la aplicación sólo procese flujos lógicos de negocios en orden secuencial, con todos los pasos procesados en tiempo humano realista, y no procesados fuera de orden, con pasos salteados, con pasos del proceso de otro usuario, o de transacciones muy rápidamente enviadas.		X	
Verificar que la aplicación tiene límites de negocio y los aplique correctamente por cada usuario, con alertas configurables y reacciones automatizadas ante ataques inusuales o automáticos.		X	

Requisitos de verificación de archivos y recursos (v16)

ACTIVIDAD	SI	NO	NA
Verificar que las URL de redireccionen y reenvíen sólo a destinos clasificados en la lista blanca, o mostrar una advertencia cuando se redirija a contenido potencialmente no confiable.		X	
Verificar que archivos no confiables enviados a la aplicación no sean utilizados directamente por comandos de I/O (Entrada/Salida) de archivos, especialmente para proteger contra manipulaciones de rutas, archivo local incluido, manipulación de tipo mime y vulnerabilidades de inyección de comandos de sistema operativo.		X	
Verificar que los archivos procedentes de fuentes no confiables sean validados para ser del tipo del cual se espera y sean analizados por escáneres antivirus para evitar la carga de contenido malicioso conocido.		X	
Verificar que datos no confiables no se utilicen en funcionalidades de reflexión, cargado de clases o inserción para prevenir vulnerabilidades de inclusión de archivos remotos/locales.	X		
Verificar que datos no confiables no se utilicen en recursos de dominios compartidos (CORS) para proteger contra el contenido remoto arbitrario.	X		
Verificar que el código de la aplicación no ejecuta datos cargados obtenidos de fuentes no confiables.		X	
Verificar que no utiliza Flash, Active-X, Silverlight, NACL, Java del lado del cliente u otras tecnologías del lado del cliente que no sean soportadas de forma nativa a través de los estándares de navegador W3C.	X		

Encuesta Desarrolladores

Para las siguientes preguntas, por favor responda con una marca **SI** realiza o **NO** las actividades mencionadas en cada ítem, en caso de no aplicar por favor marque la casilla **NA**

Arquitectura, diseño y modelado de amenazas (v1)

ACTIVIDAD	SI	NO	NA
Verificar que todos los componentes de la aplicación se encuentran identificados y asegurar que son necesarios. SI_ NO	X		
Verificar todos los componentes, tales como bibliotecas, módulos y sistemas externos, que no son parte de la aplicación pero que la misma los necesita para funcionar se han identificado.	X		
Verificar que se ha definido una arquitectura de alto nivel para la aplicación.		X	
Verificar que todos los controles de seguridad (incluyendo las bibliotecas que llaman a servicios de seguridad externos) tienen una implementación centralizada.		X	
Verificar que los componentes están separados unos de otros mediante controles de seguridad, tales como segmentación de la red, reglas de firewall, o grupos de seguridad basados en la nube.	X		
Verificar que la aplicación tiene una clara separación entre la capa de datos, la capa de control y la capa de presentación, tal que las decisiones de seguridad pueden aplicarse en sistemas confiables.	X		
Verificar que no hay ninguna lógica de negocio sensible, claves secretas u otra información propietaria en el código del lado del cliente.	X		

Requisitos de verificación de autenticación (v2)

ACTIVIDAD	SI	NO	NA
Verificar que todas las páginas y recursos requieran autenticación excepto aquellos que sean específicamente destinados a ser públicos (Principio de mediación completa).	X		
Verificar que todos los campos de credenciales no reflejen las contraseñas del usuario. Cargar la credencial por parte de la aplicación implica que la misma fue almacenada de forma reversible o en texto plano, lo que se encuentra explícitamente prohibido.	X		
Verificar que todos los controles de autenticación se realicen del lado del servidor.	X		
Verificar que los controles de autenticación fallan de forma segura para evitar que los atacantes no puedan iniciar sesión.	X		
Verificar que los campos de contraseñas permiten o fomentan el uso de frases como contraseñas (passphrases) y no impiden el uso de gestores de contraseñas, contraseñas largas o altamente complejas.		X	

Verificar que la funcionalidad de cambio de contraseña solicite la contraseña anterior, la nueva contraseña y una confirmación de la contraseña.	X		
Verificar que existen mecanismos de anti-automatización que previenen la verificación de credenciales obtenidas de forma masiva, ataques de fuerza bruta y ataques de bloqueos de cuentas		X	

Requisitos de verificación de gestión de sesiones (v3)

ACTIVIDAD	SI	NO	NA
Verificar que no se utiliza un gestor de sesiones personalizado, o que, si el gestor de sesiones es personalizado, éste sea resistente contra los ataques más comunes.		X	
Verificar que las sesiones se invalidan cuando el usuario cierra la sesión.	X		
Verificar que las sesiones se invalidan luego de un período determinado de inactividad.	X		
Verificar que todas las páginas que requieren autenticación poseen acceso fácil y visible a la funcionalidad de cierre de sesión.	X		
Verificar que el identificador de sesión nunca se revele en URLs, mensajes de error o registros de bitácora. Esto incluye verificar que la aplicación no es compatible con la re-escritura de URL incluyendo el identificador de sesión.	X		
Verificar que toda autenticación exitosa y reautenticaciones generen un nuevo identificador de sesión.	X		
Verificar que los identificadores de sesión son suficientemente largos, aleatorios y únicos para las sesiones activas.		X	

Requisitos de verificación del Control de acceso (v4)

ACTIVIDAD	SI	NO	NA
Verificar que existe el principio de privilegio mínimo - los usuarios sólo deben ser capaces de acceder a las funciones, archivos de datos, URL, controladores, servicios y otros recursos, para los cuales poseen una autorización específica. Esto implica protección contra suplantación de identidad y elevación de privilegios.	X	-	
Verificar que el acceso a registros sensibles esté protegido, tal que sólo objetos autorizados o datos sean accesibles por cada usuario (por ejemplo, proteger contra la posible manipulación hecha por usuarios sobre un parámetro para ver o modificar la cuenta de otro usuario).	X		
Verificar que la navegación del directorio esté deshabilitada a menos que esto sea deliberadamente deseado. Además, las aplicaciones no deben permitir el descubrimiento o divulgación de metadatos de archivos o directorios, como carpetas que contengan Thumbs.db, DS_Store, o directorios .git o SVN.		X	
Verificar que los controles de acceso fallen de forma segura.	X		
Verificar que las mismas reglas de control de acceso implícitas en la capa de presentación son aplicadas en el servidor.	X		

Verificar que la aplicación aplique correctamente la autorización contextual para no permitir la manipulación de parámetros de la URL.		X	

Requisitos de verificación para Manejo de entrada de datos maliciosos (v5)

ACTIVIDAD	SI	NO	NA
Verificar que el entorno de ejecución no es susceptible a desbordamientos de búfer, o que los controles de seguridad previenen desbordamientos de búfer.		X	
Verificar que las fallas de validación de entradas de datos del lado del servidor sean rechazadas y registradas.	X		
Verificar que se aplican las rutinas de validación de entradas de datos del lado del servidor.	X		
Verificar que todas las consultas de SQL, HQL, OSQL, NOSQL y procedimientos almacenados, llamadas de procedimientos almacenados están protegidos por la utilización de declaraciones preparadas o parametrización de consultas, y por lo tanto no sean susceptibles a la inyección de SQL	X		
Verificar que la aplicación no es susceptible a la inyección de comandos del sistema operativo, o que los controles de seguridad previenen la inyección de comandos del sistema operativo	X		
Verificar que la aplicación no es susceptible a ataques comunes de XML, como manipulación de consultas XPath, ataques de entidad externa XML, y ataques de inyección XML.		X	
Asegurar que todas las variables string utilizadas dentro de HTML u otro lenguaje web interpretado en cliente se encuentra apropiadamente codificada manualmente o se utiliza plantillas que automáticamente codifican contextualmente para asegurar que la aplicación no sea susceptible a ataques DOM Cross-Site Scripting (XSS).		X	
Verificar que HTML no confiable proveniente de editores WYSIWYG o similares sean debidamente sanitizados con un sanitizador de HTML y se manejen apropiadamente según la validación de entrada y codificación.		X	

Requisitos de verificación para la criptografía en el almacenamiento (v7)

ACTIVIDAD	SI	NO	NA
Verificar que todos los módulos criptográficos fallen de forma segura, y que los errores sean manejados de tal manera que no permitan ataques Oracle padding		X	
Verificar que los algoritmos criptográficos utilizados por la aplicación hayan sido validados contra FIPS 140-2 o un estándar equivalente.		X	
Verificar que los módulos criptográficos operen en su modo aprobado según sus políticas de seguridad publicadas.		X	
La información de identificación personal debe almacenarse de forma cifrada y verificar que la comunicación se lleve a cabo utilizando de canales protegidos.		X	
Verificar que contraseñas y claves criptográficas sean sobreescritas con ceros en memoria tan pronto no sean necesarias, con el fin de mitigar ataques de volcado de memoria.		X	

Requisitos de verificación de gestión y registro de errores (v8)

ACTIVIDAD	SI	NO	NA

Verificar que la aplicación no emita mensajes de error o rastros de pilas que contengan datos sensibles que podrían ayudar a un atacante, incluyendo el identificador de sesión, versiones de software/entorno y datos personales.	X		
Verificar que un registro de auditoría o similar permita la no repudiación de transacciones claves.		X	
Verificar que la lógica de manejo de errores en controles de seguridad niegue el acceso por defecto.	X		
Verificar que los controles del registro de seguridad proporcionen la capacidad para registrar los eventos de éxito y sobre todo los eventos de falla que son identificados como relevantes para la seguridad.		X	
Verificar que cada registro de evento incluya la información necesaria para permitir una eventual investigación y correlación con otros eventos.		X	

Requisitos de Verificación de Protección de Datos (v9)

ACTIVIDAD	SI	NO	NA
Verificar que todos los formularios que contengan información sensible se les haya desactivado el almacenamiento de caché en el cliente, incluyendo funciones de autocompletar.		X	
Verificar que toda información sensible es enviada al servidor en el cuerpo o cabeceras del mensaje HTTP (por ejemplo, los parámetros de la URL nunca se deben utilizar para enviar datos sensibles).	X		
Verificar que la aplicación establece encabezados anticaché adecuados según el riesgo de la aplicación, tales como las siguientes: Expires: Tue, 03 Jul 2001 06:00:00 GMTT Last-Modified: {now} GMT Cache-Control: no-store, no-cache, mustrevalidate, max-age=0 Cache-Control: post-check = 0, pre-check = 0 Pragma: no-cache		X	
Verificar que datos almacenados en el cliente (como almacenamiento local de HTML5, almacenamiento de la sesión, IndexedDB, cookies normales o las cookies de Flash) no contengan información sensible o información personal identificable.		X	
Verificar que, en el servidor, todas las copias almacenadas en caché o temporales de datos sensibles estén protegidos de accesos no autorizados o son purgados/invalidados después del acceso por parte del usuario autorizado.	X		
Verificar que la aplicación reduce al mínimo el número de parámetros en una solicitud, como campos ocultos, variables de Ajax, cookies y valores en encabezados.	X		
Verificar que el acceso a datos sensibles es registrado en bitácora, los datos son registrados acorde a las directivas de protección de datos o cuando el registro de los accesos es requerido.		X	
Verificar que existe un mecanismo para eliminar de la aplicación todo tipo de dato sensible luego de transcurrido el tiempo definido por la política de retención.		X	

Requisitos de Verificación de Seguridad de las Comunicaciones (v10)

ACTIVIDAD	SI	NO	NA
Verificar que puede construirse la cadena de confianza desde una CA (Autoridad de Certificación) para cada certificado TLS (Transport Layer Security) del servidor, y que cada certificado del servidor sea válido.	X		

Verificar que se utiliza TLS para todas las conexiones (incluyendo conexiones back-end y externas) autenticadas o que involucran funciones o información sensible, y no recaigan en protocolos inseguros o sin cifrado. Asegúrese de que la alternativa más fuerte es el algoritmo preferido.		X	
Verificar que los encabezados HTTP Strict Transport Security sean incluidos en todas las peticiones y para todos los subdominios, como Strict-Transport-Security: max-age = 15724800; includeSubdomains		X	
Verificar que una adecuada revocación de certificados, tal como el protocolo de estatus de certificado en línea (OSCP), está habilitado y configurado para determinar el estado de vigencia del certificado.		X	
Verificar que se utilicen únicamente algoritmos, cifradores y protocolos fuertes, a través de toda la cadena de confianza, incluyendo certificados raíz y certificados intermediarios de la autoridad certificadora seleccionada.		X	
Verificar que la configuración de TLS esté en línea con las mejores prácticas actuales, particularmente debido a que configuraciones comunes se convierten en inseguras a medida que transcurre el tiempo.		X	

Requisitos de verificación de configuración de seguridad HTTP (v11)

ACTIVIDAD	SI	NO	NA
Verificar que la aplicación acepte solo un conjunto definido de métodos de solicitud HTTP y que son necesarios, como GET y POST, y métodos no utilizados (por ejemplo: TRACE, PUT y DELETE) se encuentran explícitamente bloqueados.	X		
Verificar que cada respuesta HTTP contenga una cabecera content-type en la que se especifique un conjunto utilizando un conjunto de caracteres seguros (Ejemplo: UTF-8, ISO 8859-1).		X	
Verificar que los encabezados HTTP o cualquier parte de la respuesta HTTP no expongan información detallada de la versión de los componentes del sistema.	X		
Verificar que todas las respuestas del API contienen opciones X-Content-Type: nosniff y ContentDisposition: attachment; filename="api.json" (u otro nombre de archivo apropiado para el tipo de contenido).		X	
Verificar que la política de seguridad de contenido (CSPv2) está en uso de tal manera que ayude a mitigar vulnerabilidades de inyección comunes de DOM, XSS, JSON y Javascript		X	
Verificar que el encabezado "X-XSS-Protection: 1; mode=block" esté presente para habilitar a los navegadores a filtrar XSS reflejados		X	

Requisitos de verificación para Controles Malicioso (v13)

ACTIVIDAD	SI	NO	NA
Verificar que toda actividad maliciosa sea adecuadamente aislada o encajonada para retrasar y disuadir a los atacantes de atacar a otras aplicaciones		X	
Verificar que el código fuente de la aplicación y tantas bibliotecas de terceros como sean posibles, no poseen puertas traseras, huevos de pascua, o fallas de lógica en la autenticación, control de acceso, validaciones de entrada y lógica de negocio en transacciones de alto valor.	X		

Requisitos de verificación para lógica de negocios (v15)

ACTIVIDAD	SI	NO	NA
Verificar que la aplicación sólo procese flujos lógicos de negocios en orden secuencial, con todos los pasos procesados en tiempo humano realista, y no procesados fuera de orden, con pasos salteados, con pasos del proceso de otro usuario, o de transacciones muy rápidamente enviadas.	X		
Verificar que la aplicación tiene límites de negocio y los aplique correctamente por cada usuario, con alertas configurables y reacciones automatizadas ante ataques inusuales o automáticos.	X		

Requisitos de verificación de archivos y recursos (v16)

ACTIVIDAD	SI	NO	NA
Verificar que las URL de redireccionen y reenvíen sólo a destinos clasificados en la lista blanca, o mostrar una advertencia cuando se redirija a contenido potencialmente no confiable.	X		
Verificar que archivos no confiables enviados a la aplicación no sean utilizados directamente por comandos de I/O (Entrada/Salida) de archivos, especialmente para proteger contra manipulaciones de rutas, archivo local incluido, manipulación de tipo mime y vulnerabilidades de inyección de comandos de sistema operativo.	X		
Verificar que los archivos procedentes de fuentes no confiables sean validados para ser del tipo del cual se espera y sean analizados por escáneres antivirus para evitar la carga de contenido malicioso conocido.		X	
Verificar que datos no confiables no se utilicen en funcionalidades de reflexión, cargado de clases o inserción para prevenir vulnerabilidades de inclusión de archivos remotos/locales.		X	
Verificar que datos no confiables no se utilicen en recursos de dominios compartidos (CORS) para proteger contra el contenido remoto arbitrario.		X	
Verificar que el código de la aplicación no ejecuta datos cargados obtenidos de fuentes no confiables.	X		
Verificar que no utiliza Flash, Active-X, Silverlight, NACL, Java del lado del cliente u otras tecnologías del lado del cliente que no sean soportadas de forma nativa a través de los estándares de navegador W3C.		X	