**Universidad de las Ciencias Informáticas**

**Facultad 4**

# Título: Sistema de gestión para las evidencias de los objetivos del año

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

**Autores:**

Javier Ceballo Perez

Arián León Benítez

**Tutor:** M. Sc. Yordankis Matos López

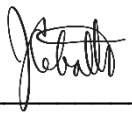
"La Habana, 2022"

*Está bien celebrar el éxito, pero es más importante prestar atención a las lecciones del fracaso.*

*Bill Gates*

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis que tiene por título: “Sistema de gestión para las evidencias de los objetivos del año” y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmo la presente a los 8 días del mes de diciembre del año 2022.

| **Javier Ceballo Perez** | **Arián León Benítez** | **Yordankis Matos López** |
| --- | --- | --- |
|  |  |  |
| Firma del Autor | Firma del Autor | Firma del Tutor |

## DEDICATORIAS

*Javier*

*Dedico esta tesis principalmente a mi madre Aracely, motivo de mi inspiración. A mi padre Julio y mi hermano Julio Cesar que siempre conté con su apoyo. A mi abuela Mima, mi abuelo Titi y demás familiares.*

*Arián*

*Dedico esta tesis a mi abuela Maximina y a la abuela que la vida me dio Martha Pastrana.*

**AGRADECIMIENTOS**

*Javier*

*Quiero darle las gracias a mis padres y a mi hermano por todo su apoyo durante mi etapa universitaria. A mis amigos Enriquito y Hemsel por siempre estar pendientes, a mi amigo Lacho que estuvo presente desde el primer hasta el último día en estos cinco años, a mi compañero de tesis y amigo Arian que siempre estuvo ahí en cada momento de desarrollo de esta tesis. Agradecer también a mi tutor Yordankis y a la profesora Yadira.*

*En fin, agradecer a todos los que de una forma u otra me han apoyado.*

*Arián*

*Quiero agradecer a mi familia y principalmente a mi mamá, mi papá y a mi tía Maitte por hacer posible mi camino por la universidad. A todos mis amigos que no voy a mencionar para que no quede nadie afuera (ellos saben que son ellos a los que me refiero). A mi compañero de tesis y amigo Javier por hacer tan ameno y divertido nuestro proceso de tesis. Y por último a nuestro Tutor Yordankis y a esos profesores que sin ser tutores nuestros siempre nos apoyaron y ayudaron.*

## Resumen

La eficiente y eficaz gestión de organizaciones, es un proceso que tiene como finalidad la optimización de los recursos y la maximización de los beneficios sean estos sociales, económicos y/o medio ambientales. Para alcanzar esas metas han surgido diferentes técnicas o herramientas que constituyen el cuerpo académico-científico de la gestión empresarial. La administración por objetivos constituye una de estas técnicas y permite la evaluación, medición y seguimiento de los resultados del trabajo en las organizaciones. La Universidad de las Ciencias Informáticas en su misión de formar profesionales altamente calificados y producir aplicaciones y servicios informáticos, define anualmente sus objetivos estratégicos agrupados en áreas de resultados claves, para cada una de las facultades que componen su estructura organizacional. El objetivo de este trabajo es desarrollar un sistema que permita la gestión de las evidencias que contribuyen al cumplimiento de los objetivos estratégicos del año. El sistema informático obtenido como resultado de este trabajo permite gestionar las evidencias generadas a lo largo del año y controlar el cumplimiento de los objetivos estratégicos. Además, está basado totalmente en tecnologías de software libre. Otra contribución importante es el apoyo a la toma de decisiones, pues permite identificar a tiempo posibles atrasos en el cumplimiento de los mismos.

**Palabras clave:** evidencias, gestión, objetivos estratégicos, sistema.

**Índice**

ÍNDICE DE ILUSTRACIONES



ÍNDICE DE TABLAS

## INTRODUCCIÓN

El uso de las Tecnologías de la Información y las Comunicaciones (TIC) ha significado a escala mundial un salto vertiginoso en el desarrollo científico técnico. Desde su llegada a los escenarios nacionales se ha convertido en un elemento indispensable para establecer las líneas de desarrollo de la sociedad cubana. El empleo de las TIC mejora la manera de trabajar del hombre, en tanto ofrece soluciones que contribuyen a potenciar los recursos disponibles y a facilitar y optimizar los procesos productivos.

En la actualidad, muchas organizaciones trabajan con una cantidad de información en continuo crecimiento y renovación, debido a esto, la gestión de sus procesos cada vez se hace más engorroso. La informatización de procesos es un aspecto que brinda soluciones automatizadas a flujos de trabajos manuales por lo que debería ser tenido en cuenta por las organizaciones cubanas. Este factor no solo es necesario para cumplir con las normas que regulan cada sector, sino que permite lograr una mayor productividad y competitividad; además de ayudar en las tareas de gestión por su aporte de información clave para la toma de decisiones estratégicas.

Actualmente, numerosas organizaciones están vinculadas al uso de los sistemas de gestión (SG). Existen diversos SG desde los más simples hasta los más sofisticados, estos últimos no solo manejan documentación, sino que permiten informatizar procesos internos dentro de las entidades u organizaciones. Estos sistemas se deben aplicar en los procesos de planificación para complementar, mejorar el análisis y las decisiones con el fin de lograr los objetivos estratégicos trazados y a largo plazo la posición de la organización, es decir, son los resultados que se esperan alcanzar en un año.

Estos objetivos estratégicos son metas a corto y largo plazo por lo que son de gran importancia para evaluar el desempeño de la organización. Se controlan mediante un proceso administrativo donde los trabajadores con un alto nivel jerárquico definen las decisiones correspondientes a la manera en que la organización funciona.

En la Universidad de Ciencias Informáticas (UCI), en la Facultad 4, se definen y aprueban los objetivos estratégicos del año en el Consejo de Dirección de la facultad. Estos objetivos se aplican a las distintas Áreas de Resultado Clave (ARC) de la organización. Deben ser planificados, realizados y controlados con el fin de conseguir los resultados que se proponen. Cada objetivo contiene uno o varios criterios de medida y a estos a su vez se le asocia un indicador para lograr el cumplimiento y control de los criterios de medida. El control del

cumplimiento de estos se realiza mediante herramientas de ofimática, consultando datos extraídos de otros sistemas con objetivos diferentes y datos enviados por correos electrónico, entre otros.

La facultad la integran profesores y especialistas, los cuales deben realizar un conjunto de actividades que pueden generar evidenciadas para cumplir con cada indicador que le fue asignado. Una vez terminado el año se conforma un documento por cada profesor o especialista a partir del cumplimiento o no de los indicadores y se emite una evaluación. Este documento lo supervisa el jefe de cada ARC, el cual lo entrega al decano de la facultad.

También, al terminar el año se elabora un documento por cada ARC a partir de los objetivos alcanzados y los criterios de medida que fueron cumplidos. Este documento también lo supervisa el jefe de cada área, el cual lo entrega al decano de la facultad.

Las evidencias de los indicadores para lograr el cumplimiento de los Criterios de medida y los Objetivos requieren de una previa consulta entre las partes involucradas lo cual se realiza a través de reuniones, despachos, y en ocasiones por medio de correo electrónico y llamadas telefónicas. Los datos obtenidos en el proceso suelen estar duplicados, desactualizados o hasta puede existir pérdida de información ya que no están centralizados. Esto trae consigo que el control y el monitoreo continuo del proceso se dificulte, situación que propicia que no se tomen las medidas correctivas a tiempo y abra las puertas al incumplimiento de algunos objetivos planeados.

A partir de la situación problemática descrita anteriormente se plantea como **problema de investigación** ¿Cómo mejorar el proceso de gestión de evidencias para contribuir al cumplimiento de los objetivos estratégico del año?

A partir del planteamiento del problema de investigación se tiene como **objeto de estudio** de la investigación: el proceso de gestión de evidencias para los objetivos estratégicos del año; enmarcado en el **campo de acción**: los sistemas de gestión para las evidencias de los objetivos estratégicos del año.

El **objetivo general** de este trabajo de diploma es desarrollar una aplicación web que permita gestionar la elaboración y chequear el cumplimiento de los objetivos estratégicos que se tracen en la facultad para un año determinado mediante la gestión de evidencias, empleando tecnologías y herramientas de software libre.

Para llegar a la solución del objetivo planteado se tomaron las siguientes tareas de investigación

- Elaboración del marco teórico de la investigación mediante el estudio del estado actual de las herramientas existentes en el mercado, dirigidas a los sistemas de gestión por objetivo de organizaciones.
- Selección de las herramientas a utilizar para desarrollar una propuesta de solución.
- Realización de los artefactos ingenieriles a partir de la metodología de desarrollo de software seleccionada.
- Implementación del sistema propuesto que permita gestionar tanto la elaboración como el chequeo del cumplimiento de los objetivos estratégicos que se tracen en la facultad para un año determinado mediante evidencias.
- Ejecución de las pruebas para garantizar la calidad del software.

**Métodos teóricos:**

- **Analítico-Sintético:** este método se utilizó para llegar a conclusiones de la investigación que se desarrolla. Mediante este se busca, investiga y se realiza un análisis de documentos, publicaciones, bibliografías e informaciones relacionadas con el objeto de estudio.

- **Modelación:** se utilizó para realizar un modelado del sistema web a desarrollar, donde fue necesario explicarle al cliente mediante modelos cómo se previó la realización del sistema y si cumplía con sus necesidades.

**Métodos Empíricos:**

- **Entrevista:** se utilizó para realizar entrevistas al personal de la institución con el objetivo de obtener información precisa sobre sus principales problemas, necesidades y posibles resultados esperados. La entrevista se encuentra en el anexo 7.

- **Observación:** se utilizó para determinar cómo se realiza todo el proceso actual de la gestión de las evidencias para lograr los objetivos estratégicos.

La estructura del presente trabajo de diploma está compuesta por los siguientes capítulos:

**Capítulo 1: Fundamentación teórica**

En este capítulo se especifican los principales conceptos relacionados con el tema, se realiza el estudio del estado del arte a nivel nacional e internacional sobre algunos sistemas de gestión que son homólogos con el sistema a desarrollar. Además, se definen las herramientas, metodología y tecnologías a utilizar durante el desarrollo del sistema.

**Capítulo 2: Análisis y Diseño de la propuesta de solución**

En este capítulo se exponen los aspectos relacionados con la fase de planificación y de diseño de la metodología seleccionada. Se realiza una estimación de esfuerzos que permite dar una idea de la duración del proceso de desarrollo. Además, se describe la propuesta de solución y se definen los requisitos del sistema y las Historias de Usuarios.

**Capítulo 3: Implementación y pruebas**

En este capítulo se abordan aspectos relacionados con la implementación del sistema en base a la metodología de desarrollo donde se precisan las tareas de ingeniería. Se documentan las pruebas de software al sistema de la gestión de las evidencias y el cumplimiento de los objetivos estratégicos del año, realizándose las pruebas de aceptación y pruebas unitarias para verificar que responda a un correcto funcionamiento, garantizar la confidencialidad y detectar fallas en el sistema.

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En este capítulo se exponen algunas definiciones y elementos teóricos necesarios para el desarrollo del sistema en cuestión. También se realiza un estudio de los sistemas homólogos a la propuesta de solución, así como se analizan y se justifican las tecnologías, herramientas y la metodología que serán usadas.

## 1.1 Conceptos fundamentales

Se hace necesario un conjunto de definiciones que conformarán el sustento teórico de la presente investigación. De esta manera se facilitará una mejor comprensión de los elementos que integran el diseño del sistema a desarrollar.

Los objetivos estratégicos según Walter Andía Valencia son objetivos de mediano y largo plazo, orientados al logro de la misión de la organización. Son los resultados más relevantes y de mayor nivel que la institución espera lograr para cumplir con su misión (Andía Valencia 2016).

Según Esperanza Carballal del Río las ARC son, simplemente, áreas o categorías esenciales para el rendimiento efectivo en la institución o empresa. Los logros, dentro de estas áreas, son decisivos para que la institución o empresa lleve a cabo con éxito su Misión y se cumpla con las expectativas generadas. Las ARC no cubren todo lo que logrará la organización. Identifican los encabezados generales. Estas son áreas donde el desempeño es vital para la situación y la supervivencia a largo plazo de la empresa (R&iacute 2001).

La administración por objetivos (APO) según Idalberto Chiavenato es un método por el cual el gerente y sus subordinados definen las metas en conjunto; las responsabilidades se especifican para cada uno en función de los resultados esperados, que constituyen los indicadores o patrones de desempeño bajo los cuales se evaluará a ambos. Analizando el resultado final, el desempeño del gerente y del subordinado puede evaluarse objetivamente y los resultados alcanzados se pueden comparar con los esperados (Chiavenato 2019).

Peter Drucker, considerado el mayor filósofo de la administración, plantea: *“La administración por objetivos, determina objetivos conjuntos y proporcionan retroalimentación sobre los resultados. Establecer objetivos desafiantes pero alcanzables, promueve la motivación y el empoderamiento de los empleados.”*

Por otra parte, un SG según José F. Naranjo es considerado como una serie de procesos, acciones y tareas que se llevan a cabo sobre un conjunto de elementos (personas, procedimientos, estrategias, planes, recursos, productos, etc.) para lograr el éxito sostenido de una organización, es decir, disponer de capacidad para satisfacer las necesidades y las expectativas de sus clientes o beneficiarios, trabajadores y de otras partes interesadas a largo plazo y de un modo equilibrado y sostenible (Naranjo 2015).

Según la Real Academia Española (RAE) una evidencia es: “certeza clara y manifiesta de lo que no se puede dudar”, entre otras aceptaciones (ASALE y RAE 2022).

Los autores de esta investigación definen una evidencia dentro de los objetivos estratégicos como un documento digital que respalda el conocimiento, la certeza, convencimiento de una aceptación y/o participación en conferencias científicas, competencias académicas, concursos, eventos, capacitaciones, cursos de Postgrado, publicaciones en diversas fuentes, entre otros, para lograr el cumplimiento de los objetivos estratégicos.

### 1.2 Análisis de sistemas homólogos

En la actualidad existen variedad de sistemas dedicados a la gestión de todo tipo de información. Para lograr una mejor comprensión de las características del sistema a desarrollar, es necesario realizar el análisis de algunos sistemas con algunas similitudes al propuesto en la presente investigación. A continuación, se expone el análisis desarrollado de los sistemas homólogos.

#### 1.2.1 Sistemas homólogos internacionales

**Alfresco**

Es un sistema de gestión de contenidos de código abierto que permite a las organizaciones capturar, almacenar, buscar y colaborar en documentos de muchos tipos distintos. Está desarrollado en Java y basado en estándares abiertos. Es utilizado como software de gestión documental para documentos, páginas web, registros, imágenes, entre otros. Entre sus características principales se encuentra la gestión de documentos, la gestión de imágenes y la gestión de registros. Alfresco fue fundado en 2005 por John Newton, cofundador de Documentum y John Powell, ex COO de Business Objects (Shariff 2009).

**Orfeo**

Es un sistema de gestión documental y de procesos licenciado como software libre bajo licencia GNU/GPL para compartir el conocimiento y mantener la creación colectiva. Permite incorporar la gestión de los documentos a los procesos de cualquier organización, automatizando procedimientos, con importantes ahorros en tiempo, costos y recursos, así como el control sobre los documentos. Esta herramienta puede instalarse en cualquier sistema Operativo (GNU/Linux, Unix, Windows, …), con diferentes bases de datos (PostgreSQL, Oracle y MS SQL Server) (Losada 2017).

### 1.2.2 Sistemas homólogos nacionales

**Sistema Automatizado para la Gestión Académica (AKADEMOS)**

El software, construido por un equipo de trabajo de la Dirección de Informatización de la UCI. Es un sistema web distribuido, de software libre y desarrollado en la plataforma .NET, que utiliza SQL Server 2000 para el almacenamiento de los datos, Internet Information Services (IIS) como servidor Web y Servicios WEB XML para el intercambio con otras aplicaciones (Costa 2007).

Sus funcionalidades se agrupan en siete módulos, de los cuales el Plan de Estudio es la entidad fundamental, pues rige todos los subprocesos (ver Anexo1). Estos módulos son:

- **Módulo de plan de estudio:** permite la gestión de diferentes especialidades o carreras al definir los planes de estudio.
- **Módulo de matrícula:** se encarga del control de los datos de los estudiantes y la gestión de los movimientos a que son sometidos.
- **Módulo de expediente:** actúa como un repositorio digital de los documentos y la información histórica de los estudiantes, disponible en todo momento.
- **Módulo de registro:** permite el control del desarrollo de un período académico con el registro de las evaluaciones y la asistencia.
- **Módulo de profesor:** mantiene un control de la plantilla de profesores.
- **Módulo de reportes:** obtiene reportes ordenados de la información almacenada en el sistema.

- **Módulo de estudiantes:** ofrece el historial académico de los años cursados por los estudiantes.

**XEDRO GESPRO**

Es una Suite orientada a la web, de software libre, que permite la planificación, seguimiento y control de productos en forma de proyectos. Cuenta con herramientas para el apoyo a la toma de decisiones a nivel de proyecto, nivel de entidad ejecutora y nivel gerencial. Se presenta en un modelo de negocios basado en servicios que combinan el uso de la solución informática para la dirección integrada de proyectos y un sistema de formación especializada en gestión de proyectos (UCI 2022).

La solución informática desarrollada bajo un ecosistema de software basado en tecnologías libres permite:

- La planificación y el control y seguimiento de los proyectos y de los recursos asociados a los mismos, alineadas con la proyección estratégica de las organizaciones.
- La planificación del alcance y el tiempo, la gestión de recursos humanos y sus competencias, la gestión de riesgos, así como la financiera de los proyectos. Gestión logística y gestión de recursos compartidos.
- El control y seguimiento de proyectos a través de la combinación de un cuadro de mando integral y un sistema para el diseño dinámico de reportes que permiten el acceso a la información del estado de los proyectos con diferentes niveles de detalles de la información.
- Gestión documental con facilidades para la gestión del expediente de los proyectos.
- Gestión de contratos y de interesados en los proyectos, que permite tanto la gestión de acuerdos con clientes como con los proveedores y garantiza integrar el control y el seguimiento de los compromisos alineados completamente con la información del estado de los proyectos (UCI 2015).

**Trabajos de diplomas**

Se realiza un análisis de los trabajos de diploma “Sistema de Gestión para el cumplimiento de los objetivos estratégicos de la Facultad 4” (Céspedes Calzado 2019) y “Sistema para la Gestión de Evidencias de Indicadores de CTI, Posgrado e Investigación (GEVIN)” (González Martínez, González Garay y Espinosa Marrero 2015), ambos con temas semejantes a la presente

investigación, uno de ellos sirvió de apoyo en la forma en que plantea la organización y estructuración referentes a las ARC y sus objetivos estratégicos; en el segundo se hace interesante investigar toda la gestión documental referente a las evidencias. Debido a este análisis se logra confirmar que los sistemas generados por ambos trabajos no se encuentran desplegados o en funcionamiento actualmente.

**Conclusiones del estudio de los sistemas**

Cabe destacar que, en el caso de Alfresco y Orfeo, aunque cuentan con múltiples características, están orientados a la gestión documental, lo cual no se ajusta en gran medida a la situación problemática ya que sus finalidades son otras. Por otra parte, AKADEMOS permite la validación y el reconocimiento oficial de la actividad científica – investigativa de profesores y especialistas en la UCI y puede actuar como un repositorio digital de estos documentos, entre otras cosas, pero no permite la gestión de los objetivos estratégicos trazados en un año por una facultad y lograr el cumplimiento de estos mediante indicadores asignados a cada profesor o especialista. Por último, XEDRO GESPRO, el cual posibilita la planificación, seguimiento y control de productos en forma de proyectos, a la vez que permite la evaluación de actividades según indicadores y aunque es una buena opción para satisfacer las necesidades de las facultades, este está orientado a proyectos y no a objetivos estratégicos por lo que no es una solución factible al problema planteado.

Debido a que ninguno de los sistemas antes mencionados satisface las necesidades actuales ni le da solución al problema de la investigación en curso, y los sistemas desarrollados por los trabajos de diplomas estudiados no se encuentran en funcionamiento, se decidió la implementación de un sistema informático que permita gestionar las evidencias para contribuir al cumplimiento de los objetivos estratégicos trazados en un año. Se toma la decisión de hacer una aplicación web debido a las ventajas que este ofrece, pues se podrá usar desde cualquier ordenador de la institución, solo haciendo uso del navegador, por lo que se obtiene una solución multiplataforma.

A continuación, se realiza un estudio de las herramientas y tecnologías que serán seleccionadas para el desarrollo de la solución, así como la metodología que guiará el proceso de desarrollo.

### 1.3 Tecnologías y herramientas

Para desarrollar el sistema informático propuesto en la presente investigación, se hace necesario indagar sobre los lenguajes, herramientas y tecnologías utilizados en el desarrollo de la propuesta de solución. A continuación, se procede con un estudio de estos.

#### 1.3.1 Lenguajes

Un lenguaje de programación es una herramienta que permite desarrollar software o programas para computadora. Los lenguajes de programación son empleados para diseñar e implementar programas encargados de definir y administrar el comportamiento de los dispositivos físicos y lógicos de una computadora. Lo anterior se logra mediante la creación e implementación de algoritmos de precisión que se utilizan como una forma de comunicación humana con la computadora (CUAED 2018).

**HTML**

Lenguaje de Marcas de Hipertextos, del inglés HyperText Markup Lenguage (HTML) es el componente más básico de la web. Define el significado y la estructura del contenido web. “Hipertexto” hace referencia a los enlaces que conectan páginas web entre sí, ya sea dentro de un único sitio web o entre sitios web. Los enlaces son un aspecto fundamental de la web. HTML utiliza "marcas" para etiquetar texto, imágenes y otro contenido para mostrarlo en un navegador Web. Un elemento HTML se distingue de otro texto en un documento mediante "etiquetas", que consisten en el nombre del elemento rodeado por "<" y ">" (Berners-Lee 2014).

**CSS**

Hojas de Estilo en Cascada (CSS*)* es el código que usas para dar estilo a tu página web. No es realmente un lenguaje de programación, tampoco es un lenguaje de marcado. Es un *lenguaje de hojas de estilo,* es decir, permite aplicar estilos de manera selectiva a elementos en documentos HTML. El CSS se puede usar para estilos de texto muy básicos como, por ejemplo, cambiar el color y el tamaño de los encabezados y los enlaces, para crear un diseño, como podría ser convertir una columna de texto en una composición con un área de contenido principal y una barra lateral para información relacionada. Incluso en la creación de efectos de animación (Wium Lie 2018).

**JavaScript**

Para el desarrollo de software para la web existen muchos lenguajes de programación, pero debido a las características del proyecto a desarrollar, se decide utilizar el lenguaje de programación JavaScript como lenguaje principal. Se utiliza tanto para el Front-End (lado del cliente) como para el Back-End (lado del servidor) debido a su versatilidad y ligereza.

La página para desarrolladores de Mozilla lo define como:
"un lenguaje de programación ligero, interpretado, o compilado (justo a tiempo) con funciones de primera clase. Si bien es más conocido como un lenguaje de scripting (secuencias de comandos) para páginas web, y es usado en muchos entornos fuera del navegador, tal como Node.js, Apache CouchDB y Adobe Acrobat. Es un lenguaje de programación basada en prototipos, multiparadigma, de un solo hilo, dinámico, con soporte para programación orientada a objetos, imperativa y declarativa" (Eich 2016).

### 1.3.2 Framework

Un framework es un esquema o marco de trabajo que ofrece una estructura base para elaborar un proyecto con objetivos específicos, una especie de plantilla que sirve como punto de partida para la organización y desarrollo de software.

Para la selección del framework de desarrollo a utilizar en el Front-End se tomaron en cuenta los siguientes parámetros de selección:

- Experiencia de los desarrolladores utilizando el framework
- Curva de aprendizaje
- Escalabilidad
- Rendimiento

Una vez definidos los parámetros, se selecciona a Vue y React como posibles framework a utilizar. A continuación, se hace un análisis de ambos para determinar cuál es la mejor opción para el desarrollo.

**Vue**

Vue es un marco de JavaScript para construir interfaces de usuario. Se basa en HTML, CSS y JavaScript estándar, y proporciona un modelo de programación declarativo y basado en

componentes que lo ayuda a desarrollar interfaces de usuario de manera eficiente, ya sea simples o complejas. Es un ecosistema que abarca gran parte las características del desarrollo fron end, vue toma en consideración lo extremadamente diversa que es la web variando de forma y escala (You 2020).

Está diseñado para ser flexible y adaptable, fue creado por Evan You un ex empleado de Google. Inicialmente surgió como una biblioteca personal, pero la comunidad hizo que el proyecto creciera a un ritmo impresionante, posicionándolo actualmente como uno de los marcos web más relevantes, junto con React (Álava Murillo 2022). Dependiendo del caso, puede usarse de diferentes formas:

- Mejora de HTML estático sin paso de compilación
- Incrustación como componentes web en cualquier página
- Aplicación de una sola página (SPA)
- Fullstack, representación del lado del servidor (SSR)
- Jamstack, generación de sitios estáticos (SSG)
- Orientación a escritorio, móvil, WebGL e incluso al terminal

**React**

React es una biblioteca javascript para crear interfaces de usuario. Es una alternativa para el desarrollo web SPA, y aplicaciones móviles, cuenta con un ecosistema de compones, módulos y herramientas que brindan facilidades, el principal enfoque que tiene es el de facilitar la creación de interfaces de usuario, de forma ágil y versátil (Walke 2013).

La biblioteca fue desarrollada por Facebook, y es mantenida por la misma, junto a desarrolladores independientes y otras compañías, actualmente muchas empresas utilizan React para las aplicaciones web, como, por ejemplo: Facebook, Instagram, Netflix, Twitter, Reddit, Uber, Paypal, entre otras. Desde su lanzamiento en 2013, su popularidad ha ido aumentando notablemente, transformándose en una de las tecnologías más utilizadas (Álava Murillo 2022).

**Comparativa Vue y React**

Vue y React comparten muchas similitudes. Ambos:

- Utilizan el DOM virtual;
- Proporcionan componentes de vista reactivos y componibles;
- Mantienen el enfoque en la librería central, con temas como el enrutamiento y la gestión global del estado generadas por librerías asociadas.

Si bien existen similitudes, como cabe esperar también se encuentran diferencias en diferentes apartados como, por ejemplo:

Rendimiento:

Tanto React como Vue ofrecen un rendimiento comparable en los casos de uso más comunes, con Vue normalmente un poco por delante debido a su implementación más ligera del DOM virtual.

En React, cuando el estado de un componente cambia, activa el re-renderizado de todo el subárbol de componentes, comenzando en ese componente como raíz. Para evitar repeticiones innecesarias de componentes hijo, necesita usar "pureComponent" o implementar "shoulComponenUpdate" siempre que pueda. También es posible que necesite utilizar estructuras de datos inmutables para que los cambios de estado sean más fáciles de optimizar. Lo que puede que en ocasiones se lleve a un estado del DOM inconsistente.

En Vue, las dependencias de un componente se rastrean automáticamente durante su renderizado, por lo que el sistema sabe con precisión qué componentes deben volver a renderizarse cuando cambia el estado.

En general, esto le elimina la necesidad de toda una clase de optimizaciones de rendimiento a los desarrolladores, y les permite centrarse más en la construcción de la aplicación en sí misma a medida que va escalando.

Utilización de HTML y CSS:

En React, todo es únicamente JavaScript. No sólo las estructuras HTML se expresan a través de JSX, sino que las tendencias recientes también apuntan a poner la gestión de CSS dentro de JavaScript. Este enfoque tiene sus propios beneficios, pero también viene con varias desventajas que no parecen valer la pena para todos los desarrolladores.

Vue adopta las tecnologías web clásicas y construye a partir de ellas.

En React, todos los componentes expresan su interfaz de usuario dentro de las funciones de render utilizando JSX, una sintaxis declarativa similar a XML que funciona con JavaScript.

Las funciones de renderizado con JSX tienen algunas ventajas:

- Puede aprovechar la potencia de un lenguaje de programación completo (JavaScript) para crear su vista. Esto incluye variables temporales, controles de flujo y referencias directas a los valores JavaScript en el ámbito de aplicación.

- El soporte de herramientas (comprobación de tipos, autocompletado del editor) para JSX es en algunos aspectos más avanzado que el disponible actualmente para las plantillas de Vue.

Vue, también tiene funciones de renderizado e incluso soporte JSX, porque a veces se necesita esa potencia. Sin embargo, como experiencia por defecto ofrece plantillas como una alternativa más simple. Cualquier HTML válido es también una plantilla Vue válida, y esto lleva a algunas ventajas propias:

- Para muchos desarrolladores que han trabajado con HTML, las plantillas se sienten más naturales para leer y escribir. La preferencia en sí misma puede ser algo subjetiva, pero si hace que el desarrollador sea más productivo, entonces el beneficio es objetivo.

- Las plantillas basadas en HTML hacen mucho más fácil la migración progresiva de las aplicaciones existentes para aprovechar las características de reactividad de Vue.

- También hace que sea mucho más fácil para los diseñadores y desarrolladores menos experimentados analizar y contribuir a la base del código.

En cuanto al CSS, a menos que extienda los componentes sobre múltiples archivos el alcance de CSS en React suele hacerse a través de soluciones CSS-en-JS. Esto introduce un nuevo paradigma de estilo orientado a componentes que es diferente del proceso normal de creación de CSS. La principal diferencia entre React y Vue aquí es que el método por defecto de estilado en Vue es a través de etiquetas “style” más familiares en componentes de un solo archivo dando el acceso completo a CSS en el mismo archivo que el resto del código de su componente.

Escala:

Para aplicaciones grandes, tanto Vue como React ofrecen soluciones de enrutamiento robustas. La comunidad de React ha sido muy innovadora en términos de soluciones para la gestión de estado como es el caso de “Redux” pero Vue ha llevado este modelo un paso más allá con “Vuex”, una solución de gestión de estado que se integra profundamente en Vue y que según los desarrolladores de Vue ofrece una experiencia de desarrollo superior.

Otra diferencia importante entre estas ofertas es que las librerías adicionales de Vue para la gestión de estados y enrutamiento están todas oficialmente soportadas y se mantienen actualizadas con la librería central. React, en cambio, opta por dejar estas preocupaciones en manos de la comunidad, creando un ecosistema más fragmentado.

React es conocido por su pronunciada curva de aprendizaje. Antes de que pueda empezar, necesita saber sobre JSX y probablemente ES2015+, ya que muchos ejemplos usan la sintaxis de clases de React. Mientras que para empezar con Vue no se necesita saber acerca de JSX, ES2015, o sistemas de compilación por lo que Vue presenta una menor curva de aprendizaje.

Luego de realizada la investigación de los Frameworks para el Front-End se llega a la conclusión de seleccionar a Vue en su versión 3 para el desarrollo del sistema puesto que destaca en su rendimiento en cuanto a las interacciones con el DOM, tiene una menor curva de aprendizaje y con su utilización se logra una mayor escalabilidad.

**Quasar**

Es un framework de código abierto basado en Vue.js que permite la creación de aplicaciones web responsivos. Cada uno de sus componentes está cuidadosamente diseñado para ofrecerle la mejor experiencia posible a sus usuarios. Está diseñado teniendo en cuenta el rendimiento y la capacidad de respuesta**.** Entre algunas de sus ventajas encontramos (Stoenescu 2015):

- Obtiene una interfaz de usuario de última generación para sus sitios web y aplicaciones listas para usar
- Es fácilmente personalizable (CSS) y extensible (JavaScript)
- Increíble comunidad su foro oficial y su chat de Discord
- Tiene un ciclo de lanzamiento regular que incluye nuevas características

Para la selección del framework de desarrollo a utilizar en el Back-End se tomaron en cuenta los siguientes parámetros de selección:

- Experiencia de los desarrolladores utilizando el framework
- Amplia documentación
- Rendimiento

Una vez definidos los parámetros, se selecciona a Express y Spring Boot como posibles framework a utilizar. A continuación, se hace un análisis de ambos para determinar cuál es el seleccionado.

**Express**

Express es un framework web flexible, utilizado para construir aplicaciones web sobre el ecosistema Node. Por defecto, utiliza el motor Pug para soportar plantillas. Es un framework relativamente pequeño que se encuentra en la parte superior de la funcionalidad del servidor web de Node para simplificar sus Interfaces de programación de aplicaciones (APIs) y añadir nuevas funciones útiles. Facilita la organización de la funcionalidad de su aplicación con middleware y enrutamiento; agrega utilidades a los objetos HTTP de Node y el renderizado de vistas HTML dinámicas; define un estándar de extensibilidad fácilmente implementado. Es fácil de configurar, implementar, controlar y proporcionar varios componentes clave para manejar las solicitudes web. Express ayuda en la creación de aplicaciones web y servidores HTTP simples ya que es un framework mínimo y flexible. Funciona como un medio para transferir las solicitudes de un cliente a una base de datos y envía las respuestas de la base de datos al cliente (Sayago y Chango 2018).

**Spring Boot**

Spring Boot es un framework que posee las bibliotecas necesarias para la creación de aplicaciones ejecutables basadas en Spring. Los componentes que posee Spring Boot permiten simplificar los pasos de la selección de las dependencias necesarias para el proyecto a desarrollar y su despliegue en el servidor. No requiere de configuración usando archivos en XML/JSON y permite la creación de servicios REST en Java de manera fácil, centrando al desarrollador en los elementos críticos de desarrollo especifico de su aplicación (Sayago y Chango 2018).

**Comparativa Express y Spring Boot**

Para comparar Express y Spring Boot se toma como referencia la investigación “Análisis comparativo para aplicaciones web basados en servicios REST: stack MEAN y stack Java EE” realizada por Jaime Sayago Heredia y Gustavo Chango Sailema Pontificia Universidad Católica del Ecuador, donde se comparan dos aplicaciones web construidas en Express y en Spring Boot respectivamente mediante pruebas basadas en los siguientes criterios (Sayago y Chango 2018):

Tiempo de respuesta

*Tabla 1 Tiempo de respuesta de las dos aplicaciones Spring Boot y Express*

| Etiqueta | Muestras | Promedio | Error % | Rendimiento | Kb/seg Recibidos |
|---|---|---|---|---|---|

| | | | | | |
|---|---|---|---|---|---|
| Aplicación con Spring Boot | 1000 | 953 | 0,00% | 350.30% | 2351.28 |
| Aplicación con Express | 1000 | 2 | 0,00% | 352.50% | 412.12 |

Como se aprecia en la tabla 1 se hicieron 1000 solicitudes de muestras. El tiempo promedio transcurrido (columna Promedio) fue de 953 milisegundos y el porcentaje de error (columna Error %) fue de 0 para la aplicación con Spring Boot. Para la aplicación con Express el tiempo promedio transcurrido (columna Promedio) fue de 2 milisegundos y el porcentaje de error (columna Error %) fue de 0. El porcentaje de error es la columna que muestra el porcentaje de solicitudes rechazadas de esta tabla. El resto de las columnas muestran la rapidez con la que se manejaron las solicitudes (columna de rendimiento). Respecto al rendimiento hay una pequeña diferencia favorable para la aplicación con Spring Boot 350.30% frente a 352.50% para la aplicación con Express. Por último, la cantidad de kilobytes recibidos (columna de kb/segundo recibidos). En esta se puede apreciar que la aplicación con Spring Boot su velocidad es de 2351.28 kb/segundo frente a 412.12 kb/segundo de la aplicación con Express (Sayago y Chango 2018).

Rendimiento

*Tabla 2 Tiempo de respuesta en milisegundos de 100000 solicitudes*

| **Concurrencia** | **10** | **50** | **100** | **500** |
|---|---|---|---|---|
| Spring Boot (respuesta en ms) | 64.216 | 180.072 | 224.170 | 1.451.031 |
| Express (respuesta en ms) | 8.395 | 34.006 | 68.301 | 697.426 |

Se observa en la tabla que es la prueba de 100000 respuestas y un nivel de concurrencia de 10, 50, 100 y 500. En la que para la aplicación con Spring Boot el tiempo medio que el servidor ha tardado en atender un grupo de peticiones fue de 64.216, 180.072, 224.170 y 1.451.031 milisegundos respectivamente. Y para la aplicación con Express este tiempo de fue de 8.395, 34.006, 68.301 y 6097.426 milisegundos. Se puede apreciar que mientras aumenta el número de solicitud de respuesta y concurrencia mejora el rendimiento de la aplicación con Express sobre Spring Boot (Sayago y Chango 2018).

Luego de realizada la investigación de los frameworks para el back-End se llega a la conclusión de seleccionar a Express para el desarrollo del sistema puesto que este destaca en su rendimiento, ligereza y mejoras en los tiempos de respuesta, también se cuenta con el objetivo

de utilizar el mismo lenguaje tanto en el Back-End como en el Front-End para así lograr una mejor comunicación entre los desarrolladores del sistema.

### 1.3.3 Librerías

**Axios**

Es una librería JavaScript que puede ejecutarse en el navegador y que facilita las operaciones como cliente HTTP, por lo que se puede configurar y realizar solicitudes a un servidor y recibir respuestas fáciles de procesar (Blanch 2022).

**Socket.IO**

La página oficial de Socket.IO lo define como:
"una biblioteca que permite la comunicación de baja latencia, bidireccional y basada en eventos entre un cliente y un servidor. Se basa en el protocolo WebSocket y proporciona garantías adicionales, como el respaldo al sondeo largo HTTP o la reconexión automática" (Arrachequesne 2022).

**Pdfmake**

La página oficial de Pdfmake lo define como:
"una biblioteca de generación de documentos PDF para uso del lado del servidor y del cliente en JavaScript puro (Pampuch 2022).

### 1.3.4 Sistemas Gestores de Base de Datos

Un sistema de gestión de base de datos (SGBD) es una aplicación que permite a los usuarios definir, crear y mantener bases de datos, proporcionando acceso controlado a las mismas. Es una herramienta que sirve de interfaz entre el usuario y las bases de datos (Ibáñez 2016).

Para la selección del SGBD se tomaron en cuenta los siguientes parámetros de selección:

- Experiencia de los desarrolladores utilizando el SGBD
- Curva de aprendizaje y escalabilidad
- Tiempo de ejecución

Una vez definidos los parámetros, se selecciona a PostgreSQL y MongoDB como posibles SGBD a utilizar. A continuación, se hace un análisis de ambos para determinar cuál es el utilizado.

**PostgreSQL**

Primeramente, decir que las bases de datos relacionales han sido una de las más utilizadas durante una gran cantidad de tiempo, cuyo núcleo se focaliza en el uso de relaciones entre datos. La información es recuperada y almacenada mediante consultas, construidas habitualmente en Structured Query Language (SQL). El funcionamiento de este tipo de bases de datos consiste en introducir los datos en registros, que se encuentran almacenados en tablas.

Las principales ventajas de las bases de datos relacionales son que garantizan la integridad referencial (al eliminar un registro, elimina los registros que dependían de este a través de relaciones) y que no haya duplicidad en los registros (gracias al uso de claves primarias y de claves únicas), además de favorecer la normalización haciéndola más comprensible. Por otro lado, las desventajas son que, en lo relacionado a datos gráficos o multimedia, presenta deficiencias y con respecto a los bloques de texto, estos no se manipulan muy bien como tipo de dato.

PostgreSQL es un SGBD relacional y orientado a objetos reconocido como uno de los más potentes del mercado. También es multiplataforma, extensible (se puede añadir funcionalidades que no vengan de serie), es escalable y funciona bajo licencia libre, es decir, se puede usar para cualquier propósito, contando con una gran comunidad que se encarga de su mantenimiento y respaldo (Ibáñez 2016).

PostgreSQL cuenta con una serie de características que la hacen destacar

- Presenta un sistema de alta concurrencia, denominado MVCC, que permite que mientras un proceso está escribiendo de una tabla, otros puedan acceder a esa tabla sin necesidad de bloqueo, dotando a cada uno de una visión consistente
- Cuenta con el sistema “Host Standby”, que permite al usuario poder conectarse al servidor y ejecutar búsquedas en la base de datos mientras esta se encuentra en modo recuperación.
- Posee la capacidad de registrar cada transacción en un Registro de Escritura Anticipada o Write Ahead Log (WAL), que es un método estándar para garantizar la integridad de los datos. Este registro permite restaurar la base de datos a cualquier punto previamente guardado.
- Por último, el uso de disparadores (triggers), que son procesos almacenados que se ejecutan, basados en una acción determinada sobre una tabla específica de la base de datos.

Es usado en diversos ámbitos como por ejemplo para el almacenamiento de datos, en sistemas de información geográfica, en bases de datos para servicios web como Wordpress y en una plataforma como servicio.

**MongoDB**

Primeramente, decir que las bases de datos no relacionales están diseñadas para modelos de datos muy específicos y que hace uso de esquemas flexibles con el fin de crear aplicaciones modernas. También se le conoce por el nombre de bases de datos NoSQL. Son bases de datos bastante conocidas debido a su facilidad sobre todo con respecto a su desarrollo, tienen una gran funcionalidad y un gran rendimiento. Surgen como alternativa a las bases de datos relacionales, diferenciándose de estas en que las no relaciones no siguen un modelo relacional y usan una forma de almacenamiento estructurada, es decir, sus tablas no poseen una estructura fija. Suelen ser, por tanto, bases de datos muchos más flexibles y abiertas. Esto permite la adaptación de este tipo de bases de datos a las necesidades específicas de los proyectos de una forma más sencilla que los modelos E/R. Este tipo de bases de datos, debido a su optimización, son recomendables para aplicaciones que necesitan grandes volúmenes de datos, baja latencia y modelos de datos flexibles. Se suelen usar en entornos distribuidos que han de estar siempre operativos y disponibles.

Las ventajas de este tipo de bases de datos residen en su flexibilidad (ideales para datos semiestructurados y no estructurados otorgando un desarrollo más rápido e intuitivo), en su escalabilidad (permitiendo aumentar sus capacidades mediante clústeres distribuidos de hardware), en su alto rendimiento (gran optimización hacia los modelos de datos específicos y patrones de acceso) y en su alta funcionalidad (posee tipos de datos que se ajustan específicamente a modelos de datos concretos). Como desventajas cabe destacar que sufren problemas de compatibilidad al no estar suficiente maduros para algunas empresas ya que, al ser relativamente novedosas, puede que no ofrezcan mucha documentación.

MongoDB es una base de datos no relacional distribuida, de código abierto y basada en documentos, lo que significa que los datos se almacenan en forma de documentos. En lugar de guardar los datos en las tablas como en el caso de PostgreSQL, estos se almacenan en estructuras de datos BSON (similar a JSON) con un esquema dinámico, permitiendo una rápida y sencilla integración de los datos.

Entre sus características principales destacan la indexación de los campos de un documento, el soporte de búsquedas por campos, consultas de rangos y expresiones regulares, la posibilidad de escalar horizontalmente balanceando la carga o replicando los datos para mantener el

funcionamiento del sistema ante fallos del hardware, el aprovechamiento de dicho balanceo de carga para el uso de MongoDB como sistema de archivos, la posibilidad de realizar consultas mediante Javascript, entre otros.
Su uso suele a estar focalizando en entornos que lo necesitan para almacenamiento y registros de eventos, para comercio electrónico, para aplicaciones móviles y juegos, para manejo de estadísticas en tiempo real, para sistemas con alto volumen de lecturas y para proyectos que hacen uso de metodologías de desarrollo ágiles o iterativos (Ibáñez 2016).

**Comparativa PostgreSQL y MongoDB**

Para comparar a PostgreSQL y MongoDB se toma como referencia la investigación "Comparación de rendimiento entre bases de datos Relacionales, NoSQL y Blockchain" realizada por Alberto Pérez Román en la Universidad de Málaga en el año 2020 (Pérez Román 2020), en donde se compara a ambos sistemas de gestión de base de datos mediante un caso de uso.

Según dicha investigación, con respecto al grado de dificultad de ambos, MongoDB sin lugar a duda es el gestor de almacenamiento que menos dificultades ha causado a la hora de implementación. PostgreSQL aporta la facilidad de gestión de la base de datos gracias a su interfaz intuitiva y de fácil uso, pero su implementación al inicio es medianamente compleja.

Para comparar el rendimiento se realizaron unas pruebas de ejecución, las cuales están divididas en tres sectores: las inserciones, las consultas y la ejecución de ambas, en el cual MongoDB obtiene mejores tiempos solo en las consultas y aunque ambos arrojan un rendimiento similar, se llega a la conclusión de que se debe tener en cuenta el número de operaciones a realizar. Como se muestra en la gráfica de la figura 1, si se realiza un número reducido de operaciones en torno a las 1000 y 5000 operaciones, se destaca MongoDB, ya que los tiempos en las inserciones son similares y no puede hacerles frente a las lecturas (Pérez Román 2020).

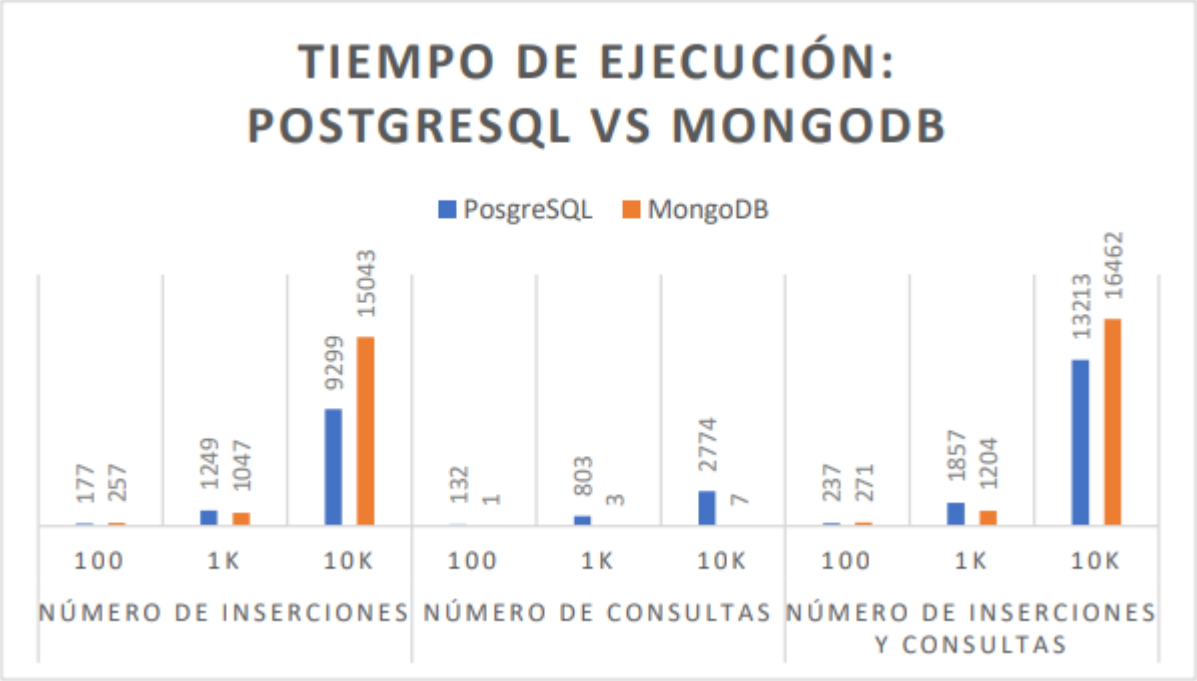



*Ilustración 1 Gráfica con los tiempos de ejecución (ms) fuente: PostgreSQL frente a MongoDB*

Luego de realizada la investigación de los SGBD se llega a la conclusión de seleccionar a MongoDB para el desarrollo del sistema puesto que este tiene menos dificultad a la hora de la implementación, destaca por su escalabilidad y se adapta más al sistema a desarrollar debido a su rendimiento en cuanto a las lecturas y en la cantidad de operaciones.

### 1.3.5 Herramientas

Para el desarrollo de un software es necesario utilizar herramientas intermediaras. Para la selección de estas se tomaron en cuenta los siguientes parámetros de selección:

- Experiencia de los desarrolladores utilizando las herramientas
- Usabilidad
- Bajo consumo de recursos

Una vez definidos los parámetros, se llega a la conclusión que para el desarrollo de esta aplicación web se emplearán los siguientes programas.

**Microsoft Visual Studio Code**

Visual Studio Code es un editor de código fuente ligero pero potente que se ejecuta en su escritorio y está disponible para Windows, macOS y Linux. Viene con soporte integrado para

JavaScript, TypeScript y Node.js y tiene un amplio ecosistema de extensiones para otros lenguajes y tiempos de ejecución. Su objetivo es proporcionar las herramientas que un desarrollador necesita para un ciclo rápido de compilación y depuración de código (Microsoft 2015).

**Node.js**

Node.js es un entorno de ejecución para JavaScript orientado a eventos asíncronos. Está diseñado para crear aplicaciones escalables (Node.js 2022).

**Git**

Es un sistema de control de versiones de código abierto y de libre uso, diseñado para manejar desde grandes hasta pequeños proyectos, con gran eficiencia y velocidad. Registra los cambios realizados en un archivo o conjunto de archivos a lo largo del tiempo, de modo que puedas recuperar versiones específicas más adelante (Torvalds 2007).

**Google Chrome**

Navegador web desarrollado por Google. Está diseñado para hacer que puedas navegar en internet de una manera simple y rápida. Con su diseño limpio y características avanzadas, Chrome se ha convertido en uno de los navegadores de Internet más populares en todo el mundo desde su lanzamiento en el año 2008. Fue diseñado específicamente para abrir aplicaciones web con velocidad y estabilidad (Alphabet 2010).

**Postman**

Postman es una plataforma gratuita si trabajas solo o de pago si quieres trabajar de manera colaborativa con tu equipo. Esta plataforma posibilita y facilita la creación y el uso de APIs. Se puede usar, por ejemplo, para obtener información sobre las respuestas HTTP, en diferentes métodos, que realicemos a APIs de diferentes temáticas (KeepCoding 2022).

**MongoDB Compass**

MongoDB Compass es una herramienta interactiva para consultar, optimizar y analizar datos en MongoDB. Es de uso gratuito y se puede ejecutar en macOS, Windows y Linux (MongoDB Inc. 2019).

### 1.4 Metodología de desarrollo de software

Las metodologías de desarrollo de software consisten en una serie de métodos y técnicas organizativas aplicadas a la creación y diseño de software informático. Estos métodos persiguen

organizar de manera eficiente a los equipos implicados para que desarrollen con éxito sus funciones teniendo en cuenta aspectos como la dificultad del proyecto, la planificación, el presupuesto, los costes, los profesionales disponibles, el lenguaje que se va a utilizar, etc.

Las metodologías ágiles de desarrollo de software se caracterizan por su enorme agilidad y flexibilidad en los procesos de trabajo. Gracias a su implementación, los equipos trabajan de una manera mucho más eficiente y productiva porque saben exactamente lo que tienen que hacer y hay una comunicación fluida entre ellos. Por otro lado, la metodología se adapta perfectamente a los imprevistos y las necesidades que surgen durante el proceso porque los ciclos se reducen y son mucho más rápidos. Estas permiten la creación de equipos de trabajo independientes y autosuficientes que están al tanto de todo el proceso de desarrollo. Además, el cliente también puede aportar y corregir porque supervisa en tiempo real cómo avanza su proyecto (Domainlogic 2022).

Algunas de las metodologías ágiles empleadas en la actualidad para el desarrollo de software son (Domainlogic 2022):

Kanban

Este método fue desarrollado por la marca de coches japonesa Toyota. Su principal característica es que divide el proceso en tareas mínimas y las coloca en un tablero de trabajo donde se pueden ver las tareas que están en curso, pendientes o finalizadas. El trabajo se centra en las tareas prioritarias y con el valor del producto siempre en primer lugar (Domainlogic 2022).

Scrum

Es bastante similar a Kaban ya que también divide las tareas y requisitos de los procesos de trabajo. Se crean bloques de tiempos cortos y fijos que no superan las cuatro semanas. Las fases de esta metodología comienzan con una planificación de cada tarea (planning sprint), después la ejecución (sprint), la reunión diaria y la demostración de resultados (sprint review) (Domainlogic 2022).

Programación extrema (XP)

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. Esta se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los

cambios. Además, se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico (Beck y Andres 2004).

**Fundamentación de la metodología a utilizar**

Después de realizarse un análisis, se seleccionó como metodología para guiar el proceso de desarrollo del software a XP ya que la metodología XP permite administrar cambios de forma óptima. Además, establece que la comunicación y satisfacción del cliente es lo principal; potenciando el trabajo en equipo donde los clientes y los desarrolladores están involucrados en el desarrollo del software. Esta metodología no depende de un jefe de proyecto que oriente al equipo de desarrollo, cuenta con la práctica de programación en pareja y posee un enfoque para proyectos de corta duración.

## 1.5 Conclusiones del capítulo

En este capítulo se abordaron un conjunto de elementos teóricos que fundamentan la solución del problema por lo que se llega a las siguientes conclusiones:

1. El análisis de los sistemas homólogos planteados concluyó que no satisfacen las necesidades actuales y demostró la necesidad de realizar un nuevo sistema que permita gestionar tanto la elaboración como el chequeo del cumplimiento de los objetivos estratégicos que se tracen en la facultad para un año determinado mediante evidencias.

2. El estudio y selección de las tecnologías y lenguajes a utilizar permitió construir la base tecnológica para desarrollar el sistema propuesto.

3. La selección de la metodología de desarrollo XP permitirá guiar el proceso de desarrollo.

## CAPÍTULO 2: ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

En el presente capítulo se describen los elementos principales que se tendrán en cuenta en el desarrollo de la propuesta de solución. Con el uso de la metodología XP se organizará el proceso de trabajo respecto a las fases y artefactos que define XP, entre los que se encuentran las historias de usuario, estimación por esfuerzo y plan de entrega por iteraciones. Además, en este capítulo quedan definidos el patrón arquitectónico, los patrones de diseño y las tarjetas de clase, responsabilidad y colaboración.

### 2.1 Descripción del proceso de gestión de las evidencias y cumplimiento de los objetivos del año actualmente

Al inicio del año la decana o decano de la Facultad 4, elabora los objetivos estratégicos a cumplir por ARC, definiendo los criterios de medidas e indicadores correspondientes a cada objetivo estratégico, en este proceso se crean las planillas con toda la información respecto a los objetivos estratégicos. Los jefes de departamentos establecen a cada profesor o especialista mediante un informe digital qué indicadores debe realizar para contribuir al cumplimiento de los objetivos estratégicos y a la vez a su evaluación personal.

A lo largo del año los profesores y especialistas van generando evidencias de las actividades relacionadas con los indicadores que les fueron establecidos, más otras actividades no contempladas en dichos indicadores. Estas son enviadas a sus respectivos jefes de departamento mediante correo electrónico.

Al final del año los jefes de departamentos apoyados en las evidencias recibidas generan las evaluaciones personales de los profesores o especialistas. Estas evidencias son usadas también para la evaluación de las ARC mediante el nivel de cumplimiento de los objetivos estratégicos.

### 2.2 Objetivos del sistema

Como paso previo al desarrollo de un producto de software se debe definir con exactitud los objetivos que se persiguen con el mismo. Estos quedan definidos de manera general en los aspectos siguientes:

1. Centralizar la información para lograr una mejor planificación.
2. Evitar por la pérdida de información cuando son generadas las evidencias.
3. Control en tiempo real del nivel de cumplimiento de los Objetivos Estratégicos, ayudando a la toma de decisiones.

4. Agilizar en gran medida la elaboración de las Evaluaciones de los profesores o especialistas.

### 2.3 Descripción del sistema

Atendiendo al proceso descrito en el epígrafe 2.1, se define como propuesta que fundamenta esta investigación: La creación de un sistema de gestión que permita contribuir en el proceso de planificación, ejecución y control de los objetivos estratégicos del año, con la integración total de la información a través de las evidencias generadas por los propios usuarios a lo largo del año además de gestionar la evaluación personal de cada usuario; nutriéndose en gran medida de las actividades realizadas para el cumplimiento de los objetivos estratégicos.

El sistema cuenta con la capacidad de gestionar las diferentes ARC que tiene la Facultad 4, a su vez, esas áreas están compuestas por los objetivos estratégicos y sus criterios de medidas. Los usuarios con los permisos correspondientes establecerán los indicadores asociados a los criterios de medidas, para luego asignarle a cada usuario los indicadores a cumplir, siendo notificados de dicha asignación por el propio sistema.

En el transcurso del año los usuarios podrán gestionar sus evidencias relacionadas con los indicadores respectivamente, haciendo que estos cambien su estado ha cumplido o no cumplido en caso de que se estuviera eliminando la evidencia, notificando al usuario responsable del área en cuestión por el propio sistema. De esta forma el sistema muestra de manera automática el nivel de cumplimiento de los objetivos estratégicos.

Al finalizar el año se genera un informe en formato PDF sobre el estado de los objetivos estratégicos proporcionado una evaluación del área. En el otro apartado del sistema también se podrá generar una evaluación personal de los usuarios igualmente en formato PDF. Para generar dicha evaluación el sistema tiene en cuenta todos los indicadores cumplidos por el usuario más los agregados por el propio usuario. El resultado de la evaluación dependerá del criterio humano pero el sistema en proporción con las evaluaciones emitidas en las diferentes categorías de los indicadores irá mostrando una propuesta de evaluación final.

Para una mejor comprensión del sistema se realizó un modelo de dominio que se presenta en la Ilustración 2.

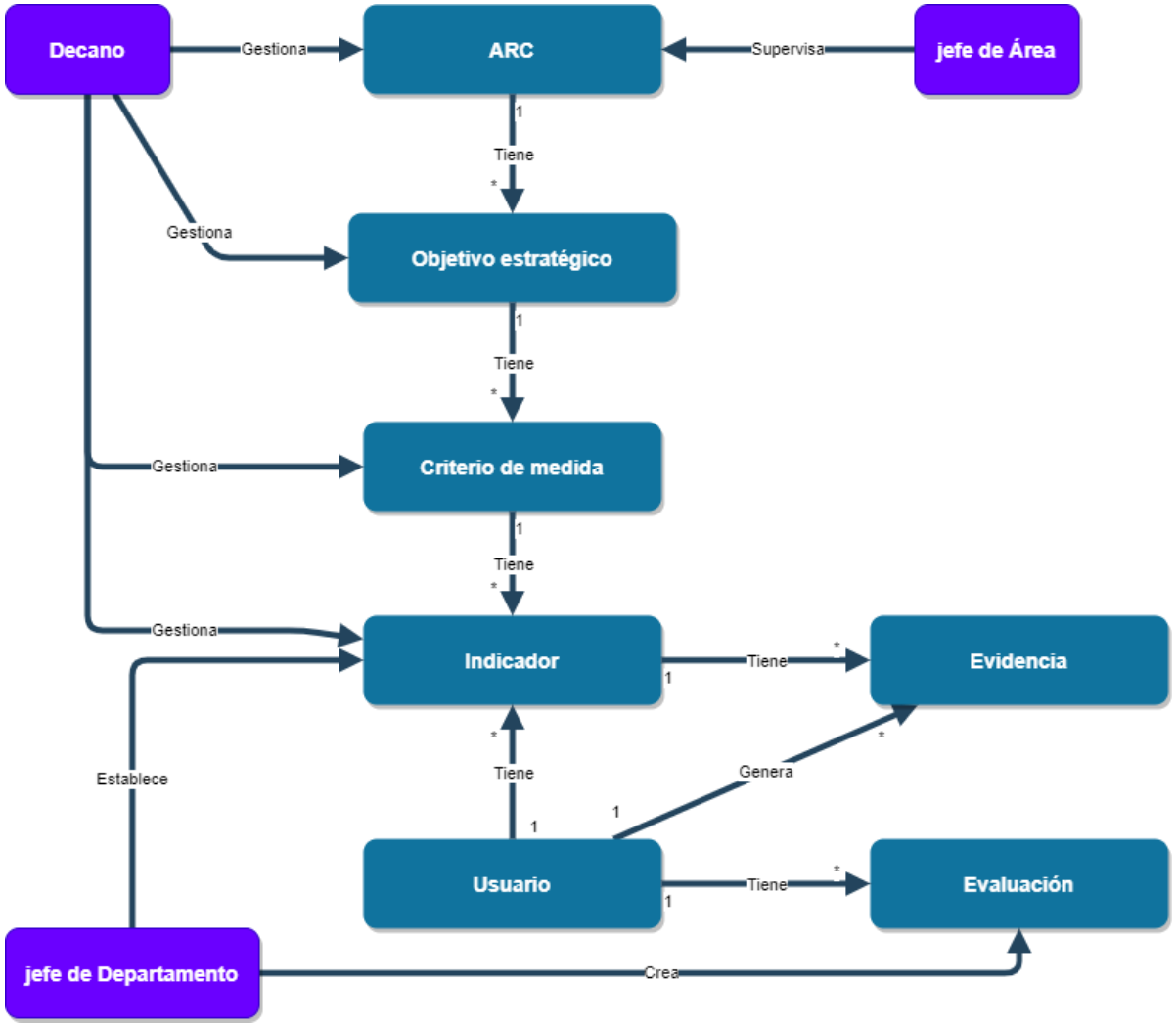



*Ilustración 2 Modelo de domino del sistema*

## 2.4 Usuarios relacionados con el sistema

Para un mejor entendimiento de la solución, se dará a conocer los usuarios que estarán relacionados de una forma u otra con el sistema informático.

*Tabla 3 Usuarios relacionados con el sistema*

| Usuarios | Descripción |
|---|---|
| **Decano** | Realiza la gestión de las diferentes áreas de la facultad y también la gestión de los usuarios. |
| **Jefe de área** | Es el encargado de velar por el cumplimiento de los objetivos estratégicos y de generar el informe de la evaluación de su área. |

| | |
|---|---|
| **Jefe de departamento** | Es el encargado de asignar los indicadores a sus trabajadores, además de evaluarlos y generar sus evaluaciones en formato PDF. |
| **Usuario Autenticado** | Accede al plan de indicadores que le fue asignado, donde también puede crear sus propios indicadores, cuenta además con la posibilidad de agregar una observación en forma de texto a cada uno de los indicadores, que luego estará presente en su evaluación. Y por último gestiona sus evidencias. |
| **Usuario Invitado** | Solo puede acceder a la página inicial la cual muestra información referente al sistema. |

### 2.5 Requisitos de software

Los requisitos de un sistema son aquellos que describen cualquier actividad o característica que este deba tener. Estos comunican las expectativas de los consumidores del producto o software.

Para dar respuesta a la problemática se obtuvieron los siguientes requisitos funcionales (RF):

**RF 1: Crear Área de Resultado Clave:** el sistema debe permitir crear un área de resultados claves de resultado clave.

**RF 2: Modificar Área de Resultado Clave:** el sistema debe permitir modificar un área de resultados claves de resultado clave.

**RF 3: Obtener Área de Resultado Clave:** el sistema debe permitir obtener un área de resultados claves de resultado clave.

**RF 4: Eliminar Área de Resultado Clave:** el sistema debe permitir eliminar un área de resultados claves de resultado clave.

**RF 5: Crear objetivo estratégico:** el sistema debe permitir crear un objetivo estratégico.

**RF 6: Modificar objetivo estratégico:** el sistema debe permitir modificar un objetivo estratégico.

**RF 7: Obtener objetivo estratégico:** el sistema debe permitir obtener un objetivo estratégico.

**RF 8: Eliminar objetivo estratégico:** el sistema debe permitir eliminar un objetivo estratégico.

**RF 9: Crear criterio de medida:** el sistema debe permitir crear un criterio de medida.

**RF 10: Modificar criterio de medida:** el sistema debe permitir modificar un criterio de medida.

**RF 11: Obtener criterio de medida:** el sistema debe permitir obtener un criterio de medida.

**RF 12: Eliminar criterio de medida:** el sistema debe permitir eliminar un criterio de medida.

**RF 13: Crear indicador:** el sistema debe permitir crear un indicador.

**RF 14: Modificar indicador:** el sistema debe permitir modificar un indicador.

**RF 15: Obtener indicador:** el sistema debe permitir obtener un indicador.

**RF 16: Eliminar indicador:** el sistema debe permitir eliminar un indicador.

**RF 17: Crear usuario:** el sistema debe permitir crear un usuario.

**RF 18: Modificar usuario:** el sistema debe permitir modificar un usuario.

**RF 19: Obtener usuario:** el sistema debe permitir obtener un usuario.

**RF 20: Eliminar usuario:** el sistema debe permitir eliminar un usuario.

**RF 21: Crear evidencia:** el sistema debe permitir crear una evidencia.

**RF 22: Modificar evidencia:** el sistema debe permitir modificar una evidencia.

**RF 23: Obtener evidencia:** el sistema debe permitir obtener una evidencia.

**RF 24: Eliminar evidencia:** el sistema debe permitir eliminar una evidencia.

**RF 25: Autenticar usuario:** el sistema debe permitir autenticar usuario.

**RF 26: Evaluar usuario:** el sistema debe permitir evaluar a un usuario.

**RF 27: Exportar evaluación:** el sistema debe permitir exportar evaluaciones en formato PDF.

**RF 28: Cambiar el estado del Criterio:** el sistema debe permitir cambiar el estado de los criterios de medidas.

**RF 29: Adjuntar archivo a la evidencia:** el sistema debe permitir adjuntar archivos a las evidencias.

**RF 30: Cambiar la contraseña:** el sistema debe permitir cambiar la contraseña.

**RF 31: Asignar indicadores:** el sistema debe permitir asignar indicadores a un usuario.

**RF 32: Crear indicador personal:** el sistema debe permitir a un usuario crear su propio indicador.

**RF 33: Cambiar estado del indicador:** el sistema debe cambiar el estado del indicador.

**RF 34: Restablecer contraseña:** el sistema debe permitir restablecer la contraseña de un usuario.

**RF 35: Buscar usuario:** el sistema debe permitir buscar a un usuario por su nombre o nombre de usuario.

**RF 36: Mostrar porcentaje:** el sistema debe mostrar el porcentaje del nivel de cumplimiento de las ACR.

**RF 37: Notificar al usuario:** el sistema debe notificar al usuario en tiempo real o posterior en caso del usuario no estar autenticado.

**RF 38: Notificar al jefe de departamento:** el sistema debe notificar al jefe de departamento en tiempo real o posterior en caso del jefe de departamento no estar autenticado.

**RF 39: Generar informe de las ARC:** el sistema debe permitir generar un informe en formato PDF de una ARC.

**RF 40: Denegar indicador:** el sistema debe permitir denegar un indicador.

**RF 41: Añadir observación:** el sistema debe permitir añadir observaciones a las evidencias.

**RF 42: Obtener ARC por año:** el sistema debe permitir obtener las ARC por año.

**RF 43: Obtener indicadores por año:** el sistema debe permitir obtener los indicadores de un usuario por año.

**RF 44: Establecer fecha a una evidencia:** el sistema debe permitir establecer fecha a una evidencia.

**RF 45: Crear un año:** el sistema debe permitir crear un año.

**RF 46: Eliminar un año:** el sistema debe permitir eliminar un año.

**RF 47: Crear un departamento:** el sistema debe permitir crear un departamento.

**RF 48: Eliminar un departamento:** el sistema debe permitir eliminar un departamento.

**RF 49: Obtener evaluación por año:** el sistema debe permitir obtener la evaluación de un usuario por año.

**RF 50: Evaluar usuario en un año:** el sistema debe permitir evaluar a un usuario en un año.

También se obtuvieron los siguientes requisitos no funcionales (RNF):

**Software**

**RNF 1:** Las computadoras de los clientes requieren de un navegador Mozilla Firefox 95.0 superior o Google Chrome 96.0 superior.

**Hardware**

**RNF 2:** Para la PC cliente se utilizó una PC Intel Pentium 4 o superior, CPU 1.2GHZ o superior, 1 GB RAM o superior, 160 GB HDD o superior.

**RNF 3:** Para la PC cliente se utilizó una PC Intel Pentium 4 o superior, CPU 1.2GHZ o superior, 1 GB RAM o superior, 160 GB HDD o superior.

**Usabilidad**

**RNF 4:** La usabilidad representa facilidad de uso por parte de los usuarios: El sistema debe permitir la fácil interacción con el mismo y llegar de manera rápida y efectiva a la información buscada. Debe, además, ser una interfaz de manejo cómodo que posibilite a los usuarios sin experiencia una rápida adaptación.

**Portabilidad**

**RNF 6:** El sistema debe funcionar en los sistemas operativos (Windows, Linux).

**Apariencia o interfaz externa**

**RNF 7:** El diseño de las interfaces tendrá pocas imágenes y colores. La información aparecerá correctamente organizada de forma tal que el usuario, pueda encontrar lo que busca rápidamente.

**Accesibilidad**

**RNF 8:** Las opciones deben ser de fácil acceso, se utilizarán iconos conocidos y conceptos de navegación sencillos además de utilizar encabezados breves y descriptivos para estructurar la información.

### 2.6 Historias de usuario

Las historias de usuario (HU) definen lo que se debe construir en el proyecto de software, tienen una prioridad asociada determinada por el cliente para indicar cuales son las más importantes en el resultado final. Estas serán divididas en tareas y su tiempo será estimado por los desarrolladores, dando paso a la elaboración del plan de iteraciones (Beck y Andres 2004).

Respecto de la información contenida en la HU, existen varias plantillas sugeridas, pero no existe un consenso al respecto. Las HU de la presente investigación, están compuesta por: el **número** que la identifica, el **nombre**, los **usuarios** que van a interactuar con la aplicación, los **puntos de estimación** que son las semanas que durará la implementación de dicha HU, la **prioridad** que va a tener para el negocio, la iteración en que será implementada, los **riesgos** en el desarrollo y una breve **descripción** de lo que realizará la misma. Y de la misma manera en la presente investigación a juicio de los autores, se asume un formato para las HU que recogen los requisitos no funcionales del sistema, el mismo está conformado por el **número** que la identifica, el **nombre** y la **descripción**.

Seguidamente, se presentan las principales HU del sistema. El resto se encuentra en el anexo número 2:

*Tabla 4 HU 1 Crear Área*

| **Historia de Usuario** | |
|---|---|
| **Número de HU:** 1 | **Nombre:** Crear Área de resultados claves |
| **Usuarios:** decano | |
| **Puntos de estimación:** 0.125 | **Iteración asignada:** 1 |
| **Prioridad del negocio:** Alta | **Riesgos en el desarrollo:** Media |
| **Descripción:** La aplicación web debe brindar al decano una vista donde se permita crear un Área de resultados claves en el sistema, ingresando en nombre de dicha Área de resultados claves y opcionalmente los objetivos que le corresponden. | |
| **Observaciones:** Para realizar esta acción el decano debe estar autenticado | |
| **Prototipo de interfaz:** | |

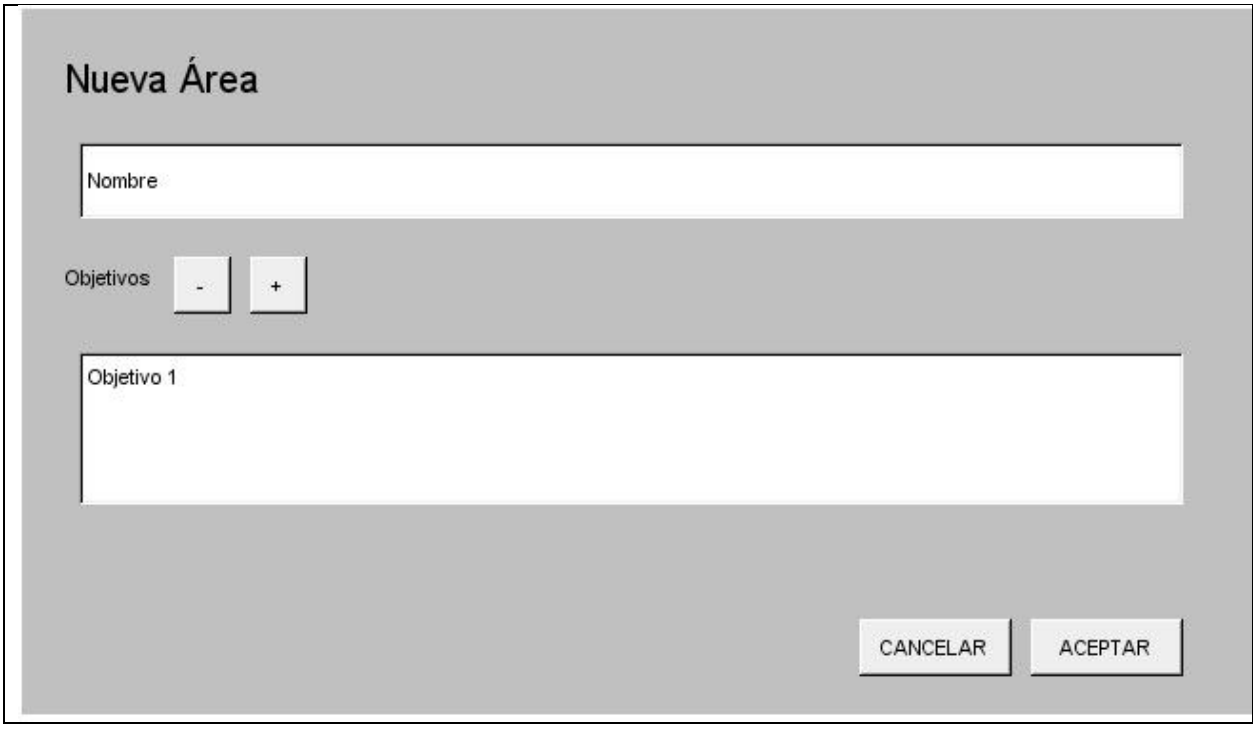



*Tabla 5 HU 21 Crear Evidencia*

| **Historia de Usuario** | |
|---|---|
| **Número de HU:** 21 | **Nombre:** Crear Evidencia |
| **Usuarios:** usuario autenticado, jefe de área y jefe de departamento | |
| **Puntos de estimación:** 0.5 | **Iteración asignada:** 1 |
| **Prioridad del negocio:** Alta | **Riesgos en el desarrollo:** Media |
| **Descripción:** La aplicación web debe brindar al usuario autenticado, jefe de área y al jefe de departamento una vista donde se permita crear una evidencia en el sistema, ingresando una descripción, una fecha y de manera opcional un archivo en cualquier formato. | |
| **Observaciones:** Para realizar esta acción el usuario autenticado, jefe de área y el jefe de departamento deben estar autenticados | |
| **Prototipo de interfaz:** | |

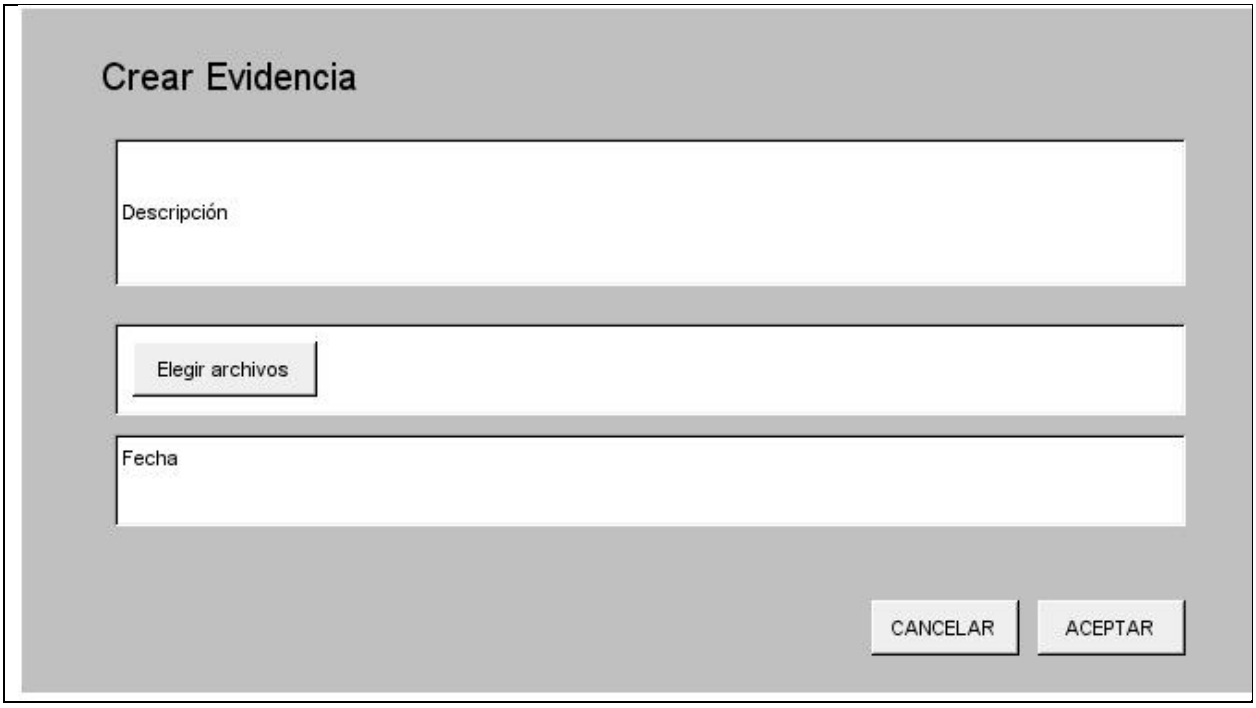
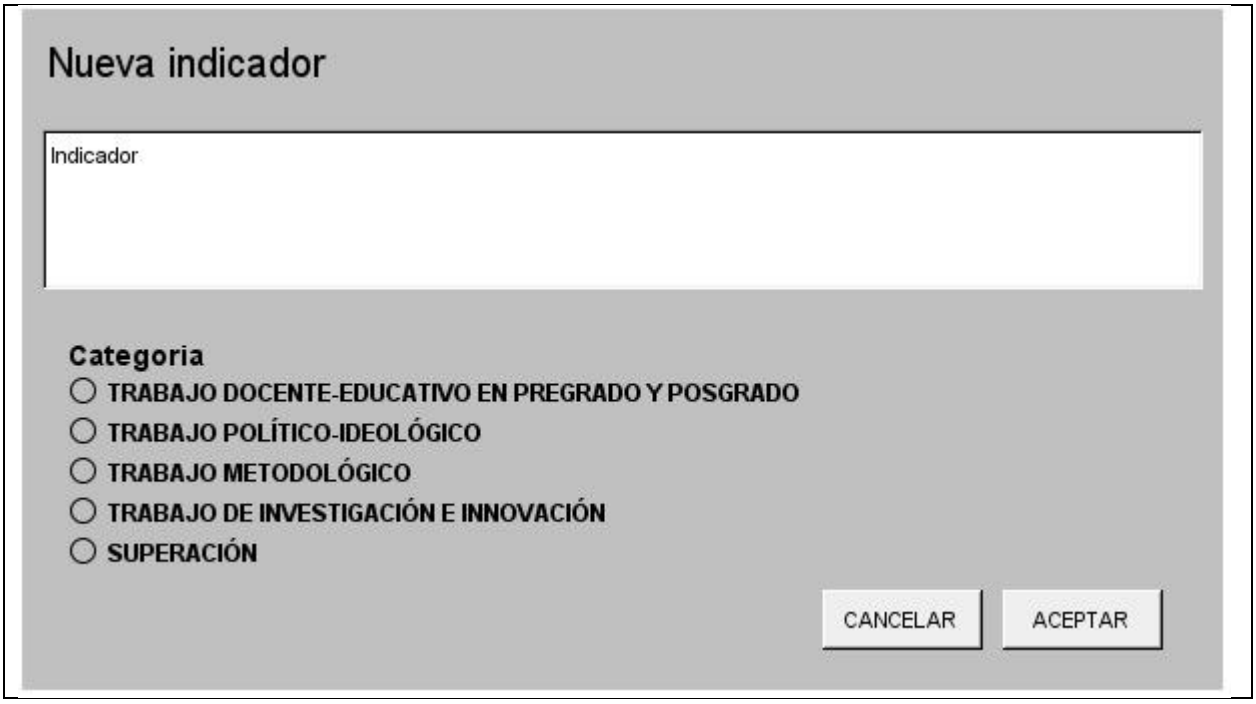



*Tabla 6 HU 13 Crear Indicador*

| **Historia de Usuario** | |
|---|---|
| **Número de HU:** 13 | **Nombre:** Crear Indicador |
| **Usuarios:** decano y jefe de área | |
| **Puntos de estimación:** 0.5 | **Iteración asignada:** 1 |
| **Prioridad del negocio:** Alta | **Riesgos en el desarrollo:** Media |
| **Descripción:** La aplicación web debe brindar al decano y al jefe de área una vista donde se permita crear un Indicador en el sistema, ingresando el nombre de dicho indicador y seleccionando la categoría que le pertenece. | |
| **Observaciones:** Para realizar esta acción el decano y el jefe de área deben estar autenticados | |
| **Prototipo de interfaz:** | |




### 2.7 Estimación de esfuerzo por HU

Después de definir las HU, los desarrolladores hacen una planificación del tiempo que necesitan para implementarlas y así determinar el tiempo total para el desarrollo de la aplicación. En el caso de la presente investigación la estimación de esfuerzo por HU está dada en un total de 9.6 semanas.

### 2.8 Iteraciones

Luego de haber realizado las HU y una estimación previa de esfuerzo, se procede a realizar la planificación de la implementación del sistema, especificando la prioridad con que se irán implementando las HU, organizadas por iteraciones, así como posibles fechas de liberación. Una iteración no es más que una pequeña versión del proyecto que se muestra como resultado, con un valor para el cliente. La versión completa del mismo no se obtiene hasta que no concluyan todas las iteraciones.

**Iteración 1:**

En esta iteración se dará cumplimiento de implementación a las HU de mayor prioridad. Al concluir esta iteración, se obtendrá el resultado de algunas de las funcionalidades básicas del sistema.

**Iteración 2:**

En esta iteración se continuará dando cumplimiento a las restantes HU de mayor prioridad y alguna de prioridad media. Con la implementación de las mismas, se obtendrán los resultados de las funcionalidades que cubren las necesidades especificadas por el cliente.

**Iteración 3:**

En esta iteración se implementan las funcionalidades con prioridad baja. Las mismas brindan al cliente la comodidad en la gestión de otras tareas asociadas a las de mayor prioridad.

### 2.8.1 Plan de duración de las iteraciones

Según define la metodología XP se crea el plan de duración de las iteraciones. En el mismo se especifican más detalladamente el orden de las HU dentro de cada iteración, así como la iteración completa de la misma.

*Tabla 7 Plan de duración de las iteraciones*

| Iteraciones | HU a Implementar | Duración |
|---|---|---|
| **1** | Crear Área de resultados claves<br>Modificar Área de resultados claves<br>Obtener Área de resultados claves<br>Eliminar Área de resultados claves<br>Crear Objetivo Estratégicos<br>Modificar Objetivo Estratégicos<br>Obtener Objetivo Estratégicos<br>Eliminar Objetivo Estratégicos<br>Crear Criterios de Medidas<br>Modificar Criterios de Medidas<br>Obtener Criterios de Medidas<br>Eliminar Criterios de Medidas<br>Crear Indicadores<br>Modificar Indicadores<br>Obtener Indicadores<br>Eliminar Indicadores<br>Crear Usuarios<br>Modificar Usuarios<br>Obtener Usuarios<br>Eliminar Usuarios<br>Crear Evidencias | 3 semanas |

| | | |
|---|---|---|
| | Modificar Evidencias<br>Obtener Evidencias<br>Eliminar Evidencias | |
| **2** | Autenticar Usuario<br>Evaluar a un usuario<br>Exportar evolución a PDF<br>Cambio de estado del Criterio<br>Adjuntar archivo a la evidencia<br>Cambiar la contraseña<br>Asignar indicadores<br>Crear indicador personal<br>Cambiar estado del indicador | 3.6 semanas |
| **3** | Restablecer contraseña<br>Buscar usuario<br>Mostrar porcentaje<br>Notificar al usuario<br>Notificar al jefe de Área<br>Informe del Área<br>Denegar indicador<br>Añadir observación<br>Obtener Áreas por año<br>Obtener Indicadores de un usuario por año<br>Establecer fecha a una evidencia<br>Crear un año<br>Eliminar un año<br>Crear un departamento<br>Eliminar un departamento<br>Obtener evaluación de un usuario en un año<br>Evaluar usuario en un año | 3 semanas |

### 2.9 Plan de entregas

El cronograma de entrega establece cuáles son las HU que serán agrupadas para conformar las entregas y el orden de estas.

*Tabla 8 Plan de entregas*

| Historias de Usuario | Fin de 1ra Iteración | Fin de 2da Iteración | Fin de 3ra Iteración |
|---|---|---|---|
| **Crear Área** | Versión 1.0 | Finalizado | Finalizado |
| **Modificar Área** | Versión 1.0 | Finalizado | Finalizado |
| **Obtener Área** | Versión 1.0 | Finalizado | Finalizado |
| **Eliminar Área** | Versión 1.0 | Finalizado | Finalizado |
| **Crear Objetivo Estratégicos** | Versión 1.0 | Finalizado | Finalizado |

| | | | |
|---|---|---|---|
| **Modificar Objetivo Estratégicos** | Versión 1.0 | Finalizado | Finalizado |
| **Obtener Objetivo Estratégicos** | Versión 1.0 | Finalizado | Finalizado |
| **Eliminar Objetivo Estratégicos** | Versión 1.0 | Finalizado | Finalizado |
| **Crear Criterios de Medidas** | Versión 1.0 | Finalizado | Finalizado |
| **Modificar Criterios de Medidas** | Versión 1.0 | Finalizado | Finalizado |
| **Obtener Criterios de Medidas** | Versión 1.0 | Finalizado | Finalizado |
| **Eliminar Criterios de Medidas** | Versión 1.0 | Finalizado | Finalizado |
| **Crear Indicadores** | Versión 1.0 | Finalizado | Finalizado |
| **Modificar Indicadores** | Versión 1.0 | Finalizado | Finalizado |
| **Obtener Indicadores** | Versión 1.0 | Finalizado | Finalizado |
| **Eliminar Indicadores** | Versión 1.0 | Finalizado | Finalizado |
| **Crear Usuarios** | Versión 1.0 | Finalizado | Finalizado |
| **Modificar Usuarios** | Versión 1.0 | Finalizado | Finalizado |
| **Obtener Usuarios** | Versión 1.0 | Finalizado | Finalizado |
| **Eliminar Usuarios** | Versión 1.0 | Finalizado | Finalizado |
| **Crear Evidencias** | Versión 1.0 | Finalizado | Finalizado |
| **Modificar Evidencias** | Versión 1.0 | Finalizado | Finalizado |
| **Obtener Evidencias** | Versión 1.0 | Finalizado | Finalizado |
| **Eliminar Evidencias** | Versión 1.0 | Finalizado | Finalizado |
| **Autenticar Usuario** | - | Versión 1.0 | Finalizado |
| **Evaluar a un usuario** | - | Versión 1.0 | Finalizado |
| **Exportar evolución a PDF** | - | Versión 1.0 | Finalizado |
| **Cambio de estado del Criterio** | - | Versión 1.0 | Finalizado |
| **Adjuntar archivo a la evidencia** | - | Versión 1.0 | Finalizado |
| **Cambiar la contraseña** | - | Versión 1.0 | Finalizado |
| **Asignar indicadores** | - | Versión 1.0 | Finalizado |
| **Crear indicador personal** | - | Versión 1.0 | Finalizado |
| **Cambiar estado del indicador** | - | Versión 1.0 | Finalizado |
| **Restablecer contraseña** | - | - | Versión 1.0 |
| **Buscar usuario** | - | - | Versión 1.0 |
| **Mostrar porcentaje** | - | - | Versión 1.0 |
| **Notificar al usuario** | - | - | Versión 1.0 |
| **Notificar al jefe de Área** | - | - | Versión 1.0 |

| | | | |
|---|---|---|---|
| **Informe del Área** | - | - | Versión 1.0 |
| **Denegar indicador** | - | - | Versión 1.0 |
| **Añadir observación** | - | - | Versión 1.0 |
| **Obtener Áreas por año** | - | - | Versión 1.0 |
| **Obtener Indicadores de un usuario por año** | - | - | Versión 1.0 |
| **Establecer fecha a una evidencia** | - | - | Versión 1.0 |
| **Crear un año** | - | - | Versión 1.0 |
| **Eliminar un año** | - | - | Versión 1.0 |
| **Crear un departamento** | - | - | Versión 1.0 |
| **Eliminar un departamento** | - | - | Versión 1.0 |
| **Obtener evaluación de un usuario en un año** | - | - | Versión 1.0 |
| **Evaluar usuario en un año** | - | - | Versión 1.0 |

### 2.10 Patrones de Arquitectura de Software

Un patrón arquitectónico es una solución general y reutilizable a un problema común en la arquitectura de software dentro de un contexto dado. Los patrones arquitectónicos son similares al patrón de diseño de software, pero tienen un alcance más amplio.

#### 2.10.1 Patrón de modelo-vista-controlador

Este patrón, también conocido como patrón MVC, divide una aplicación interactiva en 3 partes, como

1. **Modelo**: contiene la funcionalidad y los datos básicos
2. **Vista**: muestra la información al usuario (se puede definir más de una vista)
3. **Controlador**: maneja la entrada del usuario

Esto se hace para separar las representaciones internas de información de las formas en que se presenta y acepta la información del usuario. Desacopla los componentes y permite la reutilización eficiente del código (huaman 2018).

## 2.11 Patrones de Diseño

Los patrones de diseño son una solución general, reutilizable y aplicable a diferentes problemas de diseño de software. Se trata de plantillas que identifican problemas en el sistema y proporcionan soluciones apropiadas a problemas generales a los que se han enfrentado los desarrolladores durante un largo periodo de tiempo, a través de prueba y error (Miriam 2020).

### 2.11.1 Patrones GRASP

Para la asignación general de responsabilidades en el software existen patrones denominados Patrones Generales de Asignación de Responsabilidades (GRASP, por sus siglas en inglés). En la programación orientada a objeto una de las cosas más complicadas, consiste en elegir las clases adecuadas y decidir como estas clases deben interactuar. En especial cuando se utiliza metodologías rápidas como XP que se basan en el proceso de desarrollo continuo. Haciendo que sea inevitable elegir cuidadosamente las responsabilidades de cada clase en la primera codificación y, fundamentalmente, en la refactorización del programa (Mora 2003).

Seguidamente se exponen algunos patrones utilizados dentro del contexto del sistema a implementar:

- **Experto:** se evidencia en las clases que definen los modelos del sistema, las cuales cuentan con toda la información referentes a ellas y son responsable de proporcionar solo los atributos que son públicos. Dichas clases son: Area, Criterion, Evidence, Indicator, Objective, y User.

- **Creador:** se evidencia en las clases: areaController, criterionController, evidenceController, indicatorController, objectiveController y userController. Dichas clases son las encargadas de instanciar y manipular las clases modelos del sistema.

- **Controlador:** se evidencia en las clases: areaRouter, criterionRouter, evidenceRouter, indicatorRouter, objectiveRouter y userRouter. Dichas clases son las intermediarias entre una determinada petición y el algoritmo que procesa la información, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado.

- **Alta cohesión y Bajo acoplamiento:** se evidencia de manera general (en la medida de lo posible) en todas las clases del sistema.

## 2.12 Tarjetas CRC

Las tarjetas Clases, Responsabilidad y Colaboración (CRC) trabajan con la técnica de modelado basado en objetos, representando cada tarjeta CRC a un objeto, identificando las clases y sus responsabilidades. Las tarjetas están compuestas por el **nombre de la clase** colocado como **título**, en la parte izquierda se colocan las **responsabilidades** (funcionalidades) y en la parte derecha las **clases** que se implican en cada responsabilidad.

La creación de las mismas tiene gran utilidad, pues da una idea clara de la arquitectura del sistema, distribución de las clases, paquetes y la ubicación de las diferentes responsabilidades sobre la lógica del negocio (Beck y Cunningham 2004).

*Tabla 9 Tarjeta CRC areaController*

| **Tarjeta CRC** | |
|---|---|
| **Clase:** areaController | |
| **Responsabilidad** | **Colaboración** |
| • areaNames(req, res): retorna una lista con los nombres de todas las áreas de resultados claves.<br>• areaGet(req, res): retorna todas las áreas de resultados claves que existen en la base de datos.<br>• areaById(req, res): retorna un área de resultados claves en específico.<br>• areaPost(req, res): crea un área de resultados claves con los datos que se recibe.<br>• areaPut(req, res): actualiza un área de resultados claves en específico con los datos que se recibe.<br>• addObjectives(req, res): añade objetivos estratégicos a un área de resultados claves en específico.<br>• areaDelete(req, res): elimina un área en específico | areaPercentage<br>indicatorResponse<br>removeModels<br>area<br>objective |

*Tabla 10 Tarjeta CRC evidenceController*

| **Tarjeta CRC** | |
|---|---|
| **Clase:** evidenceController | |
| **Responsabilidad** | **Colaboración** |
| • evidencaGet(req, res): retorna una lista con todas las evidencias que existen en la base de datos.<br>• evidencaGetFile(req, res): retorna el archivo de una evidencia en específico.<br>• evidencaPost(req, res): crear una evidencia con los datos que se recibe.<br>• evidencaPut(req, res): actualiza una evidencia en específico con los datos que se recibe.<br>• evidencaDelete(req, res): elimina una evidencia en específico.<br>• evidencaUpload(req, res): carga y asocia un archivo a una evidencia en específico. | modifyCriterion<br>removeModels<br>uploadFile<br>evidence<br>indicator<br>user |

*Tabla 11 Tarjeta CRC indicatorController*

| **Tarjeta CRC** | |
|---|---|
| **Clase:** indicatorController | |
| **Responsabilidad** | **Colaboración** |
| • indicatorGet(req, res): retorna una lista de todos los indicadores que existen en la base de datos.<br>• indicatorByCategory(req, res): retorna una lista de los indicadores modelos ordenada por categoría.<br>• indicatorById(req, res): retorna un indicador en específico. | indicatorResponse<br>modifyCriterion<br>removeModels<br>criterion<br>indicator<br>user |

| | |
|---|---|
| • indicatorsByUser(req, res): retorna los indicadores asociados a un usuario en específico.<br>• indicatorPost(req, res): crea un indicador con los datos que se recibe.<br>• indicatorPostByCriterion(req, res): crea un indicador modelo con los datos que se recibe, asociado a un criterio de medida en específico.<br>• personalIndicatorPost(req, res): crea un indicador personal con los datos que se recibe, asociado a un usuario en específico.<br>• indicatorPut(req, res): actualiza un indicador en específico con los datos que se recibe.<br>• indicatorDelete(req, res): elimina un indicador en específico.<br>• indicatorModelDelete(req, res): elimina un indicador modelo en específico. | |

## 2.13 Conclusiones parciales

- La metodología de desarrollo seleccionada posibilitó definir los pasos y la organización para la implementación del sistema. Además, generar los artefactos correspondientes a las fases de planificación y diseño.

- La obtención de los requisitos del software en conjunto con la definición de las HU permitió identificar las funcionalidades que debe tener el sistema.

- Mediante la descripción detallada de las HU y la estimación de tiempo de esfuerzo de cada una, se pudieron definir las iteraciones enmarcadas en el plan de entrega de la realización del sistema.

- El uso del patrón arquitectónico MVC permitió lograr una estructura del software flexible a los cambios y mantenimientos, con poca dependencia entre clases.

# CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS

En el presente capítulo se describen las tareas de ingeniería generadas por cada una de las HU que se desarrollaron. También se exponen las pruebas de aceptación y las pruebas unitarias que fueron planificadas para el sistema con el objetivo de entregar un producto final probado y funcional al cliente.

## 3.1 Tareas de Ingeniería

Las tareas de ingeniería son utilizadas para identificar, controlar y definir cada una de las actividades que dan cumplimiento a las HU con el objetivo de facilitar la construcción del sistema. Para una mayor organización a continuación se describen solo algunas tareas de ingeniería en correspondencia con las iteraciones definidas, el resto pueden ser consultadas en el anexo 4.

### Tareas de Ingeniería para la iteración 1

Para la primera iteración, se definieron un total de 24 tareas de ingeniería. Se describen algunas de las tareas realizadas en esta iteración.

*Tabla 12 Tareas de Ingeniería 1*

| **Tarea** | |
|---|---|
| **Número de tarea:** 1 | **Número de Historia de Usuario:** 1 |
| **Nombre de la tarea:** Implementar la creación de una nueva área de resultados claves en el sistema | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 0.6 |
| **Programador responsable:** Arián León Benítez | |
| **Descripción:** Se implementa una funcionalidad que permite crear un área de resultados claves. | |

*Tabla 13 Tareas de Ingeniería 2*

| **Tarea** | |
|---|---|
| **Número de tarea:** 2 | **Número de Historia de Usuario:** 2 |
| **Nombre de la tarea:** Implementar la obtención de las áreas de resultados claves del sistema | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 0.6 |
| **Programador responsable:** Arián León Benítez | |
| **Descripción:** Se implementa una funcionalidad que permite obtener todas las áreas de resultados claves del sistema. | |

*Tabla 14 Tareas de Ingeniería 3*

| **Tarea** | |
|---|---|
| **Número de tarea:** 3 | **Número de Historia de Usuario:** 3 |

| | |
|---|---|
| **Nombre de la tarea:** Implementar la actualización de un área de resultados claves en el sistema | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 0.6 |
| **Programador responsable:** Arián León Benítez | |
| **Descripción:** Se implementa una funcionalidad que permite actualizar un área de resultados claves mediante su identificador. | |

*Tabla 15 Tareas de Ingeniería 4*

| **Tarea** | |
|---|---|
| **Número de tarea:** 4 | **Número de Historia de Usuario:** 4 |
| **Nombre de la tarea:** Implementar la eliminación de un área de resultados claves en el sistema | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 0.6 |
| **Programador responsable:** Arián León Benítez | |
| **Descripción:** Se implementa una funcionalidad que permite eliminar un área de resultados claves mediante su identificador. | |

**Tareas de Ingeniería para la iteración 2**

Para una segunda iteración, se elaboraron un total de 10 tareas de ingeniería. Se describen algunas de las tareas realizadas, el resto pueden ser consultadas en el anexo 4 de la investigación.

*Tabla 16 Tareas de Ingeniería 25*

| **Tarea** | |
|---|---|
| **Número de tarea:** 25 | **Número de Historia de Usuario:** 25 |
| **Nombre de la tarea:** Implementar la autenticación de un usuario en el sistema | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 1 |
| **Programador responsable:** Javier Ceballo Perez | |
| **Descripción:** Se implementa una funcionalidad que permita autenticar un usuario mediante su nombre de usuario y contraseña. | |

*Tabla 17 Tareas de Ingeniería 26*

| **Tarea** | |
|---|---|
| **Número de tarea:** 26 | **Número de Historia de Usuario:** 26 |
| **Nombre de la tarea:** Implementar el cierre de sección de un usuario en el sistema | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 1 |
| **Programador responsable:** Javier Ceballo Perez | |
| **Descripción:** Se implementa una funcionalidad que permita cerrar sección un usuario en el sistema | |

*Tabla 18 Tareas de Ingeniería 27*

| **Tarea** | |
|---|---|
| **Número de tarea:** 27 | **Número de Historia de Usuario:** 27 |
| **Nombre de la tarea:** Implementar la evaluación de un usuario en el sistema | |

| | |
|---|---|
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 1 |
| **Programador responsable:** Javier Ceballo Perez | |
| **Descripción:** Se implementa una funcionalidad que permita evaluar a un usuario por cada categoría de indicadores y devolver una calificación | |

*Tabla 19 Tareas de Ingeniería 28*

| **Tarea** | |
|---|---|
| **Número de tarea:** 28 | **Número de Historia de Usuario:** 28 |
| **Nombre de la tarea:** Implementar la exportación de una evaluación de un usuario a PDF | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 1 |
| **Programador responsable:** Javier Ceballo Perez | |
| **Descripción:** Se implementa una funcionalidad que permita exportar un PDF con la evaluación de usuario, así como todos los indicadores que tenga cumplidos el mismo | |

**Tareas de Ingeniería para la iteración 3**

Para una tercera iteración, se elaboraron un total de 17 tareas de ingeniería. Se describen algunas de las tareas realizadas, el resto pueden ser consultadas en el anexo 4 de la investigación.

*Tabla 20 Tareas de Ingeniería 36*

| **Tarea** | |
|---|---|
| **Número de tarea:** 36 | **Número de Historia de Usuario:** 36 |
| **Nombre de la tarea:** Implementar la búsqueda de usuarios en el sistema | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 1 |
| **Programador responsable:** Javier Ceballo Perez | |
| **Descripción:** Se implementa una funcionalidad que permita obtener los usuarios que coincidan sus nombres y nombres de usuario con un carácter dado | |

*Tabla 21 Tareas de Ingeniería 37*

| **Tarea** | |
|---|---|
| **Número de tarea:** 37 | **Número de Historia de Usuario:** 37 |
| **Nombre de la tarea:** Implementar obtención del porcentaje de cumplimiento de un área de resultados claves | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 1 |
| **Programador responsable:** Javier Ceballo Perez | |
| **Descripción:** Se implementa una funcionalidad que permita obtener el porcentaje de criterios de medida en estado de “Cumplido” y “Sobre cumplido” de un área de resultados claves | |

*Tabla 22 Tareas de Ingeniería 38*

| **Tarea** | |
|---|---|
| **Número de tarea:** 38 | **Número de Historia de Usuario:** 38 |
| **Nombre de la tarea:** Implementar la notificación dirigida al usuario autenticado | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 2.5 |
| **Programador responsable:** Arián León Benítez | |

| |
|---|
| **Descripción:** Se implementa una funcionalidad que permite notificar al usuario autenticado cuando se la fue asignado un indicador o se eliminó uno de sus indicadores. |

*Tabla 23 Tareas de Ingeniería 39*

| **Tarea** | |
|---|---|
| **Número de tarea:** 39 | **Número de Historia de Usuario:** 39 |
| **Nombre de la tarea:** Implementar la notificación dirigida al jefe de área | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 2.5 |
| **Programador responsable:** Arián León Benítez | |
| **Descripción:** Se implementa una funcionalidad que permite notificar al jefe de departamento cuando unos de los usuarios pertenecientes al mismo departamento que el jefe, cumplan algún de los indicadores que le fueron asignados o creen un indicador personal. | |

Luego de ser definidas las tareas ingenieriles, se hace necesario establecer un conjunto de pruebas para comprobar la calidad de la solución implementada. A continuación, se detalla este proceso.

### 3.2 Pruebas

Las pruebas de software comprenden el conjunto de actividades que se realizan para identificar posibles fallos de funcionamiento, configuración o usabilidad de la aplicación, por medio de pruebas sobre el comportamiento de esta. Constituyen a elevar la calidad de los productos desarrollados y a la seguridad de los programadores a la hora de introducir cambios o modificaciones.

La metodología XP divide las pruebas en dos grupos: pruebas unitarias, desarrolladas por los programadores, encargadas de verificar el código de forma automática y las pruebas de aceptación, destinadas a evaluar si al final de una iteración se obtuvo la funcionalidad requerida, además de comprobar que dicha funcionalidad sea la esperada por el cliente (Ramos 2021).

#### 3.2.1 Pruebas de aceptación

El objetivo de las pruebas de aceptación es validar que un sistema cumple con el funcionamiento esperado y permitir al usuario de dicho sistema que determine su aceptación, desde el punto de vista de su funcionalidad y rendimiento. Estas pruebas derivan de las Historias de Usuarios que se han implementado como parte de la liberación del software. Las pruebas de aceptación son definidas por el usuario del sistema y preparadas por el equipo de desarrollo. A continuación, se muestra una representación de las pruebas de aceptación a realizarse en cada iteración.

**Pruebas de aceptación para la Iteración I**

Para la primera iteración, se definieron un total de 24 casos de pruebas de aceptación. Se describen algunas de las pruebas realizadas, el resto pueden ser consultadas en el anexo 5.

*Tabla 24 Pruebas de aceptación HU1-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU1_P1 | **Historia de Usuario:** 1 |
| **Nombre:** Crear un área de resultados claves | |
| **Descripción:** Prueba de funcionalidad: crear un área de resultados claves | |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol decano | |
| **Pasos de ejecución:**<br>1. El decano accede a la sección de las áreas de resultados claves<br>2. El decano selecciona la opción para crear un área de resultados claves<br>3. El decano introduce los datos<br>4. El decano crea el área de resultados claves | |
| **Resultado:** Satisfactorio | |

*Tabla 25 Pruebas de aceptación HU2-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU2_P1 | **Historia de Usuario:** 2 |
| **Nombre:** Actualizar un área de resultados claves | |
| **Descripción:** Prueba de funcionalidad: actualizar un área de resultados claves | |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol decano<br>• Debe existir al menos un área de resultados claves en el sistema | |
| **Pasos de ejecución:**<br>1. El decano accede a la sección de las áreas de resultados claves<br>2. El decano selecciona la opción “modificar” perteneciente a un área de resultados claves<br>3. El decano remplaza los datos<br>4. Se actualiza el área de resultados claves | |
| **Resultado:** Satisfactorio | |

*Tabla 26 Pruebas de aceptación HU3-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU3_P1 | **Historia de Usuario:** 3 |
| **Nombre:** Obtener áreas de resultados claves | |
| **Descripción:** Prueba de funcionalidad: obtener áreas de resultados claves | |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol decano o jefe de área | |
| **Pasos de ejecución:**<br>1. El decano o jefe de área accede a la sección de las áreas de resultados claves<br>2. Automáticamente se obtiene todas las áreas de resultados claves correspondientes | |
| **Resultado:** Satisfactorio | |

*Tabla 27 Pruebas de aceptación HU4-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU4_P1 | **Historia de Usuario:** 4 |
| **Nombre:** Eliminar un área de resultados claves | |
| **Descripción:** Prueba de funcionalidad: eliminar un área de resultados claves | |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol decano<br>• Debe existir al menos un área en el sistema | |
| **Pasos de ejecución:**<br>1. El decano accede a la sección de las áreas de resultados claves<br>2. El decano selecciona la opción “eliminar” perteneciente a un área<br>3. El decano selecciona la opción “aceptar”<br>4. El decano elimina el área | |
| **Resultado:** Satisfactorio | |

**Pruebas de aceptación para la Iteración 2**

Para la segunda iteración, se definieron un total de 10 casos de pruebas de aceptación. Se describen algunas de las pruebas realizadas, el resto pueden ser consultadas en el anexo 5.

*Tabla 28 Pruebas de aceptación HU25-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU25_P1 | **Historia de Usuario:** 25 |
| **Nombre:** Autenticar usuario | |
| **Descripción:** Prueba de funcionalidad: autenticar usuario | |
| **Condiciones de ejecución:**<br>• El usuario no debe estar autenticado | |
| **Pasos de ejecución:**<br>1. El usuario accede a la página de inicio de sección<br>2. El usuario ingresa nombre de usuario y contraseña<br>3. El usuario selecciona la opción de “iniciar sección”<br>4. El usuario se autentica | |
| **Resultado:** Satisfactorio | |

*Tabla 29 Pruebas de aceptación HU25-P2*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU25_P2 | **Historia de Usuario:** 25 |
| **Nombre:** Cerrar sección | |
| **Descripción:** Prueba de funcionalidad: cerrar sección | |
| **Condiciones de ejecución:**<br>• El usuario debe estar autenticado | |
| **Pasos de ejecución:**<br>1. El usuario accede a su menú en la parte superior derecha de la pantalla<br>2. El usuario selecciona la opción “salir”<br>3. El usuario cierra la sección | |
| **Resultado:** Satisfactorio | |

*Tabla 30 Pruebas de aceptación HU26-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU26_P1 | **Historia de Usuario:** 26 |
| **Nombre:** Evaluar a un usuario | |
| **Descripción:** Prueba de funcionalidad: evaluar a un usuario | |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol jefe de departamento<br>• Debe de haber usuarios en el sistema que pertenezcan a la misma área del jefe | |
| **Pasos de ejecución:**<br>1. El jefe de departamento accede a la sección de usuarios<br>2. El jefe de departamento selecciona la opción “evaluación” que le pertenece al usuario seleccionado<br>3. El jefe de departamento selecciona la opción “evaluar”<br>4. El jefe de departamento selecciona su evaluación por categorías<br>5. El jefe de departamento selecciona la opción “aceptar”<br>6. El jefe de departamento evaluó a un usuario | |
| **Resultado:** Satisfactorio | |

*Tabla 31 Pruebas de aceptación HU27-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU27_P1 | **Historia de Usuario:** 27 |
| **Nombre:** Generar evaluación personal en PDF | |
| **Descripción:** Prueba para la funcionalidad: generar documento PDF para la evaluación personal de un usuario. | |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol de jefe de departamento. | |
| **Pasos de ejecución:**<br>1. El jefe de departamento selecciona la opción usuarios.<br>2. El jefe de departamento selecciona en el menú del usuario a evaluar la opción “Evaluación”.<br>3. El jefe de departamento selecciona la opción de Evaluar y se emite una evaluación.<br>4. Se selecciona la opción de “Exportar PDF”. | |
| **Resultado:** Satisfactorio | |

**Pruebas de aceptación para la Iteración 3**

Para la tercera iteración, se definieron un total de 18 casos de pruebas de aceptación. Se describen algunas de las pruebas realizadas, el resto pueden ser consultadas en el anexo 5.

*Tabla 32 Pruebas de aceptación HU35-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU35_P1 | **Historia de Usuario:** 35 |
| **Nombre:** Buscar usuarios | |
| **Descripción:** Prueba de funcionalidad: buscar usuarios | |

| |
|---|
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol decano o jefe de departamento<br>• Debe existir al menos un usuario en el sistema |
| **Pasos de ejecución:**<br>1. El decano o jefe de departamento accede a la sección de los usuarios<br>2. El decano o jefe de departamento selecciona el campo “buscar usuarios”<br>3. El decano o jefe de departamento introduce el texto a buscar<br>4. Se muestran los usuarios que coincidan con el texto |
| **Resultado:** Satisfactorio |

*Tabla 33 Pruebas de aceptación HU36-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU36_P1 | **Historia de Usuario:** 36 |
| **Nombre:** Mostrar porcentaje del cumplimiento de las áreas de resultados claves | |
| **Descripción:** Prueba de funcionalidad: mostrar porcentaje del cumplimiento de las áreas | |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol decano o jefe de área | |
| **Pasos de ejecución:**<br>1. El decano o jefe de área accede a la sección de las áreas<br>2. Automáticamente se obtiene el porcentaje de cumplimiento de las áreas correspondientes | |
| **Resultado:** Satisfactorio | |

*Tabla 34 Pruebas de aceptación HU37-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU37_P1 | **Historia de Usuario:** 37 |
| **Nombre:** Mostrar notificación al usuario autenticado | |
| **Descripción:** Prueba para la funcionalidad: Mostrar las notificaciones no visualizadas al usuario autenticado | |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado.<br>• El usuario autenticado debe tener al menos una notificación pendiente que visualizar. | |
| **Pasos de ejecución:**<br>1. El usuario autenticado marca en el icono de notificaciones.<br>2. El usuario autenticado despliega las notificaciones que no han sido vistas, correspondientes al usuario autenticado.<br>3. Se muestra la información correspondiente.<br>4. Se eliminan de manera automática las notificaciones visualizadas, correspondientes al usuario autenticado. | |
| **Resultado:** Satisfactorio | |

*Tabla 35 Pruebas de aceptación HU38-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU38_P1 | **Historia de Usuario:** 38 |
| **Nombre:** Mostrar notificación al jefe de departamento | |

| |
|---|
| **Descripción:** Prueba para la funcionalidad: Mostrar las notificaciones no visualizadas al jefe de departamento |
| **Condiciones de ejecución:**<br>• El jefe de departamento debe estar previamente autenticado.<br>• El jefe de departamento debe tener al menos una notificación pendiente que visualizar. |
| **Pasos de ejecución:**<br>1. El jefe de departamento marca en el icono de notificaciones.<br>2. Se despliegan las notificaciones que no han sido vistas, correspondientes al jefe de departamento.<br>3. Se muestra la información correspondiente.<br>4. Se eliminan de manera automática las notificaciones visualizadas, correspondientes al jefe de departamento. |
| **Resultado:** Satisfactorio |

**Análisis de las pruebas de aceptación**

Se desarrollaron un total de 52 casos de pruebas de aceptación. Estas pruebas fueron realizadas de forma organizada, por cada iteración definida. A continuación, se muestra en una gráfica, la cantidad de pruebas satisfactorias, no satisfactorias y solucionadas alcanzados en cada iteración.

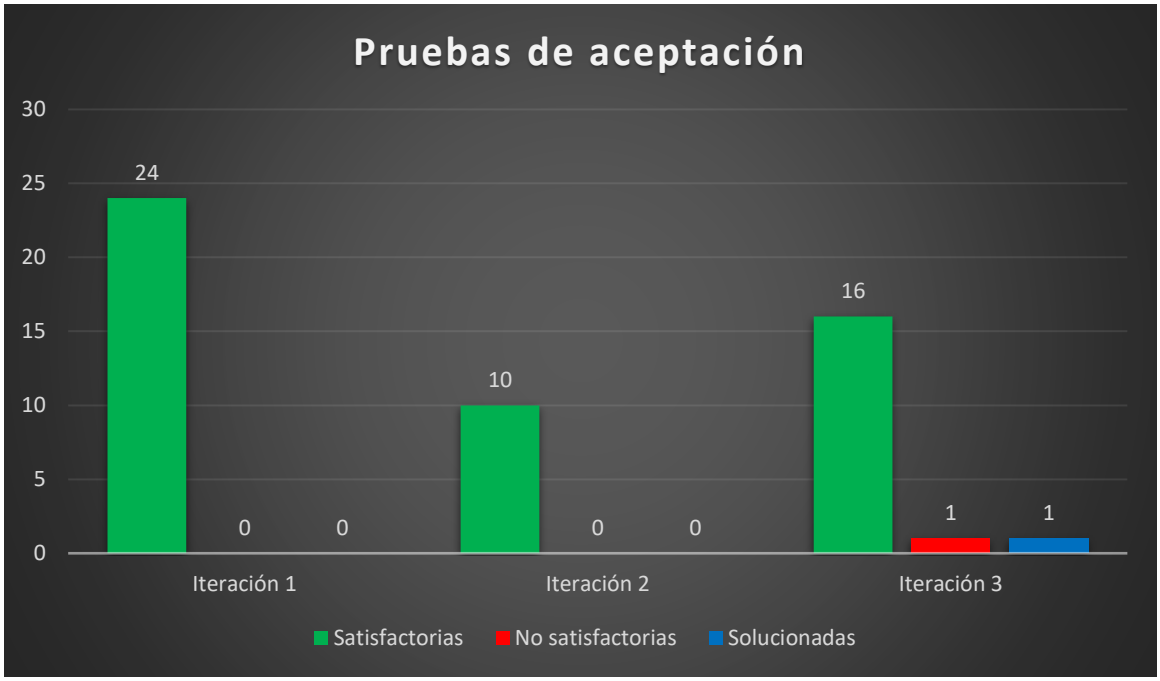



*Ilustración 3 Gráfico de las pruebas de aceptación*

Como se puede observar en la gráfica, en la primera y segunda iteraciones se realizaron 24 y 10 pruebas respectivamente, donde todas son evaluadas de resultado satisfactorio. En la tercera iteración se realizaron un total de 18 pruebas, de ellas 16 alcanzaron el nivel de satisfacción,

mientras que una de las pruebas detectó que al denegar un indicador cumplido este mantenía el estado de cumplido en el sistema; lo que permitió corregir la falla y lograr que se visualizara el estado del indicador correctamente. Con estas comprobaciones, se obtiene un 100 % de satisfacción en la iteración final del producto, comprobando el correcto funcionamiento de las funcionalidades implementadas.

### 3.2.2 Pruebas unitarias

Las pruebas unitarias son la forma de comprobar el correcto funcionamiento de una unidad de código mediante pequeñas pruebas que validad el comportamiento de un objetivo y la lógica. Para el desarrollo de estas se utilizó el framework jest debido al lenguaje utilizado. A continuación, se muestra una representación de las pruebas unitarias a realizarse en cada iteración.

**Pruebas unitarias para la Iteración 1:**

Para la primera iteración se definieron un total de 26 pruebas unitarias. A continuación, se ilustran algunas de las pruebas realizadas, el resto pueden ser consultadas en el anexo 6 de la investigación.

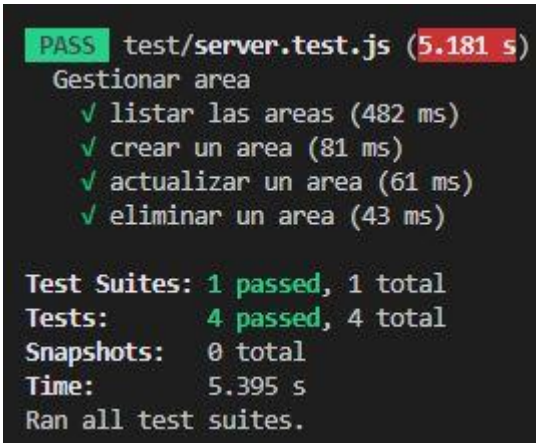



*Ilustración 4 Prueba Unitaria: Gestionar área*

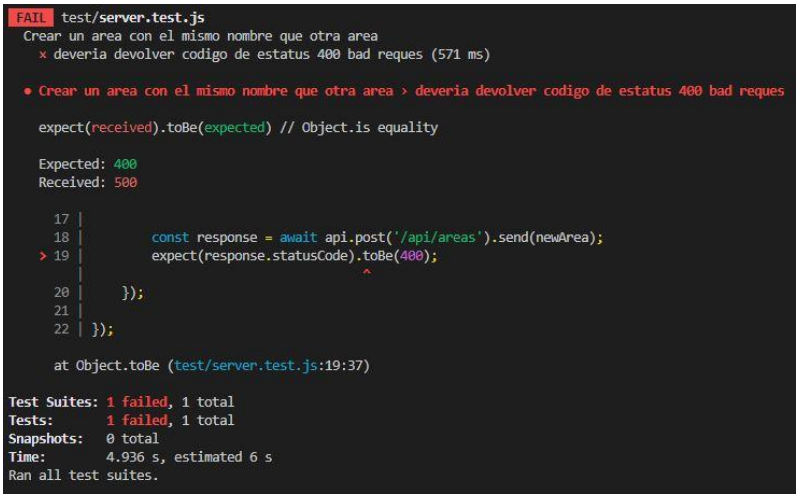



*Ilustración 5 Prueba Unitaria: Crear un área con el mismo nombre que otra área*

De las pruebas realizadas 24 fueron satisfactorias y una no satisfactoria, la cual se le acometieron los arreglos correspondientes haciendo que en su segundo intento fuera satisfactorio. Como se muestra en la siguiente ilustración.

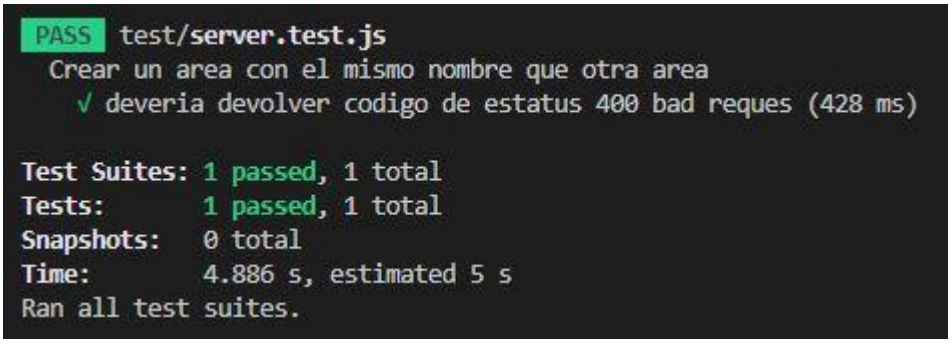



*Ilustración 6 Prueba Unitaria: Crear un área con el mismo nombre que otra área (Aceptada)*

**Pruebas unitarias para la Iteración 2:**

En la segunda iteración se definieron 10 pruebas unitarias. A continuación, se ilustran algunas de las pruebas realizadas, el resto pueden ser consultadas en el anexo 6 de la investigación.

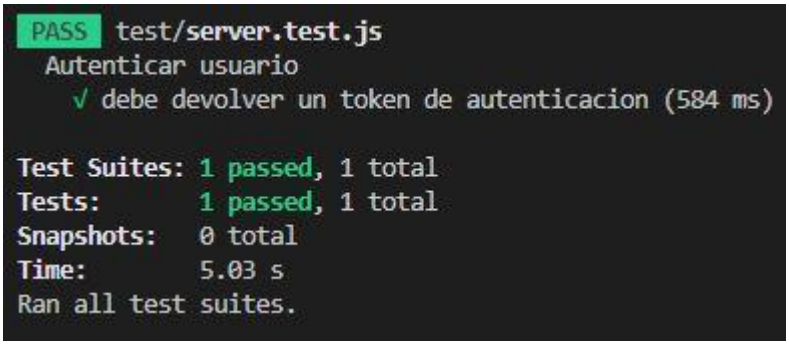



*Ilustración 7 Prueba Unitaria: Autenticar usuario*

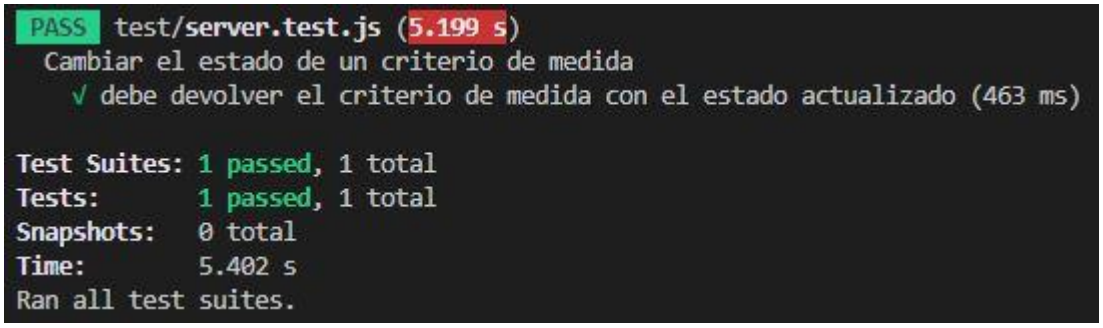



*Ilustración 8 Prueba Unitaria: Cambiar el estado de un criterio de medida*

De las pruebas realizadas nueve fueron satisfactorias y una no satisfactoria, la cual fue corregida y ejecutada nuevamente siendo satisfactoria esa vez. Cabe destacar que una de las pruebas a pesar de tener un resultado favorable, se decidió hacer ajuste de optimización y repetir dicha prueba ya que la respuesta no se obtenía con la prontitud deseada. Lográndose en su segunda ejecución una mejora en su tiempo de respuesta.

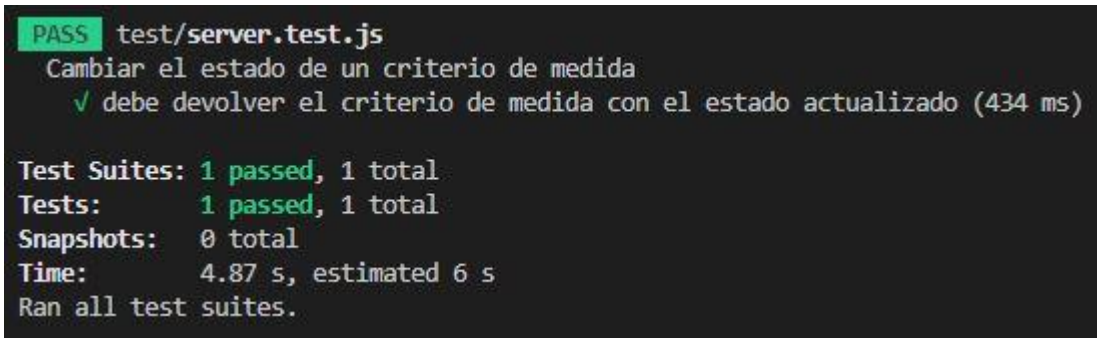



*Ilustración 9 Prueba Unitaria: Cambiar el estado de un criterio de medida (eficiente)*

**Pruebas unitarias para la Iteración 3:**

En la tercera iteración se definieron 14 pruebas unitarias. De las cuales todas obtuvieron un resultado satisfactorio. A continuación, se ilustra una de ella, el resto pueden ser consultadas en el anexo 6 de la investigación.

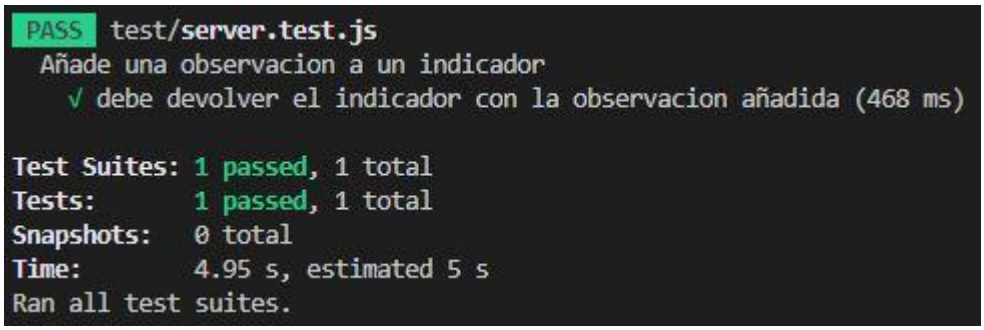



*Ilustración 10 Prueba Unitaria: Añade una observación a un indicador*

**Análisis de las pruebas de unitarias**

Se desarrollado un total de 50 pruebas unitarias. Estas pruebas fueron realizadas de forma organizada, por cada iteración definida. A continuación, se muestra en una gráfica, la cantidad de pruebas satisfactorias, no satisfactorias y solucionadas alcanzadas en cada iteración.

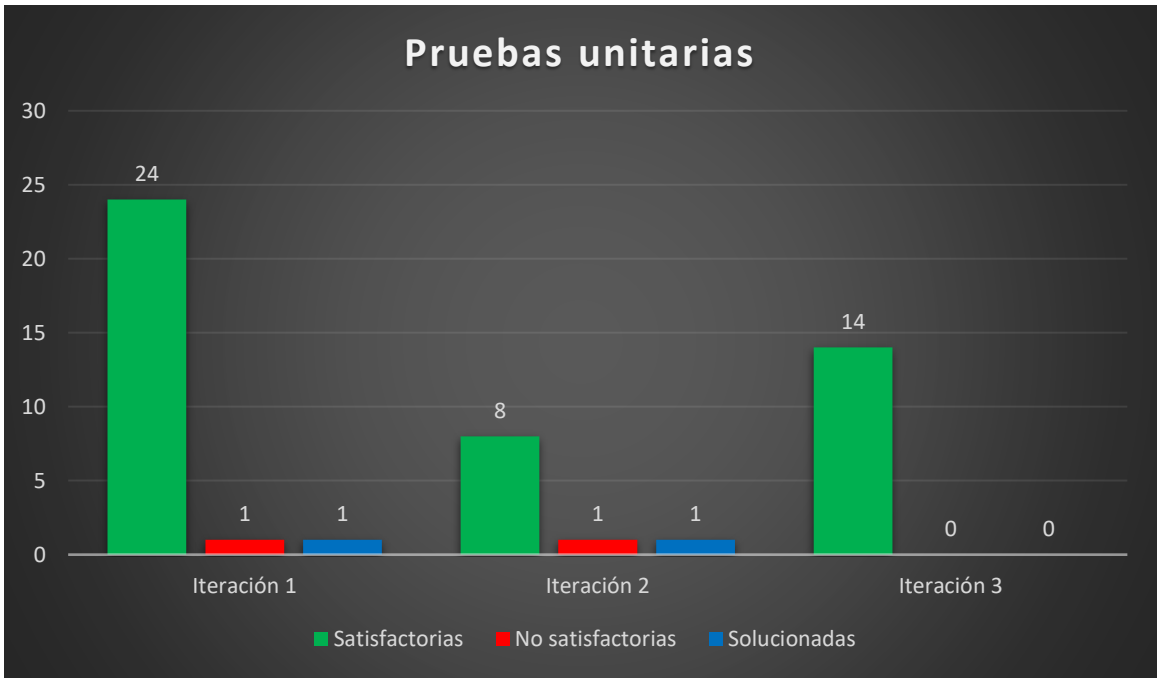



*Ilustración 11 Gráfico de las pruebas unitarias*

Como se puede observar en la representación, en la primera iteración se realizaron un total de 26 pruebas, de ellas 24 alcanzaron el nivel de satisfacción, mientras que, en una de las pruebas, detectó que, al crear un área con un nombre de otra área ya existente en el sistema, no se devolvía una respuesta controlada donde se notificaba del uso del nombre ya está ocupado. En su lugar el sistema respondía de manera no controlada, provocando la caída del sitio. Lo que permitió corregir la falla e implementar la solución de la misma. En la segunda iteración se realizaron un total de 10 pruebas, de ellas ocho alcanzaron el nivel de satisfacción, mientras que, en una de las pruebas se detectó que al crear un indicador personal no se marcaba como cumplido ya que en su implementación cuando se crea un indicador nuevo por defecto su atributo status tiene un valor de false. Esto permitió corregir la falla e implementar la solución de la misma. En tanto, en la tercera iteración se realizaron 14 donde todas son evaluadas de resultado satisfactorio. Con estas comprobaciones, se obtiene un 100 % de satisfacción en la iteración final del producto, comprobando el correcto funcionamiento de las funcionalidades implementadas.

### 3.3 Conclusiones del capítulo

- Se desarrolló el proceso de implementación del sistema mediante tareas de ingeniería permitiendo especificar los procedimientos para dar cumplimiento a las Historias de Usuario.

- Se definieron y se realizaron las pruebas de aceptación y las pruebas unitarias, las cuales permitieron detectar fallas en el sistema y corregirlas, lo que contribuyó una mejora del sistema.

## CONCLUSIONES

Luego de culminada la presente investigación se puede arribar a las siguientes conclusiones:

- El estudio realizado como parte de la fundamentación sirvió de apoyo en la toma de decisiones con vistas al desarrollo del sistema de gestión para las evidencias de los objetivos del año.

- El análisis de herramientas y tecnologías existentes aportó la base de la selección tecnológica para el desarrollo de la solución propuesta.

- La elaboración de los artefactos propuestos por la metodología de desarrollo XP permitieron una mayor comprensión y organización del sistema desarrollado.

- La implementación del sistema propuesto permitió gestionar tanto la elaboración como el chequeo del cumplimiento de los objetivos estratégicos que se tracen en la facultad para un año determinado mediante evidencias.

- La ejecución de las pruebas permitió garantizar la calidad del software, quedando evidenciada en el acta de aceptación por parte del cliente.

**RECOMENDACIONES**

- La incorporación de un módulo al sistema donde se gestione las evidencias de las actividades realizadas por los estudiantes.
- La implementación de una función que liste a los usuarios de un departamento determinado por orden del nivel del cumplimiento de indicadores, ayudando en gran medida a la toma de decisiones.

## BIBLIOGRAFÍA


ÁLAVA MURILLO, M.R., 2022. Estudio comparativo de tecnologías web de componentes, REACT.JS VS VUE.JS VS ANGULAR.JS para el proceso de desarrollo de aplicaciones web. [en línea]. bachelorThesis. S.l.: Babahoyo: UTB-FAFI. 2022. [Consulta: 2 diciembre 2022]. Disponible en: http://dspace.utb.edu.ec/handle/49000/13034.

ALPHABET, 2010. Cómo Usar Chrome: Bienvenido a Chrome. GCFGlobal.org [en línea]. [Consulta: 2 diciembre 2022]. Disponible en: https://edu.gcfglobal.org/es/como-usar-chrome/bienvenido-a-chrome/1/.

ANDÍA VALENCIA, W., 2016. Enfoque metodológico para los objetivos estratégicos en la planificación del sector público. Industrial Data, vol. 19, no. 1, pp. 28-32.

ARRACHEQUESNE, D., 2022. Introduction | Socket.IO. [en línea]. [Consulta: 6 noviembre 2022]. Disponible en: https://socket.io/docs/v4/.

ASALE, R.- y RAE, 2022. evidencia | Diccionario de la lengua española. «Diccionario de la lengua española» - Edición del Tricentenario [en línea]. [Consulta: 6 noviembre 2022]. Disponible en: https://dle.rae.es/evidencia.

BECK, K. y ANDRES, C., 2004. Extreme Programming Explained: Embrace Change. S.l.: Addison-Wesley Professional. ISBN 978-0-13-405199-4.

BECK, K. y CUNNINGHAM, W., 2004. A Laboratory For Teaching Object-Oriented Thinking. [en línea]. [Consulta: 30 octubre 2022]. Disponible en: http://c2.com/doc/oopsla89/paper.html.

BERNERS-LEE, T., 2014. HTML: Lenguaje de etiquetas de hipertexto | MDN. [en línea]. [Consulta: 6 noviembre 2022]. Disponible en: https://developer.mozilla.org/es/docs/Web/HTML.

BLANCH, A., 2022. Programación y BBDD Archivos. Blog de arsys.es [en línea]. [Consulta: 6 noviembre 2022]. Disponible en: https://www.arsys.es/blog/programacion.

CÉSPEDES CALZADO, A., 2019. Sistema de Gestión para el cumplimiento de los objetivos estratégicos de la Facultad 4. [en línea]. bachelorThesis. S.l.: Universidad de las Ciencias Informáticas. Facultad 4. [Consulta: 6 diciembre 2022]. Disponible en: https://repositorio.uci.cu/jspui/handle/123456789/10119.

CHIAVENATO, I., 2019. Introducción a la teoría general de la administración: una visión integral de la moderna administración de las organizaciones. S.l.: McGraw Hill Interamericana. ISBN 978-1-4562-7210-4.

COSTA, G.I., 2007. Akademos, un Sistema Automatizado para la Gestión Académica. Serie Científica de la Universidad de las Ciencias Informáticas [en línea], vol. 1, no. 1. [Consulta: 6 noviembre 2022]. ISSN 2306-2495. Disponible en: https://publicaciones.uci.cu/index.php/serie/article/view/253.

CUAED, U., 2018. Opción múltiple V2.1. Opción múltiple V2.1 [en línea]. [Consulta: 23 noviembre 2022]. Disponible en:

https://programas.cuaed.unam.mx/repositorio/moodle/pluginfile.php/1023/mod_resource/content/1/contenido/evaluacion/opcion_multiple/index.html.

DOMAINLOGIC, I., 2022. Metodologías de desarrollo de software 2022. Domain Logic [en línea]. [Consulta: 6 noviembre 2022]. Disponible en: https://domainlogic.io/metodologias-de-desarrollo-de-software-2022/.

EICH, B., 2016. JavaScript | MDN. [en línea]. [Consulta: 6 noviembre 2022]. Disponible en: https://developer.mozilla.org/es/docs/Web/JavaScript.

GONZÁLEZ MARTÍNEZ, A.L., GONZÁLEZ GARAY, S. y ESPINOSA MARRERO, G., 2015. "Sistema para la Gestión de Evidencias de Indicadores de CTI, Posgrado e Investigación (GEVIN) [en línea]. bachelorThesis. S.l.: s.n. [Consulta: 6 diciembre 2022]. Disponible en: https://repositorio.uci.cu/jspui/handle/ident/8907.

HUAMAN, W.C., 2018. Los 10 patrones comunes de arquitectura de software. Medium [en línea]. [Consulta: 23 noviembre 2022]. Disponible en: https://medium.com/@maniakhitoccori/los-10-patrones-comunes-de-arquitectura-de-software-d8b9047edf0b.

IBÁÑEZ, L.H., 2016. Administración de Sistemas Gestores de Base de Datos. 2ª Edición. S.l.: Grupo Editorial RA-MA.

KEEPCODING, R., 2022. ¿Qué es Postman? | KeepCoding Tech School. [en línea]. [Consulta: 6 noviembre 2022]. Disponible en: https://keepcoding.io/blog/que-es-postman/.

LOSADA, J., 2017. Detalles de ORFEO Sistema de Gestión Documental -. [en línea]. [Consulta: 4 diciembre 2022]. Disponible en: https://orfeolibre.org/inicio/detalles-de-orfeo-sistema-de-gestion-documental/.

MICROSOFT, 2015. Documentation for Visual Studio Code. [en línea]. [Consulta: 6 noviembre 2022]. Disponible en: https://code.visualstudio.com/docs.

MIRIAM, 2020. Qué son los Patrones de Diseño de software / Design Patterns. Profile Software Services [en línea]. [Consulta: 6 noviembre 2022]. Disponible en: https://profile.es/blog/patrones-de-diseno-de-software/.

MONGODB INC., 2019. MongoDB Compass. MongoDB [en línea]. [Consulta: 6 noviembre 2022]. Disponible en: https://www.mongodb.com/es/products/compass.

MORA, R.C., 2003. Patrones de GRASP. Adictos al trabajo [en línea]. [Consulta: 6 noviembre 2022]. Disponible en: https://www.adictosaltrabajo.com/2003/12/22/grasp/.

NARANJO, F.J., 2015. Sistemas de gestión: Valor estratégico de las organizaciones. Rescatado el, vol. 19.

NODE.JS, 2022. Acerca. Node.js [en línea]. [Consulta: 6 noviembre 2022]. Disponible en: https://nodejs.org/es/about/.

PAMPUCH, B., 2022. pdfmake. [en línea]. [Consulta: 6 noviembre 2022]. Disponible en: https://pdfmake.github.io/docs/0.1/.

PÉREZ ROMÁN, A., 2020. Comparación de rendimiento entre bases de datos Relacionales, NoSQL y Blockchain Comparación de rendimiento entre PostgreSQL, MongoDB y Kaleido. En: Accepted: 2020-03-17T10:50:32Z [en línea], [Consulta: 6 noviembre 2022]. Disponible en: https://riuma.uma.es/xmlui/handle/10630/19413.

RAMOS, J., 2021. Los diferentes tipos de Pruebas de software. [en línea]. [Consulta: 6 noviembre 2022]. Disponible en: https://programacionymas.com/blog/tipos-de-testing-en-desarrollo-de-software.

R&IACUTE, E.C. del, 2001. ¿Que es importante?: establecimiento de las Areas de Resultados Clave. Folletos Gerenciales, vol. 5, no. 1, pp. 10-19. ISSN 17265851.

SAYAGO, J. y CHANGO, G., 2018. Análisis comparativo para aplicaciones web basados en servicios REST: stack MEAN y stack Java EE. Pontificia Universidad Católica del Ecuador, pp. 1-19.

SHARIFF, M., 2009. Alfresco 3 Enterprise Content Management Implementation. S.l.: Packt Publishing Ltd. ISBN 978-1-84719-737-5.

STOENESCU, R., 2015. Why Quasar? Quasar Framework [en línea]. [Consulta: 6 noviembre 2022]. Disponible en: https://quasar.dev/introduction-to-quasar#what-is-quasar.

TORVALDS, L., 2007. Git - Acerca del Control de Versiones. [en línea]. [Consulta: 6 noviembre 2022]. Disponible en: https://git-scm.com/book/es/v2/Inicio---Sobre-el-Control-de-Versiones-Acerca-del-Control-de-Versiones.

UCI, 2015. Gespro - EcuRed. [en línea]. [Consulta: 6 noviembre 2022]. Disponible en: https://www.ecured.cu/Gespro.

UCI, 2022. GESPRO 13.05 | Universidad de las Ciencias Informáticas. [en línea]. [Consulta: 6 noviembre 2022]. Disponible en: https://www.uci.cu/investigacion-y-desarrollo/productos/xedro/gespro-1305.

WALKE, J., 2013. Empezando – React. [en línea]. [Consulta: 6 noviembre 2022]. Disponible en: https://es.reactjs.org/docs/getting-started.html.

WIUM LIE, H., 2018. ¿Qué es el CSS? - Aprende sobre desarrollo web | MDN. [en línea]. [Consulta: 6 noviembre 2022]. Disponible en: https://developer.mozilla.org/es/docs/Learn/CSS/First_steps/What_is_CSS.

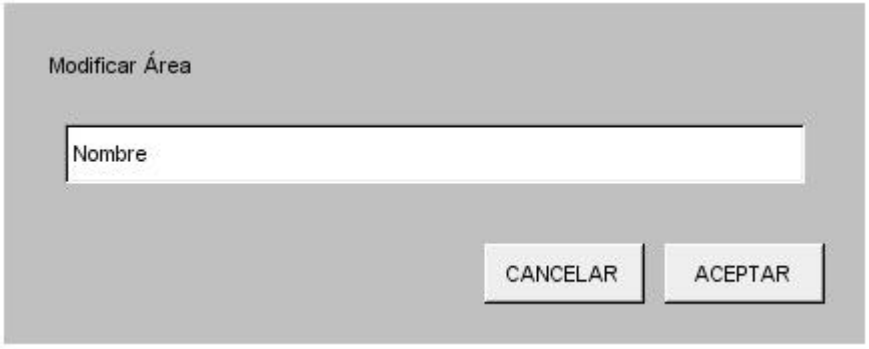
YOU, E., 2020. Introducción — Vue.js. [en línea]. [Consulta: 6 noviembre 2022]. Disponible en: https://es.vuejs.org/v2/guide/.

**ANEXOS**

## Anexo 1 Ilustraciones

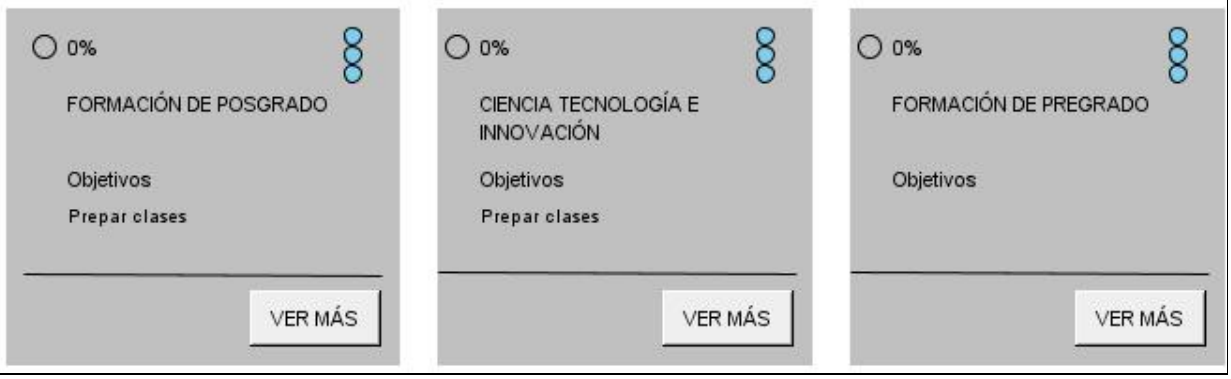

*Ilustración 12 Sistema Automatizado para la Gestión Académica*

## Anexos 2 Historias de Usuario

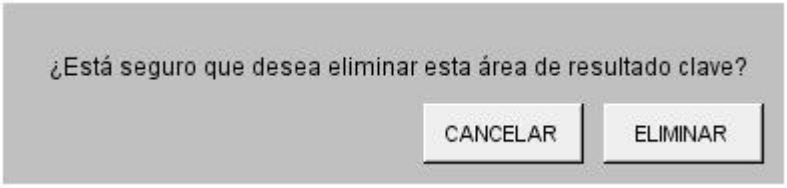
| **Historia de Usuario** | |
|---|---|
| **Número de HU:** 2 | **Nombre:** Modificar Área de resultados clave |
| **Usuarios:** decano | |
| **Puntos de estimación:** 0.125 | **Iteración asignada:** 1 |
| **Prioridad del negocio:** Alta | **Riesgos en el desarrollo:** Media |
| **Descripción:** La aplicación web debe brindar al decano una vista donde se permita modificar un Área de resultados claves en el sistema, dando la posibilidad de cambiar el nombre de dicha Área de resultados clave. | |
| **Observaciones:** Para realizar esta acción el decano debe estar autenticado | |
| **Prototipo de interfaz:**   | |

| **Historia de Usuario** | |
|---|---|
| **Número de HU:** 3 | **Nombre:** Obtener Áreas de resultados clave |
| **Usuarios:** decano | |
| **Puntos de estimación:** 0.125 | **Iteración asignada:** 1 |
| **Prioridad del negocio:** Alta | **Riesgos en el desarrollo:** Media |
| **Descripción:** La aplicación web debe brindar al decano una vista donde se permita obtener todas las Áreas de resultados claves en el sistema. | |
| **Observaciones:** Para realizar esta acción el decano debe estar autenticado | |

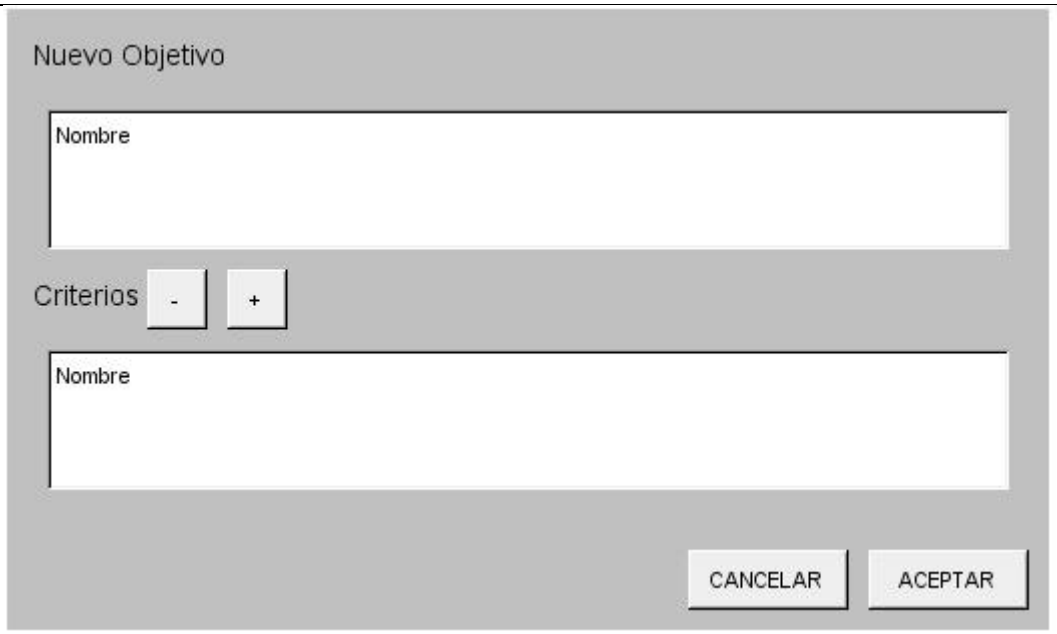
**Prototipo de interfaz:**




| **Historia de Usuario** | |
|---|---|
| **Número de HU:** 4 | **Nombre:** Eliminar Área de resultados claves |
| **Usuarios:** decano | |
| **Puntos de estimación:** 0.125 | **Iteración asignada:** 1 |
| **Prioridad del negocio:** Alta | **Riesgos en el desarrollo:** Media |
| **Descripción:** La aplicación web debe brindar al decano una vista donde se permita eliminar un Área de resultados clave en el sistema. | |
| **Observaciones:** Para realizar esta acción el decano debe estar autenticado | |
| **Prototipo de interfaz:** | |




| **Historia de Usuario** | |
|---|---|
| **Número de HU:** 5 | **Nombre:** Crear Objetivo Estratégico |
| **Usuarios:** decano | |
| **Puntos de estimación:** 0.125 | **Iteración asignada:** 1 |
| **Prioridad del negocio:** Alta | **Riesgos en el desarrollo:** Media |
| **Descripción:** La aplicación web debe brindar al decano una vista donde se permita crear un Objetivo Estratégico en el sistema, ingresando el nombre de dicho objetivo y opcionalmente los criterios de medida que le corresponden. | |
| **Observaciones:** Para realizar esta acción el decano debe estar autenticado | |
| **Prototipo de interfaz:** | |




| **Historia de Usuario** | |
|---|---|
| **Número de HU:** 6 | **Nombre:** Modificar Objetivo Estratégico |
| **Usuarios:** decano | |
| **Puntos de estimación:** 0.125 | **Iteración asignada:** 1 |
| **Prioridad del negocio:** Alta | **Riesgos en el desarrollo:** Media |
| **Descripción:** La aplicación web debe brindar al decano una vista donde se permita Modificar un Objetivo Estratégico en el sistema, dando la posibilidad de cambiar el nombre de este objetivo. | |
| **Observaciones:** Para realizar esta acción el decano debe estar autenticado | |
| **Prototipo de interfaz:**  | |



| **Historia de Usuario** | |
|---|---|
| **Número de HU:** 7 | **Nombre:** Obtener Objetivos Estratégicos |
| **Usuarios:** decano y jefe de Área | |
| **Puntos de estimación:** 0.125 | **Iteración asignada:** 1 |
| **Prioridad del negocio:** Alta | **Riesgos en el desarrollo:** Media |
| **Descripción:** La aplicación web debe brindar al decano y al jefe de Área una vista donde se permita obtener los Objetivo Estratégico en el sistema. | |

| |
|---|
| **Observaciones:** Para realizar esta acción el decano y el jefe de área deben estar autenticados |
| **Prototipo de interfaz:** |


| **Historia de Usuario** | |
|---|---|
| **Número de HU:** 8 | **Nombre:** Eliminar Objetivo Estratégico |
| **Usuarios:** decano | |
| **Puntos de estimación:** 0.125 | **Iteración asignada:** 1 |
| **Prioridad del negocio:** Alta | **Riesgos en el desarrollo:** Media |
| **Descripción:** La aplicación web debe brindar al decano una vista donde se permita eliminar un Objetivo Estratégico en el sistema. | |
| **Observaciones:** Para realizar esta acción el decano debe estar autenticado | |
| **Prototipo de interfaz:** | |

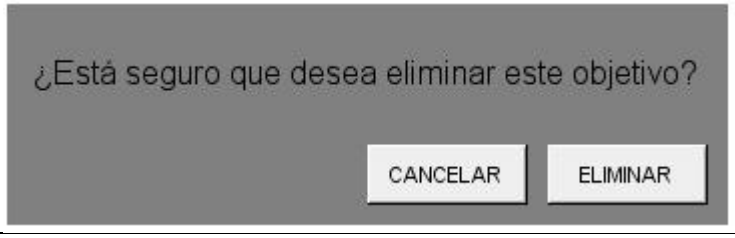



*Tabla 36 HU 9 Crear Criterio de Medida*

| **Historia de Usuario** | |
|---|---|
| **Número de HU:** 9 | **Nombre:** Crear Criterio de Medida |
| **Usuarios:** decano y jefe de área | |
| **Puntos de estimación:** 0.5 | **Iteración asignada:** 1 |
| **Prioridad del negocio:** Alta | **Riesgos en el desarrollo:** Media |
| **Descripción:** La aplicación web debe brindar al decano(a) y al jefe de Área una vista donde se permita crear un Criterio de Medida en el sistema, ingresando el nombre de dicho criterio y la cantidad de cumplimientos que este requiere para ser cumplido. | |
| **Observaciones:** Para realizar esta acción el decano y el jefe de área deben estar autenticados | |
| **Prototipo de interfaz:** | |

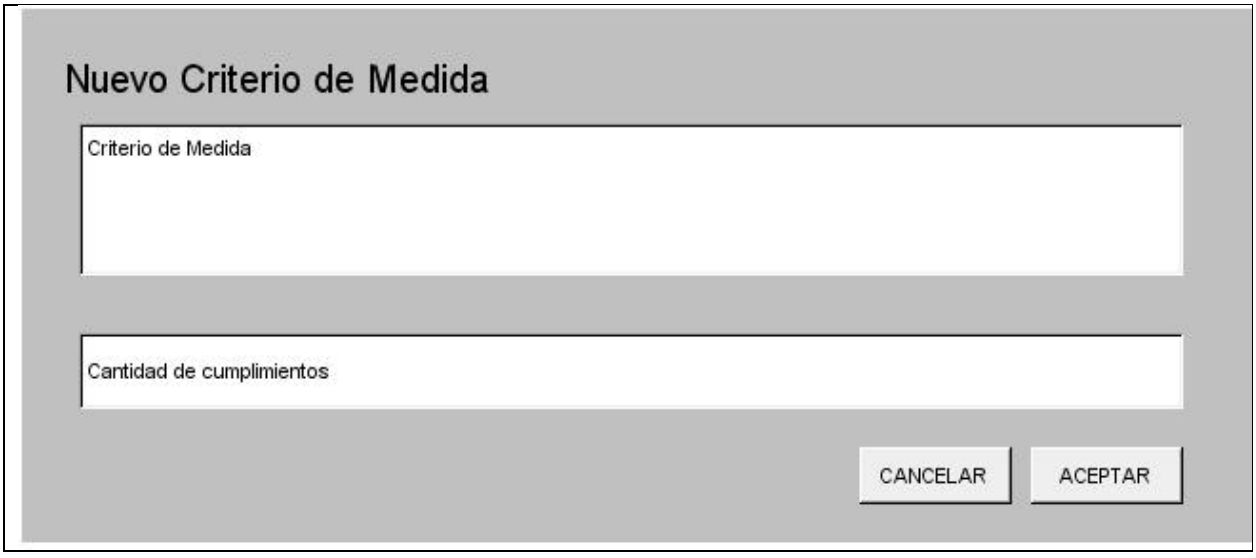



*Tabla 37 HU 10 Modificar Criterio de Medida*

| **Historia de Usuario** | |
|---|---|
| **Número de HU:** 10 | **Nombre:** Modificar Criterio de Medida |
| **Usuarios:** decano y jefe de Área | |
| **Puntos de estimación:** 0.5 | **Iteración asignada:** 1 |
| **Prioridad del negocio:** Alta | **Riesgos en el desarrollo:** Media |
| **Descripción:** La aplicación web debe brindar al decano y al jefe de área una vista donde se permita modificar un Criterio de Medida en el sistema, dando la posibilidad de cambiar el nombre de este criterio y la cantidad de cumplimientos. | |
| **Observaciones:** Para realizar esta acción el decano y el jefe de área deben estar autenticados | |
| **Prototipo de interfaz:** | |

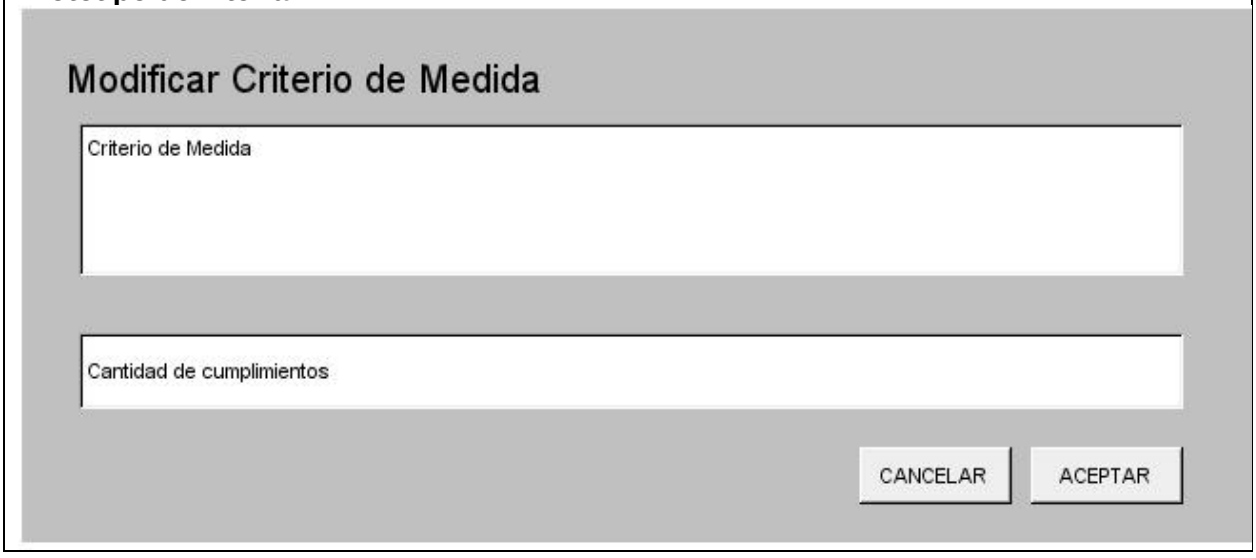



*Tabla 38 HU 11 Obtener Criterios de Medidas*

| **Historia de Usuario** | |
|---|---|
| **Número de HU:** 11 | **Nombre:** Obtener Criterios de Medidas |
| **Usuarios:** decano y jefe de Área | |

| | |
|---|---|
| **Puntos de estimación:** 0.5 | **Iteración asignada:** 1 |
| **Prioridad del negocio:** Alta | **Riesgos en el desarrollo:** Media |
| **Descripción:** La aplicación web debe brindar al decano y al jefe de área una vista donde se permita obtener los Criterios de Medidas en el sistema. | |
| **Observaciones:** Para realizar esta acción el decano y el jefe de área deben estar autenticados | |
| **Prototipo de interfaz:** | |


*Tabla 39 HU 12 Eliminar Criterio de Medida*

| Historia de Usuario | |
|---|---|
| **Número de HU:** 12 | **Nombre:** Eliminar Criterio de Medida |
| **Usuarios:** decano y jefe de área | |
| **Puntos de estimación:** 0.5 | **Iteración asignada:** 1 |
| **Prioridad del negocio:** Alta | **Riesgos en el desarrollo:** Media |
| **Descripción:** La aplicación web debe brindar al decano y al jefe de área una vista donde se permita eliminar un Criterio de Medida en el sistema. | |
| **Observaciones:** Para realizar esta acción el decano y el jefe de área deben estar autenticados | |
| **Prototipo de interfaz:** | |

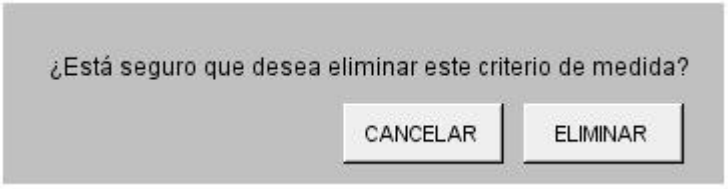



*Tabla 40 HU 14 Modificar Indicador*

| Historia de Usuario | |
|---|---|
| **Número de HU:** 14 | **Nombre:** Modificar Indicador |
| **Usuarios:** decano y jefe de área | |
| **Puntos de estimación:** 0.5 | **Iteración asignada:** 1 |
| **Prioridad del negocio:** Alta | **Riesgos en el desarrollo:** Media |

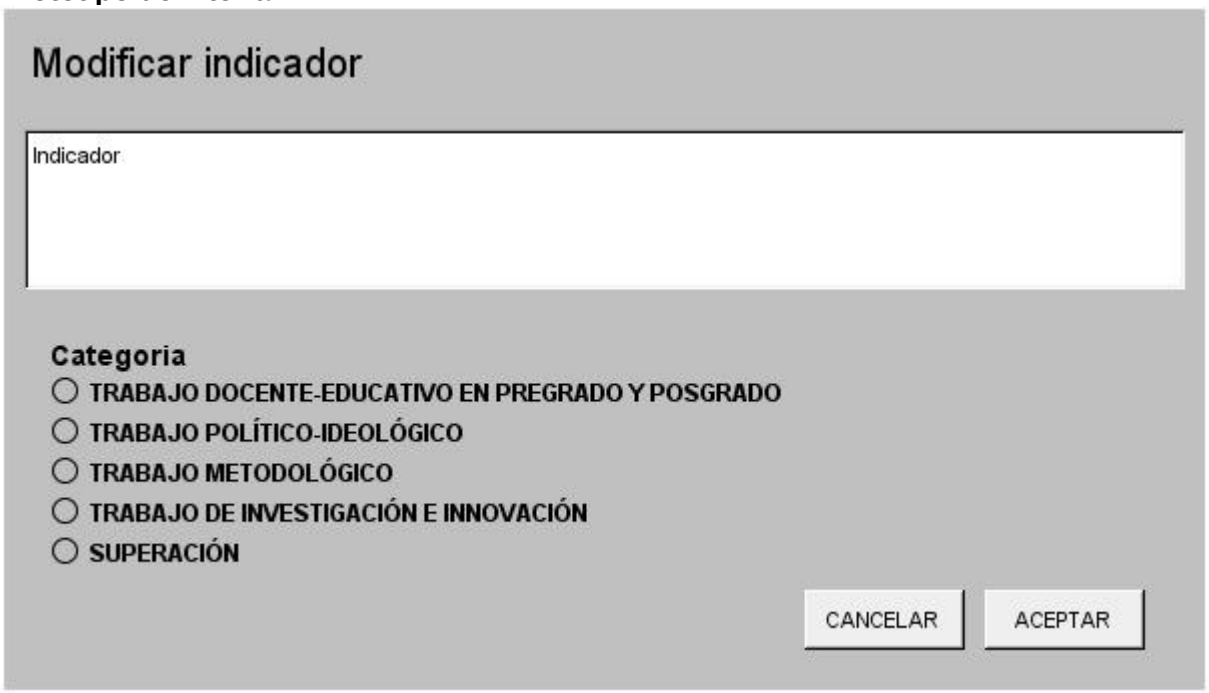
| |
|---|
| **Descripción:** La aplicación web debe brindar al decano y al jefe de área una vista donde se permita modificar un Indicador en el sistema, dando la posibilidad de cambiar el nombre de este indicador y la categoría a la que pertenece. |
| **Observaciones:** Para realizar esta acción el decano y el jefe de área deben estar autenticados |
| **Prototipo de interfaz:**   |

*Tabla 41 HU 15 Obtener Indicadores*

| **Historia de Usuario** | |
|---|---|
| **Número de HU:** 15 | **Nombre:** Obtener Indicadores |
| **Usuarios:** decano y jefe de Área | |
| **Puntos de estimación:** 0.5 | **Iteración asignada:** 1 |
| **Prioridad del negocio:** Alta | **Riesgos en el desarrollo:** Media |
| **Descripción:** La aplicación web debe brindar al decano y al jefe de área una vista donde se permita obtener los Indicadores en el sistema. | |
| **Observaciones:** Para realizar esta acción el decano y el jefe de área deben estar autenticados | |
| **Prototipo de interfaz:** | |


*Tabla 42 HU 16 Eliminar Indicador*

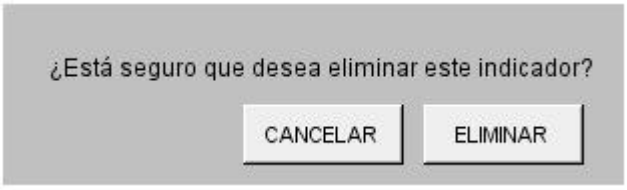
| **Historia de Usuario** | |
|---|---|
| **Número de HU:** 16 | **Nombre:** Eliminar Indicador |
| **Usuarios:** decano y jefe de área | |
| **Puntos de estimación:** 0.5 | **Iteración asignada:** 1 |
| **Prioridad del negocio:** Alta | **Riesgos en el desarrollo:** Media |
| **Descripción:** La aplicación web debe brindar al decano y al jefe de área una vista donde se permita eliminar un Indicador en el sistema. | |
| **Observaciones:** Para realizar esta acción el decano y el jefe de área deben estar autenticados | |
| **Prototipo de interfaz:**   | |

*Tabla 43 HU 17 Crear Usuario*

| **Historia de Usuario** | |
|---|---|
| **Número de HU:** 17 | **Nombre:** Crear Usuario |
| **Usuarios:** decano | |
| **Puntos de estimación:** 0.5 | **Iteración asignada:** 1 |
| **Prioridad del negocio:** Alta | **Riesgos en el desarrollo:** Media |
| **Descripción:** La aplicación web debe brindar al decano una vista donde se permita crear un usuario en el sistema, ingresando el nombre, nombre de usuario, solapín, facultad, categoría, departamento y el rol de dicho usuario. | |
| **Observaciones:** Para realizar esta acción el decano debe estar autenticado | |
| **Prototipo de interfaz:** | |


*Tabla 44 HU 18 Modificar Usuario*

| **Historia de Usuario** | |
|---|---|
| **Número de HU:** 18 | **Nombre:** Modificar Usuario |
| **Usuarios:** decano | |
| **Puntos de estimación:** 0.5 | **Iteración asignada:** 1 |
| **Prioridad del negocio:** Alta | **Riesgos en el desarrollo:** Media |
| **Descripción:** La aplicación web debe brindar al decano una vista donde se permita modificar un usuario en el sistema, dando la posibilidad de cambiar el nombre, nombre de usuario, solapín, facultad, categoría, departamento y el rol de dicho usuario. | |
| **Observaciones:** Para realizar esta acción el decano debe estar autenticado | |
| **Prototipo de interfaz:** | |


*Tabla 45 HU 19 Obtener Usuarios*

| **Historia de Usuario** | |
|---|---|
| **Número de HU:** 19 | **Nombre:** Obtener Usuarios |
| **Usuarios:** decano y jefe de departamento | |
| **Puntos de estimación:** 0.5 | **Iteración asignada:** 1 |
| **Prioridad del negocio:** Alta | **Riesgos en el desarrollo:** Media |
| **Descripción:** La aplicación web debe brindar al decano y al jefe de departamento una vista donde se permita obtener los usuarios en el sistema. | |
| **Observaciones:** Para realizar esta acción el decano debe estar autenticado | |
| **Prototipo de interfaz:** | |

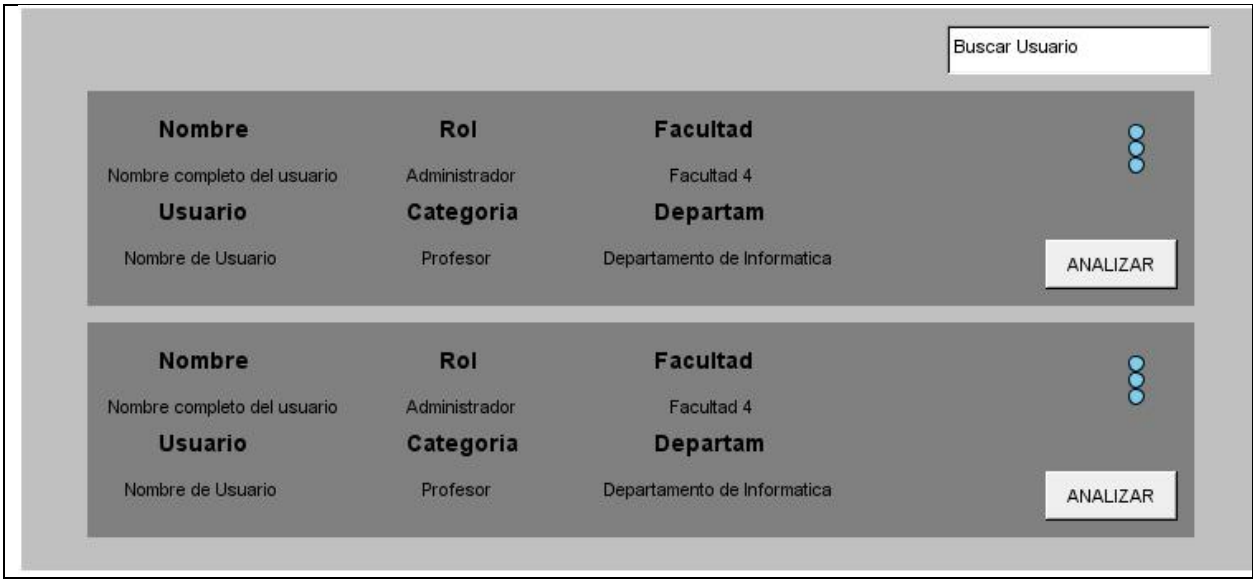



*Tabla 46 HU 20 Eliminar Usuario*

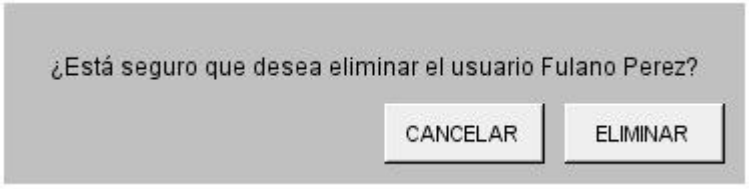
| **Historia de Usuario** | |
|---|---|
| **Número de HU:** 20 | **Nombre:** Eliminar Usuario |
| **Usuarios:** decano | |
| **Puntos de estimación:** 0.5 | **Iteración asignada:** 1 |
| **Prioridad del negocio:** Alta | **Riesgos en el desarrollo:** Media |
| **Descripción:** La aplicación web debe brindar al decano una vista donde se permita eliminar un usuario en el sistema. | |
| **Observaciones:** Para realizar esta acción el decano debe estar autenticado | |
| **Prototipo de interfaz:**  | |



*Tabla 47 HU 22 Modificar Evidencia*

| **Historia de Usuario** | |
|---|---|
| **Número de HU:** 22 | **Nombre:** Modificar Evidencia |
| **Usuarios:** usuario autenticado, jefe de área y jefe de departamento | |
| **Puntos de estimación:** 0.5 | **Iteración asignada:** 1 |
| **Prioridad del negocio:** Alta | **Riesgos en el desarrollo:** Media |
| **Descripción:** La aplicación web debe brindar al usuario autenticado, jefe de área y jefe de departamento una vista donde se permita modificar una evidencia en el sistema, dando la posibilidad de cambiar la descripción, la fecha y el archivo. | |
| **Observaciones:** Para realizar esta acción el usuario autenticado, jefe de área y jefe de departamento deben estar autenticados | |
| **Prototipo de interfaz:** | |

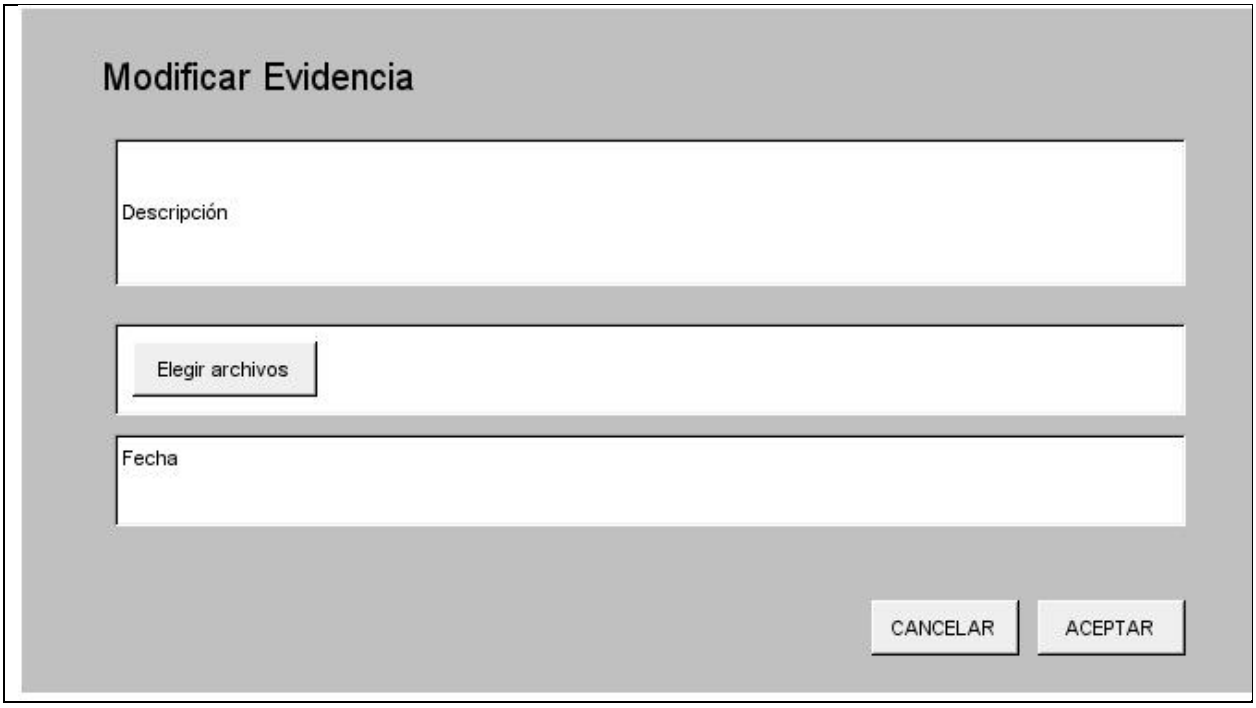



*Tabla 48 HU 23 Obtener Evidencias*

| **Historia de Usuario** | |
|---|---|
| **Número de HU:** 23 | **Nombre:** Obtener Evidencias |
| **Usuarios:** usuario autenticado, jefe de área y jefe de departamento | |
| **Puntos de estimación:** 0.5 | **Iteración asignada:** 1 |
| **Prioridad del negocio:** Alta | **Riesgos en el desarrollo:** Media |
| **Descripción:** La aplicación web debe brindar al usuario autenticado, jefe de área y jefe de departamento una vista donde se permita obtener las evidencias en el sistema. | |
| **Observaciones:** Para realizar esta acción el usuario autenticado, jefe de área y jefe de departamento área deben estar autenticados | |
| **Prototipo de interfaz:** | |

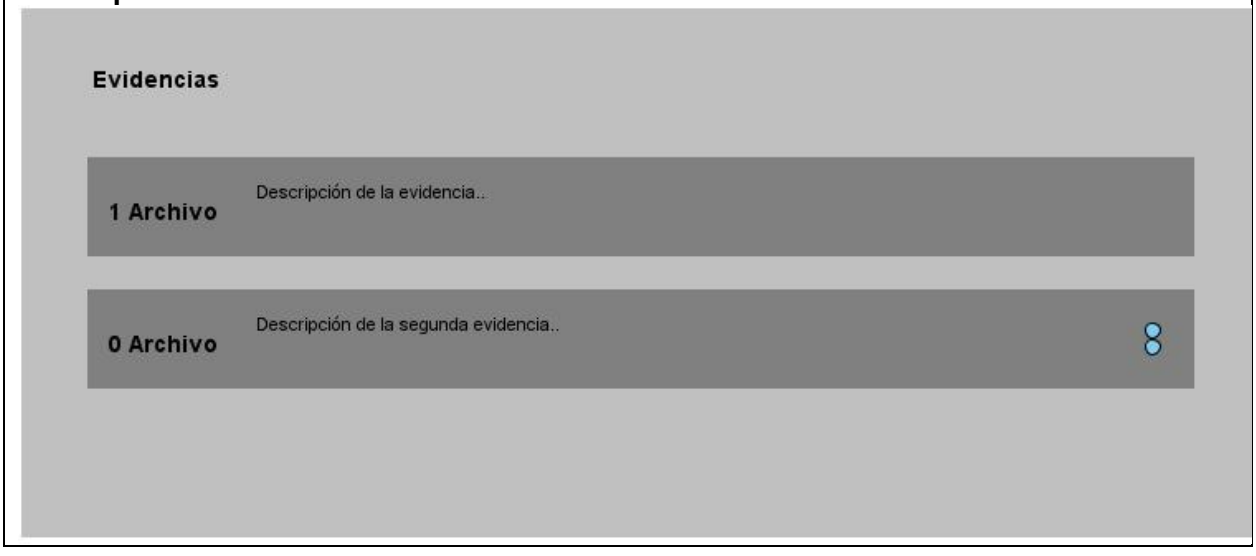


*Tabla 49 HU 24 Eliminar Evidencia*

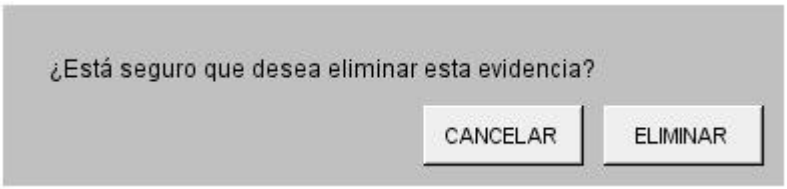
| Historia de Usuario | |
|---|---|
| **Número de HU:** 24 | **Nombre:** Eliminar Evidencia |
| **Usuarios:** usuario autenticado, jefe de área y jefe de departamento | |
| **Puntos de estimación:** 0.5 | **Iteración asignada:** 1 |
| **Prioridad del negocio:** Alta | **Riesgos en el desarrollo:** Media |
| **Descripción:** La aplicación web debe brindar al usuario autenticado, jefe de área y jefe de departamento una vista donde se permita eliminar una evidencia en el sistema. | |
| **Observaciones:** Para realizar esta acción el usuario autenticado, jefe de área y jefe de departamento área debe estar autenticados | |
| **Prototipo de interfaz:**   | |

*Tabla 50 HU 25 Autenticar Usuario*

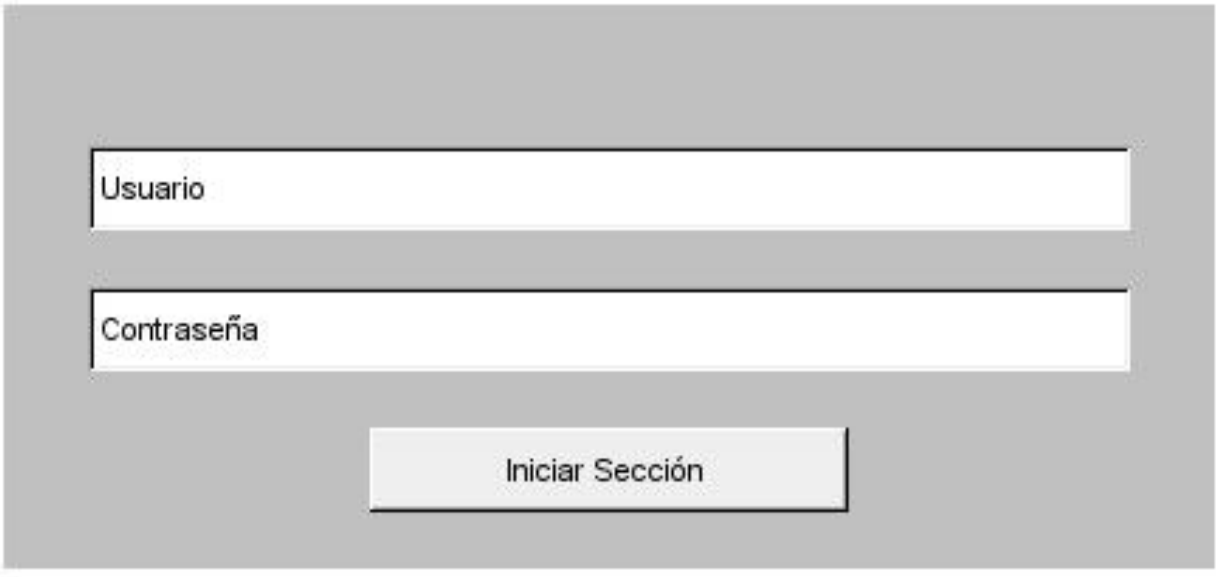
| Historia de Usuario | |
|---|---|
| **Número de HU:** 25 | **Nombre:** Autenticar Usuario |
| **Usuarios:** Usuario autenticado | |
| **Puntos de estimación:** 0.2 | **Iteración asignada:** 2 |
| **Prioridad del negocio:** Media | **Riesgos en el desarrollo:** media |
| **Descripción:** Permite a todos los profesores y especiaistas de la facultad que introduzcan su cuenta de usuario y contraseña dentro de la aplicación. | |
| **Observaciones:** Todos los usuarios deben acceder al apartado de la autenticación | |
| **Prototipo de interfaz:**   | |

*Tabla 51 HU 26 Evaluar a un usuario*

| Historia de Usuario |
|---|

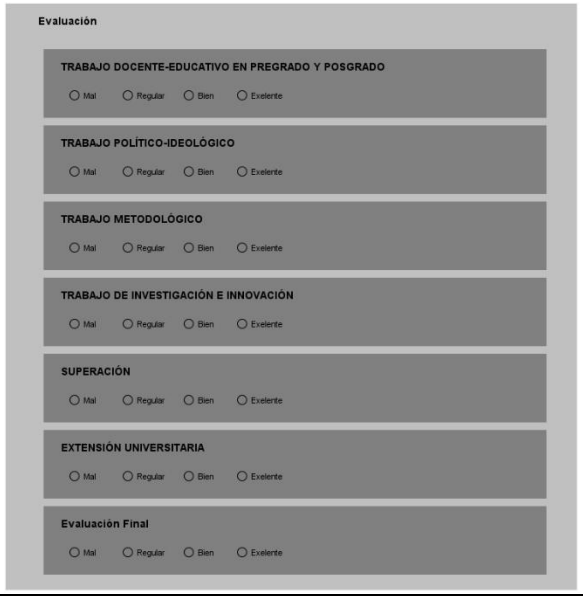
| | |
|---|---|
| **Número de HU:** 26 | **Nombre:** Evaluar a un usuario |
| **Usuarios:** jefe de departamento | |
| **Puntos de estimación:** 0.2 | **Iteración asignada:** 2 |
| **Prioridad del negocio:** Media | **Riesgos en el desarrollo:** media |
| **Descripción:** Permite a los jefes de departamento evaluar a sus trabajadores mediante los indicadores cumplidos | |
| **Prototipo de interfaz:**   | |

*Tabla 52 HU 27 Exportar evaluación a PDF*

| Historia de Usuario | |
|---|---|
| **Número de HU:** 27 | **Nombre:** Exportar evaluación a PDF |
| **Usuarios:** jefe de departaemto | |
| **Puntos de estimación:** 0.8 | **Iteración asignada:** 2 |
| **Prioridad del negocio:** Media | **Riesgos en el desarrollo:** media |
| **Descripción:** Permite a el jefe de departamento ya una vez evaluado el usuario, exportar dicha evaluación en PDF | |
| **Prototipo de interfaz:** | |

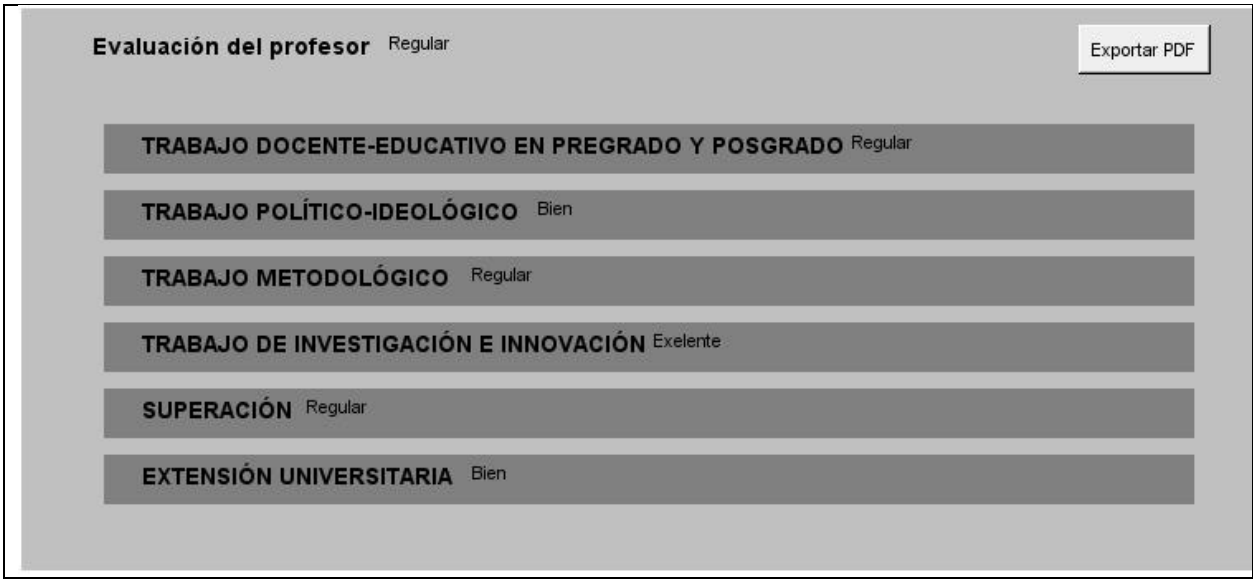



*Tabla 53 HU 28 Cambio de estado del Criterio*

| Historia de Usuario | |
|---|---|
| **Número de HU:** 28 | **Nombre:** Cambiar el estado del Criterio |
| **Usuarios:** | |
| **Puntos de estimación:** 0.5 | **Iteración asignada:** 2 |
| **Prioridad del negocio:** Alta | **Riesgos en el desarrollo:** Alta |
| **Descripción:** El sistema de manera automática cambia el estado del Criterio de Media a: No cumplido, cumplido y sobre cumplido, según se cumplan o borren los indicadores asociados al criterio | |
| **Prototipo de interfaz:** | |

*Tabla 54 HU 29 Adjuntar archivo a la evidencia*

| Historia de Usuario | |
|---|---|
| **Número de HU:** 29 | **Nombre:** Adjuntar archivo a la evidencia |
| **Usuarios:** usuario autenticado, jefe de área y jefe de departamento | |
| **Puntos de estimación:** 0.5 | **Iteración asignada:** 2 |
| **Prioridad del negocio:** Alta | **Riesgos en el desarrollo:** Alta |
| **Descripción:** El usuario autenticado, jefe de área y jefe de departamento pueden adjuntar un archivo a la evidencia, así como también una descripción de la misma | |
| **Prototipo de interfaz:** | |

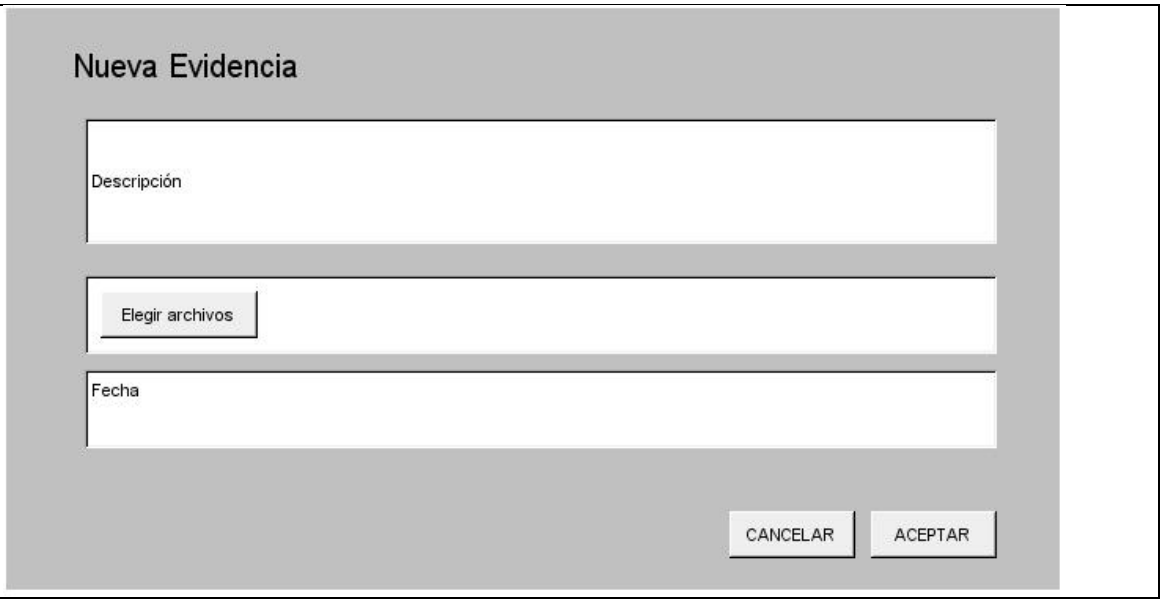



*Tabla 55 HU 30 Cambiar la contraseña*

| Historia de Usuario | |
|---|---|
| **Número de HU:** 30 | **Nombre:** Cambiar la contraseña |
| **Usuarios:** usuario autenticado, jefe de área, jefe de departamento y decano | |
| **Puntos de estimación:** 0.2 | **Iteración asignada:** 2 |
| **Prioridad del negocio:** Media | **Riesgos en el desarrollo:** Alta |
| **Descripción:** El usuario autenticado, jefe de área, jefe de departamento y decano pueden cambiar su contraseña, colocando su contraseña antigua más la nueva y una confirmación | |
| **Prototipo de interfaz:** | |

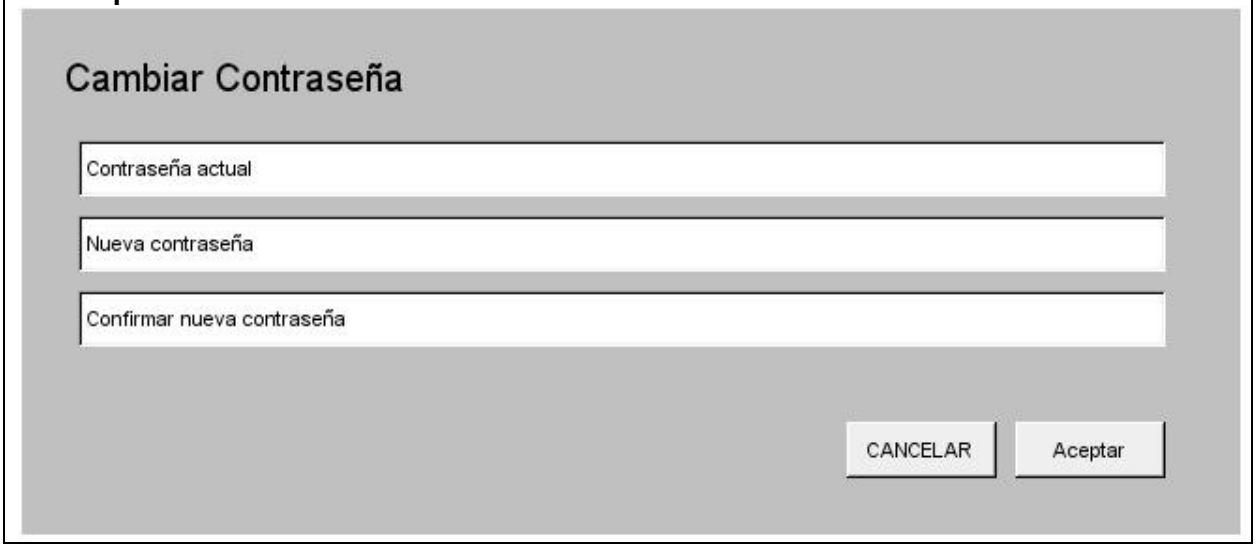



*Tabla 56 HU 31 Asignar indicadores*

| Historia de Usuario | |
|---|---|
| **Número de HU:** 31 | **Nombre:** Asignar indicadores |
| **Usuarios:** jefe de departamento | |
| **Puntos de estimación:** 0.5 | **Iteración asignada:** 2 |

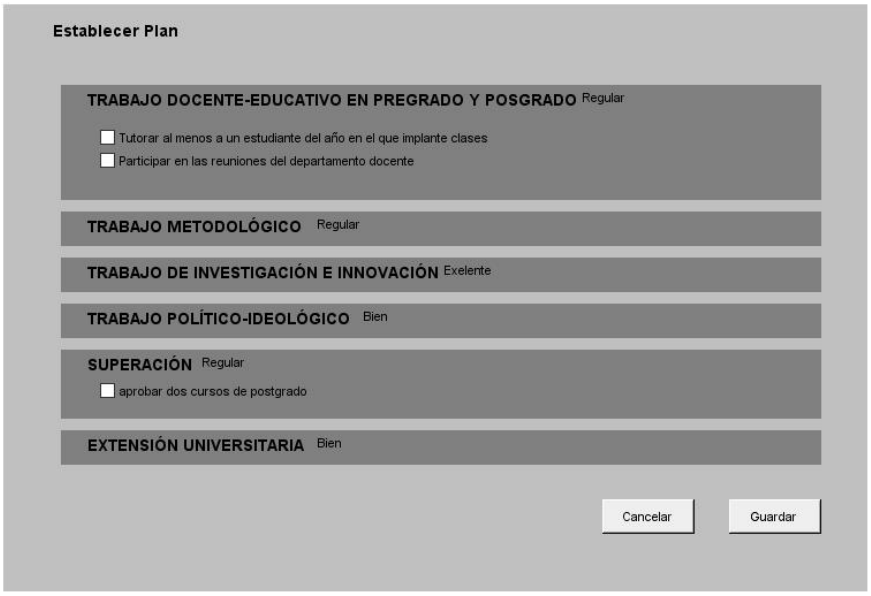
| | |
|---|---|
| **Prioridad del negocio:** Alta | **Riesgos en el desarrollo:** Alta |
| **Descripción:** El jefe de departamento le asigna los indicadores a cumplir a sus trabajadores, que ya antes fueron establecidos en el plan general | |
| **Prototipo de interfaz:**   | |

*Tabla 57 HU 32 Crear indicador personal*

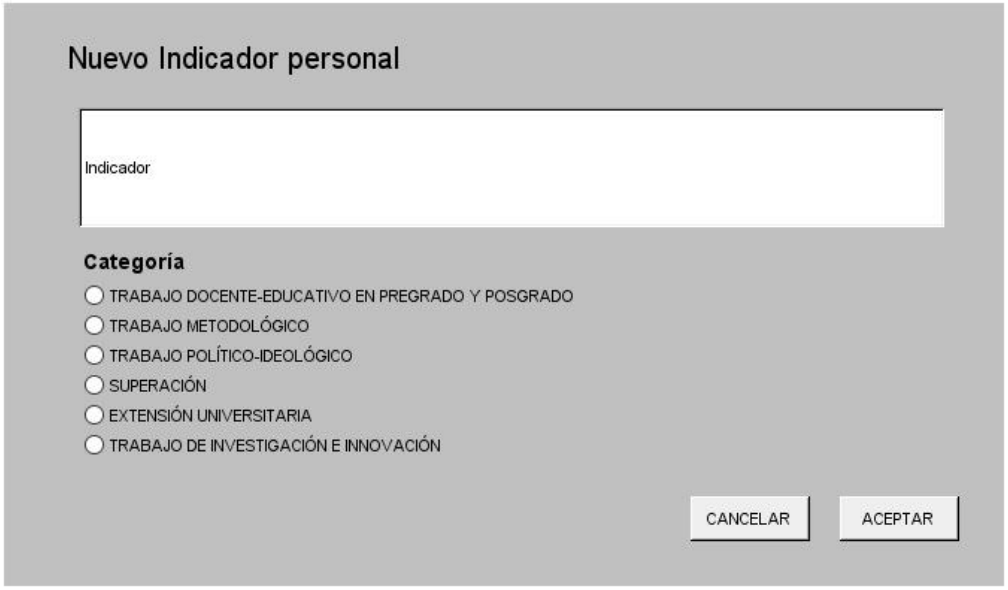
| Historia de Usuario | |
|---|---|
| **Número de HU:** 32 | **Nombre:** Crear indicador personal |
| **Usuarios:** usuario autenticado, jefe de área y jefe de departamento | |
| **Puntos de estimación:** 0.5 | **Iteración asignada:** 2 |
| **Prioridad del negocio:** Media | **Riesgos en el desarrollo:** Medio |
| **Descripción:** El usuario autenticado, jefe de área y jefe de departamento puede crear su propio indicador personal, que después dicho indicador tributa a su evaluación | |
| **Prototipo de interfaz:**   | |

*Tabla 58 HU 33 Cambiar estado del indicador*

| Historia de Usuario | |
|---|---|
| **Número de HU:** 33 | **Nombre:** Cambiar estado del indicador |
| **Usuarios:** | |
| **Puntos de estimación:** 0.2 | **Iteración asignada:** 2 |
| **Prioridad del negocio:** Alta | **Riesgos en el desarrollo:** Alta |
| **Descripción:** El sistema de manera automática cambia el estado de cumplido del indicador a verdadero o falso en caso de que se agregue su primera evidencia o se eliminen todas sus evidencias | |
| **Prototipo de interfaz:** | |

*Tabla 59 HU 34 Restablecer contraseña*

| Historia de Usuario | |
|---|---|
| **Número de HU:** 34 | **Nombre:** Restablecer contraseña |
| **Usuarios:** decano | |
| **Puntos de estimación:** 0.1 | **Iteración asignada:** 3 |
| **Prioridad del negocio:** Baja | **Riesgos en el desarrollo:** Baja |
| **Descripción:** El decano tiene la posibilidad de restablecer la contraseña a un usuario en caso de que este la haya olvidado | |
| **Prototipo de interfaz:**   | |

*Tabla 60 HU 35 Buscar usuario*

| Historia de Usuario | |
|---|---|
| **Número de HU:** 35 | **Nombre:** Buscar usuario |
| **Usuarios:** Decano y jefe de departamento | |
| **Puntos de estimación:** 0.1 | **Iteración asignada:** 3 |
| **Prioridad del negocio:** Baja | **Riesgos en el desarrollo:** Baja |
| **Descripción:** El decano y jefe de departamento tiene la posibilidad de buscar en la lista de usuarios a un usuario por su nombre o nombre de usuario | |
| **Prototipo de interfaz:** | |


*Tabla 61 HU 36 Mostrar porcentaje*

| Historia de Usuario | |
|---|---|
| **Número de HU:** 36 | **Nombre:** Mostrar porcentaje |
| **Usuarios:** | |
| **Puntos de estimación:** 0.2 | **Iteración asignada:** 3 |
| **Prioridad del negocio:** Baja | **Riesgos en el desarrollo:** Baja |
| **Descripción:** El sistema muestra por cada área de resultados claves el nivel de cumplimiento del área respectivamente en porciento. | |
| **Prototipo de interfaz:** | |

*Tabla 62 HU 37 Notificar al usuario*

| Historia de Usuario | |
|---|---|
| **Número de HU:** 37 | **Nombre:** Notificar al usuario |
| **Usuarios:** Usuario autenticado | |
| **Puntos de estimación:** 0.5 | **Iteración asignada:** 3 |
| **Prioridad del negocio:** Baja | **Riesgos en el desarrollo:** Baja |
| **Descripción:** Cuando se le asigna o elimina un indicador al usuario se lo notifica mediante un mensaje en el propio sistema al usuario en cuestión | |
| **Prototipo de interfaz:** | |

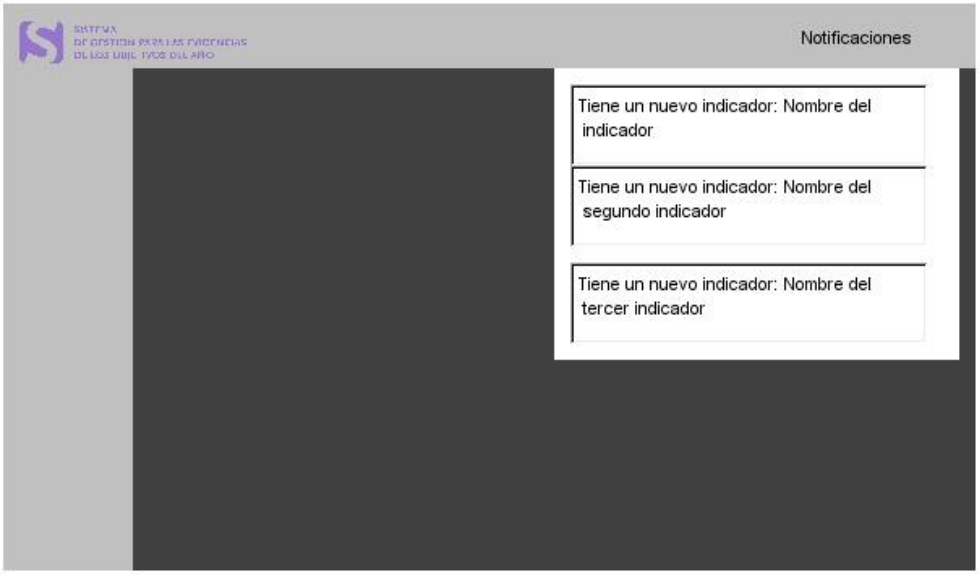



*Tabla 63 HU 38 Notificar al jefe de Área*

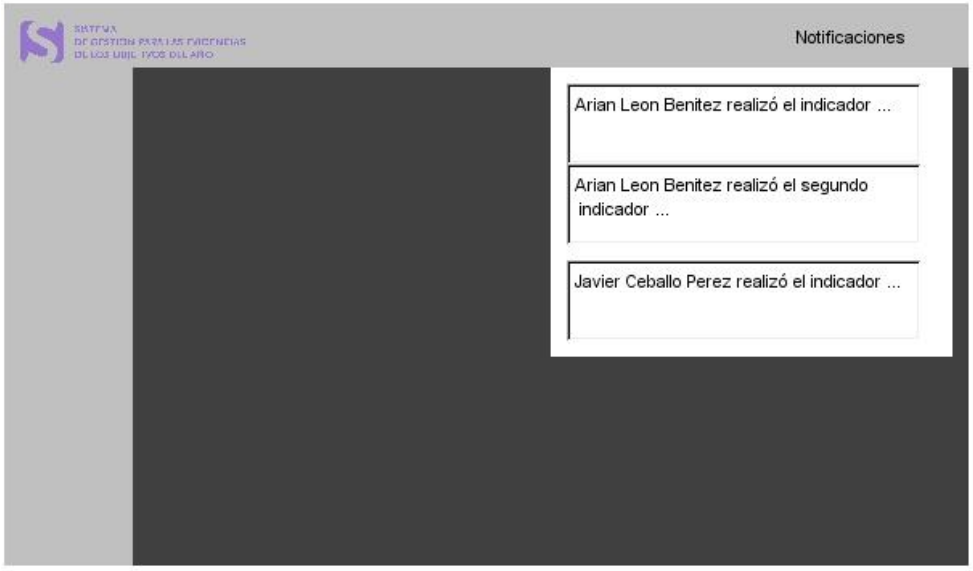
| Historia de Usuario | |
|---|---|
| **Número de HU:** 38 | **Nombre:** Notificar al jefe de departamneto |
| **Usuarios:** jefe de departamneto | |
| **Puntos de estimación:** 0.5 | **Iteración asignada:** 3 |
| **Prioridad del negocio:** Baja | **Riesgos en el desarrollo:** Baja |
| **Descripción:** Cuando un usuario cumple un indicador que le fue asignado o crea un indicador personal, al jefe de departamento al que pertenece se le notifica mediante un mensaje en el propio sistema | |
| **Prototipo de interfaz:**   | |

*Tabla 64 HU 39 Informe del Área*

| Historia de Usuario |
|---|

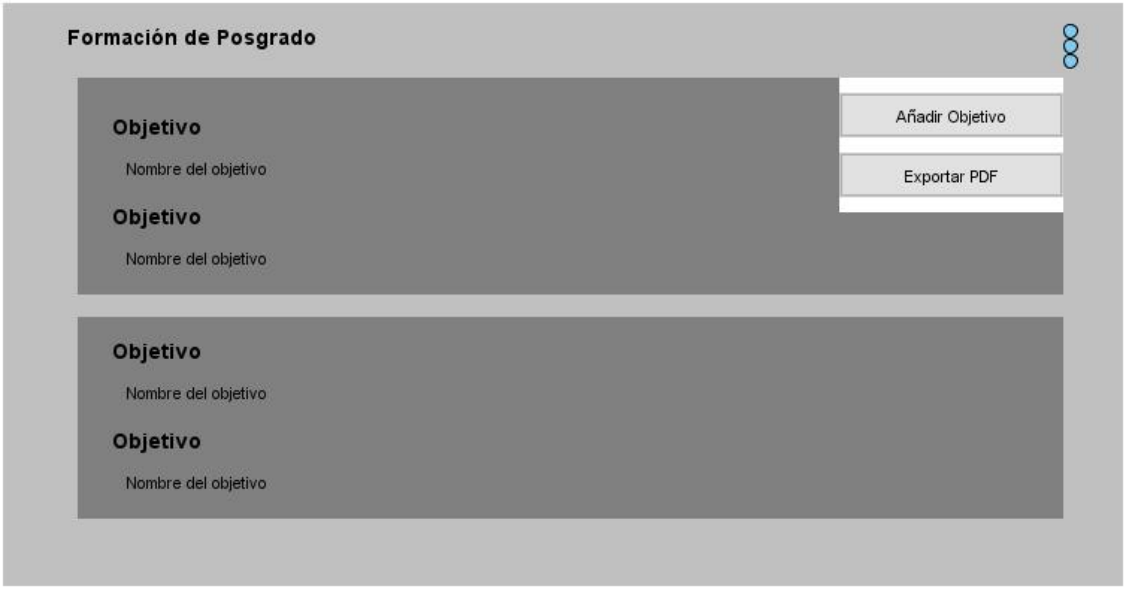
| | |
|---|---|
| **Número de HU:** 39 | **Nombre:** Informe del Área de resultados claves |
| **Usuarios:** jefe de área | |
| **Puntos de estimación:** 0.2 | **Iteración asignada:** 3 |
| **Prioridad del negocio:** Media | **Riesgos en el desarrollo:** Medio |
| **Descripción:** El jefe de área puede generar un informe detallado del cumplimiento de los objetivos estratégicos del área en formato PDF | |
| **Prototipo de interfaz:**   | |

*Tabla 65 HU 40 Denegar indicador*

| Historia de Usuario | |
|---|---|
| **Número de HU:** 40 | **Nombre:** Denegar indicador |
| **Usuarios:** jefe de departamento | |
| **Puntos de estimación:** 0.2 | **Iteración asignada:** 3 |
| **Prioridad del negocio:** Media | **Riesgos en el desarrollo:** Medio |
| **Descripción:** El jefe de departamento puede denegar un indicador que un usuario haya cumplido, cambiando el estado de cumplimiento del indicador a falso | |
| **Prototipo de interfaz:** | |

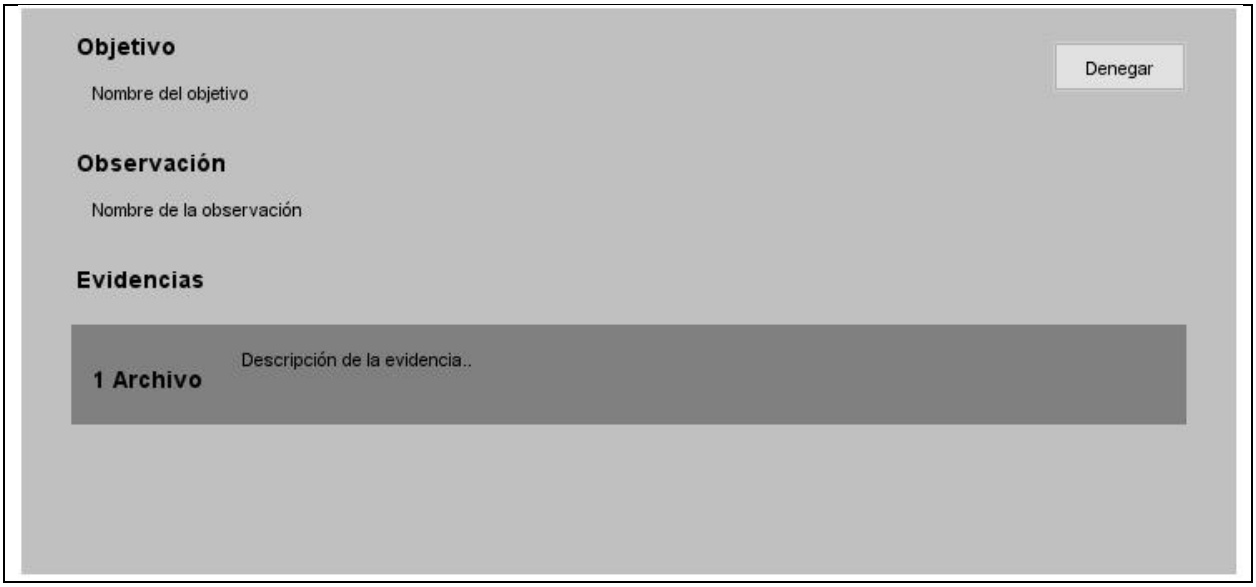



*Tabla 66 HU 41 Añadir observación*

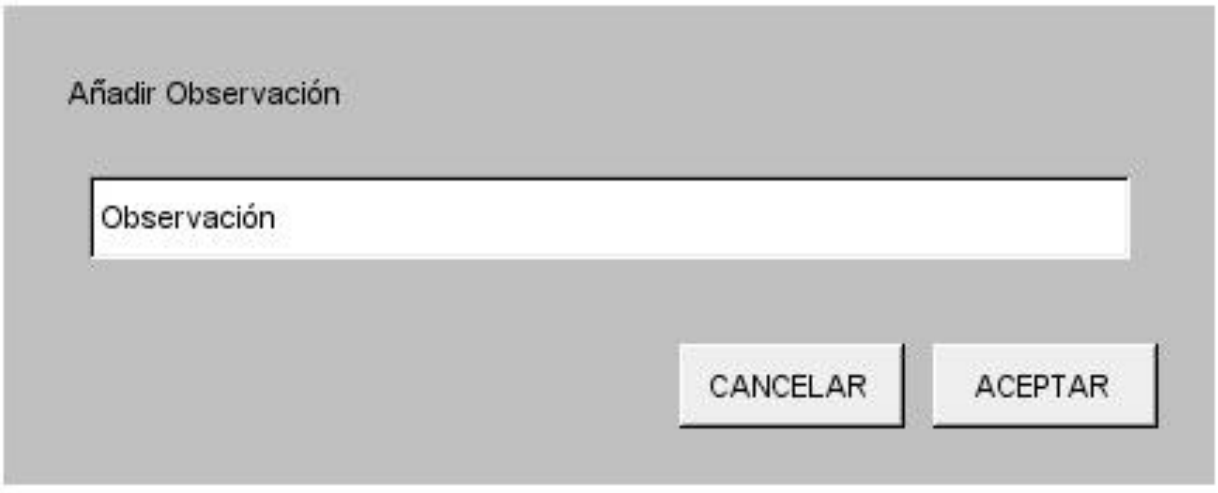
| Historia de Usuario | |
|---|---|
| **Número de HU:** 41 | **Nombre:** Añadir observación |
| **Usuarios:** usuario autenticado, jefe de área y jefe de departamento | |
| **Puntos de estimación:** 0.2 | **Iteración asignada:** 3 |
| **Prioridad del negocio:** Baja | **Riesgos en el desarrollo:** Baja |
| **Descripción:** El usuario autenticado, jefe de área y jefe de departamento tiene la posibilidad de añadir a un indicador que tenga asignado o haya sido creado por el mismo una observación en forma de texto que luego saldrá reflejado en su evaluación | |
| **Prototipo de interfaz:**  | |



*Tabla 67 HU 42 Obtener Áreas por año*

| **Historia de Usuario** | |
|---|---|
| **Número de HU:** 42 | **Nombre:** Obtener Áreas de resultados claves por año |
| **Usuarios:** Decano y jefe de área | |

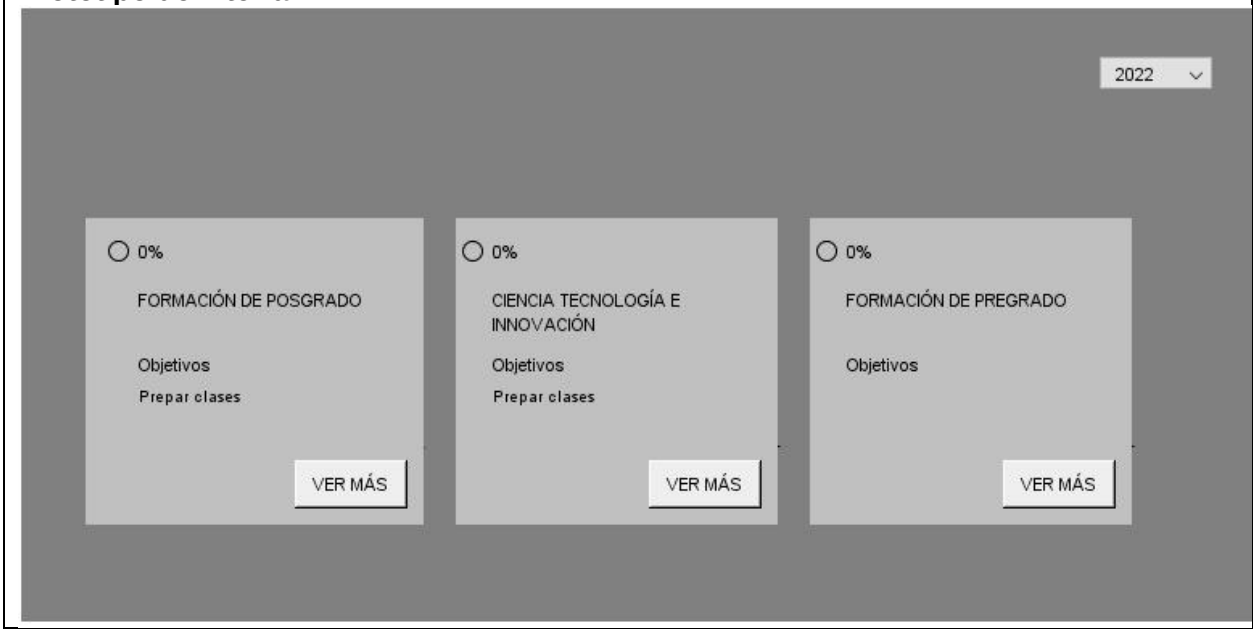
| | |
|---|---|
| **Puntos de estimación:** 0.1 | **Iteración asignada:** 3 |
| **Prioridad del negocio:** Alta | **Riesgos en el desarrollo:** Alto |
| **Descripción:** La aplicación web debe brindar al decano o jefe de área una vista donde se permita obtener todas las áreas de resultados claves de un año determinado, seleccionando dicho año. | |
| **Observaciones:** Para realizar esta acción el decano o jefe de área debe estar autenticado. | |
| **Prototipo de interfaz:**   | |

*Tabla 68 HU 43 Obtener Indicadores de un usuario por año*

| Historia de Usuario | |
|---|---|
| **Número de HU:** 43 | **Nombre:** Obtener Indicadores de un usuario por año |
| **Usuarios:** jefe de departamento y usuario autenticado | |
| **Puntos de estimación:** 0.1 | **Iteración asignada:** 3 |
| **Prioridad del negocio:** Alta | **Riesgos en el desarrollo:** Alto |
| **Descripción:** La aplicación web debe brindar al jefe de departamento o usuario autenticado una vista donde se permita obtener todos los indicadores asociados a un usuario en un año determinado. | |
| **Observaciones:** Para realizar esta acción el jefe de departamento o usuario autenticado debe estar autenticado. | |

*Tabla 69 HU 44 Establecer fecha a una evidencia*

| Historia de Usuario | |
|---|---|
| **Número de HU:** 44 | **Nombre:** Establecer fecha a una evidencia |
| **Usuarios:** jefe de área, jefe de departamento, usuario autenticado | |
| **Puntos de estimación:** 0.1 | **Iteración asignada:** 3 |
| **Prioridad del negocio:** Medio | **Riesgos en el desarrollo:** Medio |
| **Descripción:** La aplicación web debe brindar al decano una vista donde se permita establecer una fecha a una evidencia en el sistema. | |

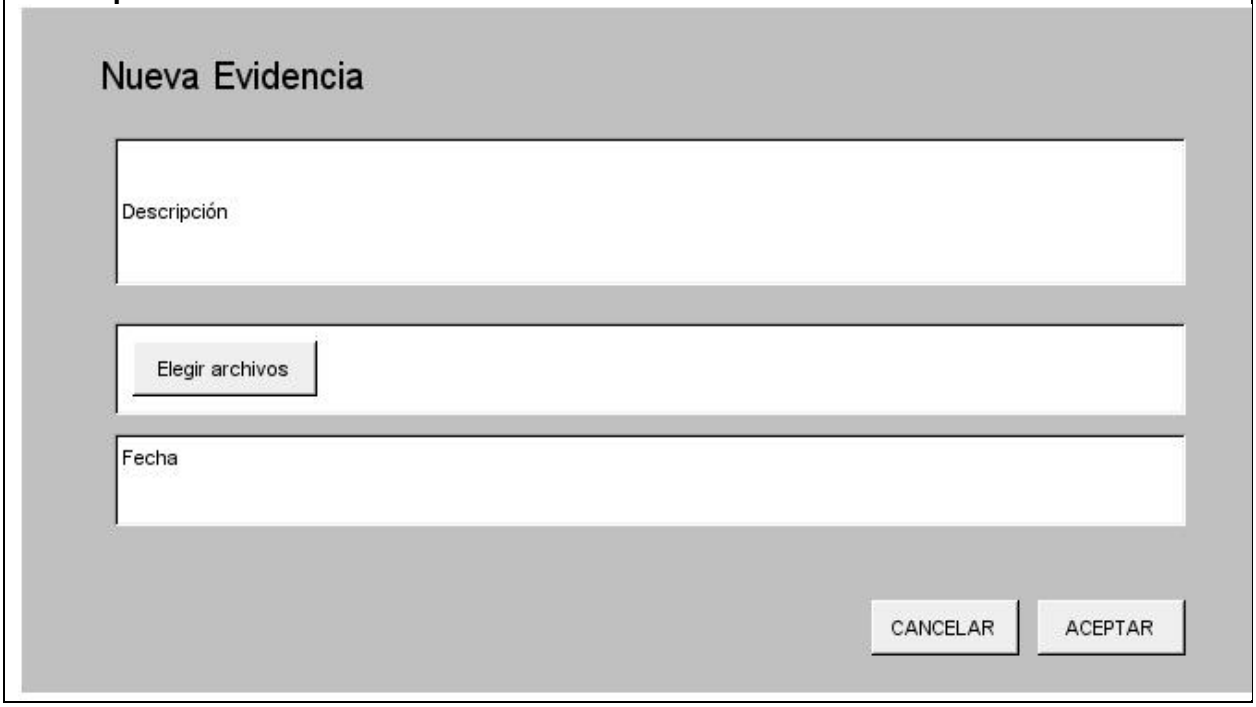
| |
|---|
| **Observaciones:** Para realizar esta acción el jefe de área, jefe de departamento y el usuario autenticado deben estar autenticados. |
| **Prototipo de interfaz:**   |

*Tabla 70 HU 45 Crear un año*

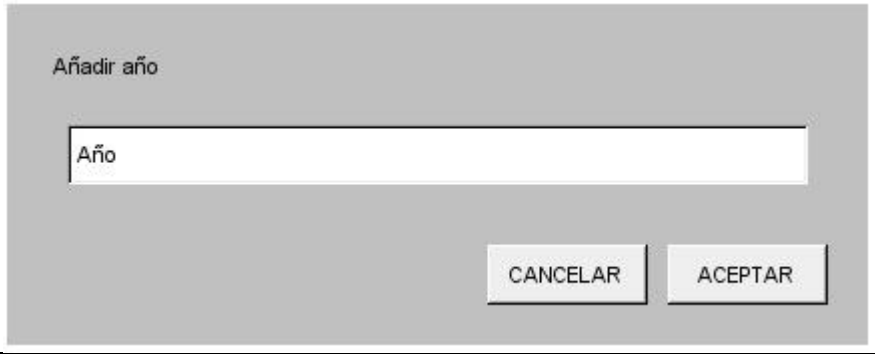
| **Historia de Usuario** | |
|---|---|
| **Número de HU:** 45 | **Nombre:** Crear un año |
| **Usuarios:** decano | |
| **Puntos de estimación:** 0.15 | **Iteración asignada:** 3 |
| **Prioridad del negocio:** Alta | **Riesgos en el desarrollo:** Alto |
| **Descripción:** La aplicación web debe brindar al decano una vista donde se permita crear un año en el sistema, ingresando en nuevo año y tomando como base los datos del año anterior. | |
| **Observaciones:** Para realizar esta acción el decano debe estar autenticado | |
| **Prototipo de interfaz:**   | |

*Tabla 71 HU 46 Eliminar un año*

| **Historia de Usuario** | |
|---|---|
| **Número de HU:** 46 | **Nombre:** Eliminar un año |

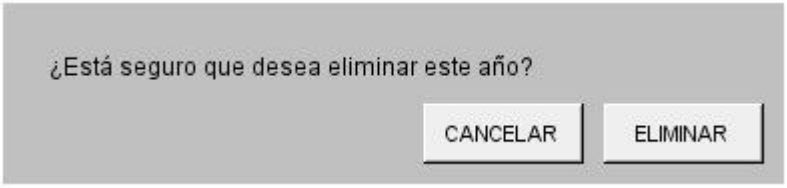
| | |
|---|---|
| **Usuarios:** decano | |
| **Puntos de estimación:** 0.15 | **Iteración asignada:** 3 |
| **Prioridad del negocio:** Alta | **Riesgos en el desarrollo:** Alto |
| **Descripción:** La aplicación web debe brindar al decano una vista donde se permita eliminar un año en el sistema. | |
| **Observaciones:** Para realizar esta acción el decano debe estar autenticado. | |
| **Prototipo de interfaz:**   | |

*Tabla 72 HU 47 Crear un departamento*

| **Historia de Usuario** | |
|---|---|
| **Número de HU:** 47 | **Nombre:** Crear un departamento |
| **Usuarios:** decano | |
| **Puntos de estimación:** 0.1 | **Iteración asignada:** 3 |
| **Prioridad del negocio:** Baja | **Riesgos en el desarrollo:** Bajo |
| **Descripción:** La aplicación web debe brindar al decano una vista donde se permita crear un departamento en el sistema, ingresando el nombre de dicho departamento. | |
| **Observaciones:** Para realizar esta acción el decano debe estar autenticado | |
| **Prototipo de interfaz:**   | |

*Tabla 73 HU 48 Eliminar un departamento*

| **Historia de Usuario** | |
|---|---|
| **Número de HU:** 48 | **Nombre:** Eliminar un departamento |
| **Usuarios:** decano | |
| **Puntos de estimación:** 0.1 | **Iteración asignada:** 3 |
| **Prioridad del negocio:** Baja | **Riesgos en el desarrollo:** Bajo |
| **Descripción:** La aplicación web debe brindar al decano una vista donde se permita eliminar un departamento en el sistema. | |

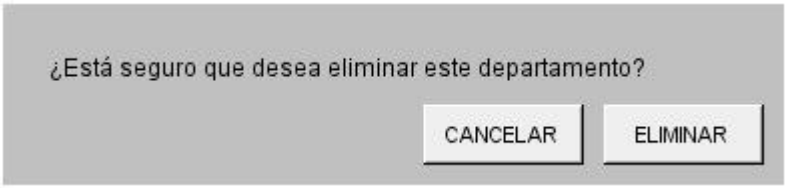
| |
|---|
| **Observaciones:** Para realizar esta acción el decano debe estar autenticado |
| **Prototipo de interfaz:**   |

*Tabla 74 HU 49 Obtener evaluación de un usuario en un año*

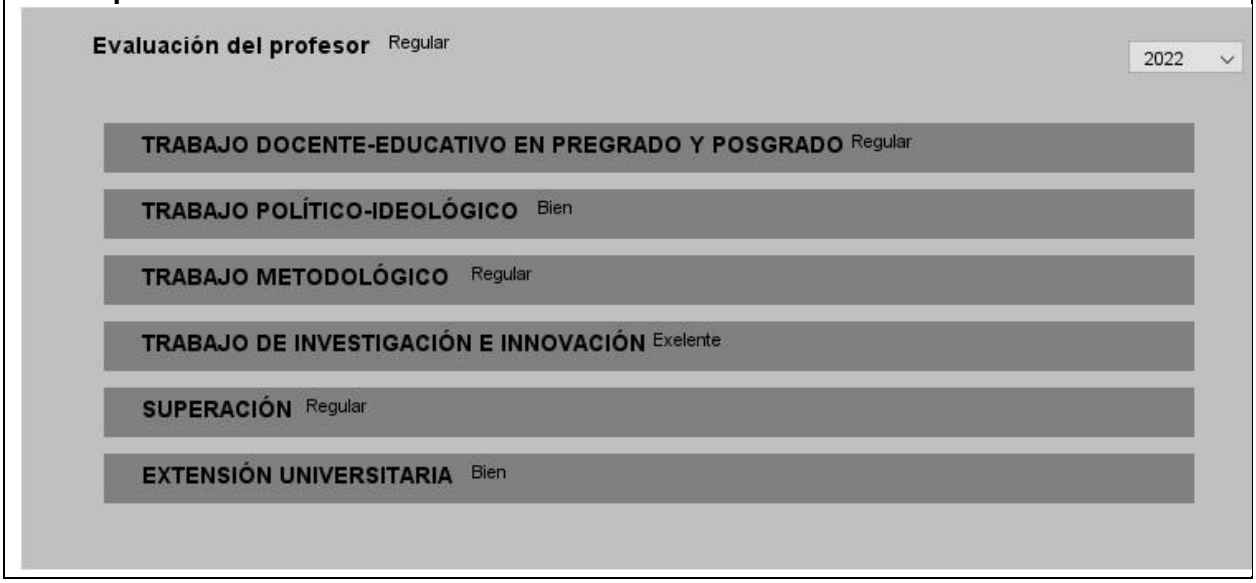
| **Historia de Usuario** | |
|---|---|
| **Número de HU:** 49 | **Nombre:** Obtener evaluación de un usuario en un año |
| **Usuarios:** jefe de departamento y usuario autenticado | |
| **Puntos de estimación:** 0.1 | **Iteración asignada:** 3 |
| **Prioridad del negocio:** Baja | **Riesgos en el desarrollo:** Baja |
| **Descripción:** La aplicación web debe brindar al jefe de departamento o al usuario autenticado una vista donde se permita obtener una evaluación de un usuario el sistema por un año determinado. | |
| **Observaciones:** Para realizar esta acción el jefe de departamento o el usuario autenticado deben estar autenticados. | |
| **Prototipo de interfaz:**   | |

*Tabla 75 HU 50 Evaluar usuario en un año*

| **Historia de Usuario** | |
|---|---|
| **Número de HU:** 50 | **Nombre:** Evaluar usuario en un año |
| **Usuarios:** jefe de departamento | |
| **Puntos de estimación:** 0.1 | **Iteración asignada:** 3 |
| **Prioridad del negocio:** Medio | **Riesgos en el desarrollo:** Medio |
| **Descripción:** La aplicación web debe brindar al jefe de departamento una vista donde se permita evaluar a un usuario el sistema por un año determinado, ingresando una evaluación por cada categoría. | |

| |
|---|
| **Observaciones:** Para realizar esta acción el jefe de departamento debe estar autenticado |
| **Prototipo de interfaz:** |

*Tabla 76 HU 51 Software*

| **Historia de Usuario** | |
|---|---|
| **Número de HU:** 51 | **Nombre:** Software |
| **Descripción:**<br>• **Cliente**<br>Navegador web Mozilla Firefox 95.0 o superior.<br>Navegador web Google Chrome 96.0 o superior.<br>• **Servidor**<br>Node.js 16 o superior.<br>MongoDB 5.0.6 o superior | |

*Tabla 77 HU 52 Hardware*

| **Historia de Usuario** | |
|---|---|
| **Número de HU:** 52 | **Nombre:** Hardware |
| **Descripción:**<br>• Microprocesador Intel Pentium 4 o superior.<br>• Memoria RAM 1 GB o superior.<br>• Disco duro con capacidad de 160 GB como mínimo.<br>• Tarjetas de red cableada, inalámbrica o modem. | |

*Tabla 78 HU 53 Apariencia o interfaz externa*

| **Historia de Usuario** | |
|---|---|
| **Número de HU:** 53 | **Nombre:** Apariencia o interfaz externa |
| **Descripción:**<br>• El diseño de las interfaces constará con pocas imágenes y colores. La información aparecerá correctamente organizada de forma tal que el usuario, pueda encontrar lo que busca rápidamente**.** | |

*Tabla 79 HU 54 Usabilidad*

| **Historia de Usuario** | |
|---|---|
| **Número de HU:** 54 | **Nombre:** Usabilidad |
| **Descripción:**<br>• El sistema debe permitir la fácil interacción con el mismo y llegar de manera rápida y efectiva a la información buscada. Debe, además, ser una interfaz de manejo cómodo que posibilite a los usuarios sin experiencia una rápida adaptación**.** | |

*Tabla 80 HU 55 Funcionalidad*

| **Historia de Usuario** | |
|---|---|
| **Número de HU:** 55 | **Nombre:** Funcionalidad |
| **Descripción:**<br>• Mínima cantidad de páginas para ejecutar todas las funciones. | |

*Tabla 81 HU 56 Seguridad*

| Historia de Usuario | |
|---|---|
| **Número de HU:** 56 | **Nombre:** Seguridad |
| **Descripción:**<br>• Se realizarán validaciones a la información contenida en el sistema. El decano, los jefes de área, jefes de departamento, profesores y especialistas se comportarán como usuarios, todos deben de estar autenticados para ver la información referente a su rol, según lo definido en el sistema. | |

*Tabla 82 HU 57 Disponibilidad*

| Historia de Usuario | |
|---|---|
| **Número de HU:** 57 | **Nombre:** Disponibilidad |
| **Descripción:**<br>• La aplicación estará disponible las 24 horas del día, los 7 días de la semana y garantizará un acceso de forma fácil y rápida para los usuarios. | |

*Tabla 83 HU 58 Portabilidad*

| Historia de Usuario | |
|---|---|
| **Número de HU:** 58 | **Nombre:** Portabilidad |
| **Descripción:**<br>• El sistema podrá ejecutarse en varios sistemas operativos como Linux y Windows. | |

## Anexos 3 Tarjetas CRC

*Tabla 84 Tarjeta CRC criterion Controller*

| Tarjeta CRC | |
|---|---|
| **Clase:** criterionController | |
| **Responsabilidad** | **Colaboración** |
| • criterionGet(req, res): retorna una lista de todos los criterios de medida que existen en la base de datos.<br>• criterionPost(req, res): crea un criterio de medida con los datos que se recibe.<br>• criterionPut(req, res): actualiza un criterio de medida en específico con los datos que se recibe.<br>• criterionDelete(req, res): elimina un criterio de medida en específico. | removeModels<br>criterion<br>objective |

*Tabla 85 Tarjeta CRC loginController*

| Tarjeta CRC |
|---|

| **Clase:** loginController | |
|---|---|
| **Responsabilidad** | **Colaboración** |
| • login(req, res): retorna un token de autorización mediante el nombre de usuario y la contraseña que se recibe. | generateJWT<br>user |

*Tabla 86 Tarjeta CRC objectiveController*

| **Tarjeta CRC** | |
|---|---|
| **Clase:** objectiveController | |
| **Responsabilidad** | **Colaboración** |
| • objetiveGet(req, res): retorna una lista de todos los objetivos estratégicos que existen en la base de datos.<br>• objetivePost(req, res): crea un objetivo estratégico con los datos que se recibe.<br>• objetivePut(req, res): actualiza un objetivo estratégico en específico con los datos que se recibe.<br>• objetiveDelete(req, res):elimina un objetivo estratégico en específico.<br>• addCriterion(req, res): añade criterios de medidas a un objetivo estratégico en específico. | removeModels<br>area<br>criterion<br>objective |

*Tabla 87 Tarjeta CRC userController*

| **Tarjeta CRC** | |
|---|---|
| **Clase:** userController | |
| **Responsabilidad** | **Colaboración** |
| • changePassword(req, res): cambiar la contraseña de un usuario en específico con los datos que se recibe.<br>• userGet(req, res): retorna una lista de todos los usuarios que existen en la base de datos | removeModels<br>indicatorResponse<br>user |

| | |
|---|---|
| • userEvaluationGet(req, res): retorna una lista de todos los indicadores personales de un usuario en específico.<br>• userNotificationsGet(req, res): retorna una lista de todos los mensajes de notificación de un usuario en específico.<br>• userPost(req, res): crea un usuario con los datos que se recibe.<br>• userPut(req, res): actualiza un usuario en específico con los datos que se recibe.<br>• userDelete(req, res): elimina un usuario en específico. | |

*Tabla 88 Tarjeta CRC areaPercentage*

| **Tarjeta CRC** | |
|---|---|
| **Clase:** areaTools | |
| **Responsabilidad** | **Colaboración** |
| • Percentage( objectives ): retorna el porcentaje de cumplimiento de un área usando la lista de objetivos que se recibe.<br>• createAllNew( lastYear, newYear): crea todas las áreas del nuevo año con toda la información referente a ellas, basado en las áreas del año anterior. | Área<br>Criterion<br>Objective<br>Indicator |

*Tabla 89 Tarjeta CRC dbValidators*

| **Tarjeta CRC** | |
|---|---|
| **Clase:** dbValidators | |
| **Responsabilidad** | **Colaboración** |
| • isRoleValid(role): verifica que el rol que se recibe por parámetro exista en la base de datos.<br>• usernameExists(username): verifica que el nombre de usuario que se recibe por parámetro exista en la base de datos. | Area<br>Criterion<br>Evidence<br>Indicator<br>Objective<br>Role<br>User |

| | |
|---|---|
| • userExistsById(id): verifica que el usuario mediante el identificador que se recibe por parámetro exista en la base de datos.<br>• areaNameExists(name): verifica que el nombre del área que se recibe por parámetro exista en la base de datos.<br>• areaExistsById(id): verifica que el área mediante el identificador que se recibe por parámetro exista en la base de datos.<br>• criterionExistsById(id): verifica que el criterio de medida mediante el identificador que se recibe por parámetro exista en la base de datos.<br>• evidenceExistsById(id): verifica que la evidencia mediante el identificador que se recibe por parámetro exista en la base de datos.<br>• indicatorExistsById(id): verifica que el indicador mediante el identificador que se recibe por parámetro exista en la base de datos.<br>• objectiveExistsById(id): verifica que el objetivo estratégico mediante el identificador que se recibe por parámetro exista en la base de datos. | |

*Tabla 90 Tarjeta CRC generateJWT*

| **Tarjeta CRC** | |
|---|---|
| **Clase:** generateJWT | |
| **Responsabilidad** | **Colaboración** |
| • jwt(uid, role, name, username, department): retorna un token de autorización que contiene en su payload la información que se recibe por parámetro. | |

*Tabla 91 Tarjeta CRC indicatorResponse*

| **Tarjeta CRC** | |
|---|---|
| **Clase:** indicatorResponse | |
| **Responsabilidad** | **Colaboración** |
| • indicatorsByCategory(categories, department, user): retorna una lista de indicadores ordenados por categorías.<br>• personalIndicators(categories, department, user): retorna una lista de indicadores peersonales ordenados por categorías. | Area<br>Indicator<br>Objective |

| | |
|---|---|
| • indicatorArea(criterionId) retorna el nombre del área de resultados claves al que pertenece el criterio del cual se recibe su identificador por parámetro. | |

*Tabla 92 Tarjeta CRC modifyCriterion*

| **Tarjeta CRC** | |
|---|---|
| **Clase:** modifyCriterion | |
| **Responsabilidad** | **Colaboración** |
| • updateCriterion(id, value): modifica el valor del ToDo usando el valor que se recibe por parametro, haciendo que cambie a su vez el estado del criterio de medida que se recibe mediante un identificador por parámetros. | criterion |

*Tabla 93 Tarjeta CRC removeModels*

| **Tarjeta CRC** | |
|---|---|
| **Clase:** removeModels | |
| **Responsabilidad** | **Colaboración** |
| • deleteEvidence(id, idIndicator): elimina la evidencia mediante el identificador que se recibe por parámetro y elimina la asociación con el indicador que se recibe mediante un identificador por parámetro.<br>• deleteIndicator(id, idUser, io): elimina el indicador mediante el identificador que se recibe por parámetro y elimina la asociación con el usuario que se recibe mediante un identificador por parámetro.<br>• deleteCriterion(id, idObjective): elimina el criterio de medida mediante el identificador que se recibe por parámetro y elimina la asociación con el objetivo estratégico que se recibe mediante un identificador por parámetro.<br>• deleteObjective(id, idArea): elimina el objetivo estratégico mediante el identificador que se recibe por parámetro y elimina la asociación con el área que se recibe mediante un identificador por parámetro.<br>• deleteArea(id): elimina el área mediante el identificador que se recibe por parámetro. | modifyCriterion<br>area<br>criterion<br>evidence<br>indicator<br>objective<br>user |

| | |
|---|---|
| • deleteUser(id): elimina el usuario mediante el identificador que se recibe por parámetro. | |

*Tabla 94 Tarjeta CRC uploadFile*

| **Tarjeta CRC** | |
|---|---|
| **Clase:** uploadFile | |
| **Responsabilidad** | **Colaboración** |
| • upload(file): guarda en una carpeta del servidor el archivo que se recibe por parámetro. | |

*Tabla 95 Tarjeta CRC evaluationController*

| **Tarjeta CRC** | |
|---|---|
| **Clase:** evaluationController | |
| **Responsabilidad** | **Colaboración** |
| • evaluationGet(req, res): retorna una lista de todas las evaluaciones que existen en la base de datos.<br>• evaluationPost(req, res): crea una evaluación con los datos que se recibe.<br>• evaluationPut(req, res): actualiza una evaluación en específico con los datos que se recibe.<br>• evaluationDelete(req, res): elimina una evaluación en específico. | removeModels<br>evaluation<br>user |

*Tabla 96 Tarjeta CRC yearController*

| **Tarjeta CRC** | |
|---|---|
| **Clase:** yearController | |
| **Responsabilidad** | **Colaboración** |
| • getLastOne(req, res): retorna el último año insertado en el sistema.<br>• newYearPost(req, res): crea un nuevo año en el sistema.<br>• yearsGet(req, res): retorna una lista con todos los años que existen en el sistema.<br>• removeOne(req, res): elimina un año del sistema en específico. | areaTools<br>year |

| | |
|---|---|
| • yearDepartamentGet (req, res): retorna una lista con los nombres de los departamntos.<br>• yearDepartamentPut(req, res): inserta un nombre de departamento a la lista de departamento.<br>• yearDepartamentDelete (req, res): elimina un nombre de departamento de la lista de nombre de departamentos. | |

## Anexos 4 Tareas de Ingeniería

*Tabla 97 Tareas de Ingeniería 5*

| **Tarea** | |
|---|---|
| **Número de tarea:** 5 | **Número de Historia de Usuario:** 5 |
| **Nombre de la tarea:** Implementar la creación de un objetivo estratégico en el sistema | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 0.6 |
| **Programador responsable:** Arián León Benítez | |
| **Descripción:** Se implementa una funcionalidad que permite crear un objetivo estratégico y asociarlo a un área usando el identificador de dicha área. | |

*Tabla 98 Tareas de Ingeniería 6*

| **Tarea** | |
|---|---|
| **Número de tarea:** 6 | **Número de Historia de Usuario:** 6 |
| **Nombre de la tarea:** Implementar la obtención de los objetivos estratégicos del sistema | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 0.6 |
| **Programador responsable:** Arián León Benítez | |
| **Descripción:** Se implementa una funcionalidad que permite obtener todos los objetivos estratégicos del sistema. | |

*Tabla 99 Tareas de Ingeniería 7*

| **Tarea** | |
|---|---|
| **Número de tarea:** 7 | **Número de Historia de Usuario:** 7 |
| **Nombre de la tarea:** Implementar la actualización de un objetivo estratégico en el sistema | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 0.6 |
| **Programador responsable:** Arián León Benítez | |
| **Descripción:** Se implementa una funcionalidad que permite actualizar un objetivo estratégico mediante su identificador. | |

*Tabla 100 Tareas de Ingeniería 8*

| **Tarea** | |
|---|---|
| **Número de tarea:** 8 | **Número de Historia de Usuario:** 8 |
| **Nombre de la tarea:** Implementar la eliminación de un objetivo estratégico en el sistema | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 0.6 |
| **Programador responsable:** Arián León Benítez | |
| **Descripción:** Se implementa una funcionalidad que permite eliminar un objetivo estratégico mediante su identificador. | |

*Tabla 101 Tareas de Ingeniería 9*

| **Tarea** | |
|---|---|
| **Número de tarea:** 9 | **Número de Historia de Usuario:** 9 |
| **Nombre de la tarea:** Implementar la creación de un criterio de medida en el sistema | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 0.6 |
| **Programador responsable:** Arián León Benítez | |
| **Descripción:** Se implementa una funcionalidad que permite crear un criterio de medida y asociarlo a un objetivo estratégico usando el identificador de dicho objetivo estratégico. | |

*Tabla 102 Tareas de Ingeniería 10*

| **Tarea** | |
|---|---|
| **Número de tarea:** 10 | **Número de Historia de Usuario:** 10 |
| **Nombre de la tarea:** Implementar la obtención de los criterios de medidas del sistema | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 0.6 |
| **Programador responsable:** Arián León Benítez | |
| **Descripción:** Se implementa una funcionalidad que permite obtener todos los criterios de medida del sistema. | |

*Tabla 103 Tareas de Ingeniería 11*

| **Tarea** | |
|---|---|
| **Número de tarea:** 11 | **Número de Historia de Usuario:** 11 |
| **Nombre de la tarea:** Implementar la actualización de un criterio de medida en el sistema | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 0.6 |
| **Programador responsable:** Arián León Benítez | |
| **Descripción:** Se implementa una funcionalidad que permite actualizar un criterio de medida mediante su identificador. | |

*Tabla 104 Tareas de Ingeniería 12*

| **Tarea** | |
|---|---|
| **Número de tarea:** 12 | **Número de Historia de Usuario:** 12 |

| | |
|---|---|
| **Nombre de la tarea:** Implementar la eliminación de un criterio de medida en el sistema | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 0.6 |
| **Programador responsable:** Arián León Benítez | |
| **Descripción:** Se implementa una funcionalidad que permite eliminar un criterio de medida mediante su identificador. | |

*Tabla 105 Tareas de Ingeniería 13*

| **Tarea** | |
|---|---|
| **Número de tarea:** 13 | **Número de Historia de Usuario:** 13 |
| **Nombre de la tarea:** Implementar la creación de un indicador en el sistema | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:**0.6 |
| **Programador responsable:** Javier Ceballo Perez | |
| **Descripción:** Se implementa una funcionalidad que permita crear un indicador y asociarlo a un criterio de medida usando el identificador de dicho criterio | |

*Tabla 106 Tareas de Ingeniería 14*

| **Tarea** | |
|---|---|
| **Número de tarea:** 14 | **Número de Historia de Usuario:** 14 |
| **Nombre de la tarea:** Implementar la obtención de los indicadores en el sistema | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:**0.6 |
| **Programador responsable:** Javier Ceballo Perez | |
| **Descripción:** Se implementa una funcionalidad que permita obtener los indicadores | |

*Tabla 107 Tareas de Ingeniería 15*

| **Tarea** | |
|---|---|
| **Número de tarea:** 15 | **Número de Historia de Usuario:** 15 |
| **Nombre de la tarea:** Implementar la actualización de un indicador en el sistema | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:**0.6 |
| **Programador responsable:** Javier Ceballo Perez | |
| **Descripción:** Se implementa una funcionalidad que permita actualizar un indicador mediante su identificador | |

*Tabla 108 Tareas de Ingeniería 16*

| **Tarea** | |
|---|---|
| **Número de tarea:** 16 | **Número de Historia de Usuario:** 16 |
| **Nombre de la tarea:** Implementar la eliminación de un indicador en el sistema | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:**0.6 |
| **Programador responsable:** Javier Ceballo Perez | |
| **Descripción:** Se implementa una funcionalidad que permita eliminar un indicador mediante su identificador | |

*Tabla 109 Tareas de Ingeniería 17*

| **Tarea** |
|---|

| | |
|---|---|
| **Número de tarea:** 17 | **Número de Historia de Usuario:** 17 |
| **Nombre de la tarea:** Implementar la creación de un usuario en el sistema | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:**0.6 |
| **Programador responsable:** Javier Ceballo Perez | |
| **Descripción:** Se implementa una funcionalidad que permita crear un usuario estableciéndole por defecto la contraseña “facultad4” | |

*Tabla 110 Tareas de Ingeniería 18*

| **Tarea** | |
|---|---|
| **Número de tarea:** 18 | **Número de Historia de Usuario:** 18 |
| **Nombre de la tarea:** Implementar la obtención de los usuarios en el sistema | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:**0.6 |
| **Programador responsable:** Javier Ceballo Perez | |
| **Descripción:** Se implementa una funcionalidad que permita obtener los usuarios del sistema | |

*Tabla 111 Tareas de Ingeniería 19*

| **Tarea** | |
|---|---|
| **Número de tarea:** 19 | **Número de Historia de Usuario:** 19 |
| **Nombre de la tarea:** Implementar la actualización de un usuario en el sistema | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:**0.6 |
| **Programador responsable:** Javier Ceballo Perez | |
| **Descripción:** Se implementa una funcionalidad que permita actualizar un usuario mediante su identificador | |

*Tabla 112 Tareas de Ingeniería 20*

| **Tarea** | |
|---|---|
| **Número de tarea:** 20 | **Número de Historia de Usuario:** 20 |
| **Nombre de la tarea:** Implementar la eliminación de un usuario en el sistema | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:**0.6 |
| **Programador responsable:** Javier Ceballo Perez | |
| **Descripción:** Se implementa una funcionalidad que permita eliminar un usuario mediante su indicador | |

*Tabla 113 Tareas de Ingeniería 21*

| **Tarea** | |
|---|---|
| **Número de tarea:** 21 | **Número de Historia de Usuario:** 21 |
| **Nombre de la tarea:** Implementar la creación de una evidencia en el sistema | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:**0.6 |
| **Programador responsable:** Javier Ceballo Perez | |
| **Descripción:** Se implementa una funcionalidad que permita crear una evidencia y asociarla a un indicador usando el identificador de dicho indicador | |

*Tabla 114 Tareas de Ingeniería 22*

| **Tarea** | |
|---|---|
| **Número de tarea:** 22 | **Número de Historia de Usuario:** 22 |
| **Nombre de la tarea:** Implementar la obtención de la evidencia en el sistema | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:**0.6 |
| **Programador responsable:** Javier Ceballo Perez | |
| **Descripción:** Se implementa una funcionalidad que permita obtener las evidencias | |

*Tabla 115 Tareas de Ingeniería 23*

| **Tarea** | |
|---|---|
| **Número de tarea:** 23 | **Número de Historia de Usuario:** 23 |
| **Nombre de la tarea:** Implementar la actualización de una evidencia en el sistema | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:**0.6 |
| **Programador responsable:** Javier Ceballo Perez | |
| **Descripción:** Se implementa una funcionalidad que permita actualizar una evidencia mediante su identificador | |

*Tabla 116 Tareas de Ingeniería 24*

| **Tarea** | |
|---|---|
| **Número de tarea:** 24 | **Número de Historia de Usuario:** 24 |
| **Nombre de la tarea:** Implementar la eliminación de una evidencia en el sistema | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:**0.6 |
| **Programador responsable:** Javier Ceballo Perez | |
| **Descripción:** Se implementa una funcionalidad que permita eliminar una evidencia mediante su identificador | |

*Tabla 117 Tareas de Ingeniería 29*

| **Tarea** | |
|---|---|
| **Número de tarea:** 29 | **Número de Historia de Usuario:** 29 |
| **Nombre de la tarea:** Implementar la adjunción de un archivo a una evidencia | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 1 |
| **Programador responsable:** Javier Ceballo Perez | |
| **Descripción:** Se implementa una funcionalidad que permita adjuntar un archivo a una evidencia mediante la identificación de dicha evidencia | |

*Tabla 118 Tareas de Ingeniería 30*

| **Tarea** | |
|---|---|
| **Número de tarea:** 30 | **Número de Historia de Usuario:** 30 |
| **Nombre de la tarea:** Implementar el cambio de contraseña de un usuario | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 1 |
| **Programador responsable:** Javier Ceballo Perez | |
| **Descripción:** Se implementa una funcionalidad que permita cambiar la contraseña de un usuario por una nueva mediante el identificador dicho usuario. | |

*Tabla 119 Tareas de Ingeniería 31*

| **Tarea** | |
|---|---|
| **Número de tarea:** 31 | **Número de Historia de Usuario:** 31 |
| **Nombre de la tarea:** Implementar la asignación de indicadores a un usuario | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 1 |
| **Programador responsable:** Javier Ceballo Perez | |
| **Descripción:** Se implementa una funcionalidad que permita asignar una lista de indicadores a un usuario mediante el identificador del usuario en cuestión | |

*Tabla 120 Tareas de Ingeniería 32*

| **Tarea** | |
|---|---|
| **Número de tarea:** 32 | **Número de Historia de Usuario:** 32 |
| **Nombre de la tarea:** Implementar la creación de un indicador personal | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 1 |
| **Programador responsable:** Javier Ceballo Perez | |
| **Descripción:** Se implementa una funcionalidad que permita crear un indicador en estado de “Cumplido” y asignarlo a un usuario mediante su identificador. | |

*Tabla 121 Tareas de Ingeniería 33*

| **Tarea** | |
|---|---|
| **Número de tarea:** 33 | **Número de Historia de Usuario:** 33 |
| **Nombre de la tarea:** Implementar el cambio de estado de un indicador | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 1 |
| **Programador responsable:** Javier Ceballo Perez | |
| **Descripción:** Se implementa una funcionalidad que permita cambiar el estado de un indicador a verdadero o false de forma automática mediante la asignación o eliminación de una evidencia a este | |

*Tabla 122 Tareas de Ingeniería 34*

| **Tarea** | |
|---|---|
| **Número de tarea:** 34 | **Número de Historia de Usuario:** 34 |
| **Nombre de la tarea:** Implementar el restablecimiento de la contraseña de un usuario | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 1 |
| **Programador responsable:** Javier Ceballo Perez | |
| **Descripción:** Se implementa una funcionalidad que permita cambiar la contraseña de un usuario a “facultad4” mediante su identificador | |

*Tabla 123 Tareas de Ingeniería 35*

| **Tarea** | |
|---|---|
| **Número de tarea:** 35 | **Número de Historia de Usuario:** 35 |
| **Nombre de la tarea:** Implementar la búsqueda de usuarios en el sistema | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 1 |
| **Programador responsable:** Javier Ceballo Perez | |

| |
|---|
| **Descripción:** Se implementa una funcionalidad que permita obtener los usuarios que coincidan sus nombres y nombres de usuario con un carácter dado |

*Tabla 124 Tareas de Ingeniería 40*

| **Tarea** | |
|---|---|
| **Número de tarea:** 40 | **Número de Historia de Usuario:** 40 |
| **Nombre de la tarea:** Implementar la denegación del cumplimiento de un indicador | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 1 |
| **Programador responsable:** Arián León Benítez | |
| **Descripción:** Se implementa una funcionalidad que permite cambiar el estado de un indicador de verdadero a falso. | |

*Tabla 125 Tareas de Ingeniería 41*

| **Tarea** | |
|---|---|
| **Número de tarea:** 41 | **Número de Historia de Usuario:** 41 |
| **Nombre de la tarea:** Implementar la asignación de una observación a un indicador | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 1 |
| **Programador responsable:** Arián León Benítez | |
| **Descripción:** Se implementa una funcionalidad que permite agregarle una observación de tipo texto a un indicador mediante el identificador de dicho indicador. | |

*Tabla 126 Tareas de Ingeniería 42*

| **Tarea** | |
|---|---|
| **Número de tarea:** 42 | **Número de Historia de Usuario:** 42 |
| **Nombre de la tarea:** Implementar la obtención de áreas de resultados claves por un año | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 0.2 |
| **Programador responsable:** Arián León Benítez | |
| **Descripción:** Se implementa una funcionalidad que permite obtener un listado de las áreas en un año determinado. | |

*Tabla 127 Tareas de Ingeniería 43*

| **Tarea** | |
|---|---|
| **Número de tarea:** 43 | **Número de Historia de Usuario:** 43 |
| **Nombre de la tarea:** Implementar la obtención de indicadores de un usuario por un año | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 0.2 |
| **Programador responsable:** Arián León Benítez | |
| **Descripción:** Se implementa una funcionalidad que permite obtener los indicadores asociados a un usuario en un año determinado. | |

*Tabla 128 Tareas de Ingeniería 44*

| **Tarea** | |
|---|---|
| **Número de tarea:** 44 | **Número de Historia de Usuario:** 44 |
| **Nombre de la tarea:** Implementar la asignación de una fecha a la evidencia | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 0.2 |
| **Programador responsable:** Arián León Benítez | |

| |
|---|
| **Descripción:** Se implementa una funcionalidad que permite asignar una fecha a la evidencia. |

*Tabla 129 Tareas de Ingeniería 45*

| **Tarea** | |
|---|---|
| **Número de tarea:** 45 | **Número de Historia de Usuario:** 45 |
| **Nombre de la tarea:** Implementar la creación de un año en el sistema | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 0.75 |
| **Programador responsable:** Arián León Benítez | |
| **Descripción:** Se implementa una funcionalidad que permite crear un año en el sistema donde usando como base las áreas del año anterior se asignan a este nuevo año, pero con los registros de actividades reiniciados. | |

*Tabla 130 Tareas de Ingeniería 46*

| **Tarea** | |
|---|---|
| **Número de tarea:** 46 | **Número de Historia de Usuario:** 46 |
| **Nombre de la tarea:** Implementar la eliminación de un año del sistema | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 0.75 |
| **Programador responsable:** Arián León Benítez | |
| **Descripción:** Se implementa una funcionalidad que permite eliminar un año del sistema y esto a su vez elimina las áreas con todo lo referentes a ellas de dicho año. | |

*Tabla 131 Tareas de Ingeniería 47*

| **Tarea** | |
|---|---|
| **Número de tarea:** 47 | **Número de Historia de Usuario:** 47 |
| **Nombre de la tarea:** Implementar la creación de un departamento | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 0.2 |
| **Programador responsable:** Arián León Benítez | |
| **Descripción:** Se implementa una funcionalidad que permite crear un departamento. | |

*Tabla 132 Tareas de Ingeniería 48*

| **Tarea** | |
|---|---|
| **Número de tarea:** 48 | **Número de Historia de Usuario:** 48 |
| **Nombre de la tarea:** Implementar la eliminación de un departamento | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 0.2 |
| **Programador responsable:** Javier Ceballo Perez | |
| **Descripción:** Se implementa una funcionalidad que permita eliminar un departamento. | |

*Tabla 133 Tareas de Ingeniería 49*

| **Tarea** | |
|---|---|
| **Número de tarea:** 49 | **Número de Historia de Usuario:** 49 |
| **Nombre de la tarea:** Implementar la obtención de una evaluación de un usuario en un año | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 0.2 |
| **Programador responsable:** Javier Ceballo Perez | |
| **Descripción:** Se implementa una funcionalidad que permita obtener una evaluación de un usuario en el sistema mediante un año determinado | |

*Tabla 134 Tareas de Ingeniería 50*

| **Tarea** | |
|---|---|
| **Número de tarea:** 50 | **Número de Historia de Usuario:** 50 |
| **Nombre de la tarea:** Implementar la evaluación de un usuario en un año | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 0.2 |
| **Programador responsable:** Javier Ceballo Perez | |
| **Descripción:** Se implementa una funcionalidad que permita evaluar un usuario en el sistema mediante un año determinado | |

## Anexos 5 Casos de prueba de aceptación

*Tabla 135 Prueba de Aceptación HU5-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU5_P1 | **Historia de Usuario:** 5 |
| **Nombre:** Crear un objetivo estratégico | |
| **Descripción:** Prueba de funcionalidad: crear un objetivo estratégico | |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol jefe de área o decano | |
| **Pasos de ejecución:**<br>1. El jefe de área o decano accede a la sección de las áreas<br>2. El jefe de área o decano selecciona la opción “ver más” perteneciente a su área<br>3. El jefe de área o decano selecciona la opción “añadir objetivos”<br>4. El jefe de área o decano introduce los datos<br>5. El jefe de área o decano crea un objetivo estratégico | |
| **Resultado:** Satisfactorio | |

*Tabla 136 Prueba de Aceptación HU6-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU6_P1 | **Historia de Usuario:** 6 |
| **Nombre:** Actualizar un objetivo estratégico | |
| **Descripción:** Prueba de funcionalidad: actualizar un objetivo estratégico | |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol jefe de área o decano<br>• Debe existir al menos un objetivo estratégico que pertenezca a dicha área | |
| **Pasos de ejecución:**<br>1. El jefe de área o decano accede a la sección de las áreas<br>2. El jefe de área o decano selecciona la opción “ver más” perteneciente a su área<br>3. El jefe de área o decano selecciona el icono de editar perteneciente a un objetivo estratégico<br>4. El jefe de área o decano reemplaza los datos<br>5. Se actualiza un objetivo estratégico | |
| **Resultado:** Satisfactorio | |

*Tabla 137 Prueba de Aceptación HU7-P1*

| **Caso de prueba de aceptación** |
|---|

| **Código:** HU7_P1 | **Historia de Usuario:** 7 |
|---|---|
| **Nombre:** Obtener objetivos estratégicos | |
| **Descripción:** Prueba de funcionalidad: obtener objetivos estratégicos | |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol jefe de área o decano<br>• Debe existir al menos un área en el sistema | |
| **Pasos de ejecución:**<br>1. El jefe de área o decano accede a la sección de las áreas<br>2. El jefe de área o decano selecciona la opción “ver más” perteneciente a su área<br>3. Se obtiene los objetivos estratégicos pertenecientes al área | |
| **Resultado:** Satisfactorio | |

*Tabla 138 Prueba de Aceptación HU8-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU8_P1 | **Historia de Usuario:** 8 |
| **Nombre:** Eliminar un objetivo estratégico | |
| **Descripción:** Prueba de funcionalidad: eliminar un objetivo estratégico | |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol jefe de área o decano<br>• Debe existir al menos un área en el sistema<br>• Debe existir al menos un objetivo estratégico que pertenezca a dicha área | |
| **Pasos de ejecución:**<br>1. El jefe de área o decano accede a la sección de las áreas<br>2. El jefe de área o decano selecciona la opción “ver más” perteneciente a su área<br>3. El jefe de área o decano selecciona la opción “eliminar objetivo” perteneciente a un objetivo estratégico<br>4. El jefe de área o decano selecciona la opción “eliminar”<br>5. El jefe de área o decano elimina un objetivo estratégico | |
| **Resultado:** Satisfactorio | |

*Tabla 139 Prueba de Aceptación HU9-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU9_P1 | **Historia de Usuario:** 9 |
| **Nombre:** Crear un criterio de medida | |
| **Descripción:** Prueba de funcionalidad: crear un criterio de medida | |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol jefe de área o decano<br>• Debe existir al menos un área en el sistema<br>• Debe existir al menos un objetivo estratégico que pertenezca a dicha área | |
| **Pasos de ejecución:**<br>1. El jefe de área o decano accede a la sección de las áreas<br>2. El jefe de área o decano selecciona la opción “ver más” perteneciente a su área<br>3. El jefe de área o decano selecciona la opción “añadir criterios de medida”<br>4. El jefe de área o decano introduce los datos<br>5. El jefe de área o decano crea un criterio de medida | |

| |
|---|
| **Resultado:** Satisfactorio |

*Tabla 140 Prueba de Aceptación HU10-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU10_P1 | **Historia de Usuario:** 10 |
| **Nombre:** actualizar un criterio de medida | |
| **Descripción:** Prueba de funcionalidad: actualizar un criterio de medida | |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol jefe de área o decano<br>• Debe existir al menos un área en el sistema<br>• Debe existir al menos un objetivo estratégico que pertenezca a dicha área<br>• Debe existir al menos un criterio de medida que pertenezca a dicho objetivo estratégico | |
| **Pasos de ejecución:**<br>1. El jefe de área o decano accede a la sección de las áreas<br>2. El jefe de área o decano selecciona la opción “ver más” perteneciente a su área<br>3. El jefe de área o decano selecciona el ícono de editar perteneciente al criterio de medida<br>4. El jefe de área o decano introduce los datos<br>5. Se actualiza un criterio de medida | |
| **Resultado:** Satisfactorio | |

*Tabla 141 Prueba de Aceptación HU11-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU11_P1 | **Historia de Usuario:** 11 |
| **Nombre:** Obtener criterios de medidas | |
| **Descripción:** Prueba de funcionalidad: obtener criterios de medidas | |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol jefe de área o decano<br>• Debe existir al menos un área en el sistema<br>• Debe existir al menos un objetivo estratégico que pertenezca a dicha área<br>• Debe existir al menos un criterio de medida que pertenezca a dicho objetivo estratégico | |
| **Pasos de ejecución:**<br>1. El jefe de área o decano accede a la sección de las áreas<br>2. El jefe de área o decano selecciona la opción “ver más” perteneciente a su área<br>3. Se obtiene los criterios de medida pertenecientes al objetivo estratégico | |
| **Resultado:** Satisfactorio | |

*Tabla 142 Prueba de Aceptación HU12-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU12_P1 | **Historia de Usuario:** 12 |
| **Nombre:** Eliminar un criterio de medida | |
| **Descripción:** Prueba de funcionalidad: eliminar un criterio de medida | |
| **Condiciones de ejecución:** | |

| |
|---|
| • El usuario debe estar previamente autenticado y contar con el rol jefe de área o decano<br>• Debe existir al menos un área en el sistema<br>• Debe existir al menos un objetivo estratégico que pertenezca a dicha área<br>• Debe existir al menos un criterio de medida que pertenezca a dicho objetivo estratégico |
| **Pasos de ejecución:**<br>1. El jefe de área o decano accede a la sección de las áreas<br>2. El jefe de área o decano selecciona la opción “ver más” perteneciente a su área<br>3. El jefe de área o decano selecciona el ícono de eliminar perteneciente al criterio de medida<br>4. El jefe de área o decano selecciona la opción “eliminar”<br>5. El jefe de área o decano elimina un criterio de medida |
| **Resultado:** Satisfactorio |

*Tabla 143 Prueba de Aceptación HU13-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU13_P1 | **Historia de Usuario:** 13 |
| **Nombre:** Crear un indicador | |
| **Descripción:** Prueba de funcionalidad: crear un indicador | |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol jefe de área o decano<br>• Debe existir al menos un área en el sistema<br>• Debe existir al menos un objetivo estratégico que pertenezca a dicha área<br>• Debe existir al menos un criterio de medida que pertenezca a dicho objetivo estratégico | |
| **Pasos de ejecución:**<br>1. El jefe de área o decano accede a la sección de las áreas<br>2. El jefe de área o decano selecciona la opción “ver más” perteneciente a su área<br>3. El jefe de área o decano selecciona el icono de añadir indicadores<br>4. El jefe de área o decano introduce los datos<br>5. El jefe de área o decano crea un indicador | |
| **Resultado:** Satisfactorio | |

*Tabla 144 Prueba de Aceptación HU14-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU14_P1 | **Historia de Usuario:** 14 |
| **Nombre:** Actualizar un indicador | |
| **Descripción:** Prueba de funcionalidad: actualizar un indicador | |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol jefe de área o decano<br>• Debe existir al menos un área en el sistema<br>• Debe existir al menos un objetivo estratégico que pertenezca a dicha área<br>• Debe existir al menos un criterio de medida que pertenezca a dicho objetivo estratégico<br>• Debe existir al menos un indicador que pertenezca a dicho criterio de medida | |

| |
|---|
| **Pasos de ejecución:**<br>1. El jefe de área o decano accede a la sección de las áreas<br>2. El jefe de área o decano selecciona la opción “ver más” perteneciente a su área<br>3. El jefe de área o decano selecciona el icono de editar indicadores<br>4. El jefe de área o decano reemplaza los datos<br>5. Se actualiza un indicador |
| **Resultado:** Satisfactorio |

*Tabla 145 Prueba de Aceptación HU15-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU15_P1 | **Historia de Usuario:** 15 |
| **Nombre:** Obtener indicadores | |
| **Descripción:** Prueba de funcionalidad: obtener indicadores | |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol jefe de área o decano<br>• Debe existir al menos un área en el sistema<br>• Debe existir al menos un objetivo estratégico que pertenezca a dicha área<br>• Debe existir al menos un criterio de medida que pertenezca a dicho objetivo estratégico<br>• Debe existir al menos un indicador que pertenezca a dicho criterio de medida | |
| **Pasos de ejecución:**<br>1. El jefe de área o decano accede a la sección de las áreas<br>2. El jefe de área o decano selecciona la opción “ver más” perteneciente a su área<br>3. Se obtiene los indicadores pertenecientes al criterio de medida | |
| **Resultado:** Satisfactorio | |

*Tabla 146 Prueba de Aceptación HU16-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU16_P1 | **Historia de Usuario:** 16 |
| **Nombre:** Eliminar un indicador | |
| **Descripción:** Prueba de funcionalidad: eliminar un indicador | |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol jefe de área o decano<br>• Debe existir al menos un área en el sistema<br>• Debe existir al menos un objetivo estratégico que pertenezca a dicha área<br>• Debe existir al menos un criterio de medida que pertenezca a dicho objetivo estratégico<br>• Debe existir al menos un indicador que pertenezca a dicho criterio de medida | |
| **Pasos de ejecución:**<br>1. El jefe de área o decano accede a la sección de las áreas<br>2. El jefe de área o decano selecciona la opción “ver más” perteneciente a su área<br>3. El jefe de área o decano selecciona el icono de eliminar indicador<br>4. El jefe de área o decano selecciona la opción “eliminar”<br>5. El jefe de área o decano elimina un indicador | |
| **Resultado:** Satisfactorio | |

*Tabla 147 Prueba de Aceptación HU17-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU17_P1 | **Historia de Usuario:** 17 |
| **Nombre:** Crear un usuario | |
| **Descripción:** Prueba de funcionalidad: crear un usuario | |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol decano(a) | |
| **Pasos de ejecución:**<br>1. El decano accede a la sección de los usuarios<br>2. El decano selecciona la opción para crear un usuario<br>3. El decano introduce los datos<br>4. El decano crea el usuario | |
| **Resultado:** Satisfactorio | |

*Tabla 148 Prueba de Aceptación HU18-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU18_P1 | **Historia de Usuario:** 18 |
| **Nombre:** Actualizar un usuario | |
| **Descripción:** Prueba de funcionalidad: actualizar un usuario | |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol decano<br>• Debe existir al menos un usuario en el sistema | |
| **Pasos de ejecución:**<br>1. El decano accede a la sección de los usuarios<br>2. El decano selecciona la opción “editar” perteneciente al usuario<br>3. El decano reemplaza los datos<br>4. Se actualiza el usuario | |
| **Resultado:** Satisfactorio | |

*Tabla 149 Prueba de Aceptación HU19-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU19_P1 | **Historia de Usuario:** 19 |
| **Nombre:** Obtener usuarios | |
| **Descripción:** Prueba de funcionalidad: obtener usuarios | |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol decano o jefe de departamento<br>• Debe existir al menos un usuario en el sistema | |
| **Pasos de ejecución:**<br>1. El decano o jefe de departamento accede a la sección de los usuarios<br>2. Se obtienen los usuarios del sistema | |
| **Resultado:** Satisfactorio | |

*Tabla 150 Prueba de Aceptación HU20-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU20_P1 | **Historia de Usuario:** 20 |
| **Nombre:** Eliminar un usuario | |

| |
|---|
| **Descripción:** Prueba de funcionalidad: eliminar un usuario |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol decano<br>• Debe existir al menos un usuario en el sistema |
| **Pasos de ejecución:**<br>1. El decano accede a la sección de los usuarios<br>2. El decano selecciona la opción “eliminar” perteneciente al usuario<br>3. El decano selecciona la opción “eliminar”<br>4. Se elimina el usuario |
| **Resultado:** Satisfactorio |

*Tabla 151 Prueba de Aceptación HU21-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU21_P1 | **Historia de Usuario:** 21 |
| **Nombre:** Crear una evidencia | |
| **Descripción:** Prueba de funcionalidad: crear una evidencia | |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol jefe de área, jefe de departamento o usuario autenticado<br>• El jefe de área, jefe de departamento o usuario autenticado deben de tener indicadores asignados | |
| **Pasos de ejecución:**<br>1. El jefe de área, jefe de departamento o usuario autenticado accede a su plan de indicadores a cumplir<br>2. El jefe de área, jefe de departamento o usuario autenticado selecciona un indicador<br>3. El jefe de área, jefe de departamento o usuario autenticado selecciona la opción para crear una evidencia<br>4. El jefe de área, jefe de departamento o usuario autenticado introduce los datos<br>5. Se crea una evidencia | |
| **Resultado:** Satisfactorio | |

*Tabla 152 Prueba de Aceptación HU22-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU22_P1 | **Historia de Usuario:** 22 |
| **Nombre:** Actualizar una evidencia | |
| **Descripción:** Prueba de funcionalidad: actualizar una evidencia | |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol jefe de área, jefe de departamento o usuario autenticado<br>• El jefe de área, jefe de departamento o usuario autenticado deben de tener indicadores asignados<br>• Los indicadores deben de tener al menos una evidencia asignada | |
| **Pasos de ejecución:**<br>1. El jefe de área, jefe de departamento o usuario autenticado accede a su plan de indicadores a cumplir<br>2. El jefe de área, jefe de departamento o usuario autenticado selecciona un indicador<br>3. El jefe de área, jefe de departamento o usuario autenticado selecciona la opción “editar” perteneciente una evidencia | |

| |
|---|
| 4. El jefe de área, jefe de departamento o usuario autenticado reemplaza los datos<br>5. Se actualiza una evidencia |
| **Resultado:** Satisfactorio |

*Tabla 153 Prueba de Aceptación HU23-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU23_P1 | **Historia de Usuario:** 23 |
| **Nombre:** Obtener evidencias | |
| **Descripción:** Prueba de funcionalidad: obtener evidencias | |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol jefe de área, jefe de departamento o usuario autenticado<br>• El jefe de área, jefe de departamento o usuario autenticado debe de tener indicadores asignados<br>• Los indicadores deben de tener al menos una evidencia asignada | |
| **Pasos de ejecución:**<br>1. El jefe de área, jefe de departamento o usuario autenticado accede a su plan de indicadores a cumplir<br>2. El jefe de área, jefe de departamento o usuario autenticado selecciona un indicador<br>3. Se obtienen las evidencias del indicador | |
| **Resultado:** Satisfactorio | |

*Tabla 154 Prueba de Aceptación HU24-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU24_P1 | **Historia de Usuario:** 24 |
| **Nombre:** Eliminar una evidencia | |
| **Descripción:** Prueba de funcionalidad: eliminar una evidencia | |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol jefe de área, jefe de departamento o usuario autenticado<br>• El jefe de área, jefe de departamento o usuario autenticado debe de tener indicadores asignados<br>• Los indicadores deben de tener al menos una evidencia asignada | |
| **Pasos de ejecución:**<br>1. El jefe de área, jefe de departamento o usuario autenticado accede a su plan de indicadores a cumplir<br>2. El jefe de área, jefe de departamento o usuario autenticado selecciona un indicador<br>3. El jefe de área, jefe de departamento o usuario autenticado selecciona la opción “eliminar” perteneciente una evidencia<br>4. El jefe de área, jefe de departamento o usuario autenticado selecciona la opción “eliminar”<br>5. Se elimina una evidencia | |
| **Resultado:** Satisfactorio | |

*Tabla 155 Prueba de Aceptación HU28-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU28_P1 | **Historia de Usuario:** 28 |
| **Nombre:** Cambio de estado de un criterio de medida automáticamente | |

| |
|---|
| **Descripción:** Prueba de funcionalidad: Pintar automáticamente de color verde, gris o violeta el ícono al que pertenece el criterio de medida dependiendo de si esta cumplido, no cumplido o sobre cumplido |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol de jefe de área, jefe de departamento o usuario autenticado<br>• El usuario debe tener uno o varios indicadores asignados<br>• Estos indicadores deben estar asociados a el criterio de medida analizado |
| **Pasos de ejecución:**<br>1. El jefe de área, jefe de departamento o usuario autenticado accede a su indicador asignado<br>2. El jefe de área, jefe de departamento o usuario autenticado cumple con uno o varios indicadores<br>3. El criterio de medida cambia su estado ha cumplido o sobre cumplido mostrando un ícono de color verde o violeta |
| **Resultado:** Satisfactorio |

*Tabla 156 Prueba de Aceptación HU29-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU29_P1 | **Historia de Usuario:** 29 |
| **Nombre:** Adjuntar archivo a una evidencia | |
| **Descripción:** Prueba de funcionalidad: adjuntar archivo a una evidencia | |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol jefe de área, jefe de departamento o usuario autenticado<br>• El jefe de área, jefe de departamento o usuario autenticado debe de tener indicadores asignados<br>• Los indicadores deben de tener al menos una evidencia asignada | |
| **Pasos de ejecución:**<br>1. El jefe de área, jefe de departamento o usuario autenticado accede a su plan de indicadores a cumplir<br>2. El jefe de área, jefe de departamento o usuario autenticado selecciona un indicador<br>3. El jefe de área, jefe de departamento o usuario autenticado selecciona la opción “editar” perteneciente una evidencia<br>4. El jefe de área, jefe de departamento o usuario autenticado selecciona la opción “elegir archivos”<br>5. El jefe de área, jefe de departamento o usuario autenticado selecciona el archivo a subir y selecciona “abrir”<br>6. Se actualiza una evidencia | |
| **Resultado:** Satisfactorio | |

*Tabla 157 Prueba de Aceptación HU30-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU30_P1 | **Historia de Usuario:** 30 |
| **Nombre:** Cambiar la contraseña de un usuario | |
| **Descripción:** Prueba de funcionalidad: cambiar la contraseña de un usuario | |
| **Condiciones de ejecución:**<br>• El usuario debe estar autenticado | |

| |
|---|
| **Pasos de ejecución:**<br>1. El usuario accede a su menú en la parte superior derecha de la pantalla<br>2. El usuario selecciona la opción “cambiar contraseña”<br>3. El usuario ingresa la contraseña actual y la nueva contraseña<br>4. El usuario selecciona la opción “aceptar”<br>5. Se cambia la contraseña |
| **Resultado:** Satisfactorio |

*Tabla 158 Prueba de Aceptación HU31-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU31_P1 | **Historia de Usuario:** 31 |
| **Nombre:** Asignar indicadores a un usuario | |
| **Descripción:** Prueba de funcionalidad: asignar indicadores a un usuario | |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol jefe de departamento<br>• Debe de haber usuarios en el sistema que pertenezcan al mismo departamento del jefe | |
| **Pasos de ejecución:**<br>1. El jefe de departamento accede a la sección de usuarios<br>2. El jefe de área selecciona la opción “analizar” que le pertenece al usuario seleccionado<br>3. El jefe de departamento selecciona la opción “establecer indicadores”<br>4. El jefe de departamento selecciona los indicadores a establecer<br>5. El jefe de departamento selecciona la opción “guardar”<br>6. El jefe de departamento establece los indicadores al usuario | |
| **Resultado:** Satisfactorio | |

*Tabla 159 Prueba de Aceptación HU32-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU32_P1 | **Historia de Usuario:** 32 |
| **Nombre:** Crear indicador personal | |
| **Descripción:** Prueba de funcionalidad: crear indicador personal | |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol jefe de área, jefe de departamento o usuario autenticado | |
| **Pasos de ejecución:**<br>1. El jefe de área, jefe de departamento o usuario autenticado accede a la sección de evaluación individual<br>2. El jefe de área, jefe de departamento o usuario autenticado selecciona la opción “añadir”<br>3. El jefe de área, jefe de departamento o usuario autenticado introduce los datos<br>4. El jefe de área, jefe de departamento o usuario autenticado selecciona la opción “aceptar”<br>5. El jefe de área, jefe de departamento o usuario autenticado creó un indicador personal | |
| **Resultado:** Satisfactorio | |

*Tabla 160 Prueba de Aceptación HU33-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU33_P1 | **Historia de Usuario:** 33 |
| **Nombre:** Cambio de estado de un indicador automáticamente | |
| **Descripción:** Prueba de funcionalidad: Pintar automáticamente de color verde o gris el ícono al que pertenece el indicador dependiendo de si esta cumplido o no | |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol de jefe de área, jefe de departamento o usuario autenticado<br>• El usuario debe tener un indicador asignado | |
| **Pasos de ejecución:**<br>1. El jefe de área, jefe de departamento o usuario autenticado accede a su indicador asignado<br>1. El jefe de área, jefe de departamento o usuario autenticado crea una evidencia asignada a dicho indicador<br>2. El indicador cambia su estado ha cumplido mostrando un ícono de color verde | |
| **Resultado:** Satisfactorio | |

*Tabla 161 Prueba de Aceptación HU34-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU34_P1 | **Historia de Usuario:** 34 |
| **Nombre:** Restablecer la contraseña de un usuario | |
| **Descripción:** Prueba de funcionalidad: restablecer la contraseña de un usuario | |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol decano<br>• Debe existir al menos un usuario en el sistema | |
| **Pasos de ejecución:**<br>1. El decano accede a la sección de los usuarios<br>2. El decano selecciona la opción “editar” perteneciente al usuario<br>3. El decano selecciona la opción “restablecer contraseña”<br>4. Se restablece la contraseña | |
| **Resultado:** Satisfactorio | |

*Tabla 162 Prueba de Aceptación HU39-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU39_P1 | **Historia de Usuario:** 39 |
| **Nombre:** Generar Informe del Área de resultados claves en PDF | |
| **Descripción:** Prueba para la funcionalidad: generar documento PDF para el Informe de un Área | |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol de jefe de área o decano. | |
| **Pasos de ejecución:**<br>1. El jefe de área o decano selecciona la opción de ver área.<br>2. Se selecciona en el menú de los tres la opción de Exportar a PDF | |
| **Resultado:** Satisfactorio | |

*Tabla 163 Prueba de Aceptación HU40-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU40_P1 | **Historia de Usuario:** 40 |
| **Nombre:** Denegar indicador de un usuario (caso negativo) | |
| **Descripción:** Prueba de funcionalidad: denegar indicador de un usuario | |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol jefe de departamento o decano<br>• Debe de haber usuarios en el sistema que pertenezcan al mismo departamento del jefe | |
| **Pasos de ejecución:**<br>1. El jefe de departamento o decano accede a la sección de usuarios<br>2. El jefe de área o decano selecciona la opción “analizar” que le pertenece al usuario seleccionado<br>3. El jefe de área o decano selecciona un indicador cumplido<br>4. El jefe de área o decano selecciona la opción “denegar”<br>5. Se mantiene el indicador cumplido en el sistema | |
| **Resultado:** No Satisfactorio | |

*Tabla 164 Prueba de Aceptación HU40-P2*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU40_P2 | **Historia de Usuario:** 40 |
| **Nombre:** Denegar indicador de un usuario (caso positivo) | |
| **Descripción:** Prueba de funcionalidad: denegar indicador de un usuario | |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol jefe de departamento o decano<br>• Debe de haber usuarios en el sistema que pertenezcan al mismo departamento del jefe | |
| **Pasos de ejecución:**<br>6. El jefe de departamento o decano accede a la sección de usuarios<br>7. El jefe de departamento o decano selecciona la opción “analizar” que le pertenece al usuario seleccionado<br>8. El jefe de departamento o decano selecciona un indicador cumplido<br>9. El jefe de departamento o decano selecciona la opción “denegar”<br>10. Se deniega el indicador | |
| **Resultado:** Satisfactorio | |

*Tabla 165 Prueba de Aceptación HU41-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU41_P1 | **Historia de Usuario:** 41 |
| **Nombre:** Añadir observación a un indicador | |
| **Descripción:** Prueba de funcionalidad: añadir observación a un indicador | |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol jefe de área, jefe de departamento o usuario autenticado<br>• El jefe de área, jefe de departamento o usuario autenticado debe de tener indicadores asignados | |

| |
|---|
| **Pasos de ejecución:**<br>1. El jefe de área, jefe de departamento o usuario autenticado accede a su plan de indicadores a cumplir<br>2. El jefe de área, jefe de departamento o usuario autenticado selecciona un indicador<br>3. El jefe de área, jefe de departamento o usuario autenticado selecciona la opción para añadir observación<br>4. El jefe de área, jefe de departamento o usuario autenticado introduce los datos<br>5. Se crea una observación |
| **Resultado:** Satisfactorio |

*Tabla 166 Prueba de Aceptación HU42-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU42_P1 | **Historia de Usuario:** 42 |
| **Nombre:** Obtener áreas de resultados clave por año | |
| **Descripción:** Prueba de funcionalidad: obtener áreas por año | |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol decano o jefe de área | |
| **Pasos de ejecución:**<br>3. El decano o jefe de área accede a la sección de las áreas<br>4. El decano o jefe de área selecciona un año<br>5. Automáticamente se obtiene todas las áreas correspondientes | |
| **Resultado:** Satisfactorio | |

*Tabla 167 Prueba de Aceptación HU43-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU43_P1 | **Historia de Usuario:** 43 |
| **Nombre:** Obtener indicadores asignados a un usuario en un año | |
| **Descripción:** Prueba de funcionalidad: obtener indicadores asignados a un usuario en un año | |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol jefe de departamento<br>• Debe de haber usuarios en el sistema que pertenezcan al mismo departamento del jefe | |
| **Pasos de ejecución:**<br>1. El jefe de departamento accede a la sección de usuarios<br>2. El jefe de departamento selecciona la opción “analizar” que le pertenece al usuario seleccionado<br>3. El jefe de departamento selecciona un año<br>4. Se obtiene los indicadores del usuario correspondiente al año seleccionado | |
| **Resultado:** Satisfactorio | |

*Tabla 168 Prueba de Aceptación HU44-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU44_P1 | **Historia de Usuario:** 44 |
| **Nombre:** Establecer fecha a una evidencia | |
| **Descripción:** Prueba de funcionalidad: establecer fecha a una evidencia | |

| |
|---|
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol jefe de departamento, jefe de área o usuario autenticado<br>• El usuario autenticado, jefe de departamento o jefe de área debe de tener indicadores asignados<br>• Los indicadores deben de tener al menos una evidencia asignada |
| **Pasos de ejecución:**<br>7. El rol jefe de departamento, jefe de área o usuario autenticado accede a su plan de indicadores a cumplir<br>8. El rol jefe de departamento, jefe de área o usuario autenticado selecciona un indicador<br>9. El rol jefe de departamento, jefe de área o usuario autenticado selecciona la opción “editar” perteneciente a una evidencia<br>10. El rol jefe de departamento, jefe de área o usuario autenticado introduce la fecha<br>11. Se actualiza la evidencia |
| **Resultado:** Satisfactorio |

*Tabla 169 Prueba de Aceptación HU45-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU45_P1 | **Historia de Usuario:** 45 |
| **Nombre:** Crear un año | |
| **Descripción:** Prueba de funcionalidad: crear un año | |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol decano | |
| **Pasos de ejecución:**<br>1. El decano accede a la sección de las áreas<br>2. El decano selecciona la opción “añadir año”<br>3. El decano introduce el año<br>4. El decano selecciona la opción para crear un año<br>5. El decano crea el año | |
| **Resultado:** Satisfactorio | |

*Tabla 170 Prueba de Aceptación HU46-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU46_P1 | **Historia de Usuario:** 46 |
| **Nombre:** Eliminar un año | |
| **Descripción:** Prueba de funcionalidad: Eliminar un año | |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol decano | |
| **Pasos de ejecución:**<br>1. El decano accede a la sección de las áreas<br>2. El decano selecciona la opción “eliminar año”<br>3. El decano selecciona el botón de aceptar | |
| **Resultado:** Satisfactorio | |

*Tabla 171 Prueba de Aceptación HU47-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU47_P1 | **Historia de Usuario:** 47 |

| |
|---|
| **Nombre:** Crear un departamento |
| **Descripción:** Prueba de funcionalidad: crear un departamento |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol decano |
| **Pasos de ejecución:**<br>5. El decano accede a la sección de las áreas<br>6. El decano selecciona la opción “gestionar departamento”<br>7. El decano introduce los datos<br>8. El decano selecciona la opción para crear un departamento<br>9. El decano crea el departamento |
| **Resultado:** Satisfactorio |

*Tabla 172 Prueba de Aceptación HU48-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU48_P1 | **Historia de Usuario:** 48 |
| **Nombre:** Eliminar un departamento | |
| **Descripción:** Prueba de funcionalidad: eliminar un departamento | |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol decano | |
| **Pasos de ejecución:**<br>1. El decano accede a la sección de las áreas<br>2. El decano selecciona la opción “gestionar departamento”<br>3. El decano selecciona la opción para eliminar un departamento correspondiente al departamento<br>4. El decano elimina el departamento | |
| **Resultado:** Satisfactorio | |

*Tabla 173 Prueba de Aceptación HU49-P1*

| **Caso de prueba de aceptación** | |
|---|---|
| **Código:** HU49_P1 | **Historia de Usuario:** 49 |
| **Nombre:** Obtener evaluación de un usuario en un año | |
| **Descripción:** Prueba de funcionalidad: obtener la evaluación de un usuario en un año determinado | |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol jefe de departamento<br>• Debe de haber usuarios en el sistema que pertenezcan al mismo departamento del jefe | |
| **Pasos de ejecución:**<br>1. El jefe de departamento accede a la sección de usuarios<br>2. El jefe de departamento selecciona la opción “evaluación” que le pertenece al usuario seleccionado<br>3. El jefe de departamento selecciona un año<br>4. Se obtiene la evaluación correspondiente al año seleccionado | |
| **Resultado:** Satisfactorio | |

*Tabla 174 Prueba de Aceptación HU50-P1*

| **Caso de prueba de aceptación** |
|---|

| **Código:** HU50_P1 | **Historia de Usuario:** 50 |
|---|---|
| **Nombre:** Evaluar a un usuario en un año | |
| **Descripción:** Prueba de funcionalidad: evaluar a un usuario en un año determinado | |
| **Condiciones de ejecución:**<br>• El usuario debe estar previamente autenticado y contar con el rol jefe de departamento<br>• Debe de haber usuarios en el sistema que pertenezcan al mismo departamento del jefe | |
| **Pasos de ejecución:**<br>7. El jefe de departamento accede a la sección de usuarios<br>8. El jefe de departamento selecciona la opción “evaluación” que le pertenece al usuario seleccionado<br>9. El jefe de departamento selecciona un año<br>10. El jefe de departamento selecciona la opción “evaluar”<br>11. El jefe de departamento selecciona su evaluación por categorías<br>12. El jefe de departamento selecciona la opción “aceptar”<br>13. El jefe de departamento evaluó a un usuario | |
| **Resultado:** Satisfactorio | |

**Anexos 6 Ilustraciones de las pruebas unitarias**

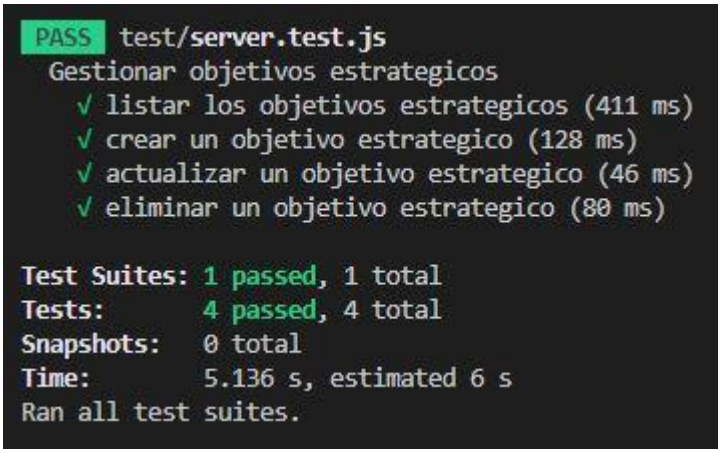



*Ilustración 13 Prueba Unitaria: Gestionar objetivos estratégicos*

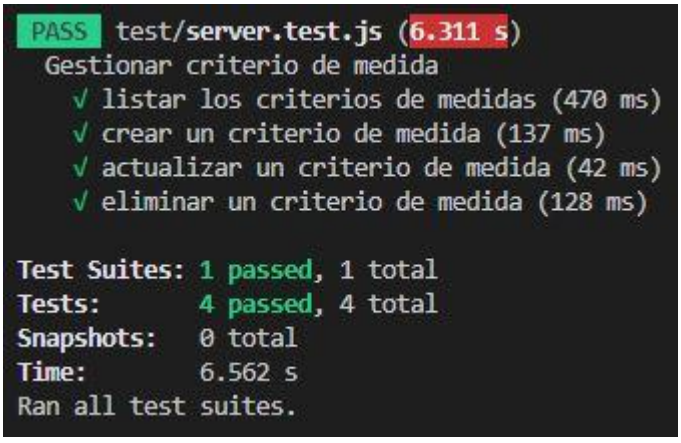



*Ilustración 14 Prueba Unitaria: Gestionar criterio de medida*

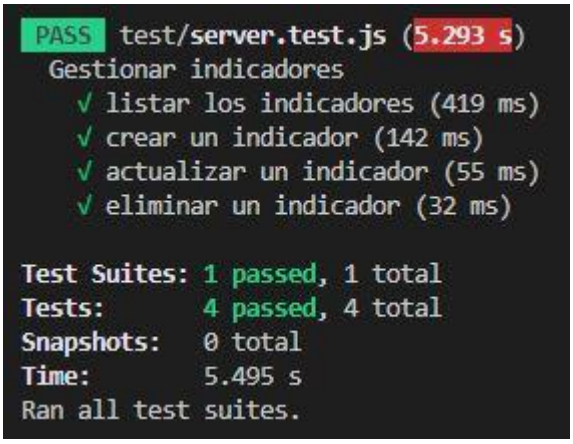



*Ilustración 15 Prueba Unitaria: Gestionar indicadores*

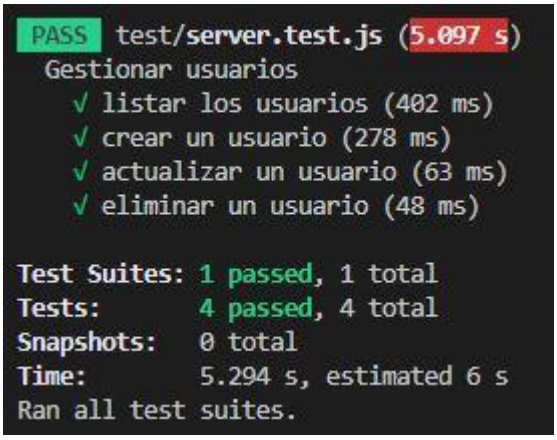



*Ilustración 16 Prueba Unitaria: Gestionar usuarios*

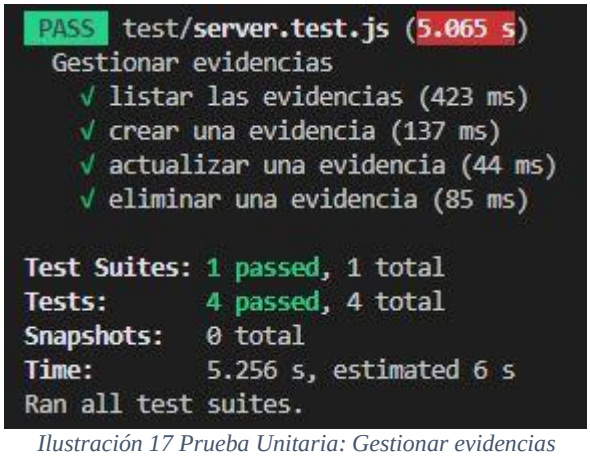



*Ilustración 17 Prueba Unitaria: Gestionar evidencias*

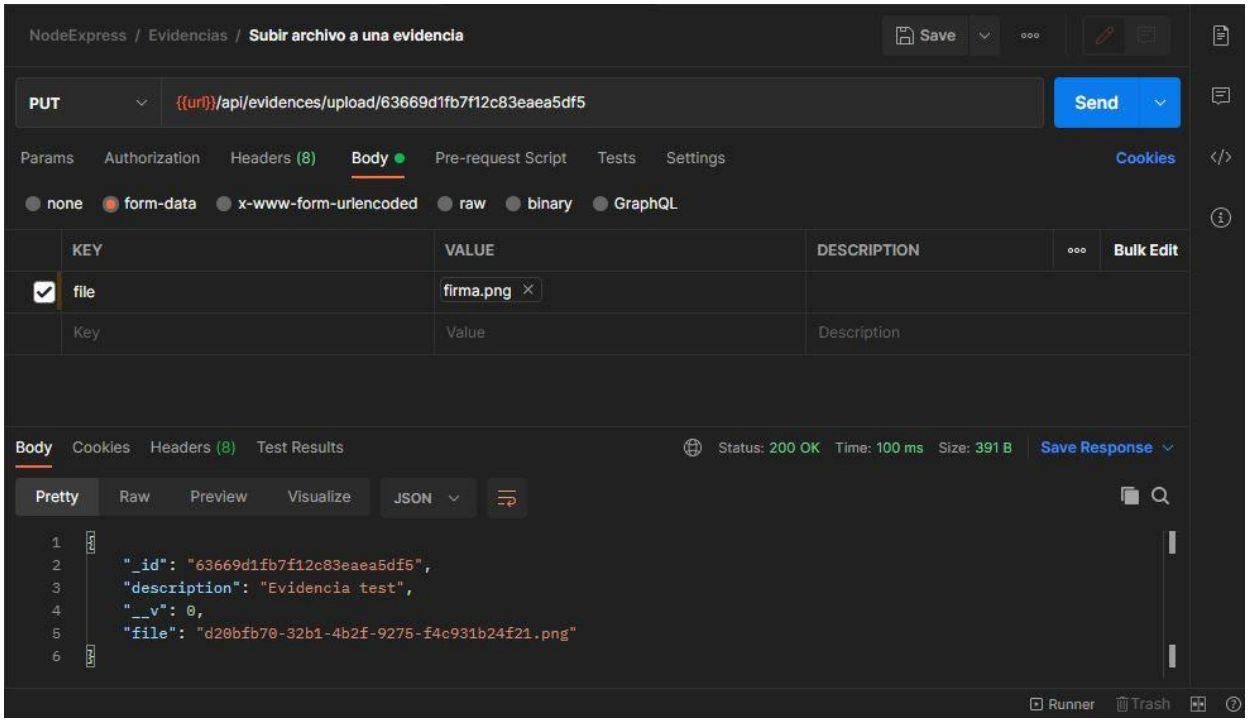



*Ilustración 18 Prueba Unitaria: Adjuntar un archivo a una evidencia*

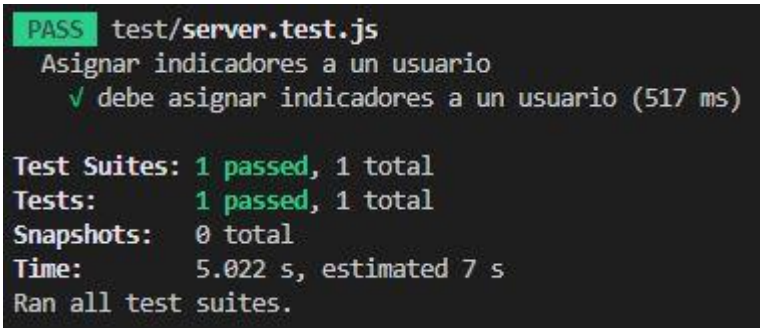



*Ilustración 19 Prueba Unitaria: Asignar indicadores a un usuario*

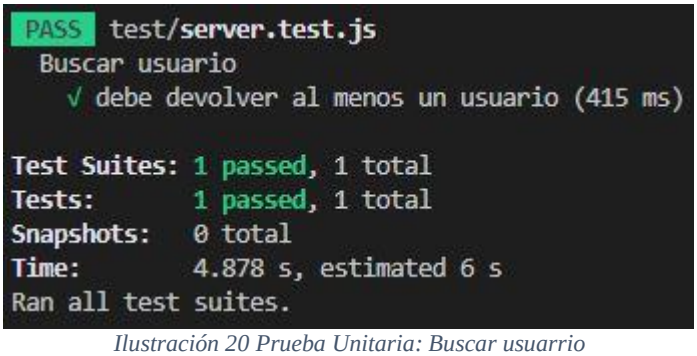



*Ilustración 20 Prueba Unitaria: Buscar usuarrio*

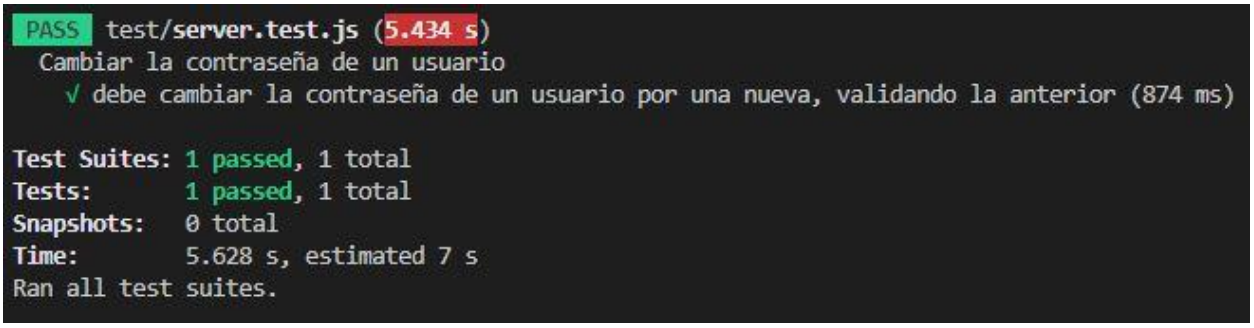



*Ilustración 21 Prueba Unitaria: Cambiar la contraseña de un usuario*

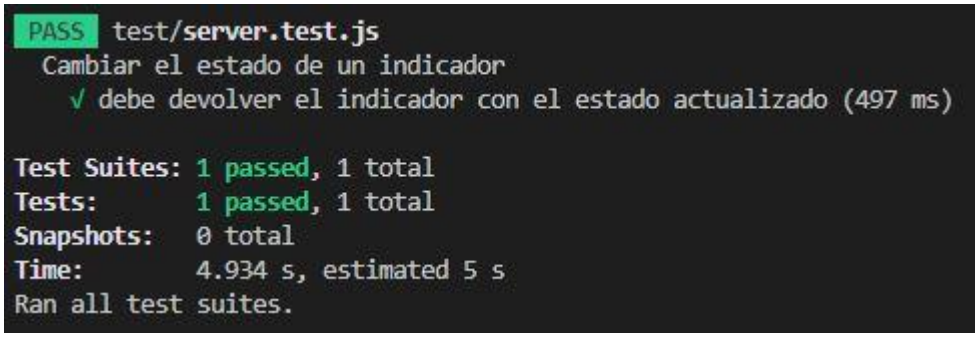



*Ilustración 22 Prueba Unitaria: Cambiar el estado de un indicador*

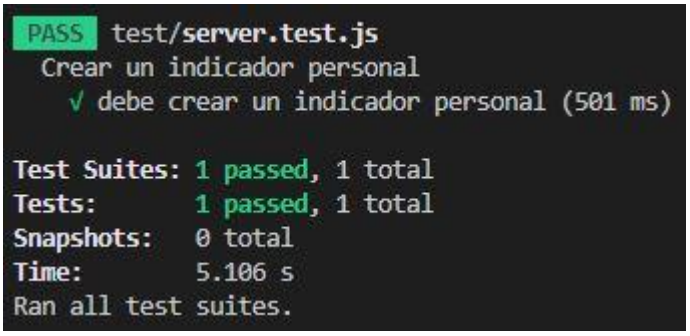



*Ilustración 23 Prueba Unitaria: Crear un indicador personal*

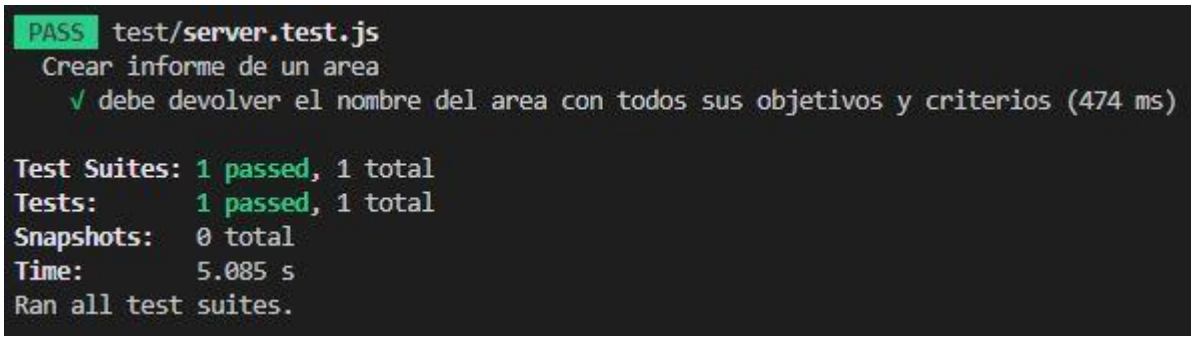



*Ilustración 24 Prueba Unitaria: Crear informe de un área*

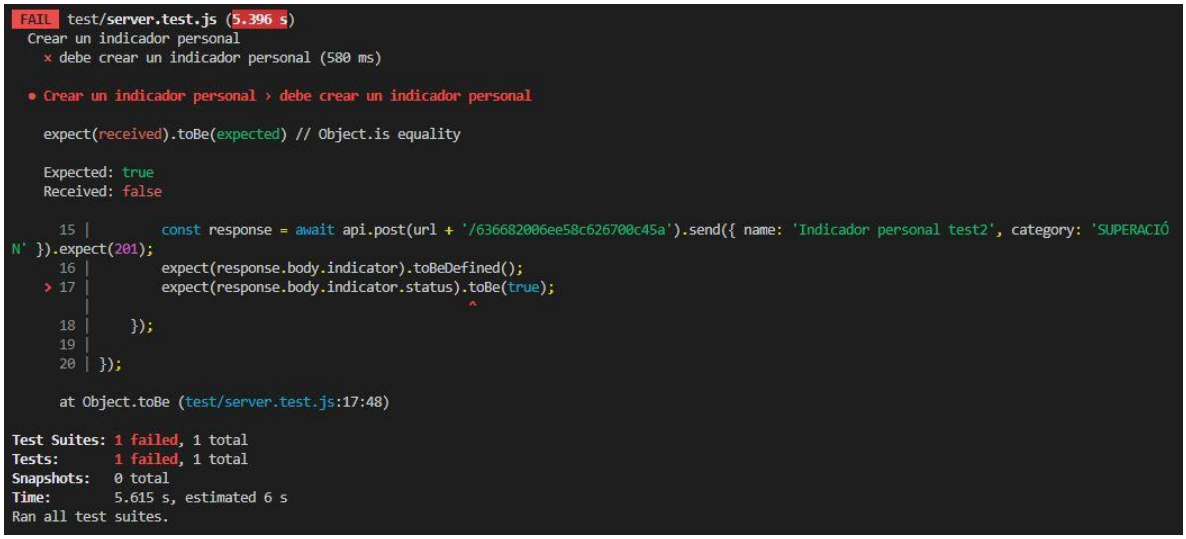



*Ilustración 25 Prueba Unitaria: Crear un indicador personal (fallido)*

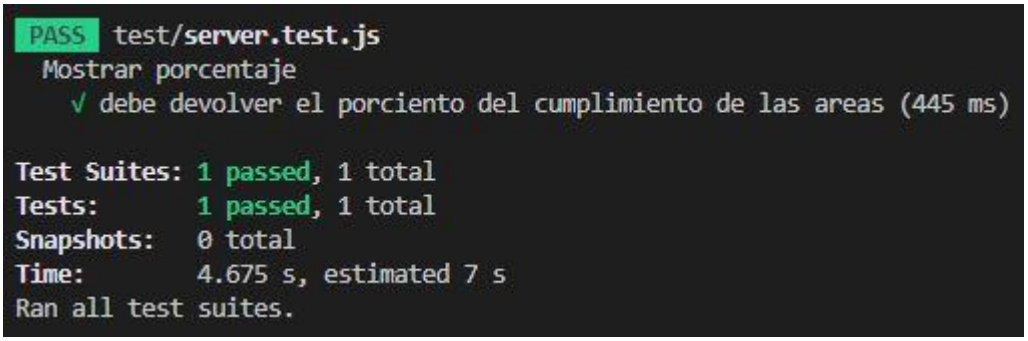



*Ilustración 26 Prueba Unitaria: Mostrar porcentaje*

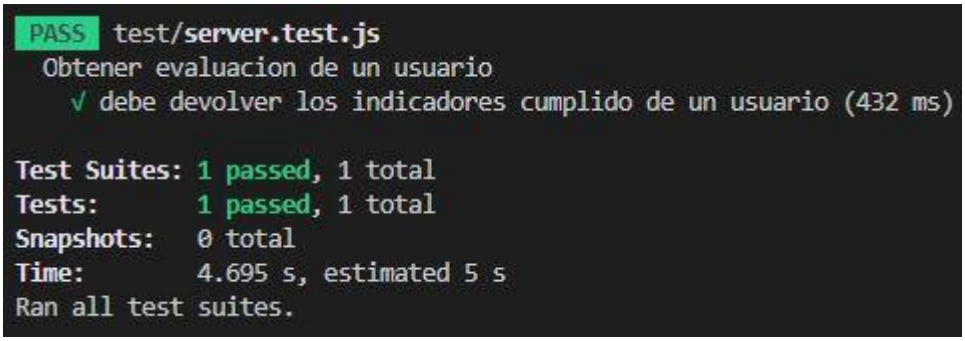



*Ilustración 27 Prueba Unitaria: Obtener evaluación de un usuario*

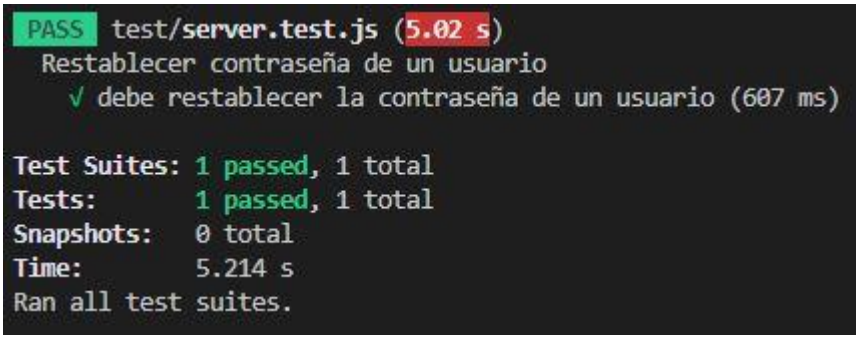



*Ilustración 28 Prueba Unitaria: Restablecer contraseña de un usuario*

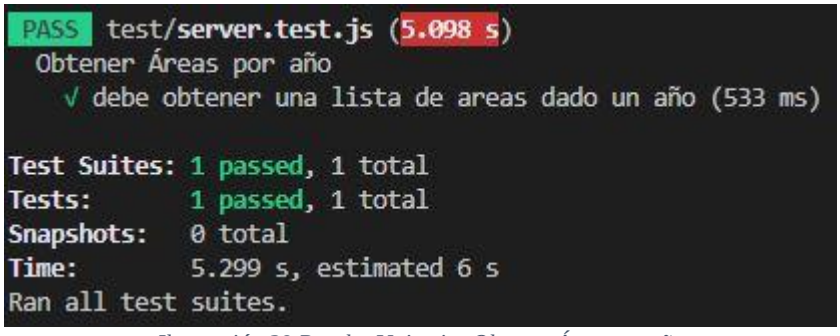



*Ilustración 29 Prueba Unitaria: Obtener Área por año*

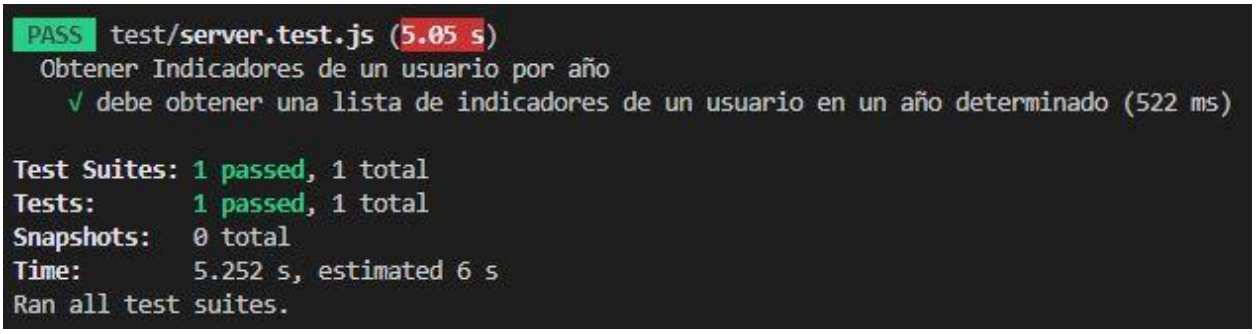



*Ilustración 30 Prueba Unitaria: Obtener Indicadores de un usuario por año*

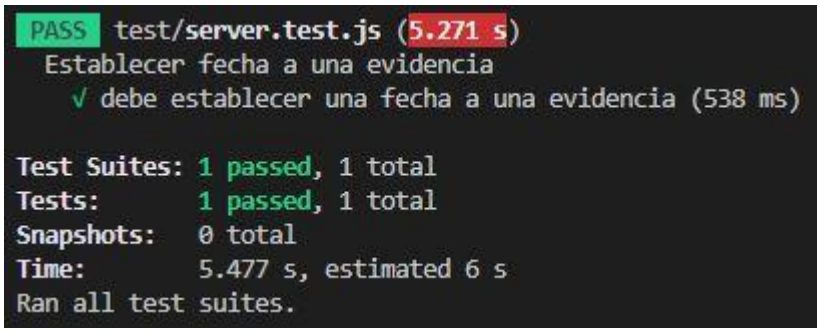



*Ilustración 31 Prueba Unitaria: Establecer fecha a una evidencia*

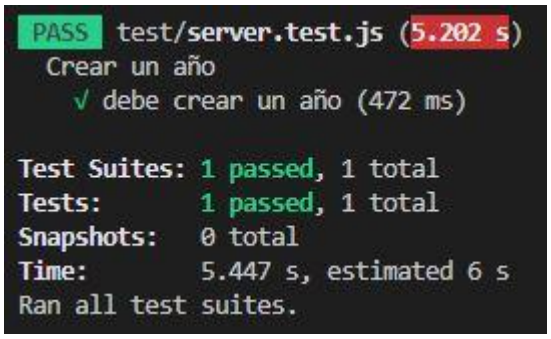



*Ilustración 32 Prueba Unitaria: Crear un año*

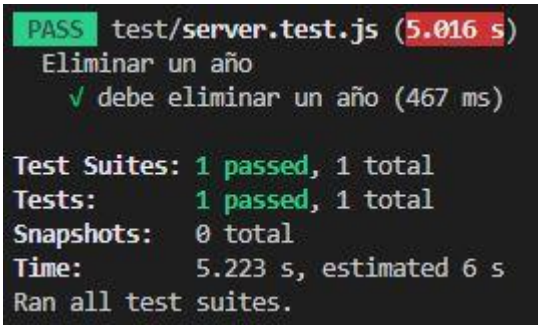



*Ilustración 33 Prueba Unitaria: Eliminar un año*

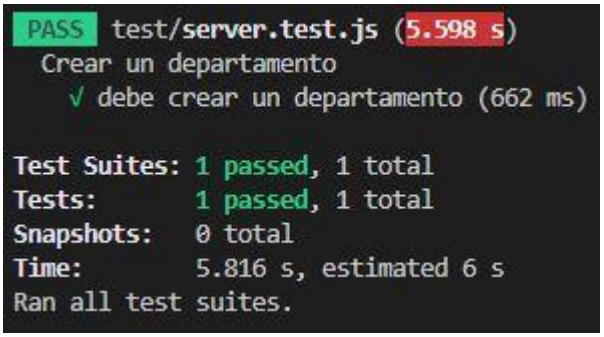



*Ilustración 34 Prueba Unitaria: Crear un departamento*

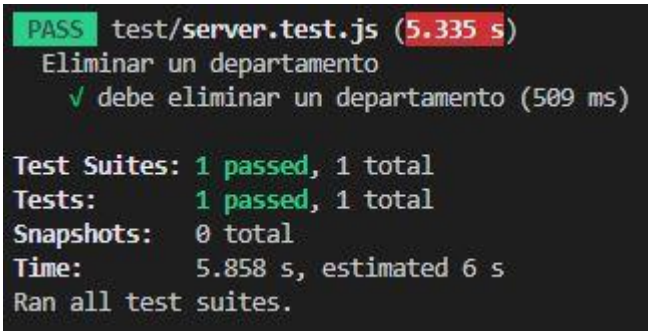



*Ilustración 35 Prueba Unitaria: Eliminar un departamento*

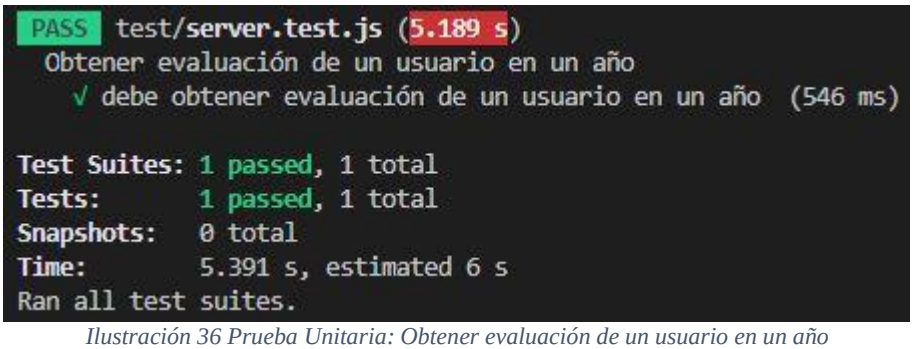



*Ilustración 36 Prueba Unitaria: Obtener evaluación de un usuario en un año*

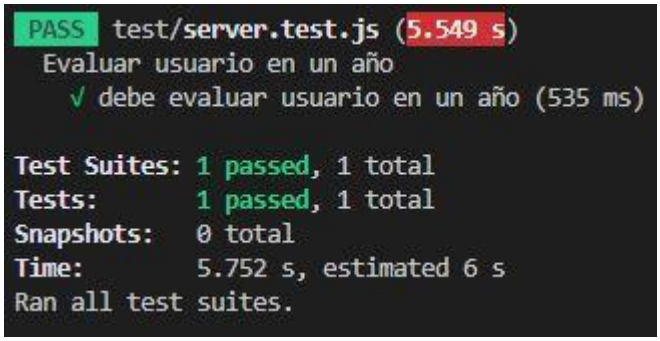



*Ilustración 37 Prueba Unitaria: Evaluar usuario en un año*

## Anexo 7 Entrevista

**Objetivo** conocer la estructura organizativa del proceso de gestión de evidencias de los objetivos estratégicos de la Facultad 4 de la UCI.

Estimado especialista:

Estamos realizando una entrevista con el propósito de conocer la estructura organizativa del proceso de gestión de evidencias de los objetivos estratégicos de la facultad 4. Le sugerimos que contribuya con nosotros.

Muchas gracias.

1. ¿Se resolverán los problemas de gestión de evidencias para cumplir los objetivos estratégicos de la Facultad 4 de la UCI con la implementación de una aplicación informática? ¿Por qué?
2. ¿Cómo ocurre actualmente la gestión de las evidencias de los objetivos estratégicos del año en la facultad?
3. ¿Para el diseño e implementación del software se necesita de alguna tecnología especifica? ¿Cuál sería la más adecuada?
4. ¿Qué problemas se presentan a la hora de realizar este proceso actualmente?

Se resolverán los problemas de gestión de evidencias para cumplir los objetivos estratégicos de la Facultad 4 de la UCI con la implementación de una aplicación informática, pues permitiría gestionar tanto la elaboración como el chequeo del cumplimiento de los objetivos estratégicos que se tracen en la facultad para un año mediante la gestión de evidencias.

Actualmente las evidencias de los indicadores se realizan a través de reuniones, despachos, y en ocasiones por medio de correos electrónicos y llamadas telefónicas.

Para el diseño e implementación del software si se necesita de alguna tecnología especifica. La más adecuada para ello sería la tecnología web ya que es la que brinda mayor flexibilidad y adaptabilidad.

Los datos obtenidos en el proceso suelen estar duplicados, desactualizados o hasta pueden existir pérdidas de información ya que no están centralizados. Esto trae consigo que el control y el monitoreo continuo del proceso se dificulte.

## Anexo 8 Acta de aceptación

Universidad de las Ciencias Informáticas | FACULTAD 4

La Habana, 15 de noviembre de 2022
"Año 64 de la Revolución"

**CARTA DE ACEPTACIÓN**

Mediante la siguiente, hacemos constar que los estudiantes Arián León Benítez y Javier Ceballo Perez, en cumplimiento del desarrollo de la tesis **Sistema de gestión para las evidencias de los objetivos del año,** hacen entrega del producto informático (aplicación web) **SGEOA.**

El cliente ha revisado y probado el software, por lo que pudo comprobar que los requerimientos del software se implementaron en su totalidad (El sistema fue instalado en el Departamento de Tecnología y el Departamento de Informática). El producto entregado, permitirá sin duda alguna, gestionar la elaboración y chequear el cumplimiento de los objetivos estratégicos que se tracen en la facultad para un año determinado mediante la gestión de evidencias.

Conforme a lo anterior se expresa la aceptación del producto entregado.

M.Sc. Yordankis Matos López
Jefe de Departamento de Tecnología

M.Sc. Yadira Ramírez Rodríguez
Jefa de Departamento de Informática