

Tipo de artículo: Artículo de revisión

Temática: Ingeniería y gestión de software

Recibido: 20/09/19 | Aceptado: 14/01/2020 | Publicado: 06/01/2020

## Diseño de herramienta para casos de pruebas funcionales en la Universidad de las Ciencias Informáticas

### Tool design for functional test cases in the University of Computer Sciences

Yadelis Velázquez Godoy<sup>1\*</sup>, Alionuska Velázquez Cintra<sup>2</sup>, Lester Collado Rolo<sup>3</sup>

<sup>1</sup> Departamento de Informática. Facultad 3. Universidad de las Ciencias Informáticas. Carretera a San Antonio de los Baños, Km 2 ½. Torrens Boyero. La Habana, Cuba. [yvelazquez@estudiantes.uci.cu](mailto:yvelazquez@estudiantes.uci.cu)

<sup>2</sup> Departamento de Informática. Facultad 3. Universidad de las Ciencias Informáticas. Carretera a San Antonio de los Baños, Km 2 ½. Torrens Boyero. La Habana, Cuba. [avelazquez@uci.cu](mailto:avelazquez@uci.cu)

<sup>3</sup> Centro de Tecnologías de Datos. Facultad de Ciencias y Tecnologías Computacionales. Universidad de las Ciencias Informáticas. Carretera a San Antonio de los Baños, Km 2 ½. Torrens Boyero. La Habana, Cuba. [lcollado@uci.cu](mailto:lcollado@uci.cu)

\* Autor para correspondencia: [yvelazquez@estudiantes.uci.cu](mailto:yvelazquez@estudiantes.uci.cu)

---

#### Resumen:

Las empresas productoras de software, como resultado de la fuerte demanda de sus productos y alta competencia, han estado inmersas en mercados muy cambiantes. Es crucial verificar y evaluar la calidad de las aplicaciones y, en este sentido, las pruebas de software son esenciales en el control de la calidad, si se quiere entregar al cliente un producto de calidad. La dinámica impuesta por el desarrollo tecnológico implica acortar los tiempos de prueba sin afectar la efectividad de las mismas. Ello ha provocado que ingenieros y probadores presten mayor atención al diseño de las mismas, como actividad fundamental para una adecuada ejecución de las pruebas. Automatizar los escenarios identificados en el desarrollo de estas actividades ayuda a reducir el riesgo de complicaciones durante las operaciones. La presente investigación muestra el diseño de una herramienta para el diseño y gestión de los casos de prueba funcionales en la Universidad de las Ciencias Informáticas (UCI).

**Palabras Clave:** automatización, calidad, casos de prueba, pruebas de software.

#### Abstract:

*Software producing companies, as a result of strong demand for their products and high competition, have been immersed in very changing markets. It is crucial to verify and evaluate the quality of applications and, in this sense, software testing is essential in quality control, if you want to deliver a quality product to the customer. In recent years, software testing has become an important tool for quality assurance of processes and products during the software development process. The dynamics imposed by technological development implies shortening test times without affecting their effectiveness. This has caused engineers and testers to pay more attention to their design, as a fundamental activity for the proper execution of the tests. Automate the scenarios identified in the development of these activities helps reduce the risk of complications during operations. This research shows the design of a tool for the design and management of functional test cases at the University of Computer Science (UCI).*

**KeyWords:** automation, quality, test cases, test software.

## **INTRODUCCIÓN**

Las Tecnologías de la Información y las Comunicaciones (TIC's) agregan valor a las actividades operacionales y de gestión empresarial en general y permiten a las empresas obtener ventajas competitivas, permanecer en el mercado y centrarse en su negocio [1]. Para obtener un resultado satisfactorio en la producción de software es imprescindible que los productos de software se desarrollen con calidad. Es por ello que las pruebas se han convertido en la actividad fundamental para su control, asumiendo un papel protagónico en el que, necesariamente, se ha hecho extensivo la creación de entidades dedicadas a probar software [2].

La industria está viviendo rápidas evoluciones en relación con los procesos, tecnologías, herramientas y nuevas tendencias. En el segmento del Aseguramiento de la Calidad (QA) y Testing del software nos enfrentamos a nuevos desafíos como la inteligencia artificial, reducción de tiempos de desarrollo o nuevas metodologías de trabajo [3].

Varios autores [3] [4] confirman la automatización de las pruebas como tendencia siendo posible reorientar esfuerzos hacia las áreas que representan la prioridad para alcanzar la calidad deseada en el producto. De este modo, las empresas reducen el número de pruebas manuales y consiguen recortar en tiempos de producción y en costos [3]. En este escenario el diseño de las pruebas adquiere mayor relevancia marcada por el creciente uso de las tecnologías emergentes.

La UCI cuenta con el Laboratorio de Pruebas de Software (LPS) el cual tiene como objetivo la verificación de los productos desarrollados por la Red de Centros de la Universidad, siendo esta la actividad de control de la calidad que tiene lugar antes de la instalación del producto en el entorno de uso. Se considera, por especialistas involucrados en los procesos de evaluación de software, que el tiempo dedicado a las tareas de diseño y gestión de casos de pruebas funcionales demanda mayor esfuerzo que el planificado en el cronograma de desarrollo del producto. A partir de la actual aceleración respecto a la evolución tecnológica y, con el objetivo de mantener los

índices de aceptación alcanzados por los productos desarrollados en la UCI y sentar las bases para asumir los retos que presupone hoy el desarrollo de software, es necesario disminuir el esfuerzo que hoy representa la ejecución de los procesos de pruebas. Al mismo tiempo. La presente investigación como objetivo diseñar una herramienta para el diseño y gestión de casos de pruebas funcionales en La Universidad de las Ciencias Informáticas (UCI).

## **MATERIALES Y MÉTODOS**

Se realizó un estudio de todas las técnicas de diseño de pruebas utilizadas, principalmente en las técnicas partición de equivalencia o clases de equivalencia (CE) y valores límites. Se analizaron las características del proceso de pruebas de software en la Universidad de las Ciencias Informáticas. Se estudiaron y compararon las herramientas que son referentes en el diseño de casos de pruebas funcionales.

### ***Las pruebas de software***

Las pruebas de software son un conjunto de herramientas, técnicas y métodos que evalúan la excelencia y el desempeño de un software. Las técnicas para encontrar problemas en un software son extensamente variadas y van desde el uso del ingenio por parte del personal ejecutor de las pruebas hasta herramientas automatizadas que ayudan a aliviar el peso y el costo de tiempo de esta actividad [1].

Estas, son seguramente la actividad más común de control de calidad realizada en los proyectos de desarrollo o mantenimiento de aplicaciones y sistemas [2]. Se vuelven más críticas en dependencia del impacto social del producto. Específicamente durante las pruebas el principal interés es mejorar la calidad del producto y para ello es necesario que todos los implicados estén a gusto con la actividad que desarrollan y el resultado final [3].

En los proyectos de desarrollo de software, las actividades de control de la calidad en su conjunto, constituyen tareas que permiten prevenir, controlar y mejorar la calidad del producto. Si el objetivo que se persigue es entregar al cliente un producto de calidad, según los parámetros establecidos por él mismo, las pruebas de software son fundamentales como actividad de control.

En un proceso de pruebas de software el conocimiento sobre el dominio de aplicación y las técnicas de prueba, así como las experiencias personales, se pueden utilizar para guiar el diseño de casos de prueba, y para reconocer los fallos. En una aproximación exploratoria de las pruebas de software, donde los casos de prueba no están definidos de antemano, el conocimiento del probador es crucial. En este caso, el conocimiento junto con el comportamiento real observado del sistema ensayado durante la prueba exploratoria se puede utilizar para crear nuevas y mejores pruebas [4] [5].

Garrepalli explica cómo varios autores coinciden en la importancia de la Gestión del Conocimiento (GC) para la selección y ejecución de técnicas de pruebas de software, que dependen en gran medida de la experiencia que tengan los probadores [5].

De manera general, la integración de las buenas prácticas de GC en el desarrollo de las actividades de pruebas resulta beneficiosa para la calidad del producto final. Durante las pruebas de software la GC individual y organizacional es fundamental a partir del papel predominante que desempeñan las personas involucradas y su expe-

riencia.

### ***Diseño de pruebas de software***

Las pruebas de software son una actividad primordial en el proceso de aseguramiento de la calidad. El conjunto de actividades de pruebas dentro del proceso de desarrollo de software, son conocidas como proceso básico de pruebas, el cual incluye [6]:

- Planeación
- Análisis y diseño de pruebas
- Ejecución de pruebas
- Evaluación de resultados
- Cierre de pruebas
- Seguimiento y control

En la etapa de análisis y diseño de pruebas se analizan los requisitos y se diseñan los casos de prueba, lo cual incluye casos positivos, casos negativos y otros elementos definidos en la IEEE 829. A la salida de esta fase se tendrán los casos de prueba.

El diseño puede comenzar una vez que las condiciones de prueba se identifiquen con datos suficientes para producir casos de prueba de alto o bajo nivel. Para un nivel de prueba dado, el diseño de la prueba puede comenzar una vez que se identifican las condiciones de la prueba y hay suficiente información disponible para escribir casos de prueba.

En el Plan de estudios para la certificación internacional del rol probador, el ISTQB recomienda las técnicas de pruebas que pueden utilizarse en los diferentes enfoques: dinámico y estático. En la figura 1 puede observarse la representación de estas técnicas para cada caso.



**Figura 1:** Definición de técnicas para el diseño de las pruebas [7]

Las pruebas funcionales están dirigidas a verificar la corrección y la completitud de una función. Dichas pruebas utilizan para su diseño las técnicas basadas en la especificación o técnicas de caja negra la cual se realiza mediante los siguientes métodos [10]:

- Partición de equivalencia (segmentación de equivalencia) o clase de equivalencia: es un método que divide el campo de entrada de un programa en clases de datos. La prueba de partición equivalente se basa en evaluar las clases de equivalencia para una condición de entrada [7] [9].
- Análisis de valores límite: parte del método de clases de equivalencia, su diferencia radica en el modo de generar los casos prueba, en este caso más que seleccionar cualquier elemento como representativo de la clase de equivalencia, el análisis de valor límite requiere que se seleccionen uno o más elementos de forma tal que cada margen de la clase de equivalencia sea sujeto a una prueba [7] [10].
- Tablas de decisión & gráficos causa-y-efecto: sintetiza procesos en los cuales se dan un conjunto de condiciones y un conjunto de acciones a tomar según el valor que toman las condiciones. Puede utilizarse como herramienta en los distintos momentos del proyecto, esto es: en la exposición de los hechos, en el análisis del sistema actual, en el diseño del nuevo sistema y en el desarrollo del software [11] [7].
- Pruebas de transición de estado: deben ejecutar todas las funciones de un estado al menos una vez. Pueden diseñarse pruebas para cubrir una secuencia típica de estados, cubrir todos los estados, ejercitar secuencias específicas de transiciones o probar transiciones inválidas [7].
- Pruebas de caso de uso: describen las interacciones entre actores (que pueden ser usuarios o sistemas) que producen un resultado que agrega algún valor. A partir de estos se pueden derivar casos de prueba [12].

El propósito principal de las pruebas de caja negra es probar la funcionalidad del sistema. Esto hace que el cum-

plimiento del objetivo de la prueba dependa de la calidad del diseño de las mismas. Los autores de la presente investigación apuestan por la combinación de varias técnicas para obtener los diseños de casos de prueba. Además, estos métodos pueden combinarse con los basados en la experiencia que utilizan el conocimiento y la habilidad del personal involucrado en el diseño de las pruebas.

### ***Herramientas que automatizan el diseño de casos de pruebas funcionales***

Todos los proyectos, por muy pequeños que sean, pueden llegar a tener una cantidad muy elevada de casos de pruebas. Considerando, además que las pruebas se repetirán varias veces debido a las pruebas de regresión, estos proyectos, necesitan de una administración, planificación y ejecución, así como de herramientas que permitan realizar pruebas automáticas. Para llevar a cabo estas tareas existen diferentes tipos de herramientas que ayudarán en todo lo posible a que el proyecto se maneje más eficientemente y que ayudarán a conseguir la calidad deseada.

Existen herramientas que se utilizan para diseñar casos de prueba, gestionar y administrar pruebas y monitorizar sistemas en pruebas. Una simple hoja de datos puede ser considerada una herramienta de pruebas. Las herramientas de gestión ayudan al probador a documentar y evaluar los casos de pruebas. Tienen como objetivo proporcionar mecanismos que permitan realizar de una manera controlada la documentación, el mantenimiento de las pruebas y la gestión de resultados [13].

Para el desarrollo de la investigación de analizaron varias herramientas que se usan para el diseño y gestión de los casos de pruebas funcionales. En la siguiente tabla se exponen los criterios analizados y resultados obtenidos para cada herramienta:

**Tabla 1:** Resumen de herramientas que diseñan y gestionan casos de pruebas funcionales [Elaboración propia]

Nombre de la Herramienta	Características
Teslink (libre) <sup>1</sup>	Es una herramienta que permite crear y gestionar casos de prueba y organizarlos dentro de planes de prueba. Se pueden ejecutar los casos de prueba a partir de los planes creados en la misma herramienta. También permite la generación de informes, así como priorizar y asignar tareas
Redmine (libre) <sup>2</sup>	Es una aplicación para gestión y planificación de proyectos con interfaz web. Está diseñada para facilitar el control y seguimiento de proyectos
HP Quality Center (pago por licencia) <sup>3</sup>	El objetivo de esta herramienta es el control de la calidad de software y tiene varios módulos que permitirán gestionar los requisitos de un proyecto, la gestión, el diseño y la creación de pruebas y la gestión de incidentes

<sup>1</sup> <http://testlink.org/>

<sup>2</sup> <http://www.redmine.org/>

<sup>3</sup> <http://www8.hp.com/es/es/software-solutions/quality-center-quality-management/>

	cias.
Mantis (libre) <sup>4</sup>	Es una herramienta para gestionar incidencias. Permite llevar un control y mantener un historial de las incidencias, así como especificar un número indeterminado de opciones de las incidencias, como los estados de éstas (abierta, cerrada, resuelta, reabierta, etc.) la severidad (baja, media, alta), etc. Es una herramienta con interfaz web.

En dichas herramientas no existen algunas funcionalidades que son vitales a tener en cuenta para la propuesta de solución durante las pruebas, tales como:

- La generación de casos de pruebas a partir de los requisitos especificados.
- Exportar los diferentes productos de trabajo en formato Excel, Word y PDF (GESPRO y Excriba).

### ***Pruebas de software funcionales en la UCI***

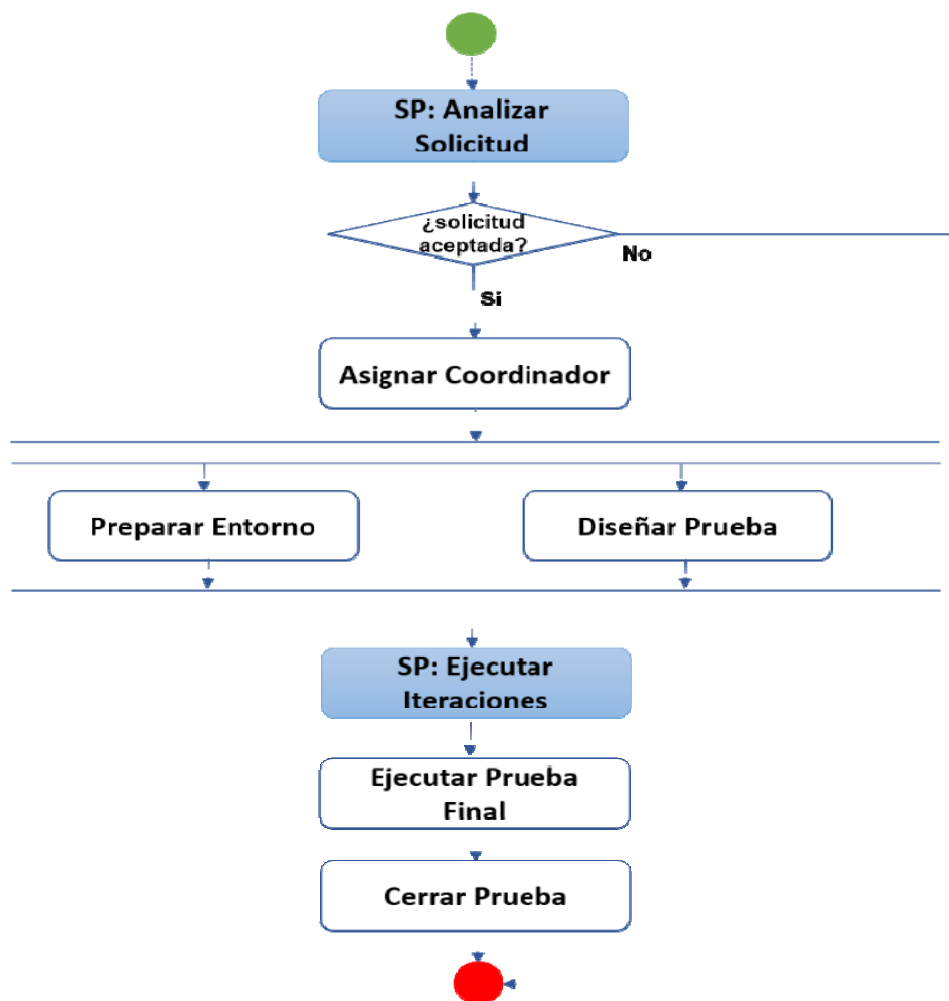
La principal misión de la UCI es formar profesionales comprometidos con su patria y altamente calificados en la rama de la informática, producir aplicaciones y servicios informáticos a partir del vínculo estudio-trabajo como modelo de formación-investigación-producción, sirviendo de soporte a la industria cubana de la informática.

Las pruebas de software en la UCI son guiadas por especialistas que funcionan como Coordinadores bajo un modelo de pruebas que combina las políticas de la actividad desarrollo-producción de la UCI con las características de calidad del producto definidas en la NC ISO/IEC 25010:2016 y las buenas prácticas de la industria [2].

El proceso de pruebas de software que tiene lugar en el LPS de la UCI se divide en 4 etapas: planificación, análisis, ejecución y evaluación de las mismas. En la figura 2 se puede observar la representación del proceso general de las pruebas.

---

<sup>4</sup> <http://www.mantisbt.org/>



**Figura. 2:** Proceso de Pruebas de Liberación del LPS de la UCI [3]

Se ha reconocido, en encuestas y entrevistas realizadas a coordinadores de pruebas, probadores y especialistas de los equipos de desarrollo (analistas en su mayoría) que intervienen en estos procesos, que las actividades desarrolladas durante la etapa de análisis son críticas para lograr una adecuada ejecución de las pruebas. A continuación, se muestran los resultados:

- Todos señalaron que la especificación de requisitos de software influye en la calidad de etapas posteriores del desarrollo del producto, identificaron además la utilidad de verificar la correspondencia entre requisitos y casos de prueba (CP), antes de ejecutar las pruebas.
- El 100 % de los analistas considera que el tiempo que emplea en el completamiento de la información relacionada con el diseño de casos de prueba es mucho. Diez de ellos señalan la automatización como solución a este problema.
- Todos los coordinadores de prueba coinciden en que el proceso de verificar la correspondencia entre los requisitos y los casos de prueba es complejo y 6 de ellos ubican la automatización de esta actividad como prioridad 1.



Por otro lado, los especialistas de proyectos vinculados a procesos de pruebas, en todos los casos, expresaron que era muy complejo mantener actualizados los casos de prueba, pues es un proceso que se realiza manualmente.

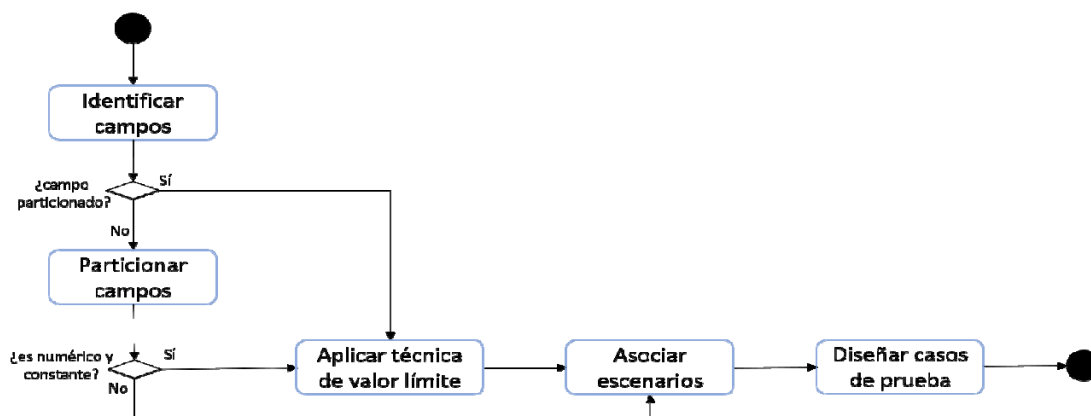
El proceso de desarrollo de software en la Universidad se encuentra certificado por CMMI - nivel 2. Actualmente este proceso transita por un período de adecuación a partir de lo establecido por el nivel 3 de CMMI para el área de Verificación [2]. Sus actividades principales se gestionan a través de la herramienta GESPRO y esta tiene incorporadas funcionalidades que permiten gestionar el listado de requisitos. Sin embargo, la herramienta no permite diseñar los casos de prueba a partir de los requisitos, comprobar la cobertura de pruebas alcanzada con el diseño realizado ni realizar trazabilidad de los defectos detectados a los requisitos, considerando las condiciones de pruebas que permitieron llegar a estos resultados.

Estos resultados ratifican la necesidad real de automatizar el proceso de diseño de casos de pruebas funcionales considerando los elementos propios del entorno de uso de la futura herramienta: proceso de desarrollo certificado y presencia de GESPRO, sin que se afecte la calidad alcanzada por el proceso de pruebas.

### *Diseño de herramienta para la automatización de casos de pruebas funcionales en la UCI*

Para el diseño de la propuesta se consideraron las buenas prácticas de la GC enfocadas al diseño de casos de prueba. Respecto al uso de las técnicas de caja negra para el diseño de las pruebas debe señalarse que el diseño de casos de pruebas descansa en las técnicas partición equivalente y análisis de valores límites, además analistas, coordinadores y probadores podrán hacer uso de las técnicas, tablas de decisión y transición de estado y las pre-condiciones y post-condiciones de los casos de uso (en los sistemas cuya metodología los incluyan como productos de trabajo).

La propuesta que se realiza incluye el desarrollo de una herramienta que permita realizar particiones a campos específicos y guardar los resultados para que puedan ser reutilizados en diferentes escenarios de prueba. En la figura 3 se muestra la representación de este proceso.



**Figura. 3:** Automatización de casos de pruebas funcionales [Elaboración propia]

### *Funcionalidades principales:*

A partir del estudio de las herramientas para gestionar casos de prueba, realizado anteriormente se identificaron

varias funcionalidades divididas en dos módulos que son vitales para el uso de un sistema con estas características en dicho proceso, además se identificaron algunas funcionalidades que responden a características propias del uso del sistema:

**Módulo Administración:**

- Registrar de usuarios
- Autenticar usuario
- Recuperar contraseña
- Mostrar trazas
- Cerrar sesión
- Gestionar usuario (Crear, Eliminar, Modificar, Listar)
- Gestionar proyectos (Crear, Eliminar, Modificar, Listar)
- Gestionar rol (Crear, Eliminar, Modificar, Listar)
- Asignar rol

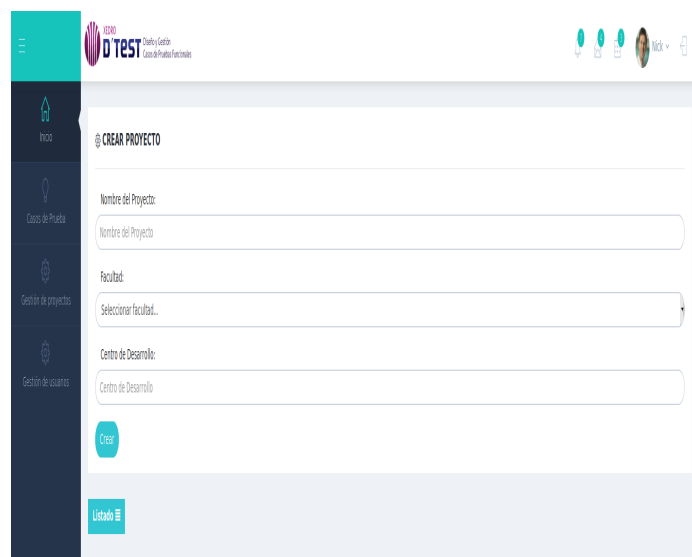
**Módulo Diseño de Casos de Pruebas Funcionales:**

- Crear particiones de un campo
- Listar campos especificados
- Gestionar CP concreto (alto nivel) (Crear, Eliminar, Modificar, Listar)
- Gestionar CP detallado (bajo nivel) (Crear, Eliminar, Modificar, Listar)
- Exportar CP

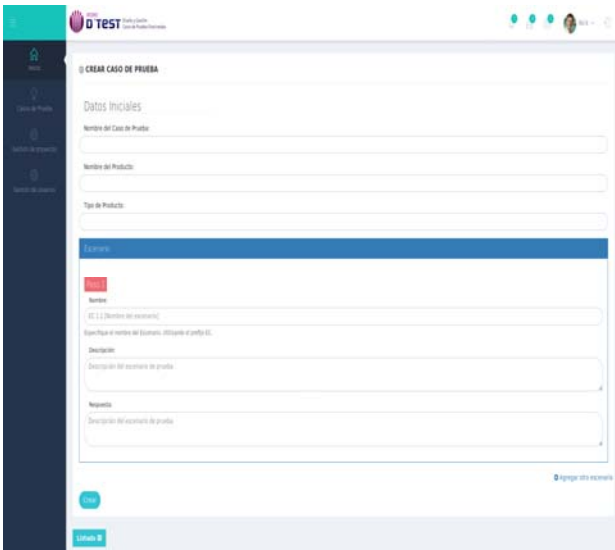
A continuación, se muestran algunas capturas de pantalla de la herramienta:



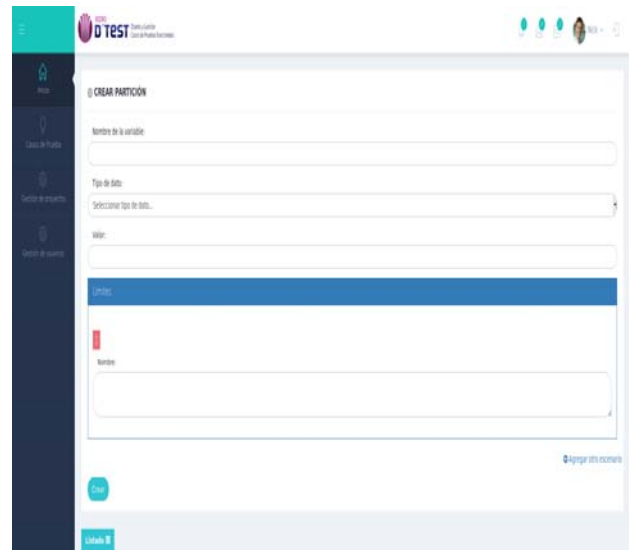
**Figura 4:** Captura del RF Autenticar Usuario



**Figura 5:** Captura del RF Gestionar proyecto



**Figura 6:** Captura RF Gestionar Caso de Prueba



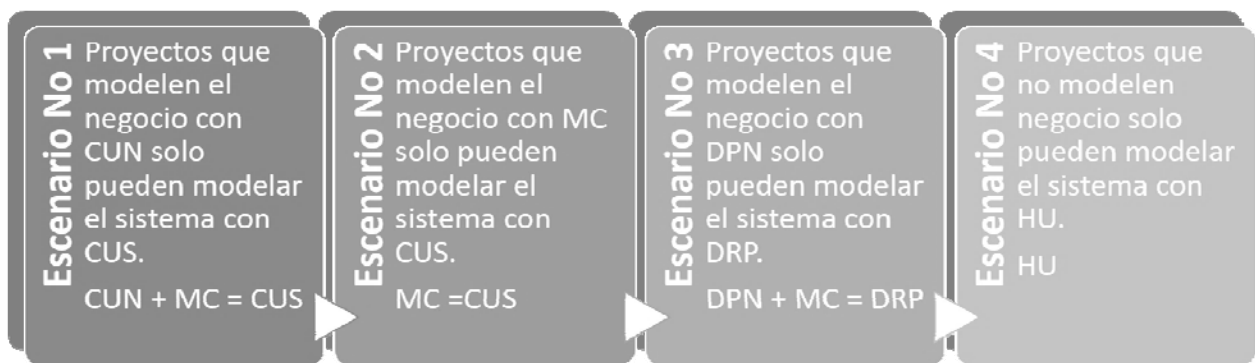
**Figura 7:** Captura del RF Crear partición

**Herramientas y tecnologías:**

**Metodología AUP-UCI**

Para el ciclo de vida de los proyectos de la UCI se utiliza una variante de la metodología AUP llamada AUP-UCI. AUP propone cuatro fases (inicio, elaboración, construcción, transición), pero modificando el objetivo de la misma, se unifican las restantes tres fases de AUP en una sola, a la que se llamó ejecución y se agrega la fase de cierre.

En la variación de la metodología AUP-UCI, existen tres formas de encapsular los requisitos Casos de Uso del Sistema (CUS), Historias de usuario y Descripción de requerimientos por proceso (DRP), agrupados en cuatro escenarios, quedando como se muestra en la **¡Error! No se encuentra el origen de la referencia..**



**Figura 8:** Escenarios metodología AUP-UCI según el Programa de Mejora [Elaboración propia]

Teniendo en cuenta las características antes mencionadas, la necesidad de una metodología que responda con facilidad a los cambios continuos y siguiendo las políticas de desarrollo de software de la UCI, se tuvo en cuenta en el desarrollo de la ingeniería de requisitos el sistema propuesto, lo planteado en la metodología AUP-UCI,

encapsulando los requerimientos a través del escenario cuatro. Para la selección de este escenario se tuvo en cuenta que el cliente estará siempre acompañando al equipo de desarrollo para convenir los detalles de los requerimientos y así poder implementarlos, probarlos y validarlos. El tratamiento de las historias de usuario es muy dinámico y flexible, lo que permite que estas puedan reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas de una manera más fácil, sin invertir tanto tiempo y esfuerzo.

### ***Lenguaje de programación del lado del cliente***

Los lenguajes de programación del lado del cliente son aquellos que pueden ser directamente interpretados por el navegador y que no necesitan un pre tratamiento. La programación del lado del cliente tiene como principal ventaja que la ejecución de la aplicación se delega al cliente, con lo cual se evita recargar al servidor de trabajo. El servidor solo envía el código y es tarea del navegador interpretarlo<sup>5</sup>.

#### *Java Script*

Es un lenguaje ligero e interpretado, orientado a objetos con funciones de primera clase, más conocido como el lenguaje de script para páginas web, pero también usado en muchos entornos sin navegador, tales como node.js, Apache CouchDB y Adobe Acrobat. Es un lenguaje script multi-paradigma, basado en prototipos, dinámico, soporta estilos de programación funcional, orientada a objetos e imperativa<sup>6</sup>.

#### *HTML*

Lenguaje de Marcado para Hipertextos (HyperText Markup Language) es el bloque de construcción más básico de una página web que se usa para crear y representarla visualmente. Determina el contenido de la misma, pero no su funcionalidad. Básicamente se trata de un conjunto de etiquetas que sirven para definir el texto y otros elementos que compondrán una página web, como imágenes, listas, vídeos<sup>7</sup>.

#### *CSS*

Las Hojas de Estilo en Cascada (Cascading Style Sheets), es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos<sup>8</sup>.

En el desarrollo de la propuesta de solución se utilizará como lenguajes de programación del lado del cliente Java Script, HTML y CSS.

### ***Lenguaje de programación del lado del servidor***

Se denominan lenguajes del lado del servidor a aquellos que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él. Es una tecnología que consiste en el procesamiento de una petición de un usuario mediante la interpretación de un script en el servidor web para

---

<sup>5</sup> <https://programacionwebisc.wordpress.com/2-2-lenguajesde-programacion-del-lado-del-cliente/>

<sup>6</sup> <https://developer.mozilla.org/es/docs/Web/JavaScript>

<sup>7</sup> [https://docs.webplatform.org/wiki/guides/the\\_basics\\_of\\_html](https://docs.webplatform.org/wiki/guides/the_basics_of_html)

<sup>8</sup> <http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo>

generar páginas HTML dinámicamente como respuesta<sup>9</sup>.

### *PHP*

Es el lenguaje de *scripting* multipropósito que está situado especialmente para el desarrollo de páginas web. Su claridad en el diseño, módulos bien organizados y mejor mantenimiento de tecnologías, lo hacen uno de los lenguajes más usados en la industria actual. Su popularidad y credibilidad puede estar relacionada al hecho de que organizaciones reputadas como la universidad de Harvard o la red social Facebook, están basadas en PHP. Esto es posible porque los sitios PHP pueden ser fácilmente mantenidos, mejorados y actualizados de vez en cuando. Dicho lenguaje es de código abierto y multiplataforma<sup>10</sup>.

En el desarrollo de la propuesta de solución se utilizará como lenguaje de programación del lado del servidor PHP debido a la gran cantidad de facilidades y utilidades que brinda ya que permite el desarrollo de aplicaciones web dinámicas con acceso a la información almacenada en la base de datos, además, permite la conexión con la mayoría de los motores de bases de datos que se utilizan en la actualidad. Posee una amplia documentación en su sitio web oficial y la colaboración de cientos de expertos lo cual permite la corrección de errores con mayor facilidad.

### *Framework*

Un framework es un conjunto de archivos y directorios que facilitan la creación de aplicaciones, ya que incorporan funcionalidades ya desarrolladas y probadas, implementadas en un determinado lenguaje de programación.

### *Symfony*

Es un *framework* diseñado para desarrollar aplicaciones web, teniendo como base el patrón Modelo Vista Controlador. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. Symfony está desarrollado completamente en PHP 5.3. Es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y Microsoft SQL Server. Se puede ejecutar tanto en plataformas \*nix (Unix, Linux, etc.) como en plataformas Windows<sup>11</sup>.

Se escogió Symfony como framework de desarrollo para la realización del sistema propuesto pues permite crear aplicaciones y sitios web rápidos y seguros de forma profesional, además es sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.

### *Sistema gestor de base de datos*

Como sistema gestor de base de datos se utiliza PostgreSQL debido a que es un sistema de gestión de bases de datos objeto-relacional. Está liberado bajo licencia BSD, además es extensible, multiplataforma y presenta

---

<sup>9</sup><https://programacionwebisc.wordpress.com/2-3-lenguajesde-programacion-del-lado-del-servidor/>

<sup>10</sup> <http://www.staffcreativa.pe/blog/ventajas-programacion-php/>

<sup>11</sup> <https://es.wikipedia.org/wiki/Symfony>

modelos de negocios rentables con instalaciones a gran escala<sup>12</sup>. Tiene ahorros considerables en costos de operación. El software ha sido diseñado y creado para tener un mantenimiento y ajuste mucho menor que los productos de los proveedores comerciales, conservando todas las características de rendimiento. Además, permite usar procedimientos almacenados, tiene un rendimiento medio, es una tecnología libre, multiplataforma y puede ser utilizado con varios entornos de desarrollo. También es de código abierto y es el más usado en la universidad y en los productos que se desarrollan en el centro.

### ***Entorno Integrado de Desarrollo***

Un Entorno de Desarrollo Integrado, traducido del inglés Integrated Development Environment (IDE), es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien utilizarse para varios. Un IDE puede denominarse como un entorno de programación que ha sido tratado como un programa aplicación. Esto significa que consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica<sup>13</sup>.

#### ***PHPStorm***

Este IDE ofrece edición en directo con tecnologías como CSS, HTML5, JavaScript o TypeScript. La popularidad de PHPStorm se puede medir por el hecho de que grandes marcas como Expedia, Yahoo, Cisco, Salesforce y wikipedia también utilizan PHPStorm como su editor de código. Este editor realmente obtiene su código y comprende profundamente su estructura, soportando todas las características del lenguaje PHP para proyectos modernos y heredados. Proporciona la mejor terminación de código, refactorizaciones, prevención de errores sobre la marcha y más. Además, es conocido por su depurador visual de configuración cero, que proporciona una visión extraordinaria de lo que sucede en su aplicación en cada paso<sup>14</sup>.

Atendiendo al estudio anterior, el IDE recomendado para la implementación es PHPStorm debido a que proporciona el análisis de código, la prevención de errores, además incluye un editor de pleno derecho de SQL con resultados de la consulta editables.

#### ***Servidor web***

Un servidor web es el encargado de aceptar las peticiones de páginas (o recursos en general) que provienen de los visitantes que acceden a nuestro sitio web y gestionar su entrega o denegación, de acuerdo con las políticas de seguridad establecidas<sup>15</sup>.

Algunas de las funciones fundamentales de un servidor web:

---

<sup>12</sup> [POSTGRESQL, PostgreSQL Global Development Group, 2012](#)

<sup>13</sup> <http://programaciondesarrollo.es/que-es-un-entorno-dedesarrollo-integrado-ide/>

<sup>14</sup> <https://www.jetbrains.com/phpstorm/>

<sup>15</sup> [http://aprenderaprogramar.com/index.php?option=com\\_content&view=article&id=542:que-es-un-servidor-y-cuales-son-los-principales-tipos-de-servidores-proxydns-webftppop3-y-smtpdhcp&catid=57:herramientas-informaticas&Itemid=179](http://aprenderaprogramar.com/index.php?option=com_content&view=article&id=542:que-es-un-servidor-y-cuales-son-los-principales-tipos-de-servidores-proxydns-webftppop3-y-smtpdhcp&catid=57:herramientas-informaticas&Itemid=179)

- ✓ Atender de manera eficiente, ya que puede recibir un gran número de peticiones HTTP, incluyendo una ejecución multitarea ya que pueden darse peticiones simultáneas. Cualquier petición compleja (por ejemplo, con acceso a base de datos) dejaría colapsado el servicio.
- ✓ Almacenar las peticiones recibidas, errores que se han producido y en general toda aquella información que puede ser registrada y analizada posteriormente para obtener las estadísticas de acceso al sitio web.

### *Apache*

Apache es un proyecto de código abierto y uso gratuito, multiplataforma, muy robusto y que destaca por su seguridad y rendimiento<sup>16</sup>. Además, Apache permite configurar un Hosting Virtual basado en IPs o en nombres, es decir, tener varios sitios web en un mismo equipo o establecer distintos niveles de control de acceso a la información incluyendo el soporte a cifrado SSL utilizando protocolo seguro HTTPS<sup>17</sup>.

Para el montaje del sistema propuesto se decide utilizar Apache dado que corre en una multitud de sistemas operativos, lo que lo hace prácticamente universal, es un servidor altamente configurable que permite personalizar la respuesta ante los posibles errores que puedan ocurrir en el servidor.

### ***Resultados principales***

El proceso diseñado fue sometido a revisión por parte de un grupo de expertos (21) con conocimientos sobre: ingeniería, gestión y calidad de software. El 91,6 % de los encuestados posee 8 años de experiencia en actividades de aseguramiento y control de calidad de software y más de 5 años vinculados al desarrollo de aplicaciones.

Los resultados obtenidos se relacionan a continuación:

- El 100 % de los expertos señala la utilidad de la propuesta a partir de la problemática planteada.
- El 100 % consideró que la propuesta estaba ajustada a las características de los desarrollos realizados por la Red de Centros de la UCI.
- Todos destacan la posibilidad de reutilizar los campos particionados en varios escenarios.

Se recomendó exportar los diseños realizados para que formen parte del expediente del proyecto.

---

<sup>16</sup> <https://es.opensuse.org/Apache>

<sup>17</sup> <https://debianhandbook.info/browse/es-ES/stable/sect.http-web-server.html>

## CONCLUSIONES

- Se destaca la importancia de automatizar el diseño de casos de pruebas como actividad que garantiza la calidad del proceso de pruebas.
- Diseñar una base de conocimientos para para recomendar escenarios de prueba a partir de los requisitos planteados y las tendencias mostradas por los reportes, puede resultar útil para el desarrollo de la actividad particionar campo.

## REFERENCIAS

- [1] D. R. Valdivia Espinoza y E. G. Valdivia Espinosa, «Estandares de calidad para pruebas de software», Universidad Nacional Mayor de San Marcos, Lima, 2005.
- [2] L. Fernandez, «Un sondeo sobre la practica actual de pruebas de software en España». REICIS Revista Española de Innovación, Calidad e Ingeniería del Software», vol. 1, 2005.
- [3] A. Velázquez, A. Estrada, y S. Merced, «Evaluación de productos de software desde la perspectiva del cliente», 2018.
- [4] É. F. DE SOUZA, R. de A. FALBO, y N. L. VIJAYKUMAR, «Knowledge management initiatives in software testing: A mapping study. Information and Software Technology», vol. 57, 2015.
- [5] C. M. Souлары, «Estrategia de Gestión del Conocimiento para el Laboratorio de Pruebas de Software de la Universidad de las Ciencias Informáticas», Universidad de las Ciencias Informáticas, 2016.
- [6] CINDY CAMPOS CHIU, «LAS PRUEBAS EN EL DESARROLLO DE SOFTWARE», UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO, 2015.
- [7] «Formación para el “Probador Certificado – Nivel Básico”». 2010.
- [8] G. Márquez Sosa, *Formación para el “Probador Certificado – Nivel Básico” de acuerdo al programa de estudios 2010 del ISTQB.* .
- [9] P. Vargas, «Casos de Prueba - Particion Equivalente», 2017.
- [10] «Técnicas de Caja Negra», 11-ago-2013.
- [11] M. J. Castilla, «TABLA DE DECISION». .
- [12] «Pruebas de caja negra ISTQB», 2016.
- [13] S. P. José Manuel, «Pruebas de Software. Fundamentos y Técnicas», jun-2015.