

## **Desarrollo de un marco de trabajo (framework) para el desarrollo de aplicaciones web en la Universidad Nacional de Costa Rica**

Erick Núñez Navarrete <sup>a</sup>, Darío Ríos Navarro <sup>b</sup>

<sup>a</sup> Universidad Nacional de Costa Rica, Profesional Analista del Centro de Gestión Informática CGI, Departamento de Tecnologías de Información DTIC, Av .1 Calle 9. Campus Omar Dengo.

[erick.nunez.navarrete@una.cr](mailto:erick.nunez.navarrete@una.cr)

<sup>b</sup> Universidad Nacional de Costa Rica, Profesional Ejecutivo en Desarrollo Tecnológico del Centro de Gestión Informática CGI, Departamento de Tecnologías de Información DTIC, Av .1 Calle 9. Campus Omar Dengo.

[dario.rios.navarro@una.cr](mailto:dario.rios.navarro@una.cr)

**Resumen.** El auge de la tecnología y el desarrollo de herramientas orientadas a la web, han identificado que Universidad Nacional de Costa Rica (UNA) debe contar con mejores sistemas que solventen las necesidades de integración que tienen uno de sus sistemas heredados, con el cual se cuenta actualmente, así como otros sistemas, implementados en otros lenguajes, especialmente aquellos que están orientados a la gestión administrativa. Para ello fue necesario, detectar en esta institución, si el desarrollo se supliría de manera interna, por medio de adquisición o de una forma híbrida. Dada una serie de experiencias por las cuales ya se había pasado, previamente; se considera la necesidad de desarrollar e implementar un software empresarial que facilite el desarrollo de muchos programadores y les brinde las herramientas que requieran a un mayor grado de simplicidad y estandarización. El framework SDKUNA (Software Development Kit de la UNA) construido y mejorado por algunos de los desarrolladores del Centro de Gestión de Informática (CGI) cuenta con una arquitectura integral que ha permitido facilitar la implementación y ha brindado una alternativa a los programadores del área para integrar los diferentes sistemas que tiene la institución. Dicho framework, que ha requerido mucho trabajo, análisis y pruebas, está mejorando el proceso de creación de aplicaciones web que tiene la Universidad Nacional de Costa Rica, y se espera que su expansión hacia otras instancias, permita crear el grado de madurez y homogeneidad que actualmente se requiere a nivel institucional.

**Palabras Clave:** UNA, SDKUNA, Universidad Nacional, framework, integración.

**Eje temático:** Soluciones TIC para la Gestión.

### **1 Introducción**

Con el avance del internet, y el progreso de las tecnologías de la información y la comunicación, la evolución que ha tenido el desarrollo de software en aplicaciones web dio un giro diferente conforme a lo que se estaba acostumbrado a ver en aplicaciones de escritorio. En los inicios del internet, la web era sencillamente una colección de páginas estáticas, documentos, etc., para consulta o descarga. El paso inmediatamente posterior en la evolución fue la inclusión de un método para elaborar páginas dinámicas que permitieran que lo mostrado tuviese carácter dinámico [1]. Esto dio hincapié al nacimiento de lo que se llamaría posteriormente aplicaciones web, que permitió que un tipo especial de aplicación cliente/servidor, donde tanto el cliente (el navegador) como el servidor (el servidor web) y el protocolo mediante el que se

comunican (HyperText Transfer Protocol (HTTP)) estuvieran estandarizados [2], y además integren el uso de scripts que permitan la interacción dinámica entre el cliente y el servidor, donde se implemente el modelo de negocio conocido [3].

La Universidad Nacional de Costa Rica, continúa teniendo algunas de sus sistemas bajo el modelo de mainframe, y cuando hablamos de mainframe u ordenador central nos estamos refiriendo a ordenadores muy potentes y caros usados principalmente para el procesamiento de grandes cantidades de datos [4], modelo mediante el cual se trabajaba con diferentes sistemas haciendo uso del lenguaje de programación COBOL [5]. Estos sistemas, aunque son muy robustos, son de difícil mantenimiento y debido al manejo de instrucciones mediante códigos que los usuarios con mayor experiencia conocen. Sin embargo, los nuevos colaboradores que ingresan a la institución, aunque se capacitan en el uso de estos sistemas, tienen una curva de aprendizaje alta. Y el desarrollo de aplicaciones web en el mercado, da a los usuarios una visión moderna de lo que se puede hacer por el desarrollo de sistemas en estos tiempos. No obstante, el departamento de tecnologías de la información y la comunicación (DTIC) de esta universidad, está consciente de que debe impulsar los procesos de mejora en el área de las TICs, con el fin de dar mejores soluciones informáticas a la institución, brindando soluciones que se adapten a las necesidades actuales de la institución.

### **1.1 ¿Qué es un framework?**

El concepto framework se emplea bajo muchos ámbitos del desarrollo de sistemas software, no solo en el ámbito de aplicaciones web. Podemos encontrar frameworks para el desarrollo de aplicaciones médicas, de visión por computador, para el desarrollo de juegos, etc. Pero una definición de un framework web es una estructura de software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. Se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta [6].

El presente artículo da a conocer, el desarrollo e implementación de un framework para aplicaciones web mediante el uso de múltiples tecnologías, que está permitiendo a los administrativos y desarrolladores de la institución, definir un nuevo modelo de trabajo más estandarizado para los programadores del área, y mostrando a los usuarios finales, un producto más duradero, con una interfaz mucho más amigable, moderno y orientado a satisfacer las necesidades que actualmente se requieren en los procesos administrativos de la Universidad Nacional de Costa Rica.

## **2 Antecedentes**

### **2.1 Necesidad de mejorar la integración de los procesos de gestión administrativa.**

Una de las principales necesidades que ha acarreado el desarrollo de sistemas en la Universidad Nacional, es que, debido a la adquisición y desarrollo de diferentes plataformas, con el paso de los años, se ha dificultado la correcta integración de cada uno de los sistemas heredados con otros sistemas soportados en otros lenguajes de programación. Otras facultades de la institución, han optado por generar sus propios sistemas, bajo diferentes estándares de programación, para su uso propio. Lo cual no permitía tener una única herramienta de programación que permitiera integrar los procesos de la organización.

Como la mayoría de sistemas administrativos, se han desarrollado bajo el modelo que brinda un sistema llamado NX, que corresponde a un sistema heredado [7], que aunque es robusto, lo cierto es que muchos de líderes de TI no piensan tener estos sistemas en sus empresas toda la vida, por eso están buscando la manera de romper esas cadenas del pasado para obtener mejores resultados, optimización de procesos y servicios [8]. Por estas razones, se buscaba hacerse una inversión en un sistema que cumpliera con las condiciones de un software moderno y adaptado a las necesidades de la institución.

## **2.2 Dificultad en el mercado para encontrar desarrolladores en sistemas heredados**

Con el avance de las tecnologías y la integración de los proyectos a la web y a la nube [9], también se ha diversificado el tipo de desarrollador necesitado en el ambiente laboral. La mayoría de los colegios y las universidades ya no están enseñando COBOL, sino que están a favor de nuevos lenguajes como Java, C ++, UNIX y Linux, y, por lo tanto, Cobol y habilidades como la unidad central son un bien escaso [10].

Los sistemas como NX que actualmente tiene la Universidad Nacional, corresponden a un lenguaje de programación COBOL, y como se identificó anteriormente, se dificulta más encontrar desarrolladores en este lenguaje. Por esa razón, los desarrolladores que fueron pioneros en el proyecto, debieron analizar además de las herramientas tecnológicas, que eventualmente deberían contratar personal que tuviera conocimiento en tecnologías modernas, para la construcción de este framework web.

## **2.2 Proyecto SIGESA**

El proyecto SIGESA (Sistema de Gestión Administrativa) nace como una necesidad de la institución de crear aquella integración, que en ese momento no estaba solventada por los sistemas actuales.

El objetivo principal del proyecto SIGESA es analizar, rediseñar, automatizar e implementar, con base en el proceso de Planificación Operativa Institucional, los procesos de los Programas de Gestión Financiera, Proveeduría Institucional y Desarrollo de Recursos Humanos con el fin de optimizar los recursos, mejorar los servicios con orientación al usuario y contribuir con la toma de decisiones institucionales [1].

Mediante un estudio de factibilidad realizado para el proyecto, se orientó a determinar el método de automatización de los procesos rediseñados e implementados, dando como resultado una de las posibles opciones [11]:

- Adquisición
- Desarrollo Interno
- Adquisición y Desarrollo Interno

La universidad Nacional ya previamente había experimentado un proceso de adquisición con el proyecto Banner, sin embargo, una de las problemáticas que ha tenido el sistema es su falta de integración con otros sistemas, como con el programa de gestión financiera, la administración de recursos humanos y otros registros de transacciones [12]. Aparte, se identificó que una de las situaciones que tenían descontentos a algunos usuarios, es que el soporte dependía de una empresa internacional, que es una dependencia que se tenía del sistema [12].

Por lo cual, se verificó que para el proyecto SIGESA, era conveniente investigar sobre algunas herramientas para hacer un desarrollo interno, que fuera llevado a cabo por

los mismos analistas y programadores del área, donde ellos mismos pudieran efectuar el soporte, y además pudieran hacer análisis de los requerimientos internos de la institución, sin la necesidad de otros proveedores externos.

### **3 Desarrollo e implementación del Framework SDKUNA**

El framework SDKUNA (Software Development Kit – UNA) satisface las necesidades que en este momento tiene la Universidad Nacional, dándole una forma estandarizada de trabajo. En sus inicios, se buscó la manera de integrar diversas tecnologías, preferiblemente de software libre que permitiera desarrollar una herramienta de entorno empresarial. Ya algunos sistemas que se desarrollaron internamente habían experimentado el uso de Enterprise Java Beans. Enterprise Java Beans (EJB) corresponde a un modelo interfaces de componentes en el servidor, que permite la rápida y robusta implementación en tecnologías Java [13]. Sin embargo, uno de los análisis que verificamos, es que esta herramienta tiene varios problemas con fugas de memoria. Algunos de los cuales podían ser solucionados mediante mejoras de rendimiento [14] y otros de los cuales no se podían solucionar de momento y se mostraban como averías. Un estudio realizado sobre uno de los sistemas de la universidad, implementado bajo el modelo EJB, indicó una seria problemática de rendimiento, que aunque se mejoró, eventualmente se destacó la posibilidad de implementar medios para reiniciar el servidor de aplicaciones que tenía este sistema, de manera tal que se evitarán los problemas de fugas de memoria. [15]

Esto también fue un factor primordial, que hizo a los desarrolladores buscarán nuevas maneras de implementar un sistema bajo otro modelo con patrones de diseño integrados que no tuviera el problema descrito anteriormente.

#### **3.1 Tecnologías Utilizadas**

Tras realizar investigación, desarrollos y pruebas sobre herramientas y tecnologías diferentes, el framework SDKUNA se implementa haciendo uso de estas tecnologías, que se han ido considerando dentro del proyecto según las necesidades deseadas a lo largo del proceso:

**Spring:** Spring es un framework de Java de código abierto muy popular y ampliamente desplegado en todo el mundo. Ayuda a los desarrolladores a crear aplicaciones de alta calidad, más rápidamente. Spring proporciona un modelo de programación coherente y consistente, el cual es bien comprendido y usado por millones de desarrolladores Java. Spring es el apoyo infraestructural a nivel de aplicación: Spring se centra en la “fontanería” de las aplicaciones empresariales [16]. Para el framework SDKUNA, Spring se concentra en la administración de las otras tecnologías, haciendo una inyección de dependencias de memoria, sobre cada uno de los objetos que conforman el sistema. Se utiliza también en la implementación de seguridad de la aplicación que es uno de los puntos fuertes del framework SDKUNA.

**Hibernate:** Hibernate es un framework de código abierto de persistencia de datos. Permite realizar mapeos relacionales de objetos y bases de datos de consultas utilizando HQL y SQL [17]. Existe una integración muy adecuada entre Spring y este framework. Para el SDKUNA, representa una herramienta muy útil para no depender directamente del tipo de gestor de base de datos, que es uno de los beneficios que presenta este framework.

**JSF:** La tecnología JavaServer Faces consiste en un conjunto de APIs (Application Programming Interface) para representar los componentes de interfaz de usuario y

administrar su estado, gestionar eventos y validación de entrada, definiendo la navegación de páginas y apoyando la internacionalización y la accesibilidad. [18]. Muchas de las funcionalidades apoyadas para la interfaz, se manejan haciendo uso de la tecnología JSF. Sin embargo, aunque la estructura del framework se desarrolla bajo el ciclo vida de JSF [19] sobre la comunicación del servidor con el cliente, se han utilizado componentes de Primefaces, que, dada su dinámica e interacción mediante AJAX, tiene interfaces más ricas y mejoradas para interactuar con los usuarios finales.

**Primefaces:** Los componentes de PrimeFaces se desarrollan con un principio de diseño que establece que "un buen componente de interfaz de usuario debe ocultar la complejidad pero mantener la flexibilidad" [20]. Estos componentes corresponden a un sabor de JSF mejorado, para brindar interfaces muy dinámicas. Entre las características más importante que presenta esta librería, es el poder integrarse con Spring, un área de ejemplos de los componentes y su amplia documentación y comunidad que brinda soporte a otros desarrolladores.

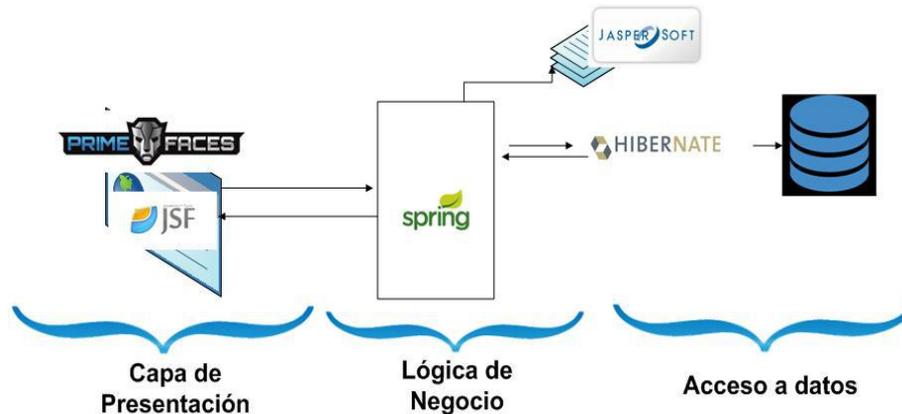
**JasperReports:** Herramienta utilizada para la generación de reportes. JasperReports es el motor de reportes de código abierto más popular del mundo. Está completamente escrito en Java y es capaz de utilizar datos procedentes de cualquier tipo de fuente de datos y producir documentos perfectos en píxeles que se pueden ver, imprimir o exportar en una variedad de formatos de documentos incluyendo HTML, PDF, Excel, Open Office y Word [21]. Con este reporteador, y con su integración en Java, se hace uso de esta herramienta para poder realizar cualquiera de las funciones de reportes que se requieran en el sistema.

**Maven:** En sí, el uso de la tecnología Maven, tiene varias funciones sobre este framework. En primera instancia se hace de esta tecnología para la gestión de dependencias [22]. Lo cual permite tener acceso a librerías ubicadas en un repositorio central, y además el poder compartir dependencia entre cada uno de los proyectos que tiene SDKUNA.

**Alfresco Activiti BPM:** Una de las funcionalidades más importantes que presenta el framework SDKUNA, es la utilización de los flujos empresariales. Esta funcionalidad, se ha adaptado a las necesidades del framework, haciendo uso de Alfresco Activiti BPM. Alfresco Activiti es una solución de gestión de procesos empresariales (BPM) orientada a desarrolladores y personas del entorno empresarial. Incorpora un motor de procesos empresariales de alto rendimiento con la flexibilidad y escalabilidad necesarias para manejar una amplia variedad de procesos críticos. La moderna solución de BPM ofrece también un potente conjunto de herramientas de usuario final y se integra en diversos sistemas de gestión de contenido empresarial, incluido Alfresco One [23]. Esta herramienta, ha permitido crear funciones en el framework, que se verificarán más adelante.

### **3.2 Estructura de SDKUNA**

A continuación, se muestra un diagrama que ejemplifica cómo están integradas algunas de las tecnologías utilizadas en la implementación del framework SDKUNA. Fig.1.



**Fig. 1.** Estructura SDKUNA considerando las tecnologías especificadas y el acceso mediante capas

### 3.1 Principales beneficios de SDKUNA

**Estandarización de la programación:** Mediante la implementación de este framework, se ha creado un estándar de desarrollo, más intuitivo y más fácil de utilizar por los desarrolladores. Lo cual les permite a estos, saber cómo están implementados y donde se encuentran ubicadas las estructuras de los paquetes y clases que tiene cada uno de los proyectos. Inclusive el uso de dependencias es un factor clave, ya que la forma en que están estructurados los proyectos ya se encuentra estandarizado.

**Patrón MVC:** El patrón MVC (Modelo, Vista, Controlador, por sus siglas en inglés) [24] permite dividir las aplicaciones en componentes, y separar la lógica del negocio de la capa de presentación visual. Debido a las tecnologías desarrolladas en el framework y a su correcta integración, se puede definir; un patrón que cumple con esas características, de tal manera que no haya una dependencia completa, si cada una de las capas son separadas. Esto se da con la intención de brindar un modelo que pueda ser escalable (en caso de necesitarlo o querer realizarlo) de manera que todos los componentes estén separados físicamente pero que se puedan integrar en forma lógica.

**Independencia del gestor de base de datos:** Al implementar un modelo bajo ORM (Object Relational Mapping) [25] que está relacionado con la implementación Hibernate, integrado con Spring, para poder ejecutar las consultas bajo un modelo de consultas SQL orientado a objetos, ha permitido que el framework no tenga dependencia del gestor de base de datos que se utilice, únicamente teniendo que modificar su configuración para indicar cuál gestor de base de datos debe utilizar.

**Integración de diferentes módulos:** Con la implementación de proyectos bajo el modelo de PARENT POM [26] que presenta Maven y su capacidad de administrar las dependencias de los proyectos, se solventó una de las necesidades más importantes que tienen los sistemas actuales de la Universidad Nacional, que es la falta de integración de los proyectos. Este desarrollo, aunado al proyecto SIGESA, ha facilitado la mayor comunicación y uso de módulos que en el pasado, no se podían utilizar de forma completa.

**Configuración de Tecnologías complejas:** Normalmente la implementación, configuración e integración de las tecnologías mencionadas anteriormente, requiere tiempo y muchas veces su instalación puede ser compleja. El framework SDKUNA,

implementa las configuraciones necesarias, para que los desarrolladores puedan acceder al desarrollo de las herramientas, sin tener que pasar por el engorroso proceso de realizar configuraciones de proyectos.

**Reutilización de código:** Parte de la integración de módulos y configuración previa de las tecnologías, es que les ha facilitado a los desarrolladores, ciertas funciones propias de cualquier sistema web, que evitan que el desarrollador tenga que realizarlas. Al consumir directamente estas funciones y haciendo uso de los medios de administración que tiene el framework, se permite reutilizar el código generalizado que se puede aplicar en otros sistemas.

**Componentes de interfaces mejorados:** La librería Primefaces, brinda diferentes componentes de interfaz enriquecidos para trabajar en las pantallas de las aplicaciones web. Pero como una mejora de los elementos que brinda dicha librería, también se han desarrollado diferentes componentes que tienen una interacción mejorada para efectos del usuario final y del estándar que implementa el framework. Podemos encontrar: barras de herramientas, tablas de datos, combos con paginación, listas de valores, entre otros.

### **3.2 Funcionalidad**

A continuación, se explican algunas de las funcionalidades que implementa el framework SDKUNA, que se han implementado a lo largo de su creación y forman parte de la documentación del sistema y del framework [27]:

**Administración de usuarios:** La administración que implementa el framework, permite la adecuada manipulación del acceso de usuarios al sistema, por medio bases de datos o haciendo uso de un directorio de usuarios o cualquier otro protocolo de acceso a directorios. Además, implementa diferentes funciones que permiten a los desarrolladores saber en cualquier momento, cual es el usuario que se encuentra autenticado.

**Administración de roles:** La administración de roles se vincula con la administración de usuarios y de recursos. Mediante el esquema propuesto se puede catalogar roles de acceso con ciertos privilegios a diversos recursos, de tal manera que haya un control adecuado de lo que los usuarios pueden ver cuando ingresan al sistema. Esta funcionalidad, al ser heredada del framework, puede ser utilizada en los diversos sistemas que la quieran utilizar, estableciendo los roles según sus necesidades.

**Administración de recursos:** La administración de recursos, está fuertemente estructurada y relacionada con los usuarios y roles del sistema. Los recursos responden a diferentes tipos de objetos con los cuales interactúa en pantalla como etiquetas, texto o campos. Principalmente, se crean recursos de página, que se relacionan con los roles, para indicar nivel de acceso de un usuario. Normalmente si no se realiza la adecuada creación de recursos de página, estos no pueden ser vistos por ningún usuario del sistema. Con la integración de listas de acceso (ACL) [28] se puede lograr la vinculación con los recursos para obtener un mayor nivel de permisos sobre ciertos objetos que aparecen en los formularios.

**Administración de listas de acceso:** Las listas de acceso, tienen diferentes términos, correspondientes al área donde se crean. En el framework, las listas de acceso de control (ACL), tienen la capacidad de establecer mejores niveles de acceso tanto a páginas como campos. Inclusive permiten mediante los usuarios y roles, determinar a qué privilegios pueden acceder estos. Son muy adecuados también, para identificar nivel de acceso de objetos e inclusive de flujos de trabajo que tiene el framework.

**Campos configurables:** Los campos configurables es una de las herramientas más dinámicas con las que cuenta el framework. Los desarrolladores implementan una extensión del formulario habitual configurado para una determinada entidad. Es decir, son campos adicionales para una tabla, pero los mismos se generan de forma dinámica, no se encuentran previamente programados en los formularios. Esto es de mucha utilidad, para implementar campos de formularios que están continuamente cambiando, según las necesidades de los usuarios.

**Botones adicionales:** En el caso de que los componentes enriquecidos, no sean suficiente para el desarrollador, por ejemplo, que los botones de las barras herramientas no complementen alguna acción necesaria en pantalla, y se requiera otro botón para alguna funcionalidad específica, el framework permite la agregación y modificación de los botones. permitiendo que los desarrolladores, siguiendo ciertas instrucciones, puedan crear sus propios botones adicionales para solventar funcionalidades específicas.

**Manejo de adjuntos:** Una de las herramientas de mayor utilización de todo el framework, son las funciones de manejo de subida y descarga de archivos adjuntos. Para cada uno de los objetos que se crean en el sistema, se pueden agregar adjuntos. Toda la información que se obtiene de los adjuntos, se almacena en un repositorio de documentos llamado Alfresco [29], el cual administra estos archivos considerando su versión.

**Lanzador de reportes:** El lanzador de reportes, se crea bajo el criterio de un lanzador de reportes con el que contaba NX. Sin embargo, este hace uso de la herramienta de JasperReports para poder ejecutar el reporte. Una característica de este modo de operación de reportes es que aparte de generar el reporte en varios formatos, también se puede programar para ser enviado por correo.

**Lanzador de procesos:** El lanzador de procesos es una herramienta idónea para realizar diferentes actividades donde las mismas quedan programadas bajo un sistema de tiempo conocido como CRON [30]. La misma herramienta cuenta con una consola de procesos para administrar los procesos que se están corriendo y con una bitácora en pantalla de los archivos de transacciones generados por los procesos. Se pueden ejecutar procesos de forma inmediata, o se pueden programar para ejecutarse posteriormente, según la expresión de tiempo seleccionada (se puede especificar, años, meses, días, horas, minutos y segundos).

**Flujos de trabajo:** Los flujos de trabajo que desarrolla el framework; están integrados por medio de la herramienta Alfresco Activiti. El proceso consta de la creación de un diagrama en formato BPMN [31]. Dicho diagrama, establece las reglas para poder llevar a cabo un flujo de trabajo. Con el uso de roles y listas de acceso, se pueden establecer en cada uno de los flujos, niveles de interacción de diferentes mandos, simulando el proceso jerárquico que se desea ejecutar. Permite identificar en donde se encuentra el flujo en determinado momento, así como permite verificar las tareas pendientes de un flujo y porque fue aceptado o rechazado un proceso del flujo al estar en proceso.

**Gráficos:** Una funcionalidad importante del framework, es el uso de gráficos. Para ello se ha creado un componente basado en JSF, que permite la generación de los mismos, considerando las diferentes entidades que se quieren relacionar en el gráfico.

**Internacionalización:** Una necesidad inherente de este tipo de framework es el poder satisfacer el desarrollo de algunas de las funcionalidades especificadas considerando el idioma. Por lo cual el módulo de internacionalización, permite la asignación de palabras en el idioma deseado para cada título, etiqueta o texto del sistema que se utilice para desplegar información al usuario final. De momento, el framework tiene disponible sus textos en inglés y español.

## **4 Impulso al desarrollo de software libre**

La Universidad Nacional (UNA) por medio de su escuela de informática, ha denotado importante interés en desarrollar el uso de software libre en la administración pública de Costa Rica [32]. De esta forma, una de las características que tiene el framework, es que casi la totalidad de sus componentes fueron construidos, haciendo uso de frameworks y librerías de software libre. La Universidad Nacional de esta forma, ha centrado sus esfuerzos en establecer mejores condiciones a nivel de arquitectura (servidores y base de datos) que propiamente en licencias para la implementación de todos los aspectos del sistema. Lo cual, representa un logro de importancia en el campo del software libre, dando a conocer que realmente, existen diversos frameworks y programas que apoyan las soluciones para desarrollos empresariales y ahorran sumas importantes de dinero en el uso y mantenimiento de licencias. Respecto al soporte, podemos destacar que muchas de las herramientas utilizadas en el framework, tienen una documentación muy completa, y están en constante mejora, debido a la actividad de las comunidades que trabajan día a día en la implementación y mejora de errores.

## **5 Resultados**

### **5.1 Implementación de proyectos**

#### **5.1.1 Sistema de declaración jurada de Horario**

El sistema de declaración jurada de horario, se implementó con el fin de hacer uso de esta herramienta para que tanto administrativos como docentes de la Universidad Nacional, puedan acceder a declarar y ajustar su horario de trabajo para la institución. El sistema de declaración jurada de horario anterior, no se encontraba integrado con otros sistemas, y no tenía la posibilidad de hacer uso de otras herramientas que solventa este sistema implementado con el framework. Una característica de importancia que este nuevo sistema adquirió con el uso del framework, es que sistema hace uso de una base de datos que está implementada en POSTGRESQL, que es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente y es considerado uno de los más potentes [33]. De esta forma, se hace uso del framework sobre este sistema, para mejorar el uso de esta herramienta de administración en la Universidad Nacional, haciendo uso de una base de datos de software y de código abierto.

#### **5.1.2 SICA**

El sistema de carrera académica (SICA) es un sistema desarrollado para la comisión académica, que surge como la necesidad de tener un sistema para evaluar y administrar los atestados de los académicos. Todo el manejo de expedientes, se hacía de manera manual, por lo cual se quiso realizar un sistema que solventará las necesidades de esta área. Actualmente el sistema que fue creado bajo la estructura del framework, y permite realizar funciones como la solicitud de un estudio, la gestión de documentos digitales y el cálculo de puntajes para establecer el grado académico que debe tener un profesor.

#### **5.1.3 SIGESA: Módulo de atracción del talento humano y Persona General.**

El módulo de atracción del talento humano, junto el módulo de persona general, han sido de los primeros módulos que se han implantado en la Universidad Nacional como parte del proyecto SIGESA. Esta parte del sistema, es la encargada de llevar a cabo los concursos internos y externos del personal que va a formar parte de un nuevo puesto en la universidad o está ingresando a laborar. También incluye las evaluaciones y resultados obtenidos de pruebas técnicas realizadas para un puesto, y brinda información a todos los funcionarios de su registro de elegibles (puestos en los cuales pueden participar). Este módulo junto con el de Persona General que guarda información de las personas físicas y jurídicas que laboran en la institución, ya está implantado en producción y se encuentra en funcionamiento.

## **6 Retos**

El proceso de desarrollo del framework SDKUNA aún debe enfrentar mayores retos para entrar en un proceso de maduración completa de la herramienta. Actualmente contamos con un framework idóneo para desarrollar diferentes funcionalidades que se ha logrado mediante el uso de diferentes herramientas del software libre. Queda por definir si en algún momento este framework, será abierto al público, para formar una comunidad de desarrollo que fortalezca el proceso de crear retroalimentación y soporte. Esto corresponde a una decisión de la administración central de la institución, donde se debe valorar que si se va a dejar completamente abierto o si la institución tendrá en cuenta alguna licencia de software libre que respalde sus derechos de autor. Otro de los retos que debe enfrentar el framework SDKUNA, es el desarrollo como una herramienta accesible que cumpla con las normas indicadas en la WCAG 2.0 [34] para alcanzar por los menos el nivel AA. Esta tarea no ha sido sencilla, debido a que requiere mucha colaboración de parte de las mejoras que se realicen principalmente a nivel de interfaz, por otras tecnologías utilizadas en el framework para que cumplas con los lineamientos de ATAG [35].

Posteriormente, se espera que se implemente una segunda etapa del proyecto en la cual se lleve la solución del framework SDKUNA a la academia, para solventar las necesidades que se requieren en esta importante área.

## **7 Conclusiones**

La arquitectura propuesta para el Framework SDKUNA ha contribuido sobre los analistas y programadores de la UNA, para satisfacer un proceso de simplificación y homogeneización en desarrollo de aplicaciones web. Fomentando una arquitectura reutilizable y un conjunto de herramientas y bibliotecas que implementan los componentes más habituales y necesarios para el desarrollo de estas aplicaciones web.

La Universidad Nacional ya está trabajando con esta herramienta que representa el esfuerzo de muchos especialistas del centro de gestión informática. Es importante dar a conocer no solamente el proyecto SIGESA que inició este proceso, sino las verdaderas ventajas que tiene el framework SDKUNA como una herramienta potencialmente útil para toda la comunidad universitaria. La Universidad Nacional debe destacar este logro e implementar el uso del framework a nivel institucional.

Una de las problemáticas indicadas en los sistemas heredados como NX, o la implementación del Sistema Banner, fue solventada mediante las características de

*Séptima Conferencia de Directores de Tecnología de Información, TICAL 2017  
Gestión de las TICs para la Investigación y la Colaboración, San José, del 3 al 5 de  
julio de 2017*

este framework, para lograr integrar varios sistemas que en el pasado no podían comunicarse de la mejor manera entre ellos.

El framework SDKUNA ha fortalecido el proceso de implementación de los sistemas en el CGI de la DTIC de la Universidad Nacional. Se ha incrementado la generación de nuevo conocimiento a través de cada una de las herramientas que componen este framework. Incentivando y creando conocimiento sobre cada uno de los analistas del área. El framework SDKUNA ha fomentado una solución a la gestión administrativa, fortaleciendo el desarrollo interno de la institución.

## **Agradecimientos**

Este trabajo se ha desarrollado gracias a la colaboración de la Dirección de Tecnología de Información y Comunicación (DTIC) de la Universidad Nacional y a todos los analistas que forman parte de proyecto SIGESA de la UNA.

## **Referencias**

- [1] Barzanallana, Rafael. Universidad de Murcia. Servicios en internet. Historia del desarrollo de aplicaciones Web. [Online]. Recuperado el 10 de abril de 2017. Available: <http://www.um.es/docencia/barzana/DIVULGACION/INFORMATICA/Historia-desarrollo-aplicaciones-web.html>
- [2] Mora Luján, Sergio. Programación en Internet: Clientes WEB. [Online]. Recuperado el 10 de abril de 2017. Available: [https://rua.ua.es/dspace/bitstream/10045/16994/1/sergio\\_lujan-programacion\\_en\\_internet\\_clientes\\_web.pdf](https://rua.ua.es/dspace/bitstream/10045/16994/1/sergio_lujan-programacion_en_internet_clientes_web.pdf)
- [3] Conallen, Jim. Modelling Web Applications with UML. [Online]. Recuperado el 10 de abril de 2017. Available: <http://www.cs.toronto.edu/km/tropos/conallen.pdf>
- [4] Cardenas Gomez, Roberto. Introducción a los mainframes. [Online]. Recuperado el 10 de abril de 2017. Available: <http://cryptomex.org/SlidesMainframes/IntroMainframes.pdf>
- [5] Coughlan, Michael. (2014) Beginning COBOL for Programmers. Editorial Apress.
- [6] Gutiérrez, Javier J. ¿Qué es un framework Web? [Online]. Recuperado el 11 de abril de 2017. Available: [http://www.lsi.us.es/~javierj/investigacion\\_ficheros/Framework.pdf](http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf)
- [7] Cervantes, José de Jesús. (2013) [Online] Recuperado el 12 de abril de 2017. Available: <http://aplicdistri.blogspot.com/2013/02/24-integracion-de-sistemas-heredados.html>
- [8] Gutiérrez, Roger. (2016) Llegó el momento de romper los sistemas heredados. [Online] Recuperado el 12 de abril de 2017. Available: <https://revistaitnow.com/llego-momento-romper-los-sistemas-heredados/>
- [9] Toro López, Francisco J. (2013) Administración de proyectos de informática. ECOE Ediciones. Bogotá, Colombia.
- [10] Aguilar, Quisi (2014) Cobol, un programador que se extingue. [Online] Recuperado el 12 de abril de 2017. Available: <https://revistaitnow.com/cobol-un-programador-que-se-extingue/>
- [11] UNA. SIGESA (Sistema de gestión administrativa) [Online] Recuperado el 12 de abril de 2017. Available: <http://www.sigesa.una.ac.cr/index.php/80-informacion-general/104-nuestros-objetivos>
- [12] Duran Cascante, Giovanni. (2012). Análisis sobre el impacto del sistema informático banner en el desempeño del programa de gestión financiera de la Universidad Nacional, a partir de su implementación en el año 2008. Tesis. Recuperado el 13 de abril de 2017. Available: <http://repositorio.uned.ac.cr/reunited/bitstream/120809/885/1/Analisis%20sobre%20el%20impacto%20del%20sistema%20informatico%20banner%20.pdf>
- [13] Oracle. Enterprise Java Beans. Recuperado el 13 de abril de 2017. Available: <http://www.oracle.com/technetwork/java/javaece/ejb/index.html>
- [14] Java Performance Tuning (2017). EJB Performance tips. Recuperado el 13 de abril de 2017. Available: [http://www.javaperformancetuning.com/tips/j2ee\\_ejb.shtml](http://www.javaperformancetuning.com/tips/j2ee_ejb.shtml)

*Séptima Conferencia de Directores de Tecnología de Información, TICAL 2017  
Gestión de las TICs para la Investigación y la Colaboración, San José, del 3 al 5 de  
julio de 2017*

- [15] UNA DTIC. Análisis de rendimiento de boleta de pago UNA. Documentación Universidad Nacional.
- [16] Springla. Spring Framework. Recuperado el 13 de abril de 2017. Available: <https://springla.io/spring/spring-framework/>
- [17] Hibernate. Hibernate. Everything data. Recuperado el 13 de abril de 2017. Available: <http://hibernate.org>
- [18] Oracle. JavaServer Faces Technology. Recuperado el 13 de abril de 2017. Available: <http://www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html>
- [19] Leonard, Anghel. (2014). Mastering JavaServer Faces 2.2. Appendix F. The JSF LifeCycle. Editorial O'reilly.
- [20] Primefaces. Why PrimeFaces. Recuperado el 13 de abril de 2017. Available: <https://www.primefaces.org/whyprimefaces/>
- [21] JasperReports® Library. Recuperado el 13 de abril de 2017. Available: <http://community.jaspersoft.com/project/jasperreports-library>
- [22] Apache Maven Project. (2017). Introduction to the Dependency Mechanism. Recuperado el 13 de abril de 2017. Available: <https://maven.apache.org/guides/introduction/introduction-to-dependency-mechanism.html>
- [23] Alfresco Activiti BPM. Recuperado el 13 de abril de 2017. Available: <https://www.alfresco.com/es/products/business-process-management/alfresco-activiti>
- [24] Chrome (s.f.). MVC Architecture. Recuperado el 13 de abril de 2017. Available: [https://developer.chrome.com/apps/app\\_frameworks](https://developer.chrome.com/apps/app_frameworks)
- [25] Hibernate. What is Object/Relational Mapping? Recuperado el 15 de abril de 2017. Available: <http://hibernate.org/orm/what-is-an-orm/>
- [26] Apache Maven Project. (2017). Parent POM. Recuperado el 15 de abril de 2017. Available: <https://maven.apache.org/pom/>
- [27] UNA DTIC. Estándar de programación SDKUNA. Documentación Universidad Nacional. Última actualización: 2017.
- [28] Glosario de Informática e Internet. Definición de ACL. Recuperado el 15 de abril de 2017. Available: <http://www.internetglosario.com/1135/ACL.html>
- [29] Alfresco Documentation. Alfresco Repository. Recuperado el 15 de abril de 2017 Available: <http://docs.alfresco.com/community5.0/concepts/dev-repository-intro.html>
- [30] (Sin autor). Cron Format. Recuperado el 15 de abril de 2017. Available: <http://www.nncron.ru/help/EN/working/cron-format.htm>
- [31] Alfresco. Alfresco presenta el motor de workflow Activiti BPMN 2.0. Recuperado el 15 de abril de 2017. Available: <https://www.alfresco.com/es/noticias/comunicados-de-prensa/alfresco-presenta-el-motor-de-workflow-activiti-bpmn-20>
- [32] Periódico La Nación. Software libre en Costa Rica conquista a gobiernos locales Recuperado el 15 de abril de 2017. Available: [http://www.nacion.com/vivir/ciencia/Software-libre-conquista-gobiernos-locales\\_0\\_1374462541.html](http://www.nacion.com/vivir/ciencia/Software-libre-conquista-gobiernos-locales_0_1374462541.html)
- [33] Martínez, Rafael. Sobre PostgreSQL. Recuperado el 15 de abril de 2017. Available: [http://www.postgresql.org.es/sobre\\_postgresql](http://www.postgresql.org.es/sobre_postgresql)
- [34] W3C. (2008) Web Content Accessibility Guidelines (WCAG) 2.0. Recuperado el 16 de abril de 2017. Available: <https://www.w3.org/TR/WCAG20/>
- [35] W3C. (2015) Authoring Tool Accessibility Guidelines (ATAG). Recuperado el 16 de abril de 2017. Available: <https://www.w3.org/WAI/intro/atag.php>