

Sistema de Gestión Académica a través del desarrollo de Modelo-Vista-Controlador

Esperanza Manrique Rojas¹, Margarita Ramírez Ramírez², Hilda Beatriz Ramírez Moreno³, María del Consuelo Salgado Soto⁴, Nora del Carmen Osuna Millán⁵, Luis Manuel Cerda Suarez⁶.

emanrique@uabc.edu.mx, maguieram@uabc.edu.mx, ramirezmb@uabc.edu.mx, csalgado@uabc.edu.mx, nora.osuna@uabc.edu.mx, luismanuelcerdasuarez@gmail.com

¹ Universidad Autónoma de Baja California, 22000, Tijuana Baja California, México.

² Universidad Autónoma de Baja California, 22000, Tijuana Baja California, México.

³ Universidad Autónoma de Baja California, 22000, Tijuana Baja California, México.

⁴ Universidad Autónoma de Baja California, 22000, Tijuana Baja California, México.

⁵ Universidad Autónoma de Baja California, 22000, Tijuana Baja California, México.

⁶ Universidad Internacional de la Rioja, 26006, Madrid, España

Pages: 1083–1093

Resumen: El presente proyecto de desarrollo de los módulos base de un sistema de gestión académica para la Facultad de Contaduría y Administración o por sus siglas SIGAF, a través de una arquitectura de desarrollo Modelo-Vista-Controlador y obedeciendo las mejores prácticas, metodologías y técnicas del desarrollo de software construido sobre tecnologías web modernas como PHP, Javascript, jQuery, Ajax, HTML5 y CSS3, utilizando marcos de trabajo como Laravel con la finalidad de realizar código sustentable y mantenible para un proyecto de desarrollo sustentable y continuo.

Palabras-clave: Backend, Programación, Sistemas, Horarios, MVC.

Academic Management System through the development of Model-View-Controller.

Abstract: The present project of development of the base modules of an academic management system for the Faculty of Accounting and Administration or by its abbreviations SIGAF, through a Model-View-Controller development architecture and obeying the best practices, methodologies and techniques of the software development built on modern web technologies such as PHP, Javascript, jQuery, Ajax, HTML5 and CSS3, using frameworks such as Laravel with the aim of making sustainable and maintainable code for a sustainable development project and continuous.

Keywords: Backend, Programming, Systems, Schedules, MVC.

1. Introducción

En las organizaciones, incluyendo las educativas, poseen sistemas de información los cuales alimentan a grandes bases de datos. Estas bases de datos con gran volumen de información tienden a convertirse en un gran problema para las organizaciones que no poseen una arquitectura de aplicaciones adecuada y de datos correctamente definida. El manejo de estos grandes volúmenes de datos alimentados por varios sistemas informáticos lleva a crear dificultades en las tareas de comunicación e intercambio de información entre aplicaciones (Narvaes & Piedra, 2018).

Actualmente en la Facultad de Contaduría y Administración de la Universidad Autónoma de Baja California, campus Tijuana, México, se imparten 4 Licenciaturas, se ofertan 240 asignaturas, imparten clases 300 a 350 docentes distribuidos en 120 grupos al semestre. Todos estos datos son organizados y combinados por el personal administrativo para crear los horarios de cada grupo semestralmente. Este proceso de generación de horarios se realizaba durante los últimos meses de cada semestre y organizado por un sistema de información local con múltiples limitaciones para el mantenimiento de módulos.

Algunas instituciones educativas han implementado sistemas de horarios a través de aplicaciones informáticas y han logrado de esta manera optimizar la gran mayoría o la totalidad los procesos de generación y gestión de horarios, estos sistemas reducen significativamente largos tiempos de trabajo y horas hombre invertidas en estas actividades. Instituciones educativas tanto regionales como extranjeras, han solventado esta problemática mediante la aplicación de la tecnología con el uso de los sistemas de información.

Con la introducción de los sistemas de información se automatizan los procesos claves en las organizaciones e instituciones al hacer más eficiente las operaciones relativas al giro en el que se desenvuelven, así la generación y gestión de los horarios tiene un área de oportunidad para adoptar este avance tecnológico, como los son los sistemas de información, en la actualidad existen soluciones informáticas y sistemas que agilizan todo este proceso de logística.

Existen instituciones donde grupos de investigadores ha propuesto iniciativas tecnológicas en el campo de la educación, con el objetivo de mejorar los procedimientos comunes de aprendizaje en las instituciones, a través del uso de tecnologías semánticas para hacer que los datos del programa de estudios sean comprensibles tanto para los humanos como para las computadoras reutilizando modelos ontológicos existentes de la mejor manera posible para facilitar la inclusión y la interoperabilidad de los nuevos datos en la red de datos (Sequicela, 2018).

Es las instituciones de educación tiene importancia los sistemas que impactan el proceso de enseñanza-aprendizaje como aquellos sistemas de gestión administrativa. Por esta razón se realizó un proyecto donde se desarrolló el software: Sistema de Gestión Académica de la Facultad de Contaduría y Administración (SIGAF), utilizando la metodología ágil SCRUM, dividiéndolo en FRONTEND, BACKEND y Diseño de Base de Datos. En este escrito se aborda las herramientas utilizadas en desarrollo BACKEND.

Backend se define como el área que soluciona procesos de información, fluidez de datos, arquitectura interna y usualmente está basado en servidores remotos (Vega, 2013).

El desarrollo de plataformas web que agilicen y sean eficientes al momento de formar los horarios, y con la aplicación correcta de las técnicas y herramientas de programación se logra construir la arquitectura adecuada que de soporte y estructura a los sistemas web. Debido a la especialización que requieren las aplicaciones, han surgido los profesionistas en BACKEND. En la actualidad, este tipo de profesión es la encargada de desarrollar la arquitectura de programación de cualquier aplicación web, además, el profesionista Backend necesita interactuar con un equipo especializado en diferentes áreas del desarrollo web, como el Frontend, Base de Datos, Análisis, Pruebas y Documentadores. Una de las actividades principales de esta área es unir el modelo de negocio (base de datos) de las aplicaciones con las interfaces donde el usuario interactúa en las plataformas y sistemas web. (Valdés, 2007). Para el desarrollo del proyecto, se diseñó un marco conceptual que se especifica en la figura 1.

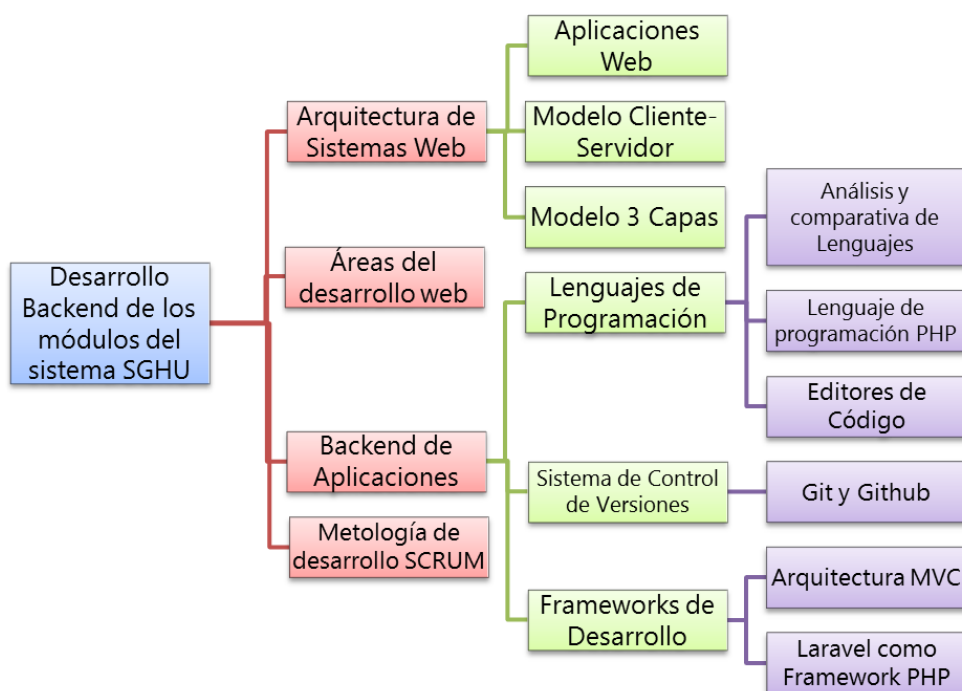


Figura 1 – Diagrama marco conceptual. Fuente Propia.

Existen lenguajes especializados para el desarrollo web, el solo hecho de programar en estos lenguajes, no asegura un orden o una estructura eficaz para el procesamiento de la información, para esto, existen modelos de programación probados y definidos, uno de los más utilizados en la actualidad es la Arquitectura N-Capas,(Paredes,2011) este estilo de programación tiene como objetivo principal, separar los diferentes aspectos del desarrollo, tales como la presentación, lógica de negocio y mecanismos de almacenamiento. Esto surge, a raíz de la desorganización de código que existía en la programación de los sitios web, comúnmente denominado como “código espagueti”, una analogía a como se presenta el desorden en la pasta como alimento (Garrido, 2004).

Los lenguajes especializados para generar aplicaciones web son generalmente permisivos, esto crea una fácil integración con otro tipo de lenguajes, este tipo de característica permite separar una aplicación web completa en varios aspectos o puntos de vista del desarrollo. Desde el punto de vista de la interfaz e interacción existen lenguajes como HTML, CSS, y Javascript, estos forman parte del desarrollo en área de denominada Frontend que solo es visible por el cliente, lo que se denomina como capa de presentación, por otro lado, el área encargada de Bases de Datos permite que la información tenga permanencia en las aplicaciones, comúnmente se denomina como capa de persistencia o datos, y por último, el Backend de una aplicación, área encargada de conectar las bases de datos con el Frontend. La aplicación de este modelo, permite tener una arquitectura robusta, separada que ayuda a la corrección y organización del código que se genera y ejecuta. (Libros Web, 2009).

El proyecto que a continuación se describe, tuvo como objetivo desarrollar la arquitectura de programación y la codificación necesaria para la implementación de cada uno de los tres módulos base que contempla el sistema SIGAF (Sistema de Gestión Académica de la Facultad de Contaduría y Administración). Estos módulos son:

- Manejo de información de los planes de estudios de la facultad que resulte a partir de la creación y gestión de las unidades de aprendizaje asignadas en un período determinado.
- Realizar la carga académica de cada programa educativo del ciclo escolar vigente, basándose en los planes de estudio previamente establecidos por el personal administrativo.
- Capturar y actualizar la disponibilidad docente, formación académica e información relevante a la institución a lo largo de su trayectoria educativa en la facultad, la cual será esencial al momento de la asignación de docentes en la creación de horarios.

Una aplicación web es un programa informático, que es accedido y ejecutado en una red como internet o intranet, mediante una interfaz comúnmente por un navegador web (Alegsa, 2013).

Una aplicación web se define en el sitio Diccionario de informática (Alegsa, 2013) “En general, el término también se utiliza para designar aquellos programas informáticos que son ejecutados en el entorno del navegador (por ejemplo, un applet de Java) o codificado con algún lenguaje soportado por el navegador (como JavaScript, combinado con HTML); confiándose en el navegador web para que reproduzca (rende-ricé) la aplicación” (Herts, 2011).

Una de las ventajas de las aplicaciones web cargadas desde internet (u otra red), es la facilidad de mantener y actualizar, sin la necesidad de distribuir e instalar un software en potencialmente miles de clientes. También tiene como ventaja la posibilidad de ser ejecutadas en múltiples plataformas.

Las aplicaciones web a diferencia de otro tipo de programa tienen las características tal como el usuario puede acceder fácilmente a estas aplicaciones mediante un navegador web (cliente) o similar con acceso a internet se actualizar y mantiene una única aplicación y todos sus usuarios verán los resultados inmediatamente (Alegsa, 2013).

La interfaz gráfica de una aplicación web puede ser compleja y funcional, gracias a las variadas tecnologías web que existen: Java, JavaScript, DHTML, Flash, Silverlight, Ajax, entre otras (Lapiente, 2013). Ajax es un ejemplo de una tecnología de desarrollo web que le da gran poder de interactividad a las aplicaciones web (Microsoft, 2013).

2. Metodología

El desarrollo y gestión del proyecto SIGAF sigue la metodología de SCRUM, esta a su vez se gestiona mediante la herramienta RALLY que esta específicamente diseñada para trabajar sobre SCRUM. En esta herramienta gratuita online (también existen versiones de pago) cada rol del equipo registra su trabajo durante cada sprint, así como el registro de las tareas necesarias para completar cada historia del usuario definida en los sprint's, Referencia (Cohn, 2014).

A continuación se listan las actividades generales del rol de desarrollo Backend de la aplicación (Rally Software, 2015):

- Investigación CVS (Git, Github).
- Investigación del framework LARAVEL para programación MVC.
- Codificación.
- Integración de interfaz y base de datos.
- Pruebas unitarias, depuración, verificación de requerimientos (Pruebas de Caja Blanca).

2.1. Metodología de desarrollo Backend

SCRUM es utilizado como línea general para el ciclo de vida del desarrollo de software, dentro de la metodología empleada para desarrollar el Backend de los módulos del sistema SIGAF, se enumeran las actividades generales para el desarrollo de forma secuencial y sistemática para la obtención del código usable, que mantendrá la arquitectura del sistema en cada módulo descrito por el análisis, para el funcionamiento del sistema en general, la metodología está enfocada a un estudio de caso, por lo que, está dirigida a comprender la dinámicas presente en un contexto singular según (Piedad, 2006). Esta particularidad hace que la metodología del estudio de caso investigado varíe acorde a la disciplina en este caso las ciencias computacionales. Para el desarrollo del sistema SIGAF se seguirá la pauta del modelo propuesto en el fundamento teórico del MVC o Modelo Vista Controlador.

Para optimizar la estructura de la metodología y mejorar la presentación del desarrollo, se divide en las siguientes fases los procedimientos y actividades a realizar en cada módulo del sistema SIGAF.

2.2. Fases de la metodología

2.2.1. Fase de preparación del entorno de desarrollo.

Contempla la preparación e instalación del entorno de desarrollo, esto se realiza solo una vez durante toda la vida del ciclo de desarrollo, por lo tanto, se ejecuta esta acción al inicio del proyecto, que consta de las siguientes actividades:

- Instalar un servidor web de prueba para peticiones http donde se harán las pruebas locales.
- Instalar el módulo de php para que interprete el código generado.
- Instalar el servicio de base de datos MYSQL para el almacenamiento y lectura de datos.
- Instalar el framework de desarrollo LARAVEL el cual tiene las siguientes dependencias: Composer como gestor de paquete, php como lenguaje base y un editor de código para la escritura, configuraciones extra como extensiones openssl para comunicación y la activación del módulo de reescritura de las url del servidor web para permitir direcciones amigables.
- Instalar un navegador web para pruebas.
- Probar la página de inicio del LARAVEL para ver que todo esté integrado y funcione correctamente.

2.2.2. Fase de análisis de insumos o entradas necesarias para la codificación.

Para codificar y unificar las interfaces de la aplicación con el modelo de negocio (Base de datos del sistema), en el Modelo-Vista-Controlador, es necesario contar con las siguientes fuentes o entradas:

- El análisis de la lógica de negocio o los requerimientos del sistema enunciados en el documento oficial de recopilación de requerimientos, así como los diagramas de proceso y casos de uso que modelan el sistema, recopilados durante la fase de análisis.
- Las interfaces o vistas de la aplicación, como es un sistema web se refiere a los documentos fuentes HTML que contienen la maquetación del diseño gráfico en plantillas funcionales para que los usuarios interactúen con el sistema.
- Los scripts del modelo de negocio (Base de datos), así como también los diagramas entidad relación que describen el comportamiento y las relaciones de las entidades dentro del contexto del sistema.
- Minutas de cambios y acuerdos realizados en las reuniones periódicas que describe la metodología SCRUM.
- Una vez obtenidos los insumos se procede al análisis de cada uno, para relacionarlos mediante el modelo descrito con anterioridad, esta actividad de recolección de insumos se realiza por cada módulo durante el proceso de desarrollo.

2.2.3. Fase de emparejamiento y cotejo de los insumos o entrada de proceso.

Las siguientes actividades aseguran que la información de los insumos mantenga una consistencia y coherencia para unificarlos en el modelo de programación:

- Verificar que los datos descritos en el análisis de requerimientos se cumplan tanto a nivel de modelo (base de datos) como a nivel de vista (interfaces), esto quiere decir, que se verifica si existe un tipo de dato en un requerimiento específico, cumpla con un campo en las entidades de la base de datos y que de igual manera sea provisto un control para el usuario en la interfaz de la vista.

- Verificar que las validaciones descritas en el análisis estén contempladas, en primer plano, durante la ejecución de acciones en la interfaz del usuario y en segundo plano, durante la integridad y mecanismos de control de la base de datos, para asegurar el almacenamiento de la información correctamente.
- Si existe alguna discrepancia, comunicar directamente con el encargado del área y resolver las posibles dudas, errores hasta obtener el resultado óptimo antes de empezar con la codificación.

2.2.4. Fase de codificación y desarrollo.

Durante cada módulo se seguirán las siguientes actividades para generar y mantener código sustentable y que tendrá como resultado la funcionalidad de la aplicación:

- Revisar los diagramas de secuencia y verificar mientras se codifica.
- Crear la estructura de organización de los archivos dependiendo el modulo.
- Escribir la parte lógica de la aplicación (controladores) y documentar, los cuales recibirán los datos y serán los encargados de procesar la información.
- Escribir en la parte del Frontend los resultados que arrojen los controladores.
- Escribir las funciones y documentar, estas funciones ejecutarán las acciones del lado del Frontend.
- Hacer pruebas de cada petición y ruta ejecutada por el usuario.
- Depurar la funcionalidad de la aplicación.
- Verificar funcionamiento con equipo de desarrollo y retroalimentar.
- Corregir y verificar de nuevo con equipo de desarrollo hasta que cumpla con el requerimiento aceptado.

2.2.5. Fase de publicación y control de versiones.

Una vez codificadas las funcionalidades de la aplicación se procedió con las publicaciones y fusiones de código con los otros miembros del equipo de desarrollo, esto con el fin de mantener actualizado en todo momento la última versión funcional del sistema, para lograr esto se seguirán las siguientes actividades:

- Cada vez que se logre la funcionalidad especificada en el análisis, se hace una confirmación al repositorio principal del proyecto a través del sistema de control de versiones GIT-GITHUB, esto con el fin de tener una organización a nivel de código del proyecto.
- Si el equipo de desarrollo sube actualizaciones en colaboración al repositorio principal del proyecto, se revisa, analiza y se aprueba para hacer una fusión respecto al código, esto provee una mejora, avance y actualización a la aplicación en desarrollo.
- Revisar periódicamente el repositorio principal para tener actualizada la aplicaciones tanto remotamente en un servidor en la nube como localmente en los equipos de cómputo.

2.3.6. Fase de retroalimentación para la metodología Scrum con usuarios finales y equipo de desarrollo.

Una vez hechas las confirmaciones al repositorio principal del proyecto, se realizan las siguientes actividades para dar seguimiento al proyecto principal:

- Reportar en la aplicación RALLY el avance de la codificación para retroalimentar al equipo.
- Revisar en juntas programadas el avance de la aplicación y los problemas que se presentan durante el avance del desarrollo del proyecto con el equipo de desarrollo y los usuarios del sistema (retroalimentación).
- Las fases descritas con anterioridad se desarrollaran por módulo hasta concluir el sistema, al final, la documentación de las funcionalidades describirá un API el cual podrá consultar el usuario para su posterior actualización o mejoras del sistema.

3. Desarrollo

Para la realización del código se utilizó un entorno de desarrollo adecuado a la arquitectura que se estableció al inicio del proyecto. Mediante el framework Laravel para PHP se puede utilizar el modelo-vista-controlador para centrar los esfuerzos solo en la codificación de la funcionalidad requerida separando apropiadamente las capas del sistema.

El framework Laravel es la médula espinal de la arquitectura MVC utilizando como lenguaje base PHP, su correcta implementación presupone agilidad, desarrollo ágil y código ordenado de acorde a SCRUM, por lo tanto su puesta en marcha y correcto funcionamiento es esencial para continuar con el desarrollo de la codificación.

Una vez analizado y cotejado los insumos se procede a codificar de la manera en que el patrón de la arquitectura indica, para esto es necesario tener en cuenta que se programan 3 capas en el modelo.

1. Los modelos: entidades del modelo relacional.
2. Los controladores: se encargan de manejar la lógica de programación.
3. Las vistas: La interfaz de usuario con que el usuario interactúa con el sistema.

El framework laravel permite crear la abstracción de las entidades de la base de datos, mediante el ORM (Object-Relational-Mapping) ELOQUENT, este permite convertir datos entre el modelo relacional(tablas, relaciones, tipo de datos, etc.) a un sistema de datos orientado a objetos, transfiere las entidades de la base de datos a clases para utilizarlas en un sistema de objetos omitiendo la parte de las conexiones y el SQL propios del lenguaje de consulta, que son necesarios para manipular las entidades del modelo.

La codificación de la inserción del número del Plan de Estudios se procede a crear los modelos en el framework lo que permitirá la interacción entre la aplicación y la base de datos.

Al término del desarrollo se tiene la estructura MVC en Laravel del módulo con la funcionalidad descrita en el análisis de Disponibilidad Docente y la funcionalidad de la interfaz interactuando con la base de datos.

4. Resultados

4.1. Resultados de Funcionalidad del Backend

Una vez desarrollado las fases de la metodología, y teniendo la retroalimentación del equipo de los problemas encontrados en la funcionalidad, el código en ejecución cumple con la

funcionalidad requerida descrita en el análisis, y el usuario final puede interactuar con los módulos mediante los controles de la interfaz, se anexa el código para este resultado.

El código generador provoca que a través de la interfaz el usuario pueda:

- Registrar un Plan de Estudios
- Crear unidades aprendizaje en los planes de estudio.
- Consultar los planes de estudio y las unidades aunadas ahí.
- Modificar los atributos de las unidades de aprendizaje.
- Eliminar de los planes de estudios una unidad de aprendizaje.
- Asignar unidades de aprendizaje a diferentes programas educativos.
- Agregar coordinaciones, números de plan de estudios, etapas y demás catálogos relacionados.
- Consultar las unidades de aprendizaje de un plan y visualizar en forma de listado cada una.
- Consultar los diferentes planes de estudios por carrera, etapa, unidad, tronco común y coordinación.

4.2. Resultados generales de la arquitectura

Durante el desarrollo de la aplicación se desarrolló un API para retornar información en formato JSON a una petición HTTP, esto con el fin de reutilizar la información en cualquier lenguaje o plataforma que no sea SIGAF.

El framework Laravel mapea todas la tablas de la BD a clases para tener un entorno de desarrollo más limpio y no mezclar consultas sql en el código de producción (Gilmore, 2015). También al termino del desarrollo se concluye con una estructura clara y concisa que no permite visualizar los elementos de programación como fragmentos de código individuales, capaz de sufrir modificaciones sin alterar el flujo de la aplicación.

La interactividad de la aplicación fue desarrollada con Javascript y jQuery para solucionar de forma práctica la dinámica entre usuario-sistema. Al término de los módulos el sistema quedó con la siguiente estructura que permite toda la funcionalidad requerida en el análisis.

5. Conclusiones

Durante el desarrollo de la aplicación se construyó un API (Interfaz de Programación de Aplicaciones) para retornar información en formato JSON (Notación Objeto de JavaScript) a una petición HTTP mediante el framework Laravel y tecnologías como Javascript, HTML5, CSS3, esto brinda a futuros desarrollos del sistema SIGAF la posibilidad de crear diferentes aplicaciones para diversos fines, con el añadido de utilizar cualquier tecnología Frontend para consumir los datos. Esto es posible gracias al equipo completo de desarrollo y a las habilidades de cada miembro.

El uso del Modelo Vista Controlador (MVC) que se utilizo, permitió que la arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos, se desarrollaran de forma independiente y muy coordinada intercambio la información necesaria para estar en completa armonía logrando el funcionamiento perfecto al articular cada sección.

Como oportunidad de aprendizaje es importante la integración del equipo de desarrollo, el presente trabajo solo forma una parte del sistema en su conjunto y es el resultado de la construcción de otras áreas como la creación de la base de datos, Frontend, por ello la comunicación entre los miembros del equipo es vital y necesaria, las metodologías describen un área de oportunidad muy grande para facilitar la creación de un sistema, sin embargo, el que ejecuta dicha metodología son personas por lo que la convivencia es necesaria entre los miembros del equipo, Cabe aclarar que durante el desarrollo de los módulos se presentaron cambios radicales por parte del cliente que repercutieron en el avance del mismo, sin embargo, el uso de la metodología SCRUM permitio la comunicación e hizo posible solventar dichos cambios de forma adecuada.

Los módulos realizados cumplen con las expectativas del proyecto y son la base para la creación de horarios del sistema SIGAF.

Referencias

- Alegsa. (2013, 12 5). *Alegsa*. Retrieved 11 16, 2010 from Aplicaciones Web: <http://www.alegsa.com.ar/Dic/aplicacion%20web.php>
- Cohn, M. (2014, Febrero 24). *User Stories & Back-End Systems*. Retrieved Julio 8, 2014 from Scrum Alliance: <https://www.scrumalliance.org/community/spotlight/mike-cohn/february-2014/user-stories-back-end-systems?feed=Scrum-Alliance-Spotlight>
- Garrido, J. S. (2004). Arquitectura y diseño de sistemas web modernos. *Revista de Ingeniería Informática del CIIRM*, 1-6.
- Garrido, J. S. (2004, Julio 10). *Arquitectura y diseño de sistemas web modernos*. Recuperado el Marzo 20, 2014 de http://pegaso.ls.fi.upm.es/~sortega/html_css/files/Arquitectura_y_diseno_de_sistemas_web_modernos.pdf
- Gilmore, W. J. (2015). *EASY LARAVEL 5*. LEANPUB.
- Hearts, C. V. (2012, Marzo 15). *Lenguajes de Programación*. From Maestros del Web: <http://www.maestrosdelweb.com/editorial/los-diferentes-lenguajes-de-programacion-para-la-web-c106224I/>
- Herts, C. V. (2011, Febrero 8). *Maestros del web*. Recuperado Abril 5, 2014 de <http://www.maestrosdelweb.com/editorial/frontend-backend-duplicando-programacion/>
- Lapuente, M. J. (8 de Diciembre de 2013). *Hipertexto*. Recuperado el 15 de Febrero de 2014, de La interfaz gráfica: <http://www.hipertexto.info/documentos/interfaz.htm>
- Libros Web. (2009, Marzo 7). *Javascript Vanilla*. Retrieved Octubre 3, 2013 from http://librosweb.es/libro/jobeeet_1_4/capitulo_4/la_arquitectura_mvc.html
- McCarthy, P. (s.f.). Obtenido de AJAX for developers: Build dynamic java applications: <http://www.ibm.com/developerworks/library/j-ajax1/>
- Microsoft. (2013, Diciembre 2). From ASP .net: Enhanced Interactivity and Responsiveness. Microsoft: <http://www.asp.net/ajax>

- Narvaez, E., & Piedra, N. (2018) Un enfoque de Linked Data para garantizar la interoperabilidad semántica e integridad de datos académicos universitarios, Recuperado November 30, de <http://ceur-ws.org/Vol-2096/paper5.pdf>.
- Paredes, M. A. (2011, Agosto 10). Arquitectura en Capas. Recuperado Febrero 20, 2014 de <http://arquitecturaencapas.blogspot.mx/2011/08/arquitectura-3-capas-programacion-por.html>
- Piedad Cristina Martínez Carazo. (2006, Marzo 12). *El método de estudio de caso*. Recuperado Septiembre 15, 2014 de http://ciruelo.uninorte.edu.co/pdf/pensamiento_gestion/20/5_El_metodo_de_estudio_de_caso.pdf
- Rally Software. (2015, Enero 22). From <https://www.rallydev.com/>
- Saquicela, V., Orellana, G., Piedra, N., Orellana, M., & Espinoza, M. (2018) Similarity Detection among Academic Contents through Semantic Technologies and Text Mining. Recuperado Noviembre 30, de <http://ceur-ws.org/Vol-2096/paper1.pdf>.
- Valdés, D. P. (2007, Noviembre 2). *Los diferentes lenguajes de programación para la web*. Recuperado Septiembre 1, 2013 de MAESTROS DEL WEB: <http://www.maestrosdelweb.com/los-diferentes-lenguajes-de-programacion-para-la-web/>
- Vega, F. (5 de 12 de 2013). *Backend Profesional*. Recuperado de <https://cursos.mejorando.la/cursos/backend-online/material/un-backend-profesional/>
- Vega, F. (5 de 12 de 2013). *Cristalab*. Recuperado de <http://www.cristalab.com/blog/que-significa-backend-y-frontend-en-el-diseno-web-c106224/>

© 2019. This work is published under
<https://creativecommons.org/licenses/by-nc-nd/4.0>(the
“License”). Notwithstanding the ProQuest Terms and
Conditions, you may use this content in accordance with the
terms of the License.