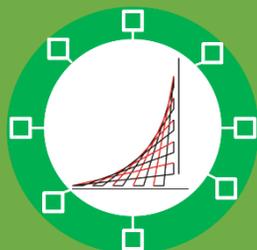


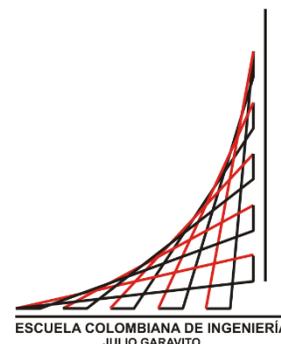
2015



Sistema Integrado para la Administración del Laboratorio de Informática: SILABINFORMA

LIBRO DE PROYECTO

LAURA CATALINA HERRERA CORREA
DIEGO ALFONSO PRIETO TORRES
SEBASTIAN CAMILO MARTÍNEZ REYES





**SISTEMA INTEGRADO PARA LA ADMINISTRACIÓN DEL
LABORATORIO DE INFORMÁTICA**

SILABINFO

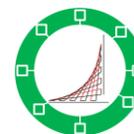
PRESENTADO POR:

**Laura Catalina Herrera Correa
Sebastian Camilo Martínez Reyes
Diego Alfonso Prieto Torres**

DIRECTOR DE PROYECTO

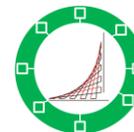
Ing. Rodrigo López Beltrán

**ESCUELA COLOMBIANA DE INGENIERIA JULIO GARAVITO
INGENIERÍA DE SISTEMAS
PROYECTO DE GRADO 2
MAYO 15 DE 2015
BOGOTA D.C**



Contenido

I.	PARTE I	5
1.	Contexto del proyecto	5
1.1	Objetivo general	5
1.2	Objetivos específicos.....	5
1.3	Alcance	5
2.	Estado del Arte: Sistema Integrado para la Administración del Laboratorio de Informática... 6	6
3.	Software definitivo.....	8
3.1	Antecedentes	8
3.2	Software actual	9
3.3	Implementación	12
4.	Plan de continuidad para el proyecto.....	13
II.	PARTE II	16
1.	Experiencias y aportes de incidentes.....	16
1.1	Definición del concepto: Periodo Académico	16
1.2	Diseño y construcción de horarios para monitores	18
1.3	Despliegue y puesta en marcha del sistema	19
2.	Selección de la infraestructura digital de trabajo	20
2.1	Capa de persistencia	20
2.2	Capa lógica	22
2.3	Capa Web	22
2.4	Transversales o de apoyo	25
III.	CONCLUSIONES.....	26
IV.	BIBLIOGRAFÍA.....	27
V.	ANEXOS	27



INTRODUCCIÓN

El presente trabajo se refiere al Proyecto de Grado llamado Sistema Integrado para la Administración del Laboratorio de Informática (SILABINFO), desarrollado en la Escuela Colombiana de Ingeniería Julio Garavito, respondiendo así al objetivo de profundizar en las áreas del conocimiento de la Ingeniería de Sistemas, y por la naturaleza del proyecto, específicamente en la línea del Desarrollo de Software.

El Proyecto de Grado nace a partir de una problemática real que se vive dentro del Laboratorio de Informática, oficina que busca dar respuesta a sus complejidades administrativas por medio del desarrollo de un software, cuyo objetivo debe ser mejorar los servicios prestados por parte del Laboratorio.

El desarrollo de este proyecto de grado se realizó en torno al interés de lograr contribuir a la comunidad académica de forma activa, no solo pensando en el desarrollo de un software, sino teniendo especial cuidado en la metodología para tal fin, ya que, como gran antecedente a este proyecto es la existencia de una solución de software similar, que por distintos problemas asociados a su desarrollo produjeron su desuso.

Desde el punto de vista académico el interés está basado en la profundización de conceptos como las arquitecturas de software, las metodologías ágiles de desarrollo así como la investigación e implementación de nuevas tecnologías que aseguren la construcción de una solución de software funcional que perdure en el tiempo. Desde el principio del proyecto, el grupo de trabajo fue consciente del alcance limitado que tendría el desarrollo de la plataforma SILABINFO debido a las restricciones de tiempo; por lo tanto, este proyecto representa la piedra angular de un trabajo que debe seguirse nutriendo con las capacidades y aptitudes de los estudiantes de la Universidad.

Finalmente el ámbito profesional fue una gran directiva para motivarnos a desarrollar este proyecto de grado, ya que, todos pretendemos continuar por la línea de desarrollo de software y vimos este trabajo como una oportunidad de aprendizaje que nutre nuestro desarrollo profesional.

Este trabajo está dividido en dos partes, la primera de ellas tiene como objetivo describir el resultado final de todas las actividades realizadas tras un año de trabajo y los planes que como autores del proyecto podemos sugerir para una posible continuación del mismo.



Por otro lado, la segunda parte de este documento es una descripción de las decisiones que como grupo de trabajo tuvimos que tomar en determinados momentos que consideramos fueron cruciales y nos llevaron a culminar con éxito los objetivos planteados. Dentro de esta sección se tratan temas que se salen del carácter técnico del proyecto y abordan temas de diseño de la plataforma, errores de comunicación e incluso inconvenientes que vivimos al momento de la puesta en marcha de la plataforma final.

Como estudiantes esperamos que disfruten leyendo este trabajo y sea una base de conocimiento para las personas que estén interesadas en la Ingeniería de Software, ya que compartimos con ustedes a través de este documento nuestras experiencias en torno al desarrollo de este proyecto, que nos enseñó mucho acerca de lo que puede ser la vida profesional. Esas enseñanzas están plasmadas en las conclusiones, las que les sugerimos leer con detalle.



I. PARTE I

1. Contexto del proyecto

1.1 Objetivo general

El proyecto SILABINFO consiste en el diseño, desarrollo y puesta en marcha de una solución de software para el Laboratorio de informática de la Escuela Colombiana de Ingeniería Julio Garavito, lo que permitirá la administración de la operación del Laboratorio y los recursos con que éste cuenta.

1.2 Objetivos específicos

- Revisar el análisis y diseño definido por el Laboratorio.
- Construir un sistema de información que responda a las necesidades actuales del Laboratorio de tal manera que soporten la operación del mismo, de acuerdo al alcance definido en la primera etapa.
- Realizar implantación y pruebas en el Laboratorio, de acuerdo al alcance definido en la primera etapa.
- Levantamiento de Línea de producción para el sistema.

1.3 Alcance

El alcance que se definió debía cumplir con los objetivos, resumidos en tres grandes aspectos: Diseñar una arquitectura de software flexible que pudiera asegurarnos la construcción de una plataforma funcional, mantenible y usable. Para ser coherentes con ello se plantearon los siguientes grupos de funcionalidades:

- Módulo de Convocatorias.
- Módulo de Monitores.
- Módulo de Actividades.
- Módulo de Ayuda.
- Módulo de Activos.
- Módulo de Administración de la aplicación.



El alcance fue definido con respecto a las principales necesidades descritas por las personas encargadas de administrar el Laboratorio de Informática.

De los módulos que se plantearon, solo uno, el módulo de ayuda, no fue posible construirlo, lo que evidencia en cierto sentido un error de planeación por parte de nosotros, pero por otro lado podemos asegurar una completitud del 85% del proyecto satisfactoriamente.

Vale la pena resaltar que el objetivo del proyecto también incluye futuros desarrollos por parte de nuevas generaciones de estudiantes, lo cual fue un criterio fundamental para el diseño de la arquitectura. Más adelante en este documento se incluyen guías para los desarrolladores que pretendan adicionar o mejorar el SILABINFO actual.

2. Estado del Arte: Sistema Integrado para la Administración del Laboratorio de Informática

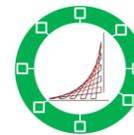
El Laboratorio de Informática de la Escuela ha tenido diversos tipos de aplicaciones que han sido implementadas para suplir sus necesidades pero las cuales han quedado sin continuidad y evolución a través de los años. Una de dichas aplicaciones es SILABINFO, un sistema que al igual que el que presentamos como producto, surge a partir de un proyecto de grado pero quedó obsoleto debido a cambios en procesos y servicios ofrecidos por el Laboratorio.

Debido a esta problemática el Laboratorio da a conocer la necesidad de crear un sistema que permita satisfacer procesos y servicios ofrecidos teniendo como principios clave: flexibilidad, funcionalidad y extensibilidad.

A partir de ello, se plantea este proyecto de grado cuyo desarrollo fue dividido en dos periodos académicos. Para comenzar el segundo periodo académico, obtuvimos como punto de partida lo siguiente:

1. Productos

- Definición y análisis de los requerimientos del sistema.
- Acta de Inicio
- Bitácora
- Definición de alcance



- Diseño de la aplicación: Diseño con el cual se implementará la plataforma teniendo en cuenta los artefactos generados desde los requerimientos del sistema.
 - Documento Arquitectónico
 - Diagramas de dominio
 - Historias de usuario

 - Desarrollo de la plataforma: La metodología y las herramientas adecuadas para la creación del software en el tiempo estipulado.
 - Manuales de instalación de la herramienta y creación de proyectos.
2. Definición de tecnologías y herramientas a usar en la etapa de desarrollo del producto.

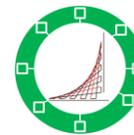
Adicionalmente, se presentaron una serie de soluciones a problemáticas que ocurrieron a lo largo de la primera etapa del proyecto:

- Problema: Actualmente Existe un mecanismo de autenticación estable y compatible, el cual será usado en el desarrollo y operación de la plataforma, por otro lado el Laboratorio con apoyo de estudiantes de proyecto de grado está realizando la centralización de los procesos de autenticación. Esto tiene implicaciones directas en el proceso de desarrollo de la plataforma, pues no debe existir acoplamiento entre el sistema y la manera de cómo se realiza la autenticación y autorización.

Solución: Se ha propuesto un nivel de desacoplamiento a todos los niveles de la aplicación, pues las necesidades del Laboratorio así lo requieren, esto incluye la manera en como la aplicación realiza la autenticación de usuarios. Se han definido contingencias y como criterio de aceptación el soporte del sistema al nuevo método de autenticación.

- Problema: Se han presentado inconvenientes a la hora de realizar la implementación de la línea de producción en los servidores del Laboratorio, la naturaleza de los inconvenientes provienen por desconocimiento técnico sobre el sistema operativo Slackware, encontramos aparente incompatibilidad entre algunos de los componentes de las nuevas versiones de Java y Angular JS con la distribución.

Solución: Se han definido estrategias para la segunda etapa, que incluyen el apoyo del personal del laboratorio, el apoyo de profesores, como el



Ingeniero Gerardo Ospina, investigación por parte del equipo de proyecto. Al día de hoy se ha tenido avance con la instalación de componentes gracias a la investigación realizada.

- Problema: Se presentaron inconvenientes a la hora de realizar la recuperación de las fuentes y manuales correspondientes a versiones anteriores de SILABINFO, los materiales provistos por la biblioteca de la universidad resultaron ser insuficientes, pues el medio de almacenamiento (CD-ROM) entregado se encontraba dañado.

Solución: Gracias al apoyo del cuerpo de monitores y personal del Laboratorio, hemos podido obtener información acerca de las versiones anteriores de la plataforma.

A partir de lo anteriormente descrito se llegó al siguiente plan de trabajo para el segundo periodo académico:

- Continuar con el desarrollo de los módulos de gestión de activos y reservas; por otro lado se realizará el primer lanzamiento de la aplicación y ponerla a disposición de la administración del Laboratorio con el propósito de la realización de pruebas. Se espera realizar el lanzamiento preliminar a mediados del primer tercio, el lanzamiento oficial se realizará en durante la semana dieciséis del semestre 2015-I.

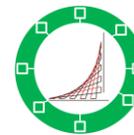
3. Software definitivo

3.1 Antecedentes

En el año 2009, dos estudiantes de Ingeniería de Sistemas dieron por terminado el proyecto que sería la primera versión estable de la plataforma SILABINFO, el cual tuvo como objetivo apoyar la administración del Laboratorio de Informática, mejorando su gestión y el servicio prestado a sus usuarios.

La plataforma fue desplegada en el servidor de producción del Laboratorio de Informática con éxito y durante algún tiempo soportó las operaciones y servicios para los que fue construida.

Sin embargo, el cambio de requerimientos funcionales, así como evoluciones tecnológicas, dieron lugar al nacimiento de modificaciones que debieron hacerse sobre dicha versión de la plataforma; estos cambios fueron presentándose de forma cada vez más frecuente y sin un correcto control



generando que SILABINFO empezara a presentar problemas de consistencia que lo llevó a fallar en la prestación de sus servicios.

Adicionalmente, la arquitectura de software diseñada en 2009 se desfiguró y fue necesario hacer un levantamiento de diagramas de dominio, de clases y de secuencia; reflejando así un problema muy serio para poder asegurar la mantenibilidad del software y su posterior evolución.

Finalmente desde el primer semestre de 2014 la plataforma SILABINFO no fue capaz de soportar las convocatorias para ese periodo académico y así mismo cayó en desuso, lo que obligó a las personas encargadas del Laboratorio de Informática retomar sus antiguas metodologías de administración e incluso a soportar muchas de ellas de forma manual.

Debido a lo anterior y a la falta de divulgación sobre el sistema implementado, la comunidad académica actual desconoce totalmente su existencia, no se volvieron a plantear proyectos para rescatar la plataforma debido a problemas de acoplamiento con tecnologías que ya no son enseñadas en la Universidad y por lo tanto no hay estudiantes que puedan asumir esta labor; bajo ese contexto inicia el proyecto de grado actualmente descrito.

3.2 Software actual

Como se menciona en la sección de alcance, se ha logrado completar un 85% de los módulos propuestos y a continuación se hace una breve descripción funcional de cada uno de ellos así como una ficha técnica de las diferentes tecnologías que fueron usadas en el desarrollo.

a. Módulo de Administración del sistema

- **Perfiles:** Permite administrar los diferentes perfiles que existen en la aplicación, es decir crear, editar los permisos asociados a cada perfil o consultar el nivel de autorización de cada uno de ellos.
- **Usuarios:** Permite consultar la información sobre cada uno de los usuarios de la aplicación así como enviar correos electrónicos a un determinado usuario. Permite cambiar el perfil asociado a cada usuario con el fin de aumentar o disminuir sus capacidades operativas en la plataforma, incluso es posible volver el estudiante automáticamente un monitor sin necesidad de que este aplique a una convocatoria.



- **Administración General:** Dentro de esta sección se configuran los mensajes de correo electrónico automáticos que la plataforma envía, así como el periodo académico sobre el que se está trabajando, los archivos que se usan dentro de toda la aplicación y finalmente la configuración de horarios para los monitores.
- **Horarios:** Esta sección responde a la capacidad de administrar el horario en general de todos los monitores para consolidar el horario del Laboratorio de Informática.

b. Módulo de convocatorias

- **Crear Convocatoria:** Permite crear una nueva convocatoria para un periodo académico definido.
- **Editar Convocatoria:** Permite editar las características de una convocatoria, por ejemplo aplazar la fecha de finalización o cerrar la convocatoria definitivamente.
- **Postularse a una convocatoria:** Un estudiante puede postularse a una convocatoria de monitoria diligenciando un formulario.
- **Ver Convocatoria:** En esa sección además poder consultar la información básica de una convocatoria, es posible ver la información de los estudiantes que se han postulado para la misma, con las opciones de aceptarlos o rechazarlos para el cargo de monitor.

c. Módulo de Monitores

- **Horario:** El monitor es capaz de administrar su propio horario, editando las actividades en cada sesión así como la naturaleza de la misma. Igualmente es posible consultar el horario de un determinado monitor.
- **Bitácora:** En ella el monitor consigna las actividades por cada sesión de trabajo, estas entradas a la bitácora pueden ser consultadas.
- **Comentarios:** Con el objetivo de mejorar de forma continua el servicio prestado por los monitores, es posible hacer comentarios sobre un monitor y consultarlos.



- **Evaluar monitor:** Es una actividad que se lleva a cabo cuando se quiere finalizar el proceso de monitoria, al monitor se le asigna una calificación cualitativa y cuantitativa respecto a su desempeño en el periodo.
- **Ver monitor:** En esta sección se puede consultar la hoja de vida de un monitor para validar todas las actividades que ha desempeñado en este cargo así como las evaluaciones que se le han hecho en cada una de ellas.

d. Módulo de Actividades

- **Crear Actividad:** Permite crear una actividad dentro del periodo académico definido.
- **Editar Actividad:** Permite editar la información básica de una actividad y además de ello adicionar monitores como participantes dentro de la actividad; así mismo permite desligar monitores de una actividad añadiéndoles una calificación a su trabajo realizado.
- **Ver Actividad:** En esta sección es donde se genera la interacción entre los diferentes participantes en la actividad a través de la posibilidad de adicionar comentarios que enriquezcan el trabajo realizado, además es posible consultar la información relacionada a la actividad.

e. Módulo de Activos

- **Ver Activos:** En esta sección se pueden ver los diferentes activos que hacen parte del Laboratorio de Informática discriminados por la categoría a la que pertenecen.
- **Prestar Activo:** Permite prestar un activo a un determinado usuario, adicionando información referente a su préstamo.
- **Bitácora del Activo:** Permite consultar todos los sucesos que ocurrieron con el activo, registrando desde su llegada hasta su salida del Laboratorio de Informática, así como los momentos en los que ha sido prestado, dañado o permanecido en mantenimiento.



3.3 Implementación

Problemática

Cuando se empezaron a definir las tecnologías que serían usadas para desarrollar SILABINFO, MySQL fue la base de datos escogida para soportar los datos de las operaciones de la plataforma, sin embargo, un riesgo no previsto en la definición inicial del proyecto fue la compra del motor de base de datos por la compañía Oracle, conocida por su potente motor de base de datos homónimo que entre otros detalles es pago y no libre como lo es MySQL y que termina siendo su mayor ventaja, ya que como estudiantes no podíamos pensar en una gran inversión de recursos económicos en el desarrollo de SILABINFO.

Por tal razón, la directora del Laboratorio de Informática manifestó su interés de cambiar de base de datos a MaríaDB, un motor de base de datos con características similares a MySQL.

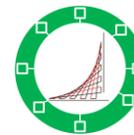
Pese a este requerimiento, nosotros investigamos los términos de la compra por parte de Oracle y fue gratificante saber que MySQL mantendrá sus características en cuanto su licencia, por tal motivo, sugerimos continuar el desarrollo de la plataforma con la misma base de datos, pero teniendo más cuidado de no acoplar en ningún momento la Base de Datos con la aplicación, para poder responder de forma segura a un eventual requerimiento de migrar la base de datos a otro motor.

Hoy en día MySQL se encuentra instalado en los servidores de producción del Laboratorio de Informática así como en máquinas que se usaron para el desarrollo, funcionando al 100% de sus capacidades ofreciendo las mismas características con las que contaba anteriormente.

Desarrollo

El trabajo realizado a nivel de base de datos se divide en tres grandes componentes, los cuales son: un buen diseño del modelo entidades, construcción y comunicación con la aplicación.

Con respecto a los diagramas de entidades se realizaron con ayuda de la herramienta Workbench, que permite trabajar con Bases de Datos MySQL; allí se plasmaron las entidades y relaciones necesarias para soportar los datos



de la plataforma SILABINFO, cuidando de hacer la abstracción fiel del contexto real.

En la parte de construcción se ubica lo que es la codificación de la base de datos, el SQL, conocido como la dimensión física de la base de datos. Se dividió esta etapa en dos DDL, donde están almacenados los scripts del esquema, la creación y eliminación de tablas; por otro lado en el DML está el script de carga inicial, necesario para que la plataforma opere y unos disparadores que se encargan de mantener las reglas del negocio.

Por último la codificación con la aplicación desarrollada en Java se realizó haciendo uso del ORM (Object Relational Mapping) Hibernate, con el cual a través de un archivo de configuración se podrán modificar las opciones de conexión.

Producción

Para el ambiente de producción se creó un nuevo esquema para la plataforma SILABINFO dentro del servidor del Laboratorio de Informática, las credenciales de la base de datos son administradas por la misma oficina.

4. Plan de continuidad para el proyecto

Es importante mencionar que el futuro del Proyecto de Grado se ve enmarcado en varias dimensiones, las cuales abarcan nuevos desarrollos, el inicio de una línea de producción de Software formalizada y una fuente de conocimiento con respecto a buenas prácticas de diseño y desarrollo de software.

Con respecto a los nuevos desarrollos, fue posible evidenciar por medio de las sustentaciones en Vitrina Académica, las oportunidades de mejora y crecimiento de la aplicación, ya que asistentes, en su mayoría estudiantes y profesores, mencionaron interesantes módulos que consideramos son una gran directriz para guiar la continuidad del proyecto de grado.

Nuevos módulos

- **Módulo de Estadísticas:** Para un usuario con perfil administrador, sería de mucha utilidad ver el rendimiento y uso de la plataforma a través de una serie de directrices sobre las operaciones que se pueden llevar a cabo y son relevantes para un mejor funcionamiento del Laboratorio de Informática, solo por mencionar algunos ejemplos: contar con el nivel de cumplimiento



de cada monitor con respecto a las actividades que se le han asignado y así poder tener una perspectiva más objetiva sobre su evaluación.

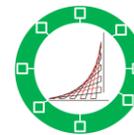
La propuesta es construir un módulo adicional llamado *Estadísticas* que me permita consultar la información y generar reportes, los cuales se puedan descargar.

Para ser consecuentes con esto, se debe hacer primero un trabajo de investigación para identificar la información relevante, así como los cambios necesarios que se deben hacer en el modelo de datos para soportar esta nueva característica.

- **Módulo de Certificaciones:** Este nuevo módulo estaría muy ligado con los monitores, ya que sería útil poder expedir certificaciones automáticas sobre una monitoria realizada sin tener que esperar a que el administrador se decida a redactarla de forma manual, por el contrario, cada estudiante en un momento determinado podría descargar las certificaciones que validan su desempeño como monitor en un momento determinado, así como las actividades que realizó en ese periodo.
- **Aplicación Móvil:** Aunque la plataforma web fue desarrollada usando Bootstrap, una tecnología que se adapta al tamaño de la pantalla desde donde se esté accediendo, hay algunos detalles visuales y funcionales que no son tan cómodos de utilizar accediendo desde el navegador del celular. Algunos estudiantes les interesó hacer la respectiva migración de la plataforma web a una aplicación móvil, nosotros hacemos una recomendación: no vale la pena hacer un desarrollo individual para cada uno de los sistemas operativos móviles disponibles en el mercado como Android e IOS por ejemplo, es mejor buscar un framework que permite adaptar el desarrollo ya hecho e integrarlo en el sistema operativo, como por ejemplo, Cordova¹, una tecnología que está evolucionando y la consideramos muy útil para este propósito.

La plataforma SILABINFO, fue desarrollada aplicando conceptos de varias áreas del conocimiento y procurando cuidar la calidad respecto a las buenas prácticas de desarrollo; por lo tanto, la arquitectura definida y el código escrito son fuente importante de investigación para los estudiantes que se están formando, así que el éxito de este proyecto de grado depende de un factor más allá de la aceptación y el uso de la herramienta en la comunidad; es imperante que se divulgue la posibilidad de trabajar con este código a modo de investigación bien sea para

¹ Apache Cordova: Es una plataforma que permite la construcción de aplicaciones móviles nativas implementando tecnologías como HTML, CSS y JavaScript.[1]



apoyarse en el trabajo de las asignaturas o como un interés personal de cada estudiante para continuar el desarrollo de la plataforma.

Finalmente, la mejor forma de continuar el proyecto es aprovechar el potencial de la arquitectura de software diseñada para suplir nuevas necesidades que surjan en el Laboratorio de Informática y permitir la evolución de la plataforma conforme las tecnologías de desarrollo lo hagan y nunca perder de vista el objetivo principal que es prestar un servicio a la comunidad educativa.



II. PARTE II

1. Experiencias y aportes de incidentes

No podemos decir que la arquitectura diseñada fue producto de una sola tarde de trabajo que nunca tuvo errores, por el contrario, fueron necesarios varios periodos de análisis para depurar e incluso hubo momentos en los cuales fue necesario rediseñar completamente módulos debido a nuevos requerimientos por parte del cliente (Director del Laboratorio) así como de reglas de negocio que no se tuvieron en cuenta y demás razones que queremos compartir con ustedes a continuación como parte del conocimiento adicional y muy valioso que nos dejó el desarrollo de la plataforma SILABINFO.

1.1 Definición del concepto: Periodo Académico

Contexto

Las fechas son una herramienta comúnmente usada para hacer el ejercicio de auditoría en los sistemas de información, en nuestro caso particular las usábamos para restringir los tiempos en los cuales una convocatoria, actividad, monitor o préstamo de un activo eran válidos; sin embargo se dieron tres eventos que casualmente fueron descubiertos al tiempo y obligaron a replantear dicha metodología: la necesidad eliminar los permisos que un monitor había adquirido dentro de la plataforma a raíz de haber sido aceptado como monitor; segundo, cerrar todas las actividades iniciadas una vez el periodo académico se finalizará y por último poder generar un reporte del estado de los activos del Laboratorio una vez ese periodo académico llegará a su fin.

Nació la necesidad entonces de identificar el periodo académico, como una entidad que describiera más que un par de fechas que indicaran el inicio y fin de un evento, ahora debía contar con un estado, que nos permitiera saber qué operaciones se podían hacer o no dependiendo del momento por el que estuviese pasando la Universidad.

Solución

La solución debería considerar el particular proceso que el Laboratorio de Informática usa para administrar sus monitores y que inicia con la convocatoria, los siguientes puntos describen lo que se tuvo en cuenta para diseñar la solución:



- No se debe permitir que exista más de una convocatoria abierta por periodo académico.
- La convocatoria de monitores se abre desde el periodo académico anterior, pero los estudiantes aceptados como monitores solo podrán ejercer desde el siguiente periodo académico. Dichos estudiantes, serán aceptados cuando el periodo académico asociado a la convocatoria se encuentre vigente.

Como estos inconvenientes se encontraron a dos meses del despliegue final de la plataforma; la solución debía ser rápida y no requerir un excesivo acoplamiento para que pudiese ser integrada con todas las entidades que necesitarán auditoría directa del periodo académico, por lo tanto se definieron los siguientes criterios de decisión:

a. Solución 1

La primera solución que se planteó fue continuar auditando las entidades a través de las fechas de inicio y fin pero aumentando las restricciones a nivel de Base de Datos en las relaciones que ameritan dicho control.

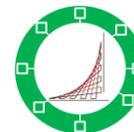
b. Solución 2

La segunda solución era crear una entidad totalmente aparte que representará el periodo académico, describiera su fecha de inicio y fin, así como su estado; es decir realizar la abstracción desde el mundo real y llevarla a la plataforma. Por otro lado debía permitir un control generalizado en la aplicación de tal manera que se evitaran inconsistencias en el uso de la misma.

Análisis de soluciones

Dado que nuestra principal preocupación era lograr crear una aplicación estable y madura que fuera capaz de auditar los hechos que sobre ella eran ejecutados, la primera solución terminó siendo muy laxa para dicho propósito, además requería un trabajo muy grande cuando se quisiera consolidar un reporte en unas determinadas fechas.

La segunda solución es más natural, abstrae la situación real de lo que ocurre con el Laboratorio de Informática sin atarse a fechas determinadas, por el contrario es flexible y permite recopilar toda la información bajo un concepto más amplio que generaliza la forma en la que opera.



Conclusión

Es importante reconocer conceptos que se alejan por completo del ámbito operativo y son útiles para el control y auditoría de los distintos procesos que se pueden ejecutar dentro de la plataforma; de no identificarse a tiempo, este tipo de cambios puede ser costoso y generalmente obligatorios para poder continuar desarrollando un sistema maduro.

1.2 Diseño y construcción de horarios para monitores

Contexto

Uno de los principales objetivos de SILABINFO es poder administrar los monitores de manera efectiva evitando que el personal del Laboratorio continúe realizando procesos de manera manual, lo cual ha generado inconsistencias y dificultades a la hora de gestionar situaciones tal como la creación de horario de servicio para los monitores en el Laboratorio teniendo en cuenta factores como clases, actividades extraescolares y preferencias de cada uno de ellos.

Solución

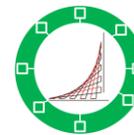
La solución planteada debía contemplar varias situaciones tales como:

- Cambio en la duración de cada sesión de trabajo en un mismo periodo: La configuración de horarios debe soportar una duración de 1.5 horas (lo común a lo largo del semestre) pero en temporada de exámenes finales, una duración de 2 horas.
- Cada monitor puede seleccionar en cuáles sesiones trabajar en su monitoria, indicando de igual manera en cuales momentos de la semana no tiene ninguna disponibilidad, ya sea por clases o eventos extraescolares a los que esté asociado.
- El administrador del Laboratorio debe poder observar las propuestas de horario realizadas por los monitores y crear así el horario definitivo.

Análisis de la solución

De acuerdo a los requerimientos mencionados se planteó la siguiente solución:

Se diseña un componente, representado en una entidad de control, la cual es la base de configuración de los horarios. Dicha entidad (HourConfig) determina



cómo debe construirse los horarios de los monitores de acuerdo a la duración de cada sesión en un determinado momento, supliendo de esta manera el primer requerimiento.

Adicional a la entidad de control se implementaron dos entidades más que permitirían la construcción del horario: la primera de ellas representa cada sesión en el periodo laborado por el monitor (Entry) allí el monitor podrá especificar si en ese lapso de tiempo tiene o disponibilidad para realizar sus labores como monitor; la segunda entidad (Schedule) encapsula todas las sesiones conformando el horario definitivo el cual finalmente se le asociará al monitor.

1.3 Despliegue y puesta en marcha del sistema

Contexto

La plataforma SILABINFO debía quedar instalada en el servidor del Laboratorio de Informática, el cual cuenta con un sistema operativo Linux Slackware y una versión de Java 1.6.0.45. Por error, nosotros no tuvimos en cuenta ese factor y sin precaución desarrollamos haciendo uso de la versión de Java 1.8.0.11. Solo hasta el día en el que debíamos desplegar la aplicación nos encontramos con este problema, que nació claramente de nuestra falta de investigación en cuando a los requerimientos técnicos sobre los cuales debía funcionar la plataforma.

Solución

a. Solución 1

La primera solución propuesta consistió en cambiar el ambiente de desarrollo para que compilara la aplicación bajo la versión de Java 1.6.0.45; previendo problemas en las dependencias de las diferentes librerías asociadas al proyecto, lo cual nos obligaría a modificar las versiones de cada una de ellas hasta que todo el cambio estuviese terminado y asegurando el correcto funcionamiento de los servicios de la aplicación.

b. Solución 2

La siguiente solución contemplaba la posibilidad de actualizar la versión de Java del servidor del Laboratorio de Informática a 1.8.0.11, pero tenía un riesgo muy grande, ya que existía la posibilidad de que al hacerlo, algunos de



los demás servicios que estaban siendo portados por el servidor dejarán de funcionar.

Análisis de la solución

La segunda solución era la más riesgosa desde todo punto de vista, por lo tanto era la menos viable, sin embargo, los administradores del servidor accedieron a hacer una prueba de esa solución sobre una copia del servidor en una máquina virtual. Realizando el ajuste la aplicación pudo ser desplegada sin errores.

Conclusión

Es importante dentro de cualquier desarrollo, contar con una adecuada investigación de tal manera que la información esté lo más completa posible, mitigando posibles errores que pueden llegar a ser bloqueantes y generar reprocesos, los cuales implican desviaciones en la planeación realizada inicialmente.

En esta oportunidad, aunque el riesgo de cambiar la versión en el servidor de producción del Laboratorio fue alta, el resultado fue satisfactorio.

2. Selección de la infraestructura digital de trabajo

La definición de los artefactos o módulos de software concretos, que en un futuro servirían como base para la organización y desarrollo de la plataforma SILABINFO, fue clave y a continuación se menciona la justificación de cada uno de ellos describiendo sus principales ventajas y desventajas.

2.1 Capa de persistencia

Dentro de la capa de persistencia se encuentra el motor de base de datos y el framework que se escogió para la comunicación entre esta capa y la capa lógica.

MYSQL

Las razones por las cuales se decidió usar MYSQL como motor de base de datos tiene que ver en primera medida con su condición de código abierto, aunque el tema entra en discusión dada la compra realizada por Oracle (durante el proceso de desarrollo de SILABINFO), lo cual cambia el tipo de licenciamiento del motor. Sin embargo, con el propósito de brindar



continuidad de la aplicación es posible cambiar el motor de base de datos, con un mínimo impacto, a MariaDB, dada su alta compatibilidad con MySQL.

Además de ello, la facilidad de configuración e instalación de MySQL soporta gran variedad de sistemas operativos, lo cual generó gran impacto en la decisión ya que sabíamos que el servidor de producción contaba con Slackware, pero nosotros como desarrolladores lo haríamos en Windows. Adicionalmente puede ser ejecutado en una máquina de escasos recursos debido al bajo costo en requerimientos para el manejo de base de datos, esto es importante ya que el servidor de producción del Laboratorio soporta otros servicios adicionales y nuestro interés es desarrollar una aplicación web que no necesite gran cantidad de recursos para su operación.

Por otro lado también existen desventajas que al no ser tan críticas no influyeron de manera significativa en la decisión, pero vale la pena mencionarlas, como por ejemplo, un gran porcentaje de las utilidades de MYSQL no están documentadas², tiene una pobre implementación de SQL al no permitir recurrencia ni las opciones de sentencias para bases de datos tipo OLAP; sin embargo cuenta con lo suficiente para crear procedimientos y disparadores que mejoren el desempeño de la base de datos.

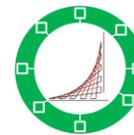
HIBERNATE

Como equipo de desarrollo sabíamos que necesitábamos contar con un ORM³ (Mapeador Objeto-Relacional) ya que la gran cantidad de código que se necesita para programar la correspondencia entre un objeto y una relación es bastante y complejo de mantener, estadísticamente representa el 35% de código de una aplicación y otorga realmente muy poca funcionalidad; por esta razón decidimos adoptar Hibernate como ORM para el desarrollo de SILABINFO.

Entre las ventajas que se derivaron de usar este framework de desarrollo están la facilidad de programación en cuanto al mantenimiento y agilidad con el que podíamos hacerlo, debido a que SILABINFO es una aplicación transaccional sin procesamiento masivo, Hibernate ayudó a hacer más fácil el proceso de traducción Objeto-Relacional, pero el factor decisivo fue la experiencia que teníamos usando este framework que se nos había enseñado previamente en la Universidad.

² Tomado de: Ventajas y desventajas MySQL [3]

³ ORM: Object-Relational Mapping. [2]



Adicionalmente, Hibernate trabaja bajo licenciamiento LGPL, amplia documentación y comunidades activas, lo cual permite un acompañamiento en el desarrollo de la aplicación.

Sin embargo, hubo problemas que hicieron que en determinado punto del proceso de desarrollo pensáramos en dejar de usar esta metodología y optar por una diferente como JDBC; entre estos problemas están la disminución de rendimiento derivada del hecho de usar un framework tan pesado, la gran cantidad de archivos de configuración que debían mantenerse cada vez que algo cambiaba a nivel de base de datos, y, por último, el poco control sobre las entidades básicas y la flexibilidad de operabilidad con la base de datos.

Decidimos seguir usando Hibernate, pero delegar muchas funcionalidades a el motor de base de datos por medio de disparadores y restricciones de tuplas, aunque acoplamos la lógica de negocio a la base de datos, se solucionaban la mayoría de problemas que se nos presentaron y aumentaba el desempeño de las operaciones, por otro lado nos dimos cuenta que un ORM como Hibernate debe ser utilizado bajo una serie de buenas prácticas que eviten el surgimiento de estos inconvenientes.

2.2 Capa lógica

Dentro de la capa lógica encontramos el lenguaje de programación en el cual se encuentra desarrollada la aplicación:

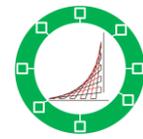
Java

Java es un lenguaje de programación orientado a objetos que tiene entre sus principales características que es multiplataforma, lo cual nos permitió poder desarrollarlo bajo ambientes en sistema operativo Windows pero implantarlo en servidores Linux.

Adicionalmente, el equipo de desarrollo tenía una alta experiencia en la creación de aplicaciones bajo este lenguaje, lo cual permitió un desarrollo ágil y basado en buenas prácticas.

2.3 Capa Web

Realmente en la Universidad se nos enseñó muy poco sobre desarrollo web, por lo cual, dependió de nosotros investigar y evaluar las diferentes alternativas que existían para llevar a cabo esta etapa del desarrollo,



teniendo claro que la usabilidad y mantenibilidad debían ser los atributos de calidad más importantes en este punto.

BOOTSTRAP

Como novatos en el tema de desarrollo y diseño web no sabíamos muy bien qué buscar, pero en las primeras etapas de investigación el concepto de diseño adaptativo dirigió los esfuerzos por encontrar un framework que ofreciera esa capacidad y además que tuviese una curva de aprendizaje corta. Rápidamente nos encontramos con Bootstrap, un framework front-end desarrollado y liberado por Twitter, que ha tenido gran aceptación y es muy utilizado en muchos proyectos por desarrolladores y agencias.

Dentro de sus características más destacadas podemos mencionar la implementación conjunta con HTML5 y CSS3, la inclusión de un innovador concepto de grillas que hace el diseño sencillo para desarrolladores como nosotros que muy pocas veces nos hemos enfrentado a la necesidad de hacer una interfaz cómoda e intuitiva para el usuario; y aunque tiene muchas más ventajas la que más llamó nuestra atención es el conjunto de buenas prácticas sobre el que está basado, que logran llevar al desarrollador a construir una plataforma que perdure en el tiempo.

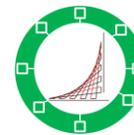
Entre las desventajas que se mencionan cuando hicimos la investigación, las principales consistían a nivel de excesiva personalización para diseñadores web obsesionados con crear sus propios componentes, sin duda alguna una desventaja que no nos preocupó en ningún momento porque sabíamos que dicha utilidad no sería determinante durante el desarrollo.

ANGULARJS

AngularJS es un framework MVC de JavaScript para el desarrollo web front-end que permite crear aplicaciones SPA (Single-Page Applications).

Bajo nuestro punto de vista AngularJS proporciona las siguientes ventajas:

- La separación lógica de la vista, de forma que el código JavaScript es completamente independiente del código HTML, permitiendo desarrollar todo el contexto visual sin depender de la lógica del negocio.
- La comunicación con el back-end, AngularJS cuenta con una implementación de peticiones REST muy fuerte y estructurada que encajaba perfectamente



con la arquitectura planteada para SILABINFO, además de incluir buenas prácticas para el manejo de este tipo de servicios.

- Facilidad a la hora de manipular los elementos del DOM (Document Object Model); uno de los principales problemas con jQuery o javascript es la necesidad de intervenir directamente con el DOM, generando un acoplamiento gigantesco con el código HTML; sin embargo AngularJS abstrae esta funcionalidad en directivas que no solo agilizan el proceso de desarrollo, sino que permite crear código reutilizable construido bajo buenas prácticas que eviten al máximo la manipulación directa del DOM.
- El código es extremadamente fácil de probar; ésta por ser la última capa y la que está de cara al usuario refleja todos los errores, por la misma razón debemos garantizar que la funcionalidad a este nivel sea óptima, AngularJS simplifica la labor ofreciendo un framework de pruebas conocido como Jasmine.
- Es compatible con otros frameworks basados en javascript y no genera incompatibilidad con ningún navegador web.

No podemos decir que no existen desventajas, incluso las que describiremos a continuación hacen pensar en desistir en el uso de un framework como estos.

- AngularJS es un framework mantenido por Google que evoluciona constantemente, dejando obsoletas versiones anteriores y causando problemas en la continuidad de las plataformas web.
- Si no se usa de forma correcta, AngularJS puede causar sobrecarga por lo cual el tiempo de carga resulta pesado para el navegador y de esta forma perdiendo cualquier beneficio del framework.
- AngularJS tiene una curva de aprendizaje alta. Aprender a usar un framework es algo restringido y que únicamente nos sirve para manejar dicho framework, mientras que aprender un lenguaje de programación es algo universal que tendrá múltiples aprovechamientos y aplicaciones.
- Pueden existir dificultades para continuar el desarrollo si los siguientes desarrolladores no conocen el framework o simplemente ya no se presta soporte para dicha versión.



Sin embargo, después de todo el camino recorrido no existe duda sobre los grandes beneficios que nos trajo el haber usado AngularJS, porque pasamos de conocer muy poco sobre desarrollo web a poder generar una plataforma madura y consistente que trabaja de forma eficiente en cualquier navegador y es fácilmente adaptable según las nuevas necesidades que aparezcan.

A modo de experiencia personal, el profesor que nos enseñó JQuery, siempre se quejaba de las complejidades de usarlo y la poca mantenibilidad que tenía una aplicación web desarrollada usando solamente javascript y jquery. AngularJS es quizá el framework que más se aleja de todo lo que alguna vez aprendimos y desarrollamos en la Universidad.

2.4 Transversales o de apoyo

Es importante mencionar que existieron muchas herramientas que estuvieron apoyando el desarrollo de la plataforma SILABINFO de forma transversal las cuales describimos a continuación, identificando su utilidad y los principales beneficios que obtuvimos de las mismas.

YEOMAN, GRUNT, BOWER

Yeoman nos permitió crear el andamiaje (scaffolding) del proyecto web, así como todos los archivos de configuración y el diseño de pruebas para mantener un front-end estable y agradable al usuario.

Con Bower obtuvimos librerías de todo tipo que fueron muy útiles en distintos momentos del desarrollo haciendo sencillo el control de versiones de las mismas y sin necesidad de sobrecargar la carga de la plataforma.

Finalmente Grunt nos permite compilar la aplicación web, haciendo pruebas y verificando que el código escrito tanto en HTML como en javascript sea correcto y no presente errores futuros en ejecución.

APACHE TOMCAT

Se escogió Apache Tomcat como servidor de despliegue por múltiples razones, es altamente configurable, es modular, multiplataforma y principalmente es la herramienta que usa el Laboratorio de Informática dentro de su servidor, pero realmente podríamos enumerar muchas más ventajas que hacen de Apache una gran herramienta de desarrollo.



BITBUCKET

Usado como herramienta para el control de versiones del proyecto, Bitbucket nos facilitó la integración de nuestro trabajo.

APACHE MAVEN

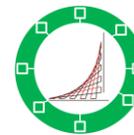
Se eligió Apache Maven como software de gestión de proyecto, el cual nos permitiría darle una estructuración clara y ordenada. De igual manera apoyaría la generación de documentación del proyecto.

TRELLO

Trello es una herramienta usada como mecanismo de comunicación entre el equipo de desarrollo, director y cliente del proyecto de grado. Por medio de ella, se gestionó el proyecto de manera colaborativa apoyando la metodología de desarrollo Scrum (Marco de desarrollo ágil).

III. CONCLUSIONES

- La aplicación de conceptos, metodologías y herramientas relacionadas a la gerencia de proyectos permitió llevar los procesos de desarrollo de software con mayor control y precisión, en cualquier momento es posible conocer el estado exacto del proyecto, las metas inmediatas y las actividades a realizar, le permite además a los interesados, bien sea usuarios, clientes, director de proyecto realizar seguimiento y control sobre el estado de avance, comunicarse de manera sencilla con los miembros del proyecto, de esta manera la comunidad es más cercana a la plataforma .
- Se evidenció la importancia y relevancia de las etapas de diseño y levantamiento de requerimientos en la construcción de sistemas de información, pues la definición del diseño arquitectónico que responda a las necesidades cambiantes del Laboratorio de informática es un proceso que debe ser realizado de manera continua, con una revisión constante y con una metodología definida.
- Se ha evidenciado que el impacto que genera un sistema como SILABINFO en la comunidad del Laboratorio de informática es alto y está relacionado directamente con el conocimiento que tiene la comunidad de la herramienta, además de cómo esta es capaz de cumplir su propósito por medio de la interacción acertada con el



usuario. Por ello se tiene especial atención en el cumplimiento de los atributos de calidad correspondientes a Usabilidad y disponibilidad.

- El entorno del Laboratorio de informática es cambiante, esto lleva a que surjan nuevas necesidades, muchas de ellas relacionadas a la administración y gestión, como implicación de ello SILABINFO debe responder de manera rápida al cambio. Es por ello que el desarrollo se enfocó en que gran parte de los atributos de calidad de SILABINFO se orientaran a la mantenibilidad y escalabilidad.
- El planteamiento asertivo del alcance es indispensable para dar claridad a todos los integrantes e interesados en el desarrollo del proyecto. Esto a su vez impide que se generen requerimientos fuera de tiempo y evita que se generen expectativas no deseadas en la entrega final.

IV. BIBLIOGRAFÍA

1. Apache Cordova. Disponible desde Internet en: <https://cordova.apache.org/>
2. Hibernate. "What is Object/Relational Mapping?" Disponible desde Internet en: <http://hibernate.org/orm/what-is-an-orm/>
3. MySQL. "Ventajas y desventajas". Disponible desde Internet en: <https://mysqldaniel.wordpress.com/ventajas-y-desventajas/>
4. Scrum. "What is Scrum?" Disponible desde Internet en: <https://www.scrum.org/resources/what-is-scrum>

V. ANEXOS

1. Manual de instalación o despliegue
2. Manual de usuario
3. Guía para futuros desarrollos
4. Pruebas y resultados