

ÍNDICE GENERAL

Introducción	1
Antecedentes	1
Justificación e importancia	1
Planteamiento del problema	1
Objetivos	2
Objetivo general	2
Objetivos específicos	2
1. Entorno Empresarial	3
1.1. Fischer, Knoblauch & Co.	3
1.2. Estructura organizacional	4
1.3. Cargo ocupado por el pasante	4
2. Marco Teórico	5
2.1. Conceptos básicos sobre el area de trabajo	5
2.1.1. E-learning o aprendizaje electrónico	5
2.1.2. B-learning o aprendizaje híbrido	6
2.1.3. Sistema de Gestión de Aprendizaje	6
2.2. Desarrollo de <i>software</i>	7
2.2.1. Modelo Vista Controlador	7
2.2.2. Arquitectura cliente-servidor	8
2.2.3. Framework o entorno de trabajo	8

3. Marco Tecnológico	9
3.1. Cliente	9
3.1.1. HTML	9
3.1.2. CSS	9
3.1.3. Javascript	9
3.1.4. Ajax	10
3.1.5. JQuery	10
3.1.6. Bootstrap	10
3.2. Servidor	10
3.2.1. PHP	10
3.2.2. Smarty	11
3.2.3. Microsoft SQL Server	11
3.2.4. Servidor HTTP Apache	11
3.3. Pruebas	11
3.3.1. PHP Unit	11
4. Marco Metodológico	12
4.1. Naturaleza del proyecto	12
4.2. Metodología ágil	12
4.3. SCRUM	13
4.3.1. Dueño del Producto o Product owner	13
4.3.2. Equipo	13
4.3.3. Facilitador o Scrum master	14
4.3.4. Stakeholders o Partes interesadas	14

4.3.5. Eventos	14
4.3.6. Sprint	14
4.3.7. Sprint Planning	14
4.3.8. Daily Scrum	15
4.3.9. Sprint Review	15
4.3.10. Sprint Retrospective	15
4.3.11. Artefactos	15
4.3.12. Product Backlog	16
4.3.13. Sprint Backlog	16
4.4. Pruebas de software	16
5. Desarrollo de las funcionalidades	17
5.1. Primer Sprint	17
5.1.1. Objetivos	17
5.1.2. Actividades	17
5.1.2.1. Familiarización con las herramientas	17
5.1.2.2. Análisis a fondo el funcionamineto del SGA	18
5.1.2.3. Levantamiento de requerimientos	19
5.1.2.4. Exploración de otras plataformas	19
5.2. Segundo sprint	19
5.2.1. Objetivos	19
5.2.2. Actividades	19
5.2.2.1. Amplicación de la base de datos	19
5.2.2.2. Creación del CRUD	20

5.2.2.3.	Integración con google maps	20
5.3.	Tercer sprint	21
5.3.1.	Objetivos	21
5.3.2.	Actividades	21
5.3.2.1.	Análisis de los cursos implementados en el sistema	21
5.3.2.2.	Desarrollo del módulo cursos del tipo seminario	21
5.3.2.3.	Creación del usuario instructor	22
5.4.	Cuarto sprint	23
5.4.1.	Objetivos	23
5.4.2.	Actividades	23
5.4.2.1.	Representación en la base de datos	23
5.4.2.2.	Creación del CRUD	24
5.4.2.3.	Creación de una sesión	24
5.4.2.4.	Actualización de una sesión	25
5.4.2.5.	Borrado de una sesión	25
5.4.2.6.	Visualización de sesiones activas de una ubicación	25
5.5.	Quinto sprint	26
5.6.	Sexto sprint	26
5.7.	Septimo sprint	26
5.8.	Octavo sprint	26
5.9.	Noveno sprint	26
5.10.	Décimo sprint	26

A. Diagramas	28
B. Screenshots de los sistemas	33

ÍNDICE DE FIGURAS

1.1. Estructura organizacional de FKC.	4
A.1. Diagrama de casos de uso.	28
A.2. Diagrama que demuestra los distintos estados en los que se puede encontrar una sesión presencial.	29
A.3. Diagrama UML parcial de la base de datos del SGA de FKC previa al proyecto.	30
A.4. Diagrama UML parcial de la base de datos final del SGA de FKC.	31
A.5. Diagrama UML parcial de la base de datos final del SGA de Bibliomed. . .	32
B.1. Vista del listado de Ubicaciones.	33
B.2. Vista de la edición de una ubicación.	34
B.3. Vista de la creación de un seminario.	35
B.4. Vista del listado de cursos del sistema para el administrador.	36
B.5. Vista de la asignación de un seminario a un grupo.	36
B.6. Vista de la creación de una sesión de un seminario.	37
B.7. Vista del listado de las sesiones de un seminario.	38
B.8. Vista de los cursos de un instructor.	38
B.9. Archivo PDF generado con la lista de los estudiantes de un curso.	39
B.10. Vista de los cursos disponibles para el usuario aprendiz.	40
B.11. Vista de los cursos disponibles para el usuario aprendiz.	40
B.12. Vista del modal mostrado con los datos de la ubicación de la sesión correspondiente.	41
B.13. Vista de una sesión confirmada por el usuario.	41

B.14.Formato de los correos enviados por el sistema.	42
B.15.Vista de las estadísticas mostradas al administrador.	42
B.16.Vista de la adaptación hecha para la creación de seminarios en el SGA de Bibliomed.	43
B.17.Vista de cursos disponibles para el aprendiz antes de la integración de el módulo de seminarios en el SGA de Bibliomed.	44
B.18.Vista de una sesión reservada en el SGA de Bibliomed.	45

CAPÍTULO 5

DESARROLLO DE LAS FUNCIONALIDADES

En este capítulo se describe el proceso de desarrollo del proyecto de pasantía. Realizado bajo las directrices de la metodología SCRUM y a lo largo de diez sprints, comprendiendo las fases: especificación y análisis de requerimientos, diseño e implementación, e implantación de los cambios realizados a Sistema de Gestión de Aprendizaje (SGA) de Fischer Knoblauch & CO (FKC). A continuación, se describen las actividades realizadas en cada fase, las dificultades encontradas, artefactos generados y las soluciones tomadas a lo largo del desarrollo de cada sprint.

5.1. Primer Sprint

5.1.1. Objetivos

- Familiarizarse con el ambiente de trabajo de la empresa.
- Aprender a usar el lenguaje de programación PHP y sus buenas prácticas.
- Analizar a fondo el funcionamiento del SGA a extender.
- Levantamiento de requerimientos del proyecto a realizar.

5.1.2. Actividades

5.1.2.1. Familiarización con las herramientas

El pasante no poseía experiencia previa con el lenguaje de programación usado en la empresa, PHP, por lo que se acordó la exploración de referencias sobre el funcionamiento y el correcto uso de dicho lenguaje.

Se usaron distintos recursos tanto literarios como web, mayormente la página web que contiene la documentación oficial del lenguaje como referencia.

5.1.2.2. Análisis a fondo el funcionamiento del SGA

Para esto el pasante tuvo que instalar las herramientas comunes de desarrollo en inglés, puesto que recibió un ambiente completamente en alemán. Entre estos: sistema operativo, manejador de las distintas bases de datos Microsoft SQL y Microsoft Access, y el navegador.

Una vez instalado el ambiente de desarrollo adecuado el pasante procedió a explorar el sistema. Rápidamente se dió cuenta que el código fuente escrito estaba muy desorganizado. Código alto acoplamiento en el que se mezclaban lógica del negocio con la presentación. constante uso de instrucciones SQL construidas dentro de cada vista susceptibles a inyecciones de SQL. Muy bajo reuso de código a lo largo de la aplicación y técnicas de programación desactualizadas para el código PHP escrito en la actualidad especialmente al momento de recuperar información de la base de datos. El código fuente no describía ninguno de los patrones de diseño que podían ayudar para la construcción de sistemas de este tipo, como composición, observador, entre otros. No existía para el sistema en cuestión ningún tipo de pruebas, ni documentación que apoyara al pasante en esta exploración.

Se descubrió el uso del lenguaje de maquetado Smarty que permite la separación de la capa lógica y la de presentación y se procedió a conseguir referencias para el aprendizaje de esta librería.

Se estudió además el esquema de la base de datos usando la herramienta SQL Management studio que genera automáticamente un esquema visual de la base de datos, donde se buscó entender los patrones con los que fue construida con el fin de mantener consistencia en las nuevas funcionalidades a desarrollar. Entre estas, implementación de las relaciones entre tablas, nombramiento de los campos, así como el tipo y tamaño de los mismos.

Asimismo se analizó la estructura de los archivos, para mantener la misma estructura con la que estaban ordenados, separando los distintos componentes de la aplicación como archivos de código PHP, Javascript, CSS y archivos estáticos. Se evidenció una estructura en el nombramiento de los archivos que se siguió a lo largo del desarrollo, colocando primero el nombre de lo que podría llamarse módulo y luego la acción específica dentro del mismo, por ejemplo: `seminar_session_create`, `seminar_session_update`, `location_create`, etc.

5.1.2.3. Levantamiento de requerimientos

Al terminar el análisis de la base de código y entender a grandes rasgos su funcionamiento y estructura se procedió a hacer el levantamiento de los requerimientos necesarios para la extensión. El objetivo era dividir el proyecto en piezas de funcionalidad con el fin de obtener una visión más clara y objetiva de las necesidades del cliente, así como un mapa que permitiera al pasante crear un plan y una estimación para la realización del proyecto. De esta reunión surgió el diagrama de casos de uso (anexo A.1.)

5.1.2.4. Exploración de otras plataformas

En esta fase también se realizó una investigación sobre la implementación de esta funcionalidad en otros SGA como e-front y moodle con el fin de tener una referencia de un producto que ya se encuentra en el mercado.

5.2. Segundo sprint

5.2.1. Objetivos

- Desarrollo del módulo ubicaciones que sirvan como locación de los seminarios.

En este sprint se desarrollo el módulo de manejo de ubicaciones representando en el diagrama de casos de uso (anexo A.1). Se decidió iniciar con este módulo por ser una funcionalidad aislada, sencilla y componente necesario para la creación de sesiones presenciales. Tomando así un acercamiento de abajo hacia arriba en el desarrollo del proyecto.

5.2.2. Actividades

5.2.2.1. Ampliación de la base de datos

Para soportar esta funcionalidad en la base de datos se creo una tabla llamada ubicación con los datos que parecieron relevantes para el cliente ilustrada en el anexo A.4. Dicha ubicación estaría relacionada con las sesiones, pudiendo una ubicación alojar distintas sesiones. El SGA permite mantener distintos clientes para dar soporte de grandes compañías

con filiales, por lo que las ubicaciones se construyen aisladas del grueso de la funcionalidad (los seminarios) para poder ser usadas luego en funcionalidades futuras o integrada en sistemas de los clientes activos de SGAs de FKC. Tienen una relación directa con la tabla cliente a través de una llave foránea. Para la clave primaria se uso un identificador creciente autogenerado.

5.2.2.2. Creación del CRUD

Se acordó que el manejo de las ubicaciones se llevara acabo en el área de administración de la aplicación, por el usuario administrador. Se agrego una nueva entrada de manejo de ubicaciones en la interfaz del adminstrador con este fin.

Se procedió entonces a la construcción de interfaces que permitieran la creación de una ubicación nueva, listar las ubicaciones existentes, modificar una ubicación existente tanto como eliminar una ubicación. Se hizo énfasis en que las interfaces creadas siguieran un aspecto consistente con las otras funciones de administración ilustradas en los apendices B.1 y B.2.

5.2.2.3. Integración con google maps

Al terminar la funcionalidad básica el dueño del producto sugirió integrar las ubicaciones del sistema con la aplicación google maps. Tarea para la cual el pasante debió investigar sobre el uso de este API. Se logró conseguir un producto gratis de esta aplicación que pemitiera mostrar ubicaciones marcadas en un mapa generado por google enganchado a un iframe en el sistema, con tan solo especificar medidas latitud y longitud, que tuvieron que ser agregadas luego a la tabla ubicación.

Para mantener el uso de esta funcionalidad de manera gratuita para la empresa el usuario debia ingresar los valores de latitud y longitud de la ubicación deseada, para esto el pasante ofreció como solución analizar gramaticalmente la URL mostrada en la aplicación web google maps por medio de expresiones regulares para extraer los valores necesarios, facilitando así al usuario el proceso de agregar una ubicación sin que la empresa tuviera que usar las funcionalidades pagas de google.

Se recomienda en un futuro el pago de este API (Javascript de google maps) si se desea facilitar aun más la interacción con mapas dentro de la aplicación. Con el beneficio de no depender que en algun momento el proveedor google modifique la estructura de sus URL

o desaparezca los datos de latitud y longitud de las mismas.

En el anexo B.2 se muestra como se integro el mapa a la vista de editar ubicación.

5.3. Tercer sprint

5.3.1. Objetivos

- Analizar la estructura de cursos actual del sistema.
- Desarrollo e integración del módulo cursos del tipo seminario.
- Crear un nuevo usuario para el sistema gestionar solo cursos presenciales.

5.3.2. Actividades

5.3.2.1. Análisis de los cursos implementados en el sistema

Luego de examinar las tablas y entender el funcionamiento de los cursos y como son asignados a los grupos (Anexo A.3) se decidió extender esta tabla como una generalización.

El sistema solo daba soporte a cursos del tipo multimedia dentro del mismo sistema, que podian ser de autoría del cliente o externos. Estos dos tipos eran manejados con un booleano “cursoDeAutoria” decisión de diseño que no daba espacio para la expansión de mas tipos de cursos. El pasante sugirió agregar un campo “tipo” a la tabla de Curso asignandole el entero 2 al tipo de curso seminario. dejando los enteros 0 y 1 a los anteriores tipos de cursos. La migración de los tipos anteriores para ser representados con el nuevo campo “tipo” fue sugerida pero los demás desarrolladores rechazaron la propuesta por provocar cambios en otras funcionalidades del sistema. el estado final de la base de datos se demuestra en el anexo A.4.

5.3.2.2. Desarrollo del módulo cursos del tipo seminario

Una vez diferenciado el nuevo tipo curso de los demás en la aplicación se procedió a dar soporte a las actividades CRUD para los seminarios. para esto se reutilizó la sección del sistema usada para la creación de cursos. Agregando la opción *seminario* a las opciones previas en la creación de un curso y cambiando la estructura de la forma HTML

usando Javascript, así como los datos enviados mediante el método POST y su respectiva validación en el servidor para el soporte de los seminarios. La vista se incluyó en el anexo B.3.

Una de las diferencias de los seminarios con los cursos multimedia es la forma en la que son aprobados, se decidió que podían existir dos tipos:

- Aprobar con tan solo asistir al curso.
- Aprobar asistiendo al curso y aprobando un examen presencial.

Para el soporte de estos dos tipos se agregó un campo “con_examen” que permite la representación.

El pasante aprovechó los métodos de eliminación, listado y asignación a grupos que el sistema ya poseía, juntándolos para el funcionamiento con los seminarios.

Para lograr esto debió separar visualmente los distintos tipos de cursos en las vistas de listado, para esto agregó un icono que los diferenciara en la vista de cursos del administrador (anexo B.4). Además agregó un filtro persistente en la sesión HTTP que permitiera al administrador la visualización de solo cursos del tipo seminario a lo largo de su conexión.

En el método asignación se agregó luego un *preview* con las sesiones disponibles para el seminario visibles en el apéndice B.5 además del icono para los seminarios.

5.3.2.3. Creación del usuario instructor

Para el manejo de la puntuación de las sesiones y el manejo de la asistencia se decidió crear un nuevo tipo de usuario en el sistema que pudiera apoyar al administrador en estas tareas. el usuario administrador tiene acceso a todas las funciones de un instructor en todos los cursos del sistema, mientras que el usuario solo tiene acceso a las sesiones en la que él mismo es el instructor.

Para esto se debió modificar el módulo de *login* para agregar la autenticación del nuevo usuario, que primero confirma si el usuario tiene una jerarquía más alta que la de instructor (un usuario de más rango que contenga sus funciones), en caso contrario, verifica si el usuario que intenta ingresar es instructor de algún curso.

El manejo de la autorización es realizado de forma manual, así que solo se debió popular

una estructura del tipo diccionario con las funcionalidades que se consideraran necesarias para el usuario instructor. En este caso, administrar los cursos de los que es instructor, funcionalidad desarrollada en el *sprint* cinco.

5.4. Cuarto sprint

5.4.1. Objetivos

- Desarrollar el módulo de sesiones de seminario.
- Integrar el nuevo módulo con los demás módulos del sistema.
- Agregar visualización de sesiones activas para una ubicación

En este sprint se desarrollo el módulo de gestión de las sesiones de un seminario representado en el diagrama de casos de uso (anexo A.1), la parte más importante del proyecto. Interactua con los demás módulos implementados y es la funcionalidad central que da vida al sistema. Es la información mas importante que se guarda en la base de datos, Los datos de las sesiones y la interacción de los usuarios con éstas.

Cabe destacar que al momento de mencionar sesión se hace referencia a las sesiones de los seminarios y no la sesión HTTP a menos que se especifique lo contrario.

5.4.2. Actividades

5.4.2.1. Representación en la base de datos

Para soportar la funcionalidad en la base de datos se creo la tabla llamada Sesión que contiene todos los datos relevantes para la descripción de una sesión. Es importante recalcar que las sesiones solo tienen sentido si forman parte de un curso del tipo seminario, cosa que se evidencia en la relación de agrupacion en el modelo final de la base de datos en el anexo A.4.

Un seminario puede estar compuesto por muchas sesiones, mientras que las sesiones no pueden ser compartidas entre los distintos seminarios, por lo que se representa como una relación 1 a muchos.

Una sesión se lleva a cabo en una ubicación por lo que se relacionan estas dos tablas. Una sesión puede suceder en solo una ubicación y las ubicaciones pueden albergar una cantidad de sesiones, por lo que la relación se modelo como 1 a muchos

Una sesión además es dictada/supervisada por un instructor por lo que se modelo como una relación 1 a muchos análoga a las demás.

La ultima relación es la mas interesante, las sesiones se relacionan directamente con los usuarios que asisten a ella (aprendices) y cada uno debe tener un espacio en la base de datos para modelar el resultado que este obtuvo en la misma. es una segunda relación de la tabla sesiones con la tabla Usuario que en este caso es muchos a muchos pues un usuario puede participar en cualquier número de sesiones que le hayan sido asignadas, mientras que a una sesión concurren la cantidad de usuarios que su capacidad permita. Se modelo con una tabla auxiliar que posee las dos foráneas y toda la información que la relación *sesión-usuario* necesitó al momento del desarrollo y la que pueda necesitar en el futuro, haciendo el diseño flexible, una de las metas del proyecto.

5.4.2.2. Creación del CRUD

El proceso básico de creación del CRUD fué muy parecido al CRUD de las ubicaciones, pero con más consideraciones a tener en cuenta, pues aquí se integraban distintas partes de la aplicación, como un instructor, que debía ser un usuario del sistema; una ubicación, en la que se llevaría a cabo la sesión creada, manejo de fechas, activación y desactivación de las sesiones. La vista de CRUD se presenta en el anexo B.6.

5.4.2.3. Creación de una sesión

Primero se procedió a integrar el módulo con las ubicaciones. Para poder crear una sesión debía estar creada en primer lugar una ubicación, por lo tanto se integró una entrada al módulo de ubicaciones desde el CRUD de la sesión que mantuviera en forma de *cookie* cual sesión estaba siendo modificada para volver a ella. La capacidad de la sesión no podra ser mayor a la capacidad de la locación seleccionada.

Además, una sesión no podria ser creada en una fecha en la que la ubicación seleccionada estuviera ocupada (tuviera otra sesión cuyo horario choque). Esta limitación dio pie a que se implementara una visualización de las sesiones activas de una ubicación.

Para el campo de instructor podia ser seleccionado cualquier usuario activo del sistema del cliente en el que se este realizando la operación. El cliente puede obtenerse por medio de la tabla curso, así que por normalización no se agregó información del cliente a la sesión como si se agrego a la ubicación.

Se agregaron campos que limitaban las confirmación y la cancelación de una sesión por parte de los aprendices, con el fin de dar seguridad al instructor de la cantidad asistentes a un curso.

Mas adelante en el sprint 7 se decidió la implementación de una opción para dar potestad al instructor de recibir o no los correos que notificaran las confirmaciones y cancelaciones por lo que se integro un campo para tal fin.

5.4.2.4. Actualización de una sesión

Las sesiones tiene un campo “activa” para que puedan ser desactivadas en caso de cualquier eventualidad manteniendo los datos de la sesión para poder asi ser reusados.

Las fechas de inicio y final de una sesión no pueden ser cambiadas si ya poseen usuarios confirmados, por lo que primero tendriar que se eliminados y notificados de la misma para poder realizar los cambios.

5.4.2.5. Borrado de una sesión

El borrado básico se realizó en este sprint pero luego tuvo restrisciones que surgieron del manejo de las calificaciones implementado en el sexto sprint y además se agregaron opciones de notificación a los usuarios de una sesión cancelada en el sprint siete. el borrado se puede realizar desde el listado de las sesiones en el administrador como se muestra en las opciones del anexo B.7. El usuario instructor no puede borrar la sesión, solo desactivarla.

5.4.2.6. Visualización de sesiones activas de una ubicación

Se agregó para facilitar el proceso de elegir una locación libre en la fecha de la sesión.

Para la visualización se uso un *plug-in* de calendario bootstrap-year-calendar creado usando *bootstrap* y *jquery*. *Plug-in* que permite la representación de los meses en distintos lenguajes asi como la personalización de sus colores, aspectos necesarios para el amplio

rango de necesidades de los clientes de FKC.

Este calendario se llena con AJAX a través de un *endpoint* creado para tal fin que devuelve las sesiones activas de la locación como un objeto json que luego se muestra en los días del calendario como se muestra en el anexo B.2. las sesiones desactivadas se muestran con un tipo de letra mas pequeño y en cursiva para diferenciarlas de las demás.

5.5. Quinto sprint

5to sprint

5.6. Sexto sprint

el sexto papa

5.7. Septimo sprint

el septimo papa

5.8. Octavo sprint

El octavito de carnaval

5.9. Noveno sprint

Hello father soy el nueveño

5.10. Décimo sprint

Soy yo papa el decimon