

$$\text{on}_{af_85,11,240}$$

## ÍNDICE GENERAL

<b>Introducción</b>	<b>1</b>
Antecedentes . . . . .	1
Justificación e importancia . . . . .	1
Planteamiento del problema . . . . .	1
Objetivos . . . . .	2
Objetivo general . . . . .	2
Objetivos específicos . . . . .	2
<b>1. Entorno Empresarial</b>	<b>3</b>
1.1. Fischer, Knoblauch & Co. . . . .	3
1.2. Estructura organizacional . . . . .	4
1.3. Cargo ocupado por el pasante . . . . .	4
<b>2. Marco Teórico</b>	<b>5</b>
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

<b>3. Marco Tecnológico</b>	<b>9</b>
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
3.2.5. Swift mailer . . . . .	11
3.3. Pruebas . . . . .	12
3.3.1. PHP Unit . . . . .	12
<b>4. Marco Metodológico</b>	<b>13</b>
4.1. Naturaleza del proyecto . . . . .	13
4.2. Metodología ágil . . . . .	13
4.3. <i>Scrum</i> . . . . .	14
4.3.1. Dueño del Producto o Product owner . . . . .	14
4.3.2. Equipo . . . . .	14
4.3.3. Facilitador o Scrum master . . . . .	15

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

5.2.2.2.	Creación del CRUD . . . . .	21
5.2.2.3.	Integración con google maps . . . . .	21
5.3.	Tercer sprint . . . . .	22
5.3.1.	Objetivos . . . . .	22
5.3.2.	Actividades . . . . .	22
5.3.2.1.	Análisis de los cursos implementados en el sistema . . . . .	22
5.3.2.2.	Desarrollo del módulo cursos del tipo seminario . . . . .	22
5.3.2.3.	Creación del usuario instructor . . . . .	23
5.4.	Cuarto sprint . . . . .	24
5.4.1.	Objetivos . . . . .	24
5.4.2.	Actividades . . . . .	24
5.4.2.1.	Representación en la base de datos . . . . .	24
5.4.2.2.	Creación del CRUD . . . . .	25
5.4.2.3.	Creación de una sesión . . . . .	25
5.4.2.4.	Actualización de una sesión . . . . .	26
5.4.2.5.	Borrado de una sesión . . . . .	26
5.4.2.6.	Visualización de sesiones activas de una ubicación . . . . .	26
5.5.	Quinto sprint . . . . .	27
5.5.1.	Objetivos . . . . .	27
5.5.2.	Actividades . . . . .	27
5.5.2.1.	Diseño de la interfaz . . . . .	27
5.5.2.2.	Listado de las sesiones disponibles . . . . .	28
5.5.2.3.	Confirmación y cancelación de una sesión . . . . .	28
5.5.2.4.	Exportar sesión al calendario . . . . .	29

5.5.2.5.	Integración con el módulo de mensajería interna del sistema	29
5.5.2.6.	Modal con los datos de la ubicación . . . . .	29
5.6.	Sexto sprint . . . . .	29
5.6.1.	Objetivos . . . . .	29
5.6.2.	Actividades . . . . .	30
5.6.2.1.	Mantener la integridad de la base de datos . . . . .	30
5.6.2.2.	Calificar una sesión . . . . .	30
5.6.2.3.	Actualizar la funcionalidad de borrado de una sesión . . .	31
5.6.2.4.	Generación de PDF . . . . .	31
5.6.2.5.	Módulo de seminarios como una opción . . . . .	31
5.7.	Septimo sprint . . . . .	32
5.7.1.	Objetivos . . . . .	32
5.7.2.	Actividades . . . . .	32
5.7.2.1.	Sesiones en línea en la base de datos . . . . .	32
5.7.2.2.	Presentación de las sesiones en línea . . . . .	33
5.7.2.3.	Notificaciones . . . . .	33
5.7.2.4.	Integración de las estadísticas . . . . .	34
5.8.	Octavo sprint . . . . .	34
5.8.1.	Objetivos . . . . .	34
5.8.2.	Actividades . . . . .	35
5.8.2.1.	Adaptación de la base de datos . . . . .	35
5.8.2.2.	Asignación de un seminario . . . . .	36
5.8.2.3.	Modificación de la interfaz del aprendiz . . . . .	36
5.8.2.4.	Aprobación de los módulos . . . . .	36

5.8.2.5. Integración con el calendario del usuario . . . . .	36
5.9. Noveno sprint . . . . .	37
5.9.1. Objetivos . . . . .	37
5.9.2. Actividades . . . . .	37
5.9.2.1. Migración para el soporte de caracteres UTF-8 . . . . .	37
5.9.2.2. Implantación del SGA de FKC . . . . .	38
5.9.2.3. Implantación del SGA Bibliomed . . . . .	38
5.10. Décimo sprint . . . . .	38
5.10.1. Objetivos . . . . .	38
5.10.2. Actividades . . . . .	39
5.10.2.1. Actualizar los esquemas de la base de datos de la empresa . . . . .	39
5.10.2.2. Realización del informe de pasantias . . . . .	39
5.11. Actividades extra . . . . .	39
5.11.1. Exportar preguntas de un SGA . . . . .	39
5.11.2. Migración a UTF-8 . . . . .	40
5.11.3. Funcionalidad en modulo <i>responsive</i> . . . . .	40
5.11.4. Imagenes de las vistas de login . . . . .	40
5.12. Dificultades encontradas . . . . .	41
<b>Conclusiones y Recomendaciones</b>	<b>42</b>
<b>A. Diagramas</b>	<b>42</b>
<b>B. Screenshots de los sistemas</b>	<b>47</b>

## ÍNDICE DE FIGURAS

1.1. Estructura organizacional de FKC. . . . .	4
A.1. Diagrama de casos de uso. . . . .	42
A.2. Diagrama que demuestra los distintos estados en los que se puede encontrar una sesión presencial. . . . .	43
A.3. Diagrama UML parcial de la base de datos del SGA de FKC previa al proyecto. . . . .	44
A.4. Diagrama UML parcial de la base de datos final del SGA de FKC. . . . .	45
A.5. Diagrama UML parcial de la base de datos final del SGA de Bibliomed. . .	46
B.1. Vista del listado de Ubicaciones. . . . .	47
B.2. Vista de la edición de una ubicación. . . . .	48
B.3. Vista de la creación de un seminario. . . . .	49
B.4. Vista del listado de cursos del sistema para el administrador. . . . .	50
B.5. Vista de la asignación de un seminario a un grupo. . . . .	50
B.6. Vista de la creación de una sesión de un seminario. . . . .	51
B.7. Vista del listado de las sesiones de un seminario. . . . .	52
B.8. Vista de los cursos de un instructor. . . . .	52
B.9. Archivo PDF generado con la lista de los estudiantes de un curso. . . . .	53
B.10. Vista de los cursos disponibles para el usuario aprendiz. . . . .	54
B.11. Vista de los cursos disponibles para el usuario aprendiz antes de la extensión realizada en la pasantía. . . . .	54
B.12. Vista del modal mostrado con los datos de la ubicación de la sesión correspondiente. . . . .	55



B.13.Vista de una sesión confirmada por el usuario. . . . .	55
B.14.Formato de los correos enviados por el sistema. . . . .	56
B.15.Vista de las estadísticas mostradas al administrador. . . . .	56
B.16.Vista de la adaptación hecha para la creación de seminarios en el SGA de Bibliomed. . . . .	57
B.17.Vista de cursos disponibles para el aprendiz antes de la integración de el módulo de seminarios en el SGA de Bibliomed. . . . .	58
B.18.Vista de una sesión reservada en el SGA de Bibliomed. . . . .	59

## CONCLUSIONES Y RECOMENDACIONES

La implementación e implantación de las funcionalidades de este proyecto de pasantías concluyó exitosamente y en el tiempo estipulado, el pasante se integró satisfactoriamente al equipo de programación de la empresa FKC con los que compartió durante el desarrollo. Derivando en que módulos para el manejo de seminarios; presenciales y en línea así como un módulo para el soporte de las ubicaciones donde estos seminarios se realizan fueran agregados al SGA base de FKC y al SGA de Bibliomed, uno de los clientes de la empresa. Permitiendo a los usuarios de dichos sistemas mas flexibilidad a la hora de idear planes de estudios dentro del sistema, integrando soporte para el *b-learning*.

El proyecto fue realizado bajo la metodología SCRUM, la cual permitió que su desarrollo fuese planificado, eficiente y organizado, llevando a cabo los cambios necesarios para adaptarse a los nuevos requerimientos que constantemente fueron surgiendo a lo largo del mismo. El proyecto estuvo comprendido por diez sprints en donde se fueron desarrollando las historias de usuario que surgieron del levantamiento inicial de los requerimientos. Ordenadas en un marco de desarrollo de abajo hacia arriba donde el pasante intentó contruir primerolas funcionalidades mas pequeñas, para luego construir con ellas las más grandes.

El desarrollo de los módulos se realizó bajo el patrón de arquitectura MVC, con la meta de mantener una estructura clara dentro del código que pueda ser mantenida y extendida por futuros desarrolladores. Durante la implementación del proyecto de pasantía, se utilizaron herramientas como el lenguaje de programación PHP, SQL y JavaScript para lograr los objetivos planteados.

Algunas recomendaciones surgen de la interacción del pasante con el SGA producido por FKC.

Se recomienda el uso de un manejador de versiones por parte del equipo de programación. Los beneficios podrían ser incalculables para la empresa pues este tipo de herramientas además de ser casi de uso obligado en el desarrollo de software desde hace mas de quince años traen consigo una serie de bondades difíciles de rechazar, mas aún dentro del *workflow* de FKC.

Entre estas:

- Manejo de distintos ambientes de trabajo (desarrollo, producción, prueba) y sus

respectivas versiones que pueden ser compartidas por todos los desarrolladores de la empresa.

- Mejor manejo de equipo, ofrece una interfaz centralizada para entender quien hizo que, cuando y cómo.
- Construcción de mejores estimaciones para el desarrollo de los proyectos.
- Una importante reducción del tiempo en la implantación del sistema comparandola con el servicio de archivos manualmente a través de FTP.
- Soporte de la base de código externo a la empresa, haciendola independiente del *hardware* interno.
- Eliminar la práctica de realización de *bakups* antes de cualquier cambio importante al sistema que consumen espacio de disco en los servidores.
- Idependencia de las bases de código de desarrollo y producción, donde desarrollo nuevo para el sistema no obstruya la implantación de un sistema nuevo o requiera el trabajo en una copia distinta de la base de código.

Si a todos estos beneficios le agregamos que FKC es una empresa que sirve a por lo menos quince otras, estos beneficios se multiplican, dado que existen distintas bases de código, repartidas en varios servidores a los que constantemente se realizan actualizaciones. Entre estas herramientas se encuentran GIT, Subversion, entre otras.

Se recomienda el uso de manejo de dependencias para los distintos proyectos de la empresa para el correcto manejo de los ambientes de desarrollo, entre

En tiempos actuales, donde la globalización impera el pasante considera importante el desarrollo del sistema completamente en el idioma inglés. De esta forma se amplia facilmente la cantidad de personas empleables por la empresa e incluso facilita la implementación de personal de apoyo fuera de las fronteras alemanas, dados los altos salarios que se necesitan para mantener un programador de este país.

Se recomienda una separación de los roles de diseñador y programador, haciendo uso eficiente del departamento de diseño gráfico dentro de la empresa. El pasante considera una buena práctica generar primero un diseño de como lucirá la interfaz y como seran las interacciones, para que luego la funcionalidad sea plasmada por el programador. Mezclar estos dos trabajos crea fases de espera entre el desarrollo de lo que el programador cree es

una interfaz válida y la revisión por parte del departamento gráfico, especialmente cuando no estan claros los requerimientos inicialmente.

Refiriendose a la base de código el pasante ofrece las siguientes recomendaciones:

El largo de muchos de los archivos y de muchas de las funciones dentro del sistema es excesivo, esto no complica a nuevos desarrolladores o a los programadores trabajar en conjunto en un mismo proyecto concatenado con la falta de un sistema de versiones antes mencionado y aumentando la cohesión de los mismos. De esta forma es difícil que un pequeño cambio no afecte alguna otra funcionalidad. Es además una de las primeras pistas que evidencian la mezcla de la lógica del negocio con el rol de los modelos y la interfaz dentro del código.

El código es en su mayoría no reusable,

Pruebas La implementación de pruebas a lo largo del sistema, el sistema no contiene pruebas que lo sustenten. No tenerlas trae consigo un conjunto de problemas en el desarrollo de software como: En el desarrollo de una nueva funcionalidad